
MRtrix Documentation

Release 3.0

MRtrix contributors

Sep 30, 2020

1	Getting help	3
2	Key features	5
3	Installation	7
4	Commands	9
5	Table of Contents	11
	Bibliography	381
	Index	385

MRtrix3 provides a large suite of tools for image processing, analysis and visualisation, with a focus on the analysis of white matter using diffusion-weighted MRI ([Tournier2019]). Features include the estimation of fibre orientation distributions using constrained spherical deconvolution ([Tournier2004]; [Tournier2007]; [Jeurissen2014]), a probabilistic streamlines algorithm for fibre tractography of white matter ([Tournier2012]), voxel-based analysis of apparent fibre density and fibre cross-section ([Raffelt2012]; [Raffelt2015]; [Raffelt2017]), quantitative structural connectivity analysis ([Smith2012]; [Smith2013]; [Smith2015]; [Christiaens2015]), and non-linear spatial registration of fibre orientation distribution images ([Raffelt2011]). *MRtrix3* also offers comprehensive visualisation tools in *mrview*.

These applications have been written from scratch in C++, using the functionality provided by *Eigen*, and *Qt*. The software is currently capable of handling DICOM, NIfTI and AnalyseAVW image formats, amongst others. The source code is distributed under the [Mozilla Public License](#).

Use of the *MRtrix3* software package in published works should be accompanied by the following citation:

J.-D. Tournier, R. E. Smith, D. Raffelt, R. Tabbara, T. Dhollander, M. Pietsch, D. Christiaens, B. Jeurissen, C.-H. Yeh, and A. Connelly. *MRtrix3*: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 202 (2019), pp. 116–37.

Tip: Make sure to use the version of this documentation that matches your version of this software. You can select the version on the lower left of this page.

CHAPTER 1

Getting help

There are a variety of sources of help and information to bring you up to speed with *MRtrix3*. These include:

- the [main MRtrix3 documentation](#) (these pages);
- our [Introduction to the Unix command-line](#) if you're unfamiliar with the terminal (though you'll readily find plenty of excellent tutorials online);
- our [Community Forum](#) for support and general discussion about the use of *MRtrix3* – you can address all *MRtrix3*-related queries there, using your GitHub or Google login to post questions.
- our [Frequently Asked Questions](#), hosted as a user-editable `wiki` category within our forum.

CHAPTER 2

Key features

While *MRtrix3* is primarily intended to be used for the analysis of diffusion MRI data, at its fundamental level it is designed as a general-purpose library for the analysis of *any* type of MRI data. As such, it provides a back-end to simplify a large number of operations, many of which will be invisible to the end-user. Specifically, *MRtrix3* features:

- a consistent *command-line interface*, with inline documentation for each command;
- universal import/export capabilities when *accessing image data* across all *MRtrix3* applications;
- *Multi-file numbered image support* to load multiple images as a single multi-dimensional dataset;
- efficient use of *Unix Pipelines* for complex workflows;
- high performance on modern multi-core systems, with multi-threading used extensively throughout *MRtrix3*;
- available on all common modern operating systems (GNU/Linux, MacOSX, Windows);
- a consistent *Coordinate system* with most operations performed in scanner/world coordinates where possible.

CHAPTER 3

Installation

MRtrix3 runs on GNU/Linux, macOS, Microsoft Windows platforms, and other Unix platforms. For most users, the simplest way to install *MRtrix3* is to use one of the pre-compiled packages. For details, please refer [the main MRtrix website](#).

If the precompiled packages are not available, we provide specific instructions for building the software from source. This is normally a simple process, but does require more compute resources and expertise. See the [relevant pages](#) for details.

CHAPTER 4

Commands

The *MRtrix3* software package includes a suite of tools for image analysis and visualisation. With the exception of *mrview* and *shview*, all *MRtrix3* executables are designed to be run via a terminal using a consistent *command-line interface*. While many of the tools and features are discussed within tutorials found in this documentation, a comprehensive *List of MRtrix3 commands* can be found in the reference section. These lists provide links to the help page (manual) for each executable, which can also be accessed by typing the `-help` option after the executable name on the terminal.

Tip: Some proficiency with the Unix command-line is required to make the best use of this software. There are many resources online to help you get started if you are not already familiar with it. We also recommend our own *Introduction to the Unix command-line*, which was written with a particular focus on the types of use that are common when using *MRtrix3*.

5.1 Before you install

5.1.1 Acknowledging this work

If you wish to include results generated using the *MRtrix3* package in a publication, please include a line such as the following to acknowledge the work of our developers:

- Processing was performed using the MRtrix3 package (Tournier et al., 2019).
J.-D. Tournier, R. E. Smith, D. Raffelt, R. Tabbara, T. Dhollander, M. Pietsch, D. Christiaens, B. Jeurissen, C.-H. Yeh, and A. Connelly. *MRtrix3*: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 202 (2019), pp. 116–37.

Note: Many individual methods included in the MRtrix3 software have been published in scientific journals and should be cited as such. Please check the references listed on the specific *application's page* to ensure the appropriate reference is included, so that the scientists behind all methods receive proper acknowledgement.

5.1.2 Warranty

The software described in this manual has no warranty, it is provided “as is”. It is your responsibility to validate the behavior of the routines and their accuracy using the source code provided, or to purchase support and warranties from commercial redistributors. Consult the [Mozilla Public License](#) for further details.

5.1.3 License

MRtrix is free software: you can redistribute it and/or modify it under the terms of the [Mozilla Public License](#) as published by the [Free Software Foundation](#), either version 2 of the License, or (at your option) any later version.

MRtrix is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the [Mozilla Public License](#) for more details. You should have received a copy of [Mozilla Public License](#) along with MRtrix. If not, see <http://mozilla.org/MPL/2.0/>.

Tip: Some proficiency with the Unix command-line is required to make the best use of this software. There are many resources online to help you get started if you are not already familiar with it. We also recommend our own [Introduction to the Unix command-line](#), which was written with a particular focus on the types of use that are common when using *MRtrix3*.

5.2 Installing *MRtrix3*

By far the quickest and simplest way to install *MRtrix3* is to use one of the pre-compiled packages. These allow you to get started straight away, with all dependencies included.

Packages of various types are available for the three main platforms (GNU/Linux, macOS, Microsoft Windows). For the latest information, please refer to [the main MRtrix website](#).



5.3 Building *MRtrix3* from source

The instructions below describe the process of compiling and installing *MRtrix3* from source. Please consult the [MRtrix3 forum](#) if you encounter any issues.

Warning: These instructions are for more advanced users who wish to install very specific versions of *MRtrix3*, or make their own modifications. Most users will find it much easier to install one of the [pre-compiled packages](#) available for their platform from the [main MRtrix3 website](#).

5.3.1 Install Dependencies

To install *MRtrix3*, you will need to have a number of dependencies available on your system, listed below. These can be installed in a number of ways, depending on your specific platform. We provide specific instructions for doing so for GNU/Linux, macOS and Microsoft Windows in the subsequent sections.

Required dependencies:

- a C++11 compliant compiler (GCC version ≥ 5 , clang);
- Python version ≥ 2.7 (≥ 3 strongly recommended due to [deprecation of Python2](#));
- The [zlib](#) compression library;
- [Eigen](#) version ≥ 3.2 (≥ 3.3 recommended);

- **Qt** version ≥ 5.5 [GUI components only];

and optionally:

- **libTIFF** version ≥ 4.0 (for TIFF support);
- **FFTW** version ≥ 3.0 (for improved performance in certain applications, currently only mrdegibbs);
- **libpng** (for PNG support).

The instructions below list the most common ways to install these dependencies on Linux, macOS, and Windows platforms.

Warning: To run the GUI components of *MRtrix3* (*mrview* & *shview*), you will also need an **OpenGL 3.3** compliant graphics card and corresponding software driver.

Note that this implies you *cannot run the GUI components over a remote X11 connection*, since it can't support OpenGL 3.3+ rendering. The most up-to-date recommendations in this context can be found in the [relevant Wiki entry](#) on the [MRtrix3 community forum](#).

Linux

The installation procedure will depend on your system. Package names may change between distributions, and between different releases of the same distribution. We provide commands to install the required dependencies on some of the most common Linux distributions below.

Warning: The commands below are **suggestions** based on what has been reported to work in the past, but may need to be tailored for your specific distribution. See below for hints on how to proceed in this case.

- Ubuntu Linux (and derivatives, e.g. Linux Mint):

```
sudo apt-get install git g++ python libeigen3-dev zlib1g-dev libqt5opengl5-dev
↳ libqt5svg5-dev libgl1-mesa-dev libfftw3-dev libtiff5-dev libpng-dev
```

Note: On Ubuntu 20.04 and newer, you'll need to replace `python` in the line above with `python-is-python3` (or `python-is-python2` if you're still using version 2.7, which is now *very* deprecated).

- RPM-based distros (Fedora, CentOS):

```
sudo yum install git g++ python eigen3-devel zlib-devel libqt5-devel libgl1-mesa-
↳ devel fftw-devel libtiff-devel libpng-devel
```

On Fedora 24, this is reported to work:

```
sudo yum install git gcc-c++ python eigen3-devel zlib-devel qt-devel mesa-libGL-
↳ devel fftw-devel libtiff-devel libpng-devel
```

- Arch Linux:

```
sudo pacman -Syu git python gcc zlib eigen qt5-svg fftw libtiff libpng
```

You may find that your package installer is unable to find the packages listed, or that the subsequent steps fail due to missing dependencies (particularly the `./configure` command). In this case, you will need to search the package database and find the correct names for these packages:

- `git`;
- an appropriate C++ compiler (e.g. GCC 5 or above, or clang);
- Python version ≥ 2.7 (version ≥ 3.0 strongly recommended);
- the `zlib` compression library and its corresponding development header/include files;
- the Eigen template library (only consists of development header/include files);
- Qt version ≥ 5.5 , its corresponding development header/include files, and the executables required to compile the code. Note that this can be broken up into several packages, depending on how your distribution has chosen to distribute this. You will need to get those that provide these Qt modules: Core, GUI, OpenGL, SVG, and the `qmake`, `rcc` & `moc` executables (note these will probably be included in one of the other packages);
- [optional] the TIFF library and utilities version ≥ 4.0 , and its corresponding development header/include files;
- [optional] the FFTW library version ≥ 3.0 , and its corresponding development header/include files;
- [optional] the PNG library and its corresponding development header/include files.

Warning: Compilers included in older distributions, e.g. Ubuntu 12.04, may not be capable of compiling *MRtrix3*, as it requires C++11 support. A solution is to install a newer compiler as provided by an optional addon package repository, e.g. the [Ubuntu toolchain PPA](#). Once the relevant repository has been added to the distribution's package manager, you'll need to update the local list of available packages (e.g. `sudo apt-get update`), followed by explicit installation of the newer version of the compiler (e.g. `sudo apt-get install g++-7`).

Note: In many instances where *MRtrix3* dependencies are installed in some non-standard fashion, the *MRtrix3* `configure` script will not be able to automatically identify the location and/or appropriate configuration of those dependencies. In such cases, the *MRtrix3* `configure` script provides a range of environment variables that can be set by the user in order to provide this information. Executing `configure -help` provides a list of such environment variables; in addition, if the script is unable to detect or utilise a particular dependency properly, it will also provide a suggestion of which environment variable may need to be set in a manner tailored for your particular system in order to provide it with the information it needs to locate that dependency.

See also:

If for whatever reasons you need to install *MRtrix3* on a system with older dependencies, and you are unable to update the software (e.g. you want to run *MRtrix3* on a centrally-managed HPC cluster), you can as a last resort use the [procedures described in this community forum post](#).

macOS

1. Update macOS to version 10.10 (Yosemite) or higher (OpenGL 3.3 will typically not work on older versions);
2. Install XCode from the [App Store](#);
3. Install Eigen3 and Qt5.

There are several alternative ways to do this, depending on your current system setup. The most convenient is probably to use your favorite package manager ([Homebrew](#) or [MacPorts](#)), or install one of these if you haven't already.

If you find your first attempt doesn't work, *please* resist the temptation to try one of the other options: in our experience, this only leads to further conflicts, which won't help installing *MRtrix3* and will make things more difficult to fix later. Once you pick one of these options, we strongly recommend you stick with it, and consult the [community forum](#) if needed for advice and troubleshooting.

- With [Homebrew](#):
 - Install Eigen3: `brew install eigen`
 - Install Qt5: `brew install qt5`
 - Install pkg-config: `brew install pkg-config`
 - Add Qt's binaries to your path: `export PATH=`brew --prefix`/opt/qt5/bin:$PATH`
- With [MacPorts](#):
 - Install Eigen3: `port install eigen3`
 - Install Qt5: `port install qt5`
 - Install pkg-config: `port install pkgconfig`
 - Add Qt's binaries to your path: `export PATH=/opt/local/libexec/qt5/bin:$PATH`
- As a last resort, you can manually install Eigen3 and Qt5: You can use this procedure if you have good reasons to avoid the other options, or if for some reason you cannot get either [Homebrew](#) or [MacPorts](#) to work.
 - Install Eigen3: download and extract the source code from eigen.tuxfamily.org
 - Install Qt5: download and install the latest version from http://download.qt.io/official_releases/qt/
 You need to select the file labelled `qt-opensource-mac-x64-clang-5.X.X.dmg`. You can choose to install it system-wide or just in your home folder, whichever suits; just remember where you installed it.
 - Make sure Qt5 tools are in your PATH (edit as appropriate): `export PATH=/path/to/Qt5/5.X.X/clang_64/bin:$PATH`
 - Set the CFLAG variable for Eigen (edit as appropriate): `export EIGEN_CFLAGS="-isystem /where/you/extracted/eigen"` Make sure *not* to include the final `/Eigen` folder in the path name: use the folder in which it resides instead!

4. Install TIFF, FFTW and PNG libraries.

- With [Homebrew](#):
 - Install TIFF: `brew install libtiff`
 - Install FFTW: `brew install fftw`
 - Install PNG: `brew install libpng`
- With [MacPorts](#):
 - Install TIFF: `port install tiff`
 - Install FFTW: `port install fftw-3`
 - Install PNG: `port install libpng`

Windows

All of these dependencies are installed below by the MSYS2 package manager.

Warning: When following the instructions below, use the ‘**MinGW-w64 Win64 shell**’; ‘MSYS2 shell’ and ‘MinGW-w64 Win32 shell’ *must* be avoided, as they will yield erroneous behaviour that is difficult to diagnose if used accidentally.

Warning: At time of writing, this MSYS2 system update will give a number of instructions, including: terminating the terminal when the update is completed, and modifying the shortcuts for executing the shell(s). Although these instructions are not as prominent as they could be, it is *vital* that they are followed correctly!

1. Download and install the most recent 64-bit MSYS2 installer from <http://msys2.github.io/> (msys2-x86_64-*.exe), and following the installation instructions from the [MSYS2 wiki](#).
2. Run the program ‘MinGW-w64 Win64 Shell’ from the start menu.
3. Update the system packages, as per the instructions:

```
pacman -Syuu
```

Close the terminal, start a new ‘MinGW-w64 Win64 Shell’, and repeat as necessary until no further packages are updated.

4. From the ‘MinGW-w64 Win64 Shell’ run:

```
pacman -S git python pkg-config mingw-w64-x86_64-gcc mingw-w64-x86_64-eigen3_
↪mingw-w64-x86_64-qt5 mingw-w64-x86_64-fftw mingw-w64-x86_64-libtiff mingw-w64-
↪x86_64-libpng
```

Sometimes pacman may fail to find a particular package from any of the available mirrors. If this occurs, you can download the relevant package from [SourceForge](#): place both the package file and corresponding .sig file into the /var/cache/pacman/pkg directory, and repeat the pacman call above.

Sometimes pacman may refuse to install a particular package, claiming e.g.:

```
error: failed to commit transaction (conflicting files)
mingw-w64-x86_64-eigen3: /mingw64 exists in filesystem
Errors occurred, no packages were upgraded.
```

Firstly, if the offending existing target is something trivial that can be deleted, this is all that should be required. Otherwise, it is possible that MSYS2 may mistake a *file* existing on the filesystem as a pre-existing *directory*; a good example is that quoted above, where pacman claims that directory /mingw64 exists, but it is in fact the two files /mingw64.exe and /mingw64.ini that cause the issue. Temporarily renaming these two files, then changing their names back after pacman has completed the installation, should solve the problem.

5.3.2 Git setup

If you intend to contribute to the development of *MRtrix3*, set up your git environment as per the [Git instructions page](#)

5.3.3 Build *MRtrix3*

1. Clone the *MRtrix3* repository:

```
git clone https://github.com/MRtrix3/mrtrix3.git
```

or if you have set up your SSH keys (for contributors):

```
git clone git@github.com:MRtrix3/mrtrix3.git
```

2. Configure the *MRtrix3* install:

```
cd mrtrix3
./configure
```

If this does not work, examine the ‘configure.log’ file that is generated by this step, it may give clues as to what went wrong.

3. Build the binaries:

```
./build
```

5.3.4 Set up *MRtrix3*

1. Update the shell startup file, so that the locations of *MRtrix3* commands and scripts will be added to your PATH environment variable.

If you are not familiar or comfortable with modification of shell files, *MRtrix3* now provides a convenience script that will perform this setup for you (assuming that you are using `bash` or equivalent interpreter). From the top level *MRtrix3* directory, run the following:

```
./set_path
```

2. Close the terminal and start another one to ensure the startup file is read (or just type ‘bash’)
3. Type `mrview` to check that everything works
4. You may also want to have a look through the [List of *MRtrix3* configuration file options](#) and set anything you think might be required on your system.

Note: The above assumes that your shell will read the `~/.bashrc` file at startup time. This is not always guaranteed, depending on how your system is configured. If you find that the above doesn’t work (e.g. typing `mrview` returns a ‘command not found’ error), try changing step 1 to instruct the `set_path` script to update PATH within a different file, for example `~/.bash_profile` or `~/.profile`, e.g. as follows:

```
./set_path ~/.bash_profile
```

5.3.5 Keeping *MRtrix3* up to date

1. You can update your installation at any time by opening a terminal in the *MRtrix3* folder, and typing:

```
git pull
./build
```

2. If this doesn’t work immediately, it may be that you need to re-run the configure script:

```
./configure
```

and re-run step 1 again.

5.4 Deploying MRtrix3

The installation instructions provided in the preceding pages produce a working install for the current user only. There are many advantages to this:

- no need for admin privileges, either for the initial install (beyond installation of dependencies), or any subsequent updates;
- users are in control of the precise version of MRtrix3 they are using for their specific projects - no system updates will interfere with their study.

However, system administrators and software distributors will want to install *MRtrix3* in a system-wide location to make it accessible to all users; and/or to deploy it to other systems without requiring a full rebuild. While *MRtrix3* does not provide an explicit command to do this, it is a trivial process:

- build the code
- copy the `bin/`, `lib/` and `share/` folders together to the desired target location
- set the `PATH` to point to the `bin/` folder.

This can be done any number of ways. The only requirement is that these 3 folders are co-located alongside each other, so that the executables can find the *MRtrix3* shared library, and the scripts can find the requisite python modules.

Note also that this structure is broadly compatible with the [Linux Filesystem Hierarchy Standard](#). It should be perfectly possible to merge the *MRtrix3* `bin/`, `lib/` and `share/` folders with the system's existing equivalent locations in `/usr/` or `/usr/local/` if desired, in which case there would be no need to explicitly set the `PATH` (assuming `/usr/bin` or `/usr/local/bin/` are already in the `PATH`). However, there is no *requirement* that it be installed anywhere in particular, and we expect most sysadmins will prefer to place them in a separate location to minimise any chance of conflict.

Below we provide step-by-step instructions for creating a single tar file that can then be copied to other systems and extracted in the desired folder:

1. Obtain, configure and build the desired version of *MRtrix3*:

```
$ git checkout http://github.com/MRtrix3/mrtrix3.git
$ cd mrtrix3
$ ./configure
$ ./build
```

2. Collate the relevant folders and their contents into a single archive file:

```
$ tar cvfz mrtrix3.tgz bin/ lib/ share/
```

3. Copy the resulting `mrtrix3.tgz` file over to the target system, into a suitable location., for example (as root):

```
$ mkdir /usr/local/mrtrix3
$ cp mrtrix3.tgz /usr/local/mrtrix3/
```

4. Extract the archive in this location (as root):

```
$ cd /usr/local/mrtrix3/
$ tar xvfz mrtrix3.tgz
```

Assuming no errors were generated, you can safely remove the `mrtrix3.tgz` file at this point.

5. Add the newly-extracted `bin/` folder to the `PATH`, e.g.:

```
$ export PATH=/usr/local/mrtrix3/bin:$PATH
```

At which point *MRtrix3* command should be available to the corresponding user.

Note that the above command will only add *MRtrix3* to the `PATH` for the current shell session. You would need to add the equivalent line to your users' startup scripts, using whichever mechanism is appropriate for your system.

5.5 HPC clusters installation

These instructions outline a few issues specific to high-performance computing (HPC) systems.

5.5.1 Installing *MRtrix3*

Most HPC clusters will run some flavour of GNU/Linux and hence a cluster administrator should be able to follow the steps outlined to *Building MRtrix3 from source*. In particular, if your sysadmin is able to install the required dependencies (the preferred option), you should be able to subsequently *Build MRtrix3*.

However, it is not uncommon for HPC systems to run stable, and hence relatively old distributions, with outdated dependencies. This is particularly problematic since *MRtrix3* relies on recent technologies (C++11, OpenGL 3.3), which are only available on recent distributions. There is therefore a good chance these dependencies simply cannot be installed (certainly not without a huge amount of effort on the part of your sysadmin). In such cases, one can instead attempt a *standalone installation* <<https://community.mrtrix.org/t/standalone-installation-on-linux/3549>>. Alternatively, if you (and your sysadmin) are comfortable with installation of dependencies from source within your home directory, you can try the [instructions suggested on the wiki section of the MRtrix3 community forum](#).

5.5.2 Remote display

Most people would expect to be able to run `mrview` on the server using X11 forwarding. Unfortunately, this will not work without some effort. The most up-to-date details on attempts to use `mrview` in this way can be found in the [relevant Wiki entry of the MRtrix3 community forum](#).

5.5.3 Configuration

There are a number of parameters that can be set in the configuration file that are highly relevant in a HPC environment, particularly when the user's home folder is stored over a network-based filesystem (as is often the case). The *MRtrix3* configuration file is located either system-wide in `/etc/mrtrix.conf`, and/or in each user's home folder in `~/.mrtrix.conf`. Entries consist of `key: value` entries, one per line, stored as ASCII text.

- **NumberOfThreads** (default: `hardware concurrency`, as reported by the system): by default, *MRtrix3* will use as many threads as the system reports being able to run concurrently. You may want to change that number to a lower value, to prevent *MRtrix3* from taking over the system entirely. This is particularly true if you anticipate many users running many *MRtrix3* commands concurrently.
- **TmpFileDir** (default: `'/tmp'`): any image data passed from one *MRtrix3* command to the next using a Unix pipeline is actually stored in a temporary file, and its filename passed to the next command. While this is fine if the filesystem holding the temporary file is locally backed and large enough, it can cause significant slowdown and bottlenecks if it resides on a networked filesystem, as the temporary file will most likely need

to be transferred in its entirety over the network and back again. Also, if the filesystem is too small, *MRtrix3* commands may abort when processing large files. In general, the `/tmp` folder is likely to be the most appropriate (especially if mounted as `tmpfs`). If however it is not locally mounted, or too small, you may want to set this folder to some other more suitable location.

- **TrackWriterBufferSize** (default: 16777216). When writing out track files, *MRtrix3* will buffer up the output and write out in chunks of 16MB, to limit the frequency of `write()` calls and the amount of IO requests. More importantly, when several instances of *MRtrix3* are generating tracks concurrently and writing to the same filesystem, frequent small writes will result in massive fragmentation of the output files. By setting a large buffer size, the chances of writes being concurrent is reduced drastically, and the output files are much less likely to be badly fragmented. Note that fragmentation can seriously affect the performance of subsequent commands that need to read affected data. Depending on the type of operations performed, it may be beneficial to use larger buffer sizes, for example 256MB. Note that larger numbers imply greater RAM usage to hold the data prior to write-out, so it is best to keep this much smaller than the total RAM capacity.

5.6 Beginner DWI tutorial

Tip: Some proficiency with the Unix command-line is required to make the best use of this software. There are many resources online to help you get started if you are not already familiar with it. We also recommend our own [Introduction to the Unix command-line](#), which was written with a particular focus on the types of use that are common when using *MRtrix3*.

Warning: This tutorial is not intended to show the optimal or even recommended way of processing. It is merely a simplified example, intended to familiarise the user with the typical command line interface of certain basic processing steps.

This tutorial will hopefully provide enough information for a novice user to get from the raw DW image data to performing some streamlines tractography. It may also be useful for experienced MRtrix users in terms of identifying some of the new command names.

For all *MRtrix3* scripts and commands, additional information on the command usage and available command-line options can be found by invoking the command with the `-help` option. Note that this tutorial includes commands and scripts for which there are relevant journal articles for citation; these are listed on the help pages also.

5.6.1 DWI geometric distortion correction

If the user has access to reversed phase-encode spin-echo image data, this can be used to correct the susceptibility-induced geometric distortions present in the diffusion images, as well as any eddy current-induced distortions and inter-volume subject motion. Procedures for this correct are not yet implemented in *MRtrix3*, though we do provide a script for interfacing with the relevant FSL tools:

```
dwifslpreproc <Input DWI series> <Output corrected DWI series> [options]
```

For more details, see the `dwifslpreproc` help file. In particular, it is necessary to manually specify what type of reversed phase-encoding acquisition has taken place (if any), and potentially provide additional relevant input images or provide details of the phase encoding scheme used in the acquisition.

5.6.2 DWI brain mask estimation

In previous versions of MRtrix, a heuristic was used to derive this mask; a dedicated command is now provided:

```
$ dwi2mask <Input DWI> <Output mask>
$ mrview <Input DWI> -roi.load <Output mask>
```

Note that if you are working with ex-vivo data, this command will likely not give the desired results. It can also give inconsistent results in cases of low SNR, strong B1 bias field, or even with good-quality images; it is recommended that the output of this command should *always* be checked (and corrected if necessary) before proceeding with further processing.

5.6.3 Response function estimation

To perform spherical deconvolution, the DWI signal emanating from a single coherently-oriented fibre bundle must be estimated. We provide a script for doing this, which has *a range of algorithms and parameters*. This example will use fairly sensible defaults:

```
$ dwi2response tournier <Input DWI> <Output response text file>
$ shview <Output response text file>
```

5.6.4 Fibre Orientation Distribution estimation

This command performs Constrained Spherical Deconvolution (CSD) based on the response function estimated previously.

```
$ dwi2fod csd <Input DWI> <Input response text file> <Output FOD image> -mask <Input_
↪DWI mask>
$ mrview <Input DWI> -odf.load_sh <Output FOD image>
```

5.6.5 Whole-brain streamlines tractography

For the sake of this tutorial, we will perform whole-brain streamlines tractography, using default reconstruction parameters.

```
$ tckgen <Input FOD image> <Output track file> -seed_image <Input DWI mask> -mask
↪<Input DWI mask> -select <Number of tracks>
$ mrview <Input DWI> -tractography.load <Output track file>
```

Note: Loading a very large number of tracks can inevitably make the `mrview` software run very slowly. When this occurs, it may be preferable to instead view only a subset of the generated tracks, e.g.:

```
$ tckedit <Track file> <Smaller track file> -number <Smaller number of tracks>
$ mrview <Input DWI> -tractography.load <Smaller track file>
```

5.6.6 Track Density Imaging (TDI)

TDI can be useful for visualising the results of tractography, particularly when a very large number of streamlines is generated.

```
$ tckmap <Input track file> <Output TDI> -vox <Voxel size in mm>
$ mrview <Output TDI>
```

5.7 Images and other data

5.7.1 Image format handling in *MRtrix3*

MRtrix3 provides a flexible data input/output back-end in the shared library, which is used across all applications. This means that all applications in *MRtrix3* can read or write images in all the supported formats - there is no need to explicitly convert the data to a given format prior to processing.

However, some specialised applications may expect additional information to be present in the input image. The MRtrix .mif/.mih formats are both capable of storing such additional information data in their header, and will hence always be supported for such applications. Most image formats however cannot carry additional information in their header (or at least, not easily) - this is in fact one of the main motivations for the development of the MRtrix image formats. In such cases, it would be necessary to use MRtrix format images. Alternatively, it may be necessary to provide the additional information using command-line arguments (this is the case particularly for the DW gradient table, when providing DWI data in NIfTI format for instance).

Image file formats are recognised by their file extension. One exception to this is DICOM: if the filename corresponds to a folder, it is assumed to contain DICOM data, and the entire folder will be scanned recursively for DICOM images.

It is also important to note that the name given as an argument will not necessarily correspond to an actual file name on disk: in many cases, images may be split over several files. What matters is that the text string provided as the *image specifier* is sufficient to unambiguously identify the full image.

Coordinate system

All *MRtrix3* applications will consistently use the same coordinate system, which is identical to the NIfTI standard. Note that this frame of reference differs from the DICOM standard (typically the x & y axis are reversed). The convention followed by *MRtrix3* applications is as follows:

dimensional	description
0 (x)	increasing from left to right
1 (y)	increasing from posterior to anterior
2 (z)	increasing from inferior to superior

All coordinates or vector components supplied to *MRtrix3* applications should be provided with reference to this coordinate system.

Multi-file numbered image support

It is possible to access a numbered series of images as a single multi-dimensional dataset, using a syntax specific to MRtrix. For example:

```
$ mrinfo MRI-volume-['[]'.nii.gz
```

will collate all images that match the pattern MRI-volume-<number>.nii.gz, sort them in ascending numerical order, and access them as a single dataset with dimensionality one larger than that contained in the images. In other words, assuming there are MRI-volume-0.nii.gz to MRI-volume-9.nii.gz, and each volume is a 3D image, the result will be a 4D dataset with 10 volumes.

Note that this isn't limited to one level of numbering:

```
$ mrconvert data-'[]'-'[]'.nii combined.mif
```

will collate all images that match the `data-number-number.nii` pattern and generate a single dataset with dimensionality two larger than its constituents.

Finally, it is also possible to explicitly request specific numbers, using *Number sequences* within the square brackets:

```
$ mrconvert data-'[10:20]'.nii combined.mif
```

Note that the single quotation marks surrounding the square brackets are required for shells that treat `[` within a string as a special character (such as `zsh`). See our [command-line tutorial](#) for more information on special character escaping.

Data types

MRtrix3 applications can read and write data in any of the common data types. Many *MRtrix3* commands also support the `-datatype` option to specify the data type for the output image. For example:

```
$ mrconvert DICOM_images/ -datatype float32 output.nii
```

Note: Not all image formats support all possible datatypes. The MRtrix image file formats are designed to handle all of the possibilities listed below, while other image formats may only support a subset. When a data type is requested that isn't supported by the image format, a hopefully suitable alternative data type will be used instead.

Below is a list of the supported data types and their specifiers for use on the command-line. Note that *MRtrix* is not sensitive to the case of the specifier: `uint16le` will work just as well as `UInt16LE`.

Specifier	Description
Bit	bitwise data
Int8	signed 8-bit (char) integer
UInt8	unsigned 8-bit (char) integer
Int16	signed 16-bit (short) integer (native endian-ness)
UInt16	unsigned 16-bit (short) integer (native endian-ness)
Int16LE	signed 16-bit (short) integer (little-endian)
UInt16LE	unsigned 16-bit (short) integer (little-endian)
Int16BE	signed 16-bit (short) integer (big-endian)
UInt16BE	unsigned 16-bit (short) integer (big-endian)
Int32	signed 32-bit int (native endian-ness)
UInt32	unsigned 32-bit int (native endian-ness)
Int32LE	signed 32-bit int (little-endian)
UInt32LE	unsigned 32-bit int (little-endian)
Int32BE	signed 32-bit int (big-endian)
UInt32BE	unsigned 32-bit int (big-endian)
Float32	32-bit floating-point (native endian-ness)
Float32LE	32-bit floating-point (little-endian)
Float32BE	32-bit floating-point (big-endian)
Float64	64-bit (double) floating-point (native endian-ness)
Float64LE	64-bit (double) floating-point (little-endian)
Float64BE	64-bit (double) floating-point (big-endian)
CFloat32	complex 32-bit floating-point (native endian-ness)
CFloat32LE	complex 32-bit floating-point (little-endian)
CFloat32BE	complex 32-bit floating-point (big-endian)
CFloat64	complex 64-bit (double) floating-point (native endian-ness)
CFloat64LE	complex 64-bit (double) floating-point (little-endian)
CFloat64BE	complex 64-bit (double) floating-point (big-endian)

The image transform

The orientation of the image with respect to the scanner axes is determined by the combination of the *image axes* and the *location of the corner voxel*. This information is encapsulated in the *transformation matrix*, commonly referred to simply as the *transform*. You can view the transform for any image using *mrinfo*, for example:

```
$ mrinfo dwi.mif
*****
Image:                "dwi.mif"
*****
Dimensions:          104 x 104 x 54 x 167
Voxel size:           2.30769 x 2.30769 x 2.3 x ?
Data strides:         [ -1 -2 3 4 ]
Format:               MRtrix
Data type:            unsigned 16 bit integer (little endian)
Intensity scaling:    offset = 0, multiplier = 1
Transform:
                    0.9999    6.887e-09    -0.01564    -116.1
                   -0.001242    0.9968    -0.07943    -89.44
                    0.01559    0.07944    0.9967    -64.27
Comments:             TOURNIER DONALD (BRI) [MR] diff60_b3000_2.3_iPat2+ADC
                    study: BRI_Temp_backup Donald
                    DOB: 09/03/1977
                    DOS: 03/10/2007 15:58:40
dw_scheme:            [ 167 entries ]
```

The ‘Transform’ field above shows the first 3 rows of the transformation matrix (technically, this is a 4×4 matrix, but the last row is always set to $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$). The first 3 columns correspond to the x , y & z image axes respectively, while the last column corresponds to the location *in real (scanner/world) space* of the corner voxel (i.e. the voxel at index $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$).

In *MRtrix3*, the transform shown always corresponds to the transformation from image coordinates *in millimeters* to scanner coordinates *in millimeters* - the voxel size is not taken into account, and the image axes are always normalised to unit amplitude. This may differ from other packages.

Furthermore, *MRtrix3* will always present the transform that best matches the real space. If the transform of the image on file represents a large rotation, such that for example the first image axis is closer to the scanner’s z axis, this transform will be modified by permutation or inversion of the axes to bring it in alignment with the expected coordinate system, so that the first axis genuinely can be interpreted as approximately left-right, etc. To achieve this, *MRtrix3* will also modify the image *Strides* to match.

Strides

A file is simply a linear array of values. Image data on the other hand are multidimensional arrays. The image values can therefore be ordered on file in many different ways. For example, we could start from the voxel at the left posterior inferior corner of the image, and store intensity values in order of traversal towards the *right*. Once the other end of the image is reached, we repeat the process for the row of values *anterior* to the last one, and repeat until the end of the slice. At this point, we store the slice *superior* to the last one, until the whole image has been stored. This ordering scheme is what is typically used in the NIfTI standard, and is commonly referred to as RAS (right anterior posterior), referring to the direction of traversal of each axis in turn. This scheme is also often referred to as *neurological*, although this term is in general much more ambiguous.

However, this is only a convention, and many other combinations are possible. For instance, it is possible to start from the *right* posterior inferior corner, and raster through along the *left* direction, then store the next row along the anterior direction, and finally the next slice in the superior direction. This scheme is what is normally used in the now deprecated Analyse format, and is commonly referred to as LAS or *radiological*.

Of course, there are many more possibilities. For instance, sagittal DICOM images will typically be stored using a PIR (posterior inferior right) order, since each sagittal slice is stored in order, etc. *MRtrix3* applications are agnostic to the order of storage, and can handle any such images provided the format is clear about what the order is.

In *MRtrix3*, the order of storage is defined by their *strides*. These refer to the number of voxels between a given voxel and the next voxel along a given dimension. For instance, in a 128×128×128 image stored using RAS ordering, the strides would be 1, 128, 16384: the next voxel along the x axis is just one voxel away, while the next voxel along the y axis is 128 values away (i.e. a whole row of x values), and so on. In contrast, if stored in LAS order, the strides would be -1, 128, 16384, indicating that the next voxel along the x axis would actually be stored one value *before* the current one.

To simplify the specification of these strides, *MRtrix3* typically expects and provides *symbolic* strides. For example, the RAS strides above would be expressed as 1, 2, 3, since this is sufficient to deduce the actual strides once the image dimensions are known. Likewise, LAS would correspond to strides of -1, 2, 3, PIR to 3, -1, -2, etc. This has the advantage that the specification of the strides is then independent of the image dimensions.

Using strides to specify ordering also allows the specification to generalise to arbitrary dimensions. For example, it is fairly common for *MRtrix3* applications to request their output for 4D images to be written with strides 2, 3, 4, 1 (if the image format supports it): this corresponds to a volume-contiguous order, whereby the values for all volumes of a given voxel are written next to each other on file; this often has performance advantages for applications that need to process all values for a given voxel concurrently (as is often the case in diffusion MRI), by allowing the hardware to make better use of resources (tractography is one such example).

Many *MRtrix3* commands accept the `-strides` option, which is used to specify the strides for the output image. For example, to generate a LAS (radiological) NIfTI image for use with FSL (along with the corresponding bvecs/bvals), you can use *mrconvert* along with the `-strides -1, 2, 3, 4` option:

```
$ mrconvert dwi.mif -strides -1,2,3,4 -export_grad_fsl bvecs bvals dwi.nii
```

Likewise, if you need to ensure the orientation is neurological (RAS), you can specify strides 1, 2, 3, 4 (or use the 1:4 shorthand). You can also specify other combinations if required: for example `-strides -2,-1,3,4` would correspond to a PLS coordinate system, `-strides 2,3,4,1` would correspond to volume-contiguous storage (with RAS for the spatial axes), etc.

The different formats supported by *MRtrix3* differ in the range of strides that they support. The *MRtrix image formats* (*.mih* / *.mif*) are the only formats to support arbitrary combinations.

Note: Not all image formats support all possible stride combinations. The *MRtrix image formats* (*.mih* / *.mif*) are designed to handle arbitrary strides, while other image formats may only support a limited subset. When strides are requested that are not supported by the image format, a hopefully suitable alternative will be used instead.

Interaction between strides and transform

There is an interaction between the strides and the image transform: if the transform matrix corresponds to a 90° rotation, this can be viewed as changing the *strides* without affecting the transform. Such a large rotation has changed the order of storage relative to the anatomical labels typically used to refer to the ordering (e.g. RAS, LAS, etc). For example, if a RAS image is modified such that its transform rotates the image axes by 90° around the y axis, this in effect implies that voxels are now ordered IAR (i.e. *right* becomes *inferior*, *anterior* remains as-is, and *superior* becomes *right*).

The *MRtrix3* back-end will indeed interpret such large rotations as affecting the strides, so that if the strides are stated as 1, 2, 3, the order of storage will always be left->right, posterior->anterior, inferior->superior *relative to the scanner axes*. Note that this also implies that the transform matrix will always be modified as necessary to bring it close to the standard coordinate system, so that the first image axis is close to the *x* axis, etc. This allows *MRtrix3* applications to operate on images in the knowledge that these axes are always anatomically as expected, without worrying about the details of *how* this information was actually stored on file.

It is important to bear this in mind when interpreting for output of *mrinfo* for example, since this produces the strides and transform *as interpreted by MRtrix3*, rather than those actually stored on file - although the two representations should be strictly equivalent. If you need to inspect the information as stored on file, use *mrinfo*'s `-config RealignTransform false` option.

5.7.2 Supported image formats

This lists the various image formats currently supported by *MRtrix3*.

MRtrix image formats (*.mih* / *.mif*)

These MRtrix-specific image formats are closely related. They consist of a text header, with data stored in binary format, either within the same file (*.mif*) or as one or more separate files (*.mih*). In both cases, the header structure is the same, as detailed below. These file formats were devised to address a number of limitations inherent in currently available formats. In particular:

- simplicity: as detailed below, the header format is deliberately kept very simple and human-readable, making it easy to debug and edit manually if needed.
- extendability: any information can be stored in the header, and will simply be ignored by the application if not recognised.

- arbitrary data organisation: voxel values can be stored in any order, making it simple to ensure for example that all FOD coefficients for a given voxel are stored contiguously on file.

Note that *MRtrix3* now includes *MatLab* functions to read and write MRtrix image files, and to load MRtrix tracks files. These are located in the `matlab` subfolder.

Compressed MRtrix image format (`.mif.gz`)

MRtrix3 also supports the compressed version of the single-file `.mif` format, both for reading and writing.

Note: While this can reduce file sizes, it does incur a runtime cost when reading or writing the image (a process that can often take longer than the operation to be performed), and will require the entire image to be loaded uncompressed into RAM (*MRtrix3* can otherwise make use of [memory-mapping](#) to keep RAM requirements to a minimum). For large files, these costs can become considerable; you may find that *MRtrix3* can process a large uncompressed image, yet run out of RAM when presented with the equivalent compressed version (in such cases, you can try using `gunzip` to uncompress the file manually before invoking the relevant *MRtrix3* command).

Header structure

The header is the first (and possibly only) data stored in the file, as ASCII-encoded text (although other encodings such as UTF8 may work equally well). Lines should be separated by Unix-style newlines (line-feed, ‘`\n`’, ASCII 0x0A), although MRtrix will also accept DOS-type newlines.

The first line should read only `mrtrix image` to indicate that this is an image in MRtrix format. The last line of the header should read only `END` (followed by a newline character) to signal the end of the header, after which all data will be considered as binary.

All lines *between* these two entries must be represented as key-value pairs, as described below.

Header key-value pairs

All following lines are in the format `key: value`, with the value entry extending up to the end of the line. All whitespace characters before and after the value entry are ignored. Some keys are required to read the images, others are optional (sensible defaults will be substituted if they are absent). Recognised keys are provided in the list below, along with the expected format of the corresponding values.

- **dim** [required]

the image dimensions, supplied as a comma-separated list of integers. The number of entries specifies the dimensionality of the image. For example: `dim: 192,256,256` specifies a 192×256×256 image.

- **vox** [required]

the voxel size along each dimension, as a comma-separated list of floating-point values. The number of entries should match that given in the `dim` entry. For example: `vox: 0.9,0.898438,0.898438`.

- **layout** [required]

specifies the organisation of the data on file. In simplest terms, it provides a way of specifying the strides required to navigate the data file, in combination with the `dim` entry. It is given as a comma-separated list of signed integers, with the sign providing the direction of data traversal with respect to voxel coordinates, and the value providing a way of specifying the order of increasing stride.

For example, assuming an image with `dim: 192,256,256`, the entry `layout: +2,-0,-1` is interpreted as: the shortest stride is along the y-axis (second entry), then the z-axis (third entry), and then along the

x-axis. Voxels are stored in the order left to right (positive stride) along the x-axis; anterior to posterior along the y-axis (negative stride); and superior to inferior (negative stride) along the z-axis. Given the image dimensions, the final strides are therefore $256 \times 256 = 65536$ for adjacent voxels along the x-axis, -1 for the y-axis, and -256 for the z-axis. This also implies that the voxel at coordinate [0 0 0] is located 65536 voxel values into the data portion of the file.

- **datatype** [required]

the datatype used to store individual voxel values. See the listing of valid [Data types](#). For example: `datatype: UInt16LE`

- **file** [required]

specifies where the binary image data are stored, in the format `file: filename offset`, with the offset provided in bytes from the beginning of the file. For example: `file: image.dat 0`.

For the single-file format (.mif), the filename should consist of a single full-stop (‘.’) to indicate the current file, and the offset should correspond to a point in the file after the END statement of the header.

For the separate header/data format (.mih), the filename should refer to an existing file in the same folder as the header (.mih) file. Multiple such entries can be supplied if the data are stored across several files.

- **transform** [optional]

used to supply the 4x4 transformation matrix specifying the orientation of the axes with respect to real space. This is supplied as a comma-separated list of floating-point values, and only the first 12 such values will be used to fill the first 3 rows of the transform matrix. Multiple such entries can be provided to fill the matrix; for example, *MRtrix3* will normally produce 3 lines for the transform, with one row of 4 values per entry:

```
transform: 0.997986,-0.0541156,-0.033109,-74.0329
transform: 0.0540858,0.998535,-0.00179436,-100.645
transform: 0.0331575,2.34007e-08,0.99945,-125.84
```

- **scaling** [optional]

used to specify how intensity values should be scaled, provided as an offset and scale. Voxel values will be read as `value_returned = offset + scale * value_read`. For example: `scaling: -1,2`. Default is 0,1 (no modification).

In addition to these keys, it is also possible to store additional key-value pairs within the header of these image files. If a particular key is not recognised by *MRtrix3*, it is simply ignored (but may be carried over to any outputs resulting from the command, depending on the particular command).

There are some keys that are utilized by particular *MRtrix3* commands in order to preserve important information as image data are passed between commands. A prominent example is `dw_scheme`, which is used to embed the diffusion gradient table within the [Image header](#).

Note: Any header key-value pairs that involve storage of either numerical data, or multiple entries within a single key-value pair, must be stored using the following convention:

- The “.” period character as the numerical decimal separator.
- The “,” comma character as the delimiter between entries.

Creation or manipulation of header data such that it does not conform to these requirements may lead to unpredictable software behaviour.

DICOM (folder or .dcm)

DICOM format is only supported for reading. *MRtrix3* applications will assume an image is in DICOM format if the image specifier provided corresponds to a folder or ends with the .dcm extension. For a folder, the application will scan the entire folder and its subfolders for DICOM files and generate a list of DICOM patients, studies and series. If a single series is found within the folder, this data set will be accessed with no further interaction required. Otherwise, the user will be prompted to select the series of interest. *MRtrix3* supports data from all major manufacturers, including Siemens mosaics and the newer single-file multi-frame format.

A separate application, *dcminfo*, is provided to view all DICOM header elements within a particular DICOM file, including Siemens' custom shadow attributes (CSA).

Note that no support is provided for reading the DICOMDIR entry due to case-sensitivity issues. DICOM data are typically stored on CD or DVD on a case-insensitive filesystem. However, Unix systems will typically not access these filesystems in a case-insensitive manner, and will fail to find the appropriate files if the case of filenames supplied in the DICOMDIR file does not match the case of the files found on the CD or DVD.

NIfTI & NIfTI-2 (.nii)

These file formats are supported both for reading and writing, and allows interoperation with other packages such as *SPM* or *FSL*. The *mrinfo* command can be used to determine whether a particular image is in NIfTI-1 or NIfTI-2 format.

Note: Use of the NIfTI format can introduce ambiguity into the transformation information used to orient and localise the image data with respect to physical space, particularly when combined with the use of multiple software packages. More information is provided in the “:ref:nifti_qform_sform” section.

Compressed NIfTI (.nii.gz)

MRtrix3 also supports compressed NIfTI images (both versions 1 & 2), for both reading and writing.

Note: While this can reduce file sizes, it does incur a runtime cost when reading or writing the image (a process that can often take longer than the operation to be performed), and will require the entire image to be loaded uncompressed into RAM (*MRtrix3* can otherwise make use of [memory-mapping](#) to keep RAM requirements to a minimum). For large files, these costs can become considerable; you may find that *MRtrix3* can process a large uncompressed image, yet run out of RAM when presented with the equivalent compressed version (in such cases, you can try using *gunzip* to uncompress the file manually before invoking the relevant *MRtrix3* command).

FreeSurfer formats (.mgz / .mgz)

MRtrix3 supports both of these formats for reading and writing.

Images stored in these formats may include [additional data structures](#) that follow the image data. These data structures provide a similar functionality to the [Header key-value pairs](#) used in the *MRtrix image formats (.mih / .mif)*.

When present in an input file, *_MRtrix3_* will import these data into [Header key-value pairs](#), with keys named “MGH_*” (each element present in the input file is named and stored individually), and the values for these data structures will be written in legible format (e.g. matrix data are stored as delimited text). The data will therefore be encapsulated within the image header and preserved (as long as formats capable of retaining this information are used subsequently). For instance:

```
$ mrinfo image.mgz
*****
Image:                "image.mgz"
*****
Dimensions:           256 x 256 x 256
Voxel size:           1 x 1 x 1
Data strides:          [ -1 3 -2 ]
Format:                MGZ (compressed MGH)
Data type:             unsigned 8 bit integer
Intensity scaling:     offset = 0, multiplier = 1
Transform:              1    -4.098e-08    6.147e-08    -129.3
                       -8.196e-08         1    7.189e-09    -118.1
                       4.377e-08   -2.133e-08         1    -147.7
MGH_TAG_AUTO_ALIGN:    0.998104,0.054096,-0.029327,2.066329
                       -0.061351,0.912803,-0.403062,-27.35524
                       0.004969,0.404097,0.914391,-5.738687
                       0,0,0,1
MGH_TAG_MRI_FRAME:     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,,0,0 0 0 0 0 0 0 0 0 0 0
↪ 0 0 0 0,0,0
MGH_TAG_PEDIR:         UNKNOWN
MGH_TE:                1.91
MGH_TI:                1100
MGH_TR:                2300
MGH_flip:              7
```

Whenever `_MRtrix3_` writes an image to one of these formats, it will check the *Header key-value pairs* for any such data that may have been created by `_MRtrix3_` when importing such an image earlier. Any such data found will be correspondingly written to the data structures following the image data, formatted such that FreeSurfer tools are capable of reading them. Other header key-value entries that do not begin with “MGH_*”, and of which FreeSurfer is not aware, will `_not_` be written to this section of any output `.mgh / .mgz` image files.

Analyse format (.img / .hdr)

This file format is supported both for reading and writing. However, when writing, the newer NIfTI standard will be used, since the Analyse format cannot store crucial information such as the image transform, and is hence deprecated. If these images are actually stored as NIfTI, they will be handled appropriately according to the standard.

Note: In order to specify an Analyse format image on the command line, type the name of its *data* file (* .img), *not* the header file.

Warning: By default, Analyse format images will be assumed to be stored using RAS (radiological) convention. This can be modified in the *Configuration file*, by setting the `Analyse.LeftToRight` entry to `true`.

5.7.3 Fixel image (directory) format

Documentation on this format has moved to a new location:

Fixel image (directory) format

5.7.4 Legacy MRtrix Sparse Format (.msh / .msf)

This is an old legacy format previously used for applications where the number of discrete elements within a voxel may vary between voxels (typically used to store fixels). This format has been superseded by the new directory-based *Fixel image (directory) format*. While all fixel-related commands now only use the new format, files stored in the legacy format can still be viewed in `mrview`.

Much like the standard *MRtrix image formats (.mih / .mif)*, there are two different image file extensions available. One (.msh) separates the image header information and raw data into separate files, while the other (.msf) encodes all information relevant to the image into a single file.

However unlike these established formats, sparse images contain *two* separate raw data fields. The first of these behaves identically to standard images: a single intensity value for every image element. The second stores sparse image data. For any particular image element, the intensity value within the standard image field defines a *pointer* to a location within the sparse image field, where the sparse data relevant for that image element can be found.

Additional image header features

These image formats have some features within the image header that differ from the standard MRtrix image formats:

- The ‘magic number’ that appears at the start of the file must read ‘mrtrix sparse image’.
- Key:value pair ‘sparse_data_name’ defines the *name* of the class used in the sparse data field. This class name is typically not reader-friendly; the value that appears is that provided by the C++ call `typeid(XYZ).name()` for a class called XYZ. This is necessary to ensure that the data stored in the sparse field can be interpreted correctly.
- Key:value pair ‘sparse_data_size’ defines the size (in bytes) of the class used to store the sparse data.
- The ‘datatype’ field MUST be a 64-bit integer, with the same endianness as the system. A 64-bit integer type is required because the standard image data provides pointers to the sparse data in memory, while the endianness is tested to ensure that the sparse data can be interpreted correctly. Note that sparse images cannot be transferred and used between systems with different endianness.
- In addition to the ‘file’ key, a second key ‘sparse_file’ is also required, which provides the path to the beginning of the sparse image data. In the .msf format, this provides an offset from the start of the file to the start of the sparse data field; in the .msh format, a second associated data file with the extension .sdaf is generated on image creation, and the path to this file is defined in the header.

Sparse data storage

Within the sparse data field, there is no delimiting information or identifying features; the image format relies on the integers stored in the standard image field to provide offset pointers to appropriate locations within the sparse field.

From the data position defined by such an offset, the first 4 bytes provide a 32-bit integer (with native endianness), which specifies the number of discrete elements stored. This is followed by data to fill precisely that number of instances of the sparse data class. Note that no endianness conversion can be performed on this data; data is read and written using a straight memory copy.

5.7.5 Tracks file format (.tck)

The format for track files is similar to that for *MRtrix image formats (.mih / .mif)*. It consists of a text header in the same key: value format, ending with a single ‘END’ statement (terminated by a newline character), and followed by binary data.

The first line of the header should read `mrtrix tracks` to indicate that the file contains tracks in MRtrix format. Further `key: value` pairs typically provide information about the parameters used to produce the tracks, and for the most part are not required to read the data. The only *required* keys are the following:

- **file** A `file:` `offset` entry is required to specify the byte offset from the beginning of the file to the start of the binary track data. At this stage, only the single-file format is supported - in other words the filename part must be specified as `.` (see above for details).
- **datatype** Specifies the datatype (and byte order). Only real floating-point data types are permitted: either 32 or 64 bits (32 is the default), and either little-endian (LE) or big-endian (BE) ordering (the native ordering of the device used to generate the file is used as default). The valid *Data types* are therefore: Float32BE, Float32LE, Float64BE, Float64LE.

While not strictly compulsory, track files generated by *MRtrix3* commands will additionally always contain the following:

- **timestamp** A floating-point value that can be effectively used as a unique identifier for the file produced. In *MRtrix3* commands this is based on the number of nanoseconds since the epoch of the system timer.
- **count** The number of streamlines stored in the file. This is commonly used to produce accurate progress information for commands that read streamline data from file. Note that even if an *MRtrix3* command is terminated prematurely, the value stored in this entry *should* reflect the number of streamlines actually stored in the file; this can however be verified for any particular file using the *MRtrix3* command *tckinfo* with the `-count` option.
- **total_count** For command *tckgen*, the value stored in this field reflects the total number of streamlines that were generated, before the application of criteria for streamline acceptance / rejection; for other commands that operate on pre-calculated streamlines data rather than generating them, this field will reflect the number of streamlines that were *input* to that command, rather than the number that were subsequently stored in the output file.

The binary track data themselves are stored as triplets of floating-point values: one triplet of values per vertex along the track. Tracks are separated using a triplet of NaN (Not A Number) values. Finally, a triplet of Inf (infinity) values is used to indicate the end of the file.

5.7.6 Track Scalar File format (.tsf)

The Track Scalar File (TSF) format is very similar to the *Tracks file format (.tck)*, in that it includes a header of key-value pairs, followed by a stream of binary data relating to streamlines, with NaN delimiting between streamlines and Inf indicating the end of the file. However rather than storing information about the *locations* of streamline vertices, this format instead encodes *some quantitative value* at the location of each streamline vertex.

It differs from the *Tracks file format (.tck)* in the following ways:

- **Header:**
 - The first line of the header should instead contain the string: `mrtrix track scalars`.
 - In addition to the `file:` and `datatype:` keys, a TSF file must also contain the `timestamp` key; the value stored here must be a *perfect match* to the value of the `timestamp` field stored in the header of the `.tck` file based on which the track scalar file is being generated.
- **Data:**
 - Rather than storing *triplets* of floating-point values, with a triplet of NaN values delimiting between streamlines and a triplet of Inf values indicating the end of the file, a `.tsf` files contains *one* floating-point value per streamline vertex, with *one* NaN value delimiting between streamlines and *one* Inf value indicating the end of the file.

When reading a `.tsf` file, validation of that file against the streamline vertex data stored in a `.tck` file on which the track scalar values are based is typically performed by comparing the `timestamp` and `count` fields in the headers

of the two files. Undefined behaviour in some instances occur can occur if an attempt is made to read a particular `.tsf` alongside some `.tck` file to which it does not correspond if these checks are not first performed. If there is doubt regarding the validity of a `.tsf` / `.tck` file pair, the *MRtrix3* command *tsfvalidate* can be used to perform a more exhaustive cross-examination of the two files.

5.8 Command-line usage

MRtrix3 generally follows a relatively standard Unix syntax, namely:

```
$ command [options] argument1 argument2 ...
```

If you need to become familiar with using the command-line, there are plenty of tutorials online to get you started. There are however a few notable features specific to *MRtrix3*, which are outlined below.

5.8.1 Ordering of options on the command-line

Options can typically occur anywhere on the command-line, in any order - they do not usually need to precede the arguments.

For instance, all three of the lines below will have the same result:

```
$ command -option1 -option2 argument1 argument2
$ command argument1 argument2 -option1 -option2
$ command -option2 argument1 argument2 -option1
```

Care must however be taken in cases where a command-line option *itself has an associated compulsory argument*. For instance, consider a command-line option `-number`, which allows the user to manually provide a numerical value in order to control some behaviour. The user's desired value *must* be provided *immediately after* `-number` appears on the command-line in order to be correctly associated with that particular option.

For instance, the following would be interpreted correctly:

```
$ command -number 10 argument1 argument2
```

But the following would *not*:

```
$ command -number argument1 10 argument2
```

The following cases would also *not* be interpreted correctly by *MRtrix3*, even though some other softwares may interpret their command-line options in such ways:

```
$ command -number10 argument1 argument2
$ command --number=10 argument1 argument2
```

There are a few cases in *MRtrix3* where the order of options on the command-line *does* matter, and hence the above demonstration does not apply:

- *mrcalc*: `mrcalc` is a stack-based calculator, and as such, the order of inputs and operations on the command-line determine how the mathematical expression is formed.
- *mrview*: `mrview` includes a number of command-line options for automatically configuring the viewing window, and importing data into its various tools. Here the order of such options does matter: the command line contents are read from left to right, and any command-line options that alter the display of a particular image or data open within a tool is applied to the *most recent* data (image or otherwise) opened by the tool associated with that option.

- *Scripts*: A subset of the Python scripts provided with *MRtrix3* require the selection of an *algorithm*, which defines the approach that the script will use to arrive at its end result based on the data provided. The name of this algorithm *must* be the *first* argument on the command-line; any command-line options provided *prior* to this algorithm name will be **silently ignored**.

5.8.2 Number sequences

Some options expect arguments in the form of *number sequences* or *floating-point lists of numbers*. The former consists of a series of integers separated by commas or colons (no spaces), with colons indicating a range, optionally with an increment (if different from 1). For example:

- 1,4,8 becomes [1 4 8]
- 3,6:12,2 becomes [3 6 7 8 9 10 11 12 2]
- 1:3:10,8:2:0 becomes [1 4 7 10 8 6 4 2 0]

Note that the sign of the increment does not matter, it will always run in the direction required.

Certain commands (e.g. `mrconvert`) also accept the special `end` keyword, which will take on the largest possible value along the corresponding axis. For example, the following command can be used to extract every other volume from an input 4D image:

```
$ mrconvert input.mif -coord 3 0:2:end output.mif
```

5.8.3 Floating-point lists

Likewise, floating-point lists consist of a comma-separated list of numbers, for example:

- 2.47,-8.2223,1.45e-3

As in the integer case, it is also possible to supply a range of values using the colon syntax, although in this case the increment is mandatory. For example:

- 3.1:2.2:10 expands to [3.1 5.3 7.5 9.7]

The `nan` keyword can also be provided to insert a Not-a-Number.

5.8.4 Using shortened option names

Options do not need to be provided in full, as long as the initial part of the option provided is sufficient to unambiguously identify it.

For example:

```
$ mrconvert -debug in.mif out.nii.gz
```

is the same as:

```
$ mrconvert -de in.mif out.nii.gz
```

but will conflict with the `-datatype` option if shortened any further:

```
$ mrconvert -d in.mif out.nii.gz
mrconvert: [ERROR] several matches possible for option "-d": "-datatype", "-debug"
```

5.8.5 Unix Pipelines

The output of one program can be fed straight through to the input of another program via [Unix pipes](#) in a single command. The appropriate syntax is illustrated in this example:

```
$ dwi2tensor /data/DICOM_folder/ - | tensor2metric - -vector ev.mif
dwi2tensor: [done] scanning DICOM folder "/data/DICOM_folder/"
dwi2tensor: [100%] reading DICOM series "ep2d_diff"...
dwi2tensor: [100%] reformatting DICOM mosaic images...
dwi2tensor: [100%] loading data for image "ACME (hm) [MR] ep2d_diff"...
dwi2tensor: [100%] estimating tensor components...
tensor2metric: [100%] computing tensor metrics...
```

This command will execute the following actions:

1. `dwi2tensor` will load the input diffusion-weighted data in DICOM format from the folder `/data/DICOM_folder/` and compute the corresponding tensor components. The resulting data set is then fed into the pipe.
2. `tensor2metric` will access the data set from the pipe, generate an eigenvector map and store the resulting data set as `ev.mif`.

The two stages of the pipeline are separated by the `|` symbol, which indicates to the system that the output of the first command is to be used as input for the next command. The image that is to be fed to or from the pipeline is specified for each program using a single dash `-` where the image would normally be specified as an argument.

Warning: If you use the `-` symbol *without* piping through to the next command, the temporary file created will *not* be deleted.

For example, with a command like this:

```
$ mrconvert image.mif -
```

`mrconvert` has no way to tell that the image it produced hasn't been passed onto another command. This means the temporary file it creates will remain on the filesystem (default: `/tmp/` on Unix, `'.'` on Windows, see [Configuration file](#)), needlessly taking up storage space. Eventually this can fill up the entire filesystem, causing subsequent commands to fail and creating problems for the system as a whole. If this happens, the script `mrtrix_cleanup` can be used to remove temporary files from a specific location.

For this to work properly, it is important to know which arguments each program will interpret as input images, and which as output images. For example, this command will fail:

```
dwi2tensor - /data/DICOM_folder/ | tensor2metric - ev.mif
```

In this example, `dwi2tensor` will hang waiting for input data (its first argument should be the input DWI data set). This will also cause `tensor2metric` to hang while it waits for `dwi2tensor` to provide some input.

Advanced pipeline usage

Such pipelines are not limited to two programs. Complex operations can be performed in one line using this technique. Here is a longer example:

```
$ dwi2tensor /data/DICOM_folder/ - | tensor2metric - -vector - | mrcalc -
mask.nii -mult - | mrview -
dwi2tensor: [done] scanning DICOM folder "/data/DICOM_folder/"
```

(continues on next page)

(continued from previous page)

```

dwi2tensor: [100%] reading DICOM series "ep2d_diff"...
dwi2tensor: [100%] reformatting DICOM mosaic images...
dwi2tensor: [100%] loading data for image "ACME (hm) [MR] ep2d_diff"...
dwi2tensor: [100%] estimating tensor components...
tensor2metric: [100%] computing tensor metrics...
mrcalc: [100%] computing: (/tmp/mrtrix-tmp-VihKrg.mif * mask.nii) ...

```

This command will execute the following actions:

1. `dwi2tensor` will load the input diffusion-weighted data in DICOM format from the folder `/data/DICOM_folder/` and compute the corresponding tensor components. The resulting data set is then fed into the pipe.
2. `tensor2metric` will access the tensor data set from the pipe, generate an eigenvector map and feed the resulting data into the next stage of the pipeline.
3. `mrcalc` will access the eigenvector data set from the pipe, multiply it by the image `mask.nii`, and feed the resulting data into the next stage of the pipeline.
4. `mrview` will access the masked eigenvector data set from the pipe and display the resulting image.

How is it implemented?

The procedure used in *MRtrix3* to feed data sets down a pipeline is somewhat different from the more traditional use of pipes. Given the large amounts of data typically contained in a data set, the ‘standard’ practice of feeding the entire data set through the pipe would be prohibitively inefficient. *MRtrix3* applications access the data via memory-mapping (when this is possible), and do not need to explicitly copy the data into their own memory space. When using pipes, *MRtrix3* applications will simply generate a temporary file and feed its filename through to the next stage once their processing is done. The next program in the pipeline will then simply read this filename and access the corresponding file. The latter program is then responsible for deleting the temporary file once its processing is done.

This implies that any errors during processing may result in undeleted temporary files. By default, these will be created within the `/tmp` folder (on Unix, or the current folder on Windows) with a filename of the form `mrtrix-tmp-XXXXXX.xyz` (note this can be changed by specifying a custom `TmpFileDir` and `TmpFilePrefix` in the [Configuration file](#)). If a piped command has failed, and no other *MRtrix* programs are currently running, these can be safely deleted.

Really advanced pipeline usage

As implemented, *MRtrix3* commands treat image file names that start with the `TmpFilePrefix` (default is `mrtrix-tmp-`) as temporary. When reading the image name from the previous stage in the pipeline, the image file name will trivially match this. But this also means that it is possible to provide such a file as a normal *argument*, and it will be treated as a temporary *pipelined* image. For example:

```

$ mrconvert /data/DICOM/ -datatype float32 -
mrconvert: [done] scanning DICOM folder "/data/DICOM/"
mrconvert: [100%] reading DICOM series "ep2d_diff"...
mrconvert: [100%] reformatting DICOM mosaic images...
mrconvert: [100%] copying from "ACME (hm) [MR] ep2d_diff" to "/tmp/mrtrix-tmp-zcDlnr.
↪mif"...
/tmp/mrtrix-tmp-zcDlnr.mif

```

Notice that the name of the temporary file is now printed on the terminal, since the command’s stdout has not be piped into another command, and we specified `-` as the second argument. You’ll also see this file is now present in the `/tmp`

folder. You can use this file by copy/pasting it as an *argument* to another *MRtrix* command (be careful though, it will be deleted once this command exits):

```
$ mrstats /tmp/mrtrix-tmp-zcDlnr.mif
      channel      mean      median      std. dev.      min      max
↪ count      [ 0 ]      1053.47      96      1324.71      0      3827
↪ 506880      [ 1 ]      173.526      84      140.645      0      549
↪ 506880
...
```

This allows for a non-linear arrangement of pipelines, whereby multiple pipelines can feed into a single command. This is achieved by using the shell's output capture feature to insert the temporary file name of one pipeline as an argument into a second pipeline. In BASH, output capture is achieved using the `$(commands)` syntax, or equivalently using backticks: ``commands``. For example:

```
$ dwi2tensor /data/DICOM/ - | tensor2metric - -mask $(dwi2mask /data/DICOM/ - |
↪ maskfilter - erode -npass 3 - ) -vec ev.mif -fa - | mrthreshold - -top 300 highFA.
↪ mif
dwi2mask: [done] scanning DICOM folder "/data/DICOM/"
dwi2tensor: [done] scanning DICOM folder "/data/DICOM/"
dwi2mask: [100%] reading DICOM series "ep2d_diff"...
dwi2tensor: [100%] reading DICOM series "ep2d_diff"...
dwi2mask: [100%] reformatting DICOM mosaic images...
dwi2tensor: [100%] reformatting DICOM mosaic images...
dwi2mask: [100%] loading data for image "ACME (hm) [MR] ep2d_diff"...
dwi2tensor: [100%] loading data for image "ACME (hm) [MR] ep2d_diff"...
dwi2mask: [100%] finding min/max of "mean b=0 image"...
dwi2mask: [done] optimising threshold...
dwi2mask: [100%] thresholding...
dwi2tensor: [100%] estimating tensor components...
dwi2mask: [100%] finding min/max of "mean b=1000 image"...
dwi2mask: [done] optimising threshold...
dwi2mask: [100%] thresholding...
dwi2mask: [done] computing dwi brain mask...
maskfilter: [100%] applying erode filter to image -...
tensor2metric: [100%] computing tensor metrics...
mrthreshold: [100%] thresholding "/tmp/mrtrix-tmp-UHvhc2.mif" at 300th top voxel...
```

In this one command, we asked the system to perform this non-linear pipeline:

```
      dwi2tensor \
                        |--> tensor2metric  ---> mrthreshold
dwi2mask ---> maskfilter /
```

More specifically:

1. `dwi2tensor` will load the input diffusion-weighted data in DICOM format from the folder `/data/DICOM/` and compute the corresponding tensor components. The resulting data set is then fed into the pipe.
 1. meanwhile, `dwi2mask` will generate a brain mask from the DWI data, and feed the result into a second pipeline.
 2. `maskfilter` will access the mask from this second pipeline, erode the mask by 3 voxels, and output the name of the temporary file for use as an *argument* by the next stage.
2. `tensor2metric` will access the tensor data set from the first pipe, generate eigenvector and FA maps within the mask provided as an *argument* by the second pipeline, store the eigenvector map in `ev.mif` and feed the

FA map into the next stage of the pipeline.

3. `mrthreshold` will access the FA image from the pipe, identify the 300 highest-valued voxels, and produce a mask of these voxels, stored in `highFA.mif`.

5.9 Configuration file

The behaviour of a number of aspects of *MRtrix3* can be controlled by the user via the *MRtrix3 configuration file*. Note, that this file is distinct from the build configuration file that is generated as part of the *MRtrix3* installation, but rather is used to specify default settings for a number of parameters, many of which relate to data visualisation when using `mrview`.

For all available configurable options, please refer to the [configuration file options](#) page.

5.9.1 Location

MRtrix3 applications will attempt to read configuration information from a two locations. The system-wide configuration file `/etc/mrtrix.conf` is read first if present, followed by the user-specific configuration `~/.mrtrix.conf`. If both system and user-specific configuration files exist, the parameters specified in the two configuration files will be aggregated, with user-specified configuration options taking precedence in the case of a conflict. In the case that a particular configuration parameter is not defined, *MRtrix3* will resort to hard-coded defaults.

5.9.2 Format

The configuration files are text files, with each line containing a key: value pair. For example

```
Analyse.LeftToRight: false
NumberOfThreads: 2
```

Note: Key names are case-sensitive.

The value entry may be interpreted by *MRtrix3* applications as either:

- **Boolean:** allowed values here are true or false
- **Integer:** any integer value
- **Floating-point:** any floating-point value
- **Text:** any text string, without any further interpretation

The list of all configuration file options can be found [here](#).

5.10 DWI denoising

MRtrix includes the command `dwidenoise`, which implements dMRI noise level estimation and denoising based on random matrix theory. The method exploits data redundancy in the patch-level PCA domain ([Veraart2016a], [Veraart2016b] and [CorderoGrande2019]). The method uses the prior knowledge that the eigenspectrum of random covariance matrices is described by the universal Marchenko-Pastur (MP) distribution.

5.10.1 Recommended use

Image denoising must be performed as the first step of the image-processing pipeline. Interpolation or smoothing in other processing steps, such as motion and distortion correction, may alter the noise characteristics and thus violate the assumptions upon which MP-PCA is based.

Typical use will be:

```
dwidenoise dwi.mif out.mif -noise noise.mif
```

where `dwi.mif` contains the raw input DWI image, `out.mif` is the denoised DWI output, and `noise.mif` is the estimated spatially-varying noise level.

We always recommend eyeballing the residuals, i.e. `out - in`, as part of the quality control. The lack of anatomy in the residual maps is a marker of accuracy and signal-preservation during denoising. The residuals can be easily obtained with

```
mrcalc dwi.mif out.mif -subtract res.mif
mrview res.mif
```

5.10.2 Advanced options

Patch size

The noise level in MRI is spatially varying, due to the proximity of the coil elements and parallel imaging. Noise level estimation and denoising therefore operates in image patches around each voxel, where the noise can be assumed to be approximately homoscedastic. The patch size can be chosen by the user with the option `-extent`. For maximal SNR gain (when using Exp2, see below) we suggest to choose $N \approx M$, where M is the no. DW volumes and N is the number of kernel elements. However, larger kernels also extend the required run time, so in large datasets it might be beneficial to select smaller sliding kernels. By default, the command will select the smallest isotropic patch size that exceeds the number of DW images in the input data, e.g., 5x5x5 for data with ≤ 125 DWI volumes, 7x7x7 for data with ≤ 343 DWI volumes, etc.

Noise level estimation

The noise level in each patch is experimentally estimated from the eigenvalue spectrum of the local data matrix. Assuming $M < N$, P signal-carrying components (also estimated), and $M - P$ noise components, the squared noise level is estimated as:

$$\sigma^2 = \frac{\lambda_{P+1} - \lambda_M}{4\sqrt{\gamma}}$$

where λ_i are the eigenvalues of the covaliance matrix, sorted in decreasing order, and γ is the matrix ratio.

`dwidenoise` implements two different versions of this estimator, based on a different definition of the matrix ratio γ :

- **Exp1** uses the definition used in the original papers [Veraart2016a] and [Veraart2016b], namely $\gamma = (M - P)/N$.
- **Exp2** uses $\gamma = (M - P)/(N - P)$ instead, which was shown in [CorderoGrande2019] to improve the estimation. This is now the default.

Complex data

Note that *dwidenoise* does not correct for non-Gaussian noise biases present in magnitude-reconstructed MRI images. Including MRI phase images can reduce such Rician bias [CorderoGrande2019], and the command now supports complex input data. To this end, users can run:

```
mrcalc dwi_magnitude.mif dwi_phase_rad.mif -polar dwi_complex.mif
dwidenoise dwi_complex.mif out.mif -noise noise.mif
mrcalc out.mif -abs out_magnitude.mif
```

Where *dwi_magnitude.mif* and *dwi_phase_rad.mif* are the magnitude and phase images respectively. Note that the code above assumes that the phase is scaled in radians.

Whilst using complex data can effectively reduce Rician noise bias, *dwidenoise* currently does not account for spatial noise correlations introduced due to in-plane acceleration. [CorderoGrande2019] addresses these additional effects at reconstruction, but this is not implemented in MRtrix3.

5.11 DWI distortion correction using dwifslpreproc

The *dwifslpreproc* script, responsible for performing general pre-processing of DWI series, has been completely re-designed as part of the *MRtrix3* 3.0_RC1 update. Although the ‘guts’ of the script are completely new, the fundamental operation - eddy current-induced distortion correction, motion correction, and (optionally) susceptibility-induced distortion correction, using FSL’s *eddy* / *topup* / *applytopup* tools, remains the same. While the user interface remains reasonably similar to that provided previously (examples to come), they are slightly different.

The major benefit of the new design is that *MRtrix3* is now capable of not only *capturing the relevant phase encoding information* from DICOM headers, but also *using that information* within *dwifslpreproc* to internally generate the necessary phase encoding table files in order to run these FSL tools. This comes with a number of benefits:

- It makes it possible to acquire and process a wider range of DWI acquisition designs, without requiring that the user laboriously manually construct the phase encoding tables that these FSL tools require.
- It means that automated pre-processing pipelines (e.g. these two [works-in-progress](#)) can be applied to provided data without requiring manual intervention to specify this information.
- Over time, as *MRtrix* 0.3.16 code is used to import DICOMs (and hence capture the phase encoding information) and the relevant code is thoroughly tested, there will be less onus on users to track and specify the type of phase encoding acquisition performed.

Note: Although the *dwifslpreproc* script is provided as part of *MRtrix3* in the hope that users will find it useful, the major image processing steps undertaken by this script are still performed using tools developed at FMRIB and provided as part of FSL. It is therefore *essential* that the appropriate references be cited whenever this script is used!

The *dwifslpreproc* script now has four major ‘modes’ of operation, that can be selected at the command-line using the *-rpe_** options. Note that exactly *one* of these options ***must*** be provided. The following are example use cases; specific parameters, file names etc. may need to be altered to reflect your particular data.

1. No variation in phase encoding

All DWI volumes are acquired with precisely the same phase encoding direction and EPI readout time, and no additional spin-echo *b=0* images are acquired over and above those interleaved within the DWI acquisition. It is therefore impossible to estimate the inhomogeneity field using *topup*, and *eddy* will perform motion and eddy current correction only.

Example DICOM image data:

```
002_-_DWI_phaseAP/
```

Old usage (i.e. prior to MRtrix 0.3.16):

```
$ dwifslpreproc AP 002_-_DWI_phaseAP/ dwi_preprocessed.mif -rpe_none
```

New usage:

```
$ dwifslpreproc 002_-_DWI_phaseAP/ dwi_preprocessed.mif -rpe_none -pe_dir AP_
↪ [ -readout_time 0.1 ]
```

Note that here (and in subsequent examples), providing the EPI readout time manually is optional (if omitted, the ‘sane’ default of 0.1s will be assumed). The precise scaling of this parameter is not expected to influence results provided that the readout time is equivalent for all $b=0$ / DWI volumes.

2. Reversed phase encode $b=0$ pair(s)

All DWI volumes are acquired with precisely the same phase encoding direction and EPI readout time. In addition, one or more pairs of spin-echo $b=0$ EPI volumes are provided, where half of these volumes have the same phase encoding direction and readout time as the DWIs, and the other half have precisely the *opposite* phase encoding direction (but the same readout time). These additional images are therefore used to estimate the inhomogeneity field, but do not form part of the output DWI series.

Example DICOM image data:

```
002_-_ep2dse_phaseAP/
003_-_ep2dse_phasePA/
004_-_DWI_phaseAP/
```

Old usage (i.e. prior to MRtrix 0.3.16):

```
$ dwifslpreproc AP 004_-_DWI_phaseAP/ dwi_preprocessed.mif -rpe_pair 002_-_
↪ ep2dse_phaseAP/ 003_-_ep2dse_phasePA/
```

New usage:

```
$ mrcat 002_-_ep2dse_phaseAP/ 003_-_ep2dse_phasePA/ b0s.mif -axis 3
$ dwifslpreproc 004_-_DWI_phaseAP/ dwi_preprocessed.mif -pe_dir AP -rpe_pair_
↪ -se_epi b0s.mif [ -readout_time 0.1 ]
```

3. Reversed phase encoding for all DWIs

For all diffusion gradient directions & b -values, two image volumes are obtained, with the opposite phase encoding direction with respect to one another. This allows for the combination of the two volumes corresponding to each unique diffusion gradient direction & strength into a single volume, where the relative compression / expansion of signal between the two volumes is exploited.

Example DICOM image data:

```
002_-_DWI_64dir_phaseLR/
003_-_DWI_64dir_phaseRL/
```

Old usage (i.e. prior to MRtrix 0.3.16):

```
$ dwifslpreproc LR 002_-_DWI_64dir_phaseLR/ dwi_preprocessed.mif -rpe_all_
↪ 003_-_DWI_64dir_phaseRL/
```

New usage:

```
$ mrcat 002_-_DWI_64dir_phaseLR/ 003_-_DWI_64dir_phaseRL/ all_DWIs.mif -axis_↵
↵3
$ dwifslpreproc all_DWIs.mif dwi_preprocessed.mif -pe_dir LR -rpe_all [ -
↵readout_time 0.1 ]
```

Note that in this particular example, the `dwifslpreproc` script will in fact extract the $b=0$ volumes from the input DWIs and use those to estimate the inhomogeneity field with `topup`. If additional $b=0$ images are also acquired, and it is desired to instead use those images to estimate the inhomogeneity field only, the `-se_epi` option can be used.

4. Arbitrary phase encoding acquisition

In cases where either:

- An up-to-date version of *MRtrix3* has been used to convert from DICOM, such that phase encoding information is embedded in the image header; or:
- Image data of unknown origin are to be processed by an automated pipeline without user intervention, and therefore phase encoding information must be provided using data files associated with the input images (such as JSON files in the [BIDS standard](#)),

it is possible for the `dwifslpreproc` script to automatically determine the appropriate steps to perform based on the phase encoding configuration of the image data presented to it.

Usage:

```
$ mrcat <all_input_DWIs> all_dwis.mif -axis 3
$ mrcat <all_extra_b=0_volumes> all_b0s.mif -axis 3 (optional)
$ dwifslpreproc all_dwis.mif dwi_preprocessed.mif -rpe_header [ -se_epi all_
↵b0s.mif ]
```

Warning: With regards to Option 4 (using phase encoding information as it is stored in the header), note that this functionality is preliminary and should not be trusted blindly. It is impossible for us to check and test all possible usage scenarios. Furthermore, if this information is imported or exported to/from the image header, this requires reorientation due to the way in which *MRtrix3* handles image orientations internally, which introduces additional mechanisms by which the tracking of phase encoding orientations may go awry. Results should therefore be checked manually if using / testing this mechanism.

When one of the options 1-3 are used, internally the `dwifslpreproc` script *generates the effective phase encoding table* given the user's images and command-line input; this is what is passed to `topup` / `applytopup` / `eddy`. If one of these options is used, but there is actually phase encoding information found within the image header(s), the script will *compare* the user's phase encoding specification against the header contents, and produce a warning if it detects a mismatch (since either the phase encoding design is not what you think it is, or the import of phase encoding information from DICOM is awry; either warrants further investigation).

5.11.1 Configuring FSL commands `topup` and `eddy`

The two principal FSL commands with which the *MRtrix3* `dwifslpreproc` script interfaces - `topup` and `eddy` - themselves have a wide range of command-line options that may be utilised to alter their behaviour. `dwifslpreproc` does not provide any command-line options to these commands over and above those absolutely necessary to perform the processing operations; as such, the default parameters / behaviour of these commands as configured in FSL will be utilised. As such, any non-default parameters or behaviour of these commands desired by the user (e.g. `eddy` outlier replacement) must be specified *explicitly*. This is achieved by using the `dwifslpreproc`

command-line options `-topup_options` and `-eddy_options`, which accept as input a string that will then be propagated down to the `topup` or `eddy` command invocation respectively.

For information on the command-line options available within these FSL commands, consult the internal help page of the command installed on your system, and/or the online FSL documentation:

- `topup`: <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/topup/TopupUsersGuide>
- `eddy`: <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/eddy/UsersGuide>

Note: Due to a quirk in the Python library `argparse` that is utilised by *MRtrix3* for command-line parsing, in some instances, some contents of a string provided at the command-line encased in double-quotes may get lost during parsing. To mitigate this, we suggest that whatever text data are provided to `topup` or `eddy` via the `-topup_options` or `-eddy_options` command-line options contain an additional space at the beginning or end of the string, like thus:

```
dwifslpreproc ... -eddy_options " --repol" ...
```

5.11.2 Using eddy's slice-to-volume motion correction capability

As of September 2017, FSL's `eddy` tool has the capability of not only estimating and correcting motion *between* DWI volumes, but also motion *within* volumes. Details of this method can be found within the relevant [publication](#). *MRtrix3* is capable of supporting this underlying `eddy` functionality within the wrapping `dwifslpreproc` script. Below are a few relevant details to assist users in getting this working:

- At time of writing, only the CUDA version of the `eddy` executable provides the slice-to-volume correction capability. Therefore, this version must be installed on your system, and CUDA itself must be appropriately set up. Note that with *MRtrix3* version 3.0_RC3, presence of the CUDA version of `eddy` will be automatically detected within your `PATH` by `dwifslpreproc`, and this version will be executed in preference to the OpenMP version.
- `eddy`'s slice-to-volume correction is triggered by the presence of the `--mporder=#` command-line option. Therefore, to activate this behaviour, the contents of the `-eddy_options` command-line option passed to `dwifslpreproc` must contain this entry.
- The timing of acquisition of each slice must be known in order to perform slice-to-volume correction. This is provided to `eddy` via the command-line option `--slspec=`, where a text file is provided that defines the order in which the slices are acquired within each volume. In `dwifslpreproc`, there are two ways in which this information can be provided:
 - If you include the string `--slspec=path/to/file.txt` within the contents of the `-eddy_options` command-line option, then `dwifslpreproc` will copy the file to which you have provided the path into the temporary directory created by the script, such that `eddy` appropriately locates that file.
 - If DICOM conversion & all subsequent processing is performed solely using *MRtrix3* commands, and *Header key-value pairs* are preserved, then where possible, *MRtrix3* will store fields "SliceEncodingDirection" and "SliceTiming" based on DICOM information (note that the naming of these fields is consistent with the [BIDS specification](#)). `dwifslpreproc` will then use these fields to internally generate the "slspec" file required by `eddy` without user intervention; as long as the user does *not* provide the `--slspec=` option within `-eddy_options`.

5.12 Response function estimation

A prerequisite for spherical deconvolution is obtaining the response function(s), which is/are used as the kernel(s) by the deconvolution algorithm. For the white matter, the response function models the signal expected for a voxel containing a single, coherently oriented bundle of axons [Tournier2004] [Tournier2007]. In case of multi-tissue variants of spherical deconvolution, response functions for other tissue types are introduced as well; typically to represent grey matter(-like) and/or CSF(-like) signals [Jeurissen2014] [Dhollander2016a].

In MRtrix3, the *dwi2response* script offers a range of algorithms to estimate these response function(s) directly from your dataset itself. This process of estimating response function(s) from the data is non-trivial. No single algorithm works for *any* possible scenario, although some have proven to be more widely applicable than others.

5.12.1 General recommendations

Choice of algorithm

While many algorithms exist, the following appear to perform well in a wide range of scenarios, based on experience and testing from both developers and the MRtrix3 community:

Single-tissue CSD: If you intend to perform (single-tissue) *Constrained spherical deconvolution* (e.g. via *dwi2fod csd*), the *tournier* algorithm is a convenient and reliable way to estimate the single-fibre white matter response function:

```
dwi2response tournier dwi.mif wm_response.txt
```

Other options include the *fa* or *tax* algorithms.

Multi-tissue CSD or global tractography: If you intend to perform a *multi-tissue* analysis, such as *Multi-shell multi-tissue constrained spherical deconvolution* (e.g. via *dwi2fod msmt_csd*) or *Global tractography* (e.g. via *tckglobal*), the *dhollander* algorithm is a convenient and reliable way to estimate the single-fibre white matter response function as well as the grey matter and CSF response functions:

```
dwi2response dhollander dwi.mif wm_response.txt gm_response.txt csf_response.txt
```

Other options include the *msmt_5tt* algorithm.

Checking the results

In general, it's always worthwhile checking your response function(s):

```
shview wm_response.txt
```

Use the left and right arrow (keyboard) keys in this viewer to switch between the different b-values ('shells') of the response function, if it has more than one b-value (this would for example be the case for the outputs of the *dhollander* algorithm).

It may also be helpful to check which voxels were selected by the algorithm to estimate the response function(s) from. For any *dwi2response* algorithm, this can be done by adding the *-voxels* option, which outputs an image of these voxels. For example, for the *tournier* algorithm:

```
dwi2response tournier dwi.mif wm_response.txt -voxels voxels.mif
```

The resulting *voxels.mif* image can be overlaid on the *dwi.mif* dataset using the *mrview* image viewer for further inspection.

5.12.2 Available algorithms

The available algorithms differ in a few general properties, related to what they deliver (as output) and require (as input), notably

- **single- versus multi-tissue:** whether they only estimate a single-fibre white matter response function (*tournier*, *tax* and *fa*) or also additional response functions for other tissue types (*dhollander* and *msmt_5tt* both output a single-fibre white matter response function as well as grey matter and CSF response functions)
- **single versus multiple b-values:** whether they only output response function(s) for a single b-value (*tournier*, *tax* and *fa*) or for all—or a selection of— b-values (*dhollander* and *msmt_5tt*)
- **input requirements:** whether they only require the DWI dataset as input (*tournier*, *dhollander*, *tax* and *fa*) or also additional input(s) (*msmt_5tt* requires a 5TT segmentation from a spatially aligned anatomical image)

Beyond these general categories, the algorithms differ mostly in the actual strategy used to determine the voxels that will be used to estimate the response function(s) from.

The *manual* choice is an exception to most of the above, in that it allows/*requires* you to provide the voxels yourself, and even allows you to provide single-fibre orientations manually as well. It should only be considered in case of exceptional kinds of data, or otherwise exceptional requirements. Caution is advised with respect to *interpretation* of spherical deconvolution results using manually defined response function(s).

The following sections provide more details on each algorithm specifically.

dhollander

This algorithm is the official implementation of the strategy proposed in [Dhollander2016b] (including improvements proposed in [Dhollander2019]) to estimate multi b-value (single-shell + b=0, or multi-shell) response functions for single-fibre white matter (*anisotropic*), grey matter and CSF (both *isotropic*), which can subsequently be used for multi-tissue (constrained) spherical deconvolution algorithms. It has the distinct advantage of requiring *only* the DWI data as input, in contrast to other multi-tissue response function estimation methods, making it the simplest and most accessible method, and a sensible default for applications that require multi-tissue responses.

This is a fully automated unsupervised algorithm that leverages the relative diffusion properties of the 3 tissue response functions with respect to each other, across all b-values and the angular domain, to select the most appropriate voxels from which to estimate the response functions. It has been used successfully in a wide range of conditions (overall data quality, pathology, developmental state of the subjects, animal data and ex-vivo data). Additional insights into its performance are presented in [Dhollander2018a]. Due to its ability to deal with the presence of extensive white matter (hyperintense) lesions, it was for example also successfully used in [Mito2018a]. The response functions as obtained in this particular way also form the basis of the 3-tissue framework to study the microstructure of lesions and other pathology [Dhollander2017] [Mito2018b].

The algorithm has been further improved in [Dhollander2019]. While the 2016 version identified the voxels to estimate the single-fibre white matter response function using the *tournier* algorithm, the new 2019 version relies on a novel strategy that optimises these voxels using properties of the signal across all b-values (and the full angular domain). It's also faster than the original approach.

In almost all cases, the algorithm runs and performs well out of the box. In *exceptional* cases where the anisotropy in the data is particularly *low* (very early development, ex-vivo data, (with) low b-value, ...), it is *sometimes* advisable to set the *-fa* parameter *lower* than its default value of 0.2. See [Dhollander2018b] for a good example of a dataset where changing this parameter was required to obtain good results. This FA threshold should be set so as to roughly separate the bulk of WM from the rest (GM and CSF). Further imperfections are corrected by the algorithm itself during a later stage.

As always, check the *-voxels* option output in unusually (challenging) cases.

For more information, refer to the *dhollander algorithm documentation*.

fa

This algorithm is an implementation of the strategy proposed in [Tournier2013] to estimate a single b-value (single-shell) response function of single-fibre white matter, which can subsequently be used for single-tissue (constrained) spherical deconvolution. The algorithm estimates this response function from the 300 voxels with the highest FA value in an eroded brain mask. There are also options to change this number or provide an absolute FA threshold.

Due to relying *only* on FA values, this strategy is relatively limited in its abilities to select the best voxels. In white matter close to CSF, for example, Gibbs ringing can affect FA values. More advanced iterative strategies, such as the *tournier* and *tax* algorithms have been proposed more recently.

For more information, refer to the *fa algorithm documentation*.

manual

This algorithm is provided for cases where none of the available automated algorithms give adequate results, for deriving multi-shell multi-tissue response functions in cases where the voxel mask for each tissue must be defined manually, or for anyone who may find it useful if trying to devise their own mechanism for response function estimation. It requires manual definition of both the single-fibre voxel mask (or just a voxel mask for isotropic tissues); the fibre directions can also be provided manually if necessary (otherwise a tensor fit will be used).

For more information, refer to the *manual algorithm documentation*.

msmt_5tt

This algorithm is a reimplementation of the strategy proposed in [Jeurissen2014] to estimate multi b-value response functions of single-fibre white matter (*anisotropic*), grey matter and CSF (both *isotropic*), which can subsequently be used for multi-tissue (constrained) spherical deconvolution. The algorithm is primarily driven by a prior (*The 5TT format*) tissue segmentation, typically obtained from a spatially aligned anatomical image. This also requires prior correction for susceptibility-induced (EPI) distortions of the DWI dataset. The algorithm selects voxels with a segmentation partial volume of at least 0.95 for each tissue type. Grey matter and CSF are further constrained by an (upper) 0.2 FA threshold. Single-fibre voxels within the WM segment are then extracted using the *tournier* algorithm (in contrast to original publication, see *Replicating original publications* below).

The input tissue segmentation can be estimated using the same *pre-processing pipeline* as required for *Anatomically-Constrained Tractography (ACT)*, namely: correction for motion and (EPI and other) distortions present in the diffusion MR data, registration of the structural to (corrected) EPI data, and spatial segmentation of the anatomical image. This process is therefore also dependent on the accuracy of each of these steps, so that the T1 image can be reliably used to select pure-tissue voxels in the DWI volumes. Failure to achieve high accuracy for each of these individual steps may result in inappropriate voxels being used for response function estimation, with concomitant errors in tissue estimates.

The *dhollander* algorithm does not rely on a number of these steps. A comparison is presented in [Dhollander2018a].

For further information, refer to the *msmt_5tt algorithm documentation*.

tax

This algorithm is a reimplementation of the iterative approach proposed in [Tax2014] to estimate a single b-value (single-shell) response function of single-fibre white matter, which can subsequently be used for single-tissue (constrained) spherical deconvolution. The algorithm iterates between performing CSD and estimating a response function from all voxels detected as being ‘single-fibre’ from the CSD result itself. The criterion for a voxel to be ‘single-fibre’ is based on the ratio of the amplitude of second tallest to the tallest peak. The method is initialised with a ‘fat’ response function; i.e., a response function that is safely deemed to be much less ‘sharp’ than the true response function.

This algorithm has occasionally been found to be unstable and converge towards suboptimal solutions. The *tourner* algorithm has been engineered with the intention to overcome some of the issues believed to be the cause of these instabilities (see some discussion on this topic [here](#) and [here](#)).

For more information, refer to the *tax algorithm documentation*.

tourner

This algorithm is a reimplementation of the iterative approach proposed in [Tournier2013] to estimate a single b-value (single-shell) response function of single-fibre white matter, which can subsequently be used for single-tissue (constrained) spherical deconvolution. The algorithm iterates between performing CSD and estimating a response function from a set of the best ‘single-fibre’ voxels, as detected from the CSD result itself. Notable differences between this implementation and the algorithm described in [Tournier2013] include:

- This implementation is initialised by a sharp $l_{\max}=4$ response function as opposed to one estimated from the 300 brain voxels with the highest FA.
- This implementation uses a more complex metric to measure how ‘single-fibre’ FODs are: $|peak1| \times (1 - |peak2| / |peak1|)^2$, as opposed to a simple ratio of the two tallest peaks. This new metric has a bias towards FODs with a larger tallest peak, to avoid favouring small, yet low SNR, FODs.
- This implementation only performs CSD on the 3000 best ‘single-fibre’ voxels (of the previous iteration) at each iteration.

While the *tourner* algorithm has a similar iterative structure as the *tax* algorithm, it was adjusted with the intention to overcome some occasional instabilities and suboptimal solutions resulting from the latter. Notable differences between the *tourner* and *tax* algorithms include:

- The *tourner* algorithm is initialised by a *sharp* ($l_{\max}=4$) response function, while the *tax* algorithm is initialised by a *fat* response function.
- This implementation of the *tourner* algorithm uses a more complex metric to measure how ‘single-fibre’ FODs are (see above), while the *tax* algorithm uses a simple ratio of the two tallest peaks.
- The *tourner* algorithm estimates the response function at each iteration only from the 300 *best* ‘single-fibre’ voxels, while the *tax* algorithm uses *all* ‘single-fibre’ voxels.

Due to these differences, the *tourner* algorithm is currently believed to be more robust in a wider range of scenarios (for further information on this topic, refer to some of the discussions [here](#) and [here](#)).

For more information, refer to the *tourner algorithm documentation*.

5.12.3 Replicating original publications

For completeness, we provide below instructions for replicating the approaches used in previous relevant publications. Note that the implementations provided below are not necessarily *exactly* as published, but aim to be close approximations nonetheless.

Spherical deconvolution and Constrained spherical deconvolution

In the original spherical deconvolution [Tournier2004] and constrained spherical deconvolution [Tournier2007] papers, the response function was estimated by extracting the 300 voxels with the highest FA values within a brain mask, eroded to avoid noisy voxels near the edge of the brain. This can be performed using the *fa* method directly:

```
dwi2response fa dwi.mif response.txt
```

where:

- `dwi.mif` is the input DWI data set,
- `response.txt` is the estimated response function, produced as output

MSMT-CSD and Global tractography

In the original multi-shell multi-tissue CSD [Jeurissen2014] and global tractography [Christiaens2015] papers, response functions were estimated using a prior tissue segmentation obtained from a coregistered structural T1 scan. For the WM response, a further hard FA threshold was used: respectively 0.7 in the MSMT-CSD paper and 0.75 in the global tractography paper. This pipeline can be replicated using the *5ttgen* command and *msmt_5tt* algorithm with the `-sfwm_fa_threshold` option in this fashion:

```
5ttgen fsl T1.mif 5tt.mif
dwi2response msmt_5tt dwi.mif 5tt.mif wm_response.txt gm_response.txt csf_response.
↪txt -sfwm_fa_threshold 0.7
```

where:

- `T1.mif` is a coregistered T1 data set from the same subject (input)
- `5tt.mif` is the resulting tissue type segmentation, used subsequently used in the response function estimation (output/input)
- `dwi.mif` is the same dwi data set as used above (input)
- `<tissue>_response.txt` is the tissue-specific response function as used above (output)

To replicate the global tractography paper, specify a value of 0.75 instead of 0.7 as shown in the command line above.

5.13 Constrained spherical deconvolution

5.13.1 Introduction

Constrained Spherical Deconvolution (CSD) [Tournier2007] estimates a white matter fibre Orientation Distribution Function (fODF) based on an estimate of the signal expected for a single-fibre white matter population (the so-called *response function*). This is used as the kernel in a deconvolution operation to extract a white matter fODF from dMRI signal measured within each voxel.

5.13.2 User guide

Prerequisites

Constrained Spherical Deconvolution as defined in [Tournier2007] relies on *single-shell* high angular resolution diffusion imaging (HARDI) data, containing at least one non-zero *b*-value. Ideally, the *b*-value used should be in the region of 2,500 – 3,000 s/mm² (at least for *in vivo* human brains), although good results have sometimes been obtained using *b* = 1000 s/mm² data.

In addition, this command expects that a suitable *single-shell single-tissue response function* has already been computed. Please refer to the [Response function estimation](#) page for details.

Invocation

Constrained Spherical Deconvolution can be performed as:

```
dwi2fod csd dwi.mif response.txt fod.mif
```

where:

- `dwi.mif` is the dwi data set (input)
- `response.txt` is the response function (input)
- `fod.mif` is the resulting fODF (output)

Typically, you will also want to use the `-mask` option to avoid unnecessary computations in non-brain voxels:

```
dwi2fod csd -mask mask.mif dwi.mif response.txt fod.mif
```

The resulting WM fODFs can be displayed together with the mean fODF amplitude map using:

```
mrview fod.mif -odf.load_sh fod.mif
```

5.14 Multi-shell multi-tissue constrained spherical deconvolution

5.14.1 Introduction

Multi-Shell Multi-Tissue Constrained Spherical Deconvolution (MSMT-CSD) exploits the unique b -value dependencies of the different macroscopic tissue types (WM/GM/CSF) to estimate a multi-tissue orientation distribution function (ODF) as explained in [Jeurissen2014]. As it includes separate compartments for each tissue type, it can produce a map of the WM/GM/CSF signal contributions directly from the DW data. In addition, the more complete modelling of the DW signal results in more accurate apparent fiber density (AFD) measures and more precise fibre orientation estimates at the tissue interfaces.

5.14.2 User guide

Prerequisites

MSMT-CSD relies on *multi-shell* high angular resolution diffusion imaging (HARDI) data, containing multiple b -values. The number of tissue types that can be resolved is limited by the number of b -values in the data (including $b=0$). To resolve the three primary tissue types in the brain (WM, GM & CSF), the acquisition should contain at least 2 shells along with the $b=0$ volumes (i.e. 3 unique b -values).

In addition, this command expects that suitable *multi-shell multi-tissue response functions* have already been computed. A number of approaches are available for this, please refer to the [Response function estimation](#) page for details.

Invocation

Multi-shell multi-tissue CSD can be performed as:

```
dwi2fod msmt_csd dwi.mif wm_response.txt wmfod.mif gm_response.txt gm.mif csf_
↪response.txt csf.mif
```

where:

- `dwi.mif` is the dwi data set (input)
- `<tissue>_response.txt` is the tissue-specific response function (input)
- `<tissue>.mif` is the tissue-specific ODF (output), typically full FODs for WM and a single scalars for GM and CSF

Note that input response functions and their corresponding output ODFs need to be specified in pairs.

Typically, you will also want to use the `-mask` option to avoid unnecessary computations in non-brain voxels:

```
dwi2fod msmt_csd -mask mask.mif dwi.mif wm_response.txt wmfod.mif gm_response.txt gm.  
↪mif csf_response.txt csf.mif
```

RGB tissue signal contribution maps can be obtained as follows:

```
mrconvert -coord 3 0 wm.mif - | mrcat csf.mif gm.mif - vf.mif
```

The resulting WM FODs can be displayed together with the tissue signal contribution map as:

```
mrview vf.mif -odf.load_sh wm.mif
```

5.15 Fibre density and cross-section - Single-tissue CSD

5.15.1 Introduction

This tutorial explains how to perform fixel-based analysis of fibre density and cross-section [Raffelt2017] using *single-tissue* spherical deconvolution. We note that high b-value (>2000s/mm²) data is recommended to aid the interpretation of apparent fibre density (AFD) being related to the intra-axonal space. See [Raffelt2012] for more details about *single-tissue* AFD.

All steps in this tutorial are written as if the commands are being **run on a cohort of images**, and make extensive use of the *for_each script to simplify batch processing*. This tutorial also assumes that the imaging dataset is organised with one directory identifying each subject, and all files within identifying the image type (i.e. processing step outcome). For example:

```
study/subjects/001_control/dwi.mif  
study/subjects/002_control/dwi.mif  
...  
study/subjects/020_control/dwi.mif  
study/subjects/021_patient/dwi.mif  
...  
study/subjects/040_patient/dwi.mif
```

Note: All commands at the start of this tutorial are run **from the subjects path**. From the step where tractography is performed on the template onwards, we change directory **to the template path**.

For all MRtrix scripts and commands, additional information on the command usage and available command-line options can be found by invoking the command with the `-help` option.

5.15.2 Pre-processsing steps

1. Denoising and unringing

If denoising and/or Gibbs ringing removal are performed as part of the preprocessing, they *must* be performed *prior* to any other processing steps: most other processing steps, in particular those that involve interpolation of the data, will invalidate the original properties of the image data that are exploited by *dwidenoise* and *mrdegibbs* at this stage, and would render the result prone to errors.

If denoising is included, it's performed as the first step:

```
for_each * : dwidenoise IN/dwi.mif IN/dwi_denoised.mif
```

If Gibbs ringing removal is included, it follows immediately after:

```
for_each * : mrdegibbs IN/dwi_denoised.mif IN/dwi_denoised_unringed.mif -axes 0,1
```

Warning: The `-axes` option to *mrdegibbs* is used to specify in which plane the slices were acquired. The `-axes 0,1` in the example above refers to the x-y plane, which is appropriate for data consisting of a stack of *axial* slices (assuming a typical human scanner and subject). For typical human data, change this to `-axes 0,2` for *coronal* slices or `-axes 1,2` for *sagittal* slices.

2. Motion and distortion correction

The *dwifslpreproc* command handles motion and distortion correction for DWI data (including eddy current distortions and optionally susceptibility-induced EPI distortions). Even though the command works seamlessly like any other *MRtrix3* command, it is in fact a script that interfaces with the *FSL* package to perform most of its core functionality and algorithms. For this command to work, *FSL* (including *eddy*) needs to be installed. Also remember to cite the relevant articles with respect to the specific algorithms (see the *dwifslpreproc* help page).

The simplest scenario is to (only) correct for motion and eddy current-induced distortions:

```
for_each * : dwifslpreproc IN/dwi_denoised_unringed.mif IN/dwi_denoised_unringed_
↳preproc.mif -rpe_none -pe_dir AP
```

Warning: The `-pe_dir` option to *dwifslpreproc* is used to specify the phase encoding direction of the acquisition. The `-pe_dir AP` in the example above refers to an *anterior-posterior* phase encoding direction, which is relatively commonly used for acquiring human data. For typical human data, change this to `-pe_dir LR` for a *left-right* phase encoding direction or `-pe_dir SI` for a *superior-inferior* phase encoding direction.

For different, more advanced, scenarios and acquisitions (e.g. the common case of correcting for susceptibility-induced EPI distortions using a pair of reverse phase-encoded b=0 images), refer to the *DWI distortion correction using dwifslpreproc* section and the *dwifslpreproc* documentation.

3. Estimate a temporary brain mask

Compute a brain mask:

```
for_each * : dwi2mask IN/dwi_denoised_preproc.mif IN/dwi_temp_mask.mif
```


5.15.3 AFD-specific pre-processing steps

To enable robust quantitative comparisons of AFD across subjects three additional steps are required. Note these can potentially be skipped if analysing *certain* other DWI fixel-based measures related to fibre density (for example CHARMED).

4. Bias field correction

Because we recommend a [global intensity normalisation](#), bias field correction is required as a pre-processing step to eliminate low frequency intensity inhomogeneities across the image. DWI bias field correction is performed by first estimating the bias field from the DWI b=0 data, then applying the field to correct all DW volumes. This can be done in a single step using the `ants` algorithm within the `dwibiascorrect` script in *MRtrix3*. The script uses a bias field correction algorithm available in *ANTS* (the N4 algorithm). *Don't* use the `fsl` algorithm with this script in this fixel-based analysis pipeline. To perform bias field correction on DW images, run:

```
for_each * : dwibiascorrect ants IN/IN/dwi_denoised_unringed_preproc.mif IN/IN/dwi_
↪denoised_unringed_preproc_unbiased.mif
```

5. Global intensity normalisation across subjects

As outlined [here](#), a global intensity normalisation is required for AFD analysis. For a single-tissue pipeline, a possible approach is to use the `dwinormalise group` script. The script performs normalisation on all subjects within a study (using a group-wise registration), and therefore the input and output arguments are directories containing all study images. First create directories to store all the input and output images. From the subjects directory:

```
mkdir -p ../dwinormalise/dwi_input
mkdir ../dwinormalise/mask_input
```

You could copy all files into this directory, however symbolic linking them will save space:

```
for_each * : ln -sr IN/dwi_denoised_unringed_preproc_unbiased.mif ../dwinormalise/dwi_
↪input/IN.mif
for_each * : ln -sr IN/dwi_temp_mask.mif ../dwinormalise/mask_input/IN.mif
```

Perform group DWI intensity normalisation:

```
dwinormalise group ../dwinormalise/dwi_input/ ../dwinormalise/mask_input/ ../
↪dwinormalise/dwi_output/ ../dwinormalise/fa_template.mif ../dwinormalise/fa_
↪template_wm_mask.mif
```

Link the output files back to the subject directories:

```
for_each ../dwinormalise/dwi_output/* : ln -sr IN PRE/dwi_denoised_unringed_preproc_
↪unbiased_normalised.mif
```

The `dwinormalise group` script also outputs the study-specific FA template and white matter mask. **It is recommended that you check that the white matter mask is appropriate** (i.e. does not contain CSF or voxels external to the brain. It needs to be a rough WM mask). If you feel the white matter mask needs to be larger or smaller you can re-run `dwinormalise group` with a different `-fa_threshold` option. Note that if your input brain masks include CSF then this can cause spurious high FA values outside the brain which are then included in the template white matter mask.

Keeping the FA template image and white matter mask is also handy if additional subjects are added to the study at a later date. New subjects can be intensity normalised in a single step by [piping](#) the following commands together. Run from the subjects directory:


```
dwi2tensor new_subject/dwi_denoised_unringed_preproc_unbiased.mif -mask new_subject/
↪dwi_temp_mask.mif - | tensor2metric - -fa - | mrregister -force ../dwinormalise/fa_
↪template.mif - -mask2 new_subject/dwi_temp_mask.mif -nl_scale 0.5,0.75,1.0 -nl_
↪niter 5,5,15 -nl_warp - /tmp/dummy_file.mif | mrtransform ../dwinormalise/fa_
↪template_wm_mask.mif -template new_subject/dwi_denoised_unringed_preproc_unbiased.
↪mif -warp - - | dwinormalise new_subject/dwi_denoised_unringed_preproc_unbiased.mif_
↪- ../dwinormalise/dwi_output/new_subject.mif
```

Note: The above command may also be useful if you wish to alter the mask and re-apply the intensity normalisation to all subjects in the study. For example, you may wish to edit the mask using the ROI tool in `mrview` to *remove* white matter regions that you hypothesise are affected by the disease (e.g. removing the corticospinal tract in a study of motor neurone disease due to T2 hyperintensity). You also may wish to redefine the mask entirely, for example in an elderly population (with enlarged ventricles) it may be possible, or even preferable, to normalise using the median $b=0$ CSF. This could be performed by manually masking partial-volume-free CSF voxels, then running the above command with the CSF mask instead of the `fa_template_wm_mask.mif`.

Warning: We strongly recommend you that you check the scale factors applied during intensity normalisation are not influenced by the variable of interest in your study. For example if one group contains global changes in white matter T2 then this may directly influence the intensity normalisation and therefore bias downstream AFD analysis. To check this, you can perform an equivalence test to ensure mean scale factors are the same between groups. To output the scale factor applied for all subjects use `mrinfo ../dwinormalise/dwi_output/* -property dwi_norm_scale_factor`.

5.15.4 Fixel-based analysis steps

6. Computing an (average) white matter response function

A robust and fully automated (unsupervised) method to obtain single-shell response functions representing single-fibre white matter from your data, is the approach proposed in [Tournier2013], which can be run by:

```
for_each * : dwi2response tournier IN/dwi_denoised_unringed_preproc_unbiased_
↪normalised.mif IN/response.txt
```

It is crucial for fixel-based analysis to only use a single *unique* response function to perform spherical deconvolution of all subjects: as all resulting fibre orientation distributions will be expressed in function of it, it can (in an abstract way) be seen as the unit of the final apparent fibre density metric. A possible way to obtain a unique response function, is to average the response functions obtained from all subjects:

```
responsemean */response.txt ../group_average_response.txt
```

There is however no strict requirement for the (one) final response function to be the average of *all* subject response functions. In certain very specific cases, it may even be wise to leave out subjects (for this step) where a response function could not reliably be obtained, or where pathology affected the brain globally.

7. Upsampling DW images

Upsampling DWI data *before* computing FODs can increase anatomical contrast and improve downstream template building, registration, tractography and statistics. We recommend upsampling to an isotropic voxel size of 1.25 mm for human brains (if your original resolution is already higher, you can skip this step):

```
for_each * : mrgrid IN/dwi_denoised_unringed_preproc_unbiased_normalised.mif regrid -
↳vox 1.25 IN/dwi_denoised_unringed_preproc_unbiased_normalised_upsampled.mif
```

8. Compute upsampled brain mask images

Compute a whole brain mask from the upsampled DW images:

```
for_each * : dwi2mask IN/dwi_denoised_unringed_preproc_unbiased_normalised_upsampled.
↳mif IN/dwi_mask_upsampled.mif
```

Warning: It is absolutely **crucial** to check at this stage that *all* individual subject masks include *all* regions of the brain that are intended to be analysed. Fibre orientation distributions will *only* be computed within these masks; and at a later step (in template space) the analysis mask will be restricted to the *intersection* of all masks, so *any* individual subject mask which excludes a certain region, will result in this region being excluded from the entire analysis (unless a more advanced pipeline is followed; see [Mitigating the effects of brain cropping](#)). Masks appearing too generous or otherwise including non-brain regions should generally not cause any concerns at this stage. Hence, if in doubt, it is advised to always err on the side of *inclusion* (of regions) at this stage. Manually correct the masks if necessary.

9. Fibre Orientation Distribution estimation (spherical deconvolution)

When performing fixel-based analysis, constrained spherical deconvolution (CSD) should be performed using the unique (average) white matter response function obtained before. Note that `dwi2fod csd` can be used, however here we use `dwi2fod msmt_csd` (even with single shell data) to benefit from the hard non-negativity constraint, which has been observed to lead to more robust outcomes:

```
for_each * : dwiextract IN/dwi_denoised_unringed_preproc_unbiased_normalised_
↳upsampled.mif - \ dwi2fod msmt_csd - ../group_average_response.txt IN/wmfod.mif -
↳mask IN/dwi_mask_upsampled.mif
```

10. Generate a study-specific unbiased FOD template

Population template creation is one of the most time consuming steps in a fixel-based analysis. If you have a very large number of subjects in your study, you can opt to create the template from a limited subset of 30-40 individuals. Typically, subjects are chosen so the generated template is representative of your population (e.g. similar number of patients and controls, though avoid patients with excessive abnormalities compared to the rest of the population). To build a template, put all FOD images in a single folder and put a set of corresponding mask images (with the same prefix as the FOD images) in another folder (using masks speeds up registration significantly):

```
mkdir -p ../template/fod_input
mkdir ../template/mask_input
```

Symbolic link all FOD images (and masks) into a single input folder. To use the entire population to build the template:

```
for_each * : ln -sr IN/wmfod.mif ../template/fod_input/PRE.mif
for_each * : ln -sr IN/dwi_mask_upsampled.mif ../template/mask_input/PRE.mif
```

If you opt to create the template from a limited subset of (e.g. 30-40) subjects and your study has multiple groups, then you can aim for a similar number of subjects from each group to make the template more representative of the

population as a whole. Assuming the subject directory labels can be used to identify members of each group, you could use:

```
for_each `ls -d *patient | sort -R | tail -20` : ln -sr IN/wmfod.mif ../template/fod_
↪input/PRE.mif ";" ln -sr IN/dwi_mask_upsampled.mif ../template/mask_input/PRE.mif
for_each `ls -d *control | sort -R | tail -20` : ln -sr IN/wmfod.mif ../template/fod_
↪input/PRE.mif ";" ln -sr IN/dwi_mask_upsampled.mif ../template/mask_input/PRE.mif
```

Run the template building script as follows:

```
population_template ../template/fod_input -mask_dir ../template/mask_input ../
↪template/wmfod_template.mif -voxel_size 1.25
```

The voxel size is typically set to match the voxel size of the input FOD images (which, in this pipeline, would typically be the resolution the preprocessed data was upsampled to, earlier on in the pipeline).

11. Register all subject FOD images to the FOD template

Register the FOD image from each subject to the FOD template:

```
for_each * : mrregister IN/wmfod.mif -mask1 IN/dwi_mask_upsampled.mif ../template/
↪wmfod_template.mif -nl_warp IN/subject2template_warp.mif IN/template2subject_warp.
↪mif
```

12. Compute the template mask (intersection of all subject masks in template space)

Different subjects have different brain coverage. To ensure subsequent analysis is performed in voxels that contain data from all subjects, we warp all subject masks into template space and compute the template mask as the intersection of all subject masks in template space. To warp all masks into template space:

```
for_each * : mrtransform IN/dwi_mask_upsampled.mif -warp IN/subject2template_warp.mif_
↪-interp nearest -datatype bit IN/dwi_mask_in_template_space.mif
```

Compute the template mask as the intersection of all warped masks:

```
mrmath */dwi_mask_in_template_space.mif min ../template/template_mask.mif -datatype_
↪bit
```

Warning: It is absolutely **crucial** to check at this stage that the resulting template mask includes *all* regions of the brain that are intended to be analysed. If this is not the case, the cause will be either an individual subject mask which did not include a certain region, or the template building process or individual subject registrations having gone wrong for one or more subjects. It is advised to **go back** to these steps, and identify and resolve the cause of the problem before continuing any further.

Note: It is possible at this stage to edit the template mask and remove certain regions that are either not part of the brain (though these are unlikely to survive the intersection step to begin with, as they would have to be present in *all* subject masks for this to happen), or otherwise not of interest. However, **do not add regions to the mask** at this stage. If there is a genuine need to do this, go back to the relevant steps which caused the exclusion of these regions to begin with (see also the above warning).

13. Compute a white matter template analysis fixel mask

In this step, we segment fixels from the FOD template. The result is the *fixel mask* that defines the fixels for which statistical analysis will later on be performed (and hence also which fixels' statistics can support others via the mechanism of connectivity-based fixel enhancement (CFE) [Raffelt2015]):

```
fod2fixel -mask ../template/template_mask.mif -fmls_peak_value 0.10 ../template/wmfod_
↳template.mif ../template/fixel_mask
```

Note: Fixel images, which appear in the pipeline from this step onwards, are stored using the *Fixel image (directory) format*, which stores all fixel data for a fixel image in a directory (i.e. a folder).

Warning: This step ultimately determines the fixel mask in which statistical analysis will be performed, and hence also which fixels' statistics can contribute to others via the CFE mechanism; so it may have a substantial impact on the final result. Essentially, it can be detrimental to the result if the threshold value specified via the `-fmls_peak_value` is *too high* and hence *excludes* genuine white matter fixels. This risk is substantially higher in voxels containing crossing fibres (and higher the more fibres are crossing in a single voxel). Even though 0.1 has been observed to be a reasonable value for single-tissue CSD population templates, it is still **strongly advised** to visualise the output fixel mask using *mrview*. Do this by opening the `index.mif` found in `../template/fixel_mask` via the *fixel plot tool*. If, with respect to known or normal anatomy, fixels are missing (especially paying attention to crossing areas), regenerate the mask with a lower value supplied to the `-fmls_peak_value` option. At the same time, try to avoid introducing too many noisy fixels, especially in gray matter areas. For single-tissue CSD, this balance is however not always easy to find. For an *adult human* brain template, and using an isotropic template voxel size of 1.25 mm, it is expected to have several *hundreds of thousands* of fixels in the fixel mask (you can check this by `mrinfo -size ../template/fixel_mask/directions.mif`, and looking at the size of the image along the first dimension).

14. Warp FOD images to template space

Note that here we warp FOD images into template space *without* FOD reorientation, as reorientation will be performed in a separate subsequent step (after fixel segmentation):

```
for_each * : mrtransform IN/wmfod.mif -warp IN/subject2template_warp.mif -reorient_
↳fod no IN/fod_in_template_space_NOT_REORIENTED.mif
```

15. Segment FOD images to estimate fixels and their apparent fibre density (FD)

Here we segment each FOD lobe to identify the number and orientation of fixels in each voxel. The output also contains the apparent fibre density (AFD) value per fixel (estimated as the FOD lobe integral):

```
for_each * : fod2fixel -mask ../template/template_mask.mif IN/fod_in_template_space_
↳NOT_REORIENTED.mif IN/fixel_in_template_space_NOT_REORIENTED -afd fd.mif
```

Note that in the following steps we will use the more generic shortened acronym Fibre Density (FD) to refer to the AFD metric.

16. Reorient fixels

Here we reorient the fixels of all subjects in template space based on the local transformation at each voxel in the warps used previously:

```
for_each * : fixelreorient IN/fixel_in_template_space_NOT_REORIENTED IN/
↳subject2template_warp.mif IN/fixel_in_template_space
```

After this step, the `fixel_in_template_space_NOT_REORIENTED` folders can be safely removed.

17. Assign subject fixels to template fixels

While each subject's data has already been (spatially) warped to the common template space, and subject fixels have been reoriented accordingly, there is still no specification of which fixels match (across subjects, and between the subject and template fixels). This step establishes exactly that, by matching the fixels of each individual subject to the single common set of template fixels (which then inherently also defines how they match across subjects). This is achieved by, for each fixel in the template fixel mask, identifying the corresponding fixel in the matching voxel of the subject image and assigning the FD value of this corresponding subject fixel to that fixel in template space. If no fixel exists or can be found in a subject that corresponds to a given template fixel then it is assigned a value of zero (as the absence of a subject fixel at this stage is most likely due to a very low, or even zero, FD). This step is performed as follows:

```
for_each * : fixelcorrespondence IN/fixel_in_template_space/fd.mif ../template/fixel_
↳mask ../template/fd PRE.mif
```

Note that the output fixel directory `../template/fd` is the same for all subjects. This makes sense, since after this operation, there is only a single remaining set of fixels (i.e. the template fixels), with corresponding FD values as obtained from each subject. This resulting directory `../template/fd` now stores these data as individual fixel data files: one for each subject, and all with respect to a single set of corresponding template fixels. This way of storing the entire population's FD data is then ready for input to `fixelcfestats` later on.

18. Compute the fibre cross-section (FC) metric

The fibre density metric, mapped directly without any modulation to the fixel template space as above, is only sensitive to the original density of intra-axonal space in each voxel. In other words, it ignores the cross-sectional size of the bundle, which is another property that would factor into the bundle's total intra-axonal space across its full cross-sectional extent, and hence influence its total capacity to carry information. In certain cases, for example, atrophy may impact this cross-sectional size, but not per se the local fibre density metric.

In this step, we compute a fixel-based metric related to morphological differences in fibre cross-section (FC), where information is derived entirely from the warps generated during registration (see [\[Raffelt2017\]](#) for more information):

```
for_each * : warp2metric IN/subject2template_warp.mif -fc ../template/fixel_mask ../
↳template/fc IN.mif
```

However, for group statistical analysis of FC we recommend calculating the $\log(\text{FC})$ to ensure data are centred around zero and normally distributed. Here, we create a separate fixel directory to store the $\log(\text{FC})$ data and copy the fixel index and directions file across:

```
mkdir ../template/log_fc
cp ../template/fc/index.mif ../template/fc/directions.mif ../template/log_fc
for_each * : mrcalc ../template/fc/IN.mif -log ../template/log_fc/IN.mif
```

Note: The FC (and hence also the $\log(\text{FC})$) as calculated here, is a *relative* metric, expressing the local fixel-wise cross-sectional size *relative* to this study's population template. While this makes it possible to interpret differences of FC *within* a single study (because only a single unique template is used in the study), the FC values should not be compared across different studies that each have their own population template. Reporting absolute quantities of FC, or absolute effect sizes of FC, also provides little information; as again, it is only meaningful with respect to the template.

19. Compute a combined measure of fibre density and cross-section (FDC)

The total capacity of a fibre bundle to carry information, is modulated both by the local fibre density at the voxel (fixel) level, as well as its cross-sectional size. Here we compute a combined metric, which factors in the effects of both FD and FC, resulting in a fibre density and cross-section (FDC) metric:

```
mkdir ../template/fdc
cp ../template/fc/index.mif ../template/fdc
cp ../template/fc/directions.mif ../template/fdc
for_each * : mrcalc ../template/fd/IN.mif ../template/fc/IN.mif -mult ../template/fdc/
↪IN.mif
```

This is also a nice example of how calculations across multiple fixel data files can be performed. However, note that this is only valid if these both share the same set of original fixels (in this case, the template fixel mask). Because the fixels also have to be stored in *exactly* the same order for this to work correctly, great care has to be taken the `index.mif` files (in this case of the `../template/fd` and `../template/fc` folders) related to all input fixel data files that are used in the `mrcalc` command are *exact copies* of each other.

20. Perform whole-brain fibre tractography on the FOD template

Statistical analysis using connectivity-based fixel enhancement (CFE) [Raffelt2015] exploits local connectivity information derived from probabilistic fibre tractography, which acts as a neighbourhood definition for threshold-free enhancement of locally clustered statistic values. To generate a whole-brain tractogram from the FOD template (note the remaining steps from here on are executed from the template directory):

```
cd ../template
tckgen -angle 22.5 -maxlen 250 -minlen 10 -power 1.0 wmfod_template.mif -seed_image_
↪template_mask.mif -mask template_mask.mif -select 20000000 -cutoff 0.10 tracks_20_
↪million.tck
```

Warning: The appropriate FOD amplitude cutoff for FOD template tractography can vary considerably between different datasets, as well as different versions of *MRtrix3* due to historical software bugs. While the value of 0.10 is suggested as a reasonable value for single-tissue data, it may be beneficial to first generate a smaller number of streamlines (e.g. 100,000) using this value, and visually confirm that the generated streamlines exhibit an appropriate extent of propagation at the ends of white matter pathways, before committing to generation of the dense tractogram.

21. Reduce biases in tractogram densities

Perform SIFT to reduce tractography biases in the whole-brain tractogram:

```
tcksift tracks_20_million.tck wmfd_template.mif tracks_2_million_sift.tck -term_
↪number 2000000
```

22. Generate fixel-fixel connectivity matrix

Generation of the fixel-fixel connectivity matrix based on the whole-brain streamlines tractogram is performed as follows:

```
fixelconnectivity fixel_mask/ tracks_2_million_sift.tck matrix/
```

The output directory should contain three images: *index.mif*, *fixels.mif* and *values.mif*; these are used to encode the fixel-fixel connectivity that is by its nature sparse.

Warning: Running `fixelconnectivity` requires quite a lot of memory (for about 500,000 fixels in the template analysis fixel mask and a typical tractogram defining the pairwise connectivity between fixels, 32GB of RAM is a typical memory requirement; you can check the number of fixels in your template analysis fixel mask by `mrinfo -size ./fixel_template/directions.mif`, and looking at the size of the image along the first dimension; also check for (and avoid) gross false positive connections in the tractogram, e.g. streamlines connecting both hemispheres in large areas where they shouldn't). Should hardware related limitations force you to reduce the number of fixels, it is *not* advised to change the threshold to derive the white matter template analysis fixel mask, as it may remove crucial fixels deep in the white matter (e.g. in crossing areas). Rather, consider slightly increasing the template voxel size, or spatially removing regions that are not of interest from the template mask (which was earlier obtained in this pipeline as the intersections of all subject masks in template space).

23. Smooth fixel data using fixel-fixel connectivity

Smoothing of fixel data is performed based on the sparse fixel-fixel connectivity matrix:

```
fixelfilter fd smooth fd_smooth -matrix matrix/
fixelfilter log_fc smooth log_fc_smooth -matrix matrix/
fixelfilter fdc smooth fdc_smooth -matrix matrix/
```

By calling the *fixelfilter* command on each fixel directory in turn, the smoothing filter will be applied to all fixel data files present in each of those directories; it is not necessary to call the *fixelfilter* command separately for each individual fixel data file.

24. Perform statistical analysis of FD, FC, and FDC

Statistical analysis using CFE is performed separately for each metric (FD, log(FC), and FDC) as follows:

```
fixelcfestats fd_smooth/ files.txt design_matrix.txt contrast_matrix.txt matrix/_
↪stats_fd/
fixelcfestats log_fc_smooth/ files.txt design_matrix.txt contrast_matrix.txt matrix/_
↪stats_log_fc/
fixelcfestats fdc_smooth/ files.txt design_matrix.txt contrast_matrix.txt matrix/_
↪stats_fdc/
```

The input `files.txt` is a text file containing the filename of each file (i.e. *not* the full path) to be analysed inside the input fixel directory, each filename on a separate line. The line ordering should correspond to the lines in the file `design_matrix.txt`.

Note: While previous versions of *fixelcfestats* performed smoothing of the input fixel data as those data were imported, this is *no longer the case*. It is expected that the fixel data provided as input to *fixelcfestats* will already have been appropriately smoothed; e.g. using *fixelfilter*.

Note: Unlike some other software packages providing a GLM (e.g. FSL *randomise*), a column of 1's (corresponding to the “global intercept”: the mean image value when all other design matrix factors are zero) will *not be automatically included*. If the model for your experiment requires such a factor, it is necessary to include it explicitly in the design matrix.

25. Visualise the results

To view the results load the population FOD template image in *mrview*, and overlay the fixel images using the vector plot tool. Note that *p*-value images are saved as $(1 - p\text{-value})$. Therefore to visualise all results at a threshold of $p < 0.05$, within the *mrview* fixel plot tool, apply a *lower* threshold at a value of 0.95.

5.16 Fibre density and cross-section - Multi-tissue CSD

5.16.1 Introduction

This tutorial explains how to perform fixel-based analysis of fibre density and cross-section [Raffelt2017] with fibre orientation distributions (FODs) computed using multi-tissue (3-tissue) CSD variants [Jeurissen2014] [Dhollander2016a]. We note that high b-value ($>2000\text{s/mm}^2$) data is recommended to aid the interpretation of apparent fibre density (AFD) being related to the intra-axonal space. See [Raffelt2012] for some details about AFD; though note that the interpretation can be altered for multi-tissue (3-tissue) CSD, depending on the context and tissues in the model.

All steps in this tutorial are written as if the commands are being **run on a cohort of images**, and make extensive use of the *for_each script to simplify batch processing*. This tutorial also assumes that the imaging dataset is organised with one directory identifying each subject, and all files within identifying the image type (i.e. processing step outcome). For example:

```
study/subjects/001_control/dwi.mif
study/subjects/002_control/dwi.mif
...
study/subjects/020_control/dwi.mif
study/subjects/021_patient/dwi.mif
...
study/subjects/040_patient/dwi.mif
```

Note: All commands at the start of this tutorial are run **from the subjects path**. From the step where tractography is performed on the template onwards, we change directory **to the template path**.

For all MRtrix scripts and commands, additional information on the command usage and available command-line options can be found by invoking the command with the `-help` option.

5.16.2 Pre-processing steps

1. Denoising and unringing

If denoising and/or Gibbs ringing removal are performed as part of the preprocessing, they *must* be performed *prior* to any other processing steps: most other processing steps, in particular those that involve interpolation of the data, will invalidate the original properties of the image data that are exploited by *dwidenoise* and *mrdegibbs* at this stage, and would render the result prone to errors.

If denoising is included, it's performed as the first step:

```
for_each * : dwidenoise IN/dwi.mif IN/dwi_denoised.mif
```

If Gibbs ringing removal is included, it follows immediately after:

```
for_each * : mrdegibbs IN/dwi_denoised.mif IN/dwi_denoised_unringed.mif -axes 0,1
```

Warning: The `-axes` option to *mrdegibbs* is used to specify in which plane the slices were acquired. The `-axes 0,1` in the example above refers to the x-y plane, which is appropriate for data consisting of a stack of *axial* slices (assuming a typical human scanner and subject). For typical human data, change this to `-axes 0,2` for *coronal* slices or `-axes 1,2` for *sagittal* slices.

2. Motion and distortion correction

The *dwifslpreproc* command handles motion and distortion correction for DWI data (including eddy current distortions and optionally susceptibility-induced EPI distortions). Even though the command works seamlessly like any other *MRtrix3* command, it is in fact a script that interfaces with the *FSL* package to perform most of its core functionality and algorithms. For this command to work, *FSL* (including *eddy*) needs to be installed. Also remember to cite the relevant articles with respect to the specific algorithms (see the *dwifslpreproc* help page).

The simplest scenario is to (only) correct for motion and eddy current-induced distortions:

```
for_each * : dwifslpreproc IN/dwi_denoised_unringed.mif IN/dwi_denoised_unringed_
↳preproc.mif -rpe_none -pe_dir AP
```

Warning: The `-pe_dir` option to *dwifslpreproc* is used to specify the phase encoding direction of the acquisition. The `-pe_dir AP` in the example above refers to an *anterior-posterior* phase encoding direction, which is relatively commonly used for acquiring human data. For typical human data, change this to `-pe_dir LR` for a *left-right* phase encoding direction or `-pe_dir SI` for a *superior-inferior* phase encoding direction.

For different, more advanced, scenarios and acquisitions (e.g. the common case of correcting for susceptibility-induced EPI distortions using a pair of reverse phase-encoded b=0 images), refer to the *DWI distortion correction using dwifslpreproc* section and the *dwifslpreproc* documentation.

3. Bias field correction

The multi-tissue FBA pipeline corrects for bias fields (and jointly performs global intensity normalisation) at the later *mtnormalise* step. The only incentive for running the (less robust and accurate) *dwibiascorrect* at this stage in the pipeline is to *improve brain mask estimation* (at the later *dwi2mask* step, in case severe bias fields are present in the data). However, cases have been reported where running *dwibiascorrect* at this stage resulted in *inferior* brain mask estimation later on. This is probably more likely in case bias fields are not as strongly present in the data. Whether *dwibiascorrect* is run at this stage or not, does not have any significant impact on the performance of *mtnormalise* later on.

If or when performing DWI bias field correction at this stage, it is achieved by first estimating the bias field from the DWI $b=0$ data, then applying the field to correct all DW volumes, which is done in a single step using the `ants` algorithm within the `dwibiascorrect` script in *MRtrix3*. The script uses a bias field correction algorithm available in *ANTs* (the N4 algorithm). *Don't* use the `fsl` algorithm with this script in this fixel-based analysis pipeline. To perform bias field correction on DW images, run:

```
for_each * : dwibiascorrect ants IN/dwi_denoised_unringed_preproc.mif IN/dwi_denoised_
↳unringed_preproc_unbiased.mif
```

5.16.3 Fixel-based analysis steps

4. Computing (average) tissue response functions

A robust and fully automated unsupervised method to obtain 3-tissue response functions representing single-fibre white matter, grey matter and CSF from the data itself, is the approach proposed in [Dhollander2016b] with the improvements of [Dhollander2019], which can be run by:

```
for_each * : dwi2response dhollander IN/dwi_denoised_unringed_preproc_unbiased.mif IN/
↳response_wm.txt IN/response_gm.txt IN/response_csf.txt
```

It is crucial for fixel-based analysis to only use a single *unique* set of the (three) response functions to perform (3-tissue) spherical deconvolution of all subjects: as the (3-tissue) spherical deconvolution results will be expressed in function of this set of response functions, they can (in an abstract way) be seen as the units of both the final apparent fibre density metric and the other compartments estimated in the model. One possible way to obtain a unique set of response functions, is to average the response functions obtained from all subjects for each tissue type:

```
responsemean */response_wm.txt ../group_average_response_wm.txt
responsemean */response_gm.txt ../group_average_response_gm.txt
responsemean */response_csf.txt ../group_average_response_csf.txt
```

There is however no strict requirement for the final set of response functions to be the average of *all* subject response functions, for each tissue type (or indeed, it doesn't even have to be the average per se). In certain very specific cases, it may even be wise to leave out subjects (for this step) where the response functions could not reliably be obtained, or where pathology affected the brain globally.

5. Upsampling DW images

Upsampling DWI data *before* computing FODs increases anatomical contrast and improves downstream template building, registration, tractography and statistics. We recommend upsampling to an isotropic voxel size of 1.25 mm for human brains (if your original resolution is already higher, you can skip this step):

```
for_each * : mrgrid IN/dwi_denoised_unringed_preproc_unbiased.mif regrid -vox 1.25 IN/
↳dwi_denoised_unringed_preproc_unbiased_upsampled.mif
```

6. Compute upsampled brain mask images

Compute a whole brain mask from the upsampled DW images:

```
for_each * : dwi2mask IN/dwi_denoised_unringed_preproc_unbiased_upsampled.mif IN/dwi_
↳mask_upsampled.mif
```

Warning: It is absolutely **crucial** to check at this stage that *all* individual subject masks include *all* regions of the brain that are intended to be analysed. Fibre orientation distributions will *only* be computed within these masks; and at a later step (in template space) the analysis mask will be restricted to the *intersection* of all masks, so *any* individual subject mask which excludes a certain region, will result in this region being excluded from the entire analysis (unless a more advanced pipeline is followed; see [Mitigating the effects of brain cropping](#)). Masks appearing too generous or otherwise including non-brain regions should generally not cause any concerns at this stage. Hence, if in doubt, it is advised to always err on the side of *inclusion* (of regions) at this stage.

Note: The earlier [dwibiascorrect](#) step is not fundamentally important in the multi-tissue fixel-based analysis pipeline, as the later [mtnormalise](#) step performs more robustly (and if [dwibiascorrect](#) is included, [mtnormalise](#) will later on typically improve the result further). While performing the earlier [dwibiascorrect](#) step typically improves [dwi2mask](#) performance, cases have been observed where the opposite is true (typically if the data contains only weak bias fields). If required, experiment by either including or excluding [dwibiascorrect](#) in the pipeline in function of the best [dwi2mask](#) outcome and manually correct the masks if necessary (by *adding* regions which [dwi2mask](#) fails to include).

7. Fibre Orientation Distribution estimation (multi-tissue spherical deconvolution)

When performing fixel-based analysis, multi-tissue constrained spherical deconvolution should be performed using the unique set of (average) tissue response functions obtained before:

```
for_each * : dwi2fod msmt_csd IN/dwi_denoised_unringed_preproc_unbiased_upsampled.mif _
↪ ../group_average_response_wm.txt IN/wmfod.mif ../group_average_response_gm.txt IN/
↪ gm.mif ../group_average_response_csf.txt IN/csf.mif -mask IN/dwi_mask_upsampled.mif
```

8. Joint bias field correction and intensity normalisation

To perform joint bias field correction and global intensity normalisation of the multi-tissue compartment parameters, use [mtnormalise](#):

```
for_each * : mtnormalise IN/wmfod.mif IN/wmfod_norm.mif IN/gm.mif IN/gm_norm.mif IN/
↪ csf.mif IN/csf_norm.mif -mask IN/dwi_mask_upsampled.mif
```

If multi-tissue CSD was performed with the same single set of (three) tissue response functions for all subjects, then the resulting output of [mtnormalise](#) makes the absolute amplitudes comparable between those subjects as well. Note that this step is **crucial** in the FBA pipeline, even if bias field correction was applied earlier using [dwibiascorrect](#), since [dwibiascorrect](#) does *not* correct for *global* intensity differences between subjects. The performance of [mtnormalise](#) is not significantly impacted by either having run [dwibiascorrect](#) before or not. In case prior bias field correction was run in the pipeline, [mtnormalise](#) will further correct for residual intensity inhomogeneities.

Warning: [mtnormalise](#) results can be sensitive to masks that contain non-brain voxels. The underlying algorithm will attempt to drive the sum of tissue volumes to unity in such voxels - despite not containing brain tissue - which can result in erroneous bias field correction if the number of such voxels is large. For this reason we recommend using conservative (i.e. less spatially extended) masks for the [mtnormalise](#) step. Unlike step 6, where inclusion of all brain voxels was encouraged even at the expense of including some non-brain voxels, for bias field estimation exclusion of non-brain voxels is of greater priority than inclusion of all brain voxels.

9. Generate a study-specific unbiased FOD template

Population template creation is one of the most time consuming steps in a fixel-based analysis. If you have a very large number of subjects in your study, you can opt to create the template from a limited subset of 30-40 individuals. Typically, subjects are chosen so the generated template is representative of your population (e.g. similar number of patients and controls, though avoid patients with excessive abnormalities compared to the rest of the population). To build a template, put all FOD images in a single folder and put a set of corresponding mask images (with the same prefix as the FOD images) in another folder (using masks speeds up registration significantly):

```
mkdir -p ../template/fod_input
mkdir ../template/mask_input
```

Symbolic link all FOD images (and masks) into a single input folder. To use the entire population to build the template:

```
for_each * : ln -sr IN/wmfod_norm.mif ../template/fod_input/PRE.mif
for_each * : ln -sr IN/dwi_mask_upsampled.mif ../template/mask_input/PRE.mif
```

If you opt to create the template from a limited subset of (e.g. 30-40) subjects and your study has multiple groups, then you can aim for a similar number of subjects from each group to make the template more representative of the population as a whole. Assuming the subject directory labels can be used to identify members of each group, you could use:

```
for_each `ls -d *patient | sort -R | tail -20` : ln -sr IN/wmfod_norm.mif ../template/
↪fod_input/PRE.mif ";" ln -sr IN/dwi_mask_upsampled.mif ../template/mask_input/PRE.
↪mif
for_each `ls -d *control | sort -R | tail -20` : ln -sr IN/wmfod_norm.mif ../template/
↪fod_input/PRE.mif ";" ln -sr IN/dwi_mask_upsampled.mif ../template/mask_input/PRE.
↪mif
```

Run the template building script as follows:

```
population_template ../template/fod_input -mask_dir ../template/mask_input ../
↪template/wmfod_template.mif -voxel_size 1.25
```

The voxel size is typically set to match the voxel size of the input FOD images (which, in this pipeline, would typically be the resolution the preprocessed data was upsampled to, earlier on in the pipeline).

10. Register all subject FOD images to the FOD template

Register the FOD image from each subject to the FOD template:

```
for_each * : mrregister IN/wmfod_norm.mif -mask1 IN/dwi_mask_upsampled.mif ../
↪template/wmfod_template.mif -nl_warp IN/subject2template_warp.mif IN/
↪template2subject_warp.mif
```

11. Compute the template mask (intersection of all subject masks in template space)

Different subjects have different brain coverage. To ensure subsequent analysis is performed in voxels that contain data from all subjects, we warp all subject masks into template space and compute the template mask as the intersection of all subject masks in template space. To warp all masks into template space:

```
for_each * : mrtransform IN/dwi_mask_upsampled.mif -warp IN/subject2template_warp.mif_
↪-interp nearest -datatype bit IN/dwi_mask_in_template_space.mif
```

Compute the template mask as the intersection of all warped masks:

```
mrmath */dwi_mask_in_template_space.mif min ../template/template_mask.mif -datatype_
↪bit
```

Warning: It is absolutely **crucial** to check at this stage that the resulting template mask includes *all* regions of the brain that are intended to be analysed. If this is not the case, the cause will be either an individual subject mask which did not include a certain region, or the template building process or individual subject registrations having gone wrong for one or more subjects. It is advised to **go back** to these steps, and identify and resolve the cause of the problem before continuing any further.

Note: It is possible at this stage to edit the template mask and remove certain regions that are either not part of the brain (though these are unlikely to survive the intersection step to begin with, as they would have to be present in *all* subject masks for this to happen), or otherwise not of interest. However, **do not add regions to the mask** at this stage. If there is a genuine need to do this, go back to the relevant steps which caused the exclusion of these regions to begin with (see also the above warning).

12. Compute a white matter template analysis fixel mask

In this step, we segment fixels from the FOD template. The result is the *fixel mask* that defines the fixels for which statistical analysis will later on be performed (and hence also which fixels' statistics can support others via the mechanism of connectivity-based fixel enhancement (CFE) [Raffelt2015]):

```
fod2fixel -mask ../template/template_mask.mif -fmls_peak_value 0.06 ../template/wmfod_
↪template.mif ../template/fixel_mask
```

Note: Fixel images, which appear in the pipeline from this step onwards, are stored using the *Fixel image (directory) format*, which stores all fixel data for a fixel image in a directory (i.e. a folder).

Warning: This step ultimately determines the fixel mask in which statistical analysis will be performed, and hence also which fixels' statistics can contribute to others via the CFE mechanism; so it may have a substantial impact on the final result. Essentially, it can be detrimental to the result if the threshold value specified via the `-fmls_peak_value` is *too high* and hence *excludes* genuine white matter fixels. This risk is substantially higher in voxels containing crossing fibres (and higher the more fibres are crossing in a single voxel). Even though 0.06 has been observed to be a decent default value for 3-tissue CSD population templates, it is still **strongly advised** to visualise the output fixel mask using *mrview*. Do this by opening the `index.mif` found in `../template/fixel_mask` via the *fixel plot tool*. If, with respect to known or normal anatomy, fixels are missing (especially paying attention to crossing areas), regenerate the mask with a lower value supplied to the `-fmls_peak_value` option (of course, avoid lowering it *too* much, as too many false or noisy fixels may be introduced). For an *adult human* brain template, and using an isotropic template voxel size of 1.25 mm, it is expected to have several *hundreds of thousands* of fixels in the fixel mask (you can check this by `mrinfo -size ../template/fixel_mask/directions.mif`, and looking at the size of the image along the first dimension).

13. Warp FOD images to template space

Note that here we warp FOD images into template space *without* FOD reorientation, as reorientation will be performed in a separate subsequent step (after fixel segmentation):

```
for_each * : mrtransform IN/wmfod_norm.mif -warp IN/subject2template_warp.mif -  
↪reorient_fod no IN/fod_in_template_space_NOT_REORIENTED.mif
```

14. Segment FOD images to estimate fixels and their apparent fibre density (FD)

Here we segment each FOD lobe to identify the number and orientation of fixels in each voxel. The output also contains the apparent fibre density (AFD) value per fixel (estimated as the FOD lobe integral):

```
for_each * : fod2fixel -mask ../template/template_mask.mif IN/fod_in_template_space_  
↪NOT_REORIENTED.mif IN/fixel_in_template_space_NOT_REORIENTED -afd fd.mif
```

Note that in the following steps we will use the more generic shortened acronym Fibre Density (FD) to refer to the AFD metric.

15. Reorient fixels

Here we reorient the fixels of all subjects in template space based on the local transformation at each voxel in the warps used previously:

```
for_each * : fixelreorient IN/fixel_in_template_space_NOT_REORIENTED IN/  
↪subject2template_warp.mif IN/fixel_in_template_space
```

After this step, the `fixel_in_template_space_NOT_REORIENTED` folders can be safely removed.

16. Assign subject fixels to template fixels

While each subject's data has already been (spatially) warped to the common template space, and subject fixels have been reoriented accordingly, there is still no specification of which fixels match (across subjects, and between the subject and template fixels). This step establishes exactly that, by matching the fixels of each individual subject to the single common set of template fixels (which then inherently also defines how they match across subjects). This is achieved by, for each fixel in the template fixel mask, identifying the corresponding fixel in the matching voxel of the subject image and assigning the FD value of this corresponding subject fixel to that fixel in template space. If no fixel exists or can be found in a subject that corresponds to a given template fixel then it is assigned a value of zero (as the absence of a subject fixel at this stage is most likely due to a very low, or even zero, FD). This step is performed as follows:

```
for_each * : fixelcorrespondence IN/fixel_in_template_space/fd.mif ../template/fixel_  
↪mask ../template/fd PRE.mif
```

Note that the output fixel directory `../template/fd` is the same for all subjects. This makes sense, since after this operation, there is only a single remaining set of fixels (i.e. the template fixels), with corresponding FD values as obtained from each subject. This resulting directory `../template/fd` now stores these data as individual fixel data files: one for each subject, and all with respect to a single set of corresponding template fixels. This way of storing the entire population's FD data is then ready for input to `fixelcfestats` later on.

17. Compute the fibre cross-section (FC) metric

The fibre density metric, mapped directly without any modulation to the fixel template space as above, is only sensitive to the original density of intra-axonal space in each voxel. In other words, it ignores the cross-sectional size of the bundle, which is another property that would factor into the bundle's total intra-axonal space across its full cross-sectional extent, and hence influence its total capacity to carry information. In certain cases, for example, atrophy may impact this cross-sectional size, but not per se the local fibre density metric.

In this step, we compute a fixel-based metric related to morphological differences in fibre cross-section (FC), where information is derived entirely from the warps generated during registration (see [Raffelt2017] for more information):

```
for_each * : warp2metric IN/subject2template_warp.mif -fc ../template/fixel_mask ../
↳template/fc IN.mif
```

However, for group statistical analysis of FC we recommend calculating the $\log(\text{FC})$ to ensure data are centred around zero and normally distributed. Here, we create a separate fixel directory to store the $\log(\text{FC})$ data and copy the fixel index and directions file across:

```
mkdir ../template/log_fc
cp ../template/fc/index.mif ../template/fc/directions.mif ../template/log_fc
for_each * : mrcalc ../template/fc/IN.mif -log ../template/log_fc/IN.mif
```

Note: The FC (and hence also the $\log(\text{FC})$) as calculated here, is a *relative* metric, expressing the local fixel-wise cross-sectional size *relative* to this study's population template. While this makes it possible to interpret differences of FC *within* a single study (because only a single unique template is used in the study), the FC values should not be compared across different studies that each have their own population template. Reporting absolute quantities of FC, or absolute effect sizes of FC, also provides little information; as again, it is only meaningful with respect to the template.

18. Compute a combined measure of fibre density and cross-section (FDC)

The total capacity of a fibre bundle to carry information, is modulated both by the local fibre density at the voxel (fixel) level, as well as its cross-sectional size. Here we compute a combined metric, which factors in the effects of both FD and FC, resulting in a fibre density and cross-section (FDC) metric:

```
mkdir ../template/fdc
cp ../template/fc/index.mif ../template/fdc
cp ../template/fc/directions.mif ../template/fdc
for_each * : mrcalc ../template/fd/IN.mif ../template/fc/IN.mif -mult ../template/fdc/
↳IN.mif
```

This is also a nice example of how calculations across multiple fixel data files can be performed. However, note that this is only valid if these both share the same set of original fixels (in this case, the template fixel mask). Because the fixels also have to be stored in *exactly* the same order for this to work correctly, great care has to be taken the `index.mif` files (in this case of the `../template/fd` and `../template/fc` folders) related to all input fixel data files that are used in the `mrcalc` command are *exact copies* of each other.

19. Perform whole-brain fibre tractography on the FOD template

Statistical analysis using connectivity-based fixel enhancement (CFE) [Raffelt2015] exploits local connectivity information derived from probabilistic fibre tractography, which acts as a neighbourhood definition for threshold-free enhancement of locally clustered statistic values. To generate a whole-brain tractogram from the FOD template (note the remaining steps from here on are executed from the template directory):

```
cd ../template
tckgen -angle 22.5 -maxlen 250 -minlen 10 -power 1.0 wmfod_template.mif -seed_image_
↳template_mask.mif -mask template_mask.mif -select 20000000 -cutoff 0.06 tracks_20_
↳million.tck
```


Warning: The appropriate FOD amplitude cutoff for FOD template tractography can vary considerably between different datasets, as well as different versions of *MRtrix3* due to historical software bugs. While the value of 0.06 is suggested as a reasonable value for multi-tissue data, it may be beneficial to first generate a smaller number of streamlines (e.g. 100,000) using this value, and visually confirm that the generated streamlines exhibit an appropriate extent of propagation at the ends of white matter pathways, before committing to generation of the dense tractogram.

20. Reduce biases in tractogram densities

Perform SIFT to reduce tractography biases in the whole-brain tractogram:

```
tcksift tracks_20_million.tck wmfod_template.mif tracks_2_million_sift.tck -term_
↪number 2000000
```

21. Generate fixel-fixel connectivity matrix

Generation of the fixel-fixel connectivity matrix based on the whole-brain streamlines tractogram is performed as follows:

```
fixelconnectivity fixel_mask/ tracks_2_million_sift.tck matrix/
```

The output directory should contain three images: *index.mif*, *fixels.mif* and *values.mif*; these are used to encode the fixel-fixel connectivity that is by its nature sparse.

Warning: Running `fixelconnectivity` requires quite a lot of memory (for about 500,000 fixels in the template analysis fixel mask and a typical tractogram defining the pairwise connectivity between fixels, 32GB of RAM is a typical memory requirement; you can check the number of fixels in your template analysis fixel mask by `mrinfo -size ./fixel_template/directions.mif`, and looking at the size of the image along the first dimension; also check for (and avoid) gross false positive connections in the tractogram, e.g. streamlines connecting both hemispheres in large areas where they shouldn't). Should hardware related limitations force you to reduce the number of fixels, it is *not* advised to change the threshold to derive the white matter template analysis fixel mask, as it may remove crucial fixels deep in the white matter (e.g. in crossing areas). Rather, consider slightly increasing the template voxel size, or spatially removing regions that are not of interest from the template mask (which was earlier obtained in this pipeline as the intersections of all subject masks in template space).

22. Smooth fixel data using fixel-fixel connectivity

Smoothing of fixel data is performed based on the sparse fixel-fixel connectivity matrix:

```
fixelfilter fd smooth fd_smooth -matrix matrix/
fixelfilter log_fc smooth log_fc_smooth -matrix matrix/
fixelfilter fdc smooth fdc_smooth -matrix matrix/
```

By calling the *fixelfilter* command on each fixel directory in turn, the smoothing filter will be applied to all fixel data files present in each of those directories; it is not necessary to call the *fixelfilter* command separately for each individual fixel data file.

23. Perform statistical analysis of FD, FC, and FDC

Statistical analysis using CFE is performed separately for each metric (FD, log(FC), and FDC) as follows:

```
fixelcfestats fd_smooth/ files.txt design_matrix.txt contrast_matrix.txt matrix/_
↪stats_fd/
fixelcfestats log_fc_smooth/ files.txt design_matrix.txt contrast_matrix.txt matrix/_
↪stats_log_fc/
fixelcfestats fdc_smooth/ files.txt design_matrix.txt contrast_matrix.txt matrix/_
↪stats_fdc/
```

The input `files.txt` is a text file containing the filename of each file (i.e. *not* the full path) to be analysed inside the input fixel directory, each filename on a separate line. The line ordering should correspond to the lines in the file `design_matrix.txt`.

Note: While previous versions of `fixelcfestats` performed smoothing of the input fixel data as those data were imported, this is *no longer the case*. It is expected that the fixel data provided as input to `fixelcfestats` will already have been appropriately smoothed; e.g. using `fixelfilter`.

Note: Unlike some other software packages providing a GLM (e.g. FSL `randomise`), a column of 1's (corresponding to the “global intercept”: the mean image value when all other design matrix factors are zero) will *not be automatically included*. If the model for your experiment requires such a factor, it is necessary to include it explicitly in the design matrix.

24. Visualise the results

To view the results load the population FOD template image in `mrview`, and overlay the fixel images using the vector plot tool. Note that p -value images are saved as $(1 - p\text{-value})$. Therefore to visualise all results at a threshold of $p < 0.05$, within the `mrview` fixel plot tool, apply a *lower* threshold at a value of 0.95.

5.17 Fixel image (directory) format

Images for representing discrete multi-fibre models are sparse in nature (i.e. different voxels may have different numbers of fibre populations - a.k.a “*fixels*”, as described in the “*Fixels*” (and “*Dixels*”) page), and different models have different parameter requirements per fixel (e.g. orientation, volume fraction, fanning, tensors etc). This fixel image format overcomes several issues in storing such data in either traditional 4D images or a custom format (such as the legacy *Legacy MRtrix Sparse Format* (`.msh / .msf`)).

5.17.1 Requirements

This new format has been designed with the following requirements in mind:

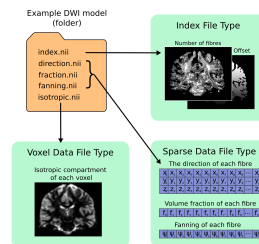
- **Space saving.** Because different voxels may have different numbers of fixels, it is inefficient to store data using 4-dimensional images, since the size of the 4th dimension must accommodate the voxel with the highest number of fixels. A sparse representation on disk is therefore more efficient.
- **Easily read and written** by other software packages to enable inter-operability of fixel-based DWI models.

- **Flexible** enough to allow for both fixel-specific model parameters (e.g. volume fractions, fanning), and voxel-specific parameters (e.g. hindered isotropic compartment). The format should also support any number of model parameters.
- **Self-documenting.** Users should be able to easily infer what kind of data is included in the model. Developers should also easily understand the data layout, without having to read in special fields in the image header.
- **Minimise the need for supporting commands.** We wanted to avoid the need to have dedicated commands for performing basic operations on the data (e.g. math/calculator operations, thresholding, histogram generation etc).
- **Extendability.** Users should be able to add components to an existing sparse image. E.g. a mask to label fixels of interest, or additional test-statistic output from a group analysis.

5.17.2 Specifications

In the fixel format we have opted to leverage the file system by storing data belonging to a single sparse DWI model inside a single *directory/folder* (in contrast to the old *Legacy MRtrix Sparse Format (.msh / .msf)* where all data is stored inside a single *file*). Effectively the directory becomes the ‘dataset’. While this implies that all data files must be kept together inside the directory, and can be tampered with (or accidentally deleted) by users, we believe the transparency and accessibility of the data is beneficial and enables all of the above requirements to be met.

All files types saved inside the format are in either **NIfTI-2 format** (for maximum compatibility with other packages) or **MRtrix image formats (.mih / .mif)**. To help describe the format and the layout of the files within the directory, we have used an example of how a ball and racket-like model may be stored:



5.17.3 Fixel format file types

Index File

- 4D image ($i \times j \times k \times 2$).
- The index file is required, with fixed naming (`index.nii` or `index.mif`).
- The first 3D volume in the 4th dimension stores the *number of elements (fixels)* per voxel.
- The second volume in the 4th dimension stores the *index of the first element (fixel)* in that voxel; indices for the subsequent elements in each voxel are inferred to be sequential.

Fixel Data File

- 3D image ($n \times p \times 1$) where n is the total number of elements in the image, and p is the number of parameters per element (e.g. 3 for `direction.nii`, 1 for `volume.nii`, or 6 for a multi-tensor model).
- For each voxel, data for the elements within that voxel must be stored within sequential indices in the first dimension.

- Easily identified as a data file type because the size of the image is 1 in the 3rd dimension
- Any number of Fixel Data File types may be present in the directory. In the example image above, the volume fraction and fanning angle parameters have been saved as separate files; however the format is flexible and allows for multiple parameters, p, per element.
- Any naming convention can be used for Fixel Data Files, with the exception of:
 - The directions file (see below).
 - If a particular set of commands expect to write and subsequently read one or more data files with a fixed name, then manually renaming such files may prevent the operation of that set of commands.

Note: The number of fixels in a whole-brain image can be very large (> 100,000). The NIfTI-1 format therefore cannot be used in this context, as it restricts the total number of voxels along any dimension of the image to 65,535. This is why either NIfTI-2 or *MRtrix image formats (.mih / .mif)* must be used.

Fixel Direction File

- **All fixel-based DWI models must specify the direction of each fixel.**
- Directions for each fixel must be saved within a single file named either `directions.nii` or `directions.mif`.
- This can be considered as a special type of fixel data file, with dimensions (n x 3 x 1).
- Directions must be specified with respect to the *scanner coordinate frame*, in *cartesian coordinates*.

Voxel Data File

- 3D or 4D image
- Any number of Voxel Data Files may be stored in the directory
- Must have the same resolution and header transform as the index image
- Naming of files is flexible
- The 4th dimension is optional, but allows for multiple parameters per voxel to be stored (e.g. 6 tensor coefficients of the ‘hindered’ compartment in CHARMED)

5.17.4 Usage

Because the fixel format leverages the file system to store all fixel data within a single directory, interacting with fixel data in *MRtrix3* may require user input and output arguments to be either: 1) the path to the fixel format directory; or 2) specific fixel data files within the directory. For example, `fod2fixel` requires the name of the containing directory *and* the names of the output fixel data files to be stored inside the directory:

```
fod2fixel patient01/fod.mif patient01/fixel_directory -afd afd.mif -disp dispersion.
↪mif
```

In this example, a new fixel format directory is created, with the name `patient01_fixel_directory`; this includes creation of the requisite index and directions files. Fixel Data Files for two fixel-specific measures (AFD and dispersion) are then additionally saved inside this directory.

Other commands, such as `fixel2voxel`, may only require the fixel data file:

```
fixel2voxel patient01/fixel_directory/afd.mif sum patient01/total_afd.mif
```

A major benefit of the directory-based format is that existing commands for operating on traditional images can be used to manipulate fixel data. For example, to threshold fixels based on their AFD value:

```
mrthreshold patient01/fixel_directory/afd.mif -abs 0.1 patient01/fixel_directory/afd_
↪mask.mif
```

Other commands like `mrhistogram`, `mrcalc` and `mrstats` can also be used on Fixel Data Files. For example, to compute the mean dispersion over all fixels in a mask:

```
mrstats -output mean -mask patient01/fixel_directory/afd_mask.mif patient01/fixel_
↪directory/dispersion.mif
```

5.17.5 Viewing fixel data in mrview

Fixel data can be visualised using the “Fixel Plot” tool in `mrview`. Any image within the fixel directory can be opened by the file chooser. By default the fixels will be coloured based on the file selected when loaded (e.g. if you select the fixel directions file, fixels will be colour-coded by direction; if `afd.mif` is selected they will be coloured by AFD value). Irrespective of the file selected to view the fixel file, all other fixel file types in the fixel directory will be detected and available for use in colour-coding and thresholding fixels via a combo box in the Fixel Plot tool. This enables advanced visualisations such as thresholding fixels by p-value while colour-coding by effect size.

5.18 Mitigating the effects of brain cropping

In some datasets, the DWI images do not provide coverage of the entire brain cerebrum and cerebellum, due to poor placement of the imaging field of view and/or subject movement leading to that area of the brain shifting outside the FoV for at least one DWI volume. In such cases, it would be erroneous to infer a change in any fixel-wise metric in that area if subjects for which no valid image data are available were to contribute to the assessment. It is therefore necessary to properly track where each subject possesses valid image data and where they do not, and to modify the processing of and inference from such data accordingly.

The way to achieve this at the point of statistical inference is for the fixel data to contain the value NaN (Not A Number) in any location where a valid quantitative metric could not be obtained for that particular subject. Such data are removed from the GLM on a fixel-by-fixel basis. This technique is explained in greater detail and demonstrated in [Smith2019a].

The following instructions describe a way to modify to a typical Fixel-Based Analysis pipeline, in order to ensure that any location in the template image where image data for a particular subject may have been affected by such cropping will contain the value NaN. It is however recommended that you manually inspect the results of these processing steps in order to ensure that the manipulations of the data are operating as intended.

1. Replace zero-filled values in the DWI with NaN

When the FSL command `eddy` (invoked by *MRtrix3*’s *dwifslpreproc*) cannot reconstruct valid image data for all DWI volumes in a particular voxel, it fills that voxel with zero values. The following sequence of commands identifies such voxels, and replaces the values stored within those voxels with NaN:

```
mrmath dwi.mif norm - -axis 3 | mrthreshold - - -abs 0.0 -comparison gt -nan |
↪mrcalc dwi.mif - -mult dwi_nan.mif
```

Note that this command must be run *after* *dwifslpreproc*, but *before* any upsampling of the DWI data: the latter introduces an interpolation step, such that some voxels in the upsampled image will be decreased in intensity due to this effect, but will not be precisely zero.

If data upsampling is performed subsequent to this step, regions of the image containing these NaN values will become *larger*. This occurs because any voxels for which performing 3D interpolation will attempt to sample from an input voxel containing the value NaN will itself obtain a value of NaN.

2. Ignore other instructions elsewhere regarding brain masks

Filling voxels outside of the brain with values of NaN achieves a comparable effect to providing a brain mask: voxels containing such values will not contribute to various calculations just as though they were to lie outside of a provided brain mask. As such, explicitly providing a brain mask that does not exclude any voxels not already excluded by step 1 would not have any consequence.

3. Use NaN fill value in *mrtransform*

When transforming subject FOD data to template space, instruct the *mrtransform* command to fill voxels in template space outside of the input image FoV with the value NaN, rather than zeroes (using the `-nan` option); this will ensure that any voxel in template space for which valid subject data are not available will contain the value NaN, regardless of which step in the pipeline led to that fact.

4. Substitute template mask with number of valid subjects

Upon generation of the study-specific population template, the intersection of all subject brain masks in template space will *not* be utilised. Indeed it is not entirely appropriate to transform individual subjects' brain masks to template space, as the results of such would not reflect the propagation of NaN values described at the end of point 1.

It may however instead be useful to know, for each voxel in template space, how many subjects possess valid image data in that location:

```
for_each * : mrconvert IN/fod_in_template_space_NOT_REORIENTED.mif -coord 3 0 -
↪axes 0,1,2 - "|" mrcalc - -finite IN/valid_data_template_space_mask.mif -
↪datatype bit
mrmath */valid_data_template_space_mask.mif sum ../template/valid_data_num_
↪subjects.mif
```

It may then be useful to apply a threshold to this image (*mrthreshold*) in order to inform the derivation of a voxel mask for statistical inference; e.g. one may wish to exclude altogether from analysis those voxels with less than some number of subjects; this choice is left open to the researcher.

5. Propagate NaN values to fixel quantitative metrics

For voxels in template space for which no valid data are available for a particular subject, we want fixel data to contain the value NaN rather than 0.0. This is done by projecting the voxel mask representing those voxels for which valid subject data are available into the template fixel mask, and then modifying the fixel values accordingly:

```
mkdir ../template/valid_data_masks/
for_each * : voxel2fixel IN/valid_data_template_space_mask.mif ../template/fd/ ../
↪template/valid_data_masks/ PRE.mif

mkdir ../template/fd_nan/
cp ../template/fd/index.mif ../template/fd/directions.mif ../template/fd_nan/
for_each * : mrcalc ../template/fd/PRE.mif ../template/valid_data_masks/PRE.mif -
↪div ../template/fd_nan/PRE.mif
```

This is performed *after* the *fixelcorrespondence* step, and must be performed independently for each fixel metric of interest.

6. When running *fixelcfestats*, the presence of NaN values in the input data will be detected automatically, and this fact will be reported to the user.

This command should be expected to take approximately 4 to 5 times longer to complete than typical usage where all input data are finite.

5.19 Expressing the effect size relative to controls

The apparent Fibre Density (FD) and Fibre Density and Cross-section (FDC) are relative measures and have arbitrary units. Therefore the units of `abs_effect.mif` output from *fixelcfstats* are not directly interpretable. In a patient-control group comparison t-test, one way to present results is to express the absolute effect size as a percentage relative to the control group mean.

To compute FD and FDC percentage decrease effect size use:

```
mrcalc stats_fd/abs_effect.mif control_mean.mif -div 100 -mult stats_fd/percentage_
↪effect.mif
```

The mean value in the control group can be obtained in one of two ways:

1. Use the relevant beta coefficient image from *fixelcfstats*

If there is a column in the design matrix that contains the value 1 for all subjects in the control group and 0 for all other subjects, *and* if any & all nuisance regressors were de-meaned prior to inserting them into the design matrix, then the relevant beta coefficient image provided by *fixelcfstats* (e.g. `stats_fd/beta0.mif`) can be interpreted directly as the control group mean.

2. Explicitly calculate the control group mean

Using the smoothed data from *fixelfilter*, the *mrmath* command with the `sum` operation can be used to explicitly compute the mean across the set of images corresponding to the control group.

Because the Fibre Cross-section (FC) measure is a scale factor it is slightly more complicated to compute the percentage decrease. The FC ratio between two subjects (or groups) tells us the direct scale factor between them.

For example, for a given fixel if the patient group mean FC is 0.7, and control mean is 1.4, then this implies encompassing fibre tract in the patients is half as big as the controls: $0.7/1.4 = 0.5$. I.e. this is a 50% reduction wrt to the controls: $1 - (FC_{patients}/FC_{controls})$

Because we perform FBA of $\log(FC)$, the `abs_effect` that is output from *fixelcfstats* is: $abs_effect = \log(FC_{controls}) - \log(FC_{patients}) = \log(FC_{controls}/FC_{patients})$. Therefore to get the percentage effect we need to perform $1 - 1/\exp(abs_effect)$:

```
mrcalc 1 1 stats_fc/abs_effect.mif -exp -div -sub stats_log_fc/percentage_effect.mif
```

5.20 Displaying results with streamlines

Fixels rendered directly as lines using the *fixel plot tool* of *mrview* are appropriate for viewing 2D slices; however, to better appreciate all the fibre pathways affected and to visualise the full extent of the results in 3D, it's also possible to use a visualisation based on the whole-brain template-derived tractogram [Raffelt2017].

First use `tckedit` to reduce the whole-brain template tractogram to a sensible number of streamlines (2 million is quite a lot for certain graphics cards to render smoothly). This step assumes you have the same folder structure and filenames from the FBA tutorials. From the `template` directory:

```
tckedit tracks_2_million_sift.tck -num 200000 tracks_200k_sift.tck
```

Map fixel values to streamline points, save them in a “track scalar file”. For example:

```
fixel2tsf stats_fdc/fwe_pvalue.mif tracks_200k_sift.tck fdc_fwe_pvalue.tsf
fixel2tsf stats_fdc/abs_effect_size.mif tracks_200k_sift.tck fdc_abs_effect_size.tsf
```

Visualise track scalar files using the tractogram tool in MRview. First load the streamlines (`tracks_200k_sift.tck`). Then right click and select ‘colour by (track) scalar file’. For example you might load the `abs_effect_size.tsf` file. Then to dynamically threshold (remove) streamline points by p-value select the “Thresholds” dropdown and select “Separate Scalar file” to load `fwe_pvalue.tsf`.

Note that you can also threshold and view all brain fixels by deselecting “crop to slice” in the fixel plot tool. However it can be harder to appreciate the specific pathways affected. The downside to viewing and colouring results by streamline, then viewing all streamlines (uncropped to slice), is that without transparency you only see the colours on the outside of the significant pathways, where normally the effect size/p-value is most severe in the ‘core’ of the fibre pathway.

Good examples of both fixel as well as streamline visualisations of FBA results can be found in [Mito2018a].

5.21 Anatomically-Constrained Tractography (ACT)

This page describes the recommended processing steps for taking advantage of the Anatomically-Constrained Tractography (ACT) framework [Smith2012], the image format used, and the commands available for manipulating these data. There are also instructions for anyone looking to make use of alternative tissue segmentation approaches.

5.21.1 Pre-processing steps

DWI distortion correction

For the anatomical information to be incorporated accurately during the tractography reconstruction process, any geometric distortions present in the diffusion images must be corrected. The FSL 5.0 commands `topup` and `eddy` are effective in performing this correction based on a reversed phase-encode acquisition, though their interfaces can be daunting. We therefore provide a wrapper script, `dwifslpreproc`, which interfaces with these tools to perform correction of multiple forms of image distortion (motion, eddy current and inhomogeneity). Please read the *DWI distortion correction using dwifslpreproc* page, and the `dwifslpreproc` help page for further details.

Image registration

My personal preference is to register the T1-contrast anatomical image to the diffusion image series before any further processing of the T1 image is performed. By registering the T1 image to the diffusion series rather than the other way around, reorientation of the diffusion gradient table is not necessary; and by doing this registration before subsequent T1 processing, any subsequent images derived from the T1 are inherently aligned with the diffusion image series. This registration should be rigid-body only; if the DWI distortion correction is effective, a higher-order registration is likely to only introduce errors.

DWI pre-processing

Because the anatomical image is used to limit the spatial extent of streamlines propagation rather than a binary mask derived from the diffusion image series, I highly recommend dilating the DWI brain mask prior to computing FODs; this is to make sure that any errors in derivation of the DWI mask do not leave gaps in the FOD data within the brain white matter, and therefore result in erroneous streamlines termination.

Tissue segmentation

So far I have had success with using FSL tools to also perform the anatomical image segmentation; FAST is not perfect, but in most cases it's good enough, and most alternative software I tried provided binary mask images only, which is not ideal. The `5ttgen` script using the `fsl` algorithm interfaces with FSL to generate the necessary image data from the raw T1 image, using BET, FAST and FIRST. Note that this script also crops the resulting image so that it contains no more than the extracted brain (as this reduces the file size and therefore improves memory access performance during tractography); if you want the output image to possess precisely the same dimensions as the input T1 image, you can use the `-nocrop` option.

5.21.2 Using ACT

Once the necessary pre-processing steps are completed, using ACT is simple: just provide the tissue-segmented image to the `tckgen` command using the `-act` option.

In addition, since the propagation and termination of streamlines is primarily handled by the 5TT image, it is no longer necessary to provide a mask using the `-mask` option. In fact, for whole-brain tractography, it is recommended that you `_not_` provide such an image when using ACT: depending on the accuracy of the DWI brain mask, its inclusion may only cause erroneous termination of streamlines inside the white matter due to exiting this mask. If the mask encompasses all of the white matter, then its inclusion does not provide any additional information to the tracking algorithm.

5.21.3 The 5TT format

When the ACT framework is invoked, it expects the tissue information to be provided in a particular format; this is referred to as the 'five-tissue-type (5TT)' format. This is a 4D, 32-bit floating-point image, where the dimension of the fourth axis is 5; that is, there are five 3D volumes in the image. These five volumes correspond to the different tissue types. In all brain voxels, the sum of these five volumes should be 1.0, and outside the brain it should be zero. The tissue type volumes must appear in the following order for the anatomical priors to be applied correctly during tractography:

0. Cortical grey matter
1. Sub-cortical grey matter
2. White matter
3. CSF
4. Pathological tissue

The first four of these are described in the ACT NeuroImage paper. The fifth can be optionally used to manually delineate regions of the brain where the architecture of the tissue present is unclear, and therefore the type of anatomical priors to be applied are also unknown. For any streamline entering such a region, *no anatomical priors are applied* until the streamline either exists that region, or stops due to some other streamlines termination criterion.

The following binaries are provided for working with the 5TT format:

- `5tt2gmwmi`: Produces a mask image suitable for seeding streamlines from the grey matter - white matter interface (GMWMI). The resulting image should then be provided to the `tckgen` command using the `-seed_gmwmi` option.
- `5tt2vis`: Produces a 3D greyscale image suitable for visualisation purposes.
- `5ttcheck`: Check that one or more input images conform to the 5TT format.

- `5ttedit`: Allows the user to edit the tissue segmentations. Useful for manually correcting tissue segmentations that are known to be erroneous (e.g. dark blobs in the white matter being labelled as grey matter); see the command's help page for more details.

5.21.4 Alternative tissue segmentation software

Users who wish to experiment with using tissue segmentations from different software sources are encouraged to do so; if a particular approach is shown to be effective we can add an appropriate script to MRtrix. The `5ttgen` script has a second algorithm, `freesurfer`, which demonstrates how the output of different software can be manipulated to provide the tissue segmentations in the appropriate format. It is however not recommended to actually use this alternative algorithm for patient studies; many midbrain structures are not segmented by FreeSurfer, so the tracking may not behave as desired.

Users who wish to try manipulating the tissue segmentations from some alternative software into the 5TT format may find it most convenient to make a copy of one of the existing algorithms within the `lib/mrtrix3/_5ttgen/` directory, and modify accordingly. The `5ttgen` script will automatically detect the presence of the new algorithm, and make it available at the command-line.

5.22 Spherical-deconvolution Informed Filtering of Tractograms (SIFT)

SIFT [Smith2013], or ‘Spherical-deconvolution Informed Filtering of Tractograms’, is a novel approach for improving the quantitative nature of whole-brain streamlines reconstructions. By producing a reconstruction where the streamlines densities are proportional to the fibre densities as estimated by spherical deconvolution throughout the white matter, the number of streamlines connecting two regions becomes a proportional estimate of the cross-sectional area of the fibres connecting those two regions. We therefore hope that this method will attract usage in a range of streamlines tractography applications.

The actual usage of SIFT can be found in the help page of the `ticksift` command. In this page I'll outline some issues that are worth thinking about if you are looking to apply this method.

5.22.1 DWI bias field correction

DWI volumes often have a non-negligible *B1* bias field, mostly due to high-density receiver coils. If left uncorrected, SIFT will incorrectly interpret this as a spatially-varying fibre density. Therefore bias field correction is highly recommended. We generally estimate the bias field based on the mean $b=0$ image, and apply the estimated field to all DWI volumes. This can currently be achieved using the `dwibiascorrect` script, which can employ either the FAST tool in FSL or the N4 algorithm in ANTS to perform the field estimate.

5.22.2 Number of streamlines pre / post SIFT

In diffusion MRI streamlines tractography, we generate discrete samples from a continuous fibre orientation field. The more streamlines we generate, the better our reconstruction of that field. Furthermore, the greater number of streamlines we generate, the less influence the discrete quantification of connectivity has on the connectome (e.g. would rather be comparing 1,000 v.s. 2,000 streamlines to 1 v.s. 2; it's less likely to be an artefact of random / discrete sampling). So the more streamlines the better, at the cost of execution speed & hard drive consumption.

However we also have the added confound of SIFT. The larger the number of streamlines that can be fed to SIFT the better, as it can make better choices regarding which streamlines to keep/remove; but it also introduces a memory constraint. SIFT can deal with approximately 4-8 million streamlines per GB of RAM (depending on the seeding

mechanism used and the spatial resolution of your diffusion images), so ideally you'll want access to dedicated high-performance computing hardware. On top of this, there's the issue of how many streamlines to have remaining in the reconstruction after SIFT; the more streamlines that SIFT removes, the better the streamlines reconstruction will fit the image data, but the more likely you are to run into quantisation issues with the resulting tractogram.

So when you design your image processing pipeline, you need to consider the compromise between these factors:

- Initially generating a larger number of streamlines is beneficial for both the quality and the density of the filtered reconstruction, at the expense of longer computation time (both in generating the streamlines, and running SIFT), and a higher RAM requirement for running SIFT.
- Filtering a greater number of streamlines will always produce a superior fit to the image data, at the expense of having a lower-density reconstruction to work with afterwards, and a slightly longer computation time.

Unfortunately there's no single answer of how many streamlines are required, as it will depend on the diffusion model, tractography algorithm, and spatial extent of your target regions / connectome parcellation granularity. There are a couple of papers / abstracts on the topic if you look hard enough, but nothing definitive, and nothing involving SIFT. I would recommend testing using your own data to find numbers that are both adequate in terms of test-retest variability, and computationally reasonable.

Personally I have been using a FreeSurfer parcellation (84 nodes), generating 100 million streamlines and filtering to 10 million using SIFT (I'm a physicist; I like orders of magnitude). In retrospect, I would say that when using white matter seeding, filtering by a factor of 10 is inadequate (i.e. the fit of the reconstruction to the data is not good enough); and with grey matter - white matter interface seeding, a final number of 10 million is inadequate (the streamlines are mostly very short, so the appearance of the reconstruction is quite sparse). Another alternative is 'dynamic seeding', which uses the SIFT model during tractogram generation to only seed streamlines in pathways that are poorly reconstructed (see the `-seed_dynamic` option in `tckgen`); this provides a better initial estimate, so the percentage of streamlines that need to be removed in order to achieve a good fit is reduced. I will leave it to the end user to choose numbers that they deem appropriate (unless we do a paper on the topic, in which case you will use our published values without question).

5.22.3 Normalising connection density between subjects

An ongoing issue with our Apparent Fibre Density (AFD) work is how to guarantee that a smaller FOD in a subject actually corresponds to a reduced density of fibres. Structural connectome studies have a similar issue with regards to streamline counts; Even if SIFT is applied, this only guarantees correct proportionality between different connection pathways within a subject, not necessarily between subjects. The simplest and most common solution is simply to use an identical number of streamlines for every subject in connectome construction; however this isn't perfect:

- The distribution of streamlines lengths may vary between subjects, such that the reconstructed streamlines 'density' differs.
- A subject may have decreased fibre density throughout the brain, but be morphologically normal; if the same number of streamlines are generated, this difference won't be reflected in the tractogram post-SIFT.
- If the white matter volume varies between subjects, but the actual number of fibres within a given volume is consistent, then the subject with a larger brain may have an elevated total number of fibre connections; this would also be missed if the number of streamlines were fixed between subjects.

It's also possible to scale by the total white matter volume of each subject; this would however fail to take into account any differences in the density of fibres within a fixed volume between subjects.

An alternative approach is to try to achieve normalisation of FOD amplitudes across subjects, as is done using AFD. This requires a couple of extra processing steps, namely inter-subject intensity normalisation and use of a group average response function, which are also far from error-free. But if this can be achieved, it means that a fixed density of streamlines should be used to reconstruct a given FOD amplitude between subjects, and then the cross-sectional area of fibres represented by each streamline is also identical between subjects; this can be achieved by terminating SIFT at a given value of the proportionality coefficient using the `-term_mu` option. One potential disadvantage of

this approach (in addition to the issues associated with intensity normalisation) is that using a group average response function instead of the individual subject response may result in spurious peaks or incorrect relative volume fractions in the FODs, which could influence the tracking results.

Ideally, a diffusion model would provide the absolute partial volume of each fibre population, rather than a proportional quantity: this could then be used directly in SIFT. However the diffusion models that do provide such information tend to get the crossing fibre geometry wrong in the first place...

If anyone has any ideas on how to solve this pickle, let us know.

5.22.4 No DWI distortion correction available

SIFT should ideally be used in conjunction with ACT; by passing the ACT 5TT image to `tcksift` using the `-act` option, the command will automatically derive a processing mask that will limit the contribution of non-pure-white-matter voxels toward the model. Without this information, non-pure-white-matter voxels adversely affect both streamlines tractography, and the construction of the SIFT model.

If you are looking to apply SIFT without correction of DWI geometric distortions (and therefore without reliable high-resolution co-registered anatomical image data), these are some points that you may wish to consider:

- The spatial extent of the DWI mask may have a large influence on your streamlines tractography results. Therefore greater care should perhaps be taken to validate this mask, including manual editing if necessary.
- It is possible to manually provide a processing mask to `tcksift` using the `-proc_mask` option. If users are capable of heuristically generating an approximate white matter partial volume image from the DWI data alone, this may be appropriate information to provide to the SIFT model.

5.22.5 Use of SIFT for quantifying pathways of interest

In some circumstances, researchers may be interested in the connection density of one or two specific pathways of interest, rather than that of the whole brain. SIFT is still applicable in this scenario; however the SIFT algorithm itself is only applicable to whole-brain fibre-tracking data. Therefore, the workflow in this scenario should be: * Generate a whole-brain tractogram; * Apply SIFT; * Extract the pathway(s) of interest using `tckedit`. * Get the streamline count using `tckinfo`.

The SIFT algorithm is *not directly applicable to targeted tracking data*. The underlying biophysical model in SIFT assumes that the estimated density of each fibre population in every voxel of the image should be proportionally reconstructed by streamlines; if only a subset of pathways in the brain are permitted to be reconstructed by the tractography algorithm, this will clearly not be the case, so application of SIFT in this instance will provide erroneous results.

5.23 Structural connectome construction

Included in this new version of MRtrix are some useful tools for generating structural connectomes based on streamlines tractography. Here I will describe the steps taken to produce a connectome, and some issues that should be taken into consideration. Note that I will **not** be going into appropriate parcellations or network measures or anything like that; once you've generated your connectomes, you're on your own.

5.23.1 Preparing a parcellation image for connectome generation

Parcellations are typically provided as an integer image, where each integer corresponds to a particular node, and voxels where there is no parcellation node have a value of 0. However, for all of the parcellation schemes I've looked at thus far, the values used for the nodes do not increase monotonically from 1, but rather have some non-linear distribution; a text file (or 'lookup table') is then provided that links node indices to structure names. This is however

undesirable for connectome construction; it would be preferable for the node indices to increase monotonically from 1, so that each integer value corresponds to a row/column position in the connectome matrix.

This functionality is provided in the command `labelconvert`. It takes as its input a parcellation image that has been provided by some other software package, and converts the label indices; this is done so that the code that actually generates the connectome can be ‘dumb and blind’, i.e. the integer values at the streamline endpoints correspond to the row & column of the connectome matrix that should be incremented. In addition, this processing chain design provides flexibility in terms of both the source of the parcellation data, and the way in which the user wishes to customise the layout of their connectome.

Please consult the tutorial [labelconvert: Explanation & demonstration](#) for a guide on how to use the `labelconvert` command.

5.23.2 Generating the connectome

The command `tck2connectome` is responsible for converting the tractogram into a connectome matrix, based on the provided parcellation image. By default, the streamline count is used as the connectivity metric; run `tck2connectome -help` to see alternative heuristics / measures.

A factor in structural connectome production commonly overlooked or not reported in the literature is the mechanism used to assign streamlines to grey matter parcels. If done incorrectly this can have a large influence on the resulting connectomes. This is one aspect where *Anatomically-Constrained Tractography (ACT)* really shines; because streamlines can only terminate precisely at the grey matter - white matter interface, within sub-cortical grey matter, or at the inferior edge of the image, this assignment becomes relatively trivial. The default assignment mechanism is a radial search outwards from the streamline termination point, out to a maximum radius of 4mm; and the streamline endpoint is only assigned to the first non-zero node index. If you do not have the image data necessary to use the ACT framework, see the ‘No DWI distortion correction available’ section below.

5.23.3 SIFT and the structural connectome

If you are generating structural connectomes, you should be using *Spherical-deconvolution Informed Filtering of Tractograms (SIFT)*.

5.23.4 Extracting pathways of interest from a connectome

The command `connectome2tck` can be used to extract specific connections of interest from a connectome for further interrogation or visualisation. Note that since the resulting connectome matrix does not encode precisely which parcellation node pair each streamline was assigned to, the streamlines are re-assigned to parcellation nodes as part of this command. Run `connectome2tck -help` to see the various ways in which streamlines may be selected from the connectome.

Also: Beware of running this command on systems with distributed network file storage. This particular command uses an un-buffered file output when writing the streamlines files, which re-opens the output file and writes data for individual streamlines at a time (necessary as many files may be generated at once); such systems tend to be optimised for large-throughput writes, so this command may cause performance issues.

5.23.5 No DWI distortion correction available

If you can’t perform DWI susceptibility distortion correction, it severely limits how accurately you can estimate the structural connectome. If this is the case for you, below is a few points that are worth considering.

Non-linear registration

Rather than actually correcting the DWI geometric distortions, some people try to do a non-linear registration between DWI and T1 images. In general I'm against this: the registration is fairly ill-posed due to the differing contrasts, and an off-the-shelf non-linear registration will have too many degrees of freedom. Pursue at your own risk.

Grey matter parcellation

With good spatial alignment, parcellations that highlight only the cortical ribbon (e.g. FreeSurfer) are highly accurate and effective, and the assignment of streamlines to those parcellations will also be robust if ACT is used. But without these, residual registration errors may have a large influence, and assigning streamlines to parcellations only as thick as the cortex may also be erroneous (streamlines may terminate prior to the parcel, or travel through and extend well beyond it). A parcellation with large-volume nodes that is based on atlas registration (e.g. AAL) is likely more appropriate in this case.

Assignment of streamlines to parcellation nodes

Without ACT, streamlines will terminate pretty much anywhere within the DWI brain mask. Not only this, but they may traverse multiple parcellation nodes, turn around within a node and traverse elsewhere, terminate just prior to entering a node, all sorts of weirdness. I have provided a few assignment mechanisms that you can experiment with - run `tck2connectome -help` to see the list and parameters for each. Alternatively if anyone has a better idea for how this could potentially be done, I'd love to hear it.

5.24 Using the connectome visualisation tool

The connectome tool bar in *MRtrix3* has been designed from scratch, with the intention of providing a simple, data-driven mechanism for visually assessing individual connectomes as well as the results of network-based group statistics. The interface may therefore vary considerably from other connectome visualisation packages, and may be intimidating for new users who simply want to 'see the connectome'. I hope I can convince you in this tutorial that the design of this tool allows you, the user, to dictate exactly *how* you want to visualise the connectome, rather than being forced to conform to a particular prior expectation of how such things should be visualised.

5.24.1 Initialising the tool

My suspicion is that new users will load the tool, and immediately think: 'Where do I load my connectome?'. Well, let's take a step backwards. If you were to give the software a connectome matrix, with no other data, there would be no way to visualise that connectome in the space of an MR image: the software has no information about the spatial locations of the nodes upon which that connectome is based. So the first step is actually to load an image to provide the tool with this information, using the "Node image" button at the top of the toolbar. The desired image is the output of the `labelconvert` command, as detailed in the [Structural connectome construction](#) guide: the tool uses this image to localise each parcel in 3D space in preparation for visualisation. Alternatively, you can load the relevant parcellation image from the command-line when you first run `mrview`, using the `-connectome.init` option.

Attention: If you still do not see anything in the `mrview` main window, this is likely because you have not yet opened a primary image in `mrview`. This is currently necessary for `mrview` to correctly set up the camera positioning. The easiest solution is to open your parcellation image not only to initialise the connectome tool, but also as a standard image in `mrview`; then simply *hide* the main image using the 'View' menu.

With the basis parcellation image loaded, the tool will display the location of each node; note however that all of the nodes are exactly the same colour, and exactly the same size, and there are no connections shown between them - it's an entirely dis-connected network. This makes sense - we haven't actually provided the tool with any information regarding which connections are present and which are absent. We can also do the opposite: change the "Edge visualisation" - "Visibility" from 'None' to 'All', and now the software shows every edge in the connectome non-discriminantly.

Therefore, we need some mechanism of informing the software of which edges should be drawn, and which should not. Most logically, this could be achieved by loading a structural connectome, and perhaps applying some threshold. So now, for the "Edge visualisation" - "Visibility" option, select "Matrix file", and load your connectome. The software now uses the data from this external file to threshold which edges are drawn and which are not, and also allows you to vary that threshold interactively. (You can also load a connectome matrix from the command line using the `-connectome.load` option.)

The connectome still however has a binary appearance; every edge in the connectome is either present or absent, and they all have the same size and the same colour. We know that our connectome contains weights distributed over a wide scale, and would like to be able to see this as part of our visualisation; for instance, we may decide that more dense connections should have a 'hot' colour appearance, whereas less dense connections should be darker. We can achieve this by changing the "Edge visualisation" - "Colour" from 'Fixed' to 'Matrix file', and selecting an appropriate matrix file (perhaps the same file as was used for the visibility threshold, perhaps not).

For most users, connectome data will be loaded using the 'open' button in the 'connectome matrices' section, or at the command-line when `mrview` is first run using the `-connectome.load` option.

5.24.2 Basis of connectome visualisation customisation

With the above steps completed, you should obtain a fairly rudimentary visualisation of the connectome you have loaded. The plethora of buttons and gadgets in the connectome tool user interface is however a clue regarding the scope of customisation available for precisely how the connectome data will be displayed.

As an example, consider the 'Edge visualisation - Colour' entry. These options control how the colour of each individual edge in the connectome will be determined, based on the data the tool is provided with. Clicking on the main combo box shows that there are a few options available:

- *Fixed*: Use the same fixed colour to display all visible edges.
- *By direction*: The XYZ spatial offset between the two nodes connected by an edge is used to derive an RGB colour (much like the default streamlines colouring).
- *Connectome*: The colour of each edge will depend on the value for that edge in the connectome you have loaded, based on some form of value -> colour mapping (a 'colour map').
- *Matrix file*: Operates similarly to the *connectome* option; except that the value for each edge is drawn from a matrix file that is *not* the connectome matrix you have loaded (though it must be based on the same parcellation to have any meaning). So for instance: You could load a structural connectome file as your connectome matrix and show only those edges where the connection density is above a certain threshold, but then set the *colour* of each edge based on a *different matrix file* that contains functional connectivity values.

If the *Connectome* or *Matrix file* options are used, it is also possible to alter the colour map used, and modify the values at which the edges will reach the colours at either extreme of the colour map.

Hopefully, this simple demonstration will be enough to highlight the design principle of this tool, and therefore the frame of mind necessary to use it effectively:

What **data do I want to determine a specific **visual property** of my connectome?**

There is tremendous power in separating these two entities. For instance, consider a use case where I have performed network-based group statistics, and wish to visualise my result. I may choose to threshold the connectome edges based on statistical significance, but set the width of the connections based on the mean connection strength to get an idea

of the density of connections in the detected network, but set the colour of each edge based on the effect size to see which components of the network are most affected. I can even automatically hide any nodes that are not involved in the detected network by selecting “Node visualisation” - “Visibility” - ‘Degree ≥ 1 ’.

5.24.3 Importing detailed node information

When the parcellation image is first loaded, the software has no information regarding the designations of the underlying nodes, so it simply labels them as “Node 1”, “Node 2” etc.. To show the anatomical name of each node in the list, you must load the connectome lookup table that was used as the target output in the `labelconvert` step during [structural connectome construction]. This file provides a list of node indices and their corresponding names, so is perfect for subsequent assessment of the resulting connectomes, whether using this tool or in other contexts (e.g. Matlab). Such a lookup table may also include a pre-defined colour for each node, which can then be used during visualisation by selecting “Node Visualisation -> Colour -> LUT”.

5.24.4 Advanced visualisation

There are a couple of neat tricks that can be used to produce impressive-looking visualisations, but need some pre-processing or careful consideration in order to achieve them.

Visualising edges as streamlines / streamtubes

Rather than drawing a straight line between connected nodes to represent an edge, it is possible with tractography-based connectome construction to instead represent each connection based on the structural trajectory by which those nodes are inter-connected. This can be achieved as follows:

- When generating the connectome using `tck2connectome`, use the `-out_assignments` option. This will produce a text file where each line contains the indices of the two nodes to which that particular streamline was assigned.
- Use the `connectome2tck` command to produce a single track file, where every streamline represents the mean, or *exemplar*, trajectory between two nodes. This is achieved using two command-line options: `-exemplars` to instruct the command to generate the exemplar trajectory for each edge, rather than keeping all streamlines (you will need to provide your parcellation image); and `-files single` to instruct the command to place all computed exemplars into a single output file.
- In the `mrview` connectome toolbar, select “Edge visualisation” - “Geometry” - ‘Streamlines / Streamtubes’, and select the exemplar track file just generated.

Visualising nodes as triangulated meshes

Although the node parcellations are represented as volumetric segmentations, and we do not yet have support for importing mesh-based parcellations, it is still possible to visualise the connectome nodes using a mesh-based representation. This is done by explicitly converting the volume of each parcel to a triangulated mesh. The process is as follows:

- Compute a triangular mesh for each node, and store the results in a single file. The command is called `label2mesh`. Note that the output file *must* be in the `.obj` file format: this is the only format currently supported that is capable of storing multiple mesh objects in a single file.
- (Optional) Smooth the meshes to make them more aesthetically pleasing (the results of the conversion process used in `label2mesh` appear very ‘blocky’). Apply the `meshfilter` command, using the `smooth` operator. Again, the output must be in the `.obj` format.

- In the `mrview` connectome toolbar, select “Node visualisation” - “Geometry” - ‘Mesh’, and select the mesh file just generated.

Using node selection to highlight features of interest

The table in the connectome toolbar that lists the node names and colours can also be used to select and highlight particular nodes. In most cases, this will simply be an additional ‘toy’ for navigating the data; however it’s also possible that this capability will prove to be a powerful tool for demonstrating network features.

In any connectome visualisation software, when the user selects one or more particular nodes of interest, some modification must be applied to the visual features of the nodes in order to ‘highlight’ the nodes of interest. In many cases, this may be hard-wired to behave in a particular way. In the case of `mrview` in *MRtrix3*, this highlighting mechanism is entirely flexible: the user can control the visual modifications applied to both those network elements selected and those not selected. For instance, you may choose for nodes to become completely opaque when you select them, while other un-selected nodes remain transparent; or they may grow in size with respect to the rest of the connectome; or they may change in colour to highlight them; or those nodes not selected may disappear entirely. This flexibility is accessed via the “Selection visualisation settings” button, which will open a dialog window providing access to these settings.

As manual selection applies to nodes only, the behaviour for edges is as follows:

- When no nodes are selected, all edges are drawn according to their standard settings.
- If a single node is selected, all edges emanating from that node are considered to be ‘selected’, and the relevant visual modifiers will be applied.
- If two or more nodes are selected, only connections exclusively connecting between the nodes of interest are considered to be ‘selected’.

Node visualisation using matrices

When using external data files to control the visual properties of the connectome, most commonly *vector files* will be used to determine visual properties of nodes, and *matrix files* will be used to determine visual properties of edges. These provide precisely one scalar value per connectome element, and therefore provide a static visual configuration.

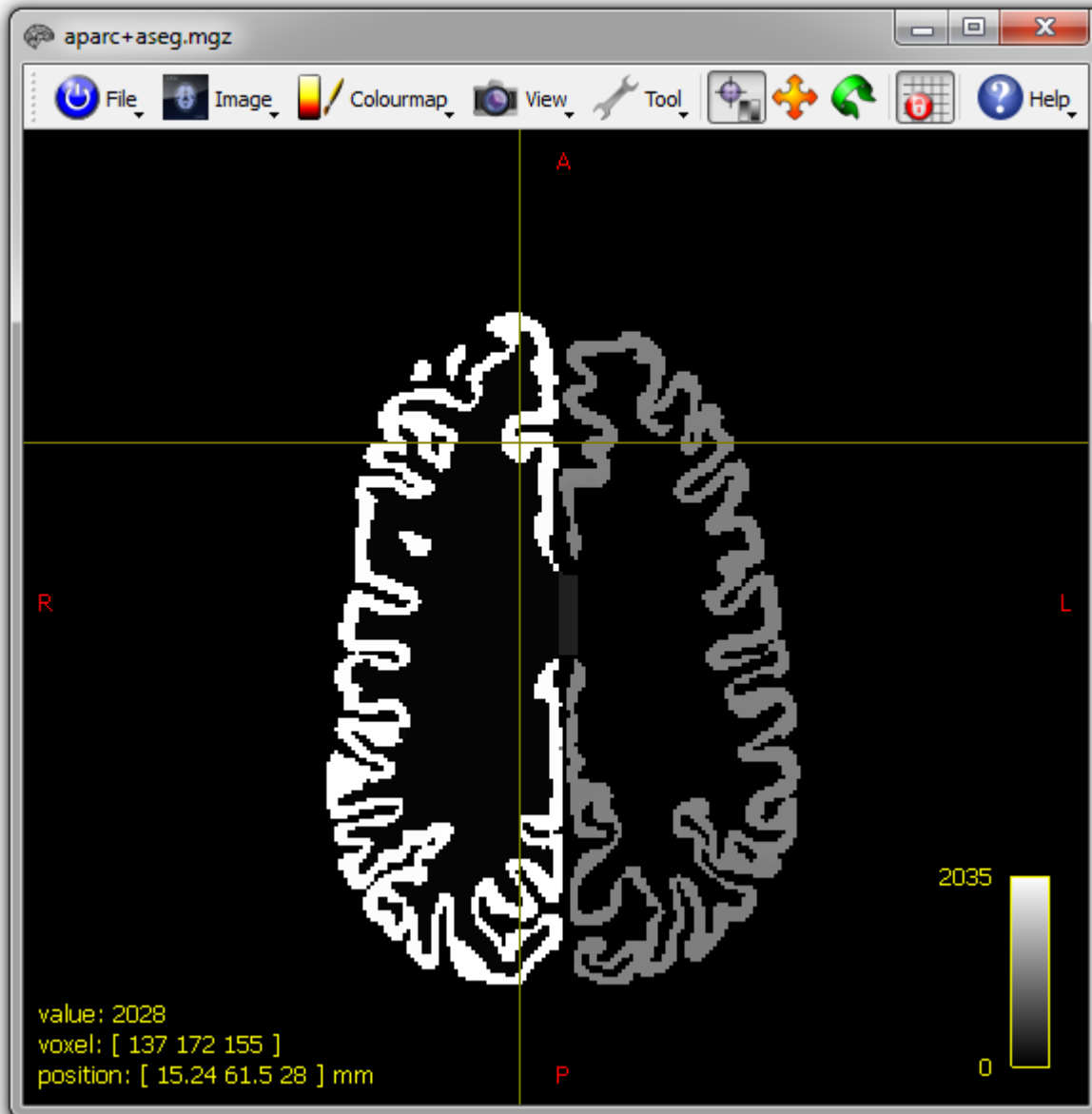
It is however also possible to set any visual property of the connectome nodes based on a *matrix file*. In this scenario, the values to be drawn from the matrix - and hence their influence on the relevant visual property of the nodes - depends on the *current node selection*. That is: once you select a node of interest, the software extracts the relevant row from the matrix, and uses only that row to influence the node visual property to which it has been assigned. In the case where multiple nodes of interest are selected, an additional drop-down menu is provided, that allows you to prescribe how those multiple rows of matrix data are combined in order to produce a single scalar value per node, which can then be used to influence its relevant visual property.

5.25 `labelconvert`: Explanation & demonstration

The `labelconvert` (previously `labelconfig`) step in *Structural connectome construction* has proven to be a hurdle for many. It may be a ‘unique’ step in so far as that other software packages probably deal with this step implicitly, but in MRtrix we prefer things to be explicit and modular. So here I’ll go through an example to demonstrate exactly what this command does.

5.25.1 Worked example

For this example, let's imagine that we're going to generate a structural connectome for Bert, the quintessential FreeSurfer subject. Also, we're going to generate the connectome based on the Desikan-Killiany atlas. The default FreeSurfer pipeline provides the volumetric image `aparc+aseg.mgz`; this is the file that will be used to define the nodes of our connectome.



Looking at the raw image itself, each node possesses a particular intensity, corresponding to a particular integer value. If we focus on the superior frontal gyrus in the right hemisphere, we can see that the image intensity is 2028 for this structure.

This immediately presents a problem for constructing a connectome: if any streamline encountering this region were written to row/column 2028, our connectome would be enormous, and consist mostly of zeroes (as most indices between 1 and 2028 do not correspond to any structure). Therefore, what we'd prefer is to map the unique integer

index of this structure to a particular row/column index of the connectome; this should be done in such a way that all structures of interest have a unique integer value between 1 and N , where N is the number of nodes in the connectome.

Now looking at the file `FreeSurferColorLUT.txt` provided with FreeSurfer, we see the following:

```
...
2026   ctx-rh-rostralanteriorcingulate      80  20  140  0
2027   ctx-rh-rostralmiddlefrontal          75  50  125  0
2028   ctx-rh-superiorfrontal                20  220 160  0
2029   ctx-rh-superiorparietal               20  180 140  0
2030   ctx-rh-superiortemporal              140  220 220  0
...
```

This gives us a *meaningful name* for this structure based on the integer index. It also gives us some colour information, but let's not worry about that for now.

Our goal then is to determine a *new integer index* for this structure, that will determine the row/column of our connectome matrix that this structure corresponds to. This is dealt with by mapping the structure indices of this lookup table to a *new* lookup table. For this example, let's imagine that we're using the default MRtrix lookup table for the FreeSurfer Desikan-Killiany atlas segmentation: this is provided at `shared/mrtrix3/labelconvert/fs_default.txt`. Examining this file in detail, we see the following:

```
...
74     R.RACG   ctx-rh-rostralanteriorcingulate 80  20  140 255
75     R.RMFG   ctx-rh-rostralmiddlefrontal    75  50  125 255
76     R.SFG    ctx-rh-superiorfrontal          20  220 160 255
77     R.SPG    ctx-rh-superiorparietal         20  180 140 255
78     R.STG    ctx-rh-superiortemporal         140  220 220 255
...
```

(This file is in a slightly different format to `FreeSurferColorLUT.txt`; don't worry about this for the time being)

This file contains the *same structure name* as the FreeSurfer look-up table, but it is assigned a *different integer index* (76)! What's going on?

The following is what the `labelconvert` command is actually going to do under the bonnet, using these two lookup table files:

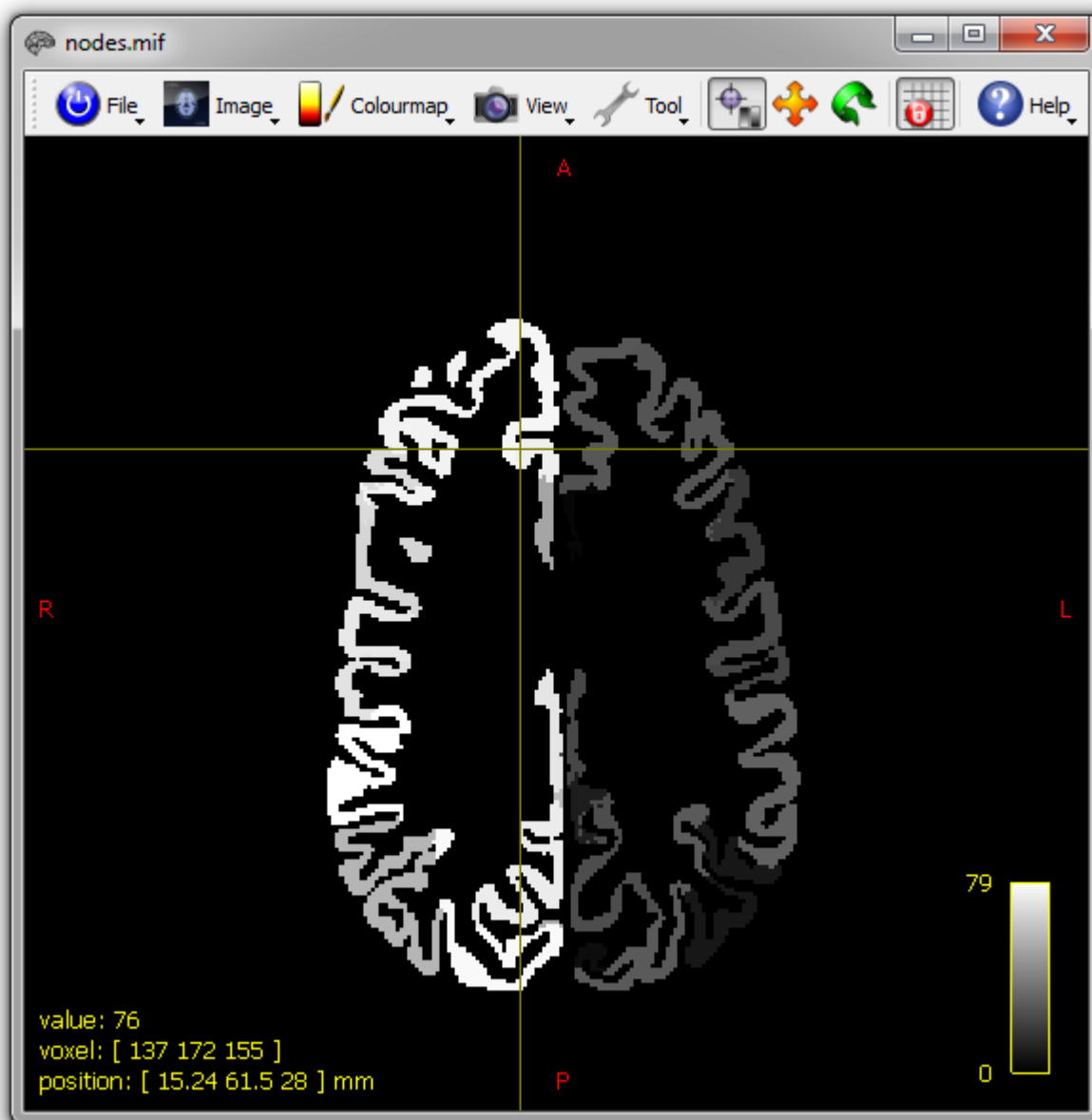
1. Read the integer value at each voxel of the input image
2. Convert the integer value into a string, based on the *input lookup table file* (`FreeSurferColorLUT.txt`)
3. Find this string in the *output lookup table file* (`fs_default.txt`)
4. Write the integer index stored in the *output lookup table file* for this structure to the voxel in the output image

This is what the actual command call looks like:

```
labelconvert $FREESURFER_HOME/subjects/bert/mri/aparc+aseg.mgz $FREESURFER_HOME/
↪FreeSurferColorLUT.txt ~/mrtrix3/share/mrtrix3/labelconfig/fs_default.txt bert_
↪parcels.mif
```

And this is what the resulting image looks like:

The integer labels of the underlying grey matter parcels have been *converted* from the input lookup table to the output lookup table (hence the name `labelconvert`). They now increase monotonically from 1 to the maximum index, with no 'gaps' (i.e. unused integer values) in between. Therefore, when you construct your connectome using `tk2connectome`, the connectome matrix will only be as big as it needs to be to store all of the node-node connectivity information.



5.25.2 Design rationale

Making this step of re-indexing parcels explicit in connectome construction has a few distinct advantages:

- You can use parcellations from any software / atlas: just provide the structure index / name lookup table that comes with whatever software / atlas provides the parcellation, and define an appropriate target lookup table that defines which index you want each structure to map to.
- `tck2connectome` can be ‘dumb and blind’: it reads the integer indices at either end of the streamline, and that’s the row/column of the connectome matrix that needs to be incremented.
- You can have your grey matter parcels appear in any order in your matrices: just define a new lookup table file. Doing this prior to connectome construction is less likely to lead to heartache than re-ordering the rows and columns in e.g. Matlab, where you may lose track of which matrices have been re-ordered and which have not.
- You can remove structures from the connectome, or merge multiple structures into a single parcel, just by omitting or duplicating indices appropriately in the target lookup table file.
- Looking at your matrices and need to find out what structure corresponds to a particular row/column? Just look at the config file!

Obviously if your parcellation image already has node indices that increase monotonically from 1, and you’re happy enough with the numerical order of the nodes, you don’t actually need to use the `labelconvert` step at all.

5.25.3 Custom design connectomes

Some notes for anybody that wishes to define their own configuration files (either for re-ordering nodes, changing selection of nodes, or using parcellations from alternative sources):

- If you wish to omit nodes from your connectome (e.g. the cerebellar hemispheres), you may be better off making these nodes the largest indices in your connectome, but then cropping them from the connectome matrices retrospectively, rather than omitting them from the parcellation image entirely: If you were to do the latter, streamlines that would otherwise be assigned to your unwanted nodes may instead be erroneously assigned to the nearest node that is part of your connectome (exactly what happens here will depend on the streamline-node assignment mechanism used).
- The command `labelconvert` is capable of reading in look-up tables in a number of formats. If you wish to define your own lookup table, you will need to conform to one of these formats in order for MRtrix commands to be able to import it. If you are using an atlas where the look-up table does not conform to any of these formats (and hence MRtrix refuses to import it), you can either manually manipulate it into a recognized format, or if it is likely that multiple users will be using that parcellation scheme, we may choose to add a parser to the MRtrix code: contact the developers directly if this is the case.

5.26 Global tractography

5.26.1 Introduction

Global tractography is the process of finding the full track configuration that best explains the measured DWI data. As opposed to streamline tracking, global tractography is less sensitive to noise, and the density of the resulting tractogram is directly related to the data at hand.

As of version 3.0, MRtrix supports global tractography using a multi-tissue spherical convolution model, as introduced in [Christiaens2015]. This method extends the method of [Reisert2011] to multi-shell response functions, estimated from the data, and adopts the multi-tissue model presented in [Jeurissen2014] to account for partial voluming.

5.26.2 User guide

Prerequisites

This global tractography implementation relies on *multi-shell* high angular resolution diffusion imaging (HARDI) data, containing at least 3 unique *b*-values (i.e 2 shells along with the *b*=0 volumes).

In addition, this command expects that suitable *multi-shell multi-tissue response functions* have already been computed. A number of approaches are available for this, please refer to the [Response function estimation](#) page for details.

Invocation

For multi-shell DWI data, the most common use will be:

```
tckglobal dwi.mif wm_response.txt -riso csf_response.txt -riso gm_response.txt -mask_
↪mask.mif -niter 1e9 -fod fod.mif -fiso fiso.mif tracks.tck
```

In this example, `dwi.mif` is the input dataset, including the gradient table, and `tracks.tck` is the output tractogram. `wm_response.txt`, `gm_response.txt` and `csf_response.txt` are the corresponding tissue response functions (as estimated in a previous [Response function estimation](#) step). Optional output images `fod.mif` and `fiso.mif` contain the predicted WM fODF and isotropic tissue fractions of CSF and GM respectively, estimated as part of the global optimization and thus affected by spatial regularization.

Parameters

`-niter`: The number of iterations in the optimization. Although the default value is deliberately kept low, a full brain reconstruction will require at least 100 million iterations.

`-lmax`: Maximal order of the spherical harmonics basis.

`-length`: Length of each track segment (particle), which determines the resolution of the reconstruction.

`-weight`: Weight of each particle. Decreasing its value by a factor of two will roughly double the number of reconstructed tracks, albeit at increased computation time.

Particle potential `-ppot`: The particle potential essentially associates a *cost* to each particle, relative to its weight. As such, we are in fact trying to reconstruct the data as well as possible, with as few particles as needed. This ensures that there is sufficient *proof* for each individual particle, and hence avoids that a bit of noise in the data spurs generation of new (random) particles. Think of it as a parameter that balances sensitivity versus specificity. A higher particle potential requires more *proof* in the data and therefore leads to higher specificity; a smaller value increases sensitivity.

Connection potential `-cpot`: The connection potential is the driving force for connecting segments and hence building tracks. Higher values increase connectivity, at the cost of increased invalid connections.

Ancillary outputs

`-fod`: Outputs the predicted fibre orientation distribution function (fODF) as an image of spherical harmonics coefficients. This fODF is estimated as part of the global track optimization, and therefore incorporates the spatial regularization that it imposes. Internally, the fODF is represented as a discrete sum of apodized point spread functions (aPSF) oriented along the directions of all particles in the voxel, akin to track orientation distribution imaging (TODI, [Dhollander2014]). This internal representation is used to predict the DWI signal upon every change to the particle configuration.

`-fiso`: Outputs the estimated density of all isotropic tissue components, as multiple volumes in one 4-D image in the same order as their respective `-riso` kernels were provided.

`-eext`: Outputs the residual data energy image, including the L1-penalty imposed by the particle potential.

5.27 ISMRM tutorial - Structural connectome for Human Connectome Project (HCP)

Warning: This page describes an outdated example from 2015, currently retained only for historical reference. The steps described in this page are not up to date with current optimised practices. We do not recommend to use the steps below in connectomics processing for any project, using either HCP or other data.

This document duplicates the information provided during the *MRtrix3* demonstration at ISMRM 2015 in Toronto. We will generate a structural connectome for quintessential Human Connectome Project subject 100307. Some of these instructions will be specific to HCP data, others will be more general recommendations.

Note that this page is being retained as a reference of the steps demonstrated during the ISMRM 2015 meeting; it does *not* constitute an up-to-date ‘recommended’ processing pipeline for HCP data.

5.27.1 Necessary files

To duplicate our methods and results, you will need to download the appropriate files, accessible through the following steps:

- <https://db.humanconnectome.org/>
- WU-Minn HCP Data - 900 Subjects + 7T
- Download Image Data: Single subject
- Session Type: 3T MRI
- Processing level: Preprocessed
- Package Type: MSM-Sulc + MSM-All
- add Structural Preprocessed and Diffusion Preprocessed to queue

The actual files within these compressed downloads that we will make use of are:

Diffusion preprocessed files

- bvals
- bvecs
- data.nii.gz
- nodif_brain_mask.nii.gz

Structural preprocessed files

- aparc+aseg.nii.gz
- T1w_acpc_dc_restore_brain.nii.gz

5.27.2 Structural image processing

1. Generate a tissue-segmented image appropriate for [Anatomically-Constrained Tractography](#):

```
5ttgen fsl T1w_acpc_dc_restore_brain.nii.gz 5TT.mif -premasked
```

Note that it is *not necessary* to use a tissue-segmented image that has the same resolution as the diffusion images; *MRtrix3* will happily acquire interpolated values from each of them separately as tracking is performed. This allows ACT to exploit the higher spatial resolution of the tissue-segmented anatomical image, but still use the diffusion image information at its native resolution also.

2. Collapse the multi-tissue image into a 3D greyscale image for visualisation:

```
5tt2vis 5TT.mif vis.mif; mrview vis.mif
```

If the tissue segmentation image contains clearly erroneous tissue labels, you can delineate them manually using the ROI editor tool in *mrview*, then apply your corrections to the tissue data using the *5ttedit* command.

3. Modify the integer values in the parcellated image, such that the numbers in the image no longer correspond to entries in FreeSurfer's colour lookup table, but rows and columns of the connectome:

```
labelconvert aparc+aseg.nii.gz FreeSurferColorLUT.txt fs_default.txt nodes.mif
```

File *FreeSurferColorLUT.txt* is provided with FreeSurfer in its root directory. The target lookup table file (*fs_default.txt* in this case) is a handy text file that provides a structure name for every row / column of the connectome matrix: it is provided as part of *MRtrix3*, and located at *shared/mrtrix3/labelconvert/fs_default.txt* within the *MRtrix3* folder.

4. Replace FreeSurfer's estimates of sub-cortical grey matter structures with estimates from FSL's FIRST tool:

```
labelsgmfix nodes.mif T1w_acpc_dc_restore_brain.nii.gz fs_default.txt  
nodes_fixSGM.mif -premasked
```

5.27.3 Diffusion image processing

1. Convert the diffusion images into a non-compressed format (not strictly necessary, but will make subsequent processing faster), embed the diffusion gradient encoding information within the image header, re-arrange the data strides to make volume data contiguous in memory for each voxel, and convert to floating-point representation (makes data access faster in subsequent commands):

```
mrconvert data.nii.gz DWI.mif -fslgrad bvecs bvals -datatype float32 -strides  
0,0,0,1
```

2. Generate a mean $b=0$ image (useful for visualisation):

```
dwiextract DWI.mif - -bzero | mrmath - mean meanb0.mif -axis 3
```

(If you are not familiar with the 'l' piping symbol, read more about it [here](#))

3. [Estimate the response function](#); note that here we are estimating *multi-shell, multi-tissue* response functions:

```
dwi2response msmt_5tt DWI.mif 5TT.mif RF_WM.txt RF_GM.txt RF_CSF.txt -voxels  
RF_voxels.mif
```

mrview meanb0.mif -overlay.load RF_voxels.mif -overlay.opacity 0.5 (check appropriateness of response function voxel selections)

4. Perform Multi-Shell, Multi-Tissue Constrained Spherical Deconvolution:

```
dwi2fod msmt_csd DWI.mif RF_WM.txt WM_FODs.mif RF_GM.txt GM.mif RF_CSF.txt  
CSF.mif -mask nodif_brain_mask.nii.gz
```

```
mrconvert WM_FODs.mif - -coord 3 0 | mrcat CSF.mif GM.mif - tissueRGB.mif
-axis 3
```

This generates a 4D image with 3 volumes, corresponding to the tissue densities of CSF, GM and WM, which will then be displayed in *mrview* as an RGB image with CSF as red, GM as green and WM as blue (as was presented in the MSMT CSD manuscript).

```
mrview tissueRGB.mif -odf.load_sh WM_FODs.mif (visually make sure that both the tissue segmen-
tations and the white matter FODs are sensible)
```

5.27.4 Connectome generation

1. Generate the initial tractogram:

```
tckgen WM_FODs.mif 100M.tck -act 5TT.mif -backtrack -crop_at_gmwmi
- -seed_dynamic WM_FODs.mif -maxlength 250 -select 100M -cutoff 0.06
```

Explicitly setting the maximum length is highly recommended for HCP data, as the default heuristic - 100 times the voxel size - would result in a maximum length of 125mm, which would preclude the reconstruction of some longer pathways.

We also suggest a reduced FOD amplitude cutoff threshold for tracking when using the MSMT CSD algorithm in conjunction with ACT; this allows streamlines to reach the GM-WM interface more reliably, and does not result in significant false positives since the MSMT algorithm does not produce many erroneous small FOD lobes.

2. Apply the [Spherical-deconvolution Informed Filtering of Tractograms \(SIFT\)](#) algorithm

This method reduces the overall streamline count, but provides more biologically meaningful estimates of structural connection density:

```
tcksift 100M.tck WM_FODs.mif 10M_SIFT.tck -act 5TT.mif -term_number 10M
```

If your system does not have adequate RAM to perform this process, the first recommendation is to reduce the spatial resolution of the FOD image and provide this alternative FOD image to SIFT (this should have little influence on the outcome of the algorithm, but will greatly reduce memory consumption):

```
mrgrid WM_FODs.mif regrid FOD_downsampled.mif -scale 0.5 -interp sinc
```

If this still does not adequately reduce RAM usage, you will need to reduce the number of input streamlines to a level where your processing hardware can successfully execute the *tcksift* command, e.g.:

```
tckedit 100M.tck 50M.tck -number 50M
```

Alternatively, if you're feeling brave, you can give [SIFT2](#) a try...

3. Map streamlines to the parcellated image to produce a connectome:

```
tck2connectome 10M_SIFT.tck nodes_fixSGM.mif connectome.csv
```

```
mrview nodes_fixSGM.mif -connectome.init nodes_fixSGM.mif -connectome.load
connectome.csv
```

5.28 Diffusion gradient scheme handling

An essential piece of information for DWI processing is the diffusion-weighted (DW) gradient scheme, also known as the “*DW gradient table*”, the “*DW encoding*”, the “*b-vectors*”, the “*bvecs*”, and other variations on the theme. This table provides information about the diffusion sensitisation gradients applied during acquisition of each imaging volume in a DWI dataset, usually in the form of the *b*-value and the (unit) vector for the DW gradient direction. In this

page we will describe the details of how this information is typically stored / represented, and how *MRtrix3* handles / manipulates this data.

5.28.1 Gradient table storage

MRtrix3 allows the DW gradient table to be read directly from, or written to, the image headers for specific image formats; notably *DICOM* (*folder or .dcm*) (read-only) and the *MRtrix image formats* (*.mih / .mif*) (read/write). *MRtrix3* applications will automatically make use of this information when it is available for the input dataset through storage of the table within the *image header*, without requiring explicit intervention from the user. In addition, *MRtrix3* commands can also import or export this information from/to two different external file formats: typically referred to as the *MRtrix format* and the *FSL format*. These differ in a number of respects, as outlined below.

MRtrix format

This format consists of a single ASCII text file, with no restrictions on the filename. It consists of one row per entry (i.e. per DWI volume), with each row consisting of 4 space-separated floating-point values; these correspond to $[x \ y \ z \ b]$, where $[x \ y \ z]$ are the components of the gradient vector, and b is the b -value in units of s/mm^2 . A typical *MRtrix* format DW gradient table file might look like this:

Listing 1: **grad.b:**

0	0	0	0
0	0	0	0
-0.0509541	0.0617551	-0.99679	3000
0.011907	0.955047	0.296216	3000
-0.525115	0.839985	0.136671	3000
-0.785445	-0.6111	-0.0981447	3000
0.060862	-0.456701	0.887536	3000
0.398325	0.667699	0.6289	3000
-0.680604	0.689645	-0.247324	3000
0.237399	0.969995	0.0524565	3000
0.697302	0.541873	-0.469195	3000
-0.868811	0.407442	0.28135	3000
...			

It is important to note that in this format, the direction vectors are assumed to be provided with respect to *real* or *scanner* coordinates. This is the same convention as is used in the DICOM format. Also note that the file does not need to have the file type extension *.b* (or any other particular suffix); this is simply a historical convention.

Image header

When using the *MRtrix image formats* (*.mih / .mif*), *MRtrix3* has the capability of *embedding* the diffusion gradient table *within the header of the image file*. This provides significant advantages when performing image processing:

- The table accompanies the image data at all times, which means that the user is not responsible for tracking which diffusion gradient table corresponds to which image file, or whether or not a particular gradient table file reflects some manipulation that has been applied to an image.
- In *MRtrix3* commands that require a diffusion gradient table, and/or make modifications to the image data that require corresponding modifications to the diffusion gradient table, these data will be utilised (and/or modified) *automatically*, without requiring explicit intervention from the user.

For these reasons, the general recommendation of the *MRtrix3* team is to make use of the *MRtrix image formats* (*.mih / .mif*) whenever possible.

This embedding is achieved by writing an entry into the Image *Header key-value pairs*, using the key `dw_scheme`. The value of this entry is the complete diffusion gradient table, stored in the *MRtrix format*. However, this entry should generally *not be accessed or manipulated directly* by users; instead, users should rely on the internal handling of these data as performed by *MRtrix3* commands, or where relevant, use the command-line options provided as part of specific *MRtrix3 commands*, as detailed later.

FSL format

This format consists of a pair of ASCII text files, typically named `bvecs` & `bvals` (or variations thereof). The `bvals` file consists of a single row of space-separated floating-point values, all in one row, with one value per volume in the DWI dataset. The `bvecs` file consists of 3 rows of space-separated floating-point values, with the first row corresponding to the *x*-component of the DW gradient vectors, one value per volume in the dataset; the second row corresponding to the *y*-component, and the third row to the *z*-component. A typical pair of FSL format DW gradient files might look like:

Listing 2: `bvecs`:

```
0 0 -4.30812931665e-05 -0.00028279245503 -0.528846962834659 -0.781281266220383 0.
→014299684287952 0.36785999072309 -0.66507232482745 0.237350171404029 0.
→721877079467007 -0.880754419294581 0 -0.870185851757858 ...
0 0 -0.002606397951389 -0.97091525561761 -0.846605326714759 0.615840299891175 0.
→403330065122241 -0.70377676751476 -0.67378508548543 -0.971399047063277 -0.
→513131073140676 -0.423391107245363 0 -0.416501756655988 ...
0 0 -0.999996760803023 0.23942421337746 0.059831733802001 -0.101684552642539 0.
→914942902775223 0.60776414747636 -0.32201498900359 0.007004078617919 -0.
→464317089148873 0.212157919445896 0 -0.263255013300656 ...
```

Listing 3: `bvals`:

```
0 0 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 ...
```

It is important to note that in this format, the gradient vectors are provided *with respect to the image axes*, **not** in real or scanner coordinates (actually, it's a little bit more complicated than that, refer to the [FSL wiki](#) for details). This is a rich source of confusion, since seemingly innocuous changes to the image can introduce inconsistencies in the *b*-vectors. For example, simply reformatting the image from sagittal to axial will effectively rotate the *b*-vectors, since this operation changes the image axes. It is also important to remember that a particular `bvals`/`bvecs` pair is only valid for the particular image that it corresponds to.

5.28.2 Using the DW gradient table in *MRtrix3* applications

Querying the DW gradient table

As mentioned above, *MRtrix3* will use the DW gradient table from the image headers when it is available. Currently, only the *DICOM (folder or .dcm)* and *MRtrix image formats (.mih / .mif)* support this. The DW gradient table can be queried for any particular image using the `mrinfo` command in combination with the `-dwgrad` option. For example:

```
$ mrinfo DICOM/ -dwgrad
mrinfo: [done] scanning DICOM folder "DICOM/"
mrinfo: [100%] reading DICOM series "BRI 64 directions ep2d_diff_3scan_trace_p2"
      0      0      0      0
-0.999994 0.00167109 0.00300897      3000
      -0      0.999996 0.00299996      3000
0.0261389 0.65148 -0.758215      3000
```

(continues on next page)

(continued from previous page)

-0.590138	-0.767763	-0.249553	3000
0.236087	-0.527069	-0.816371	3000
0.893005	-0.261931	-0.36597	3000
-0.797405	0.126351	-0.590068	3000
-0.233751	0.930868	-0.280794	3000
-0.936406	0.141569	-0.321095	3000
-0.505355	-0.845584	0.17206	3000
-0.346203	-0.848909	0.39937	3000
-0.457204	-0.633042	0.624678	3000
0.48716	-0.391994	-0.780395	3000
0.617871	0.674589	-0.403938	3000
0.577709	-0.102522	0.809779	3000
0.825818	-0.523076	-0.210752	3000
...			

Exporting the DW gradient table

This information can also be exported from the image headers using the `-export_grad_mrtrix` option (for the *MRtrix format*) or `-export_grad_fsl` option (for the *FSL format*) in commands that support it. For example:

```
$ mrinfo dwi.mif -export_grad_mrtrix grad.b
```

results in a `grad.b` file in *MRtrix format*, while:

```
$ mrconvert DICOM/ dwi.nii.gz -export_grad_fsl bvecs bvals
mrconvert: [done] scanning DICOM folder "DICOM/"
mrconvert: [100%] reading DICOM series "BRI 64 directions ep2d_diff_3scan_trace_p2"
mrconvert: [100%] reformatting DICOM mosaic images
mrconvert: [100%] copying from "DICOM data...ns ep2d_diff_3scan_trace_p2" to "dwi.nii.
→gz"
mrconvert: [100%] compressing image "dwi.nii.gz"
```

converts the DWI data in the `DICOM/` folder to *Compressed NIfTI (.nii.gz)*, and exports the DW gradient table to *FSL format* if found in the DICOM headers, resulting in a pair of `bvecs` & `bvals` files.

Importing the DW gradient table

If the *image header* already contain the DW information, then no further action is required - the *MRtrix3* application will be able to find it and use it directly. If this is not the case (e.g. the image format does not support including it in the header), or the information contained is not correct, *MRtrix3* applications also allow the DW gradient table to be imported using the `-grad` option (for the *MRtrix format*) or the `-fslgrad` option (for the *FSL format*). Note that this will override the information found in the image headers if it was there. This can be used during conversion using `mrconvert`, or at the point of use. For example:

```
$ mrconvert dwi.nii -fslgrad dwi_bvecs dwi_bvals dwi.mif
```

will convert the `dwi.nii` from *NIfTI & NIfTI-2 (.nii)* to *MRtrix image formats (.mih / .mif)*, embedding the DW gradient table information found in the `dwi_bvecs` & `dwi_bvals` files (in *FSL format*) directly into the output image header. As another example:

```
$ dwi2tensor DICOM/ -grad encoding.b tensor.nii
```

will process the DWI dataset found in the `DICOM/` folder (in *DICOM (folder or .dcm)* format), but *override* any DW gradient information in the DICOM data with the table stored in the *MRtrix format* file `encoding.b`.

5.28.3 Operations performed by *MRtrix3* when handling DW gradient tables

Most *MRtrix3* applications that don't actually need to interpret the DW gradient table will typically simply pass the information through to the output unmodified. Any information found in the input image header – including the DW gradient table – is simply written to the output image header if the image format supports it (i.e. if the output is in *MRtrix image formats (.mih / .mif)* – DICOM is not supported for writing). If the output image format does not allow storing the DW gradient table in the image header, the `-export_grad_mrtrix` or `-export_grad_fsl` options can be used to write it out to separate files, ready for use with third-party applications, or directly within *MRtrix3* if users prefer to keep their data organised in this way.

However, any *MRtrix3* application that manipulates the DW gradient table in any way (for example, using the `-grad` or `-fslgrad` option) will perform a number of sanity checks and modifications to the information in the DW gradient table, depending on the nature of the operation, and its original format. This includes applications such as *mrconvert*, *mrinfo*, *mrcat*, and other most obvious DW-specific applications such as *dwi2tensor* and *dwi2fod*.

The specific steps performed by *MRtrix3* include:

- verifying that the number of volumes in the DWI dataset matches the number of entries in the DW gradient table;
- normalising the gradient vectors to unit amplitude;
- if required, scaling the *b*-values by the square of the gradient vector amplitude – see *b-value scaling* for details.
- where relevant, verifying that the DW gradient tables contains the data in a shell structure, by clustering similar *b*-values together (see *mrinfo*'s `-shell_bvalues` and `-shell_sizes` options);

Note: *mrinfo* will also perform most of these checks. While there is no technical reason for it to interpret the DW gradient information, in practice it is generally helpful to view the information as it would be interpreted by other *MRtrix3* applications. If you need to display the raw DW gradient table before any modification, use *mrinfo* with the `-property dw_scheme` option.

5.28.4 *b*-value scaling

On MRI scanners that do not explicitly allow for multi-shell datasets, a common workaround is to set the scanning protocol according to the largest desired *b*-value, but use gradient vector directions that have *less than unit norm*. This results in diffusion sensitisation gradients with reduced strength, and hence images with lower *b*-values.

For example, if this was the desired gradient table:

0	0	0	0
1	0	0	700
1	0	0	2800

This could be achieved on some systems by supplying this custom diffusion vectors file, now nominally containing only *b* = 0 and *b* = 2800 s/mm²:

0	0	0	0
0.5	0	0	2800
1	0	0	2800

By default, *MRtrix3* applications will detect this and **automatically** scale the *b*-values by the squared amplitude of the gradient vectors if required (so that the stored gradient table is equivalent to the first example), in order to more sensibly reflect the nature of the image data. Note that this is only applied if the DW gradient table looks like it corresponds to a multi-shell scheme, which is detected heuristically based on whether the gradient vector norms deviate from unity by more than 1%.

While this scaling allows such datasets to be processed seamlessly, it may introduce unexpected variations in the *b*-values for other datasets. Alternatively, if the provided diffusion gradient table is malformed, and contains the correct *b*-values but non-unity-norm directions, this scaling will result in a reported diffusion gradient table that contains *b*-values other than those expected.

If this scaling becomes a problem (e.g. for third-party applications), this feature can be explicitly enabled or disabled using the `-bvalue_scaling` option in *mrconvert* when initially importing or converting the raw data.

When using the FSL format

In this format, the gradient vectors are provided relative to the image axes (as detailed in the [FSL wiki](#)). To convert them to the internal representation used in *MRtrix3* (and in the *MRtrix format* gradient table), these vectors need to be transformed into the real / scanner coordinate system. To do this requires knowledge of the DWI dataset these vectors correspond to, in particular the image transform. In essence, this consists of rotating the gradient vectors according to the rotation part of the transform (i.e. the top-left 3×3 part of the matrix). This will introduce differences between the components of the gradient vectors when stored in *MRtrix format* compared to the *FSL format*, particularly for images not acquired in a pure axial orientation (i.e. images where the rotation part of the image transform is identity). Indeed, as mentioned earlier, there is an additional confound related to the handed-ness of the coordinate system; see the [FSL wiki](#) for details.

Warning: Never perform a manual conversion between MRtrix and FSL gradient table formats using a text editor or basic shell script. This poses a risk of introducing an unwanted rotation / reflection of the gradient directions, with concomitant errors in later processing.

Note that in this operation, what matters is the transform as stored in the NIfTI headers (i.e. the `sform / qform`); the transform as reported by *mrinfo* can differ substantially from this (while still being consistent with the data), as the *MRtrix3* image loading backend will try to provide the image transform in a near-axial orientation (by inverting / exchanging columns of the transform, and adjusting the *Strides* to match - see [The image transform](#) for details). To find out the actual transform that was stored in the NIfTI header, use *mrinfo* with the `-config RealignTransform false` option.

5.29 Phase encoding scheme handling

From version 3.0RC1 onwards, *MRtrix3* is capable of importing information from DICOM relating to the phase encoding of the acquired images, and encoding this information within key-value fields inside an image header. This information can then later be used by the *dwifslpreproc* script, specifically using its `-rpe_header` command-line option, to automatically set up and execute FSL's *topup* and *eddy* commands without requiring explicit input from the user regarding the phase encoding design of the imaging experiment. This page explains how this information is encoded and manipulated by *MRtrix3*.

Note: Due to variations in sequences, acquisition protocols, DICOM encoding, and image manipulation operations, *MRtrix3* cannot be guaranteed to obtain or retain the phase encoding information correctly in all circumstances. Manual inspection of these data and/or the outcomes of image processing when making use of these data is advocated. If

circumstances leading to incorrect import or encoding of this information in *MRtrix3* is reproducible, please report this to the *MRtrix3* developers so that the software can be updated to support such data.

5.29.1 Phase encoding information storage

The phase encoding information for a particular image file can be stored in one of two ways.

- The most convenient of these is storage of (one or more) key-value field(s) encapsulated within the image header, just as can be used for the diffusion gradient scheme. This ensures that the information is retained through image processing, as each *MRtrix3* command passes the header entries of the input image through to the output image.
- Alternatively, this information can be stored within a JSON file that accompanies the relevant image file(s). This information would then typically be imported / exported using the `-json_import` and `-json_export` options in `mrconvert`.

Precisely *how* this phase encoding information is *encoded* however depends on the nature of the phase encoding information for that image; specifically, whether the phase encoding information is *identical* for all volumes within an image file (or if it contains just one volume), or whether the phase encoding information *varies* between volumes within the image series.

Fixed phase encoding

In the case where both the phase encoding *direction* and the EPI *total readout time* are equivalent for all volumes within an image, this information is encoded within two fields: “PhaseEncodingDirection” and “TotalReadoutTime”. These fields are consistent with *BIDS* (the Brain Imaging Data Structure).

PhaseEncodingDirection can take one of six values: “i”, “i-”, “j”, “j-”, “k”, “k-”. These correspond to the first, second and third axes of the corresponding image. Exactly how these axes map to “real” / “scanner” space axes (i.e. left-right, posterior-anterior, inferior-superior) depends on the *The image transform* that is stored in the respective image header, which may potentially have been modified internally by *MRtrix3* on image load (discussed further in the *Non-axial acquisitions* section). Here, we begin with the most simple use case:

Take an image that conforms to the RAS (Right-Anterior-Superior) convention common to both NIfTI and *MRtrix3*. If the phase encoding is applied A>>P (anterior-posterior), this is the *second* spatial axis, or axis “j”. However, the phase encoding is also *reversed* along that axis (“RAS” indicates that voxel positions along the second axis *increase* when moving toward the anterior of the brain, whereas “A>>P” indicates the opposite). Hence, in this example, PhaseEncodingDirection would have the value “j-”.

Note: The phase encoding direction is defined specifically with respect to *image axes*. It is therefore *not* affected by the image *strides*, which only affect how the data for these axes are arranged when they are stored as a one-dimensional list of values within a file.

Note: The phase encoding direction does *not* relate to “x”, “y” and “z” axis directions in “real” / “scanner” space, as do other representations of orientation information in *MRtrix3*. This is because phase encoding specifically affects the appearance of the image *along the image axis in which phase encoding was applied*. The mapping from image axes to real-space axes is determined by the image header transform.

To demonstrate this: Imagine if an EPI image were to undergo a rigid-body rotation. The EPI field inhomogeneity distortions would *still align* with the relevant image axis after this rotation; but the effective x-y-z direction of phase encoding in real-space would change.

`TotalReadoutTime` provides the total time required for the EPI readout train. Specifically, this is the time between the centre of the first echo, and the centre of the last echo, in the train; this is consistent with BIDS, and is sometimes referred to as the “FSL definition”, since it is consistent with relevant calculations performed within FSL tools. It should be defined in seconds.

Variable phase encoding

If the phase encoding direction and/or the total readout time varies between different volumes within a single image series, then the two key-value fields described above are not sufficient to fully encode this information. In this situation, *MRtrix3* will instead use a key-value entry “`pe_scheme`” (similar to the “`dw_scheme`” entry used for the diffusion gradient scheme).

This information is stored as a *table*, where each row contains the phase encoding direction and the readout time for the corresponding volume; the number of rows in this table must therefore be equal to the number of volumes in the image. In each row, the first three numbers encode the phase encoding direction, and the fourth number is the total readout time. The direction is specified as a unit direction in the image coordinate system; for instance, a phase encoding direction of A>>P would be encoded as `[0 -1 0]`.

5.29.2 Non-axial acquisitions

When images are acquired in such a manner that the axes do not approximately correspond to RAS convention (i.e. first axis increases from left to right, second axis increases from posterior to anterior, third axis increases from inferior to superior), *MRtrix3* will automatically alter the axis *Strides* & transform in order to make the image *appear* as close to an axial acquisition as possible. This is briefly mentioned in *The image transform* section. The behaviour may also be observed by running `mrinfo` with and without the `-config RealignTransform false` option, which temporarily disables this behaviour.

Because phase encoding is defined with respect to the image axes, any transformation of image axes must correspondingly be applied to the phase encoding data. When the phase encoding information is stored within the image data, *MRtrix3* should automatically manipulate such phase encoding information in order to maintain correspondence with the image data.

Where management of such information becomes more complex and prone to errors is when it is included in the sidcar information of a JSON file, e.g. as is commonly now utilised alongside NIfTI images such as in the BIDS format. This becomes more complex at both read and write stages, each in their own complex way.

1. When *reading* a JSON file, *MRtrix3* will take the transformation that was applied to the corresponding input image, and apply that to the phase encoding information.

This has two curious consequences:

1. Running:

```
mrconvert image.nii -json_import image.json - | mrinfo - | grep ↵
↵PhaseEncodingDirection
```

and:

```
cat image.json | grep PhaseEncodingDirection
```

may produce different results. This is because once imported via the `-json_import` option, the phase encoding direction is altered to reflect *how MRtrix3 interprets the image data*, rather than how they are actually stored on file.

2. Running:


```
mrconvert image.nii - | mrconvert - -json_import image.json image.mif
```

may produce the incorrect result. This is because information regarding the transformation that is applied to the NIfTI image in the *first* `mrconvert` call in order to approximate an axial acquisition is *no longer available* in the *second* `mrconvert` call. When the `-json_import` command-line option is used, it is interpreted with respect to the input image *for that command* - which, in the above case, is an MRtrix piped image to which the axial transformation has *already been applied* - and so must *always* be used immediately in conjunction with loading the image with which that JSON file is associated.

2. When *writing* a JSON file, *MRtrix3* will attempt to modify the phase encoding information in order to conform to the limitations of the output image format alongside which the JSON file is intended to reside.

Unlike the *MRtrix image formats* (*.mih / .mif*), *NIfTI & NIfTI-2* (*.nii*) do not support arbitrary image strides. When writing image data with non-trivial strides to a NIfTI image, *MRtrix3* will reorder the three spatial axes in order to approximate an axial alignment, performing the corresponding modifications to the image transform. Any exported JSON file must therefore also have the same transformation applied.

This has the curious consequence that:

```
mrconvert input.mif output.mif -json_export output.json
cat output.json | grep PhaseEncodingDirection
```

and:

```
mrconvert input.mif output.nii -json_export output.json
cat output.json | grep PhaseEncodingDirection
```

may produce different results. A JSON file *must* only be interpreted in conjunction with the singular image file alongside which it was generated. In the case of the MRtrix image format, we suggest relying instead on the storage of sidcar information within the *Header key-value pairs* rather than in such an external file, as it allows *MRtrix3* to apply any requisite modifications to such data to echo modifications to the image without user intervention.

Note: This concept also has consequences for the `dwifslpreproc` script when manually providing the phase encoding direction. The axis and sign of phase encoding provided to the script must reflect the direction of phase encoding *after* *MRtrix3* has performed this transformation, i.e. as it is read by any *MRtrix3* command or as it appears in `mrview`, *not* the actual encoding of axes within the file.

5.29.3 Manipulation of phase encoding data

The primary purpose of storing this phase encoding information is to automate the correction of EPI susceptibility distortions. However this can only occur if the information stored is not invalidated through the manipulation of the corresponding image data. Therefore, any *MRtrix3* command that is capable of manipulating the image data in such a way as to invalidate the phase encoding information will *automatically* modify this phase encoding information appropriately. This includes modifying the representation of this information between the fixed and variable phase encoding cases.

Consider, for instance, a pair of `b=0` images, where the first was acquired with phase encoding direction `A>>P`, and the second was acquired using phase encoding direction `P>>A`:

```
$ mrinfo AP.mif
*****
Image:                AP.mif
*****
```

(continues on next page)

(continued from previous page)

```

...
PhaseEncodingDirection: j-
TotalReadoutTime: 0.0575
...

$ mrinfo PA.mif
*****
Image:                PA.mif
*****
...
PhaseEncodingDirection: j
TotalReadoutTime: 0.0575
...

```

Now watch what happens when we concatenate these two images together:

```

$ mrcat AP.mif PA.mif AP_PA_pair.mif -axis 3
mrcat: [100%] concatenating "AP.mif"
mrcat: [100%] concatenating "PA.mif"
# mrinfo AP_PA_pair.mif
*****
Image:                AP_PA_pair.mif
*****
...
pe_scheme:            0,-1,0,0.0575
                     0,1,0,0.0575
...

```

When the two input images are concatenated, *MRtrix3* additionally concatenates the phase encoding information of the input volumes; since it detects that these are not consistent between volumes, it stores this information using the `pe_scheme` header entry, rather than `PhaseEncodingDirection` and `TotalReadoutTime`.

The `mrconvert` command has a number of additional functionalities that can be used to manipulate this information:

- The `-import_pe_table` and `-export_pe_table` options can be used to import/export the phase encoding information from / to file as a table, i.e. in the format used for the `pe_scheme` header entry described above. Note that even if all volumes in the image have the same phase encoding direction and total readout time, these options will still import / export these data in table format.
- The `-import_pe_eddy` and `-export_pe_eddy` options can be used to import/export the phase encoding information in the format required by FSL's eddy tool. The [FSL documentation page](#) describes this format in more detail.
- The `-json_import` and `-json_export` options can be used to import/export *all* header key-value entries from/to an external JSON file. This may be useful in particular for operating within the BIDS specification. There is a caveat here: If you use the `-json_export` option on an image with *fixed* phase encoding, the `PhaseEncodingDirection` and `TotalReadoutTime` fields will be written as expected by BIDS; however if the image contains *variable* phase encoding, then the `pe_scheme` header entry will be written to the JSON file, and this will not be appropriately interpreted by other BIDS tools.
- The `-set_property` option may be useful to *override* these header entries if they are deemed incorrect by some other source of information.

5.30 Global intensity normalisation

Several existing DWI models derive quantitative measures by fitting a model to the ratio of the DW signal to the $b=0$ signal within each voxel. Such a voxel-wise division by the original $b=0$ signal removes intensity variations due to T2-weighting and RF inhomogeneity. However, unless all compartments within white matter (e.g. intra- and extra-axonal space, myelin, cerebral spinal fluid (CSF) and grey matter partial volumes) are modelled accurately (i.e. with appropriate assumptions/modelling of both the compartment diffusion and T2), the proportion of one compartment in a voxel may influence another. For *example*, if CSF partial volume (e.g. at the border of white matter and the ventricles) is not taken into account, then a voxel-wise division by the $b=0$ (which has a long T2 and appears much brighter in CSF than in white matter in the T2-weighted $b=0$ image), will artificially overreduce the DW signal from the white matter intra-axonal (restricted) compartment, ultimately changing several derived quantitative measures.

A previous work investigating differences in Apparent Fibre Density (AFD) [Raffelt2012] opted to instead perform a *global* intensity normalisation between subjects. This avoids the aforementioned issues, but also comes with its own set of challenges and assumptions inherent to specific strategies to deal with intensity normalisation for diffusion MRI data. Aside from the problem of how to define a reference region for global intensity normalisation (that is unbiased with respect to the groups in the analysis), the data must also be *bias field* corrected, to eliminate low frequency (spatially smooth) intensity inhomogeneities across the image.

In theory, an approach to global intensity normalisation could for example be to normalise using the median CSF $b=0$ intensity for each subject as a reference (under the assumption that the CSF T2 is unlikely to be affected by pathology). However, in practice it is surprisingly difficult to obtain a robust partial-volume-free estimate of the CSF intensity due to the typical low resolution of DW images. For healthy participants less than 50 years old, reasonably small ventricles make it quite difficult to identify *pure* CSF voxels at 2-2.5mm resolutions. Therefore, performing global intensity normalisation using the median *white matter* $b=0$ intensity may be easier to achieve. While the white matter $b=0$ intensity may be influenced by pathology-induced changes in T2, the assumption then becomes that such changes would be (spatially) quite local and therefore have little influence on the median white matter $b=0$ value.

The *dwinormalise group* script is included in MRtrix to perform an automatic global normalisation using the *median white matter $b=0$ intensity*. The script input requires two folders: a folder containing all DW images in the study (in .mif format) and a folder containing the corresponding whole brain mask images (with the same filename prefix). The script runs by first computing diffusion tensor Fractional Anisotropy (FA) maps, registering these to a groupwise template, then thresholding the template FA map to obtain an *approximate* white matter mask. The mask is then transformed back into the space of each subject image and used in the *dwinormalise individual* command to normalise the input DW images to have the same $b=0$ white matter median value. All intensity normalised data will be output in a single folder. As previously mentioned, all DWI data must be bias field corrected *before* applying *dwinormalise group*, for example using *dwibiascorrect*. Users are well advised to (manually) check the results of *dwinormalise group* closely though, as occasional instabilities have been observed in the outcomes of particular subjects.

In case of pipelines that include a multi-tissue spherical deconvolution algorithm yielding compartment estimates for multiple different tissues [Jeurissen2014] [Dhollander2016a], a *new* command called *mtnormalise* can be used instead, which performs multi-tissue informed intensity normalisation in the log-domain, correcting simultaneously for both global intensity differences as well as bias fields. The benefit of the *mtnormalise* command is that normalisation can be performed *independently* on each subject, and therefore does *not* require a computationally expensive (and potentially not entirely accurate) registration step to a group template.

5.31 Spherical Harmonics

For Spherical Deconvolution (SD) as implemented in MRtrix, processing is done in the Spherical Harmonic (SH) basis; this mathematical formulation provides a smooth representation of data distributed on the sphere. When we do SD, the resulting Fibre Orientation Distributions (FODs) are written to an image. These FOD images contain coefficients in this SH basis, that when interpreted correctly, produce the FOD butterflies we all know and love. If you've ever looked at the raw image volumes from an FOD image, you'll know that all but the first one are basically not interpretable.

5.31.1 What are Spherical Harmonics?

Spherical harmonics are special functions defined on the surface of a sphere. They form a complete orthonormal set and can therefore be used to represent any well-behaved spherical function. In many ways, they are the equivalent to the Fourier series for functions defined over spherical (rather than Cartesian) coordinates. They are defined as:

$$Y_l^m(\theta, \phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos \theta) e^{im\phi}$$

with integer *order* l and *phase* m , where $l \geq 0$ and $-l \leq m \leq l$ (note that the terms *degree* and *order* are also commonly used to denote l & m respectively), and associated Legendre polynomials P_l^m . The harmonic order l corresponds to the angular frequency of the basis function; for example, all $l = 2$ SH basis functions feature 2 full oscillations around some equator on the sphere. The harmonic phase m correspond to different orthogonal modes at this frequency; e.g. where the oscillations occur around a different plane.

Any well-behaved function on the sphere $f(\theta, \phi)$ can be expressed as its spherical harmonic expansion:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_l^m Y_l^m(\theta, \phi)$$

For smooth functions that have negligible high angular frequency content, the series can be truncated at some suitable maximum harmonic order l_{\max} with little to no loss of accuracy:

$$f(\theta, \phi) = \sum_{l=0}^{l_{\max}} \sum_{m=-l}^l c_l^m Y_l^m(\theta, \phi)$$

The spherical harmonic series therefore provides a compact representation for smooth functions on the sphere. Moreover, due to its formulation, it has many compelling mathematical properties that simplify otherwise complex operations, including notably spherical (de)convolution.

5.31.2 Formulation used in MRtrix3

Due to the nature of the problems addressed in diffusion MRI, in *MRtrix3* a simplified version of the SH series is used:

1. The data involved are real (the phase information is invariably discarded due to its instability to motion), so we can use a real basis with no imaginary components.
2. The problems involved all exhibit antipodal symmetry (i.e. symmetry about the origin, $f(\mathbf{x}) = f(-\mathbf{x})$), so we can ignore all odd order terms in the series (since these correspond to strictly antisymmetric terms).

The SH basis functions $Y_{lm}(\theta, \phi)$ used in *MRtrix3* are therefore:

$$Y_{lm}(\theta, \phi) = \begin{cases} 0 & \text{if } l \text{ is odd,} \\ \sqrt{2} \operatorname{Im} [Y_l^{-m}(\theta, \phi)] & \text{if } m < 0, \\ Y_l^0(\theta, \phi) & \text{if } m = 0, \\ \sqrt{2} \operatorname{Re} [Y_l^m(\theta, \phi)] & \text{if } m > 0, \end{cases}$$

Storage conventions

Images that contain spherical harmonic coefficients are stored as 4-dimensional images, with each voxel's coefficients stored along the fourth axis. Only the even degree coefficients are stored (since odd l coefficients are assumed to be zero).

The SH coefficients c_{lm} corresponding to the basis functions $Y_{lm}(\theta, \phi)$ are stored in the corresponding image volume at index V_{lm} according to the following equation:

$$V_{lm} = \frac{1}{2}l(l+1) + m$$

The first few volumes of the image therefore correspond to SH coefficients as follows:

Volume V_{lm}	c_{lm}
0	$l = 0, m = 0$
1	$l = 2, m = -2$
2	$l = 2, m = -1$
3	$l = 2, m = 0$
4	$l = 2, m = 1$
5	$l = 2, m = 2$
6	$l = 4, m = -4$
7	$l = 4, m = -3$
...	...

The total number of volumes N in the image depends on the highest angular frequency band included in the series, referred to as the maximal spherical harmonic order l_{\max} :

$$N = \frac{1}{2}(l_{\max} + 1)(l_{\max} + 2)$$

l_{\max}	N
0	1
2	6
4	15
6	28
8	45
10	66
12	91
...	...

Representation of response functions

The response functions required when performing spherical (de)convolution correspond to axially symmetric kernels (they typically represent the ideal signal for a coherently aligned bundle of fibres aligned with the z axis). Due to this symmetry, all $m \neq 0$ coefficients can be assumed to be zero. Therefore, response functions can be fully represented using only their even order l , zero phase $m = 0$ coefficients (the so-called *zonal* harmonics).

Response functions are stored in plain text files. A vector of values stored on one row of such a file are interpreted as these even l , $m = 0$ terms. The number of coefficients stored for a response function of a given maximal harmonic order l_{\max} is $1 + l_{\max}/2$.

Response files can contain multiple rows, in which case they are assumed to represent a *multi-shell* response, with one set of coefficients per b -value, in order of increasing b -value (i.e. the first row would normally correspond to the $b = 0$ ‘shell’, with all $l > 0$ terms set to zero). The b -values themselves are not stored in the response file, but are assumed to match the values in the DW encoding of the diffusion MRI dataset to be processed.

5.31.3 Differences with previous version of MRtrix

An important difference between the old (0.2.x) and new (0.3.x and 3.x.x) versions of MRtrix is a change to the Spherical Harmonic (SH) basis functions. This change has important consequences for data that may be used interchangeably between the two versions.

Important: note that although it is possible to use and display FODs generated using MRtrix 0.2.x in the newer *MRtrix3* applications (and vice-versa), the FODs will *NOT* be correct. Moreover, it is very difficult to tell the difference by simple visual inspection - the FODs may still *look* reasonable, but will give incorrect results if used for tractography or in quantitative analyses. To ensure your images are correct, you should use the *shbasis* application included in *MRtrix3*, as described below.

The problem

Here's where it gets tricky. In all previous versions of MRtrix, there was a 'bug' in the SH basis functions. Mathematically, the basis was 'non-orthonormal' (although still orthogonal), due to the omission of the $\sqrt{2}$ terms in the definitions above. You don't necessarily need to know what this means, just appreciate that this formulation of the basis, although entirely self-consistent, was not optimal for some operations.

This 'bug' didn't actually cause any problems; the previous version of MRtrix was self-consistent in its handling of the issue throughout the code. It was annoying for any users transferring data between MRtrix and other packages though. For the release of the new *MRtrix3*, we have decided to correct the underlying error in the SH basis once and for all, as there are various mathematical operations that are greatly simplified when the basis is orthonormal. This does however introduce a problem for anyone that has done prior image processing using the old MRtrix 0.2 and wants to be able to use that data with *MRtrix3*: if you have image data that was generated using the *old* SH basis, but read it using MRtrix code that was compiled using the *new* SH basis, the data will *not be interpreted correctly*.

The solution

There is a solution, but it takes a bit of manual labour on your part. We have provided a new command called *shbasis*. This command will read your image data, and tell you which SH basis it thinks your image data are stored in (or if it's unable to make this decision).

Furthermore, it includes a command-line option for *changing* the SH basis of the underlying image data: `-convert`. The most important choice for this option is `-convert native`. This option identifies the SH basis that *MRtrix3* is compiled for (this is the new orthonormal basis by default); and if the image data is not currently stored in this basis, it *modifies the image data in-place* so that it conforms to the correct basis.

Any data that you generate after this update has occurred will automatically be produced in the new SH basis, and therefore will not need to be converted using *shbasis*. However if you are uncertain whether or not a particular image does or does not need to be converted, *shbasis* can always be used to verify whether or not the image data are in the correct SH basis; and if you provide the `-convert native` option despite the image data already being in the new SH basis, no modification of the image data will take place.

My recommendation is therefore as follows. When you commit to using the new version of MRtrix, you should go through *all* of your diffusion image data on *all* systems that you use, and run `shbasis -convert native` on all images that contain spherical harmonic data (only FOD images; raw DWIs / response functions / TDIs / etc. do not need to be converted).

Also: Remember that data previously generated will not be interpreted correctly by *MRtrix3* commands without the SH basis conversion? The same applies in the other direction. So if you load FOD images that have either been generated using *MRtrix*, or have been previously converted using *shbasis*, commands from the previous version of MRtrix (0.2) won't interpret them correctly. We hope that once we have feature completeness in *MRtrix3*, the old version will no longer be necessary, and therefore this will not be a problem.

Dealing with problematic data

In some circumstances, the *shbasis* command will give an error something like this:

```
shbasis [WARNING]: Cannot make unambiguous decision on SH basis of image csd.
mif (power ratio regressed to l=0 is 1.58446)
```

`shbasis` uses a data-driven approach to automatically determine the SH basis that the image data are currently stored in; however a number of issues can arise that lead to a breakdown of the numerical assumption that it is based on, and it can no longer make this decision.

If this occurs, but you are confident that your image data are in the old non-orthonormal basis and need to be converted to the new orthonormal basis, you can run: `shbasis <image> -convert force_oldtonew`. This will inform `shbasis` that even though it's unable to determine the current SH basis, you're confident that you do know it, and therefore it should perform the conversion anyway. It will give you a couple of loud warnings just to make sure you appreciate the danger in what you're doing, so you should only ever use this setting for problematic data; for the vast majority of conversions, `-convert native` is much better.

5.32 Maximum spherical harmonic degree l_{\max}

The l_{\max} value refers to the highest harmonic order l included in the *Spherical Harmonics* series. It determines the highest angular frequency band to be included in the series: higher l_{\max} values allow sharper details to be represented, but require more coefficients to be stored and processed.

5.32.1 What determines l_{\max} for my image data?

For any command or script operating on data in the spherical harmonic basis, it should be possible to manually set the maximum harmonic degree of the output using the `-lmax` command-line option. If this is *not* provided, then an appropriate value will be determined automatically.

The mechanisms by which this automatic determination of l_{\max} occurs are as follows:

- Determine the maximum value for l_{\max} that is supported by the number of DWI volumes in the shell being processed (or the total number of non- $b=0$ volumes in a single-shell acquisition). This is the number of coefficients required to store an antipodally-symmetric spherical harmonic function:

l_{\max}	Required volumes
2	6
4	15
6	28
8	45
10	66
12	91
...	...

- If l_{\max} exceeds 8, reduce to 8. This is primarily based on the findings in [this paper](#).
- Check the condition of the transformation between DWIs and spherical harmonics. If the transformation is ill-conditioned (usually indicating that the diffusion sensitisation gradient directions are not evenly distributed over the sphere or half-sphere), reduce l_{\max} until the transformation is well-conditioned.

As an example: concatenating two repeats of a 30 direction acquisition to produce 60 volumes will *not* support an $l_{\max}=8$ fit: the angular resolution of the data set is equivalent to 30 *unique* directions, and so $l_{\max}=6$ would be selected (and this would be accompanied by a command-line warning to the user).

- In the case of spherical deconvolution, the l_{\max} selected for FOD estimation will also be reduced if l_{\max} of the provided response function is less than that calculated as above.

The *exception* to these rules is the new `amp2response` command, which is now called by default in all `dwi2response` script algorithms. This command converts *amplitudes* on the half-sphere (most likely in the form of raw DWI image intensities) into a *response function* intended for use in spherical deconvolution. This command behaves differently for two reasons in combination:

- The image data from multiple voxels are combined together in a single fitting procedure, therefore having a much greater number of samples when performing the transformation.
- The data are transformed not to the spherical harmonic basis, but directly to the *zonal* spherical harmonic basis (this is the spherical harmonic basis containing only the $m = 0$ terms). This basis requires far fewer coefficients for any given value of l_{\max} : 2 for $l_{\max}=2$, 3 for $l_{\max}=4$, 4 for $l_{\max}=6$, 5 for $l_{\max}=8$ and so on.

The value of l_{\max} that can be used in this command is therefore practically unconstrained; though the power in higher harmonic degrees is much smaller than that in lower degrees. The command is currently configured to select $l_{\max}=10$ by default, regardless of b -value; interested readers can find the discussion [here](#).

5.32.2 Reduced l_{\max} in particular subjects

If you find that certain subjects within a cohort have a reduced l_{\max} compared to the rest of the cohort when using any command relating to spherical harmonics, the most likely cause is premature termination of the diffusion sequence during scanning of those subjects, resulting in a reduced number of diffusion volumes, and therefore a reduced l_{\max} according to the table above.

5.32.3 Setting l_{\max} in different applications

The range of permissible values for l_{\max} depends on the particular command being used; e.g.:

- For any command that maps image data directly to spherical harmonics, it is impossible to set l_{\max} to a value higher than that supported by the image data. The transformation from DWI data to spherical harmonics simply cannot be done in such a case, as the problem is under-determined. You can of course set l_{\max} to a lower value than that supported by the data.
- In spherical deconvolution, it *is* possible to set a higher l_{\max} than that supported by the data - so-called *super-resolved* spherical deconvolution. Here, additional information is provided by the non-negativity constraint to make estimation of additional spherical harmonic coefficients possible.
- If performing Track Orientation Density Imaging (TODI) using `tkmap -tod`, then the apodized point spread functions (aPSFs) can be generated at any value of l_{\max} for which aPSF data are available (currently $l_{\max}=16$, since the angular resolution of the original image data is not a limiting factor here).
- As described previously, the `amp2response` command is a special case, and the maximum permissible l_{\max} is vastly greater than the maximum practical value.

5.33 “Fixels” (and “Dixels”)

Internally we have created a couple of new terms that we find invaluable when discussing diffusion MRI processing methods and statistics. We’d like to share these with our user base in the hope that others will gain advantages from using the same terminology, and also so that we all know what everyone else is talking about! Anyone using *MRtrix3* to develop their own software may also see these terms scattered throughout the library code, so will need to know what they represent.

All *MRtrix3* users should be familiar with the terms ‘pixel’ and ‘voxel’; these are abbreviations of “picture element” and “volume element”, corresponding to the smallest element within a 2D picture and 3D volume respectively. However in Diffusion MRI we also deal with *orientation* information within each image volume element; so we wanted terminology to allow us to convey the types of discrete elements that we deal with on a daily basis.

We have settled on the following two terms. The first of these, ‘fixel’, will appear frequently throughout the *MRtrix3* documentation and in online discussions, and will therefore satisfy the requirements of the majority of users. The second, ‘dixel’, is typically reserved for internal technical discussion; however due to its occasional usage (and its

inconsistent use in early presentations, see final note below), we are additionally providing its full definition here for interested readers.

5.33.1 ‘Fixel’: *Fibre bundle element*

The term *fixel* refers to a *specific fibre bundle* within a *specific voxel*. Alternatively, consistently with the definitions of ‘pixel’ and ‘voxel’, it can be thought of as a “fibre bundle element”: the smallest discrete component of a fibre bundle. Each fixel is parameterized by the voxel in which it resides, the estimated mean orientation of the underlying fibres attributed to that bundle, a fibre density (or partial volume fraction), and potentially other metrics.

In reality, fixels have been used in the field of Diffusion MRI for a long time: multi-tensor fitting, ball-and-sticks, any diffusion model that is capable of fitting multiple anisotropic elements to each image voxel, can be considered as estimating fixels. However in the past, researchers have resorted either to lengthy descriptive labels in an attempt to express the nature of the data being manipulated, or have adopted existing terms, which can lead to confusion with the original sense of the terms. Furthermore, these labels are not applied inconsistently between publications; we hope that the term ‘fixel’, being unambiguous with other interpretations of “fibre bundle” or “fascicle” or other examples, will slowly become the standard term for describing these data.

Historically, in MRtrix we are accustomed to dealing with FODs that are *continuous* functions on the sphere, rather than having a discrete number of fibre directions in each voxel. However, if the FOD is *segmented* in any way (either through peak-finding as shown in [this paper](#) and implemented in the `sh2peaks` command, the segmentation algorithm described in the appendices of the [SIFT NeuroImage paper](#) and provided in the `fod2fixel` command, or more advanced methods), each discrete feature of a particular FOD can be labelled a ‘fixel’, as each represents a set of fibres within that voxel that form a coherent bundle in orientation space.

The term ‘fixel’ has now appeared in the literature with the publication of the statistical method, [Connectivity-based Fixel Enhancement](#), as well as the more general framework of [Fixel-Based Analysis](#), which together allow for the inference of group differences not just at the voxel level, but the *fixel* level; that is, if only one fibre bundle within a crossing-fibre voxel is affected in a cohort, we hope to both identify the bundle affected, and quantify the group effect that is specific to that bundle.

5.33.2 ‘Dixel’: *Directional Element*

This term is used less frequently, and hence may not be relevant for all readers. If you have not seen it used before, you may in fact prefer to avoid the following text in order to keep things simple. . .

Imagine a single image voxel, the data for which is in fact a function on the sphere (i.e. varies with orientation). We now take samples of that function along a set of pre-defined directions on the unit sphere. Each of those samples is referred to as a ‘dixel’: a *directional element* within a *specific voxel*. Each dixel is described by the voxel in which it resides, the direction along which the relevant spherical function was sampled, and the intensity of the function in that direction.

Importantly, it is the *combination* of the voxel location and sampling direction that describe the dixel. If a different direction were used to sample the spherical function, this would be a different dixel with a different associated value; likewise, if the spherical function in an adjacent voxel were sampled along the same direction, that would also be a different dixel with a different associated value. Each dixel is a unique sample of a particular spatially-varying spherical function.

Most commonly, the term ‘dixel’ is used to refer to the situation where a *set of directions* on the unit sphere has been used to sample some function; for instance, sampling the amplitudes of a Fibre Orientation Distribution (FOD), which is otherwise a continuous function expressed in the Spherical Harmonic (SH) basis. However, by the definition of the term, ‘dixel’ could also be used to describe a single voxel within a particular image volume in a HARDI experiment; if the HARDI signal in a single voxel is considered to be discrete samples of the orientation dependence of the diffusion signal in that voxel, then each of those samples could be labelled a ‘dixel’.

Therefore, the fundamental disambiguation between ‘fixels’ and ‘dixels’ is as follows:

- A ‘dixel’ is typically assumed to represent a sample of a spherical function along some pre-determined direction, where that direction belongs to some dense *basis set* of equally-distributed unit directions that has been used to sample an otherwise continuous (hemi-)spherical function.
- ‘Fixel’, on the other hand, is used to describe a *set of fibres* within a voxel that are sufficiently similar in orientation that they are indistinguishable from one another, and therefore form a fibre ‘bundle’ within that voxel.

Some observations / contexts in which the term ‘dixel’ may be useful:

- The `mrview` “ODF overlay” tool is capable of loading “Dixel ODFs”. These can be either a set of direction-based samples on the sphere, or it can be used to directly visualise the diffusion signal within a particular *b*-value shell, since both of these cases correspond to a set of directions on the unit hemisphere, where each direction has associated with it an ‘intensity’ / ‘amplitude’.
- In the original Apparent Fibre Density (AFD) [manuscript](#), the statistical analysis was performed by performing a t-test in each of 200 directions in each voxel, and then detecting connected clusters in position & orientation space. This can be thought of as “dixel-based cluster statistics”.
- In the FOD segmentation method provided in the `fod2fixel` command mentioned earlier, the algorithm first samples the amplitude of the FOD along a set of 1,281 directions, before identifying fixels based on accumulating these directions / samples. So this process can be thought of as converting the FOD from a continuous SH representation, to a dixel representation, then finally to a fixel representation.

Note: During the development of many of the aforementioned methods, a [presentation](#) was made at ISMRM demonstrating “Tractographic threshold-free cluster enhancement” (this is now referred to as “Connectivity-based Fixel Enhancement (CFE)”). During the presentation itself, the term ‘dixel’ was used to refer to a specific direction within a specific voxel; but a direction that corresponds to a particular fibre bundle in that voxel. You may observe that this definition is in fact consistent with what we have labelled here as a ‘fixel’, rather than a ‘dixel’; this is because at the time when this presentation was made, these two terms had not yet been disambiguated. The definitions made within this documentation page are what will be used from now on by the *MRtrix3* developers; and we hope by the wider community as well.

5.34 Motivation for `afdconnectivity`

Due to the interest in the `afdconnectivity` command, I thought I’d explain the reasoning behind the approach, the rationale behind the improvements made in commit 40ccdb62, and the argument for why we recommend the use of *Spherical-deconvolution Informed Filtering of Tractograms (SIFT)* as an alternative if possible.

The `afdconnectivity` command was originally written as a ‘hack’ for a colleague who wanted to obtain quantitative measures of ‘connectivity’ in the absence of EPI distortion correction. Without EPI distortion correction *Anatomically-Constrained Tractography (ACT)* cannot be applied, and consequently streamlines may terminate within white matter. Streamline count (as a measure of connectivity) between two grey matter regions will therefore not include those streamlines that terminate in white matter (and therefore the estimated connectivity may not be accurate).

The `afdconnectivity` command attempts to get around this issue by estimating a measure of ‘connectivity’ as follows:

- The integral of a discrete lobe of an FOD (*fixel*) is proportional to the volume of the MR-visible tissue (intracellular at high *b*-value) aligned in that direction.
- By taking a set of streamlines corresponding to a pathway of interest, and summing the integrals of all FOD lobes traversed by the bundle, you obtain an estimate of the total fibre volume of the pathway of interest.
- If you then divide by the length of the bundle (taken as the mean streamline length), you get an estimate of the cross-sectional area of the bundle, which is a measure of ‘connectivity’ independent of fibre length.

The major problem with this approach is the assumption that *all* of the fibre volume in each fixel traversed by the streamlines of interest belong to the bundle of interest; clearly not the case in various circumstances. The changes I have made to `afdconnectivity` are aimed at improving the behaviour in the presence of partial volume and erroneous streamlines.

The default behaviour is as before: determine a fixel mask using some bundle of streamlines, sum the apparent fibre density (a volume) of the fixels within the mask, and divide by mean streamline length (to get an estimate of cross-sectional area of the pathway).

Now, you can optionally provide a whole-brain fibre-tracking data set using the `-wbft` option (your bundle `.tck` file should then be a subset of this tractogram). In this case, the program determines the total streamlines density attributed to each fixel, and for those fixels traversed by the streamlines of interest, some fraction of the fibre volume of that fixel is contributed to the result. This fraction is determined for each fixel by the ratio of streamlines density from the bundle of interest, to the total streamlines density from the tractogram. The fibre volume of each fixel is therefore divided ‘fairly’ between the bundle of interest and the rest of the tractogram.

Although this may be an improvement in many circumstances, it’s still not our recommended method. Effectively what’s happening in this scenario is that for each streamline, a fibre volume is determined, based on its ‘fair share’ of each fixel it traverses. However this means that the effective cross-sectional area of that streamline is *allowed to vary drastically along its length*; this is clearly not physically realistic. Furthermore, due to the relative over- or under-reconstruction of different pathways in whole-brain fibre-tracking, there’s no guarantee that this proportional ‘sharing’ of fibre volume between streamlines is biologically accurate.

Now consider the alternative: filtering a tractogram using *Spherical-deconvolution Informed Filtering of Tractograms (SIFT)*, then selecting a subset of the remaining streamlines corresponding to your pathway of interest. By the model underlying SIFT, each streamline represents a constant cross-sectional area of fibres; so the *streamline count* becomes your estimate of bundle cross-sectional area and therefore ‘connectivity’ (with the SIFT proportionality coefficient providing the conversion between streamline count and AFD if you so choose).

This argument also holds if you are looking to use the image output from `afdconnectivity`, which provides the estimated fibre volume of the pathway of interest within each voxel. I have already stated why this is a poor interpretation with the default `afdconnectivity` behaviour; it’s improved with use of the `-wbft` option, but is noisy in regions where fixels are traversed by very few streamlines, and still may not share the fibre volume of each fixel appropriately. Again, SIFT provides the better alternative: an equivalent map can be produced by selecting your streamlines of interest post-SIFT, and running `tckmap -precise` (sums streamline lengths within each voxel rather than counting streamlines). Remember: a product of cross-sectional area and length gives a volume!

This is also an important message for interpretation of AFD results, both in this context and others. FOD amplitude (in any guise) is *in no way* a measure of “tissue integrity”, no matter how many quotation marks you use; it’s a measure of *density*. This is the reasoning behind the modulation step in *AFD*, and is the entire premise behind the SIFT method.

Anyways, rant over. We are considering writing a technical note that will discuss this issue, so we are trusting the *MRtrix3* beta user base not to do anything scientifically unethical with this information / command until we can create the relevant article for citation.

5.35 DICOM handling

MRtrix3 includes its own fast DICOM handling backend, allowing all *MRtrix3* applications to seamlessly support DICOM images as input. While this works well in most cases, it can fail in some circumstances. Issues specific to this format are outlined below.

5.35.1 How MRtrix3 handles DICOM data

When interpreting the argument provided for an input image to any *MRtrix3* command, the image handling backend will assume that the data are in DICOM format if the argument corresponds to a *folder*, or ends with the `.dcm` suffix.

The DICOM handling backend will then quickly scan through the files (recursively through the entire folder if one was provided), and build a table of contents. This consists of a *tree*, containing one or more *patients*, each containing one or more *studies*, each of which consists of one or more image *series*. When multiple choices are possible (e.g. multiple series are available), the application will present a menu to select the data of interest. For example:

```
$ mrinfo "Siemens Trio/"
mrinfo: [done] scanning DICOM folder "Siemens Trio/"
Select series ('q' to abort):
  0 -    9 MR images 10:20:52 localizer_sf (*fl2d1) [1]
  1 -   21 MR images 10:24:39 diff 20DW 2NEX ALL (*ep_b1000#17) [2]
  2 -   54 MR images 10:24:40 diff 20DW 2NEX ALL_ADC (*ep_b0_1000) [3]
  3 -   54 MR images 10:24:40 diff 20DW 2NEX ALL_TRACEW (*ep_b1000t) [4]
  4 -   54 MR images 10:24:40 diff 20DW 2NEX ALL_FA (*ep_b0_1000) [5]
  5 -   21 MR images 10:28:52 ep2d_diff_MDDW_AT_WIP (ep_b1000#2) [7]
  6 -   54 MR images 10:28:53 ep2d_diff_MDDW_AT_WIP_ADC (ep_b0_1000) [8]
  7 -   54 MR images 10:28:53 ep2d_diff_MDDW_AT_WIP_TRACEW (ep_b1000t) [9]
  8 -   54 MR images 10:28:54 ep2d_diff_MDDW_AT_WIP_FA (ep_b0_1000) [10]
?
```

The user should then enter the integer corresponding to the study of interest, for example (following from the above):

```
...
? 5
mrinfo: [100%] reading DICOM series "ep2d_diff_MDDW_AT_WIP"
*****
Image:          "TOURNIER DONALD (1) [MR] ep2d_diff_MDDW_AT_WIP"
*****
Dimensions:      84 x 84 x 54 x 21
Voxel size:      2.5 x 2.5 x 2.5 x ?
Data strides:    [ -1 -2 3 4 ]
Format:          DICOM
Data type:        unsigned 16 bit integer (little endian)
Intensity scaling: offset = 0, multiplier = 1
Transform:
                 0.9986   4.186e-08   -0.05229   -99.76
                -0.002193   0.9991   -0.04188   -83.83
                 0.05224   0.04193   0.9978   -43.07

EchoTime:        0.08
PhaseEncodingDirection: j-
TotalReadoutTime: 0.0252
comments:        TOURNIER DONALD (1) [MR] ep2d_diff_MDDW_AT_WIP
                  study: Head Brain_basic
                  DOS: 11/05/2007 10:28:52
dw_scheme:       0,0,0,0
[21 entries]     -0.99949686000000004,-0.0050327100000000001,-0.
→0050327100000000001,1000
...
                -0.11020774999999999,0.27551937999999998,0.95513387000000005,1000
                -0.031670320000000002,0.79175793999999999,0.57006572,1000
```

What happens at this stage is a deeper scan through those files that relate specifically to the data selected, to gather all the information required to read the data correctly. This two-stage process allows *MRtrix3* to scan through large datasets rapidly to allow the user to quickly select just those datasets of interest.

Selecting multiple matching series as a single dataset

It is also possible to load multiple series as a single dataset, when the image dimensions and other parameters match. This is done by using *Number sequences* to specify the series of interest. For example:

```
$ mrinfo "Siemens Trio/"
mrinfo: [done] scanning DICOM folder "Siemens Trio/"
Select series ('q' to abort):
  0 -    9 MR images 10:20:52 localizer_sf (*fl2d1) [1]
  1 -   21 MR images 10:24:39 diff 20DW 2NEX ALL (*ep_b1000#17) [2]
  2 -   54 MR images 10:24:40 diff 20DW 2NEX ALL_ADC (*ep_b0_1000) [3]
  3 -   54 MR images 10:24:40 diff 20DW 2NEX ALL_TRACEW (*ep_b1000t) [4]
  4 -   54 MR images 10:24:40 diff 20DW 2NEX ALL_FA (*ep_b0_1000) [5]
  5 -   21 MR images 10:28:52 ep2d_diff_MDDW_AT_WIP (ep_b1000#2) [7]
  6 -   54 MR images 10:28:53 ep2d_diff_MDDW_AT_WIP_ADC (ep_b0_1000) [8]
  7 -   54 MR images 10:28:53 ep2d_diff_MDDW_AT_WIP_TRACEW (ep_b1000t) [9]
  8 -   54 MR images 10:28:54 ep2d_diff_MDDW_AT_WIP_FA (ep_b0_1000) [10]
  9 -   21 MR images 10:35:01 diff 20DW 2NEX AT TE110 (*ep_b1000#1) [12]
 10 -   54 MR images 10:35:01 diff 20DW 2NEX AT TE110_ADC (*ep_b0_1000) [13]
 11 -   54 MR images 10:35:02 diff 20DW 2NEX AT TE110_TRACEW (*ep_b1000t) [14]
 12 -   54 MR images 10:35:02 diff 20DW 2NEX AT TE110_FA (*ep_b0_1000) [15]
 13 -   21 MR images 10:39:39 diff 20DW 2NEX AT TE80 (*ep_b1000#4) [17]
 14 -   54 MR images 10:39:51 diff 20DW 2NEX AT TE80_ADC (*ep_b0_1000) [18]
 15 -   54 MR images 10:39:51 diff 20DW 2NEX AT TE80_TRACEW (*ep_b1000t) [19]
 16 -   54 MR images 10:39:51 diff 20DW 2NEX AT TE80_FA (*ep_b0_1000) [20]
? 4,12,16
mrinfo: [100%] reading DICOM series "diff 20DW 2NEX ALL_FA"
mrinfo: [100%] reading DICOM series "diff 20DW 2NEX AT TE110_FA"
mrinfo: [100%] reading DICOM series "diff 20DW 2NEX AT TE80_FA"
*****
Image:                "TOURNIER DONALD (1) [MR] diff 20DW 2NEX ALL FA"
*****
Dimensions:          84 x 84 x 54 x 3
Voxel size:           2.5 x 2.5 x 2.5 x ?
Data strides:         [ -1 -2 3 4 ]
Format:               DICOM
Data type:             unsigned 16 bit integer (little endian)
Intensity scaling:    offset = 0, multiplier = 1
Transform:             0.9986   4.186e-08   -0.05229   -99.76
                     -0.002193   0.9991   -0.04188   -83.83
                     0.05224   0.04193   0.9978   -43.07
EchoTime:              0.08
PhaseEncodingDirection: j-
TotalReadoutTime:     0.0249
comments:              TOURNIER DONALD (1) [MR] diff 20DW 2NEX ALL FA
                      study: Head Brain_basic
                      DOB: 09/03/1977
                      DOS: 11/05/2007 10:24:40
```

In the above example, the application accessed 3 FA maps produced with different echo times as a single 4D dataset, consisting of DICOM series 4, 12, 16.

Loading DICOM data from scripts

It is good practice to write scripts to perform the full analysis from the raw data, so that the analysis can be performed afresh if required, and so that the exact steps taken at every stage of the analysis are recorded. However, access to

DICOM data requires user interaction to select the right series for each subject. There are two ways this can be done within *MRtrix3*:

1. Piping known selections to the command

The first time the data are inspected, the relevant selections can be recorded and piped as input to the commands, allowing their use in scripts. For example, assuming we have a data folder containing lots of data, and we are interested in Donald's T1 scan:

```
$ mrinfo DICOM_folder/
mrinfo: [done] scanning DICOM folder "DICOM_folder/"
Select patient (q to abort):
  1 - WILLATS LISA (000188) 06/04/1981
  2 - TOURNIER DONALD (BRI) 09/03/1977
? 2
patient: TOURNIER DONALD (BRI) 09/03/1977
Select series ('q' to abort):
  0 - 9 MR images 15:31:22 localiser (*f12d1) [1]
  1 - 160 MR images 15:37:34 t1_mpr_1mm iso qk (*tfl3d1_ns) [2]
  2 - 60 MR images 15:38:33 AX MPR T1 (*tfl3d1_ns) [3]
  3 - 60 MR images 15:38:56 COR MPR T1 (*tfl3d1_ns) [4]
  4 - 51 MR images 15:39:28 SAG MPR T1 (*tfl3d1_ns) [5]
  5 - 8 MR images 15:46:57 svb_se_30 PWM NWS (*tfl3d1_ns) [6]
  6 - 8 MR images 15:52:32 svb_se_30 PWM WS 32 ACQ (*tfl3d1_ns) [7]
  7 - 167 MR images 15:58:40 diff60_b3000_2.3_iPat2+ADC (*ep_b3000#93) [8]
  8 - 54 MR images 16:16:50 diff60_b3000_2.3_iPat2+ADC_ADC (*ep_b0_3000) [9]
  9 - 108 MR images 16:16:50 diff60_b3000_2.3_iPat2+ADC_TRACEW (*ep_b3000t) [10]
 10 - 54 MR images 16:16:51 diff60_b3000_2.3_iPat2+ADC_FA (*ep_b0_3000) [11]
? 1
mrinfo: [100%] reading DICOM series "t1_mpr_1mm iso qk"
*****
Image: "TOURNIER DONALD (BRI) [MR] t1_mpr_1mm iso qk"
*****
Dimensions: 160 x 256 x 256
Voxel size: 1 x 1 x 1
Data strides: [ 3 -1 -2 ]
Format: DICOM
Data type: unsigned 16 bit integer (little endian)
Intensity scaling: offset = 0, multiplier = 1
Transform: 0.9987 0.05056 0.003483 -85.68
           -0.05056 0.9987 -0.0001763 -106.9
           -0.003487 9.906e-09 1 -130.2
EchoTime: 0.00255
PhaseEncodingDirection: j-
TotalReadoutTime: 0
comments: TOURNIER DONALD (BRI) [MR] t1_mpr_1mm iso qk
          study: BRI_Temp_backup Donald
          DOB: 09/03/1977
          DOS: 03/10/2007 15:37:34
```

We can see that the relevant series is obtained using the choices 2 (to get the second patient) and 1 (to get the second series for that patient). This can be scripted using the `echo` command to *pipe* these numbers directly to the relevant command, with no further user interaction required, for example:

```
$ echo "2 1" | mrconvert DICOM_folder/ T1_anat.nii
mrconvert: [done] scanning DICOM folder "DICOM_folder/"
```

(continues on next page)

(continued from previous page)

```

Select patient (q to abort):
  1 - WILLATS LISA (000188) 06/04/1981
  2 - TOURNIER DONALD (BRI) 09/03/1977
? patient: TOURNIER DONALD (BRI) 09/03/1977
Select series ('q' to abort):
  0 -      9 MR images 15:31:22 localiser (*fl2d1) [1]
  1 -    160 MR images 15:37:34 t1_mpr_lmm iso qk (*tfl3d1_ns) [2]
  2 -     60 MR images 15:38:33 AX MPR T1 (*tfl3d1_ns) [3]
  3 -     60 MR images 15:38:56 COR MPR T1 (*tfl3d1_ns) [4]
  4 -     51 MR images 15:39:28 SAG MPR T1 (*tfl3d1_ns) [5]
  5 -      8 MR images 15:46:57 svs_se_30 PWM NWS (*tfl3d1_ns) [6]
  6 -      8 MR images 15:52:32 svs_se_30 PWM WS 32 ACQ (*tfl3d1_ns) [7]
  7 -   167 MR images 15:58:40 diff60_b3000_2.3_iPat2+ADC (*ep_b3000#93) [8]
  8 -    54 MR images 16:16:50 diff60_b3000_2.3_iPat2+ADC_ADC (*ep_b0_3000) [9]
  9 -   108 MR images 16:16:50 diff60_b3000_2.3_iPat2+ADC_TRACEW (*ep_b3000t) [10]
 10 -    54 MR images 16:16:51 diff60_b3000_2.3_iPat2+ADC_FA (*ep_b0_3000) [11]
mrconvert: [100%] reading DICOM series "t1_mpr_lmm iso qk"
mrconvert: [100%] copying from "TOURNIER D...BRI) [MR] t1_mpr_lmm iso qk" to "T1_anat.
↪nii"

```

2. Using environment variables

It is also possible to select datasets by string matching on specific DICOM fields. There are currently four selectors available:

- DICOM_PATIENT to match PatientName
- DICOM_ID to match PatientID
- DICOM_STUDY to match StudyName
- DICOM_SERIES to match SeriesName

To use a selector, simply set the corresponding environment variable to the desired pattern. It is simplest in practice to set it directly before the command, on the same line – this ensures that the environment variable does not apply to other commands than that intended. The matching pattern can include the `*` and/or `?` wildcard characters, and you can use several selectors at the same time. Note that matching is *not* case-sensitive. If you set any selector, you won't be prompted for input, but the command will abort if there is no match or if there are too many matches.

Example usage:

```
$ DICOM_SERIES='diff*iPat2' DICOM_PATIENT='*donald*' mrinfo dicom/
```

5.35.2 When the DICOM import goes wrong

Errors can occur in the DICOM import for several reasons. In some cases, we can identify the problem in the *MRtrix3* code and provide a fix to handle these data. In other cases, the data are simply not complete, not standards-compliant, or stored using encodings that *MRtrix3* doesn't currently handle.

The application crashes

If running a simple command such as:

```
$ mrconvert DICOM/ out.nii
mrconvert: [SYSTEM FATAL CODE: SIGSEGV (11)] Segmentation fault: Invalid memory access
```

crashes without a relevant error message, then this is an overt bug that needs fixing within the *MRtrix3* code. Even if the data are not DICOM-compliant, the code should nonetheless be able to detect this and exit gracefully with a clear indication of what the problem is. In these cases, please send the problematic data sets to members of the *MRtrix3* team for inspection.

ERROR: missing image frames for DICOM image

Note: This also applies to the “dimensions mismatch in DICOM series” error message.

DICOM data are often stored with individual slices in separate files. Unfortunately, there is no requirement in the DICOM standard that the files for a given dataset should all reside within the same folder. This means that it’s not uncommon for files belonging to the same series to be spread over multiple different sub-folders. This makes it all too easy for some of the images in a DICOM series to go missing, due to users forgetting to copy over all of the folders. Another way this can happen is when users copy the data from their DICOM client (e.g. PACS system) before the DICOM sender has finished sending the data (these transfers can take a long time...). Attempts to read the data will fail with a message like this:

```
$ mrinfo DICOM/
mrinfo: [done] scanning DICOM folder "DICOM/"
mrinfo: [100%] reading DICOM series "DWI_60"
mrinfo: [ERROR] missing image frames for DICOM image "Joe Bloggs [MR] DWI_60"
mrinfo: [ERROR] error opening image "DICOM/"
```

In these cases, it is simply not possible to load the data, since there are missing frames within it. The only solution here is to go back to the data source, find the missing data, and try again.

ERROR: no diffusion encoding information found in image

This indicates that *MRtrix3* was unable to find any information regarding the DW gradient directions (bvecs/bvals) in the DICOM headers, leading to errors like:

```
$ dwi2tensor DICOM/ dt.mif
dwi2tensor: [done] scanning DICOM folder "DICOM/"
dwi2tensor: [100%] reading DICOM series "DWI_60"
dwi2tensor: [ERROR] no diffusion encoding information found in image "Joe Bloggs [MR]
↳DWI_60"
```

This can happen for a number of reasons:

- the information is simply not present. This can happen with custom sequences not explicitly designed to provide this information, or lack of support for providing this information from some manufacturers. The only possible solution in this case is to obtain the DW information from a different source, and provide it to *MRtrix3* manually using the `-grad` or `-fslgrad` options in those commands that support it.
- the information is present, but in a format that *MRtrix3* doesn’t yet support. This is a very rare occurrence these days, but still possible. If you’re convinced your data should contain this information, please get in touch with members of the *MRtrix3* team, so we can take a look and see if support can be added to the code.
- the information *was* present, but has been stripped out by third-party software, in particular anonymisation packages. These typically work by stripping out all potentially patient-identifiable information. This often

includes removing any *private* (vendor-specific) DICOM entries, since it's not possible for a computer program to guarantee that these entries contain no sensitive information. Unfortunately, these entries often do contain important information, notably the DW gradient information. In these cases, the only sensible solution is to request the raw non-anonymised data, convert these correctly, and anonymise the *converted* images.

- the image series is simply *not* a DWI series. Double-check that you in fact trying to read the correct dataset.

ERROR: unsupported transfer syntax

The DICOM standard specifies a default *transfer syntax* to encode the information and the imaging data themselves. However, it also allows specifies a number of other storage formats to store the imaging data, notably compressed formats such as different variants of **JPEG** and **MPEG** (see the official [DICOM standard](#) for details). Importantly, these compressed formats are not mandatory: a compliant DICOM implementation does not need to support these features. This makes it entirely possible (and indeed, quite common) for a fully DICOM-compliant implementation to produce data that cannot be understood by another fully DICOM-compliant implementation – a less than ideal situation...

MRtrix3 does not currently support non-default transfer syntaxes – only those that the standard defines as mandatory, and variants thereof. For reference, these are:

- Implicit VR Little Endian (1.2.840.10008.1.2)
- Explicit VR Little Endian (1.2.840.10008.1.2.1)
- Explicit VR Big Endian (1.2.840.10008.1.2.2)

Any other transfer syntax will be flagged as unsupported, and *MRtrix3* will be unable to read the data, providing an error message similar to this:

```
$ mrinfo DICOM
mrinfo: [done] scanning DICOM folder "DICOM"
mrinfo: [ERROR] unable to read DICOM images in "DICOM":
mrinfo: [ERROR] unsupported transfer syntax found in DICOM data
mrinfo: [ERROR] consider using third-party tools to convert your data to standard_
↳uncompressed encoding
mrinfo: [ERROR] See the MRtrix3 documentation on DICOM handling for details:
mrinfo: [ERROR] http://mrtrix.readthedocs.io/en/latest/tips_and_tricks/dicom_
↳handling.html#error-unsupported-transfer-syntax
mrinfo: [ERROR] error opening image "DICOM"
```

Thankfully, other tools exist that should be able to convert the data to a format that *MRtrix3* (and other DICOM tools) will read. The **dcmtk** DICOM toolkit in particular provides the **dcmdjpeg** command to decompress data stored using JPEG transfer syntax. On Linux, a directory of such files can be decompressed as follows (amend the various **PATH** as required for your system):

```
$ export PATH=/opt/dcmtk/bin:$PATH
$ export DCMDICTPATH=/opt/dcmtk/share/dcmtk/dicom.dic

$ for img in dcmdir/*
> do
>     dcmdjpeg $img ${img}.tmp
>     mv ${img}.tmp $img
> done
```

MRtrix3 commands should now be able to read the directory successfully:

```
$ mrinfo dcmdir
mrinfo: [done] scanning DICOM folder "data/driss/t1"
```

(continues on next page)

(continued from previous page)

```
mrinfo: [100%] reading DICOM series "AX FSPGR 3D ASSET C+"
...
```

5.36 Batch processing with `for_each`

Image processing often involves executing the same command on many different subjects or time points within a study. *MRtrix3* includes a Python script called `for_each` to simplify this process. The main benefit of using `for_each` compared to a bash `for` loop is a simpler and less verbose syntax. However other benefits include multi-threaded job execution (to exploit modern multi-core CPUs when the command being run is not already multi-threaded), and automatic identification of path basenames and prefixes. To view the full help page run `for_each` on the command line with no arguments.

5.36.1 Example 1 - using `IN`

Many people like to organise their imaging datasets with one directory per subject. For example:

```
study/001_patient/dwi.mif
study/002_patient/dwi.mif
study/003_patient/dwi.mif
study/004_control/dwi.mif
study/005_control/dwi.mif
study/006_control/dwi.mif
```

The `for_each` script can be used to run the same command on each subject, for example:

```
.. code-block:: console
```

```
$ for_each study/* : dwidenoise IN/dwi.mif IN/dwi_denoised.mif
```

The first part of the command above is the `for_each` script name, followed by the pattern matching string (`study/*`) to identify all the files (which in this case are directories) to be looped over. The colon is used to separate the invocation of `for_each`, along with its inputs and any command-line options, from the command to be executed. In this example the `dwidenoise` command will be run multiple times, by substituting the keyword `IN` with each of the directories that match the pattern (`study/001_patient`, `study/002_patient`, etc.).

5.36.2 Example 2 - using `NAME`

Other people may prefer to organise their imaging datasets with one folder per image type and have all subjects inside. For example:

```
study/dwi/001_patient.mif
study/dwi/002_patient.mif
study/dwi/003_patient.mif
study/dwi/004_control.mif
study/dwi/005_control.mif
study/dwi/006_control.mif
```

The `NAME` keyword can be used in this situation to obtain the basename of the file path. For example:

```
$ mkdir study/dwi_denoised
$ for_each study/dwi/* : dwidenoise IN study/dwi_denoised/NAME
```

Here, the IN keyword will be substituted with the full string from the matching pattern (`study/dwi/001_patient.mif`, `study/dwi/002_patient.mif`, etc), however the NAME keyword will be replaced with the *basename* of the matching pattern (`001_patient.mif`, `002_patient.mif`, etc).

Alternatively, the same result can be achieved by running `for_each` from inside the `study/dwi` directory. In this case NAME would not be required. For example:

```
$ mkdir study/dwi_denoised
$ cd study/dwi
$ for_each * : dwidenoise IN ../dwi_denoised/IN
```

5.36.3 Example 3 - using PRE

For this example let us assume we want to convert all `dwi.mif` files from example 2 to NIfTI file format (`*.nii`). This can be performed using:

```
$ for_each study/dwi/* : mrconvert IN study/dwi/PRE.nii
$ rm *.mif
```

There the PRE keyword will be replaced by the file basename, without the file extension.

5.36.4 Example 4 - Sequential Processing

As an example of a single `for_each` command running multiple sequential commands (e.g. with the bash `;`, `|`, `&&`, `||` operators), let's assume in the previous example we wanted to remove the `*.mif` files as they were converted. We could use the `&&` operator, which means “run next command only if current command succeeds without error”.

```
$ for_each study/dwi/* : mrconvert IN study/dwi/PRE.nii "&&" rm IN
```

The `&&` operator here must be escaped with quotes in order to prevent the shell from interpreting it. Bash operator characters can also be escaped with the `\"` character; for example, to *pipe an image* between two MRtrix commands (assuming the data set directory layout from example 1):

```
$ for_each study/* : dwiextract -bzero IN/dwi.mif - \| mrmath - mean -axis 3 IN/mean_
↪b0.mif
```

5.36.5 Example 5 - Parallel Processing

To run multiple jobs at once, use the standard *MRtrix3* command-line option `-nthreads N`, where N is the number of concurrent jobs required. For example:

```
$ for_each study/* -nthreads 8 : dwidenoise IN/dwi.mif IN/dwi_denoised.mif
```

will run up to 8 of the required jobs in parallel. Note that unlike in other *MRtrix3* commands where command-line options can be placed anywhere on the command-line, in this particular context the `-nthreads` option must be specified *before* the colon separator. This is necessary in order for the `for_each` script to recognise that this command-line option applies to its own operation, as opposed to the command that `for_each` is responsible for invoking. To demonstrate this, consider the following usage:

```
$ for_each study/* : dwidenoise IN/dwi.mif IN/dwi_denoised.mif -nthreads 8
```

Here, `for_each` would execute the `dwidenoise` command entirely *sequentially*, once for each input; but each time it is run, `dwidenoise` would be instructed to use 8 threads.

Indeed these two usages can in theory be *combined*. Imagine that a hypothetical *MRtrix3* command, “`dwidostuff`”, tends to not be capable in practise of utilising any more than four threads, regardless of how many threads are in fact available on your hardware / explicitly invoked. However you have a system with eight hardware threads, and wish to utilise them all as much as possible. In such a scenario, you could use:

```
$ for_each study/* -nthreads 2 : dwidostuff IN/dwi.mif IN/dwi_stuffdone.mif -nthreads_
↪ 4
```

This would instruct `for_each` to always have *two* jobs running in parallel, each of which will be explicitly instructed to use *four* threads.

Note that most *MRtrix3* commands are multi-threaded, and will generally succeed in individually using all available CPU cores, in which case running multiple jobs in parallel using `for_each` is unlikely to provide a benefit in computation time (or it may in fact be detrimental). If however a particular command is known to be single-threaded (or have only limited multi-threading capability), and your system possesses enough RAM to support running multiple instances of that command at once, this usage may yield a considerable reduction in total processing time.

5.37 External modules

The *MRtrix3* build process allows for the easy development of separate modules, compiled against the *MRtrix3* core (or indeed against any other *MRtrix3* module). This allows developers to maintain their own repository, or compile stand-alone commands provided by developers / other users, without affecting their core *MRtrix3* installation. The obvious benefit is that developers can keep their own developments private if they wish to, and the *MRtrix3* core can be kept as lean as possible.

5.37.1 Filesystem structure

A module simply consists of a separate directory, which contains its own `cmd/` folder, and potentially also its own `src/` folder if required. The build process is then almost identical to that for the *MRtrix3* core, with a few differences.

To demonstrate this, we construct a module residing alongside the core installation:

```
$ mkdir ~/mymodule
$ cd ~/mymodule/
$ mkdir cmd
```

Assuming our module consists of a single `mycommand.cpp` C++ file, it should be placed in the `cmd/` folder (see below for details). This results in the following folder structure:

```
mymodule/
|-- cmd/
|   |-- mycommand.cpp
```

The most relevant difference is how the build script is invoked. For a module, compilation is started by invoking the *MRtrix3* core’s build script, but with the module’s top-level folder being the current working directory. For example, if the *MRtrix3* core resides in the directory `~/mrtrix3/`, and the module resides in `~/mymodule`, then the module can be compiled by typing:

```
$ cd ~/mymodule
$ ../mrtrix3/build
```

This will compile all `.cpp` files found in the `cmd/` folder, along with any potentially dependencies in the `src/` folder (if applicable), and place the resulting executables in the `bin/` folder (which will be created if not already present).

Note: Once compiled, the newly created executables will only run if they remain in the same location relative to the *MRtrix3* core folder. This is because the runtime search path for the *MRtrix3* dynamic library is set within each executable to search in the core *MRtrix3* `lib/` folder (for our example, this is `../../mrtrix3/lib/`). If you need to move your *MRtrix3* and module installations, make sure to maintain their relative paths.

Linking to the MRtrix3 core (C++ code only)

For routine use of modules containing C++ code, it is more convenient to set up a reference to the core *MRtrix3* installation. This can be done in two ways:

- **Using a symbolic link** to the *MRtrix3* core's build script:

```
$ cd ~/mymodule
$ ln -s ../mrtrix3/build
```

This results in the following folder structure:

```
mymodule/
|-- bin/
|-- build -> ../mrtrix3/build
|-- cmd/
|   |-- mycommand.cpp
```

The link can then be invoked directly, and the build script will detect that it is compiling a module:

```
$ ./build
```

Note that this approach will NOT work on Windows / MSYS2 installations, due to the lack of support for symbolic links. In this case, use the alternative approach below.

- **Using a text file** containing the path to the *MRtrix3* core's build script:

```
$ cd ~/mymodule
$ echo ../mrtrix3/build > build
$ chmod +x build
```

The last command ensures that the file is executable. This results in the following folder structure:

```
mymodule/
|-- bin/
|-- build
|-- cmd/
|   |-- mycommand.cpp
```

The new executable file can then be invoked directly, and the build script will again detect that it is compiling a module:

```
$ ./build
```

5.37.2 Handling Python commands

The instructions above relate to modules containing C++ code. Modules can also contain Python scripts, in which case additional steps will be required to ensure the module's scripts use the core *MRtrix3* python libraries. There are several ways to do this, depending on your circumstances. In all cases, the aim is to ensure that the correct `mrtrix3.py` module can be located and imported (i.e. that the `import mrtrix3` line succeeds).

- **Symbolic link to the MRtrix3 core `mrtrix3.py` (recommended):** the core installation contains a `bin/mrtrix3.py` file that will be imported preferentially for any script co-located within the same `bin/` folder. It will in turn locate and import the actual `mrtrix3` module, which is located in the core `lib/mrtrix3` folder. If a symbolic link to that file is placed in the module's `bin/` folder, it will locate the correct modules. For example:

```
$ cd ~/mymodule/
$ ln -sr ../mrtrix3/bin/mrtrix3.py bin/
```

This results in the following folder structure:

```
mymodule/
|-- bin/
|   |-- mrtrix3.py -> ../../mrtrix3/bin/mrtrix3.py
|-- build -> ../mrtrix3/build
|-- cmd/
|   |-- mycommand.cpp
```

- **Copy of the MRtrix3 core `mrtrix3.py` file:** in some cases, it may not be possible or convenient to use a symbolic link as described above. This is the case particularly on Windows / MSYS2 installations, or when distributing an independent module. In this case, a *copy* of the core *MRtrix3* `bin/mrtrix3.py` can be placed in the module's `bin/` folder:

```
$ cd ~/mymodule/
$ cp ../mrtrix3/bin/mrtrix3.py bin/
```

This results in the following folder structure:

```
mymodule/
|-- bin/
|   |-- mrtrix3.py
|-- build -> ../mrtrix3/build
|-- cmd/
|   |-- mycommand.cpp
```

In this case, the script will fail to detect the *MRtrix3* modules in the normal way, and will instead rely on the `build` symbolic link or file to locate the core libraries. For this to work, the module must therefore have been set up as suggested in the previous section: either with a `build` symbolic link pointing the core *MRtrix3* build script, or with a `build` file containing the path to the core build script. The location of the core *MRtrix3* build script is then sufficient to locate the core Python libraries, since they should reside in a known location relative to that script.

- **Use the `PYTHONPATH` environment variable:** some users may prefer to set the `PYTHONPATH` environment variable to point to the core *MRtrix3* `lib/` folder. This is the more usual way of locating modules in Python, and will work here also:

```
$ export PYTHONPATH=~/mrtrix3/lib
```

Note: While the `PYTHONPATH` environment variable will work, there are good reasons not to use this approach. If you have multiple versions of *MRtrix3* installed on one system, and use this approach, then the Python modules

within whichever of those *MRtrix3* versions is added to PYTHONPATH will *always* be imported, regardless of the version of *MRtrix3* against which any particular external module is *intended* to run. Creation of a `bin/mrtrix3.py` symbolic link or copy is therefore preferable, as it allows different external modules to run against different *MRtrix3* installations.

5.37.3 Adding code to the module

New code can be added to this new module as follows:

- **Stand-alone .cpp file:** a single C++ code file destined to be compiled into a binary executable should have the `.cpp` file extension, and be placed into the `cmd/` directory of the module. Execution of the `build` script in the module root directory should then detect the presence of this file, and generate an executable file in the corresponding `bin/` directory.
- **Stand-alone Python file:** A stand-alone Python script designed to make use of the *MRtrix3* Python APIs will typically not have any file extension, and will have its first line set to `#!/usr/bin/env python`. Such files should be placed directly into the `bin/` directory. It will also typically be necessary to mark the file as executable before the system will allow it to run:

```
$ chmod +x bin/example_script
```

(Replace `example_script` with the name of the script file you have added)

- **More complex modules:** If the requisite code for a particular functionality cannot reasonably be fully encapsulated within a single file, additional files will need to be added to the module. For C++ code, these will need to be added to the `src/` directory. For further details, refer to the relevant [developer documentation](#).

For example, the following steps take the `example_script` Python script and `example_binary.cpp` C++ files, previously downloaded by the user into the `~/Downloads/` folder, place them in the appropriate locations in the module created as described above, ensure the Python script is executable, and build the C++ executable:

```
$ cd ~/mymodule
$ cp ~/Downloads/example_script bin/
$ cp ~/Downloads/example_binary.cpp cmd/
$ chmod +x bin/example_script
$ ./build
[1/2] [CC] tmp/cmd/example_binary.o
[2/2] [LD] bin/example_binary
```

This results in the following folder structure:

```
mymodule/
|-- bin/
|   |-- example_binary
|   |-- example_script
|   |-- mrtrix3.py -> ../../mrtrix3/bin/mrtrix3.py
|-- build -> ../mrtrix3/build
|-- cmd/
|   |-- example_binary.cpp
|-- tmp/
|   |-- (directories)
```

Both example command executables – `example_binary` and `example_script` – now reside in directory `~/mymodule/bin/`. The `example_binary` executable will be linked against the core *MRtrix3* library (in the `~/mrtrix3/lib` folder), and the `example_script` Python script will import modules from the core *MRtrix3* Python module (in the `~/mrtrix3/lib/mrtrix3` folder) – neither will run if these libraries are not found.

5.37.4 Adding modules to PATH

Because these binaries are not placed into the same directory as those provided as part of the core *MRtrix3* installation, simply typing the name of the command into the terminal will not work, as your system will not yet be configured to look for executable files in this new location. You can solve this in one of three ways:

1. Provide the *full path* to the binary file when executing it. So for instance, instead of typing:

```
$ example_binary argument1 argument2 ...
```

you would use:

```
$ ~/mymodule/bin/example_binary argument1 argument2 ...
```

While this may be inconvenient in some circumstances, in others it can be beneficial, as it is entirely explicit and clear as to exactly which version of the command is being run. This is especially useful when experimenting with different versions of a command, where the name of the command has not changed.

2. Use the `set_path` script provided with *MRtrix3* to automatically add the location of the module's `bin/` directory to `PATH` whenever a terminal session is created. To do this, execute your core *MRtrix3* installation's `set_path` script while residing in the top-level directory of the module:

```
$ cd ~/mymodule
$ ../mrtrix3/set_path
```

3. Manually add the location of the `bin/` directory of this new module to your system's `PATH` environment variable. Most likely you will want this location to be already stored within `PATH` whenever you open a new terminal; therefore you will most likely want to add a line such as that below to the appropriate configuration file for your system (e.g. `~/.bashrc` or `~/.bash_profile`; the appropriate file will depend on your particular system):

```
$ export PATH=~/.mymodule/bin:$PATH
```

Obviously you will need to modify this line according to the location on your file system where you have installed the module.

5.38 Troubleshooting

As of the initial *MRtrix3* release (version 3.0.0), the various “Troubleshooting” pages that once formed a section of the software documentation are now maintained within a [distinct Wiki category](#) on the **MRtrix3** [community forum](#). This will allow the developers to more quickly deploy updates to documentation and recommendations, as well as allow users of the software to contribute modifications or new pages themselves.

Please [visit the Wiki](#) and search for an existing solution to any issues you face, and/or contribute to the discussion!

5.39 List of MRtrix3 commands

5.39.1 5tt2gmwmi

Synopsis

Generate a mask image appropriate for seeding streamlines on the grey matter-white matter interface

Usage

```
5tt2gmwmi [ options ] 5tt_in mask_out
```

- *5tt_in*: the input 5TT segmented anatomical image
- *mask_out*: the output mask image

Options

- **-mask_in image** Filter an input mask image according to those voxels that lie upon the grey matter - white matter boundary. If no input mask is provided, the output will be a whole-brain mask image calculated using the anatomical image only.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 2012, 62, 1924-1938

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.2 5tt2vis

Synopsis

Generate an image for visualisation purposes from an ACT 5TT segmented anatomical image

Usage

```
5tt2vis [ options ] input output
```

- *input*: the input 4D tissue-segmented image
- *output*: the output 3D image for visualisation

Options

- **-bg value** image intensity of background (default: 0)
- **-cgm value** image intensity of cortical grey matter (default: 0.5)
- **-sgm value** image intensity of sub-cortical grey matter (default: 0.75)
- **-wm value** image intensity of white matter (default: 1)
- **-csf value** image intensity of CSF (default: 0.15)
- **-path value** image intensity of pathological tissue (default: 2)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.3 5ttcheck

Synopsis

Thoroughly check that one or more images conform to the expected ACT five-tissue-type (5TT) format

Usage

```
5ttcheck [ options ] input [ input ... ]
```

- *input*: the 5TT image(s) to be tested

Options

- **-voxels prefix** output mask images highlighting voxels where the input does not conform to 5TT requirements

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.4 5ttedit

Synopsis

Manually set the partial volume fractions in an ACT five-tissue-type (5TT) image using mask images

Usage

```
5ttedit [ options ] input output
```

- *input*: the 5TT image to be modified
- *output*: the output modified 5TT image

Options

- **-cgm image** provide a mask of voxels that should be set to cortical grey matter
- **-sgm image** provide a mask of voxels that should be set to sub-cortical grey matter
- **-wm image** provide a mask of voxels that should be set to white matter
- **-csf image** provide a mask of voxels that should be set to CSF
- **-path image** provide a mask of voxels that should be set to pathological tissue
- **-none image** provide a mask of voxels that should be cleared (i.e. are non-brain); note that this will supersede all other provided masks

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.

- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.5 5ttgen

Synopsis

Generate a 5TT image suitable for ACT

Usage

```
5ttgen algorithm [ options ] ...
```

- *algorithm*: Select the algorithm to be used to complete the script operation; additional details and options become available once an algorithm is nominated. Options are: freesurfer, fsl, gif, hsvs

Description

5ttgen acts as a ‘master’ script for generating a five-tissue-type (5TT) segmented tissue image suitable for use in Anatomically-Constrained Tractography (ACT). A range of different algorithms are available for completing this task. When using this script, the name of the algorithm to be used must appear as the first argument on the command-line after ‘5ttgen’. The subsequent compulsory arguments and options available depend on the particular algorithm being invoked.

Each algorithm available also has its own help page, including necessary references; e.g. to see the help page of the ‘fsl’ algorithm, type ‘5ttgen fsl’.

Options

Options common to all 5ttgen algorithms

- **-nocrop** Do NOT crop the resulting 5TT image to reduce its size (keep the same dimensions as the input image)
- **-sgm_amyg_hipp** Represent the amygdalae and hippocampi as sub-cortical grey matter in the 5TT image

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 2012, 62, 1924-1938

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.6 5ttgen freesurfer

Synopsis

Generate the 5TT image based on a FreeSurfer parcellation image

Usage

```
5ttgen freesurfer input output [ options ]
```

- *input*: The input FreeSurfer parcellation image (any image containing ‘aseg’ in its name)
- *output*: The output 5TT image

Options

Options specific to the ‘freesurfer’ algorithm

- **-lut** Manually provide path to the lookup table on which the input parcellation image is based (e.g. FreeSurfer-ColorLUT.txt)

Options common to all 5ttgen algorithms

- **-nocrop** Do NOT crop the resulting 5TT image to reduce its size (keep the same dimensions as the input image)
- **-sgm_amyg_hipp** Represent the amygdalae and hippocampi as sub-cortical grey matter in the 5TT image

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.

- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 2012, 62, 1924-1938

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.7 5ttgen fsl

Synopsis

Use FSL commands to generate the 5TT image based on a T1-weighted image

Usage

```
5ttgen fsl input output [ options ]
```

- *input*: The input T1-weighted image
- *output*: The output 5TT image

Options

Options specific to the ‘fsl’ algorithm

- **-t2 <T2 image>** Provide a T2-weighted image in addition to the default T1-weighted image; this will be used as a second input to FSL FAST

- **-mask** Manually provide a brain mask, rather than deriving one in the script
- **-premasked** Indicate that brain masking has already been applied to the input image

Options common to all 5ttgen algorithms

- **-nocrop** Do NOT crop the resulting 5TT image to reduce its size (keep the same dimensions as the input image)
- **-sgm_amyg_hipp** Represent the amygdalae and hippocampi as sub-cortical grey matter in the 5TT image

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 2012, 62, 1924-1938
- Smith, S. M. Fast robust automated brain extraction. *Human Brain Mapping*, 2002, 17, 143-155
- Zhang, Y.; Brady, M. & Smith, S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 2001, 20, 45-57
- Patenaude, B.; Smith, S. M.; Kennedy, D. N. & Jenkinson, M. A Bayesian model of shape and appearance for subcortical brain segmentation. *NeuroImage*, 2011, 56, 907-922
- Smith, S. M.; Jenkinson, M.; Woolrich, M. W.; Beckmann, C. F.; Behrens, T. E.; Johansen-Berg, H.; Bannister, P. R.; De Luca, M.; Drobnjak, I.; Flitney, D. E.; Niazy, R. K.; Saunders, J.; Vickers, J.; Zhang, Y.; De Stefano, N.; Brady, J. M. & Matthews, P. M. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 2004, 23, S208-S219

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.8 5ttgen gif

Synopsis

Generate the 5TT image based on a Geodesic Information Flow (GIF) segmentation image

Usage

```
5ttgen gif input output [ options ]
```

- *input*: The input Geodesic Information Flow (GIF) segmentation image
- *output*: The output 5TT image

Options

Options common to all 5ttgen algorithms

- **-nocrop** Do NOT crop the resulting 5TT image to reduce its size (keep the same dimensions as the input image)
- **-sgm_amyg_hipp** Represent the amygdalae and hippocampi as sub-cortical grey matter in the 5TT image

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. NeuroImage, 2012, 62, 1924-1938

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Matteo Mancini (m.mancini@ucl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.9 5ttgen hsvs

Synopsis

Generate a 5TT image based on Hybrid Surface and Volume Segmentation (HSVS), using FreeSurfer and FSL tools

Usage

```
5ttgen hsvs input output [ options ]
```

- *input*: The input FreeSurfer subject directory
- *output*: The output 5TT image

Options

- **-template** Provide an image that will form the template for the generated 5TT image
- **-hippocampi** Select method to be used for hippocampi (& amygdalae) segmentation; options are: sub-fields,first,aseg
- **-thalami** Select method to be used for thalamic segmentation; options are: nuclei,first,aseg
- **-white_stem** Classify the brainstem as white matter

Options common to all 5ttgen algorithms

- **-nocrop** Do NOT crop the resulting 5TT image to reduce its size (keep the same dimensions as the input image)
- **-sgm_amyg_hipp** Represent the amygdalae and hippocampi as sub-cortical grey matter in the 5TT image

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 2012, 62, 1924-1938
- Smith, R.; Skoch, A.; Bajada, C.; Caspers, S.; Connelly, A. Hybrid Surface-Volume Segmentation for improved Anatomically-Constrained Tractography. In *Proc OHBM 2020*
- Fischl, B. Freesurfer. *NeuroImage*, 2012, 62(2), 774-781

- If FreeSurfer hippocampal subfields module is utilised: Iglesias, J.E.; Augustinack, J.C.; Nguyen, K.; Player, C.M.; Player, A.; Wright, M.; Roy, N.; Frosch, M.P.; Mc Kee, A.C.; Wald, L.L.; Fischl, B.; and Van Leemput, K. A computational atlas of the hippocampal formation using ex vivo, ultra-high resolution MRI: Application to adaptive segmentation of in vivo MRI. *NeuroImage*, 2015, 115, 117-137
- If FreeSurfer hippocampal subfields module is utilised and includes amygdalae segmentation: Saygin, Z.M. & Kliemann, D.; Iglesias, J.E.; van der Kouwe, A.J.W.; Boyd, E.; Reuter, M.; Stevens, A.; Van Leemput, K.; Mc Kee, A.; Frosch, M.P.; Fischl, B.; Augustinack, J.C. High-resolution magnetic resonance imaging reveals nuclei of the human amygdala: manual segmentation to automatic atlas. *NeuroImage*, 2017, 155, 370-382
- If -thalami nuclei is used: Iglesias, J.E.; Insausti, R.; Lerma-Usabiaga, G.; Bocchetta, M.; Van Leemput, K.; Greve, D.N.; van der Kouwe, A.; ADNI; Fischl, B.; Caballero-Gaudes, C.; Paz-Alonso, P.M. A probabilistic atlas of the human thalamic nuclei combining ex vivo MRI and histology. *NeuroImage*, 2018, 183, 314-326
- If ACPCDetect is installed: Ardekani, B.; Bachman, A.H. Model-based automatic detection of the anterior and posterior commissures on MRI scans. *NeuroImage*, 2009, 46(3), 677-682

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.10 afdconnectivity

Synopsis

Obtain an estimate of fibre connectivity between two regions using AFD and streamlines tractography

Usage

```
afdconnectivity [ options ] image tracks
```

- *image*: the input FOD image.
- *tracks*: the input track file defining the bundle of interest.

Description

This estimate is obtained by determining a fibre volume (AFD) occupied by the pathway of interest, and dividing by the streamline length.

If only the streamlines belonging to the pathway of interest are provided, then ALL of the fibre volume within each voxel selected will contribute to the result. If the -wbft option is used to provide whole-brain fibre-tracking (of which

the pathway of interest should contain a subset), only the fraction of the fibre volume in each fixel estimated to belong to the pathway of interest will contribute to the result.

Use `-quiet` to suppress progress messages and output fibre connectivity value only.

For valid comparisons of AFD connectivity across scans, images **MUST** be intensity normalised and bias field corrected, and a common response function for all subjects must be used.

Note that the sum of the AFD is normalised by streamline length to account for subject differences in fibre bundle length. This normalisation results in a measure that is more related to the cross-sectional volume of the tract (and therefore ‘connectivity’). Note that SIFT-ed tract count is a superior measure because it is unaffected by tangential yet unrelated fibres. However, AFD connectivity may be used as a substitute when Anatomically Constrained Tractography is not possible due to uncorrectable EPI distortions, and SIFT may therefore not be as effective.

Options

- **-wbft tracks** provide a whole-brain fibre-tracking data set (of which the input track file should be a subset), to improve the estimate of fibre bundle volume in the presence of partial volume
- **-afd_map image** output a 3D image containing the AFD estimated for each voxel.
- **-all_fixels** if whole-brain fibre-tracking is NOT provided, then if multiple fixels within a voxel are traversed by the pathway of interest, by default the fixel with the greatest streamlines density is selected to contribute to the AFD in that voxel. If this option is provided, then ALL fixels with non-zero streamlines density will contribute to the result, even if multiple fixels per voxel are selected.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.11 amp2response

Synopsis

Estimate response function coefficients based on the DWI signal in single-fibre voxels

Usage

```
amp2response [ options ] amps mask directions response
```

- *amps*: the amplitudes image
- *mask*: the mask containing the voxels from which to estimate the response function
- *directions*: a 4D image containing the estimated fibre directions
- *response*: the output zonal spherical harmonic coefficients

Description

This command uses the image data from all selected single-fibre voxels concurrently, rather than simply averaging their individual spherical harmonic coefficients. It also ensures that the response function is non-negative, and monotonic (i.e. its amplitude must increase from the fibre direction out to the orthogonal plane).

If multi-shell data are provided, and one or more b-value shells are not explicitly requested, the command will generate a response function for every b-value shell (including b=0 if present).

Options

- **-isotropic** estimate an isotropic response function (lmax=0 for all shells)
- **-noconstraint** disable the non-negativity and monotonicity constraints
- **-directions path** provide an external text file containing the directions along which the amplitudes are sampled

DW shell selection options

- **-shells b-values** specify one or more b-values to use during processing, as a comma-separated list of the desired approximate b-values (b-values are clustered to allow for small deviations). Note that some commands are incompatible with multiple b-values, and will report an error if more than one b-value is provided. **WARNING:** note that, even though the b=0 volumes are never referred to as shells in the literature, they still have to be explicitly included in the list of b-values as provided to the -shell option! Several algorithms which include the b=0 volumes in their computations may otherwise return an undesired result.

- **-lmax values** specify the maximum harmonic degree of the response function to estimate (can be a comma-separated list for multi-shell data)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Smith, R. E.; Dhollander, T. & Connelly, A. Constrained linear least squares estimation of anisotropic response function for spherical deconvolution. ISMRM Workshop on Breaking the Barriers of Diffusion MRI, 23.

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.12 amp2sh

Synopsis

Convert a set of amplitudes (defined along a set of corresponding directions) to their spherical harmonic representation

Usage

```
amp2sh [ options ] amp SH
```

- *amp*: the input amplitude image.
- *SH*: the output spherical harmonics coefficients image.

Description

The spherical harmonic decomposition is calculated by least-squares linear fitting to the amplitude data.

The directions can be defined either as a DW gradient scheme (for example to compute the SH representation of the DW signal), a set of [az el] pairs as output by the `dirgen` command, or a set of [x y z] directions in Cartesian coordinates. The DW gradient scheme or direction set can be supplied within the input image header or using the `-gradient` or `-directions` option. Note that if a direction set and DW gradient scheme can be found, the direction set will be used by default.

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Options

- **-lmax order** set the maximum harmonic order for the output series. By default, the program will use the highest possible lmax given the number of diffusion-weighted images, up to a maximum of 8.
- **-normalise** normalise the DW signal to the b=0 image
- **-directions file** the directions corresponding to the input amplitude image used to sample AFD. By default this option is not required providing the direction set is supplied in the amplitude image. This should be supplied as a list of directions [az el], as generated using the `dirgen` command, or as a list of [x y z] Cartesian coordinates.
- **-rician noise** correct for Rician noise induced bias, using noise map supplied

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

DW shell selection options

- **-shells b-values** specify one or more b-values to use during processing, as a comma-separated list of the desired approximate b-values (b-values are clustered to allow for small deviations). Note that some commands are incompatible with multiple b-values, and will report an error if more than one b-value is provided. **WARNING:** note that, even though the b=0 volumes are never referred to as shells in the literature, they still have to be explicitly included in the list of b-values as provided to the `-shell` option! Several algorithms which include the b=0 volumes in their computations may otherwise return an undesired result.

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.13 connectome2tck

Synopsis

Extract streamlines from a tractogram based on their assignment to parcellated nodes

Usage

```
connectome2tck [ options ] tracks_in assignments_in prefix_out
```

- *tracks_in*: the input track file
- *assignments_in*: input text file containing the node assignments for each streamline
- *prefix_out*: the output file / prefix

Description

The compulsory input file “assignments_in” should contain a text file where there is one row for each streamline, and each row contains a list of numbers corresponding to the parcels to which that streamline was assigned (most typically there will be two entries per streamline, one for each endpoint; but this is not strictly a requirement). This file will most typically be generated using the tck2connectome command with the -out_assignments option.

Example usages

- *Default usage:*

```
$ connectome2tck tracks.tck assignments.txt edge-
```

The command will generate one track file for every edge in the connectome, with the name of each file indicating the nodes connected via that edge; for instance, all streamlines connecting nodes 23 and 49 will be written to file “edge-23-49.tck”.

- *Extract only the streamlines between nodes 1 and 2:*

```
$ connectome2tck tracks.tck assignments.txt tracks_1_2.tck -nodes 1,2 -exclusive -  
↪files single
```

Since only a single edge is of interest, this example provides only the two nodes involved in that edge to the -nodes option, adds the -exclusive option so that only streamlines for which both assigned nodes are in the list of nodes of interest are extracted (i.e. only streamlines connecting nodes 1 and 2 in this example), and writes the result to a single output track file.

- *Extract the streamlines connecting node 15 to all other nodes in the parcellation, with one track file for each edge:*

```
$ tck2connectome tracks.tck assignments.txt from_15_to_ -nodes 15 -keep_self
```

The command will generate the same number of track files as there are nodes in the parcellation: one each for the streamlines connecting node 15 to every other node; i.e. “from_15_to_1.tck”, “from_15_to_2.tck”, “from_15_to_3.tck”, etc.. Because the -keep_self option is specified, file “from_15_to_15.tck” will also be generated, containing those streamlines that connect to node 15 at both endpoints.

- *For every node, generate a file containing all streamlines connected to that node:*

```
$ connectome2tck tracks.tck assignments.txt node -files per_node
```

Here the command will generate one track file for every node in the connectome: “node1.tck”, “node2.tck”, “node3.tck”, etc.. Each of these files will contain all streamlines that connect the node of that index to another node in the connectome (it does not select all tracks connecting a particular node, since the -keep_self option was omitted and therefore e.g. a streamline that is assigned to node 41 will not be present in file “node41.tck”).

Each streamline in the input tractogram will in fact appear in two different output track files; e.g. a streamline connecting nodes 8 and 56 will be present both in file “node8.tck” and file “node56.tck”.

- *Get all streamlines that were not successfully assigned to a node pair:*

```
$ connectome2tck tracks.tck assignments.txt unassigned.tck -nodes 0 -keep_self -
↪files single
```

Node index 0 corresponds to streamline endpoints that were not successfully assigned to a node. As such, by selecting all streamlines that are assigned to “node 0” (including those streamlines for which neither endpoint is assigned to a node due to use of the `-keep_self` option), the single output track file will contain all streamlines for which at least one of the two endpoints was not successfully assigned to a node.

- *Generate a single track file containing edge exemplar trajectories:*

```
$ tck2connectome tracks.tck assignments.txt exemplars.tck -files single -
↪exemplars nodes.mif
```

This produces the track file that is required as input when attempting to display connectome edges using the streamlines or streamtubes geometries within the meview connectome tool.

Options

Options for determining the content / format of output files

- **-nodes list** only select tracks that involve a set of nodes of interest (provide as a comma-separated list of integers)
- **-exclusive** only select tracks that exclusively connect nodes from within the list of nodes of interest
- **-files option** select how the resulting streamlines will be grouped in output files. Options are: `per_edge`, `per_node`, `single` (default: `per_edge`)
- **-exemplars image** generate a mean connection exemplar per edge, rather than keeping all streamlines (the parcellation node image must be provided in order to constrain the exemplar endpoints)
- **-keep_unassigned** by default, the program discards those streamlines that are not successfully assigned to a node. Set this option to generate corresponding outputs containing these streamlines (labelled as node index 0)
- **-keep_self** by default, the program will not output streamlines that connect to the same node at both ends. Set this option to instead keep these self-connections.

Options for importing / exporting streamline weights

- **-tck_weights_in path** specify a text scalar file containing the streamline weights
- **-prefix_tck_weights_out prefix** provide a prefix for outputting a text file corresponding to each output file, each containing only the streamline weights relevant for that track file

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.

- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.14 connectomeedit

Synopsis

Perform basic operations on a connectome

Usage

```
connectomeedit [ options ] input operation output
```

- *input*: the input connectome.
- *operation*: the operation to apply, one of: `to_symmetric`, `upper_triangular`, `lower_triangular`, `transpose`, `zero_diagonal`.
- *output*: the output connectome.

Options

Standard options

- **-info** display information messages.

- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Matteo Frigo (matteo.frigo@inria.fr)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.15 connectomestats

Synopsis

Connectome group-wise statistics at the edge level using non-parametric permutation testing

Usage

```
connectomestats [ options ] input algorithm design contrast output
```

- *input*: a text file listing the file names of the input connectomes
- *algorithm*: the algorithm to use in network-based clustering/enhancement. Options are: nbs, tfnbs, none
- *design*: the design matrix
- *contrast*: the contrast matrix
- *output*: the filename prefix for all output.

Description

For the TFNBS algorithm, default parameters for statistical enhancement have been set based on the work in: Vinokur, L.; Zalesky, A.; Raffelt, D.; Smith, R.E. & Connelly, A. A Novel Threshold-Free Network-Based Statistics Method: Demonstration using Simulated Pathology. OHBM, 2015, 4144; and: Vinokur, L.; Zalesky, A.; Raffelt, D.; Smith, R.E. & Connelly, A. A novel threshold-free network-based statistical method: Demonstration and parameter optimisation using in vivo simulated pathology. In Proc ISMRM, 2015, 2846. Note however that not only was the optimisation of these parameters not very precise, but the outcomes of statistical inference (for both this algorithm and the NBS method) can vary markedly for even small changes to enhancement parameters. Therefore the specificity of results obtained using either of these methods should be interpreted with caution.

In some software packages, a column of ones is automatically added to the GLM design matrix; the purpose of this column is to estimate the “global intercept”, which is the predicted value of the observed variable if all explanatory variables were to be zero. However there are rare situations where including such a column would not be appropriate for a particular experimental design. Hence, in MRtrix3 statistical inference commands, it is up to the user to determine whether or not this column of ones should be included in their design matrix, and add it explicitly if necessary. The contrast matrix must also reflect the presence of this additional column.

Options

Options relating to shuffling of data for nonparametric statistical inference

- **-notest** don't perform statistical inference; only output population statistics (effect size, stdev etc)
- **-errors spec** specify nature of errors for shuffling; options are: ee,ise,both (default: ee)
- **-exchange_within file** specify blocks of observations within each of which data may undergo restricted exchange
- **-exchange_whole file** specify blocks of observations that may be exchanged with one another (for independent and symmetric errors, sign-flipping will occur block-wise)
- **-strong** use strong familywise error control across multiple hypotheses
- **-nshuffles number** the number of shuffles (default: 5000)
- **-permutations file** manually define the permutations (relabelling). The input should be a text file defining a $m \times n$ matrix, where each relabelling is defined as a column vector of size m , and the number of columns, n , defines the number of permutations. Can be generated with the `palm_quickperms` function in PALM (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM>). Overrides the `-nshuffles` option.
- **-nonstationarity** perform non-stationarity correction
- **-skew_nonstationarity value** specify the skew parameter for empirical statistic calculation (default for this command is 1)
- **-nshuffles_nonstationarity number** the number of shuffles to use when precomputing the empirical statistic image for non-stationarity correction (default: 5000)
- **-permutations_nonstationarity file** manually define the permutations (relabelling) for computing the empirical statistics for non-stationarity correction. The input should be a text file defining a $m \times n$ matrix, where each relabelling is defined as a column vector of size m , and the number of columns, n , defines the number of permutations. Can be generated with the `palm_quickperms` function in PALM (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM>). Overrides the `-nshuffles_nonstationarity` option.

Options for controlling TFCE behaviour

- **-tfce_dh value** the height increment used in the tfce integration (default: 0.1)
- **-tfce_e value** tfce extent exponent (default: 0.4)
- **-tfce_h value** tfce height exponent (default: 3)

Options related to the General Linear Model (GLM)

- **-variance file** define variance groups for the G-statistic; measurements for which the expected variance is equivalent should contain the same index
- **-ftests path** perform F-tests; input text file should contain, for each F-test, a row containing ones and zeros, where ones indicate the rows of the contrast matrix to be included in the F-test.
- **-fonly** only assess F-tests; do not perform statistical inference on entries in the contrast matrix
- **-column path** (*multiple uses permitted*) add a column to the design matrix corresponding to subject edge-wise values (note that the contrast matrix must include an additional column for each use of this option); the text file provided via this option should contain a file name for each subject

Additional options for connectomestats

- **-threshold value** the t-statistic value to use in threshold-based clustering algorithms

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- If using the NBS algorithm: Zalesky, A.; Fornito, A. & Bullmore, E. T. Network-based statistic: Identifying differences in brain networks. *NeuroImage*, 2010, 53, 1197-1207
- If using the TFNBS algorithm: Baggio, H.C.; Abos, A.; Segura, B.; Campabadal, A.; Garcia-Diaz, A.; Uribe, C.; Compta, Y.; Marti, M.J.; Valldeoriola, F.; Junque, C. Statistical inference in brain graphs using threshold-free network-based statistics. *HBM*, 2018, 39, 2289-2302

- If using the `-nonstationary` option: Salimi-Khorshidi, G.; Smith, S.M. & Nichols, T.E. Adjusting the effect of nonstationarity in cluster-based and TFCE inference. *Neuroimage*, 2011, 54(3), 2006-19

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.16 dcmedit

Synopsis

Edit DICOM file in-place

Usage

```
dcmedit [ options ] file
```

- *file*: the DICOM file to be edited.

Description

Note that this command simply replaces the existing values without modifying the DICOM structure in any way. Replacement text will be truncated if it is too long to fit inside the existing tag.

WARNING: this command will modify existing data! It is recommended to run this command on a copy of the original data set to avoid loss of data.

Options

- **-anonymise** remove any identifiable information, by replacing the following tags: - any tag with Value Representation PN will be replaced with ‘anonymous’ - tag (0010,0030) PatientBirthDate will be replaced with an empty string **WARNING:** there is no guarantee that this command will remove all identifiable information, since such information may be contained in any number of private vendor-specific tags. You will need to double-check the results independently if you need to ensure anonymity.
- **-id text** replace all ID tags with string supplied. This consists of tags (0010, 0020) PatientID and (0010, 1000) OtherPatientIDs
- **-tag group element newvalue** (*multiple uses permitted*) replace specific tag.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.17 dcminfo

Synopsis

Output DICOM fields in human-readable format

Usage

```
dcminfo [ options ] file
```

- *file*: the DICOM file to be scanned.

Options

- **-all** print all DICOM fields.
- **-csa** print all Siemens CSA fields (excluding Phoenix unless requested)
- **-phoenix** print Siemens Phoenix protocol information
- **-tag group element** (*multiple uses permitted*) print field specified by the group & element tags supplied. Tags should be supplied as Hexadecimal (i.e. as they appear in the -all listing).

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.18 dirflip

Synopsis

Invert the polarity of individual directions so as to optimise a unipolar electrostatic repulsion model

Usage

```
dirflip [ options ] in out
```

- *in*: the input files for the directions.
- *out*: the output files for the directions.

Description

The orientations themselves are not affected, only their polarity; this is necessary to ensure near-optimal distribution of DW directions for eddy-current correction.

Options

- **-permutations num** number of permutations to try (default: 100000000)
- **-cartesian** Output the directions in Cartesian coordinates [x y z] instead of [az el].

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.19 **dirgen**

Synopsis

Generate a set of uniformly distributed directions using a bipolar electrostatic repulsion model

Usage

```
dirgen [ options ] ndir dirs
```

- *ndir*: the number of directions to generate.
- *dirs*: the text file to write the directions to, as [az el] pairs.

Description

Directions are distributed by analogy to an electrostatic repulsion system, with each direction corresponding to a single electrostatic charge (for -unipolar), or a pair of diametrically opposed charges (for the default bipolar case). The energy of the system is determined based on the Coulomb repulsion, which assumes the form $1/r^{\text{power}}$, where r is the distance between any pair of charges, and p is the power assumed for the repulsion law (default: 1). The minimum energy state is obtained by gradient descent.

Options

- **-power exp** specify exponent to use for repulsion power law (default: 1). This must be a power of 2 (i.e. 1, 2, 4, 8, 16, ...).
- **-niter num** specify the maximum number of iterations to perform (default: 10000).
- **-restarts num** specify the number of restarts to perform (default: 10).
- **-unipolar** optimise assuming a unipolar electrostatic repulsion model rather than the bipolar model normally assumed in DWI
- **-cartesian** Output the directions in Cartesian coordinates [x y z] instead of [az el].

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).

- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Jones, D.; Horsfield, M. & Simmons, A. Optimal strategies for measuring diffusion in anisotropic systems by magnetic resonance imaging. *Magnetic Resonance in Medicine*, 1999, 42: 515-525

Papadakis, N. G.; Murrills, C. D.; Hall, L. D.; Huang, C. L.-H. & Adrian Carpenter, T. Minimal gradient encoding for robust estimation of diffusion anisotropy. *Magnetic Resonance Imaging*, 2000, 18: 671-679

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.20 dirmerge

Synopsis

Splice / merge multiple sets of directions in such a way as to maintain near-optimality upon truncation

Usage

```
dirmerge [ options ] subsets bvalue files [ bvalue files ... ] out
```

- *subsets*: the number of subsets (eg. phase encoding directions) per b-value
- *bvalue files*: the b-value and sets of corresponding files, in order
- *out*: the output directions file, with each row listing the X Y Z gradient directions, the b-value, and an index representing the phase encode direction

Options

- **-unipolar_weight value** set the weight given to the unipolar electrostatic repulsion model compared to the bipolar model (default: 0.2).

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.21 `dirorder`

Synopsis

Reorder a set of directions to ensure near-uniformity upon truncation

Usage

```
dirorder [ options ] input output
```

- *input*: the input directions file
- *output*: the output directions file

Description

The intent of this command is to reorder a set of gradient directions such that if a scan is terminated prematurely, at any point, the acquired directions will still be close to optimally distributed on the half-sphere.

Options

- **-cartesian** Output the directions in Cartesian coordinates [x y z] instead of [az el].

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.22 dirsplit

Synopsis

Split a set of evenly distributed directions (as generated by `dirgen`) into approximately uniformly distributed subsets

Usage

```
dirsplit [ options ] dirs out [ out ... ]
```

- *dirs*: the text file containing the directions.
- *out*: the output partitioned directions

Options

- **-permutations num** number of permutations to try (default: 100000000)
- **-cartesian** Output the directions in Cartesian coordinates [x y z] instead of [az el].

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.23 dirstat

Synopsis

Report statistics on a direction set

Usage

```
dirstat [ options ] dirs
```

- *dirs*: the text file or image containing the directions.

Description

This command will accept as inputs:

- directions file in spherical coordinates (ASCII text, [az el] space-separated values, one per line);
- directions file in Cartesian coordinates (ASCII text, [x y z] space-separated values, one per line);
- DW gradient files (MRtrix format: ASCII text, [x y z b] space-separated values, one per line);
- image files, using the DW gradient scheme found in the header (or provided using the appropriate command line options below).

By default, this produces all relevant metrics for the direction set provided. If the direction set contains multiple shells, metrics are provided for each shell separately.

Metrics are produced assuming a unipolar or bipolar electrostatic repulsion model, producing the potential energy (total, mean, min & max), and the nearest-neighbour angles (mean, min & max). The condition number is also produced for the spherical harmonic fits up to the highest harmonic order supported by the number of volumes. Finally, the norm of the mean direction vector is provided as a measure of the overall symmetry of the direction set (important with respect to eddy-current resilience).

Specific metrics can also be queried independently via the “-output” option, using these shorthands: U/B for unipolar/bipolar model, E/N for energy and nearest-neighbour respectively, t/-/+ for total/min/max respectively (mean implied otherwise); SHn for condition number of SH fit at order n (with n an even integer); ASYM for asymmetry index (norm of mean direction vector); N for the number of directions.

Example usages

- *Default usage:*

```
$ dirstat directions.txt
```

This provides a pretty-printed list of all metrics available.

- *Write a single metric of interest to standard output:*

```
$ dirstat grad.b -shell 3000 -output SH8
```

requests the condition number of SH fit of b=3000 shell directions at SH order 8

- *Write multiple metrics of interest to standard output:*

```
$ dirstat dwi.mif -output BN,BN-,BN+
```

requests the mean, min and max nearest-neighbour angles assuming a bipolar model.

Options

- **-output list** output selected metrics as a space-delimited list, suitable for use in scripts. This will produce one line of values per selected shell. Valid metrics are as specified in the description above.

DW shell selection options

- **-shells b-values** specify one or more b-values to use during processing, as a comma-separated list of the desired approximate b-values (b-values are clustered to allow for small deviations). Note that some commands are incompatible with multiple b-values, and will report an error if more than one b-value is provided. **WARNING:** note that, even though the b=0 volumes are never referred to as shells in the literature, they still have to be explicitly included in the list of b-values as provided to the -shell option! Several algorithms which include the b=0 volumes in their computations may otherwise return an undesired result.

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.24 dwi2adc

Synopsis

Convert mean dwi (trace-weighted) images to mean ADC maps

Usage

```
dwi2adc [ options ] input output
```

- *input*: the input image.
- *output*: the output image.

Options

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).

- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.25 dwi2fod

Synopsis

Estimate fibre orientation distributions from diffusion data using spherical deconvolution

Usage

```
dwi2fod [ options ] algorithm dwi response odf [ response odf ... ]
```

- *algorithm*: the algorithm to use for FOD estimation. (options are: csd,msmt_csd)
- *dwi*: the input diffusion-weighted image
- *response odf*: pairs of input tissue response and output ODF images

Description

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Example usages

- *Perform single-shell single-tissue CSD:*

```
$ dwi2fod csd dwi.mif response_wm.txt wmfod.mif
```

This algorithm is designed for single-shell data and only uses a single b-value. The response function text file provided should only contain a single row, corresponding to the b-value used for CSD.

- *Perform multi-shell multi-tissue CSD:*

```
$ dwi2fod msmt_csd dwi.mif response_wm.txt wmfod.mif response_gm.txt gm.mif_
↪response_csf.txt csf.mif
```

This example is the most common use case of multi-tissue CSD, estimating a white matter FOD, and grey matter and CSF compartments. This algorithm requires at least three unique b-values to estimate three tissue compartments. Each response function text file should have a number of rows equal to the number of b-values used. If only two unique b-values are available, it's also possible to estimate only two tissue compartments, e.g., white matter and CSF.

Options

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

DW shell selection options

- **-shells b-values** specify one or more b-values to use during processing, as a comma-separated list of the desired approximate b-values (b-values are clustered to allow for small deviations). Note that some commands are incompatible with multiple b-values, and will report an error if more than one b-value is provided. WARNING: note that, even though the b=0 volumes are never referred to as shells in the literature, they still have to be explicitly included in the list of b-values as provided to the -shell option! Several algorithms which include the b=0 volumes in their computations may otherwise return an undesired result.

Options common to more than one algorithm

- **-directions file** specify the directions over which to apply the non-negativity constraint (by default, the built-in 300 direction set is used). These should be supplied as a text file containing [az el] pairs for the directions.
- **-lmax order** the maximum spherical harmonic order for the output FOD(s). For algorithms with multiple outputs, this should be provided as a comma-separated list of integers, one for each output image; for single-output algorithms, only a single integer should be provided. If omitted, the command will use the lmax of the corresponding response function (i.e based on its number of coefficients), up to a maximum of 8.
- **-mask image** only perform computation within the specified binary brain mask image.

Options for the Constrained Spherical Deconvolution algorithm

- **-filter spec** the linear frequency filtering parameters used for the initial linear spherical deconvolution step (default = [1 1 1 0 0]). These should be supplied as a text file containing the filtering coefficients for each even harmonic order.
- **-neg_lambda value** the regularisation parameter lambda that controls the strength of the non-negativity constraint (default = 1).
- **-norm_lambda value** the regularisation parameter lambda that controls the strength of the constraint on the norm of the solution (default = 1).
- **-threshold value** the threshold below which the amplitude of the FOD is assumed to be zero, expressed as an absolute amplitude (default = 0).
- **-niter number** the maximum number of iterations to perform for each voxel (default = 50). Use '-niter 0' for a linear unconstrained spherical deconvolution.

Options for the Multi-Shell, Multi-Tissue Constrained Spherical Deconvolution algorithm

- **-norm_lambda value** the regularisation parameter lambda that controls the strength of the constraint on the norm of the solution (default = 1e-10).
- **-neg_lambda value** the regularisation parameter lambda that controls the strength of the non-negativity constraint (default = 1e-10).
- **-predicted_signal image** output the predicted dwi image.

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- If using csd algorithm: Tournier, J.-D.; Calamante, F. & Connelly, A. Robust determination of the fibre orientation distribution in diffusion MRI: Non-negativity constrained super-resolved spherical deconvolution. *NeuroImage*, 2007, 35, 1459-1472
- If using msmt_csd algorithm: Jeurissen, B; Tournier, J-D; Dhollander, T; Connelly, A & Sijbers, J. Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data. *NeuroImage*, 2014, 103, 411-426

Tournier, J.-D.; Calamante, F., Gadian, D.G. & Connelly, A. Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution. *NeuroImage*, 2004, 23, 1176-1185

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com) and Ben Jeurissen (ben.jeurissen@uantwerpen.be)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.26 dwi2mask

Synopsis

Generates a whole brain mask from a DWI image

Usage

```
dwi2mask [ options ] input output
```

- *input*: the input DWI image containing volumes that are both diffusion weighted and b=0
- *output*: the output whole-brain mask image

Description

All diffusion weighted and b=0 volumes are used to obtain a mask that includes both brain tissue and CSF.

In a second step peninsula-like extensions, where the peninsula itself is wider than the bridge connecting it to the mask, are removed. This may help removing artefacts and non-brain parts, e.g. eyes, from the mask.

Options

- **-clean_scale value** the maximum scale used to cut bridges. A certain maximum scale cuts bridges up to a width (in voxels) of 2x the provided scale. Setting this to 0 disables the mask cleaning step. (Default: 2)

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Dhollander T, Raffelt D, Connelly A. Unsupervised 3-tissue response function estimation from single-shell or multi-shell diffusion MR data without a co-registered T1 image. ISMRM Workshop on Breaking the Barriers of Diffusion MRI, 2016, 5

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au), Thijs Dhollander (thijs.dhollander@gmail.com) and Ben Jeurissen (ben.jeurissen@uantwerpen.be)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.27 dwi2response

Synopsis

Estimate response function(s) for spherical deconvolution

Usage

```
dwi2response algorithm [ options ] ...
```

- *algorithm*: Select the algorithm to be used to complete the script operation; additional details and options become available once an algorithm is nominated. Options are: dhollander, fa, manual, msmt_5tt, tax, tournier

Description

dwi2response offers different algorithms for performing various types of response function estimation. The name of the algorithm must appear as the first argument on the command-line after ‘dwi2response’. The subsequent arguments and options depend on the particular algorithm being invoked.

Each algorithm available has its own help page, including necessary references; e.g. to see the help page of the ‘fa’ algorithm, type ‘dwi2response fa’.

Options

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

General dwi2response options

- **-mask** Provide an initial mask for response voxel selection
- **-voxels** Output an image showing the final voxel selection(s)
- **-shells** The b-value(s) to use in response function estimation (comma-separated list in case of multiple b-values, b=0 must be included explicitly)
- **-lmax** The maximum harmonic degree(s) for response function estimation (comma-separated list in case of multiple b-values)

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and Thijs Dhollander (thijs.dhollander@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.28 dwi2response dhollander

Synopsis

Unsupervised estimation of WM, GM and CSF response functions that does not require a T1 image (or segmentation thereof)

Usage

```
dwi2response dhollander input out_sfwm out_gm out_csf [ options ]
```

- *input*: Input DWI dataset
- *out_sfwm*: Output single-fibre WM response function text file
- *out_gm*: Output GM response function text file
- *out_csf*: Output CSF response function text file

Description

This is an improved version of the Dhollander et al. (2016) algorithm for unsupervised estimation of WM, GM and CSF response functions, which includes the Dhollander et al. (2019) improvements for single-fibre WM response function estimation (prior to this update, the “dwi2response tournier” algorithm had been utilised specifically for the single-fibre WM response function estimation step).

Options

Options for the ‘dhollander’ algorithm

- **-erode** Number of erosion passes to apply to initial (whole brain) mask. Set to 0 to not erode the brain mask. (default: 3)
- **-fa** FA threshold for crude WM versus GM-CSF separation. (default: 0.2)
- **-sfwm** Final number of single-fibre WM voxels to select, as a percentage of refined WM. (default: 0.5 per cent)
- **-gm** Final number of GM voxels to select, as a percentage of refined GM. (default: 2 per cent)
- **-csf** Final number of CSF voxels to select, as a percentage of refined CSF. (default: 10 per cent)
- **-wm_algo algorithm** Use external dwi2response algorithm for WM single-fibre voxel selection (options: fa, tax, tournier) (default: built-in Dhollander 2019)

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

General dwi2response options

- **-mask** Provide an initial mask for response voxel selection
- **-voxels** Output an image showing the final voxel selection(s)
- **-shells** The b-value(s) to use in response function estimation (comma-separated list in case of multiple b-values, b=0 must be included explicitly)
- **-lmax** The maximum harmonic degree(s) for response function estimation (comma-separated list in case of multiple b-values)

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Dhollander, T.; Raffelt, D. & Connelly, A. Unsupervised 3-tissue response function estimation from single-shell or multi-shell diffusion MR data without a co-registered T1 image. ISMRM Workshop on Breaking the Barriers of Diffusion MRI, 2016, 5
- If -wm_algo option is not used: Dhollander, T.; Mito, R.; Raffelt, D. & Connelly, A. Improved white matter response function estimation for 3-tissue constrained spherical deconvolution. Proc Intl Soc Mag Reson Med, 2019, 555

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Thijs Dhollander (thijs.dhollander@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.29 dwi2response fa

Synopsis

Use the old FA-threshold heuristic for single-fibre voxel selection and response function estimation

Usage

```
dwi2response fa input output [ options ]
```

- *input*: The input DWI
- *output*: The output response function text file

Options

Options specific to the ‘fa’ algorithm

- **-erode** Number of brain mask erosion steps to apply prior to threshold (not used if mask is provided manually)
- **-number** The number of highest-FA voxels to use
- **-threshold** Apply a hard FA threshold, rather than selecting the top voxels

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

General dwi2response options

- **-mask** Provide an initial mask for response voxel selection
- **-voxels** Output an image showing the final voxel selection(s)
- **-shells** The b-value(s) to use in response function estimation (comma-separated list in case of multiple b-values, b=0 must be included explicitly)
- **-lmax** The maximum harmonic degree(s) for response function estimation (comma-separated list in case of multiple b-values)

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Tournier, J.-D.; Calamante, F.; Gadian, D. G. & Connelly, A. Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution. *NeuroImage*, 2004, 23, 1176-1185

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.30 dwi2response manual

Synopsis

Derive a response function using an input mask image alone (i.e. pre-selected voxels)

Usage

```
dwi2response manual input in_voxels output [ options ]
```

- *input*: The input DWI
- *in_voxels*: Input voxel selection mask
- *output*: Output response function text file

Options

Options specific to the ‘manual’ algorithm

- **-dirs** Manually provide the fibre direction in each voxel (a tensor fit will be used otherwise)

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

General dwi2response options

- **-mask** Provide an initial mask for response voxel selection
- **-voxels** Output an image showing the final voxel selection(s)
- **-shells** The b-value(s) to use in response function estimation (comma-separated list in case of multiple b-values, b=0 must be included explicitly)
- **-lmax** The maximum harmonic degree(s) for response function estimation (comma-separated list in case of multiple b-values)

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.31 dwi2response msmt_5tt

Synopsis

Derive MSMT-CSD tissue response functions based on a co-registered five-tissue-type (5TT) image

Usage

```
dwi2response msmt_5tt input in_5tt out_wm out_gm out_csf [ options ]
```

- *input*: The input DWI
- *in_5tt*: Input co-registered 5TT image
- *out_wm*: Output WM response text file
- *out_gm*: Output GM response text file
- *out_csf*: Output CSF response text file

Options

Options specific to the ‘msmt_5tt’ algorithm

- **-dirs** Manually provide the fibre direction in each voxel (a tensor fit will be used otherwise)
- **-fa** Upper fractional anisotropy threshold for GM and CSF voxel selection (default: 0.2)
- **-pvf** Partial volume fraction threshold for tissue voxel selection (default: 0.95)
- **-wm_algo algorithm** dwi2response algorithm to use for WM single-fibre voxel selection (options: fa, tax, tournier; default: tournier)
- **-sfwm_fa_threshold** Sets -wm_algo to fa and allows to specify a hard FA threshold for single-fibre WM voxels, which is passed to the -threshold option of the fa algorithm (warning: overrides -wm_algo option)

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

General dwi2response options

- **-mask** Provide an initial mask for response voxel selection
- **-voxels** Output an image showing the final voxel selection(s)
- **-shells** The b-value(s) to use in response function estimation (comma-separated list in case of multiple b-values, b=0 must be included explicitly)
- **-lmax** The maximum harmonic degree(s) for response function estimation (comma-separated list in case of multiple b-values)

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Jeurissen, B.; Tournier, J.-D.; Dhollander, T.; Connelly, A. & Sijbers, J. Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data. *NeuroImage*, 2014, 103, 411-426
- Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.32 dwi2response tax

Synopsis

Use the Tax et al. (2014) recursive calibration algorithm for single-fibre voxel selection and response function estimation

Usage

```
dwi2response tax input output [ options ]
```

- *input*: The input DWI
- *output*: The output response function text file

Options

Options specific to the ‘tax’ algorithm

- **-peak_ratio** Second-to-first-peak amplitude ratio threshold
- **-max_iters** Maximum number of iterations
- **-convergence** Percentile change in any RF coefficient required to continue iterating

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

General dwi2response options

- **-mask** Provide an initial mask for response voxel selection
- **-voxels** Output an image showing the final voxel selection(s)
- **-shells** The b-value(s) to use in response function estimation (comma-separated list in case of multiple b-values, b=0 must be included explicitly)

- **-lmax** The maximum harmonic degree(s) for response function estimation (comma-separated list in case of multiple b-values)

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Tax, C. M.; Jeurissen, B.; Vos, S. B.; Viergever, M. A. & Leemans, A. Recursive calibration of the fiber response function for spherical deconvolution of diffusion MRI data. *NeuroImage*, 2014, 86, 67-80

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.33 dwi2response tournier

Synopsis

Use the Tournier et al. (2013) iterative algorithm for single-fibre voxel selection and response function estimation

Usage

```
dwi2response tournier input output [ options ]
```

- *input*: The input DWI
- *output*: The output response function text file

Options

Options specific to the ‘tournier’ algorithm

- **-number** Number of single-fibre voxels to use when calculating response function
- **-iter_voxels** Number of single-fibre voxels to select when preparing for the next iteration (default = 10 x value given in -number)
- **-dilate** Number of mask dilation steps to apply when deriving voxel mask to test in the next iteration
- **-max_iters** Maximum number of iterations

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

General dwi2response options

- **-mask** Provide an initial mask for response voxel selection
- **-voxels** Output an image showing the final voxel selection(s)
- **-shells** The b-value(s) to use in response function estimation (comma-separated list in case of multiple b-values, b=0 must be included explicitly)
- **-lmax** The maximum harmonic degree(s) for response function estimation (comma-separated list in case of multiple b-values)

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Tournier, J.-D.; Calamante, F. & Connelly, A. Determination of the appropriate b-value and number of gradient directions for high-angular-resolution diffusion-weighted imaging. NMR Biomedicine, 2013, 26, 1775-1786

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.34 dwi2tensor

Synopsis

Diffusion (kurtosis) tensor estimation

Usage

```
dwi2tensor [ options ] dwi dt
```

- *dwi*: the input dwi image.
- *dt*: the output dt image.

Description

By default, the diffusion tensor (and optionally its kurtosis) is fitted to the log-signal in two steps: firstly, using weighted least-squares (WLS) with weights based on the empirical signal intensities; secondly, by further iterated weighted least-squares (IWLS) with weights determined by the signal predictions from the previous iteration (by default, 2 iterations will be performed). This behaviour can be altered in two ways:

- The **-ols** option will cause the first fitting step to be performed using ordinary least-squares (OLS); that is, all measurements contribute equally to the fit, instead of the default behaviour of weighting based on the empirical signal intensities.
- The **-iter** option controls the number of iterations of the IWLS procedure. If this is set to zero, then the output model parameters will be those resulting from the first fitting step only: either WLS by default, or OLS if the **-ols** option is used in conjunction with **-iter 0**.

The tensor coefficients are stored in the output image as follows: volumes 0-5: D11, D22, D33, D12, D13, D23

If diffusion kurtosis is estimated using the **-dkt** option, these are stored as follows: volumes 0-2: W1111, W2222, W3333 volumes 3-8: W1112, W1113, W1222, W1333, W2223, W2333 volumes 9-11: W1122, W1133, W2233 volumes 12-14: W1123, W1223, W1233

Options

- **-ols** perform initial fit using an ordinary least-squares (OLS) fit (see Description).
- **-mask image** only perform computation within the specified binary brain mask image.
- **-b0 image** the output b0 image.
- **-dkt image** the output dkf image.
- **-iter integer** number of iterative reweightings for IWLS algorithm (default: 2) (see Description).
- **-predicted_signal image** the predicted dwi image.

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).

- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

References based on fitting algorithm used:

- OLS, WLS: Basser, P.J.; Mattiello, J.; LeBihan, D. Estimation of the effective self-diffusion tensor from the NMR spin echo. J Magn Reson B., 1994, 103, 247–254.
- IWLS: 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, 335-346

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Ben Jeurissen (ben.jeurissen@uantwerpen.be)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.35 dwibiascorrect

Synopsis

Perform B1 field inhomogeneity correction for a DWI volume series

Usage

```
dwibiascorrect algorithm [ options ] ...
```

- *algorithm*: Select the algorithm to be used to complete the script operation; additional details and options become available once an algorithm is nominated. Options are: ants, fsl

Options

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format

- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

Options common to all dwibiascorrect algorithms

- **-mask image** Manually provide a mask image for bias field estimation
- **-bias image** Output the estimated bias field

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.36 dwibiascorrect ants

Synopsis

Perform DWI bias field correction using the N4 algorithm as provided in ANTs

Usage

```
dwibiascorrect ants input output [ options ]
```

- *input*: The input image series to be corrected
- *output*: The output corrected image series

Options

Options for ANTs N4BiasFieldCorrection command

- **-ants.b [100,3]** N4BiasFieldCorrection option -b. [initial mesh resolution in mm, spline order] This value is optimised for human adult data and needs to be adjusted for rodent data.
- **-ants.c [1000,0.0]** N4BiasFieldCorrection option -c. [numberOfIterations,convergenceThreshold]
- **-ants.s 4** N4BiasFieldCorrection option -s. shrink-factor applied to spatial dimensions

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

Options common to all dwibiascorrect algorithms

- **-mask image** Manually provide a mask image for bias field estimation
- **-bias image** Output the estimated bias field

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Tustison, N.; Avants, B.; Cook, P.; Zheng, Y.; Egan, A.; Yushkevich, P. & Gee, J. N4ITK: Improved N3 Bias Correction. IEEE Transactions on Medical Imaging, 2010, 29, 1310-1320

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.37 dwibiascorrect fsl

Synopsis

Perform DWI bias field correction using the ‘fast’ command as provided in FSL

Usage

```
dwibiascorrect fsl input output [ options ]
```

- *input*: The input image series to be corrected
- *output*: The output corrected image series

Description

The FSL ‘fast’ command only estimates the bias field within a brain mask, and cannot extrapolate this smoothly-varying field beyond the defined mask. As such, this algorithm by necessity introduces a hard masking of the input DWI. Since this attribute may interfere with the purpose of using the command (e.g. correction of a bias field is commonly used to improve brain mask estimation), use of this particular algorithm is generally not recommended.

Options

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

Options common to all dwibiascorrect algorithms

- **-mask image** Manually provide a mask image for bias field estimation
- **-bias image** Output the estimated bias field

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Zhang, Y.; Brady, M. & Smith, S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 2001, 20, 45-57
 - Smith, S. M.; Jenkinson, M.; Woolrich, M. W.; Beckmann, C. F.; Behrens, T. E.; Johansen-Berg, H.; Bannister, P. R.; De Luca, M.; Drobnjak, I.; Flitney, D. E.; Niazy, R. K.; Saunders, J.; Vickers, J.; Zhang, Y.; De Stefano, N.; Brady, J. M. & Matthews, P. M. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 2004, 23, S208-S219
- Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137
-

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.38 dwicat

Synopsis

Concatenating multiple DWI series accounting for differential intensity scaling

Usage

```
dwicat inputs output [ options ]
```

- *inputs*: Multiple input diffusion MRI series
- *output*: The output image series (all DWIs concatenated)

Description

This script concatenates two or more 4D DWI series, accounting for the fact that there may be differences in intensity scaling between those series. This intensity scaling is corrected by determining scaling factors that will make the overall image intensities in the b=0 volumes of each series approximately equivalent.

Options

- **-mask image** Provide a binary mask within which image intensities will be matched

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Lena Dorfschmidt (ld548@cam.ac.uk) and Jakub Vohryzek (jakub.vohryzek@queens.ox.ac.uk) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.39 dwidenoise

Synopsis

dMRI noise level estimation and denoising using Marchenko-Pastur PCA

Usage

```
dwidenoise [ options ] dwi out
```

- *dwi*: the input diffusion-weighted image.
- *out*: the output denoised DWI image.

Description

DWI data denoising and noise map estimation by exploiting data redundancy in the PCA domain using the prior knowledge that the eigenspectrum of random covariance matrices is described by the universal Marchenko-Pastur (MP) distribution. Fitting the MP distribution to the spectrum of patch-wise signal matrices hence provides an estimator of the noise level ‘sigma’, as was first shown in Veraart et al. (2016) and later improved in Cordero-Grande et al. (2019). This noise level estimate then determines the optimal cut-off for PCA denoising.

Important note: image denoising must be performed as the first step of the image processing pipeline. The routine will fail if interpolation or smoothing has been applied to the data prior to denoising.

Note that this function does not correct for non-Gaussian noise biases present in magnitude-reconstructed MRI images. If available, including the MRI phase data can reduce such non-Gaussian biases, and the command now supports complex input data.

Options

- **-mask image** Only process voxels within the specified binary brain mask image.
- **-extent window** Set the patch size of the denoising filter. By default, the command will select the smallest isotropic patch size that exceeds the number of DW images in the input data, e.g., 5x5x5 for data with ≤ 125 DWI volumes, 7x7x7 for data with ≤ 343 DWI volumes, etc.
- **-noise level** The output noise map, i.e., the estimated noise level ‘sigma’ in the data. Note that on complex input data, this will be the total noise level across real and imaginary channels, so a scale factor $\sqrt{2}$ applies.
- **-datatype float32/float64** Datatype for the eigenvalue decomposition (single or double precision). For complex input data, this will select complex float32 or complex float64 datatypes.
- **-estimator Exp1/Exp2** Select the noise level estimator (default = Exp2), either: * Exp1: the original estimator used in Veraart et al. (2016), or * Exp2: the improved estimator introduced in Cordero-Grande et al. (2019).

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.

- **-version** display version information and exit.

References

- Veraart, J.; Novikov, D.S.; Christiaens, D.; Ades-aron, B.; Sijbers, J. & Fieremans, E. Denoising of diffusion MRI using random matrix theory. *NeuroImage*, 2016, 142, 394-406, doi: 10.1016/j.neuroimage.2016.08.016
- Veraart, J.; Fieremans, E. & Novikov, D.S. Diffusion MRI noise mapping using random matrix theory. *Magn. Res. Med.*, 2016, 76(5), 1582-1593, doi: 10.1002/mrm.26059
- Cordero-Grande, L.; Christiaens, D.; Hutter, J.; Price, A.N.; Hajnal, J.V. Complex diffusion-weighted image estimation via matrix recovery under general noise models. *NeuroImage*, 2019, 200, 391-404, doi: 10.1016/j.neuroimage.2019.06.039
- Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Daan Christiaens (daan.christiaens@kcl.ac.uk) & Jelle Veraart (jelle.veraart@nyumc.org) & J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2016 New York University, University of Antwerp, and the MRtrix3 contributors

Permission is hereby granted, free of charge, to any non-commercial entity ('Recipient') obtaining a copy of this software and associated documentation files (the 'Software'), to the Software solely for non-commercial research, including the rights to use, copy and modify the Software, subject to the following conditions:

1. The above copyright notice and this permission notice shall be included by Recipient in all copies or substantial portions of the Software.
2. THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
3. In no event shall NYU be liable for direct, indirect, special, incidental or consequential damages in connection with the Software. Recipient will defend, indemnify and hold NYU harmless from any claims or liability resulting from the use of the Software by recipient.
4. Neither anything contained herein nor the delivery of the Software to recipient shall be deemed to grant the Recipient any right or licenses under any patents or patent application owned by NYU.
5. The Software may only be used for non-commercial research and may not be used for clinical care.
6. Any publication by Recipient of research involving the Software shall cite the references listed below.

5.39.40 dwiextract

Synopsis

Extract diffusion-weighted volumes, b=0 volumes, or certain shells from a DWI dataset

Usage

```
dwiextract [ options ] input output
```

- *input*: the input DW image.
- *output*: the output image (diffusion-weighted volumes by default).

Example usages

- *Calculate the mean $b=0$ image from a 4D DWI series:*

```
$ dwiextract dwi.mif - -bzero | mrmath - mean mean_bzero.mif -axis 3
```

The dwiextract command extracts all volumes for which the b-value is (approximately) zero; the resulting 4D image can then be provided to the mrmath command to calculate the mean intensity across volumes for each voxel.

Options

- **-bzero** Output $b=0$ volumes (instead of the diffusion weighted volumes, if -singleshell is not specified).
- **-no_bzero** Output only non $b=0$ volumes (default, if -singleshell is not specified).
- **-singleshell** Force a single-shell (single non $b=0$ shell) output. This will include $b=0$ volumes, if present. Use with -bzero to enforce presence of $b=0$ volumes (error if not present) or with -no_bzero to exclude them.

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm^2 . If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

DW shell selection options

- **-shells b-values** specify one or more b-values to use during processing, as a comma-separated list of the desired approximate b-values (b-values are clustered to allow for small deviations). Note that some commands are incompatible with multiple b-values, and will report an error if more than one b-value is provided. **WARNING:** note that, even though the $b=0$ volumes are never referred to as shells in the literature, they still have to be explicitly included in the list of b-values as provided to the -shell option! Several algorithms which include the $b=0$ volumes in their computations may otherwise return an undesired result.

DW gradient table export options

- **-export_grad_mrtrix path** export the diffusion-weighted gradient table to file in MRtrix format

- **-export_grad_fsl bvecs_path bvals_path** export the diffusion-weighted gradient table to files in FSL (bvecs / bvals) format

Options for importing phase-encode tables

- **-import_pe_table file** import a phase-encoding table from file
- **-import_pe_eddy config indices** import phase-encoding information from an EDDY-style config / index file pair

Options for selecting volumes based on phase-encoding

- **-pe desc** select volumes with a particular phase encoding; this can be three comma-separated values (for i,j,k components of vector direction) or four (direction & total readout time)

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) and Thijs Dhollander (thijs.dhollander@gmail.com) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.41 dwifslpreproc

Synopsis

Perform diffusion image pre-processing using FSL’s eddy tool; including inhomogeneity distortion correction using FSL’s topup tool if possible

Usage

```
dwifslpreproc input output [ options ]
```

- *input*: The input DWI series to be corrected
- *output*: The output corrected image series

Description

This script is intended to provide convenience of use of the FSL software tools topup and eddy for performing DWI pre-processing, by encapsulating some of the surrounding image data and metadata processing steps. It is intended to simply these processing steps for most commonly-used DWI acquisition strategies, whilst also providing support for some more exotic acquisitions. The “example usage” section demonstrates the ways in which the script can be used based on the (compulsory) `-rpe_*` command-line options.

The “-topup_options” and “-eddy_options” command-line options allow the user to pass desired command-line options directly to the FSL commands topup and eddy. The available options for those commands may vary between versions of FSL; users can interrogate such by querying the help pages of the installed software, and/or the FSL online documentation: (topup) <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/topup/TopupUsersGuide> ; (eddy) <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/eddy/UsersGuide>

The script will attempt to run the CUDA version of eddy; if this does not succeed for any reason, or is not present on the system, the CPU version will be attempted instead. By default, the CUDA eddy binary found that indicates compilation against the most recent version of CUDA will be attempted; this can be over-ridden by providing a soft-link “eddy_cuda” within your path that links to the binary you wish to be executed.

Note that this script does not perform any explicit registration between images provided to topup via the `-se_epi` option, and the DWI volumes provided to eddy. In some instances (motion between acquisitions) this can result in erroneous application of the inhomogeneity field during distortion correction. Use of the `-align_seepi` option is advocated in this scenario, which ensures that the first volume in the series provided to eddy is also the first volume in the series provided to eddy, guaranteeing alignment. But a prerequisite for this approach is that the image contrast within the images provided to the `-se_epi` option must match the `b=0` volumes present within the input DWI series: this means equivalent TE, TR and flip angle (note that differences in multi-band factors between two acquisitions may lead to differences in TR).

Example usages

- *A basic DWI acquisition, where all image volumes are acquired in a single protocol with fixed phase encoding:*

```
$ dwifslpreproc DWI_in.mif DWI_out.mif -rpe_none -pe_dir ap -readout_time 0.55
```

Due to use of a single fixed phase encoding, no EPI distortion correction can be applied in this case.

- *DWIs all acquired with a single fixed phase encoding; but additionally a pair of $b=0$ images with reversed phase encoding to estimate the inhomogeneity field:*

```
$ mrcat b0_ap.mif b0_pa.mif b0_pair.mif -axis 3; dwifslpreproc DWI_in.mif DWI_out.  
↪mif -rpe_pair -se_epi b0_pair.mif -pe_dir ap -readout_time 0.72 -align_seepi
```

Here the two individual $b=0$ volumes are concatenated into a single 4D image series, and this is provided to the script via the `-se_epi` option. Note that with the `-rpe_pair` option used here, which indicates that the SE-EPI image series contains one or more pairs of $b=0$ images with reversed phase encoding, the FIRST HALF of the volumes in the SE-EPI series must possess the same phase encoding as the input DWI series, while the second half are assumed to contain the opposite phase encoding direction but identical total readout time. Use of the `-align_seepi` option is advocated as long as its use is valid (more information in the Description section).

- *All DWI directions & b -values are acquired twice, with the phase encoding direction of the second acquisition protocol being reversed with respect to the first:*

```
$ mrcat DWI_lr.mif DWI_rl.mif DWI_all.mif -axis 3; dwifslpreproc DWI_all.mif DWI_  
↪out.mif -rpe_all -pe_dir lr -readout_time 0.66
```

Here the two acquisition protocols are concatenated into a single DWI series containing all acquired volumes. The direction indicated via the `-pe_dir` option should be the direction of phase encoding used in acquisition of the FIRST HALF of volumes in the input DWI series; ie. the first of the two files that was provided to the `mrcat` command. In this usage scenario, the output DWI series will contain the same number of image volumes as ONE of the acquired DWI series (ie. half of the number in the concatenated series); this is because the script will identify pairs of volumes that possess the same diffusion sensitisation but reversed phase encoding, and perform explicit recombination of those volume pairs in such a way that image contrast in regions of inhomogeneity is determined from the stretched rather than the compressed image.

- *Any acquisition scheme that does not fall into one of the example usages above:*

```
$ mrcat DWI_*.mif DWI_all.mif -axis 3; mrcat b0_*.mif b0_all.mif -axis 3; ↪  
↪dwifslpreproc DWI_all.mif DWI_out.mif -rpe_header -se_epi b0_all.mif -align_  
↪seepi
```

With this usage, the relevant phase encoding information is determined entirely based on the contents of the relevant image headers, and `dwifslpreproc` prepares all metadata for the executed FSL commands accordingly. This can therefore be used if the particular DWI acquisition strategy used does not correspond to one of the simple examples as described in the prior examples. This usage is predicated on the headers of the input files containing appropriately-named key-value fields such that MRtrix3 tools identify them as such. In some cases, conversion from DICOM using MRtrix3 commands will automatically extract and embed this information; however this is not true for all scanner vendors and/or software versions. In the latter case it may be possible to manually provide these metadata; either using the `-json_import` command-line option of `dwifslpreproc`, or the `-json_import` or one of the `-import_pe_*` command-line options of MRtrix3's `mrconvert` command (and saving in .mif format) prior to running `dwifslpreproc`.

Options

- **-pe_dir PE** Manually specify the phase encoding direction of the input series; can be a signed axis number (e.g. -0, 1, +2), an axis designator (e.g. RL, PA, IS), or NIfTI axis codes (e.g. i-, j, k)
- **-readout_time time** Manually specify the total readout time of the input series (in seconds)
- **-se_epi image** Provide an additional image series consisting of spin-echo EPI images, which is to be used exclusively by topup for estimating the inhomogeneity field (i.e. it will not form part of the output image series)
- **-align_seepi** Achieve alignment between the SE-EPI images used for inhomogeneity field estimation, and the DWIs (more information in Description section)
- **-json_import file** Import image header information from an associated JSON file (may be necessary to determine phase encoding information)
- **-topup_options "TopupOptions"** Manually provide additional command-line options to the topup command (provide a string within quotation marks that contains at least one space, even if only passing a single command-line option to topup)
- **-eddy_options "EddyOptions"** Manually provide additional command-line options to the eddy command (provide a string within quotation marks that contains at least one space, even if only passing a single command-line option to eddy)
- **-eddy_mask image** Provide a processing mask to use for eddy, instead of having dwifslpreproc generate one internally using dwi2mask
- **-eddy_slspec file** Provide a file containing slice groupings for eddy's slice-to-volume registration
- **-eddyqc_text directory** Copy the various text-based statistical outputs generated by eddy, and the output of eddy_qc (if installed), into an output directory
- **-eddyqc_all directory** Copy ALL outputs generated by eddy (including images), and the output of eddy_qc (if installed), into an output directory

Options for specifying the acquisition phase-encoding design; note that one of the -rpe_* options MUST be provided

- **-rpe_none** Specify that no reversed phase-encoding image data is being provided; eddy will perform eddy current and motion correction only
- **-rpe_pair** Specify that a set of images (typically b=0 volumes) will be provided for use in inhomogeneity field estimation only (using the -se_epi option)
- **-rpe_all** Specify that ALL DWIs have been acquired with opposing phase-encoding
- **-rpe_header** Specify that the phase-encoding information can be found in the image header(s), and that this is the information that the script should use

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

Options for exporting the diffusion gradient table

- **-export_grad_mrtrix grad** Export the final gradient table in MRtrix format
- **-export_grad_fsl bvecs bvals** Export the final gradient table in FSL bvecs/bvals format

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Andersson, J. L. & Sotiropoulos, S. N. An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging. *NeuroImage*, 2015, 125, 1063-1078
- Smith, S. M.; Jenkinson, M.; Woolrich, M. W.; Beckmann, C. F.; Behrens, T. E.; Johansen-Berg, H.; Bannister, P. R.; De Luca, M.; Drobnjak, I.; Flitney, D. E.; Niazy, R. K.; Saunders, J.; Vickers, J.; Zhang, Y.; De Stefano, N.; Brady, J. M. & Matthews, P. M. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 2004, 23, S208-S219
- If performing recombination of diffusion-weighted volume pairs with opposing phase encoding directions: Skare, S. & Bammer, R. Jacobian weighting of distortion corrected EPI data. *Proceedings of the International Society for Magnetic Resonance in Medicine*, 2010, 5063
- If performing EPI susceptibility distortion correction: Andersson, J. L.; Skare, S. & Ashburner, J. How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging. *NeuroImage*, 2003, 20, 870-888
- If including “-repol” in -eddy_options input: Andersson, J. L. R.; Graham, M. S.; Zsoldos, E. & Sotiropoulos, S. N. Incorporating outlier detection and replacement into a non-parametric framework for movement and distortion correction of diffusion MR images. *NeuroImage*, 2016, 141, 556-572

- If including “-mporder” in -eddy_options input: Andersson, J. L. R.; Graham, M. S.; Drobnjak, I.; Zhang, H.; Filippini, N. & Bastiani, M. Towards a comprehensive framework for movement and distortion correction of diffusion MR images: Within volume movement. *NeuroImage*, 2017, 152, 450-466
- If using -eddyqc_test or -eddyqc_all option and eddy_quad is installed: Bastiani, M.; Cottaar, M.; Fitzgibbon, S.P.; Suri, S.; Alfaro-Almagro, F.; Sotiropoulos, S.N.; Jbabdi, S.; Andersson, J.L.R. Automated quality control for within and between studies diffusion MRI data using a non-parametric framework for movement and distortion correction. *NeuroImage*, 2019, 184, 801-812

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.42 dwigradcheck

Synopsis

Check the orientation of the diffusion gradient table

Usage

```
dwigradcheck input [ options ]
```

- *input*: The input DWI series to be checked

Options

- **-mask image** Provide a brain mask image
- **-number** Set the number of tracks to generate for each test

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

Options for exporting the diffusion gradient table

- **-export_grad_mrtrix grad** Export the final gradient table in MRtrix format
- **-export_grad_fsl bvecs bvals** Export the final gradient table in FSL bvecs/bvals format

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Jeurissen, B.; Leemans, A.; Sijbers, J. Automated correction of improperly rotated diffusion gradient orientations in diffusion weighted MRI. Medical Image Analysis, 2014, 18(7), 953-962

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.43 dwinormalise

Synopsis

Perform various forms of intensity normalisation of DWIs

Usage

```
dwinormalise algorithm [ options ] ...
```

- *algorithm*: Select the algorithm to be used to complete the script operation; additional details and options become available once an algorithm is nominated. Options are: group, individual

Description

This script provides access to different techniques for globally scaling the intensity of diffusion-weighted images. The different algorithms have different purposes, and different requirements with respect to the data with which they must be provided & will produce as output. Further information on the individual algorithms available can be accessed via their individual help pages; eg. “dwinormalise group -help”.

Options

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.44 dwinormalise group

Synopsis

Performs a global DWI intensity normalisation on a group of subjects using the median $b=0$ white matter value as the reference

Usage

```
dwinormalise group input_dir mask_dir output_dir fa_template wm_mask [ options ]
```

- *input_dir*: The input directory containing all DWI images
- *mask_dir*: Input directory containing brain masks, corresponding to one per input image (with the same file name prefix)
- *output_dir*: The output directory containing all of the intensity normalised DWI images
- *fa_template*: The output population specific FA template, which is threshold to estimate a white matter mask
- *wm_mask*: The output white matter mask (in template space), used to estimate the median $b=0$ white matter value for normalisation

Description

The white matter mask is estimated from a population average FA template then warped back to each subject to perform the intensity normalisation. Note that bias field correction should be performed prior to this step.

All input DWI files must contain an embedded diffusion gradient table; for this reason, these images must all be in either .mif or .mif.gz format.

Options

- **-fa_threshold** The threshold applied to the Fractional Anisotropy group template used to derive an approximate white matter mask (default: 0.4)

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.45 dwinormalise individual

Synopsis

Intensity normalise a DWI series based on the b=0 signal within a supplied mask

Usage

```
dwinormalise individual input_dwi input_mask output_dwi [ options ]
```

- *input_dwi*: The input DWI series
- *input_mask*: The mask within which a reference b=0 intensity will be sampled
- *output_dwi*: The output intensity-normalised DWI series

Options

- **-intensity** Normalise the b=0 signal to a specified value (Default: 1000)
- **-percentile** Define the percentile of the b=0 image intensities within the mask used for normalisation; if this option is not supplied then the median value (50th percentile) will be normalised to the desired intensity value

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.46 dwishellmath

Synopsis

Apply an mrmath operation to each b-value shell in a DWI series

Usage

```
dwishellmath input operation output [ options ]
```

- *input*: The input diffusion MRI series
- *operation*: The operation to be applied to each shell; this must be one of the following: mean, median, sum, product, rms, norm, var, std, min, max, absmax, magmax
- *output*: The output image series

Description

The output of this command is a 4D image, where each volume corresponds to a b-value shell (in order of increasing b-value), and the intensities within each volume correspond to the chosen statistic having been computed from across the DWI volumes belonging to that b-value shell.

Example usages

- *To compute the mean diffusion-weighted signal in each b-value shell:*

```
$ dwishellmath dwi.mif mean shellmeans.mif
```

Options

Options for importing the diffusion gradient table

- **-grad** Provide the diffusion gradient table in MRtrix format
- **-fslgrad bvecs bvals** Provide the diffusion gradient table in FSL bvecs/bvals format

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Daan Christiaens (daan.christiaens@kcl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.47 fixel2peaks

Synopsis

Convert data in the fixel directory format into a 4D image of 3-vectors

Usage

```
fixel2peaks [ options ] in out
```

- *in*: the input fixel information
- *out*: the output peaks image

Description

If a fixel data file is provided as input, then the 3-vectors in the output image will be scaled based on the data in that file. If the input is instead the fixel directory, or the index or directions file, then all output 3-vectors will possess unit norm.

Options

- **-number value** maximum number of fixels in each voxel (default: based on input data)
- **-nan** fill excess peak data with NaNs rather than zeroes

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.48 fixel2sh

Synopsis

Convert a fixel-based sparse-data image into an spherical harmonic image

Usage

```
fixel2sh [ options ] fixel_in sh_out
```

- *fixel_in*: the input fixel data file.
- *sh_out*: the output sh image.

Description

This command generates spherical harmonic data from fixels that can be visualised using the ODF tool in MRview. The output ODF lobes are scaled according to the values in the input fixel image.

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Options

- **-lmax order** set the maximum harmonic order for the output series (Default: 8)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.

- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) & David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.49 fixel2tsf

Synopsis

Map fixel values to a track scalar file based on an input tractogram

Usage

```
fixel2tsf [ options ] fixel_in tracks tsf
```

- *fixel_in*: the input fixel data file (within the fixel directory)
- *tracks*: the input track file
- *tsf*: the output track scalar file

Description

This command is useful for visualising all brain fixels (e.g. the output from `fixelcfestats`) in 3D.

Options

- **-angle value** the max angular threshold for computing correspondence between a fixel direction and track tangent (default = 45 degrees)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.50 fixel2voxel

Synopsis

Convert a fixel-based sparse-data image into some form of scalar image

Usage

```
fixel2voxel [ options ] fixel_in operation image_out
```

- *fixel_in*: the input fixel data file
- *operation*: the operation to apply, one of: mean, sum, product, min, max, absmax, magmax, count, complexity, sf, dec_unit, dec_scaled, none.
- *image_out*: the output scalar image.

Description

Fixel data can be reduced to voxel data in a number of ways:

- Some statistic computed across all fixel values within a voxel: mean, sum, product, min, max, absmax, magmax
- The number of fixels in each voxel: count
- Some measure of crossing-fibre organisation: complexity, sf ('single-fibre')
- A 4D directionally-encoded colour image: dec_unit, dec_scaled
- A 4D image containing all fixel data values in each voxel unmodified: none

The -weighted option deals with the case where there is some per-fixel metric of interest that you wish to collapse into a single scalar measure per voxel, but each fixel possesses a different volume, and you wish for those fixels with greater volume to have a greater influence on the calculation than fixels with lesser volume. For instance, when estimating a voxel-based measure of mean axon diameter from per-fixel mean axon diameters, a fixel's mean axon diameter should be weighed by its relative volume within the voxel in the calculation of that voxel mean.

Options

- **-number N** use only the largest N fixels in calculation of the voxel-wise statistic; in the case of operation "none", output only the largest N fixels in each voxel.
- **-fill value** for "none" operation, specify the value to fill when number of fixels is fewer than the maximum (default: 0.0)
- **-weighted fixel_in** weight the contribution of each fixel to the per-voxel result according to its volume.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Reference for 'complexity' operation: Riffert, T. W.; Schreiber, J.; Anwender, A. & Knosche, T. R. Beyond Fractional Anisotropy: Extraction of bundle-specific structural metrics from crossing fibre models. NeuroImage, 2014, 100, 176-191

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) & David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.51 fixelcfestats

Synopsis

Fixel-based analysis using connectivity-based fixel enhancement and non-parametric permutation testing

Usage

```
fixelcfestats [ options ] in_fixel_directory subjects design contrast connectivity_
↪out_fixel_directory
```

- *in_fixel_directory*: the fixel directory containing the data files for each subject (after obtaining fixel correspondence)
- *subjects*: a text file listing the subject identifiers (one per line). This should correspond with the filenames in the fixel directory (including the file extension), and be listed in the same order as the rows of the design matrix.
- *design*: the design matrix
- *contrast*: the contrast matrix, specified as rows of weights
- *connectivity*: the fixel-fixel connectivity matrix
- *out_fixel_directory*: the output directory where results will be saved. Will be created if it does not exist

Description

Unlike previous versions of this command, where a whole-brain tractogram file would be provided as input in order to generate the fixel-fixel connectivity matrix and smooth fixel data, this version expects to be provided with the directory path to a pre-calculated fixel-fixel connectivity matrix (likely generated using the MRtrix3 command `fixelconnectivity`), and for the input fixel data to have already been smoothed (likely using the MRtrix3 command `fixelfilter`).

Note that if the `-mask` option is used, the output fixel directory will still contain the same set of fixels as that present in the input fixel template, in order to retain fixel correspondence. However a consequence of this is that all fixels in the template will be initially visible when the output fixel directory is loaded in `mrview`. Those fixels outside the processing mask will immediately disappear from view as soon as any data-file-based fixel colouring or thresholding is applied.

In some software packages, a column of ones is automatically added to the GLM design matrix; the purpose of this column is to estimate the “global intercept”, which is the predicted value of the observed variable if all explanatory variables were to be zero. However there are rare situations where including such a column would not be appropriate for a particular experimental design. Hence, in MRtrix3 statistical inference commands, it is up to the user to determine whether or not this column of ones should be included in their design matrix, and add it explicitly if necessary. The contrast matrix must also reflect the presence of this additional column.

Options

- **-mask file** provide a fixel data file containing a mask of those fixels to be used during processing

Options relating to shuffling of data for nonparametric statistical inference

- **-notest** don't perform statistical inference; only output population statistics (effect size, stdev etc)
- **-errors spec** specify nature of errors for shuffling; options are: ee,ise,both (default: ee)
- **-exchange_within file** specify blocks of observations within each of which data may undergo restricted exchange
- **-exchange_whole file** specify blocks of observations that may be exchanged with one another (for independent and symmetric errors, sign-flipping will occur block-wise)
- **-strong** use strong familywise error control across multiple hypotheses
- **-nshuffles number** the number of shuffles (default: 5000)
- **-permutations file** manually define the permutations (relabelling). The input should be a text file defining a $m \times n$ matrix, where each relabelling is defined as a column vector of size m , and the number of columns, n , defines the number of permutations. Can be generated with the `palm_quickperms` function in PALM (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM>). Overrides the `-nshuffles` option.
- **-nonstationarity** perform non-stationarity correction
- **-skew_nonstationarity value** specify the skew parameter for empirical statistic calculation (default for this command is 1)
- **-nshuffles_nonstationarity number** the number of shuffles to use when precomputing the empirical statistic image for non-stationarity correction (default: 5000)
- **-permutations_nonstationarity file** manually define the permutations (relabelling) for computing the empirical statistics for non-stationarity correction. The input should be a text file defining a $m \times n$ matrix, where each relabelling is defined as a column vector of size m , and the number of columns, n , defines the number of permutations. Can be generated with the `palm_quickperms` function in PALM (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM>) Overrides the `-nshuffles_nonstationarity` option.

Parameters for the Connectivity-based Fixel Enhancement algorithm

- **-cfe_dh value** the height increment used in the cfe integration (default: 0.1)
- **-cfe_e value** cfe extent exponent (default: 2)
- **-cfe_h value** cfe height exponent (default: 3)
- **-cfe_c value** cfe connectivity exponent (default: 0.5)
- **-cfe_legacy** use the legacy (non-normalised) form of the cfe equation

Options related to the General Linear Model (GLM)

- **-variance file** define variance groups for the G-statistic; measurements for which the expected variance is equivalent should contain the same index
- **-ftests path** perform F-tests; input text file should contain, for each F-test, a row containing ones and zeros, where ones indicate the rows of the contrast matrix to be included in the F-test.
- **-fonly** only assess F-tests; do not perform statistical inference on entries in the contrast matrix
- **-column path** (*multiple uses permitted*) add a column to the design matrix corresponding to subject fixel-wise values (note that the contrast matrix must include an additional column for each use of this option); the text file provided via this option should contain a file name for each subject

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Raffelt, D.; Smith, R.E.; Ridgway, G.R.; Tournier, J.D.; Vaughan, D.N.; Rose, S.; Henderson, R.; Connelly, A. Connectivity-based fixel enhancement: Whole-brain statistical analysis of diffusion MRI measures in the presence of crossing fibres. *Neuroimage*, 2015, 15(117):40-55

- If not using the `-cfe_legacy` option: Smith, R.E.; Dimond, D.; Vaughan, D.; Parker, D.; Dhollander, T.; Jackson, G.; Connelly, A. Intrinsic non-stationarity correction for Fixel-Based Analysis. In *Proc OHBM 2019* M789
- If using the `-nonstationary` option: Salimi-Khorshidi, G. Smith, S.M. Nichols, T.E. Adjusting the effect of nonstationarity in cluster-based and TFCE inference. *NeuroImage*, 2011, 54(3), 2006-19

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.52 fixelconnectivity

Synopsis

Generate a fixel-fixel connectivity matrix

Usage

```
fixelconnectivity [ options ] fixel_directory tracks matrix
```

- *fixel_directory*: the directory containing the fixels between which connectivity will be quantified
- *tracks*: the tracks used to determine fixel-fixel connectivity
- *matrix*: the output fixel-fixel connectivity matrix directory path

Description

This command will generate a directory containing three images, which encodes the fixel-fixel connectivity matrix. Documentation regarding this format and how to use it will come in the future.

Options

Options that influence generation of the connectivity matrix / matrices

- **-threshold value** a threshold to define the required fraction of shared connections to be included in the neighbourhood (default: 0.01)
- **-angle value** the max angle threshold for assigning streamline tangents to fixels (Default: 45 degrees)
- **-mask file** provide a fixel data file containing a mask of those fixels to be computed; fixels outside the mask will be empty in the output matrix

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).

- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.53 fixelconvert

Synopsis

Convert between the old format fixel image (.msf / .msh) and the new fixel directory format

Usage

```
fixelconvert [ options ] fixel_in fixel_out
```

- *fixel_in*: the input fixel file / directory.
- *fixel_out*: the output fixel file / directory.

Example usages

- *Convert from the old file format to the new directory format:*

```
$ fixelconvert old_fixels.msf new_fixels/ -out_size
```

This performs a simple conversion from old to new format, and additionally writes the contents of the “size” field within old-format fixel images stored using the “FixelMetric” class (likely all of them) as an additional fixel data file.

- *Convert multiple files from old to new format, preserving fixel correspondence:*

```
$ for_each *.msf : fixelconvert IN NAME_new/ -template template_fixels/
```

In this example, the `for_each` script is used to execute the `fixelconvert` command once for each of a series of input files in the old fixel format, generating a new output fixel directory for each. Importantly here though, the `-template` option is used to ensure that the ordering of fixels within these output directories is identical, such that fixel data files can be exchanged between them (e.g. accumulating fixel data files across subjects into a single template fixel directory)

- *Convert from the new directory format to the old file format:*

```
$ fixelconvert new_fixels/ old_fixels.msf -value parameter.mif -in_size new_
↪fixels/afd.mif
```

Conversion from the new directory format will contain the value 1.0 for all output fixels in both the “size” and “value” fields of the “FixelMetric” class, unless the `-in_size` and/or `-value` options are used respectively to indicate which fixel data files should be used as the source(s) of this information.

Options

Options for converting from old to new format

- **-name string** assign a different name to the value field output (Default: `value`). Do not include the file extension.
- **-nii** output the index, directions and data file in NIfTI format instead of `.mif`
- **-out_size** also output the ‘size’ field from the old format
- **-template path** specify an existing fixel directory (in the new format) to which the new output should conform

Options for converting from new to old format

- **-value path** nominate the data file to import to the ‘value’ field in the old format
- **-in_size path** import data for the ‘size’ field in the old format

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.54 fixelcorrespondence

Synopsis

Obtain fixel-fixel correspondence between a subject fixel image and a template fixel mask

Usage

```
fixelcorrespondence [ options ] subject_data template_directory output_directory_
↪output_data
```

- *subject_data*: the input subject fixel data file. This should be a file inside the fixel directory
- *template_directory*: the input template fixel directory.
- *output_directory*: the fixel directory where the output file will be written.
- *output_data*: the name of the output fixel data file. This will be placed in the output fixel directory

Description

It is assumed that the subject image has already been spatially normalised and is aligned with the template. The output fixel image will have the same fixels (and directions) of the template.

Options

- **-angle value** the max angle threshold for computing inter-subject fixel correspondence (Default: 45 degrees)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.55 fixelcrop

Synopsis

Crop/remove fixels from sparse fixel image using a binary fixel mask

Usage

```
fixelcrop [ options ] input_fixel_directory input_fixel_mask output_fixel_directory
```

- *input_fixel_directory*: input fixel directory, all data files and directions file will be cropped and saved in the output fixel directory
- *input_fixel_mask*: the input fixel data file defining which fixels to crop. Fixels with zero values will be removed
- *output_fixel_directory*: the output directory to store the cropped directions and data files

Description

The mask must be input as a fixel data file the same dimensions as the fixel data file(s) to be cropped.

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) & Rami Tabarra (rami.tabarra@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.56 fixelfilter

Synopsis

Perform filtering operations on fixel-based data

Usage

```
fixelfilter [ options ] input filter output
```

- *input*: the input: either a fixel data file, or a fixel directory (see Description)
- *filter*: the filtering operation to perform; options are: connect, smooth
- *output*: the output: either a fixel data file, or a fixel directory (see Description)

Description

If the first input to the command is a specific fixel data file, then a filtered version of only that file will be generated by the command. Alternatively, if the input is the location of a fixel directory, then the command will create a duplicate of the fixel directory, and apply the specified filter operation to all fixel data files within the directory.

Options

- **-matrix file** provide a fixel-fixel connectivity matrix for filtering operations that require it

Options specific to the “connect” filter

- **-threshold_value value** specify a threshold for the input fixel data file values (default = 0.5)
- **-threshold_connectivity value** specify a fixel-fixel connectivity threshold for connected-component analysis (default = 0.10000000000000001)

Options specific to the “smooth” filter

- **-fwhm value** the full-width half-maximum (FWHM) of the spatial component of the smoothing filter (default = 10mm)
- **-minweight value** apply a minimum threshold to smoothing weights (default = 0.01)
- **-mask image** only perform smoothing within a specified binary fixel mask

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.

- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.57 fixelreorient

Synopsis

Reorient fixel directions

Usage

```
fixelreorient [ options ] fixel_in warp fixel_out
```

- *fixel_in*: the input fixel directory
- *warp*: a 4D deformation field used to perform reorientation. Reorientation is performed by applying the Jacobian affine transform in each voxel in the warp, then re-normalising the vector representing the fixel direction
- *fixel_out*: the output fixel directory. If the the input and output directories are the same, the existing directions file will be replaced (providing the **-force** option is supplied). If a new directory is supplied then the fixel directions and all other fixel data will be copied to the new directory.

Description

Reorientation is performed by transforming the vector representing the fixel direction with the Jacobian (local affine transform) computed at each voxel in the warp, then re-normalising the vector.

Options

Standard options

- **-info** display information messages.

- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.58 fod2dec

Synopsis

Generate FOD-based DEC maps, with optional panchromatic sharpening and/or luminance/perception correction

Usage

```
fod2dec [ options ] input output
```

- *input*: The input FOD image (spherical harmonic coefficients).
- *output*: The output DEC image (weighted RGB triplets).

Description

By default, the FOD-based DEC is weighted by the integral of the FOD. To weight by another scalar map, use the `-contrast` option. This option can also be used for panchromatic sharpening, e.g., by supplying a T1 (or other sensible) anatomical volume with a higher spatial resolution.

Options

- **-mask image** Only perform DEC computation within the specified mask image.
- **-contrast image** Weight the computed DEC map by the provided image contrast. If the contrast has a different image grid, the DEC map is first resliced and renormalised. To achieve panchromatic sharpening, provide an image with a higher spatial resolution than the input FOD image; e.g., a T1 anatomical volume. Only the DEC is subject to the mask, so as to allow for partial colouring of the contrast image. Default when this option is *not* provided: integral of input FOD, subject to the same mask/threshold as used for DEC computation.
- **-lum** Correct for luminance/perception, using default values Cr,Cg,Cb = 0.3,0.5,0.2 and gamma = 2.2 (*not* correcting is the theoretical equivalent of Cr,Cg,Cb = 1,1,1 and gamma = 2).
- **-lum_coefs values** The coefficients Cr,Cg,Cb to correct for luminance/perception. Note: this implicitly switches on luminance/perception correction, using a default gamma = 2.2 unless specified otherwise.
- **-lum_gamma value** The gamma value to correct for luminance/perception. Note: this implicitly switches on luminance/perception correction, using a default Cr,Cg,Cb = 0.3,0.5,0.2 unless specified otherwise.
- **-threshold value** FOD amplitudes below the threshold value are considered zero.
- **-no_weight** Do not weight the DEC map; just output the unweighted colours. Reslicing and renormalising of colours will still happen when providing the -contrast option as a template.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Dhollander T, Smith RE, Tournier JD, Jeurissen B, Connelly A. Time to move on: an FOD-based DEC map to replace DTI's trademark DEC FA. *Proc Intl Soc Mag Reson Med*, 2015, 23, 1027

Dhollander T, Raffelt D, Smith RE, Connelly A. Panchromatic sharpening of FOD-based DEC maps by structural T1 information. *Proc Intl Soc Mag Reson Med*, 2015, 23, 566

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Thijs Dhollander (thijs.dhollander@gmail.com)

Copyright: Copyright (C) 2014 The Florey Institute of Neuroscience and Mental Health, Melbourne, Australia. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

5.39.59 fod2fixel

Synopsis

Perform segmentation of continuous Fibre Orientation Distributions (FODs) to produce discrete fixels

Usage

```
fod2fixel [ options ] fod fixel_directory
```

- *fod*: the input fod image.
- *fixel_directory*: the output fixel directory

Options

Metric values for fixel-based sparse output images

- **-afd image** output the total Apparent Fibre Density per fixel (integral of FOD lobe)
- **-peak_amp image** output the amplitude of the FOD at the maximal peak per fixel
- **-disp image** output a measure of dispersion per fixel as the ratio between FOD lobe integral and maximal peak amplitude

FOD FMLS segmenter options

- **-fmls_integral value** threshold absolute numerical integral of positive FOD lobes. Any lobe for which the integral is smaller than this threshold will be discarded. Default: 0.
- **-fmls_peak_value value** threshold peak amplitude of positive FOD lobes. Any lobe for which the maximal peak amplitude is smaller than this threshold will be discarded. Default: 0.1.
- **-fmls_no_thresholds** disable all FOD lobe thresholding; every lobe where the FOD is positive will be retained.
- **-fmls_lobe_merge_ratio value** Specify the ratio between a given FOD amplitude sample between two lobes, and the smallest peak amplitude of the adjacent lobes, above which those lobes will be merged. This is the amplitude of the FOD at the ‘bridge’ point between the two lobes, divided by the peak amplitude of the smaller of the two adjoining lobes. A value of 1.0 will never merge two lobes into one; a value of 0.0 will always merge lobes unless they are bisected by a zero-valued crossing. Default: 1.

Other options for fod2fixel

- **-mask image** only perform computation within the specified binary brain mask image.
- **-maxnum number** maximum number of fixels to output for any particular voxel (default: no limit)
- **-nii** output the directions and index file in nii format (instead of the default mif)

- **-dirpeak** define the fixel direction as that of the lobe's maximal peak as opposed to its weighted mean direction (the default)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Reference for the FOD segmentation method: Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. SIFT: Spherical-deconvolution informed filtering of tractograms. NeuroImage, 2013, 67, 298-312 (Appendix 2)
- Reference for Apparent Fibre Density (AFD): Raffelt, D.; Tournier, J.-D.; Rose, S.; Ridgway, G.R.; Henderson, R.; Crozier, S.; Salvado, O.; Connelly, A. Apparent Fibre Density: a novel measure for the analysis of diffusion-weighted magnetic resonance images. Neuroimage, 2012, 15;59(4), 3976-94

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.60 for_each

Synopsis

Perform some arbitrary processing step for each of a set of inputs

Usage

```
for_each inputs colon command [ options ]
```

- *inputs*: Each of the inputs for which processing should be run
- *colon*: Colon symbol (":") delimiting the `for_each` inputs & command-line options from the actual command to be executed
- *command*: The command string to run for each input, containing any number of substitutions listed in the Description section

Description

This script greatly simplifies various forms of batch processing by enabling the execution of a command (or set of commands) independently for each of a set of inputs. Part of the way that this is achieved is by providing basic text substitutions, which simplify the formation of valid command strings based on the unique components of the input strings on which the script is instructed to execute. The available substitutions are listed below (note that the `-test` command-line option can be used to ensure correct command string formation prior to actually executing the commands):

- **IN**: The full matching pattern, including leading folders. For example, if the target list contains a file "folder/image.mif", any occurrence of "IN" will be substituted with "folder/image.mif".
- **NAME**: The basename of the matching pattern. For example, if the target list contains a file "folder/image.mif", any occurrence of "NAME" will be substituted with "image.mif".
- **PRE**: The prefix of the input pattern (the basename stripped of its extension). For example, if the target list contains a file "folder/my.image.mif.gz", any occurrence of "PRE" will be substituted with "my.image".
- **UNI**: The unique part of the input after removing any common prefix and common suffix. For example, if the target list contains files: "folder/001dwi.mif", "folder/002dwi.mif", "folder/003dwi.mif", any occurrence of "UNI" will be substituted with "001", "002", "003".

Note that due to a limitation of the Python "argparse" module, any command-line **OPTIONS** that the user intends to provide specifically to the `for_each` script must appear **BEFORE** providing the list of inputs on which `for_each` is intended to operate. While command-line options provided as such will be interpreted specifically by the `for_each` script, any command-line options that are provided **AFTER** the **COLON** separator will form part of the executed **COMMAND**, and will therefore be interpreted as command-line options having been provided to that underlying command.

Example usages

- *Demonstration of basic usage syntax:*

```
$ for_each folder/*.mif : mrinfo IN
```

This will run the "mrinfo" command for every .mif file present in "folder". Note that the compulsory colon symbol is used to separate the list of items on which `for_each` is being instructed to operate, from the command that is intended to be run for each input.

- *Multi-threaded use of `for_each`:*

```
$ for_each -nthreads 4 freesurfer/subjects/* : recon-all -subjid NAME -all
```

In this example, `for_each` is instructed to run the FreeSurfer command ‘`recon-all`’ for all subjects within the ‘`subjects`’ directory, with four subjects being processed in parallel at any one time. Whenever processing of one subject is completed, processing for a new unprocessed subject will commence. This technique is useful for improving the efficiency of running single-threaded commands on multi-core systems, as long as the system possesses enough memory to support such parallel processing. Note that in the case of multi-threaded commands (which includes many MRtrix3 commands), it is generally preferable to permit multi-threaded execution of the command on a single input at a time, rather than processing multiple inputs in parallel.

- *Excluding specific inputs from execution:*

```
$ for_each *.nii -exclude 001.nii : mrconvert IN PRE.mif
```

Particularly when a wildcard is used to define the list of inputs for `for_each`, it is possible in some instances that this list will include one or more strings for which execution should in fact not be performed; for instance, if a command has already been executed for one or more files, and then `for_each` is being used to execute the same command for all other files. In this case, the `-exclude` option can be used to effectively remove an item from the list of inputs that would otherwise be included due to the use of a wildcard (and can be used more than once to exclude more than one string). In this particular example, `mrconvert` is instructed to perform conversions from NIFTI to MRtrix image formats, for all except the first image in the directory. Note that any usages of this option must appear **AFTER** the list of inputs. Note also that the argument following the `-exclude` option can alternatively be a regular expression, in which case any inputs for which a match to the expression is found will be excluded from processing.

- *Testing the command string substitution:*

```
$ for_each -test * : mrconvert IN PRE.mif
```

By specifying the `-test` option, the script will print to the terminal the results of text substitutions for all of the specified inputs, but will not actually execute those commands. It can therefore be used to verify that the script is receiving the intended set of inputs, and that the text substitutions on those inputs lead to the intended command strings.

Options

- **-exclude “regex”** (*multiple uses permitted*) Exclude one specific input string / all strings matching a regular expression from being processed (see Example Usage)
- **-test** Test the operation of the `for_each` script, by printing the command strings following string substitution but not actually executing them

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.

- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.61 label2colour

Synopsis

Convert a parcellated image (where values are node indices) into a colour image

Usage

```
label2colour [ options ] nodes_in colour_out
```

- *nodes_in*: the input node parcellation image
- *colour_out*: the output colour image

Description

Many software packages handle this colouring internally within their viewer program; this binary explicitly converts a parcellation image into a colour image that should be viewable in any software.

Options

- **-lut file** Provide the relevant colour lookup table (if not provided, nodes will be coloured randomly)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.62 label2mesh

Synopsis

Generate meshes from a label image

Usage

```
label2mesh [ options ] nodes_in mesh_out
```

- *nodes_in*: the input node parcellation image
- *mesh_out*: the output mesh file

Options

- **-blocky** generate ‘blocky’ meshes with precise delineation of voxel edges, rather than the default Marching Cubes approach

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.63 labelconvert

Synopsis

Convert a connectome node image from one lookup table to another

Usage

```
labelconvert [ options ] path_in lut_in lut_out image_out
```

- *path_in*: the input image
- *lut_in*: the connectome lookup table corresponding to the input image
- *lut_out*: the target connectome lookup table for the output image
- *image_out*: the output image

Description

Typical usage is to convert a parcellation image provided by some other software, based on the lookup table provided by that software, to conform to a new lookup table, particularly one where the node indices increment from 1, in preparation for connectome construction; examples of such target lookup table files are provided in `share//mrtrix3//labelconvert//`, but can be created by the user to provide the desired node set // ordering // colours.

Example usages

- *Convert a Desikan-Killiany parcellation image as provided by FreeSurfer to have nodes incrementing from 1:*

```
$ labelconvert aparc+aseg.mgz FreeSurferColorLUT.txt mrtrix3//share//mrtrix3//
↪labelconvert//fs_default.txt nodes.mif
```

Paths to the files in the example above would need to be revised according to their locations on the user's system.

Options

- **-spine image** provide a manually-defined segmentation of the base of the spine where the streamlines terminate, so that this can become a node in the connection matrix.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).

- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.64 labelsgmfix

Synopsis

In a FreeSurfer parcellation image, replace the sub-cortical grey matter structure delineations using FSL FIRST

Usage

```
labelsgmfix parc t1 lut output [ options ]
```

- *parc*: The input FreeSurfer parcellation image
- *t1*: The T1 image to be provided to FIRST
- *lut*: The lookup table file that the parcellated image is based on
- *output*: The output parcellation image

Options

- **-premasked** Indicate that brain masking has been applied to the T1 input image
- **-sgm_amyg_hipp** Consider the amygdalae and hippocampi as sub-cortical grey matter structures, and also replace their estimates with those from FIRST

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- Patenaude, B.; Smith, S. M.; Kennedy, D. N. & Jenkinson, M. A Bayesian model of shape and appearance for subcortical brain segmentation. *NeuroImage*, 2011, 56, 907-922
- Smith, S. M.; Jenkinson, M.; Woolrich, M. W.; Beckmann, C. F.; Behrens, T. E.; Johansen-Berg, H.; Bannister, P. R.; De Luca, M.; Drobnjak, I.; Flitney, D. E.; Niazy, R. K.; Saunders, J.; Vickers, J.; Zhang, Y.; De Stefano, N.; Brady, J. M. & Matthews, P. M. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 2004, 23, S208-S219
- Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. The effects of SIFT on the reproducibility and biological accuracy of the structural connectome. *NeuroImage*, 2015, 104, 253-265

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.65 labelstats

Synopsis

Compute statistics of parcels within a label image

Usage

```
labelstats [ options ] input
```

- *input*: the input label image

Options

- **-output choice** output only the field specified; options are: mass,centre
- **-voxelspace** report parcel centres of mass in voxel space rather than scanner space

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.66 maskdump

Synopsis

Print out the locations of all non-zero voxels in a mask image

Usage

```
maskdump [ options ] input[ output ]
```

- *input*: the input image.
- *output*: the (optional) output text file.

Description

If no destination file is specified, the voxel locations will be printed to stdout.

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.67 maskfilter

Synopsis

Perform filtering operations on 3D / 4D mask images

Usage

```
maskfilter [ options ] input filter output
```

- *input*: the input image.
- *filter*: the type of filter to be applied (clean, connect, dilate, erode, median)
- *output*: the output image.

Description

The available filters are: clean, connect, dilate, erode, median.

Each filter has its own unique set of optional parameters.

Options

Options for mask cleaning filter

- **-scale value** the maximum scale used to cut bridges. A certain maximum scale cuts bridges up to a width (in voxels) of 2x the provided scale. (Default: 2)

Options for connected-component filter

- **-axes axes** specify which axes should be included in the connected components. By default only the first 3 axes are included. The axes should be provided as a comma-separated list of values.
- **-largest** only retain the largest connected component
- **-connectivity** use 26-voxel-neighbourhood connectivity (Default: 6)

Options for dilate / erode filters

- **-npass value** the number of times to repeatedly apply the filter

Options for median filter

- **-extent voxels** specify the extent (width) of kernel size in voxels. This can be specified either as a single value to be used for all axes, or as a comma-separated list of the extent for each axis. The default is 3x3x3.

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au), David Raffelt (david.raffelt@florey.edu.au), Thijs Dhollander (thijs.dhollander@gmail.com) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.68 mesh2voxel

Synopsis

Convert a mesh surface to a partial volume estimation image

Usage

```
mesh2voxel [ options ] source template output
```

- *source*: the mesh file; note vertices must be defined in realspace coordinates
- *template*: the template image
- *output*: the output image

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 2012, 62, 1924-1938

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.69 meshconvert

Synopsis

Convert meshes between different formats, and apply transformations

Usage

```
meshconvert [ options ] input output
```

- *input*: the input mesh file
- *output*: the output mesh file

Options

- **-binary** write the output mesh file in binary format (if supported)
- **-transform mode image** transform vertices from one coordinate space to another, based on a template image; options are: first2real, real2first, voxel2real, real2voxel, fs2real

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.70 meshfilter

Synopsis

Apply filter operations to meshes

Usage

```
meshfilter [ options ] input filter output
```

- *input*: the input mesh file
- *filter*: the filter to apply. Options are: smooth
- *output*: the output mesh file

Description

While this command has only one filter operation currently available, it nevertheless presents with a comparable interface to the MRtrix3 commands maskfilter and mrfilter commands.

Example usages

- *Apply a mesh smoothing filter (currently the only filter available):*

```
$ meshfilter input.vtk smooth output.vtk
```

The usage of this command may cause confusion due to the generic interface despite only one filtering operation being currently available. This simple example usage is therefore provided for clarity.

Options

Options for mesh smoothing filter

- **-smooth_spatial value** spatial extent of smoothing (default: 10mm)
- **-smooth_influence value** influence factor for smoothing (default: 10)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.71 mraverageheader

Synopsis

Calculate the average (unbiased) coordinate space of all input images

Usage

```
mraverageheader [ options ] input [ input ... ] output
```

- *input*: the input image(s).
- *output*: the output image

Options

- **-padding value** boundary box padding in voxels. Default: 0
- **-resolution type** subsampling of template compared to smallest voxel size in any input image. Valid options are ‘mean’: unbiased but loss of resolution for individual images possible, and ‘max’: smallest voxel size of any input image defines the resolution. Default: mean
- **-fill** set the intensity in the first volume of the average space to 1

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Maximilian Pietsch (maximilian.pietsch@kcl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.72 mrcalc

Synopsis

Apply generic voxel-wise mathematical operations to images

Usage

```
mrcalc [ options ] operand [ operand ... ]
```

- *operand*: an input image, intensity value, or the special keywords ‘rand’ (random number between 0 and 1) or ‘randn’ (random number from unit std.dev. normal distribution) or the mathematical constants ‘e’ and ‘pi’.

Description

This command will only compute per-voxel operations. Use ‘mrmath’ to compute summary statistics across images or along image axes.

This command uses a stack-based syntax, with operators (specified using options) operating on the top-most entries (i.e. images or values) in the stack. Operands (values or images) are pushed onto the stack in the order they appear (as arguments) on the command-line, and operators (specified as options) operate on and consume the top-most entries in the stack, and push their output as a new entry on the stack.

As an additional feature, this command will allow images with different dimensions to be processed, provided they satisfy the following conditions: for each axis, the dimensions match if they are the same size, or one of them has size one. In the latter case, the entire image will be replicated along that axis. This allows for example a 4D image of size [X Y Z N] to be added to a 3D image of size [X Y Z], as if it consisted of N copies of the 3D image along the 4th axis (the missing dimension is assumed to have size 1). Another example would a single-voxel 4D image of size [1 1 1 N], multiplied by a 3D image of size [X Y Z], which would allow the creation of a 4D image where each volume consists of the 3D image scaled by the corresponding value for that volume in the single-voxel image.

Example usages

- *Double the value stored in every voxel:*

```
$ mrcalc a.mif 2 -mult r.mif
```

This performs the operation: $r = 2 * a$ for every voxel a,r in images a.mif and r.mif respectively.

- *A more complex example:*

```
$ mrcalc a.mif -neg b.mif -div -exp 9.3 -mult r.mif
```

This performs the operation: $r = 9.3 * \exp(-a/b)$

- *Another complex example:*

```
$ mrcalc a.mif b.mif -add c.mif d.mif -mult 4.2 -add -div r.mif
```

This performs: $r = (a+b)/(c*d+4.2)$.

- *Rescale the densities in a SH l=0 image:*

```
$ mrcalc ODF_CSF.mif 4 pi -mult -sqrt -div ODF_CSF_scaled.mif
```

This applies the spherical harmonic basis scaling factor: $1.0/\sqrt{4\pi}$, such that a single-tissue voxel containing the same intensities as the response function of that tissue should contain the value 1.0.

Options

basic operations

- **-abs** (*multiple uses permitted*) $| \%1 |$: return absolute value (magnitude) of real or complex number
- **-neg** (*multiple uses permitted*) $\neg \%1$: negative value
- **-add** (*multiple uses permitted*) $(\%1 + \%2)$: add values
- **-subtract** (*multiple uses permitted*) $(\%1 - \%2)$: subtract nth operand from (n-1)th
- **-multiply** (*multiple uses permitted*) $(\%1 * \%2)$: multiply values
- **-divide** (*multiple uses permitted*) $(\%1 / \%2)$: divide (n-1)th operand by nth
- **-min** (*multiple uses permitted*) $\min (\%1, \%2)$: smallest of last two operands
- **-max** (*multiple uses permitted*) $\max (\%1, \%2)$: greatest of last two operands

comparison operators

- **-lt** (*multiple uses permitted*) $(\%1 < \%2)$: less-than operator (true=1, false=0)
- **-gt** (*multiple uses permitted*) $(\%1 > \%2)$: greater-than operator (true=1, false=0)
- **-le** (*multiple uses permitted*) $(\%1 \leq \%2)$: less-than-or-equal-to operator (true=1, false=0)
- **-ge** (*multiple uses permitted*) $(\%1 \geq \%2)$: greater-than-or-equal-to operator (true=1, false=0)
- **-eq** (*multiple uses permitted*) $(\%1 == \%2)$: equal-to operator (true=1, false=0)
- **-neq** (*multiple uses permitted*) $(\%1 != \%2)$: not-equal-to operator (true=1, false=0)

conditional operators

- **-if** (*multiple uses permitted*) $(\%1 ? \%2 : \%3)$: if first operand is true (non-zero), return second operand, otherwise return third operand
- **-replace** (*multiple uses permitted*) $(\%1, \%2 \rightarrow \%3)$: Wherever first operand is equal to the second operand, replace with third operand

power functions

- **-sqrt** (*multiple uses permitted*) $\sqrt{ \%1 }$: square root
- **-pow** (*multiple uses permitted*) $\%1^{\%2}$: raise (n-1)th operand to nth power

nearest integer operations

- **-round** (*multiple uses permitted*) round (%1) : round to nearest integer
- **-ceil** (*multiple uses permitted*) ceil (%1) : round up to nearest integer
- **-floor** (*multiple uses permitted*) floor (%1) : round down to nearest integer

logical operators

- **-not** (*multiple uses permitted*) !%1 : NOT operator: true (1) if operand is false (i.e. zero)
- **-and** (*multiple uses permitted*) (%1 && %2) : AND operator: true (1) if both operands are true (i.e. non-zero)
- **-or** (*multiple uses permitted*) (%1 || %2) : OR operator: true (1) if either operand is true (i.e. non-zero)
- **-xor** (*multiple uses permitted*) (%1 ^^ %2) : XOR operator: true (1) if only one of the operands is true (i.e. non-zero)

classification functions

- **-isnan** (*multiple uses permitted*) isnan (%1) : true (1) if operand is not-a-number (NaN)
- **-isinf** (*multiple uses permitted*) isinf (%1) : true (1) if operand is infinite (Inf)
- **-finite** (*multiple uses permitted*) finite (%1) : true (1) if operand is finite (i.e. not NaN or Inf)

complex numbers

- **-complex** (*multiple uses permitted*) (%1 + %2 i) : create complex number using the last two operands as real,imaginary components
- **-polar** (*multiple uses permitted*) (%1 /_ %2) : create complex number using the last two operands as magnitude,phase components (phase in radians)
- **-real** (*multiple uses permitted*) real (%1) : real part of complex number
- **-imag** (*multiple uses permitted*) imag (%1) : imaginary part of complex number
- **-phase** (*multiple uses permitted*) phase (%1) : phase of complex number (use -abs for magnitude)
- **-conj** (*multiple uses permitted*) conj (%1) : complex conjugate
- **-proj** (*multiple uses permitted*) proj (%1) : projection onto the Riemann sphere

exponential functions

- **-exp** (*multiple uses permitted*) exp (%1) : exponential function
- **-log** (*multiple uses permitted*) log (%1) : natural logarithm
- **-log10** (*multiple uses permitted*) log10 (%1) : common logarithm

trigonometric functions

- **-cos** (*multiple uses permitted*) cos (%1) : cosine
- **-sin** (*multiple uses permitted*) sin (%1) : sine
- **-tan** (*multiple uses permitted*) tan (%1) : tangent
- **-acos** (*multiple uses permitted*) acos (%1) : inverse cosine
- **-asin** (*multiple uses permitted*) asin (%1) : inverse sine
- **-atan** (*multiple uses permitted*) atan (%1) : inverse tangent

hyperbolic functions

- **-cosh** (*multiple uses permitted*) cosh (%1) : hyperbolic cosine
- **-sinh** (*multiple uses permitted*) sinh (%1) : hyperbolic sine
- **-tanh** (*multiple uses permitted*) tanh (%1) : hyperbolic tangent
- **-acosh** (*multiple uses permitted*) acosh (%1) : inverse hyperbolic cosine
- **-asinh** (*multiple uses permitted*) asinh (%1) : inverse hyperbolic sine
- **-atanh** (*multiple uses permitted*) atanh (%1) : inverse hyperbolic tangent

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.73 mrcat

Synopsis

Concatenate several images into one

Usage

```
mrcat [ options ] image1 image2 [ image2 ... ] output
```

- *image1*: the first input image.
- *image2*: additional input image(s).
- *output*: the output image.

Example usages

- *Concatenate individual 3D volumes into a single 4D image series:*

```
$ mrcat volume*.mif series.mif
```

The wildcard characters will find all images in the current working directory with names that begin with “volume” and end with “.mif”; the mrcat command will receive these as a list of input file names, from which it will produce a 4D image where the input volumes have been concatenated along axis 3 (the fourth axis; the spatial axes are 0, 1 & 2).

Options

- **-axis axis** specify axis along which concatenation should be performed. By default, the program will use the last non-singleton, non-spatial axis of any of the input images - in other words axis 3 or whichever axis (greater than 3) of the input images has size greater than one.

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.74 mrcentroid

Synopsis

Determine the centre of mass / centre of gravity of an image

Usage

```
mrcentroid [ options ] input
```

- *input*: the input image

Options

- **-mask image** only include voxels within a mask in the calculation
- **-voxelspace** report image centre of mass in voxel space rather than scanner space

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@floreys.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.75 mrcheckerboardmask

Synopsis

Create bitwise checkerboard image

Usage

```
mrcheckerboardmask [ options ] input output
```

- *input*: the input image to be used as a template.
- *output*: the output binary image mask.

Options

- **-tiles value** specify the number of tiles in any direction
- **-invert** invert output binary mask.
- **-nan** use NaN as the output zero value.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Max Pietsch (maximilian.pietsch@kcl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.76 mrclusterstats

Synopsis

Voxel-based analysis using permutation testing and threshold-free cluster enhancement

Usage

```
mrclusterstats [ options ] input design contrast mask output
```

- *input*: a text file containing the file names of the input images, one file per line
- *design*: the design matrix
- *contrast*: the contrast matrix
- *mask*: a mask used to define voxels included in the analysis.
- *output*: the filename prefix for all output.

Description

In some software packages, a column of ones is automatically added to the GLM design matrix; the purpose of this column is to estimate the “global intercept”, which is the predicted value of the observed variable if all explanatory variables were to be zero. However there are rare situations where including such a column would not be appropriate for a particular experimental design. Hence, in MRtrix3 statistical inference commands, it is up to the user to determine whether or not this column of ones should be included in their design matrix, and add it explicitly if necessary. The contrast matrix must also reflect the presence of this additional column.

Options

Options relating to shuffling of data for nonparametric statistical inference

- **-notest** don’t perform statistical inference; only output population statistics (effect size, stdev etc)
- **-errors spec** specify nature of errors for shuffling; options are: ee,ise,both (default: ee)
- **-exchange_within file** specify blocks of observations within each of which data may undergo restricted exchange
- **-exchange_whole file** specify blocks of observations that may be exchanged with one another (for independent and symmetric errors, sign-flipping will occur block-wise)
- **-strong** use strong familywise error control across multiple hypotheses
- **-nshuffles number** the number of shuffles (default: 5000)

- **-permutations file** manually define the permutations (relabelling). The input should be a text file defining a $m \times n$ matrix, where each relabelling is defined as a column vector of size m , and the number of columns, n , defines the number of permutations. Can be generated with the `palm_quickperms` function in PALM (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM>). Overrides the `-nshuffles` option.
- **-nonstationarity** perform non-stationarity correction
- **-skew_nonstationarity value** specify the skew parameter for empirical statistic calculation (default for this command is 1)
- **-nshuffles_nonstationarity number** the number of shuffles to use when precomputing the empirical statistic image for non-stationarity correction (default: 5000)
- **-permutations_nonstationarity file** manually define the permutations (relabelling) for computing the empirical statistics for non-stationarity correction. The input should be a text file defining a $m \times n$ matrix, where each relabelling is defined as a column vector of size m , and the number of columns, n , defines the number of permutations. Can be generated with the `palm_quickperms` function in PALM (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM>) Overrides the `-nshuffles_nonstationarity` option.

Options for controlling TFCE behaviour

- **-tfce_dh value** the height increment used in the tfce integration (default: 0.1)
- **-tfce_e value** tfce extent exponent (default: 0.5)
- **-tfce_h value** tfce height exponent (default: 2)

Options related to the General Linear Model (GLM)

- **-variance file** define variance groups for the G-statistic; measurements for which the expected variance is equivalent should contain the same index
- **-ftests path** perform F-tests; input text file should contain, for each F-test, a row containing ones and zeros, where ones indicate the rows of the contrast matrix to be included in the F-test.
- **-fonly** only assess F-tests; do not perform statistical inference on entries in the contrast matrix
- **-column path** (*multiple uses permitted*) add a column to the design matrix corresponding to subject voxel-wise values (note that the contrast matrix must include an additional column for each use of this option); the text file provided via this option should contain a file name for each subject

Additional options for `mrclusterstats`

- **-threshold value** the cluster-forming threshold to use for a standard cluster-based analysis. This disables TFCE, which is the default otherwise.
- **-connectivity** use 26-voxel-neighbourhood connectivity (Default: 6)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.

- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- If not using the **-threshold** command-line option: Smith, S. M. & Nichols, T. E. Threshold-free cluster enhancement: Addressing problems of smoothing, threshold dependence and localisation in cluster inference. *NeuroImage*, 2009, 44, 83-98
- If using the **-nonstationary** option: Salimi-Khorshidi, G. Smith, S.M. Nichols, T.E. Adjusting the effect of nonstationarity in cluster-based and TFCE inference. *Neuroimage*, 2011, 54(3), 2006-19

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.77 mrcolour

Synopsis

Apply a colour map to an image

Usage

```
mrcolour [ options ] input map output
```

- *input*: the input image
- *map*: the colourmap to apply; choices are: gray,hot,cool,jet,inferno,viridis,pet,colour,rgb
- *output*: the output image

Description

Under typical usage, this command will receive as input a 3D greyscale image, and output a 4D image with 3 volumes corresponding to red-green-blue components; other use cases are possible, and are described in more detail below.

By default, the command will automatically determine the maximum and minimum intensities of the input image, and use that information to set the upper and lower bounds of the applied colourmap. This behaviour can be overridden by manually specifying these bounds using the `-upper` and `-lower` options respectively.

Options

- **-upper value** manually set the upper intensity of the colour mapping
- **-lower value** manually set the lower intensity of the colour mapping
- **-colour values** set the target colour for use of the ‘colour’ map (three comma-separated floating-point values)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.78 mrconvert

Synopsis

Perform conversion between different file types and optionally extract a subset of the input image

Usage

```
mrconvert [ options ] input output
```

- *input*: the input image.
- *output*: the output image.

Description

If used correctly, this program can be a very useful workhorse. In addition to converting images between different formats, it can be used to extract specific studies from a data set, extract a specific region of interest, or flip the images. Some of the possible operations are described in more detail below.

Note that for both the `-coord` and `-axes` options, indexing starts from 0 rather than 1. E.g. `-coord 3 <#>` selects volumes (the fourth dimension) from the series; `-axes 0,1,2` includes only the three spatial axes in the output image.

Additionally, for the second input to the `-coord` option and the `-axes` option, you can use any valid number sequence in the selection, as well as the ‘end’ keyword (see the main documentation for details); this can be particularly useful to select multiple coordinates.

The `-vox` option is used to change the size of the voxels in the output image as reported in the image header; note however that this does not re-sample the image based on a new voxel size (that is done using the `mrgrid` command).

By default, the intensity scaling parameters in the input image header are passed through to the output image header when writing to an integer image, and reset to 0,1 (i.e. no scaling) for floating-point and binary images. Note that the `-scaling` option will therefore have no effect for floating-point or binary output images.

The `-axes` option specifies which axes from the input image will be used to form the output image. This allows the permutation, omission, or addition of axes into the output image. The axes should be supplied as a comma-separated list of axis indices. If an axis from the input image is to be omitted from the output image, it must either already have a size of 1, or a single coordinate along that axis must be selected by the user by using the `-coord` option. Examples are provided further below.

The `-bvalue_scaling` option controls an aspect of the import of diffusion gradient tables. When the input diffusion-weighting direction vectors have norms that differ substantially from unity, the b-values will be scaled by the square of their corresponding vector norm (this is how multi-shell acquisitions are frequently achieved on scanner platforms). However in some rare instances, the b-values may be correct, despite the vectors not being of unit norm (or conversely, the b-values may need to be rescaled even though the vectors are close to unit norm). This option allows the user to control this operation and override MRtrix3’s automatic detection.

Example usages

- *Extract the first volume from a 4D image, and make the output a 3D image:*

```
$ mrconvert in.mif -coord 3 0 -axes 0,1,2 out.mif
```

The `-coord 3 0` option extracts, from axis number 3 (which is the fourth axis since counting begins from 0; this is the axis that steps across image volumes), only coordinate number 0 (i.e. the first volume). The `-axes 0,1,2` ensures that only the first three axes (i.e. the spatial axes) are retained; if this option were not used in this example, then image `out.mif` would be a 4D image, but it would only consist of a single volume, and `mrinfo` would report its size along the fourth axis as 1.

- *Extract slice number 24 along the AP direction:*

```
$ mrconvert volume.mif slice.mif -coord 1 24
```

MRtrix3 uses a RAS (Right-Anterior-Superior) axis convention, and internally reorients images upon loading in order to conform to this as far as possible. So for non-exotic data, axis 1 should correspond (approximately) to the anterior-posterior direction.

- *Extract only every other volume from a 4D image:*

```
$ mrconvert all.mif every_other.mif -coord 3 1:2:end
```

This example demonstrates two features: Use of the colon syntax to conveniently specify a number sequence (in the format ‘start:step:stop’); and use of the ‘end’ keyword to generate this sequence up to the size of the input image along that axis (i.e. the number of volumes).

- *Alter the image header to report a new isotropic voxel size:*

```
$ mrconvert in.mif isotropic.mif -vox 1.25
```

By providing a single value to the `-vox` option only, the specified value is used to set the voxel size in mm for all three spatial axes in the output image.

- *Alter the image header to report a new anisotropic voxel size:*

```
$ mrconvert in.mif anisotropic.mif -vox 1,,3.5
```

This example will change the reported voxel size along the first and third axes (ideally left-right and inferior-superior) to 1.0mm and 3.5mm respectively, and leave the voxel size along the second axis (ideally anterior-posterior) unchanged.

- *Turn a single-volume 4D image into a 3D image:*

```
$ mrconvert 4D.mif 3D.mif -axes 0,1,2
```

Sometimes in the process of extracting or calculating a single 3D volume from a 4D image series, the size of the image reported by `mrinfo` will be “X x Y x Z x 1”, indicating that the resulting image is in fact also 4D, it just happens to contain only one volume. This example demonstrates how to convert this into a genuine 3D image (i.e. `mrinfo` will report the size as “X x Y x Z”).

- *Insert an axis of size 1 into the image:*

```
$ mrconvert XYZD.mif XYZ1D.mif -axes 0,1,2,-1,3
```

This example uses the value -1 as a flag to indicate to `mrconvert` where a new axis of unity size is to be inserted. In this particular example, the input image has four axes: the spatial axes X, Y and Z, and some form of data D is stored across the fourth axis (i.e. volumes). Due to insertion of a new axis, the output image is 5D: the three spatial axes (XYZ), a single volume (the size of the output image along the fourth axis will be 1), and data D will be stored as volume groups along the fifth axis of the image.

- *Manually reset the data scaling parameters stored within the image header to defaults:*


```
$ mrconvert with_scaling.mif without_scaling.mif -scaling 0.0,1.0
```

This command-line option alters the parameters stored within the image header that provide a linear mapping from raw intensity values stored in the image data to some other scale. Where the raw data stored in a particular voxel is I , the value within that voxel is interpreted as: $\text{value} = \text{offset} + (\text{scale} \times I)$. To adjust this scaling, the relevant parameters must be provided as a comma-separated 2-vector of floating-point values, in the format “offset,scale” (no quotation marks). This particular example sets the offset to zero and the scale to one, which equates to no rescaling of the raw intensity data.

Options

Options for manipulating fundamental image properties

- **-coord axis selection** (*multiple uses permitted*) retain data from the input image only at the coordinates specified in the selection along the specified axis. The selection argument expects a number sequence, which can also include the ‘end’ keyword.
- **-vox sizes** change the voxel dimensions reported in the output image header
- **-axes axes** specify the axes from the input image that will be used to form the output image
- **-scaling values** specify the data scaling parameters used to rescale the intensity values

Options for handling JSON (JavaScript Object Notation) files

- **-json_import file** import data from a JSON file into header key-value pairs
- **-json_export file** export data from an image header key-value pairs into a JSON file

Options to modify generic header entries

- **-clear_property key** (*multiple uses permitted*) remove the specified key from the image header altogether.
- **-set_property key value** (*multiple uses permitted*) set the value of the specified key in the image header.
- **-append_property key value** (*multiple uses permitted*) append the given value to the specified key in the image header (this adds the value specified as a new line in the header value).
- **-copy_properties source** clear all generic properties and replace with the properties from the image / file specified.

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be,

uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-bvalue_scaling mode** enable or disable scaling of diffusion b-values by the square of the corresponding DW gradient norm (see Description). Valid choices are yes/no, true/false, 0/1 (default: automatic).

DW gradient table export options

- **-export_grad_mrtrix path** export the diffusion-weighted gradient table to file in MRtrix format
- **-export_grad_fsl bvecs_path bvals_path** export the diffusion-weighted gradient table to files in FSL (bvecs / bvals) format

Options for importing phase-encode tables

- **-import_pe_table file** import a phase-encoding table from file
- **-import_pe_eddy config indices** import phase-encoding information from an EDDY-style config / index file pair

Options for exporting phase-encode tables

- **-export_pe_table file** export phase-encoding table to file
- **-export_pe_eddy config indices** export phase-encoding information to an EDDY-style config / index file pair

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.

- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.79 mrdegibbs

Synopsis

Remove Gibbs Ringing Artifacts

Usage

```
mrdegibbs [ options ] in out
```

- *in*: the input image.
- *out*: the output image.

Description

This application attempts to remove Gibbs ringing artefacts from MRI images using the method of local subvoxel-shifts proposed by Kellner et al. (see reference below for details).

This command is designed to run on data directly after it has been reconstructed by the scanner, before any interpolation of any kind has taken place. You should not run this command after any form of motion correction (e.g. not after `dwifslpreproc`). Similarly, if you intend running `dwidenoise`, you should run denoising before this command to not alter the noise structure, which would impact on `dwidenoise`'s performance.

Note that this method is designed to work on images acquired with full k-space coverage. Running this method on partial Fourier ('half-scan') data may lead to suboptimal and/or biased results, as noted in the original reference below. There is currently no means of dealing with this; users should exercise caution when using this method on partial Fourier data, and inspect its output for any obvious artefacts.

Options

- **-axes list** select the slice axes (default: 0,1 - i.e. x-y).
- **-nshifts value** discretization of subpixel spacing (default: 20).
- **-minW value** left border of window used for TV computation (default: 1).
- **-maxW value** right border of window used for TV computation (default: 3).

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Kellner, E; Dhital, B; Kiselev, V.G & Reisert, M. Gibbs-ringing artifact removal based on local subvoxel-shifts. Magnetic Resonance in Medicine, 2016, 76, 1574–1581.

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Ben Jeurissen (ben.jeurissen@uantwerpen.be) & J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.80 mrdump

Synopsis

Print out the values within an image

Usage

```
mrdump [ options ] input[ output ]
```

- *input*: the input image.
- *output*: the (optional) output text file.

Description

If no destination file is specified, the voxel locations will be printed to stdout.

Options

- **-mask image** only write the image values within voxels specified by a mask image

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.81 mredit

Synopsis

Directly edit the intensities within an image from the command-line

Usage

```
mredit [ options ] input[ output ]
```

- *input*: the input image
- *output*: the (optional) output image

Description

A range of options are provided to enable direct editing of voxel intensities based on voxel / real-space coordinates. If only one image path is provided, the image will be edited in-place (use at own risk); if input and output image paths are provided, the output will contain the edited image, and the original image will not be modified in any way.

Options

- **-plane axis coord value** (*multiple uses permitted*) fill one or more planes on a particular image axis
- **-sphere position radius value** (*multiple uses permitted*) draw a sphere with radius in mm
- **-voxel position value** (*multiple uses permitted*) change the image value within a single voxel
- **-scanner** indicate that coordinates are specified in scanner space, rather than as voxel coordinates

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).

- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.82 mrfilter

Synopsis

Perform filtering operations on 3D / 4D MR images

Usage

```
mrfilter [ options ]  input filter output
```

- *input*: the input image.
- *filter*: the type of filter to be applied
- *output*: the output image.

Description

The available filters are: fft, gradient, median, smooth, normalise, zclean.

Each filter has its own unique set of optional parameters.

For 4D images, each 3D volume is processed independently.

Options

Options for FFT filter

- **-axes list** the axes along which to apply the Fourier Transform. By default, the transform is applied along the three spatial axes. Provide as a comma-separated list of axis indices.
- **-inverse** apply the inverse FFT
- **-magnitude** output a magnitude image rather than a complex-valued image
- **-centre_zero** re-arrange the FFT results so that the zero-frequency component appears in the centre of the image, rather than at the edges

Options for gradient filter

- **-stdev sigma** the standard deviation of the Gaussian kernel used to smooth the input image (in mm). The image is smoothed to reduced large spurious gradients caused by noise. Use this option to override the default stdev of 1 voxel. This can be specified either as a single value to be used for all 3 axes, or as a comma-separated list of 3 values, one for each axis.
- **-magnitude** output the gradient magnitude, rather than the default x,y,z components
- **-scanner** define the gradient with respect to the scanner coordinate frame of reference.

Options for median filter

- **-extent size** specify extent of median filtering neighbourhood in voxels. This can be specified either as a single value to be used for all 3 axes, or as a comma-separated list of 3 values, one for each axis (default: 3x3x3).

Options for normalisation filter

- **-extent size** specify extent of normalisation filtering neighbourhood in voxels. This can be specified either as a single value to be used for all 3 axes, or as a comma-separated list of 3 values, one for each axis (default: 3x3x3).

Options for smooth filter

- **-stdev mm** apply Gaussian smoothing with the specified standard deviation. The standard deviation is defined in mm (Default 1 voxel). This can be specified either as a single value to be used for all axes, or as a comma-separated list of the stdev for each axis.
- **-fwhm mm** apply Gaussian smoothing with the specified full-width half maximum. The FWHM is defined in mm (Default 1 voxel * 2.3548). This can be specified either as a single value to be used for all axes, or as a comma-separated list of the FWHM for each axis.
- **-extent voxels** specify the extent (width) of kernel size in voxels. This can be specified either as a single value to be used for all axes, or as a comma-separated list of the extent for each axis. The default extent is $2 * \text{ceil}(2.5 * \text{stdev} / \text{voxel_size}) - 1$.

Options for zclean filter

- **-zupper num** define high intensity outliers: default: 2.5
- **-zlower num** define low intensity outliers: default: 2.5
- **-bridge num** number of voxels to gap to fill holes in mask: default: 4
- **-maskin image** initial mask that defines the maximum spatial extent and the region from which to sample the intensity range.
- **-maskout image** Output a refined mask based on a spatially coherent region with normal intensity range.

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au), David Raffelt (david.raffelt@florey.edu.au) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.83 mrgrid

Synopsis

Modify the grid of an image without interpolation (cropping or padding) or by regridding to an image grid with modified orientation, location and or resolution. The image content remains in place in real world coordinates.

Usage

```
mrgrid [ options ] input operation output
```

- *input*: input image to be regridded.
- *operation*: the operation to be performed, one of: regrid, crop, pad.
- *output*: the output image.

Description

- **regrid**: This operation performs changes of the voxel grid that require interpolation of the image such as changing the resolution or location and orientation of the voxel grid. If the image is down-sampled, the appropriate smoothing is automatically applied using Gaussian smoothing unless nearest neighbour interpolation is selected or oversample is changed explicitly. The resolution can only be changed for spatial dimensions.
- **crop**: The image extent after cropping, can be specified either manually for each axis dimensions, or via a mask or reference image. The image can be cropped to the extent of a mask. This is useful for axially-acquired brain images, where the image size can be reduced by a factor of 2 by removing the empty space on either side of the brain. Note that cropping does not extend the image beyond the original FOV unless explicitly specified (via `-crop_unbound` or negative `-axis` extent).
- **pad**: Analogously to cropping, padding increases the FOV of an image without image interpolation. Pad and crop can be performed simultaneously by specifying signed specifier argument values to the `-axis` option.

This command encapsulates and extends the functionality of the superseded commands ‘`mrpad`’, ‘`mrcrop`’ and ‘`mrresize`’. Note the difference in `-axis` convention used for ‘`mrcrop`’ and ‘`mrpad`’ (see `-axis` option description).

Example usages

- *Crop and pad the first axis:*

```
$ mrgrid in.mif crop -axis 0 10,-5 out.mif
```

This removes 10 voxels on the lower and pads with 5 on the upper bound, which is equivalent to padding with the negated specifier (`mrgrid in.mif pad -axis 0 -10,5 out.mif`).

- *Right-pad the image to the number of voxels of a reference image:*

```
$ mrgrid in.mif pad -as ref.mif -all_axes -axis 3 0,0 out.mif -fill nan
```

This pads the image on the upper bound of all axes except for the volume dimension. The headers of `in.mif` and `ref.mif` are ignored and the output image uses NAN values to fill in voxels outside the original range of `in.mif`.

- *Regrid and interpolate to match the voxel grid of a reference image:*

```
$ mrgrid in.mif regrid -template ref.mif -scale 1,1,0.5 out.mif -fill nan
```

The `-template` instructs to regrid `in.mif` to match the voxel grid of `ref.mif` (voxel size, grid orientation and voxel centres). The `-scale` option overwrites the voxel scaling factor yielding voxel sizes in the third dimension that are twice as coarse as those of the template image.

Options

Regridding options (involves image interpolation, applied to spatial axes only)

- **-template image** match the input image grid (voxel spacing, image size, header transformation) to that of a reference image. The image resolution relative to the template image can be changed with one of `-size`, `-voxel`, `-scale`.
- **-size dims** define the size (number of voxels) in each spatial dimension for the output image. This should be specified as a comma-separated list.
- **-voxel size** define the new voxel size for the output image. This can be specified either as a single value to be used for all spatial dimensions, or as a comma-separated list of the size for each voxel dimension.
- **-scale factor** scale the image resolution by the supplied factor. This can be specified either as a single value to be used for all dimensions, or as a comma-separated list of scale factors for each dimension.
- **-interp method** set the interpolation method to use when reslicing (choices: nearest, linear, cubic, sinc. Default: cubic).
- **-oversample factor** set the amount of over-sampling (in the target space) to perform when regridding. This is particularly relevant when downsampling a high-resolution image to a low-resolution image, to avoid aliasing artefacts. This can consist of a single integer, or a comma-separated list of 3 integers if different oversampling factors are desired along the different axes. Default is determined from ratio of voxel dimensions (disabled for nearest-neighbour interpolation).

Pad and crop options (no image interpolation is performed, header transformation is adjusted)

- **-as reference image** pad or crop the input image on the upper bound to match the specified reference image grid. This operation ignores differences in image transformation between input and reference image.
- **-uniform number** pad or crop the input image by a uniform number of voxels on all sides
- **-mask image** crop the input image according to the spatial extent of a mask image. The mask must share a common voxel grid with the input image but differences in image transformations are ignored. Note that even though only 3 dimensions are cropped when using a mask, the bounds are computed by checking the extent for all dimensions. Note that by default a gap of 1 voxel is left at all edges of the image to allow valid trilinear interpolation. This gap can be modified with the `-uniform` option but by default it does not extend beyond the FOV unless `-crop_unbound` is used.
- **-crop_unbound** Allow padding beyond the original FOV when cropping.
- **-axis index spec** (*multiple uses permitted*) pad or crop the input image along the provided axis (defined by index). The specifier argument defines the number of voxels added or removed on the lower or upper end of the axis (`-axis index delta_lower,delta_upper`) or acts as a voxel selection range (`-axis index start:stop`). In both modes, values are relative to the input image (overriding all other extent-specifying options). Negative delta specifier values trigger the inverse operation (pad instead of crop and vice versa) and negative range specifier trigger padding. Note that the deprecated commands `'mrcrop'` and `'mrpad'` used range-based and delta-based -axis indices, respectively.

- **-all_axes** Crop or pad all, not just spatial axes.

General options

- **-fill number** Use number as the out of bounds value. nan, inf and -inf are valid arguments. (Default: 0.0)

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Max Pietsch (maximilian.pietsch@kcl.ac.uk) & David Raffelt (david.raffelt@florey.edu.au) & Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.84 mrhistmatch

Synopsis

Modify the intensities of one image to match the histogram of another

Usage

```
mrhistmatch [ options ] type input target output
```

- *type*: type of histogram matching to perform; options are: scale,linear,nonlinear
- *input*: the input image to be modified
- *target*: the input image from which to derive the target histogram
- *output*: the output image

Options

Image masking options

- **-mask_input image** only generate input histogram based on a specified binary mask image
- **-mask_target image** only generate target histogram based on a specified binary mask image

Non-linear histogram matching options

- **-bins num** the number of bins to use to generate the histograms

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).

- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- If using inverse contrast normalization for inter-modal (DWI - T1) registration: Bhushan, C.; Haldar, J. P.; Choi, S.; Joshi, A. A.; Shattuck, D. W. & Leahy, R. M. Co-registration and distortion correction of diffusion and anatomical images based on inverse contrast normalization. *NeuroImage*, 2015, 115, 269-280
- Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137
-

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.85 mrhistogram

Synopsis

Generate a histogram of image intensities

Usage

```
mrhistogram [ options ] image hist
```

- *image*: the input image from which the histogram will be computed
- *hist*: the output histogram file

Options

Histogram generation options

- **-bins num** Manually set the number of bins to use to generate the histogram.
- **-template file** Use an existing histogram file as the template for histogram formation
- **-mask image** Calculate the histogram only within a mask image.
- **-ignorezero** ignore zero-valued data during histogram construction.

Additional options for mrhistogram

- **-allvolumes** generate one histogram across all image volumes, rather than one per image volume

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.86 mrinfo

Synopsis

Display image header information, or extract specific information from the header

Usage

```
mrinfo [ options ] image [ image ... ]
```

- *image*: the input image(s).

Description

By default, all information contained in each image header will be printed to the console in a reader-friendly format.

Alternatively, command-line options may be used to extract specific details from the header(s); these are printed to the console in a format more appropriate for scripting purposes or piping to file. If multiple options and/or images are provided, the requested header fields will be printed in the order in which they appear in the help page, with all requested details from each input image in sequence printed before the next image is processed.

The command can also write the diffusion gradient table from a single input image to file; either in the MRtrix or FSL format (bvecs/bvals file pair; includes appropriate diffusion gradient vector reorientation)

The `-dwgrad`, `-export_*` and `-shell_*` options provide (information about) the diffusion weighting gradient table after it has been processed by the MRtrix3 back-end (vectors normalised, b-values scaled by the square of the vector norm, depending on the `-bvalue_scaling` option). To see the raw gradient table information as stored in the image header, i.e. without MRtrix3 back-end processing, use “`-property dw_scheme`”.

The `-bvalue_scaling` option controls an aspect of the import of diffusion gradient tables. When the input diffusion-weighting direction vectors have norms that differ substantially from unity, the b-values will be scaled by the square of their corresponding vector norm (this is how multi-shell acquisitions are frequently achieved on scanner platforms). However in some rare instances, the b-values may be correct, despite the vectors not being of unit norm (or conversely, the b-values may need to be rescaled even though the vectors are close to unit norm). This option allows the user to control this operation and override MRtrix3’s automatic detection.

Options

- **-all** print all properties, rather than the first and last 2 of each.
- **-name** print the file system path of the image
- **-format** image file format
- **-ndim** number of image dimensions
- **-size** image size along each axis
- **-spacing** voxel spacing along each image dimension
- **-datatype** data type used for image data storage
- **-strides** data strides i.e. order and direction of axes data layout
- **-offset** image intensity offset
- **-multiplier** image intensity multiplier
- **-transform** the transformation from image coordinates [mm] to scanner / real world coordinates [mm]

Options for exporting image header fields

- **-property key** (*multiple uses permitted*) any text properties embedded in the image header under the specified key (use ‘all’ to list all keys found)
- **-json_keyval file** export header key/value entries to a JSON file
- **-json_all file** export all header contents to a JSON file

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-bvalue_scaling mode** enable or disable scaling of diffusion b-values by the square of the corresponding DW gradient norm (see Description). Valid choices are yes/no, true/false, 0/1 (default: automatic).

DW gradient table export options

- **-export_grad_mrtrix path** export the diffusion-weighted gradient table to file in MRtrix format
- **-export_grad_fsl bvecs_path bvals_path** export the diffusion-weighted gradient table to files in FSL (bvecs / bvals) format
- **-dwgrad** the diffusion-weighting gradient table, as interpreted by MRtrix3
- **-shell_bvalues** list the average b-value of each shell
- **-shell_sizes** list the number of volumes in each shell
- **-shell_indices** list the image volumes attributed to each b-value shell

Options for exporting phase-encode tables

- **-export_pe_table file** export phase-encoding table to file
- **-export_pe_eddy config indices** export phase-encoding information to an EDDY-style config / index file pair
- **-petable** print the phase encoding table

Handling of piped images

- **-nodelete** don't delete temporary images or images passed to mrinfo via Unix pipes

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.

- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (d.tournier@brain.org.au) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.87 mrmath

Synopsis

Compute summary statistic on image intensities either across images, or along a specified axis of a single image

Usage

```
mrmath [ options ] input [ input ... ] operation output
```

- *input*: the input image(s).
- *operation*: the operation to apply, one of: mean, median, sum, product, rms, norm, var, std, min, max, absmax, magmax.
- *output*: the output image.

Description

Supported operations are:

mean, median, sum, product, rms (root-mean-square value), norm (vector 2-norm), var (unbiased variance), std (unbiased standard deviation), min, max, absmax (maximum absolute value), magmax (value with maximum absolute value, preserving its sign).

This command is used to traverse either along an image axis, or across a set of input images, calculating some statistic from the values along each traversal. If you are seeking to instead perform mathematical calculations that are done independently for each voxel, please see the ‘mrcalc’ command.

Example usages

- Calculate a 3D volume representing the mean intensity across a 4D image series:

```
$ mrmath 4D.mif mean 3D_mean.mif -axis 3
```

This is a common operation for calculating e.g. the mean value within a specific DWI b-value. Note that axis indices start from 0; thus, axes 0, 1 & 2 are the three spatial axes, and axis 3 operates across volumes.

- Generate a Maximum Intensity Projection (MIP) along the inferior-superior direction:

```
$ mrmath input.mif max MIP.mif -axis 2
```

Since a MIP is literally the maximal value along a specific projection direction, axis-aligned MIPs can be generated easily using mrmath with the ‘max’ operation.

Options

- **-axis index** perform operation along a specified axis of a single input image
- **-keep_unary_axes** Keep unary axes in input images prior to calculating the stats. The default is to wipe axes with single elements.

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.88 mrmetric

Synopsis

Computes a dissimilarity metric between two images

Usage

```
mrmetric [ options ] image1 image2
```

- *image1*: the first input image.
- *image2*: the second input image.

Description

Currently only the mean squared difference is fully implemented.

Options

- **-space iteration method** voxel (default): per voxel image1: scanner space of image 1 image2: scanner space of image 2 average: scanner space of the average affine transformation of image 1 and 2
- **-interp method** set the interpolation method to use when reslicing (choices: nearest, linear, cubic, sinc. Default: linear).
- **-metric method** define the dissimilarity metric used to calculate the cost. Choices: diff (squared differences), cc (non-normalised negative cross correlation aka negative cross covariance). Default: diff. cc is only implemented for -space average and -interp linear and cubic.
- **-mask1 image** mask for image 1
- **-mask2 image** mask for image 2
- **-nonormalisation** do not normalise the dissimilarity metric to the number of voxels.

- **-overlap** output number of voxels that were used.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) and Max Pietsch (maximilian.pietsch@kcl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.89 mrregister

Synopsis

Register two images together using a symmetric rigid, affine or non-linear transformation model

Usage

```
mrregister [ options ] image1 image2[ contrast1 contrast2 ... ]
```

- *image1 image2*: input image 1 (‘moving’) and input image 2 (‘template’)

- *contrast1 contrast2*: optional list of additional input images used as additional contrasts. Can be used multiple times. contrastX and imageX must share the same coordinate system.

Description

By default this application will perform an affine, followed by non-linear registration.

FOD registration (with apodised point spread reorientation) will be performed by default if the number of volumes in the 4th dimension equals the number of coefficients in an antipodally symmetric spherical harmonic series (e.g. 6, 15, 28 etc). The `-no_reorientation` option can be used to force reorientation off if required.

Non-linear registration computes warps to map from both image1->image2 and image2->image1. Similar to Avants (2008) Med Image Anal. 12(1): 26–41, registration is performed by matching both the image1 and image2 in a ‘midway space’. Warps can be saved as two deformation fields that map directly between image1->image2 and image2->image1, or if using `-nl_warp_full` as a single 5D file that stores all 4 warps image1->mid->image2, and image2->mid->image1. The 5D warp format stores x,y,z deformations in the 4th dimension, and uses the 5th dimension to index the 4 warps. The affine transforms estimated (to midway space) are also stored as comments in the image header. The 5D warp file can be used to reinitialise subsequent registrations, in addition to transforming images to midway space (e.g. for intra-subject alignment in a 2-time-point longitudinal analysis).

Options

- **-type choice** the registration type. Valid choices are: rigid, affine, nonlinear, rigid_affine, rigid_nonlinear, affine_nonlinear, rigid_affine_nonlinear (Default: affine_nonlinear)
- **-transformed image** (*multiple uses permitted*) image1 after registration transformed and regridded to the space of image2. Note that `-transformed` needs to be repeated for each contrast if multi-contrast registration is used.
- **-transformed_midway image1_transformed image2_transformed** (*multiple uses permitted*) image1 and image2 after registration transformed and regridded to the midway space. Note that `-transformed_midway` needs to be repeated for each contrast if multi-contrast registration is used.
- **-mask1 filename** a mask to define the region of image1 to use for optimisation.
- **-mask2 filename** a mask to define the region of image2 to use for optimisation.
- **-nan** use NaN as out of bounds value. (Default: 0.0)

Rigid registration options

- **-rigid file** the output text file containing the rigid transformation as a 4x4 matrix
- **-rigid_1tomidway file** the output text file containing the rigid transformation that aligns image1 to image2 in their common midway space as a 4x4 matrix
- **-rigid_2tomidway file** the output text file containing the rigid transformation that aligns image2 to image1 in their common midway space as a 4x4 matrix
- **-rigid_init_translation type** initialise the translation and centre of rotation Valid choices are: mass (aligns the centers of mass of both images, default), geometric (aligns geometric image centres) and none.
- **-rigid_init_rotation type** initialise the rotation Valid choices are: search (search for the best rotation using mean squared residuals), moments (rotation based on directions of intensity variance with respect to centre of mass), none (default).
- **-rigid_init_matrix file** initialise either the rigid, affine, or syn registration with the supplied rigid transformation (as a 4x4 matrix in scanner coordinates). Note that this overrides rigid_init_translation and rigid_init_rotation initialisation

- **-rigid_scale factor** use a multi-resolution scheme by defining a scale factor for each level using comma separated values (Default: 0.25,0.5,1.0)
- **-rigid_niter num** the maximum number of gradient descent iterations per stage. This can be specified either as a single number for all multi-resolution levels, or a single value for each level. (Default: 1000)
- **-rigid_metric type** valid choices are: diff (intensity differences), Default: diff
- **-rigid_metric.diff.estimator type** Valid choices are: l1 (least absolute: $|x|$), l2 (ordinary least squares), lp (least powers: $|x|^{1.2}$), Default: l2
- **-rigid_lmax num** explicitly set the lmax to be used per scale factor in rigid FOD registration. By default FOD registration will use lmax 0,2,4 with default scale factors 0.25,0.5,1.0 respectively. Note that no reorientation will be performed with lmax = 0.
- **-rigid_log file** write gradient descent parameter evolution to log file

Affine registration options

- **-affine file** the output text file containing the affine transformation as a 4x4 matrix
- **-affine_1tomidway file** the output text file containing the affine transformation that aligns image1 to image2 in their common midway space as a 4x4 matrix
- **-affine_2tomidway file** the output text file containing the affine transformation that aligns image2 to image1 in their common midway space as a 4x4 matrix
- **-affine_init_translation type** initialise the translation and centre of rotation Valid choices are: mass (aligns the centers of mass of both images), geometric (aligns geometric image centres) and none. (Default: mass)
- **-affine_init_rotation type** initialise the rotation Valid choices are: search (search for the best rotation using mean squared residuals), moments (rotation based on directions of intensity variance with respect to centre of mass), none (Default: none).
- **-affine_init_matrix file** initialise either the affine, or syn registration with the supplied affine transformation (as a 4x4 matrix in scanner coordinates). Note that this overrides affine_init_translation and affine_init_rotation initialisation
- **-affine_scale factor** use a multi-resolution scheme by defining a scale factor for each level using comma separated values (Default: 0.25,0.5,1.0)
- **-affine_niter num** the maximum number of gradient descent iterations per stage. This can be specified either as a single number for all multi-resolution levels, or a single value for each level. (Default: 1000)
- **-affine_metric type** valid choices are: diff (intensity differences), Default: diff
- **-affine_metric.diff.estimator type** Valid choices are: l1 (least absolute: $|x|$), l2 (ordinary least squares), lp (least powers: $|x|^{1.2}$), Default: l2
- **-affine_lmax num** explicitly set the lmax to be used per scale factor in affine FOD registration. By default FOD registration will use lmax 0,2,4 with default scale factors 0.25,0.5,1.0 respectively. Note that no reorientation will be performed with lmax = 0.
- **-affine_log file** write gradient descent parameter evolution to log file

Advanced linear transformation initialisation options

- **-init_translation.unmasked1** disregard mask1 for the translation initialisation (affects 'mass')
- **-init_translation.unmasked2** disregard mask2 for the translation initialisation (affects 'mass')

- **-init_rotation.unmasked1** disregard mask1 for the rotation initialisation (affects ‘search’ and ‘moments’)
- **-init_rotation.unmasked2** disregard mask2 for the rotation initialisation (affects ‘search’ and ‘moments’)
- **-init_rotation.search.angles angles** rotation angles for the local search in degrees between 0 and 180. (Default: 2,5,10,15,20)
- **-init_rotation.search.scale scale** relative size of the images used for the rotation search. (Default: 0.15)
- **-init_rotation.search.directions num** number of rotation axis for local search. (Default: 250)
- **-init_rotation.search.run_global** perform a global search. (Default: local)
- **-init_rotation.search.global.iterations num** number of rotations to investigate (Default: 10000)

Advanced linear registration stage options

- **-linstage.iterations num or comma separated list** number of iterations for each registration stage, not to be confused with -rigid_niter or -affine_niter. This can be used to generate intermediate diagnostics images (-linstage.diagnostics.prefix) or to change the cost function optimiser during registration (without the need to repeatedly resize the images). (Default: 1 == no repetition)
- **-linstage.optimiser.first algorithm** Cost function optimisation algorithm to use at first iteration of all stages. Valid choices: bbgd (Barzilai-Borwein gradient descent) or gd (simple gradient descent). (Default: bbgd)
- **-linstage.optimiser.last algorithm** Cost function optimisation algorithm to use at last iteration of all stages (if there are more than one). Valid choices: bbgd (Barzilai-Borwein gradient descent) or gd (simple gradient descent). (Default: bbgd)
- **-linstage.optimiser.default algorithm** Cost function optimisation algorithm to use at any stage iteration other than first or last iteration. Valid choices: bbgd (Barzilai-Borwein gradient descent) or gd (simple gradient descent). (Default: bbgd)
- **-linstage.diagnostics.prefix file prefix** generate diagnostics images after every registration stage

Non-linear registration options

- **-nl_warp warp1 warp2** the non-linear warp output defined as two deformation fields, where warp1 can be used to transform image1->image2 and warp2 to transform image2->image1. The deformation fields also encapsulate any linear transformation estimated prior to non-linear registration.
- **-nl_warp_full image** output all warps used during registration. This saves four different warps that map each image to a midway space and their inverses in a single 5D image file. The 4th image dimension indexes the x,y,z component of the deformation vector and the 5th dimension indexes the field in this order: image1->midway, midway->image1, image2->midway, midway->image2. Where image1->midway defines the field that maps image1 onto the midway space using the reverse convention. When linear registration is performed first, the estimated linear transform will be included in the comments of the image header, and therefore the entire linear and non-linear transform can be applied (in either direction) using this output warp file with mrtransform
- **-nl_init image** initialise the non-linear registration with the supplied warp image. The supplied warp must be in the same format as output using the -nl_warp_full option (i.e. have 4 deformation fields with the linear transforms in the image header)
- **-nl_scale factor** use a multi-resolution scheme by defining a scale factor for each level using comma separated values (Default: 0.25,0.5,1.0)
- **-nl_niter num** the maximum number of iterations. This can be specified either as a single number for all multi-resolution levels, or a single value for each level. (Default: 50)

- **-nl_update_smooth stdev** regularise the gradient update field with Gaussian smoothing (standard deviation in voxel units, Default 2.0)
- **-nl_disp_smooth stdev** regularise the displacement field with Gaussian smoothing (standard deviation in voxel units, Default 1.0)
- **-nl_grad_step num** the gradient step size for non-linear registration (Default: 0.5)
- **-nl_lmax num** explicitly set the lmax to be used per scale factor in non-linear FOD registration. By default FOD registration will use lmax 0,2,4 with default scale factors 0.25,0.5,1.0 respectively. Note that no reorientation will be performed with lmax = 0.
- **-diagnostics_image path** write intermediate images for diagnostics purposes

FOD registration options

- **-directions file** the directions used for FOD reorientation using apodised point spread functions (Default: 60 directions)
- **-noreorientation** turn off FOD reorientation. Reorientation is on by default if the number of volumes in the 4th dimension corresponds to the number of coefficients in an antipodally symmetric spherical harmonic series (i.e. 6, 15, 28, 45, 66 etc)

Multi-contrast options

- **-mc_weights weights** relative weight of images used for multi-contrast registration. Default: 1.0 (equal weighting)

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- If FOD registration is being performed: Raffelt, D.; Tournier, J.-D.; Fripp, J.; Crozier, S.; Connelly, A. & Salvado, O. Symmetric diffeomorphic registration of fibre orientation distributions. *NeuroImage*, 2011, 56(3), 1171-1180
- Raffelt, D.; Tournier, J.-D.; Crozier, S.; Connelly, A. & Salvado, O. Reorientation of fiber orientation distributions using apodized point spread functions. *Magnetic Resonance in Medicine*, 2012, 67, 844-855
- Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137
-

Author: David Raffelt (david.raffelt@florey.edu.au) & Max Pietsch (maximilian.pietsch@kcl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.90 mrstats

Synopsis

Compute images statistics

Usage

```
mrstats [ options ] image
```

- *image*: the input image from which statistics will be computed.

Options

Statistics options

- **-output field** (*multiple uses permitted*) output only the field specified. Multiple such options can be supplied if required. Choices are: mean, median, std, std_rv, min, max, count. Useful for use in scripts. Both std options refer to the unbiased (sample) standard deviation. For complex data, min, max and std are calculated separately for real and imaginary parts, std_rv is based on the real valued variance (equals sqrt of sum of variances of imaginary and real parts).
- **-mask image** only perform computation within the specified binary mask image.
- **-ignorezero** ignore zero values during statistics calculation

Additional options for mrstats

- **-allvolumes** generate statistics across all image volumes, rather than one set of statistics per image volume

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.91 mrthreshold

Synopsis

Create bitwise image by thresholding image intensity

Usage

```
mrthreshold [ options ] input[ output ]
```

- *input*: the input image to be thresholded
- *output*: the (optional) output binary image mask

Description

The threshold value to be applied can be determined in one of a number of ways:

- If no relevant command-line option is used, the command will automatically determine an optimal threshold;
- The `-abs` option provides the threshold value explicitly;
- The `-percentile`, `-top` and `-bottom` options enable more fine-grained control over how the threshold value is determined.

The `-mask` option only influences those image values that contribute toward the determination of the threshold value; once the threshold is determined, it is applied to the entire image, irrespective of use of the `-mask` option. If you wish for the voxels outside of the specified mask to additionally be excluded from the output mask, this can be achieved by providing the `-out_masked` option.

The four operators available through the “-comparison” option (“lt”, “le”, “ge” and “gt”) correspond to “less-than” (<), “less-than-or-equal” (<=), “greater-than-or-equal” (>=) and “greater-than” (>). This offers fine-grained control over how the thresholding operation will behave in the presence of values equivalent to the threshold. By default, the command will select voxels with values greater than or equal to the determined threshold (“ge”); unless the `-bottom` option is used, in which case after a threshold is determined from the relevant lowest-valued image voxels, those voxels with values less than or equal to that threshold (“le”) are selected. This provides more fine-grained control than the `-invert` option; the latter is provided for backwards compatibility, but is equivalent to selection of the opposite comparison within this selection.

If no output image path is specified, the command will instead write to standard output the determined threshold value.

Options

Threshold determination mechanisms

- **-abs value** specify threshold value as absolute intensity
- **-percentile value** determine threshold based on some percentile of the image intensity distribution
- **-top count** determine threshold that will result in selection of some number of top-valued voxels
- **-bottom count** determine & apply threshold resulting in selection of some number of bottom-valued voxels (note: implies threshold application operator of “le” unless otherwise specified)

Threshold determination modifiers

- **-allvolumes** compute a single threshold for all image volumes, rather than an individual threshold per volume
- **-ignorezero** ignore zero-valued input values during threshold determination
- **-mask image** compute the threshold based only on values within an input mask image

Threshold application modifiers

- **-comparison choice** comparison operator to use when applying the threshold; options are: lt,le,ge,gt (default = “le” for -bottom; “ge” otherwise)
- **-invert** invert the output binary mask (equivalent to flipping the operator; provided for backwards compatibility)
- **-out_masked** mask the output image based on the provided input mask image
- **-nan** set voxels that fail the threshold to NaN rather than zero (output image will be floating-point rather than binary)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- If not using any explicit thresholding mechanism: Ridgway, G. R.; Omar, R.; Ourselin, S.; Hill, D. L.; Warren, J. D. & Fox, N. C. Issues with threshold masking in voxel-based morphometry of atrophied brains. *NeuroImage*, 2009, 44, 99-111

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.92 mrtransform

Synopsis

Apply spatial transformations to an image

Usage

```
mrtransform [ options ] input output
```

- *input*: input image to be transformed.
- *output*: the output image.

Description

If a linear transform is applied without a template image the command will modify the image header transform matrix. FOD reorientation (with apodised point spread functions) can be performed if the number of volumes in the 4th dimension equals the number of coefficients in an antipodally symmetric spherical harmonic series (e.g. 6, 15, 28 etc). For such data, the `-reorient_fod yes/no` option must be used to specify if reorientation is required.

The output image intensity can be modulated using the (local or global) volume change if a linear or nonlinear transformation is applied. ‘FOD’ modulation preserves the apparent fibre density across the fibre bundle width and can only be applied if FOD reorientation is used. Alternatively, non-directional scaling by the Jacobian determinant can be applied to any image type.

If a DW scheme is contained in the header (or specified separately), and the number of directions matches the number of volumes in the images, any transformation applied using the `-linear` option will also be applied to the directions.

When the `-template` option is used to specify the target image grid, the image provided via this option will not influence the axis data strides of the output image; these are determined based on the input image, or the input to the `-strides` option.

Options

Affine transformation options

- **-linear transform** specify a linear transform to apply, in the form of a 3x4 or 4x4 ascii file. Note the standard ‘reverse’ convention is used, where the transform maps points in the template image to the moving image. Note that the reverse convention is still assumed even if no `-template` image is supplied
- **-flip axes** flip the specified axes, provided as a comma-separated list of indices (0:x, 1:y, 2:z).
- **-inverse** apply the inverse transformation
- **-half** apply the matrix square root of the transformation. This can be combined with the inverse option.
- **-replace file** replace the linear transform of the original image by that specified, rather than applying it to the original image. The specified transform can be either a template image, or a 3x4 or 4x4 ascii file.
- **-identity** set the header transform of the image to the identity matrix

Regridding options

- **-template image** reslice the input image to match the specified template image grid.
- **-midway_space** reslice the input image to the midway space. Requires either the **-template** or **-warp** option. If used with **-template** and **-linear** option the input image will be resliced onto the grid halfway between the input and template. If used with the **-warp** option the input will be warped to the midway space defined by the grid of the input warp (i.e. half way between image1 and image2)
- **-interp method** set the interpolation method to use when reslicing (choices: nearest, linear, cubic, sinc. Default: cubic).
- **-oversample factor** set the amount of over-sampling (in the target space) to perform when regridding. This is particularly relevant when downsampling a high-resolution image to a low-resolution image, to avoid aliasing artefacts. This can consist of a single integer, or a comma-separated list of 3 integers if different oversampling factors are desired along the different axes. Default is determined from ratio of voxel dimensions (disabled for nearest-neighbour interpolation).

Non-linear transformation options

- **-warp image** apply a non-linear 4D deformation field to warp the input image. Each voxel in the deformation field must define the scanner space position that will be used to interpolate the input image during warping (i.e. pull-back/reverse warp convention). If the **-template image** is also supplied the deformation field will be resliced first to the template image grid. If no **-template** option is supplied then the output image will have the same image grid as the deformation field. This option can be used in combination with the **-affine** option, in which case the affine will be applied first)
- **-warp_full image** warp the input image using a 5D warp file output from mrregister. Any linear transforms in the warp image header will also be applied. The **-warp_full** option must be used in combination with either the **-template** option or the **-midway_space** option. If a **-template image** is supplied then the full warp will be used. By default the image1->image2 transform will be applied, however the **-from 2** option can be used to apply the image2->image1 transform. Use the **-midway_space** option to warp the input image to the midway space. The **-from** option can also be used to define which warp to use when transforming to midway space
- **-from image** used to define which space the input image is when using the **-warp_mid** option. Use **-from 1** to warp from image1 or **-from 2** to warp from image2

Fibre orientation distribution handling options

- **-modulate intensity modulation method** Valid choices are: fod and jac. fod: modulate FODs during reorientation to preserve the apparent fibre density across fibre bundle widths before and after the transformation. jac: modulate the image intensity with the determinant of the Jacobian of the warp of linear transformation to preserve the total intensity before and after the transformation.
- **-directions file** directions defining the number and orientation of the apodised point spread functions used in FOD reorientation (Default: 300 directions)
- **-reorient_fod boolean** specify whether to perform FOD reorientation. This is required if the number of volumes in the 4th dimension corresponds to the number of coefficients in an antipodally symmetric spherical harmonic series with $l_{max} \geq 2$ (i.e. 6, 15, 28, 45, 66 volumes).

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

DW gradient table export options

- **-export_grad_mrtrix path** export the diffusion-weighted gradient table to file in MRtrix format
- **-export_grad_fsl bvecs_path bvals_path** export the diffusion-weighted gradient table to files in FSL (bvecs / bvals) format

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Additional generic options for mrtransform

- **-nan** Use NaN as the out of bounds value (Default: 0.0)
- **-no_reorientation** deprecated, use -reorient_fod instead

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.

- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- If FOD reorientation is being performed: Raffelt, D.; Tournier, J.-D.; Crozier, S.; Connelly, A. & Salvado, O. Reorientation of fiber orientation distributions using apodized point spread functions. *Magnetic Resonance in Medicine*, 2012, 67, 844-855
- If FOD modulation is being performed: Raffelt, D.; Tournier, J.-D.; Rose, S.; Ridgway, G.R.; Henderson, R.; Crozier, S.; Salvado, O.; Connelly, A.; Apparent Fibre Density: a novel measure for the analysis of diffusion-weighted magnetic resonance images. *NeuroImage*, 2012, 15;59(4), 3976-94

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com) and David Raffelt (david.raffelt@florey.edu.au) and Max Pietsch (maximilian.pietsch@kcl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.93 mrtrix_cleanup

Synopsis

Clean up residual temporary files & scratch directories from MRtrix3 commands

Usage

```
mrtrix_cleanup path [ options ]
```

- *path*: Path from which to commence filesystem search

Description

This script will search the file system at the specified location (and in sub-directories thereof) for any temporary files or directories that have been left behind by failed or terminated MRtrix3 commands, and attempt to delete them.

Note that the script’s search for temporary items will not extend beyond the user-specified filesystem location. This means that any built-in or user-specified default location for MRtrix3 piped data and scripts will not be automatically searched. Cleanup of such locations should instead be performed explicitly: e.g. “mrtrix_cleanup /tmp/” to remove residual piped images from /tmp/.

This script should not be run while other MRtrix3 commands are being executed: it may delete temporary items during operation that may lead to unexpected behaviour.

Options

- **-test** Run script in test mode: will list identified files / directories, but not attempt to delete them
- **-failed file** Write list of items that the script failed to delete to a text file

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.94 mrview

Synopsis

The MRtrix image viewer

Usage

```
mrview [ options ] [ image ... ]
```

- *image*: An image to be loaded.

Description

Any images listed as arguments will be loaded and available through the image menu, with the first listed displayed initially. Any subsequent command-line options will be processed as if the corresponding action had been performed through the GUI.

Note that because images loaded as arguments (i.e. simply listed on the command-line) are opened before the GUI is shown, subsequent actions to be performed via the various command-line options must appear after the last argument. This is to avoid confusion about which option will apply to which image. If you need fine control over this, please use the `-load` or `-select_image` options. For example:

```
$ mrview -load image1.mif -interpolation 0 -load image2.mif -interpolation 0
```

or

```
$ mrview image1.mif image2.mif -interpolation 0 -select_image 2 -interpolation 0
```

Options

View options

- **-mode index** (*multiple uses permitted*) Switch to view mode specified by the integer index, as per the view menu.
- **-load image** (*multiple uses permitted*) Load image specified and make it current.
- **-reset** (*multiple uses permitted*) Reset the view according to current image. This resets the FOV, projection and focus.
- **-fov value** (*multiple uses permitted*) Set the field of view, in mm.
- **-focus x,y,z or boolean** (*multiple uses permitted*) Either set the position of the crosshairs in scanner coordinates, with the new position supplied as a comma-separated list of floating-point values or show or hide the focus cross hair using a boolean value as argument.
- **-target x,y,z** Set the target location for the viewing window (the scanner coordinate that will appear at the centre of the viewing window)
- **-voxel x,y,z** (*multiple uses permitted*) Set the position of the crosshairs in voxel coordinates, relative the image currently displayed. The new position should be supplied as a comma-separated list of floating-point values.
- **-volume idx** (*multiple uses permitted*) Set the volume index for the image displayed, as a comma-separated list of integers.

- **-plane index** (*multiple uses permitted*) Set the viewing plane, according to the mapping 0: sagittal; 1: coronal; 2: axial.
- **-lock yesno** (*multiple uses permitted*) Set whether view is locked to image axes (0: no, 1: yes).
- **-select_image index** (*multiple uses permitted*) Switch to image number specified, with reference to the list of currently loaded images.
- **-autoscale** (*multiple uses permitted*) Reset the image scaling to automatically determined range.
- **-interpolation boolean** (*multiple uses permitted*) Enable or disable image interpolation in main image.
- **-colourmap index** (*multiple uses permitted*) Switch the image colourmap to that specified, as per the colourmap menu.
- **-noannotations** (*multiple uses permitted*) Hide all image annotation overlays
- **-comments boolean** (*multiple uses permitted*) Show or hide image comments overlay.
- **-voxelinfo boolean** (*multiple uses permitted*) Show or hide voxel information overlay.
- **-orientationlabel boolean** (*multiple uses permitted*) Show or hide orientation label overlay.
- **-colourbar boolean** (*multiple uses permitted*) Show or hide colourbar overlay.
- **-imagevisible boolean** (*multiple uses permitted*) Show or hide the main image.
- **-intensity_range min,max** (*multiple uses permitted*) Set the image intensity range to that specified.

Window management options

- **-size width,height** (*multiple uses permitted*) Set the size of the view area, in pixel units.
- **-position x,y** (*multiple uses permitted*) Set the position of the main window, in pixel units.
- **-fullscreen** Start fullscreen.
- **-exit** Quit MRView.

Sync Options

- **-sync.focus** Sync the focus with other MRView windows that also have this turned on.

Debugging options

- **-fps** Display frames per second, averaged over the last 10 frames. The maximum over the last 3 seconds is also displayed.

Overlay tool options

- **-overlay.load image** (*multiple uses permitted*) Loads the specified image on the overlay tool.
- **-overlay.opacity value** (*multiple uses permitted*) Sets the overlay opacity to floating value [0-1].
- **-overlay.colourmap index** (*multiple uses permitted*) Sets the colourmap of the overlay as indexed in the colourmap dropdown menu.
- **-overlay.colour R,G,B** (*multiple uses permitted*) Specify a manual colour for the overlay, as three comma-separated values

- **-overlay.intensity Min,Max** (*multiple uses permitted*) Set the intensity windowing of the overlay
- **-overlay.threshold_min value** (*multiple uses permitted*) Set the lower threshold value of the overlay
- **-overlay.threshold_max value** (*multiple uses permitted*) Set the upper threshold value of the overlay
- **-overlay.no_threshold_min** (*multiple uses permitted*) Disable the lower threshold for the overlay
- **-overlay.no_threshold_max** (*multiple uses permitted*) Disable the upper threshold for the overlay
- **-overlay.interpolation value** (*multiple uses permitted*) Enable or disable overlay image interpolation.

ROI editor tool options

- **-roi.load image** (*multiple uses permitted*) Loads the specified image on the ROI editor tool.
- **-roi.opacity value** (*multiple uses permitted*) Sets the overlay opacity to floating value [0-1].
- **-roi.colour R,G,B** (*multiple uses permitted*) Sets the colour of the ROI overlay

Tractography tool options

- **-tractography.load tracks** (*multiple uses permitted*) Load the specified tracks file into the tractography tool.
- **-tractography.thickness value** (*multiple uses permitted*) Line thickness of tractography display, [-1.0, 1.0], default is 0.0.
- **-tractography.geometry value** (*multiple uses permitted*) The geometry type to use when rendering tractograms (options are: pseudotubes, lines, points)
- **-tractography.opacity value** (*multiple uses permitted*) Opacity of tractography display, [0.0, 1.0], default is 1.0.
- **-tractography.slab value** (*multiple uses permitted*) Slab thickness of tractography display, in mm. -1 to turn off crop to slab.
- **-tractography.lighting value** (*multiple uses permitted*) Toggle the use of lighting of tractogram geometry
- **-tractography.tsf_load tsf** (*multiple uses permitted*) Load the specified tractography scalar file.
- **-tractography.tsf_range RangeMin,RangeMax** (*multiple uses permitted*) Set range for the tractography scalar file. Requires -tractography.tsf_load already provided.
- **-tractography.tsf_thresh ThresholdMin,ThesholdMax** (*multiple uses permitted*) Set thresholds for the tractography scalar file. Requires -tractography.tsf_load already provided.

ODF tool options

- **-odf.load_sh image** (*multiple uses permitted*) Loads the specified SH-based ODF image on the ODF tool.
- **-odf.load_tensor image** (*multiple uses permitted*) Loads the specified tensor image on the ODF tool.
- **-odf.load_dixel image** (*multiple uses permitted*) Loads the specified dixel-based image on the ODF tool.

Fixel plot tool options

- **-fixel.load image** (*multiple uses permitted*) Load a fixel file (any file inside a fixel directory, or an old .msf / .msh legacy format file) into the fixel tool.

Connectome tool options

- **-connectome.init image** Initialise the connectome tool using a parcellation image.
- **-connectome.load path** (*multiple uses permitted*) Load a matrix file into the connectome tool.

Screen Capture tool options

- **-capture.folder path** (*multiple uses permitted*) Set the output folder for the screen capture tool.
- **-capture.prefix string** (*multiple uses permitted*) Set the output file prefix for the screen capture tool.
- **-capture.grab** (*multiple uses permitted*) Start the screen capture process.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Calamante, F. & Connelly, A. MRtrix: Diffusion tractography in crossing fiber regions. Int. J. Imaging Syst. Technol., 2012, 22, 53-66

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com), Dave Raffelt (david.raffelt@florey.edu.au), Robert E. Smith (robert.smith@florey.edu.au), Rami Tabbara (rami.tabbara@florey.edu.au), Max Pietsch (maximilian.pietsch@kcl.ac.uk), Thijs Dhollander (thijs.dhollander@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.95 mtnormalise

Synopsis

Multi-tissue informed log-domain intensity normalisation

Usage

```
mtnormalise [ options ] input output [ input output ... ]
```

- *input output*: list of all input and output tissue compartment files (see example usage).

Description

This command takes as input any number of tissue components (e.g. from multi-tissue CSD) and outputs corresponding normalised tissue components corrected for the effects of (residual) intensity inhomogeneities. Intensity normalisation is performed by optimising the voxel-wise sum of all tissue compartments towards a constant value, under constraints of spatial smoothness (polynomial basis of a given order). Different to the Raffelt et al. 2017 abstract, this algorithm performs this task in the log-domain instead, with added gradual outlier rejection, different handling of the balancing factors between tissue compartments and a different iteration structure.

The `-mask` option is mandatory and is optimally provided with a brain mask (such as the one obtained from `dwi2mask` earlier in the processing pipeline). Outlier areas with exceptionally low or high combined tissue contributions are accounted for and reoptimised as the intensity inhomogeneity estimation becomes more accurate.

Example usages

- *Default usage (for 3-tissue CSD compartments)*:

```
$ mtnormalise wmfod.mif wmfod_norm.mif gm.mif gm_norm.mif csf.mif csf_norm.mif -
↪mask mask.mif
```

Note how for each tissue compartment, the input and output images are provided as a consecutive pair.

Options

- **-mask image** the mask defines the data used to compute the intensity normalisation. This option is mandatory.
- **-order number** the maximum order of the polynomial basis used to fit the normalisation field in the log-domain. An order of 0 is equivalent to not allowing spatial variance of the intensity normalisation factor. (default: 3)
- **-niter number** set the number of iterations. The first (and potentially only) entry applies to the main loop. If supplied as a comma-separated list of integers, the second entry applies to the inner loop to update the balance factors (default: 15,7).
- **-reference number** specify the (positive) reference value to which the summed tissue compartments will be normalised. (default: 0.282095, SH DC term for unit angular integral)
- **-balanced** incorporate the per-tissue balancing factors into scaling of the output images (NOTE: use of this option has critical consequences for AFD intensity normalisation; should not be used unless these consequences are fully understood)

Debugging options

- **-check_norm image** output the final estimated spatially varying intensity level that is used for normalisation.
- **-check_mask image** output the final mask used to compute the normalisation. This mask excludes regions identified as outliers by the optimisation process.
- **-check_factors file** output the tissue balance factors computed during normalisation.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Raffelt, D.; Dhollander, T.; Tournier, J.-D.; Tabbara, R.; Smith, R. E.; Pierre, E. & Connelly, A. Bias Field Correction and Intensity Normalisation for Quantitative Analysis of Apparent Fibre Density. In Proc. ISMRM, 2017, 26, 3541

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Thijs Dhollander (thijs.dhollander@gmail.com), Rami Tabbara (rami.tabbara@florey.edu.au), David Raffelt (david.raffelt@florey.edu.au), Jonas Rosnarho-Tornstrand (jonas.rosnarho-tornstrand@kcl.ac.uk) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.96 peaks2amp

Synopsis

Extract amplitudes from a peak directions image

Usage

```
peaks2amp [ options ] directions amplitudes
```

- *directions*: the input directions image. Each volume corresponds to the x, y & z component of each direction vector in turn.
- *amplitudes*: the output amplitudes image.

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.97 peaks2fixel

Synopsis

Convert peak directions image to a fixel directory

Usage

```
peaks2fixel [ options ] directions fixels
```

- *directions*: the input directions image; each volume corresponds to the x, y & z component of each direction vector in turn.
- *fixels*: the output fixel directory.

Options

- **-dataname path** the name of the output fixel data file encoding peak amplitudes

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.98 population_template

Synopsis

Generates an unbiased group-average template from a series of images

Usage

```
population_template input_dir template [ options ]
```

- *input_dir*: Input directory containing all images used to build the template
- *template*: Corresponding output template image. For multi-contrast registration, provide multiple paired input_dir and template arguments. Example: WM_dir WM_template.mif GM_dir GM_template.mif

Description

First a template is optimised with linear registration (rigid and/or affine, both by default), then non-linear registration is used to optimise the template further.

Options

Input, output and general options

- **-type** Specify the types of registration stages to perform. Options are “rigid” (perform rigid registration only which might be useful for intra-subject registration in longitudinal analysis), “affine” (perform affine registration) and “nonlinear” as well as combinations of registration types: “rigid_affine”, “rigid_nonlinear”, “affine_nonlinear”, “rigid_affine_nonlinear”. Default: rigid_affine_nonlinear
- **-voxel_size** Define the template voxel size in mm. Use either a single value for isotropic voxels or 3 comma separated values.
- **-initial_alignment** Method of alignment to form the initial template. Options are “mass” (default), “robust_mass” (requires masks), “geometric” and “none”.
- **-mask_dir** Optionally input a set of masks inside a single directory, one per input image (with the same file name prefix). Using masks will speed up registration significantly. Note that masks are used for registration, not for aggregation. To exclude areas from aggregation, NaN-mask your input images.
- **-warp_dir** Output a directory containing warps from each input to the template. If the folder does not exist it will be created
- **-transformed_dir** Output a directory containing the input images transformed to the template. If the folder does not exist it will be created. For multi-contrast registration, provide comma separated list of directories.
- **-linear_transformations_dir** Output a directory containing the linear transformations used to generate the template. If the folder does not exist it will be created

- **-template_mask** Output a template mask. Only works if `-mask_dir` has been input. The template mask is computed as the intersection of all subject masks in template space.
- **-noreorientation** Turn off FOD reorientation in `mrregister`. Reorientation is on by default if the number of volumes in the 4th dimension corresponds to the number of coefficients in an antipodally symmetric spherical harmonic series (i.e. 6, 15, 28, 45, 66 etc)
- **-leave_one_out** Register each input image to a template that does not contain that image. Valid choices: 0, 1, auto. (Default: auto (true if `n_subjects` larger than 2 and smaller than 15))
- **-aggregate** Measure used to aggregate information from transformed images to the template image. Valid choices: mean, median. Default: mean
- **-aggregation_weights** Comma separated file containing weights used for weighted image aggregation. Each row must contain the identifiers of the input image and its weight. Note that this weighs intensity values not transformations (shape).
- **-nanmask** Optionally apply masks to (transformed) input images using NaN values to specify include areas for registration and aggregation. Only works if `-mask_dir` has been input.
- **-copy_input** Copy input images and masks into local scratch directory.

Options for the non-linear registration

- **-nl_scale** Specify the multi-resolution pyramid used to build the non-linear template, in the form of a list of scale factors (default: 0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0). This implicitly defines the number of template levels
- **-nl_lmax** Specify the `lmax` used for non-linear registration for each scale factor, in the form of a list of integers (default: 2,2,2,2,2,2,2,4,4,4,4,4,4,4). The list must be the same length as the `nl_scale` factor list
- **-nl_niter** Specify the number of registration iterations used within each level before updating the template, in the form of a list of integers (default: 5,5,5,5,5,5,5,5,5,5,5,5,5,5,5). The list must be the same length as the `nl_scale` factor list
- **-nl_update_smooth** Regularise the gradient update field with Gaussian smoothing (standard deviation in voxel units, Default 2.0 x `voxel_size`)
- **-nl_disp_smooth** Regularise the displacement field with Gaussian smoothing (standard deviation in voxel units, Default 1.0 x `voxel_size`)
- **-nl_grad_step** The gradient step size for non-linear registration (Default: 0.5)

Options for the linear registration

- **-linear_no_pause** Do not pause the script if a linear registration seems implausible
- **-linear_estimator** Specify estimator for intensity difference metric. Valid choices are: 11 (least absolute: $|x|$), 12 (ordinary least squares), lp (least powers: $|x|^{1.2}$), Default: None (no robust estimator used)
- **-rigid_scale** Specify the multi-resolution pyramid used to build the rigid template, in the form of a list of scale factors (default: 0.3,0.4,0.6,0.8,1.0,1.0). This and `affine_scale` implicitly define the number of template levels
- **-rigid_lmax** Specify the `lmax` used for rigid registration for each scale factor, in the form of a list of integers (default: 2,2,2,4,4,4). The list must be the same length as the `linear_scale` factor list
- **-rigid_niter** Specify the number of registration iterations used within each level before updating the template, in the form of a list of integers (default:50 for each scale). This must be a single number or a list of same length as the `linear_scale` factor list

- **-affine_scale** Specify the multi-resolution pyramid used to build the affine template, in the form of a list of scale factors (default: 0.3,0.4,0.6,0.8,1.0,1.0). This and rigid_scale implicitly define the number of template levels
- **-affine_lmax** Specify the lmax used for affine registration for each scale factor, in the form of a list of integers (default: 2,2,2,4,4,4). The list must be the same length as the linear_scale factor list
- **-affine_niter** Specify the number of registration iterations used within each level before updating the template, in the form of a list of integers (default:500 for each scale). This must be a single number or a list of same length as the linear_scale factor list

Multi-contrast options

- **-mc_weight_initial_alignment** Weight contribution of each contrast to the initial alignment. Comma separated, default: 1.0
- **-mc_weight_rigid** Weight contribution of each contrast to the objective of rigid registration. Comma separated, default: 1.0
- **-mc_weight_affine** Weight contribution of each contrast to the objective of affine registration. Comma separated, default: 1.0
- **-mc_weight_nl** Weight contribution of each contrast to the objective of nonlinear registration. Comma separated, default: 1.0

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) & Max Pietsch (maximilian.pietsch@kcl.ac.uk) & Thijs Dhollander (thijs.dhollander@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.99 responsemean

Synopsis

Calculate the mean response function from a set of text files

Usage

```
responsemean inputs output [ options ]
```

- *inputs*: The input response functions
- *output*: The output mean response function

Description

Example usage: `responsemean input_response1.txt input_response2.txt input_response3.txt ... output_average_response.txt`

All response function files provided must contain the same number of unique b-values (lines), as well as the same number of coefficients per line.

As long as the number of unique b-values is identical across all input files, the coefficients will be averaged. This is performed on the assumption that the actual acquired b-values are identical. This is however impossible for the `responsemean` command to determine based on the data provided; it is therefore up to the user to ensure that this requirement is satisfied.

Options

- **-legacy** Use the legacy behaviour of former command ‘`average_response`’: average response function coefficients directly, without compensating for global magnitude differences between input files

Additional standard options for Python scripts

- **-nocleanup** do not delete intermediate files during script execution, and do not delete scratch directory at script completion.
- **-scratch /path/to/scratch/** manually specify the path in which to generate the scratch directory.
- **-continue <ScratchDir> <LastFile>** continue the script from a previous execution; must provide the scratch directory path, and the name of the last successfully-generated file.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status. Alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files.
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.100 sh2amp

Synopsis

Evaluate the amplitude of an image of spherical harmonic functions along specified directions

Usage

```
sh2amp [ options ] input directions output
```

- *input*: the input image consisting of spherical harmonic (SH) coefficients.
- *directions*: the list of directions along which the SH functions will be sampled, generated using the `dirgen` command
- *output*: the output image consisting of the amplitude of the SH functions along the specified directions.

Description

The input image should consist of a 4D or 5D image, with SH coefficients along the 4th dimension according to the convention below. If 4D (or size 1 along the 5th dimension), the program expects to be provided with a single shell of directions. If 5D, each set of coefficients along the 5th dimension is understood to correspond to a different shell.

The directions can be provided as: - a 2-column ASCII text file contained azimuth / elevation pairs (as produced by `dirgen`) - a 3-column ASCII text file containing x, y, z Cartesian direction vectors (as produced by `dirgen -cart`) - a 4-column ASCII text file containing the x, y, z, b components of a full DW encoding scheme (in MRtrix format, see main documentation for details). - an image file whose header contains a valid DW encoding scheme

If a full DW encoding is provided, the number of shells needs to match those found in the input image of coefficients (i.e. its size along the 5th dimension). If needed, the `-shell` option can be used to pick out the specific shell(s) of interest.

If the input image contains multiple shells (its size along the 5th dimension is greater than one), the program will expect the direction set to contain multiple shells, which can only be provided as a full DW encodings (the last two options in the list above).

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Options

- **-nonnegative** cap all negative amplitudes to zero

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.101 sh2peaks

Synopsis

Extract the peaks of a spherical harmonic function in each voxel

Usage

```
sh2peaks [ options ] SH output
```

- *SH*: the input image of SH coefficients.
- *output*: the output image. Each volume corresponds to the x, y & z component of each peak direction vector in turn.

Description

Peaks of the spherical harmonic function in each voxel are located by commencing a Newton search along each of a set of pre-specified directions

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Options

- **-num peaks** the number of peaks to extract (default: 3).
- **-direction phi theta** (*multiple uses permitted*) the direction of a peak to estimate. The algorithm will attempt to find the same number of peaks as have been specified using this option.
- **-peaks image** the program will try to find the peaks that most closely match those in the image provided.
- **-threshold value** only peak amplitudes greater than the threshold will be considered.
- **-seeds file** specify a set of directions from which to start the multiple restarts of the optimisation (by default, the built-in 60 direction set is used)
- **-mask image** only perform computation within the specified binary brain mask image.
- **-fast** use lookup table to compute associated Legendre polynomials (faster, but approximate).

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Jeurissen, B.; Leemans, A.; Tournier, J.-D.; Jones, D.K.; Sijbers, J. Investigating the prevalence of complex fiber configurations in white matter tissue with diffusion magnetic resonance imaging. *Human Brain Mapping*, 2013, 34(11), 2747-2766

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.102 sh2power

Synopsis

Compute the total power of a spherical harmonics image

Usage

```
sh2power [ options ] SH power
```

- *SH*: the input spherical harmonics coefficients image.
- *power*: the output power image.

Description

This command computes the sum of squared SH coefficients, which equals the mean-squared amplitude of the spherical function it represents.

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Options

- **-spectrum** output the power spectrum, i.e., the power contained within each harmonic degree ($l=0, 2, 4, \dots$) as a 4-D image.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.103 sh2response

Synopsis

Generate an appropriate response function from the image data for spherical deconvolution

Usage

```
sh2response [ options ] SH mask directions response
```

- *SH*: the spherical harmonic decomposition of the diffusion-weighted images
- *mask*: the mask containing the voxels from which to estimate the response function
- *directions*: a 4D image containing the direction vectors along which to estimate the response function
- *response*: the output axially-symmetric spherical harmonic coefficients

Description

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Options

- **-lmax value** specify the maximum harmonic degree of the response function to estimate
- **-dump file** dump the m=0 SH coefficients from all voxels in the mask to the output file, rather than their mean

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.104 shbasis

Synopsis

Examine the values in spherical harmonic images to estimate (and optionally change) the SH basis used

Usage

```
shbasis [ options ] SH [ SH ... ]
```

- *SH*: the input image(s) of SH coefficients.

Description

In previous versions of MRtrix, the convention used for storing spherical harmonic coefficients was a non-orthonormal basis (the $m \neq 0$ coefficients were a factor of $\sqrt{2}$ too large). This error has been rectified in newer versions of MRtrix, but will cause issues if processing SH data that was generated using an older version of MRtrix (or vice-versa).

This command provides a mechanism for testing the basis used in storage of image data representing a spherical harmonic series per voxel, and allows the user to forcibly modify the raw image data to conform to the desired basis.

Note that the “force_*” conversion choices should only be used in cases where this command has previously been unable to automatically determine the SH basis from the image data, but the user themselves are confident of the SH basis of the data.

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Options

- **-convert mode** convert the image data in-place to the desired basis; options are: old,new,force_oldtonew,force_newtoold.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.105 shconv

Synopsis

Perform spherical convolution

Usage

```
shconv [ options ] odf response [ odf response ... ] SH_out
```

- *odf response*: pairs of input ODF image and corresponding responses
- *SH_out*: the output spherical harmonics coefficients image.

Description

Provided with matching pairs of response function and ODF images (containing SH coefficients), perform spherical convolution to provide the corresponding SH coefficients of the signal.

If multiple pairs of inputs are provided, their contributions will be summed into a single output.

If the responses are multi-shell (with one line of coefficients per shell), the output will be a 5-dimensional image, with the SH coefficients of the signal in each shell stored at different indices along the 5th dimension.

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

The spherical harmonic coefficients are stored according the conventions described the main documentation, which can be found at the following link: https://mrtrix.readthedocs.io/en/3.0.2/concepts/spherical_harmonics.html

Options

Data type options

- **-datatype spec** specify output image data type. Valid choices are: float32, float32le, float32be, float64, float64le, float64be, int64, uint64, int64le, uint64le, int64be, uint64be, int32, uint32, int32le, uint32le, int32be, uint32be, int16, uint16, int16le, uint16le, int16be, uint16be, cfloat32, cfloat32le, cfloat32be, cfloat64, cfloat64le, cfloat64be, int8, uint8, bit.

Stride options

- **-strides spec** specify the strides of the output data in memory; either as a comma-separated list of (signed) integers, or as a template image from which the strides shall be extracted and used. The actual strides produced will depend on whether the output image format can support it.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.106 shview

Synopsis

View spherical harmonics surface plots

Usage

```
shview [ options ] [ coefs ]
```

- *coefs*: a text file containing the even order spherical harmonics coefficients to display.

Options

- **-response** assume SH coefficients file only contains $m=0$ terms (zonal harmonics). Used to display the response function as produced by `estimate_response`

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.107 tck2connectome

Synopsis

Generate a connectome matrix from a streamlines file and a node parcellation image

Usage

```
tck2connectome [ options ] tracks_in nodes_in connectome_out
```

- *tracks_in*: the input track file
- *nodes_in*: the input node parcellation image
- *connectome_out*: the output .csv file containing edge weights

Example usages

- *Default usage:*

```
$ tck2connectome tracks.tck nodes.mif connectome.csv -tck_weights_in weights.csv -  
→out_assignments assignments.txt
```

By default, the metric of connectivity quantified in the connectome matrix is the number of streamlines; or, if tcksift2 is used, the sum of streamline weights via the `-tck_weights_in` option. Use of the `-out_assignments` option is recommended as this enables subsequent use of the `connectome2tck` command.

- *Generate a matrix consisting of the mean streamline length between each node pair:*

```
$ tck2connectome tracks.tck nodes.mif distances.csv -scale_length -stat_edge mean
```

By multiplying the contribution of each streamline to the connectome by the length of that streamline, and then, for each edge, computing the mean value across the contributing streamlines, one obtains a matrix where the value in each entry is the mean length across those streamlines belonging to that edge.

- *Generate a connectome matrix where the value of connectivity is the “mean FA”:*

```
$ tcksample tracks.tck FA.mif mean_FA_per_streamline.csv -stat_tck mean;_  
→tck2connectome tracks.tck nodes.mif mean_FA_connectome.csv -scale_file mean_FA_  
→per_streamline.csv -stat_edge mean
```

Here, a connectome matrix that is “weighted by FA” is generated in multiple steps: firstly, for each streamline, the value of the underlying FA image is sampled at each vertex, and the mean of these values is calculated to produce a single scalar value of “mean FA” per streamline; then, as each streamline is assigned to nodes within the connectome, the magnitude of the contribution of that streamline to the matrix is multiplied by the mean FA value calculated prior for that streamline; finally, for each connectome edge, across the values of “mean FA” that were contributed by all of the streamlines assigned to that particular edge, the mean value is calculated.

- *Generate the connectivity fingerprint for streamlines seeded from a particular region:*

```
$ tck2connectome fixed_seed_tracks.tck nodes.mif fingerprint.csv -vector
```

This usage assumes that the streamlines being provided to the command have all been seeded from the (effectively) same location, and as such, only the endpoint of each streamline (not their starting point) is assigned based on the provided parcellation image. Accordingly, the output file contains only a vector of connectivity values rather than a matrix, since each streamline is assigned to only one node rather than two.

Options

Structural connectome streamline assignment option

- **-assignment_end_voxels** use a simple voxel lookup value at each streamline endpoint
- **-assignment_radial_search radius** perform a radial search from each streamline endpoint to locate the nearest node. Argument is the maximum radius in mm; if no node is found within this radius, the streamline endpoint is not assigned to any node. Default search distance is 4mm.
- **-assignment_reverse_search max_dist** traverse from each streamline endpoint inwards along the streamline, in search of the last node traversed by the streamline. Argument is the maximum traversal length in mm (set to 0 to allow search to continue to the streamline midpoint).
- **-assignment_forward_search max_dist** project the streamline forwards from the endpoint in search of a parcellation node voxel. Argument is the maximum traversal length in mm.
- **-assignment_all_voxels** assign the streamline to all nodes it intersects along its length (note that this means a streamline may be assigned to more than two nodes, or indeed none at all)

Structural connectome metric options

- **-scale_length** scale each contribution to the connectome edge by the length of the streamline
- **-scale_invlength** scale each contribution to the connectome edge by the inverse of the streamline length
- **-scale_invnodevol** scale each contribution to the connectome edge by the inverse of the two node volumes
- **-scale_file path** scale each contribution to the connectome edge according to the values in a vector file

Options for outputting connectome matrices

- **-symmetric** Make matrices symmetric on output
- **-zero_diagonal** Set matrix diagonal to zero on output

Other options for tck2connectome

- **-stat_edge statistic** statistic for combining the values from all streamlines in an edge into a single scale value for that edge (options are: sum,mean,min,max; default=sum)
- **-tck_weights_in path** specify a text scalar file containing the streamline weights
- **-keep_unassigned** By default, the program discards the information regarding those streamlines that are not successfully assigned to a node pair. Set this option to keep these values (will be the first row/column in the output matrix)
- **-out_assignments path** output the node assignments of each streamline to a file; this can be used subsequently e.g. by the command connectome2tck
- **-vector** output a vector representing connectivities from a given seed point to target nodes, rather than a matrix of node-node connectivities

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

If using the default streamline-parcel assignment mechanism (or `-assignment_radial_search` option): Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. The effects of SIFT on the reproducibility and biological accuracy of the structural connectome. *NeuroImage*, 2015, 104, 253-265

If using `-scale_invlength` or `-scale_invnodevol` options: Hagmann, P.; Cammoun, L.; Gigandet, X.; Meuli, R.; Honey, C.; Wedeen, V. & Sporns, O. Mapping the Structural Core of Human Cerebral Cortex. *PLoS Biology* 6(7), e159

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.108 tck2fixel

Synopsis

Compute a fixel TDI map from a tractogram

Usage

```
tck2fixel [ options ] tracks fixel_folder_in fixel_folder_out fixel_data_out
```

- *tracks*: the input tracks.
- *fixel_folder_in*: the input fixel folder. Used to define the fixels and their directions
- *fixel_folder_out*: the fixel folder to which the output will be written. This can be the same as the input folder if desired
- *fixel_data_out*: the name of the fixel data image.

Options

- **-angle value** the max angle threshold for assigning streamline tangents to fixels (Default: 45 degrees)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.109 tckconvert

Synopsis

Convert between different track file formats

Usage

```
tckconvert [ options ] input output
```

- *input*: the input track file.
- *output*: the output track file.

Description

The program currently supports MRtrix .tck files (input/output), ascii text files (input/output), VTK polydata files (input/output), and RenderMan RIB (export only).

Note that ascii files will be stored with one streamline per numbered file. To support this, the command will use the multi-file numbering syntax, where square brackets denote the position of the numbering for the files, for example:

```
$ tckconvert input.tck output-[''] .txt
```

will produce files named output-0000.txt, output-0001.txt, output-0002.txt, ...

Options

- **-scanner2voxel reference** if specified, the properties of this image will be used to convert track point positions from real (scanner) coordinates into voxel coordinates.
- **-scanner2image reference** if specified, the properties of this image will be used to convert track point positions from real (scanner) coordinates into image coordinates (in mm).
- **-voxel2scanner reference** if specified, the properties of this image will be used to convert track point positions from voxel coordinates into real (scanner) coordinates.
- **-image2scanner reference** if specified, the properties of this image will be used to convert track point positions from image coordinates (in mm) into real (scanner) coordinates.

Options specific to PLY writer

- **-sides sides** number of sides for streamlines
- **-increment increment** generate streamline points at every (increment) points

Options specific to RIB writer

- **-dec** add DEC as a primvar

Options for both PLY and RIB writer

- **-radius radius** radius of the streamlines

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Daan Christiaens (daan.christiaens@kcl.ac.uk), J-Donald Tournier (jdtournier@gmail.com), Philip Broser (philip.broser@me.com), Daniel Blezek (daniel.blezek@gmail.com).

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.110 tckdfc

Synopsis

Perform the Track-Weighted Dynamic Functional Connectivity (TW-dFC) method

Usage

```
tckdfc [ options ] tracks fmri output
```

- *tracks*: the input track file.
- *fmri*: the pre-processed fMRI time series
- *output*: the output TW-dFC image

Description

This command generates a Track-Weighted Image (TWI), where the contribution from each streamline to the image is the Pearson correlation between the fMRI time series at the streamline endpoints.

The output image can be generated in one of two ways (note that one of these two command-line options **MUST** be provided):

- “Static” functional connectivity (-static option): Each streamline contributes to a static 3D output image based on the correlation between the signals at the streamline endpoints using the entirety of the input time series.
- “Dynamic” functional connectivity (-dynamic option): The output image is a 4D image, with the same number of volumes as the input fMRI time series. For each volume, the contribution from each streamline is calculated based on a finite-width sliding time window, centred at the timepoint corresponding to that volume.

Note that the -backtrack option in this command is similar, but not precisely equivalent, to back-tracking as can be used with Anatomically-Constrained Tractography (ACT) in the tckgen command. However, here the feature does not change the streamlines trajectories in any way; it simply enables detection of the fact that the input fMRI image may not contain a valid timeseries underneath the streamline endpoint, and where this occurs, searches from the streamline endpoint inwards along the streamline trajectory in search of a valid timeseries to sample from the input image.

Options

Options for toggling between static and dynamic TW-dFC methods; note that one of these options **MUST be provided**

- **-static** generate a “static” (3D) output image.
- **-dynamic shape width** generate a “dynamic” (4D) output image; must additionally provide the shape and width (in volumes) of the sliding window.

Options for setting the properties of the output image

- **-template image** an image file to be used as a template for the output (the output image will have the same transform and field of view).
- **-vox size** provide either an isotropic voxel size (in mm), or comma-separated list of 3 voxel dimensions.
- **-stat_vox type** define the statistic for choosing the final voxel intensities for a given contrast type given the individual values from the tracks passing through each voxel Options are: sum, min, mean, max (default: mean)

Other options for affecting the streamline sampling & mapping behaviour

- **-backtrack** if no valid timeseries is found at the streamline endpoint, back-track along the streamline trajectory until a valid timeseries is found
- **-upsample factor** upsample the tracks by some ratio using Hermite interpolation before mapping (if omitted, an appropriate ratio will be determined automatically)

Standard options

- **-info** display information messages.

- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Calamante, F.; Smith, R.E.; Liang, X.; Zalesky, A.; Connelly, A Track-weighted dynamic functional connectivity (TW-dFC): a new method to study time-resolved functional connectivity. *Brain Struct Funct*, 2017, doi: 10.1007/s00429-017-1431-1

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.111 tckedit

Synopsis

Perform various editing operations on track files

Usage

```
tckedit [ options ] tracks_in [ tracks_in ... ] tracks_out
```

- *tracks_in*: the input track file(s)
- *tracks_out*: the output track file

Description

This command can be used to perform various types of manipulations on track data. A range of such manipulations are demonstrated in the examples provided below.

Example usages

- *Concatenate data from multiple track files into one:*

```
$ tckedit *.tck all_tracks.tck
```

Here the wildcard operator is used to select all files in the current working directory that have the .tck filetype suffix; but input files can equivalently be specified one at a time explicitly.

- *Extract a reduced number of streamlines:*

```
$ tckedit in_many.tck out_few.tck -number 1k -skip 500
```

The number of streamlines requested would typically be less than the number of streamlines in the input track file(s); if it is instead greater, then the command will issue a warning upon completion. By default the streamlines for the output file are extracted from the start of the input file(s); in this example the command is instead instructed to skip the first 500 streamlines, and write to the output file streamlines 501-1500.

- *Extract streamlines based on selection criteria:*

```
$ tckedit in.tck out.tck -include ROI1.mif -include ROI2.mif -minlength 25
```

Multiple criteria can be added in a single invocation of tckedit, and a streamline must satisfy all criteria imposed in order to be written to the output file. Note that both -include and -exclude options can be specified multiple times to provide multiple waypoints / exclusion masks.

- *Select only those streamline vertices within a mask:*

```
$ tckedit in.tck cropped.tck -mask mask.mif
```

The -mask option is applied to each streamline vertex independently, rather than to each streamline, retaining only those streamline vertices within the mask. As such, use of this option may result in a greater number of output streamlines than input streamlines, as a single input streamline may have the vertices at either endpoint retained but some vertices at its midpoint removed, effectively cutting one long streamline into multiple shorter streamlines.

Options

Region Of Interest processing options

- **-include spec** (*multiple uses permitted*) specify an inclusion region of interest, as either a binary mask image, or as a sphere using 4 comma-separated values (x,y,z,radius). Streamlines must traverse ALL inclusion regions to be accepted.
- **-include_ordered image** (*multiple uses permitted*) specify an inclusion region of interest, as either a binary mask image, or as a sphere using 4 comma-separated values (x,y,z,radius). Streamlines must traverse ALL inclusion_ordered regions in the order they are specified in order to be accepted.
- **-exclude spec** (*multiple uses permitted*) specify an exclusion region of interest, as either a binary mask image, or as a sphere using 4 comma-separated values (x,y,z,radius). Streamlines that enter ANY exclude region will be discarded.

- **-mask spec** (*multiple uses permitted*) specify a masking region of interest, as either a binary mask image, or as a sphere using 4 comma-separated values (x,y,z,radius). If defined, streamlines exiting the mask will be truncated.

Streamline length threshold options

- **-maxlength value** set the maximum length of any streamline in mm
- **-minlength value** set the minimum length of any streamline in mm

Streamline count truncation options

- **-number count** set the desired number of selected streamlines to be propagated to the output file
- **-skip count** omit this number of selected streamlines before commencing writing to the output file

Thresholds pertaining to per-streamline weighting

- **-maxweight value** set the maximum weight of any streamline
- **-minweight value** set the minimum weight of any streamline

Other options specific to tckedit

- **-inverse** output the inverse selection of streamlines based on the criteria provided; i.e. only those streamlines that fail at least one selection criterion, and/or vertices that are outside masks if provided, will be written to file
- **-ends_only** only test the ends of each streamline against the provided include/exclude ROIs

Options for handling streamline weights

- **-tck_weights_in path** specify a text scalar file containing the streamline weights
- **-tck_weights_out path** specify the path for an output text scalar file containing streamline weights

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.112 tckgen

Synopsis

Perform streamlines tractography

Usage

```
tckgen [ options ] source tracks
```

- *source*: The image containing the source data. The type of image data required depends on the algorithm used (see Description section).
- *tracks*: the output file containing the tracks generated.

Description

By default, tckgen produces a fixed number of streamlines, by attempting to seed from new random locations until the target number of streamlines have been selected (in other words, after all inclusion & exclusion criteria have been applied), or the maximum number of seeds has been exceeded (by default, this is 1000 x the desired number of selected streamlines). Use the `-select` and/or `-seeds` options to modify as required. See also the Seeding options section for alternative seeding strategies.

Below is a list of available tracking algorithms, the input image data that they require, and a brief description of their behaviour:

- **FACT: Fiber Assigned by Continuous Tracking.** A deterministic algorithm that takes as input a 4D image, with 3xN volumes, where N is the maximum number of fiber orientations in a voxel. Each triplet of volumes represents a 3D vector corresponding to a fiber orientation; the length of the vector additionally indicates some measure of density or anisotropy. As streamlines move from one voxel to another, the fiber orientation most collinear with the streamline orientation is selected (i.e. there is no intra-voxel interpolation).

- **iFOD1:** First-order Integration over Fiber Orientation Distributions. A probabilistic algorithm that takes as input a Fiber Orientation Distribution (FOD) image represented in the Spherical Harmonic (SH) basis. At each streamline step, random samples from the local (trilinear interpolated) FOD are taken. A streamline is more probable to follow orientations where the FOD amplitude is large; but it may also rarely traverse orientations with small FOD amplitude.
- **iFOD2 (default):** Second-order Integration over Fiber Orientation Distributions. A probabilistic algorithm that takes as input a Fiber Orientation Distribution (FOD) image represented in the Spherical Harmonic (SH) basis. Candidate streamline paths (based on short curved “arcs”) are drawn, and the underlying (trilinear-interpolated) FOD amplitudes along those arcs are sampled. A streamline is more probable to follow a path where the FOD amplitudes along that path are large; but it may also rarely traverse orientations where the FOD amplitudes are small, as long as the amplitude remains above the FOD amplitude threshold along the entire path.
- **NullDist1 / NullDist2:** Null Distribution tracking algorithms. These probabilistic algorithms expect as input the same image that was used when invoking the corresponding algorithm for which the null distribution is sought. These algorithms generate streamlines based on random orientation samples; that is, no image information relating to fiber orientations is used, and streamlines trajectories are determined entirely from random sampling. The NullDist2 algorithm is designed to be used in conjunction with iFOD2; NullDist1 should be used in conjunction with any first-order algorithm.
- **SD_STREAM:** Streamlines tractography based on Spherical Deconvolution (SD). A deterministic algorithm that takes as input a Fiber Orientation Distribution (FOD) image represented in the Spherical Harmonic (SH) basis. At each streamline step, the local (trilinear-interpolated) FOD is sampled, and from the current streamline tangent orientation, a Newton optimisation on the sphere is performed in order to locate the orientation of the nearest FOD amplitude peak.
- **SeedTest:** A dummy streamlines algorithm used for testing streamline seeding mechanisms. Any image can be used as input; the image will not be used in any way. For each seed point generated by the seeding mechanism(s), a streamline containing a single point corresponding to that seed location will be written to the output track file.
- **Tensor_Det:** A deterministic algorithm that takes as input a 4D diffusion-weighted image (DWI) series. At each streamline step, the diffusion tensor is fitted to the local (trilinear-interpolated) diffusion data, and the streamline trajectory is determined as the principal eigenvector of that tensor.
- **Tensor_Prob:** A probabilistic algorithm that takes as input a 4D diffusion-weighted image (DWI) series. Within each image voxel, a residual bootstrap is performed to obtain a unique realisation of the DWI data in that voxel for each streamline. These data are then sampled via trilinear interpolation at each streamline step, the diffusion tensor model is fitted, and the streamline follows the orientation of the principal eigenvector of that tensor.

Note that the behaviour of the `-angle` option varies slightly depending on the order of integration: for any first-order method, this angle corresponds to the deviation in streamline trajectory per step; for higher-order methods, this corresponds to the change in underlying fibre orientation between the start and end points of each step.

Options

- **-algorithm name** specify the tractography algorithm to use. Valid choices are: FACT, iFOD1, iFOD2, NullDist1, NullDist2, SD_Stream, Seedtest, Tensor_Det, Tensor_Prob (default: iFOD2).

Streamlines tractography options

- **-select number** set the desired number of streamlines to be selected by `tckgen`, after all selection criteria have been applied (i.e. inclusion/exclusion ROIs, min/max length, etc). `tckgen` will keep seeding streamlines until this number of streamlines have been selected, or the maximum allowed number of seeds has been exceeded (see `-seeds` option). By default, 5000 streamlines are to be selected. Set to zero to disable, which will result in streamlines being seeded until the number specified by `-seeds` has been reached.

- **-step size** set the step size of the algorithm in mm (defaults: for first-order algorithms, 0.1 x voxelsize; if using RK4, 0.25 x voxelsize; for iFOD2: 0.5 x voxelsize).
- **-angle theta** set the maximum angle in degrees between successive steps (defaults: 60 for deterministic algorithms; 15 for iFOD1 / nulldist1; 45 for iFOD2 / nulldist2)
- **-minlength value** set the minimum length of any track in mm (defaults: without ACT, 5 x voxelsize; with ACT, 2 x voxelsize).
- **-maxlength value** set the maximum length of any track in mm (default: 100 x voxelsize).
- **-cutoff value** set the FOD amplitude / fixel size / tensor FA cutoff for terminating tracks (defaults: 0.1 for FOD-based algorithms; 0.1 for fixel-based algorithms; 0.1 for tensor-based algorithms; threshold multiplied by 0.5 when using ACT).
- **-trials number** set the maximum number of sampling trials at each point (only used for iFOD1 / iFOD2) (default: 1000).
- **-noprecomputed** do NOT pre-compute legendre polynomial values. Warning: this will slow down the algorithm by a factor of approximately 4.
- **-rk4** use 4th-order Runge-Kutta integration (slower, but eliminates curvature overshoot in 1st-order deterministic methods)
- **-stop** stop propagating a streamline once it has traversed all include regions
- **-downsample factor** downsample the generated streamlines to reduce output file size (default is (samples-1) for iFOD2, no downsampling for all other algorithms)

Tractography seeding mechanisms; at least one must be provided

- **-seed_image image** (*multiple uses permitted*) seed streamlines entirely at random within a mask image
- **-seed_sphere spec** (*multiple uses permitted*) spherical seed as four comma-separated values (XYZ position and radius)
- **-seed_random_per_voxel image num_per_voxel** (*multiple uses permitted*) seed a fixed number of streamlines per voxel in a mask image; random placement of seeds in each voxel
- **-seed_grid_per_voxel image grid_size** (*multiple uses permitted*) seed a fixed number of streamlines per voxel in a mask image; place seeds on a 3D mesh grid (grid_size argument is per axis; so a grid_size of 3 results in 27 seeds per voxel)
- **-seed_rejection image** (*multiple uses permitted*) seed from an image using rejection sampling (higher values = more probable to seed from)
- **-seed_gmwmi image** (*multiple uses permitted*) seed from the grey matter - white matter interface (only valid if using ACT framework). Input image should be a 3D seeding volume; seeds drawn within this image will be optimised to the interface using the 5TT image provided using the -act option.
- **-seed_dynamic fod_image** determine seed points dynamically using the SIFT model (must not provide any other seeding mechanism). Note that while this seeding mechanism improves the distribution of reconstructed streamlines density, it should NOT be used as a substitute for the SIFT method itself.

Tractography seeding options and parameters

- **-seeds number** set the number of seeds that tckgen will attempt to track from. If this option is NOT provided, the default number of seeds is set to 1000x the number of selected streamlines. If -select is NOT also specified, tckgen will continue tracking until this number of seeds has been attempted. However, if -select is also specified,

tckgen will stop when the number of seeds attempted reaches the number specified here, OR when the number of streamlines selected reaches the number requested with the `-select` option. This can be used to prevent the program from running indefinitely when no or very few streamlines can be found that match the selection criteria. Setting this to zero will cause tckgen to keep attempting seeds until the number specified by `-select` has been reached.

- **-max_attempts_per_seed number** set the maximum number of times that the tracking algorithm should attempt to find an appropriate tracking direction from a given seed point. This should be set high enough to ensure that an actual plausible seed point is not discarded prematurely as being unable to initiate tracking from. Higher settings may affect performance if many seeds are genuinely impossible to track from, as many attempts will still be made in vain for such seeds. (default: 1000)
- **-seed_cutoff value** set the minimum FA or FOD amplitude for seeding tracks (default is the same as the normal `-cutoff`).
- **-seed_unidirectional** track from the seed point in one direction only (default is to track in both directions).
- **-seed_direction dir** specify a seeding direction for the tracking (this should be supplied as a vector of 3 comma-separated values).
- **-output_seeds path** output the seed location of all successful streamlines to a file

Region Of Interest processing options

- **-include spec** (*multiple uses permitted*) specify an inclusion region of interest, as either a binary mask image, or as a sphere using 4 comma-separated values (x,y,z,radius). Streamlines must traverse ALL inclusion regions to be accepted.
- **-include_ordered image** (*multiple uses permitted*) specify an inclusion region of interest, as either a binary mask image, or as a sphere using 4 comma-separated values (x,y,z,radius). Streamlines must traverse ALL inclusion_ordered regions in the order they are specified in order to be accepted.
- **-exclude spec** (*multiple uses permitted*) specify an exclusion region of interest, as either a binary mask image, or as a sphere using 4 comma-separated values (x,y,z,radius). Streamlines that enter ANY exclude region will be discarded.
- **-mask spec** (*multiple uses permitted*) specify a masking region of interest, as either a binary mask image, or as a sphere using 4 comma-separated values (x,y,z,radius). If defined, streamlines exiting the mask will be truncated.

Anatomically-Constrained Tractography options

- **-act image** use the Anatomically-Constrained Tractography framework during tracking; provided image must be in the 5TT (five-tissue-type) format
- **-backtrack** allow tracks to be truncated and re-tracked if a poor structural termination is encountered
- **-crop_at_gmwmi** crop streamline endpoints more precisely as they cross the GM-WM interface

Options specific to the iFOD tracking algorithms

- **-power value** raise the FOD to the power specified (defaults are: 1.0 for iFOD1; 1.0/nsamples for iFOD2).

Options specific to the iFOD2 tracking algorithm

- **-samples number** set the number of FOD samples to take per step (Default: 4).

DW gradient table import options

- **-grad file** Provide the diffusion-weighted gradient scheme used in the acquisition in a text file. This should be supplied as a 4xN text file with each line is in the format [X Y Z b], where [X Y Z] describe the direction of the applied gradient, and b gives the b-value in units of s/mm². If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.
- **-fslgrad bvecs bvals** Provide the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files. If a diffusion gradient scheme is present in the input image header, the data provided with this option will be instead used.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

References based on streamlines algorithm used:

- FACT: Mori, S.; Crain, B. J.; Chacko, V. P. & van Zijl, P. C. M. Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Annals of Neurology*, 1999, 45, 265-269
- iFOD1 or SD_STREAM: Tournier, J.-D.; Calamante, F. & Connelly, A. MRtrix: Diffusion tractography in crossing fiber regions. *Int. J. Imaging Syst. Technol.*, 2012, 22, 53-66
- iFOD2: Tournier, J.-D.; Calamante, F. & Connelly, A. Improved probabilistic streamlines tractography by 2nd order integration over fibre orientation distributions. *Proceedings of the International Society for Magnetic Resonance in Medicine*, 2010, 1670
- Nulldist1 / Nulldist2: Morris, D. M.; Embleton, K. V. & Parker, G. J. Probabilistic fibre tracking: Differentiation of connections from chance events. *NeuroImage*, 2008, 42, 1329-1339
- Tensor_Det: Basser, P. J.; Pajevic, S.; Pierpaoli, C.; Duda, J. & Aldroubi, A. In vivo fiber tractography using DT-MRI data. *Magnetic Resonance in Medicine*, 2000, 44, 625-632
- Tensor_Prob: Jones, D. Tractography Gone Wild: Probabilistic Fibre Tracking Using the Wild Bootstrap With Diffusion Tensor MRI. *IEEE Transactions on Medical Imaging*, 2008, 27, 1268-1274

References based on command-line options:

- -rk4: Basser, P. J.; Pajevic, S.; Pierpaoli, C.; Duda, J. & Aldroubi, A. In vivo fiber tractography using DT-MRI data. *Magnetic Resonance in Medicine*, 2000, 44, 625-632

- `-act`, `-backtrack`, `-seed_gmwmi`: Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 2012, 62, 1924-1938
- `-seed_dynamic`: Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. SIFT2: Enabling dense quantitative assessment of brain white matter connectivity using streamlines tractography. *NeuroImage*, 2015, 119, 338-351

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com) and Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.113 tckglobal

Synopsis

Multi-Shell Multi-Tissue Global Tractography

Usage

```
tckglobal [ options ] source response tracks
```

- *source*: the image containing the raw DWI data.
- *response*: the response of a track segment on the DWI signal.
- *tracks*: the output file containing the tracks generated.

Description

This command will reconstruct the global white matter fibre tractogram that best explains the input DWI data, using a multi-tissue spherical convolution model.

Example use:

```
$ tckglobal dwi.mif wmr.txt -riso csfr.txt -riso gmr.txt -mask mask.mif -niter 1e9 -fod fod.mif -fiso fiso.mif tracks.tck
```

in which `dwi.mif` is the input image, `wmr.txt` is an anisotropic, multi-shell response function for WM, and `csfr.txt` and `gmr.txt` are isotropic response functions for CSF and GM. The output tractogram is saved to `tracks.tck`. Optional output images `fod.mif` and `fiso.mif` contain the predicted WM fODF and isotropic tissue fractions of CSF and GM respectively, estimated as part of the global optimization and thus affected by spatial regularization.

Options

Input options

- **-grad scheme** specify the diffusion encoding scheme (required if not supplied in the header).
- **-mask image** only reconstruct the tractogram within the specified brain mask image.
- **-riso response** (*multiple uses permitted*) set one or more isotropic response functions. (multiple allowed)

Parameters

- **-lmax order** set the maximum harmonic order for the output series. (default = 8)
- **-length size** set the length of the particles (fibre segments). (default = 1mm)
- **-weight w** set the weight by which particles contribute to the model. (default = 0.1)
- **-ppot u** set the particle potential, i.e., the cost of adding one segment, relative to the particle weight. (default = 0.05)
- **-cpot v** set the connection potential, i.e., the energy term that drives two segments together. (default = 0.5)
- **-t0 start** set the initial temperature of the metropolis hastings optimizer. (default = 0.1)
- **-t1 end** set the final temperature of the metropolis hastings optimizer. (default = 0.001)
- **-niter n** set the number of iterations of the metropolis hastings optimizer. (default = 10M)

Output options

- **-fod odF** Predicted fibre orientation distribution function (fODF). This fODF is estimated as part of the global track optimization, and therefore incorporates the spatial regularization that it imposes. Internally, the fODF is represented as a discrete sum of apodized point spread functions (aPSF) oriented along the directions of all particles in the voxel, used to predict the DWI signal from the particle configuration.
- **-noapo** disable spherical convolution of fODF with apodized PSF, to output a sum of delta functions rather than a sum of aPSFs.
- **-fiso iso** Predicted isotropic fractions of the tissues for which response functions were provided with -riso. Typically, these are CSF and GM.
- **-eext eext** Residual external energy in every voxel.
- **-etrend stats** internal and external energy trend and cooling statistics.

Advanced parameters, if you really know what you're doing

- **-balance b** balance internal and external energy. (default = 0) Negative values give more weight to the internal energy, positive to the external energy.
- **-density lambda** set the desired density of the free Poisson process. (default = 1)
- **-prob prob** set the probabilities of generating birth, death, randshift, optshift and connect proposals respectively. (default = 0.25,0.05,0.25,0.1,0.35)
- **-beta b** set the width of the Hanning interpolation window. (in [0, 1], default = 0) If used, a mask is required, and this mask must keep at least one voxel distance to the image bounding box.

- **-lambda lam** set the weight of the internal energy directly. (default = 1) If provided, any value of -balance will be ignored.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Christiaens, D.; Reisert, M.; Dhollander, T.; Sunaert, S.; Suetens, P. & Maes, F. Global tractography of multi-shell diffusion-weighted imaging data using a multi-tissue model. *NeuroImage*, 2015, 123, 89-101

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Daan Christiaens (daan.christiaens@kcl.ac.uk)

Copyright: Copyright (C) 2015 KU Leuven, Dept. Electrical Engineering, ESAT/PSI, Herestraat 49 box 7003, 3000 Leuven, Belgium

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

5.39.114 tckinfo

Synopsis

Print out information about a track file

Usage

```
tckinfo [ options ] tracks [ tracks ... ]
```

- *tracks*: the input track file.

Options

- **-count** count number of tracks in file explicitly, ignoring the header

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.115 tckmap

Synopsis

Use track data as a form of contrast for producing a high-resolution image

Usage

```
tckmap [ options ] tracks output
```

- *tracks*: the input track file.
- *output*: the output track-weighted image

Description

Note: if you run into limitations with RAM usage, make sure you output the results to a .mif file or .mih / .dat file pair - this will avoid the allocation of an additional buffer to store the output for write-out.

Options

Options for the header of the output image

- **-template image** an image file to be used as a template for the output (the output image will have the same transform and field of view).
- **-vox size** provide either an isotropic voxel size (in mm), or comma-separated list of 3 voxel dimensions.
- **-datatype spec** specify output image data type.

Options for the dimensionality of the output image

- **-dec** perform track mapping in directionally-encoded colour (DEC) space
- **-dixel path** map streamlines to dixels within each voxel; requires either a number of dixels (references an internal direction set), or a path to a text file containing a set of directions stored as azimuth/elevation pairs
- **-tod lmax** generate a Track Orientation Distribution (TOD) in each voxel; need to specify the maximum spherical harmonic degree lmax to use when generating Apodised Point Spread Functions

Options for the TWI image contrast properties

- **-contrast type** define the desired form of contrast for the output image Options are: tdi, length, invlength, scalar_map, scalar_map_count, fod_amp, curvature, vector_file (default: tdi)
- **-image image** provide the scalar image map for generating images with 'scalar_map' / 'scalar_map_count' contrast, or the spherical harmonics image for 'fod_amp' contrast
- **-vector_file path** provide the vector data file for generating images with 'vector_file' contrast
- **-stat_vox type** define the statistic for choosing the final voxel intensities for a given contrast type given the individual values from the tracks passing through each voxel. Options are: sum, min, mean, max (default: sum)
- **-stat_tck type** define the statistic for choosing the contribution to be made by each streamline as a function of the samples taken along their lengths. Only has an effect for 'scalar_map', 'fod_amp' and 'curvature' contrast types. Options are: sum, min, mean, max, median, mean_nonzero, gaussian, ends_min, ends_mean, ends_max, ends_prod (default: mean)
- **-fwhm_tck value** when using gaussian-smoothed per-track statistic, specify the desired full-width half-maximum of the Gaussian smoothing kernel (in mm)

- **-map_zero** if a streamline has zero contribution based on the contrast & statistic, typically it is not mapped; use this option to still contribute to the map even if this is the case (these non-contributing voxels can then influence the mean value in each voxel of the map)
- **-backtrack** when using `-stat_tck ends_*`, if the streamline endpoint is outside the FoV, backtrack along the streamline trajectory until an appropriate point is found

Options for the streamline-to-voxel mapping mechanism

- **-upsample factor** upsample the tracks by some ratio using Hermite interpolation before mapping (If omitted, an appropriate ratio will be determined automatically)
- **-precise** use a more precise streamline mapping strategy, that accurately quantifies the length through each voxel (these lengths are then taken into account during TWI calculation)
- **-ends_only** only map the streamline endpoints to the image
- **-tck_weights_in path** specify a text scalar file containing the streamline weights

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- For TDI or DEC TDI: Calamante, F.; Tournier, J.-D.; Jackson, G. D. & Connelly, A. Track-density imaging (TDI): Super-resolution white matter imaging using whole-brain track-density mapping. *NeuroImage*, 2010, 53, 1233-1243
- If using `-contrast length` and `-stat_vox mean`: Pannek, K.; Mathias, J. L.; Bigler, E. D.; Brown, G.; Taylor, J. D. & Rose, S. E. The average pathlength map: A diffusion MRI tractography-derived index for studying brain pathology. *NeuroImage*, 2011, 55, 133-141
- If using `-dixel` option with TDI contrast only: Smith, R.E., Tournier, J.-D., Calamante, F., Connelly, A. A novel paradigm for automated segmentation of very large whole-brain probabilistic tractography data sets. In *proc. ISMRM*, 2011, 19, 673
- If using `-dixel` option with any other contrast: Pannek, K., Raffelt, D., Salvado, O., Rose, S. Incorporating directional information in diffusion tractography derived maps: angular track imaging (ATI). In *Proc. ISMRM*, 2012, 20, 1912

- If using -tod option: Dhollander, T., Emsell, L., Van Hecke, W., Maes, F., Sunaert, S., Suetens, P. Track Orientation Density Imaging (TODI) and Track Orientation Distribution (TOD) based tractography. *NeuroImage*, 2014, 94, 312-336
- If using other contrasts / statistics: Calamante, F.; Tournier, J.-D.; Smith, R. E. & Connelly, A. A generalised framework for super-resolution track-weighted imaging. *NeuroImage*, 2012, 59, 2494-2503
- If using -precise mapping option: Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. SIFT: Spherical-deconvolution informed filtering of tractograms. *NeuroImage*, 2013, 67, 298-312 (Appendix 3)

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.116 tckresample

Synopsis

Resample each streamline in a track file to a new set of vertices

Usage

```
tckresample [ options ] in_tracks out_tracks
```

- *in_tracks*: the input track file
- *out_tracks*: the output resampled tracks

Description

It is necessary to specify precisely ONE of the command-line options for controlling how this resampling takes place; this may be either increasing or decreasing the number of samples along each streamline, or may involve changing the positions of the samples according to some specified trajectory.

Note that because the length of a streamline is calculated based on the sums of distances between adjacent vertices, resampling a streamline to a new set of vertices will typically change the quantified length of that streamline; the magnitude of the difference will typically depend on the discrepancy in the number of vertices, with less vertices leading to a shorter length (due to taking chordal lengths of curved trajectories).

Options

Streamline resampling options

- **-upsample ratio** increase the density of points along the length of each streamline by some factor (may improve mapping streamlines to ROIs, and/or visualisation)
- **-downsample ratio** increase the density of points along the length of each streamline by some factor (decreases required storage space)
- **-step_size value** re-sample the streamlines to a desired step size (in mm)
- **-num_points count** re-sample each streamline to a fixed number of points
- **-endpoints** only output the two endpoints of each streamline
- **-line num start end** resample tracks at 'num' equidistant locations along a line between 'start' and 'end' (specified as comma-separated 3-vectors in scanner coordinates)
- **-arc num start mid end** resample tracks at 'num' equidistant locations along a circular arc specified by points 'start', 'mid' and 'end' (specified as comma-separated 3-vectors in scanner coordinates)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.117 tcksample

Synopsis

Sample values of an associated image along tracks

Usage

```
tcksample [ options ] tracks image values
```

- *tracks*: the input track file
- *image*: the image to be sampled
- *values*: the output sampled values

Description

By default, the value of the underlying image at each point along the track is written to either an ASCII file (with all values for each track on the same line), or a track scalar file (.tsf). Alternatively, some statistic can be taken from the values along each streamline and written to a vector file.

Options

- **-stat_tck statistic** compute some statistic from the values along each streamline (options are: mean, median, min, max)
- **-nointerp** do not use trilinear interpolation when sampling image values
- **-precise** use the precise mechanism for mapping streamlines to voxels (obviates the need for trilinear interpolation) (only applicable if some per-streamline statistic is requested)
- **-use_tdi_fraction** each streamline is assigned a fraction of the image intensity in each voxel based on the fraction of the track density contributed by that streamline (this is only appropriate for processing a whole-brain tractogram, and images for which the quantitative parameter is additive)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.

- **-help** display this information page and exit.
- **-version** display version information and exit.

References

- If using **-precise** option: Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. SIFT: Spherical-deconvolution informed filtering of tractograms. *NeuroImage*, 2013, 67, 298-312
- Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137
-

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.118 tcksift

Synopsis

Filter a whole-brain fibre-tracking data set such that the streamline densities match the FOD lobe integrals

Usage

```
tcksift [ options ] in_tracks in_fod out_tracks
```

- *in_tracks*: the input track file
- *in_fod*: input image containing the spherical harmonics of the fibre orientation distributions
- *out_tracks*: the output filtered tracks file

Options

- **-nofilter** do NOT perform track filtering - just construct the model in order to provide output debugging images
- **-output_at_counts counts** output filtered track files (and optionally debugging images if **-output_debug** is specified) at specific numbers of remaining streamlines; provide as comma-separated list of integers

Options for setting the processing mask for the SIFT fixel-streamlines comparison model

- **-proc_mask image** provide an image containing the processing mask weights for the model; image spatial dimensions must match the fixel image
- **-act image** use an ACT five-tissue-type segmented anatomical image to derive the processing mask

Options affecting the SIFT model

- **-fd_scale_gm** provide this option (in conjunction with -act) to heuristically downsize the fibre density estimates based on the presence of GM in the voxel. This can assist in reducing tissue interface effects when using a single-tissue deconvolution algorithm
- **-no_dilate_lut** do NOT dilate FOD lobe lookup tables; only map streamlines to FOD lobes if the precise tangent lies within the angular spread of that lobe
- **-make_null_lobes** add an additional FOD lobe to each voxel, with zero integral, that covers all directions with zero / negative FOD amplitudes
- **-remove_untracked** remove FOD lobes that do not have any streamline density attributed to them; this improves filtering slightly, at the expense of longer computation time (and you can no longer do quantitative comparisons between reconstructions if this is enabled)
- **-fd_thresh value** fibre density threshold; exclude an FOD lobe from filtering processing if its integral is less than this amount (streamlines will still be mapped to it, but it will not contribute to the cost function or the filtering)

Options to make SIFT provide additional output files

- **-csv file** output statistics of execution per iteration to a .csv file
- **-out_mu file** output the final value of SIFT proportionality coefficient mu to a text file
- **-output_debug** provide various output images for assessing & debugging performance etc.
- **-out_selection path** output a text file containing the binary selection of streamlines

Options to control when SIFT terminates filtering

- **-term_number value** number of streamlines - continue filtering until this number of streamlines remain
- **-term_ratio value** termination ratio - defined as the ratio between reduction in cost function, and reduction in density of streamlines. Smaller values result in more streamlines being filtered out.
- **-term_mu value** terminate filtering once the SIFT proportionality coefficient reaches a given value

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.

- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. SIFT: Spherical-deconvolution informed filtering of tractograms. *NeuroImage*, 2013, 67, 298-312

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.119 tcksift2

Synopsis

Optimise per-streamline cross-section multipliers to match a whole-brain tractogram to fixel-wise fibre densities

Usage

```
tcksift2 [ options ] in_tracks in_fod out_weights
```

- *in_tracks*: the input track file
- *in_fod*: input image containing the spherical harmonics of the fibre orientation distributions
- *out_weights*: output text file containing the weighting factor for each streamline

Options

Options for setting the processing mask for the SIFT fixel-streamlines comparison model

- **-proc_mask image** provide an image containing the processing mask weights for the model; image spatial dimensions must match the fixel image
- **-act image** use an ACT five-tissue-type segmented anatomical image to derive the processing mask

Options affecting the SIFT model

- **-fd_scale_gm** provide this option (in conjunction with -act) to heuristically downsize the fibre density estimates based on the presence of GM in the voxel. This can assist in reducing tissue interface effects when using a single-tissue deconvolution algorithm
- **-no_dilate_lut** do NOT dilate FOD lobe lookup tables; only map streamlines to FOD lobes if the precise tangent lies within the angular spread of that lobe
- **-make_null_lobes** add an additional FOD lobe to each voxel, with zero integral, that covers all directions with zero / negative FOD amplitudes
- **-remove_untracked** remove FOD lobes that do not have any streamline density attributed to them; this improves filtering slightly, at the expense of longer computation time (and you can no longer do quantitative comparisons between reconstructions if this is enabled)
- **-fd_thresh value** fibre density threshold; exclude an FOD lobe from filtering processing if its integral is less than this amount (streamlines will still be mapped to it, but it will not contribute to the cost function or the filtering)

Options to make SIFT provide additional output files

- **-csv file** output statistics of execution per iteration to a .csv file
- **-out_mu file** output the final value of SIFT proportionality coefficient mu to a text file
- **-output_debug** provide various output images for assessing & debugging performance etc.
- **-out_coeffs path** output text file containing the weighting coefficient for each streamline

Regularisation options for SIFT2

- **-reg_tikhonov value** provide coefficient for regularising streamline weighting coefficients (Tikhonov regularisation) (default: 0)
- **-reg_tv value** provide coefficient for regularising variance of streamline weighting coefficient to fixels along its length (Total Variation regularisation) (default: 0.1)

Options for controlling the SIFT2 optimisation algorithm

- **-min_td_frac fraction** minimum fraction of the FOD integral reconstructed by streamlines; if the reconstructed streamline density is below this fraction, the fixel is excluded from optimisation (default: 0.1)
- **-min_iters count** minimum number of iterations to run before testing for convergence; this can prevent premature termination at early iterations if the cost function increases slightly (default: 10)

- **-max_iters count** maximum number of iterations to run before terminating program
- **-min_factor factor** minimum weighting factor for an individual streamline; if the factor falls below this number the streamline will be rejected entirely (factor set to zero) (default: 0)
- **-min_coeff coeff** minimum weighting coefficient for an individual streamline; similar to the ‘-min_factor’ option, but using the exponential coefficient basis of the SIFT2 model; these parameters are related as: $\text{factor} = e^{(\text{coeff})}$. Note that the -min_factor and -min_coeff options are mutually exclusive - you can only provide one. (default: -inf)
- **-max_factor factor** maximum weighting factor that can be assigned to any one streamline (default: inf)
- **-max_coeff coeff** maximum weighting coefficient for an individual streamline; similar to the ‘-max_factor’ option, but using the exponential coefficient basis of the SIFT2 model; these parameters are related as: $\text{factor} = e^{(\text{coeff})}$. Note that the -max_factor and -max_coeff options are mutually exclusive - you can only provide one. (default: inf)
- **-max_coeff_step step** maximum change to a streamline’s weighting coefficient in a single iteration (default: 1)
- **-min_cf_decrease frac** minimum decrease in the cost function (as a fraction of the initial value) that must occur each iteration for the algorithm to continue (default: 2.5e-05)
- **-linear** perform a linear estimation of streamline weights, rather than the standard non-linear optimisation (typically does not provide as accurate a model fit; but only requires a single pass)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Smith, R. E.; Tournier, J.-D.; Calamante, F. & Connelly, A. SIFT2: Enabling dense quantitative assessment of brain white matter connectivity using streamlines tractography. *NeuroImage*, 2015, 119, 338-351

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.120 tckstats

Synopsis

Calculate statistics on streamlines lengths

Usage

```
tckstats [ options ] tracks_in
```

- *tracks_in*: the input track file

Options

- **-output field** (*multiple uses permitted*) output only the field specified. Multiple such options can be supplied if required. Choices are: mean, median, std, min, max, count. Useful for use in scripts.
- **-histogram path** output a histogram of streamline lengths
- **-dump path** dump the streamlines lengths to a text file
- **-ignorezero** do not generate a warning if the track file contains streamlines with zero length
- **-tck_weights_in path** specify a text scalar file containing the streamline weights

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.121 tcktransform

Synopsis

Apply a spatial transformation to a tracks file

Usage

```
tcktransform [ options ] tracks transform output
```

- *tracks*: the input track file.
- *transform*: the image containing the transform.
- *output*: the output track file

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.122 tensor2metric

Synopsis

Generate maps of tensor-derived parameters

Usage

```
tensor2metric [ options ] tensor
```

- *tensor*: the input tensor image.

Options

- **-adc image** compute the mean apparent diffusion coefficient (ADC) of the diffusion tensor. (sometimes also referred to as the mean diffusivity (MD))
- **-fa image** compute the fractional anisotropy (FA) of the diffusion tensor.
- **-ad image** compute the axial diffusivity (AD) of the diffusion tensor. (equivalent to the principal eigenvalue)
- **-rd image** compute the radial diffusivity (RD) of the diffusion tensor. (equivalent to the mean of the two non-principal eigenvalues)
- **-cl image** compute the linearity metric of the diffusion tensor. (one of the three Westin shape metrics)
- **-cp image** compute the planarity metric of the diffusion tensor. (one of the three Westin shape metrics)
- **-cs image** compute the sphericity metric of the diffusion tensor. (one of the three Westin shape metrics)
- **-value image** compute the selected eigenvalue(s) of the diffusion tensor.
- **-vector image** compute the selected eigenvector(s) of the diffusion tensor.
- **-num sequence** specify the desired eigenvalue/eigenvector(s). Note that several eigenvalues can be specified as a number sequence. For example, ‘1,3’ specifies the principal (1) and minor (3) eigenvalues/eigenvectors (default = 1).

- **-modulate choice** specify how to modulate the magnitude of the eigenvectors. Valid choices are: none, FA, eigval (default = FA).
- **-mask image** only perform computation within the specified binary brain mask image.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Basser, P. J.; Mattiello, J. & Lebihan, D. MR diffusion tensor spectroscopy and imaging. Biophysical Journal, 1994, 66, 259-267

Westin, C. F.; Peled, S.; Gudbjartsson, H.; Kikinis, R. & Jolesz, F. A. Geometrical diffusion measures for MRI from tensor basis analysis. Proc Intl Soc Mag Reson Med, 1997, 5, 1742

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: Thijs Dhollander (thijs.dhollander@gmail.com) & Ben Jeurissen (ben.jeurissen@uantwerpen.be) & J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.123 transformcalc

Synopsis

Perform calculations on linear transformation matrices

Usage

```
transformcalc [ options ] inputs [ inputs ... ] operation output
```

- *inputs*: the input(s) for the specified operation
- *operation*: the operation to perform, one of: invert, half, rigid, header, average, interpolate, decompose, align_vertices_rigid, align_vertices_rigid_scale (see description section for details).
- *output*: the output transformation matrix.

Example usages

- *Invert a transformation:*

```
$ transformcalc matrix_in.txt invert matrix_out.txt
```

- *Calculate the matrix square root of the input transformation (halfway transformation):*

```
$ transformcalc matrix_in.txt half matrix_out.txt
```

- *Calculate the rigid component of an affine input transformation:*

```
$ transformcalc affine_in.txt rigid rigid_out.txt
```

- *Calculate the transformation matrix from an original image and an image with modified header:*

```
$ transformcalc mov mapmovhdr header output
```

- *Calculate the average affine matrix of a set of input matrices:*

```
$ transformcalc input1.txt ... inputN.txt average matrix_out.txt
```

- *Create interpolated transformation matrix between two inputs:*

```
$ transformcalc input1.txt input2.txt interpolate matrix_out.txt
```

Based on matrix decomposition with linear interpolation of translation, rotation and stretch described in: Shoemaker, K., Hill, M., & Duff, T. (1992). Matrix Animation and Polar Decomposition. Matrix, 92, 258-264. doi:10.1.1.56.1336

- *Decompose transformation matrix M into translation, rotation and stretch and shear ($M = T * R * S$):*

```
$ transformcalc matrix_in.txt decompose matrixes_out.txt
```

The output is a key-value text file containing: scaling: vector of 3 scaling factors in x, y, z direction; shear: list of shear factors for xy, xz, yz axes; angles: list of Euler angles about static x, y, z axes in radians in the range $[0:\pi][-\pi:\pi][-\pi:\pi]$; angle_axis: angle in radians and rotation axis; translation : translation vector along x, y, z axes in mm; R: composed rotation matrix ($R = \text{rot}_x * \text{rot}_y * \text{rot}_z$); S: composed scaling and shear matrix

- *Calculate transformation that aligns two images based on sets of corresponding landmarks:*

```
$ transformcalc input moving.txt fixed.txt align_vertices_rigid rigid.txt
```

Similarly, 'align_vertices_rigid_scale' produces an affine matrix (rigid and global scale). Vertex coordinates are in scanner space, corresponding vertices must be stored in the same row of moving.txt and fixed.txt. Requires 3 or more vertices in each file. Algorithm: Kabsch 'A solution for the best rotation to relate two sets of vectors' DOI:10.1107/S0567739476001873

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Max Pietsch (maximilian.pietsch@kcl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.124 transformcompose

Synopsis

Compose any number of linear transformations and/or warps into a single transformation

Usage

```
transformcompose [ options ] input [ input ... ] output
```

- *input*: the input transforms (either linear or non-linear warps).
- *output*: the output file (may be a linear transformation text file, or a deformation warp field image, depending on usage)

Description

Any linear transforms must be supplied as a 4x4 matrix in a text file (e.g. as per the output of `mrregister`). Any warp fields must be supplied as a 4D image representing a deformation field (e.g. as output from `mrrregister -nl_warp`).

Input transformations should be provided to the command in the order in which they would be applied to an image if they were to be applied individually.

If all input transformations are linear, and the `-template` option is not provided, then the file output by the command will also be a linear transformation saved as a 4x4 matrix in a text file. If a template image is supplied, then the output will always be a deformation field. If at least one of the inputs is a warp field, then the output will be a deformation field, which will be defined on the grid of the last input warp image supplied if the `-template` option is not used.

Options

- **-template image** define the output grid defined by a template image

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.125 transformconvert

Synopsis

Convert linear transformation matrices

Usage

```
transformconvert [ options ] input [ input ... ] operation output
```

- *input*: the input(s) for the specified operation
- *operation*: the operation to perform, one of: flirt_import, itk_import
- *output*: the output transformation matrix.

Description

This command allows to convert transformation matrices provided by other registration softwares to a format usable in MRtrix3. Example usages are provided below.

Example usages

- *Convert a transformation matrix produced by FSL's flirt command into a format usable by MRtrix3:*

```
$ transformconvert transform_flirt.mat flirt_in.nii flirt_ref.nii flirt_import_  
↪transform_mrtrix.txt
```

The two images provided as inputs for this operation must be in the correct order: first the image that was provided to flirt via the -in option, second the image that was provided to flirt via the -ref option.

- *Convert a plain text transformation matrix file produced by ITK's affine registration (e.g. ANTS, Slicer) into a format usable by MRtrix3:*

```
$ transformconvert transform_itk.txt itk_import transform_mrtrix.txt
```

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.

- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Max Pietsch (maximilian.pietsch@kcl.ac.uk)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.126 tsfdivide

Synopsis

Divide corresponding values in track scalar files

Usage

```
tsfdivide [ options ] input1 input2 output
```

- *input1*: the first input track scalar file.
- *input2*: the second input track scalar file.
- *output*: the output track scalar file

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).

- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.127 tsfinfo

Synopsis

Print out information about a track scalar file

Usage

```
tsfinfo [ options ] tracks [ tracks ... ]
```

- *tracks*: the input track scalar file.

Options

- **-count** count number of tracks in file explicitly, ignoring the header
- **-ascii prefix** save values of each track scalar file in individual ascii files, with the specified prefix.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.

- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.128 tsfmult

Synopsis

Multiply corresponding values in track scalar files

Usage

```
tsfmult [ options ] input1 input1 output
```

- *input1*: the first input track scalar file.
- *input1*: the second input track scalar file.
- *output*: the output track scalar file

Options

Standard options

- **-info** display information messages.

- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.129 tsfsmooth

Synopsis

Gaussian filter a track scalar file

Usage

```
tsfsmooth [ options ] input output
```

- *input*: the input track scalar file.
- *output*: the output track scalar file

Options

- **-stdev sigma** apply Gaussian smoothing with the specified standard deviation. The standard deviation is defined in units of track points (default: 4)

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.130 tsfthreshold

Synopsis

Threshold and invert track scalar files

Usage

```
tsfthreshold [ options ] input T output
```

- *input*: the input track scalar file.
- *T*: the desired threshold
- *output*: the binary output track scalar file

Options

- **-invert** invert the output mask

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.131 tsfvalidate

Synopsis

Validate a track scalar file against the corresponding track data

Usage

```
tsfvalidate [ options ] tsf tracks
```

- *tsf*: the input track scalar file
- *tracks*: the track file on which the TSF is based

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.132 vectorstats

Synopsis

Statistical testing of vector data using non-parametric permutation testing

Usage

```
vectorstats [ options ] input design contrast output
```

- *input*: a text file listing the file names of the input subject data
- *design*: the design matrix
- *contrast*: the contrast matrix
- *output*: the filename prefix for all output

Description

This command can be used to perform permutation testing of any form of data. The data for each input subject must be stored in a text file, with one value per row. The data for each row across subjects will be tested independently, i.e. there is no statistical enhancement that occurs between the data; however family-wise error control will be used.

In some software packages, a column of ones is automatically added to the GLM design matrix; the purpose of this column is to estimate the “global intercept”, which is the predicted value of the observed variable if all explanatory variables were to be zero. However there are rare situations where including such a column would not be appropriate for a particular experimental design. Hence, in MRtrix3 statistical inference commands, it is up to the user to determine whether or not this column of ones should be included in their design matrix, and add it explicitly if necessary. The contrast matrix must also reflect the presence of this additional column.

Options

Options relating to shuffling of data for nonparametric statistical inference

- **-notest** don't perform statistical inference; only output population statistics (effect size, stdev etc)
- **-errors spec** specify nature of errors for shuffling; options are: ee,ise,both (default: ee)
- **-exchange_within file** specify blocks of observations within each of which data may undergo restricted exchange
- **-exchange_whole file** specify blocks of observations that may be exchanged with one another (for independent and symmetric errors, sign-flipping will occur block-wise)
- **-strong** use strong familywise error control across multiple hypotheses
- **-nshuffles number** the number of shuffles (default: 5000)
- **-permutations file** manually define the permutations (relabelling). The input should be a text file defining a $m \times n$ matrix, where each relabelling is defined as a column vector of size m , and the number of columns, n , defines the number of permutations. Can be generated with the `palm_quickperms` function in PALM (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/PALM>). Overrides the `-nshuffles` option.

Options related to the General Linear Model (GLM)

- **-variance file** define variance groups for the G-statistic; measurements for which the expected variance is equivalent should contain the same index
- **-ftests path** perform F-tests; input text file should contain, for each F-test, a row containing ones and zeros, where ones indicate the rows of the contrast matrix to be included in the F-test.
- **-fonly** only assess F-tests; do not perform statistical inference on entries in the contrast matrix

- **-column path** (*multiple uses permitted*) add a column to the design matrix corresponding to subject element-wise values (note that the contrast matrix must include an additional column for each use of this option); the text file provided via this option should contain a file name for each subject

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.133 voxel2fixel

Synopsis

Map the scalar value in each voxel to all fixels within that voxel

Usage

```
voxel2fixel [ options ] image_in fixel_directory_in fixel_directory_out fixel_data_
↪out
```

- *image_in*: the input image.
- *fixel_directory_in*: the input fixel directory. Used to define the fixels and their directions
- *fixel_directory_out*: the fixel directory where the output will be written. This can be the same as the input directory if desired
- *fixel_data_out*: the name of the fixel data image.

Description

This command is designed to enable CFE-based statistical analysis to be performed on voxel-wise measures.

Options

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.134 voxel2mesh

Synopsis

Generate a surface mesh representation from a voxel image

Usage

```
voxel2mesh [ options ] input output
```

- *input*: the input image.
- *output*: the output mesh file.

Description

This command utilises the Marching Cubes algorithm to generate a polygonal surface that represents the isocontour(s) of the input image at a particular intensity. By default, an appropriate threshold will be determined automatically from the input image, however the intensity value of the isocontour(s) can instead be set manually using the `-threshold` option.

If the `-blocky` option is used, then the Marching Cubes algorithm will not be used. Instead, the input image will be interpreted as a binary mask image, and polygonal surfaces will be generated at the outer faces of the voxel clusters within the mask.

Options

- **-blocky** generate a ‘blocky’ mesh that precisely represents the voxel edges
- **-threshold value** manually set the intensity threshold for the Marching Cubes algorithm

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.135 warp2metric

Synopsis

Compute fixel-wise or voxel-wise metrics from a 4D deformation field

Usage

```
warp2metric [ options ] in
```

- *in*: the input deformation field

Options

- **-fc template_fixel_directory output_fixel_directory output_fixel_data** use an input template fixel image to define fibre orientations and output a fixel image describing the change in fibre cross-section (FC) in the perpendicular plane to the fixel orientation. e.g. `warp2metric warp.mif -fc fixel_template_directory output_fixel_directory fc.mif`
- **-jmat output** output a Jacobian matrix image stored in column-major order along the 4th dimension. Note the output jacobian describes the warp gradient w.r.t the scanner space coordinate system
- **-jdet output** output the Jacobian determinant instead of the full matrix

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.

- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Raffelt, D.; Tournier, JD.; Smith, RE.; Vaughan, DN.; Jackson, G.; Ridgway, GR. Connelly, A. Investigating White Matter Fibre Density and Morphology using Fixel-Based Analysis. *Neuroimage*, 2017, 144, 58-73, doi: 10.1016/j.neuroimage.2016.09.029

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.136 warpconvert

Synopsis

Convert between different representations of a non-linear warp

Usage

```
warpconvert [ options ] in type out
```

- *in*: the input warp image.
- *type*: the conversion type required. Valid choices are: deformation2displacement, displacement2deformation, warpfull2deformation, warpfull2displacement
- *out*: the output warp image.

Description

A deformation field is defined as an image where each voxel defines the corresponding position in the other image (in scanner space coordinates). A displacement field stores the displacements (in mm) to the other image from the each voxel's position (in scanner space). The warpfull file is the 5D format output from `mrregister -nl_warp_full`, which contains linear transforms, warps and their inverses that map each image to a midway space.

Options

- **-template image** define a template image when converting a warpfull file (which is defined on a grid in the midway space between image 1 & 2). For example to generate the deformation field that maps image1 to image2, then supply image2 as the template image
- **-midway_space** to be used only with `warpfull2deformation` and `warpfull2displacement` conversion types. The output will only contain the non-linear warp to map an input image to the midway space (defined by the warpfull grid). If a linear transform exists in the warpfull file header then it will be composed and included in the output.
- **-from image** to be used only with `warpfull2deformation` and `warpfull2displacement` conversion types. Used to define the direction of the desired output field. Use `-from 1` to obtain the image1->image2 field and from 2 for image2->image1. Can be used in combination with the `-midway_space` option to produce a field that only maps to midway space.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.137 warpcorrect

Synopsis

Replaces voxels in a deformation field that point to a specific out of bounds location with nan,nan,nan

Usage

```
warpcorrect [ options ] in out
```

- *in*: the input warp image.
- *out*: the output warp image.

Description

This can be used in conjunction with the warpinit command to compute a MRtrix compatible deformation field from non-linear transformations generated by any other registration package.

Options

- **-marker coordinates** single value or a comma separated list of values that define out of bounds voxels in the input warp image. Default: (0,0,0).
- **-tolerance value** numerical precision used for L2 matrix norm comparison. Default: 9.99999975e-06.

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: David Raffelt (david.raffelt@florey.edu.au) & Max Pietsch (mail@maxpietsch.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.138 warpinit

Synopsis

Create an initial warp image, representing an identity transformation

Usage

```
warpinit [ options ] template warp
```

- *template*: the input template image.
- *warp*: the output warp image.

Description

This is useful to obtain the warp fields from other normalisation applications, by applying the transformation of interest to the warp field generated by this program.

The image generated is a 4D image with the same spatial characteristics as the input template image. It contains 3 volumes, with each voxel containing its own x,y,z coordinates.

Note that this command can be used to create 3 separate X,Y,Z images directly (which may be useful to create images suitable for use in the registration program) using the following syntax:

```
$ warpinit template.mif warp-[''].nii
```

Options

Standard options

- **-info** display information messages.

- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the MRTRIX_QUIET environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 2019, 202, 116137

Author: J-Donald Tournier (jdtournier@gmail.com)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.

5.39.139 warpinvert

Synopsis

Invert a non-linear warp field

Usage

```
warpinvert [ options ] in out
```

- *in*: the input warp image.
- *out*: the output warp image.

Description

By default, this command assumes that the input warp field is a deformation field, i.e. each voxel stores the corresponding position in the other image (in scanner space), and the calculated output warp image will also be a deformation field. If the input warp field is instead a displacement field, i.e. where each voxel stores an offset from which to sample the other image (but still in scanner space), then the `-displacement` option should be used; the output warp field will additionally be calculated as a displacement field in this case.

Options

- **-template image** define a template image grid for the output warp
- **-displacement** indicates that the input warp field is a displacement field; the output will also be a displacement field

Standard options

- **-info** display information messages.
- **-quiet** do not display information messages or progress status; alternatively, this can be achieved by setting the `MRTRIX_QUIET` environment variable to a non-empty string.
- **-debug** display debugging messages.
- **-force** force overwrite of output files (caution: using the same file as input and output might cause unexpected behaviour).
- **-nthreads number** use this number of threads in multi-threaded applications (set to 0 to disable multi-threading).
- **-config key value** (*multiple uses permitted*) temporarily set the value of an MRtrix config file entry.
- **-help** display this information page and exit.
- **-version** display version information and exit.

References

Tournier, J.-D.; Smith, R. E.; Raffelt, D.; Tabbara, R.; Dhollander, T.; Pietsch, M.; Christiaens, D.; Jeurissen, B.; Yeh, C.-H. & Connelly, A. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 2019, 202, 116137

Author: Robert E. Smith (robert.smith@florey.edu.au) and David Raffelt (david.raffelt@florey.edu.au)

Copyright: Copyright (c) 2008-2020 the MRtrix3 contributors.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. See the Mozilla Public License v. 2.0 for more details.

For more details, see <http://www.mrtrix.org/>.


















Lang	Command	Synopsis
	<i>5tt2gmwmi</i>	Generate a mask image appropriate for seeding streamlines on the grey matter-white matter interface
	<i>5tt2vis</i>	Generate an image for visualisation purposes from an ACT 5TT segmented anatomical image
	<i>5ttcheck</i>	Thoroughly check that one or more images conform to the expected ACT five-tissue-type (5TT) format
	<i>5ttedit</i>	Manually set the partial volume fractions in an ACT five-tissue-type (5TT) image using mask images
	<i>5ttgen</i>	Generate a 5TT image suitable for ACT
	<i>afdconnectivity</i>	Obtain an estimate of fibre connectivity between two regions using AFD and streamlines tractography
	<i>amp2response</i>	Estimate response function coefficients based on the DWI signal in single-fibre voxels
	<i>amp2sh</i>	Convert a set of amplitudes (defined along a set of corresponding directions) to their spherical harmonics
	<i>connectome2tck</i>	Extract streamlines from a tractogram based on their assignment to parcellated nodes
	<i>connectomeedit</i>	Perform basic operations on a connectome
	<i>connectomestats</i>	Connectome group-wise statistics at the edge level using non-parametric permutation testing
	<i>dcmedit</i>	Edit DICOM file in-place
	<i>dcminfo</i>	Output DICOM fields in human-readable format
	<i>dirflip</i>	Invert the polarity of individual directions so as to optimise a unipolar electrostatic repulsion model
	<i>dirgen</i>	Generate a set of uniformly distributed directions using a bipolar electrostatic repulsion model
	<i>dirmerge</i>	Splice / merge multiple sets of directions in such a way as to maintain near-optimality upon truncation
	<i>dirorder</i>	Reorder a set of directions to ensure near-uniformity upon truncation
	<i>dirsplrit</i>	Split a set of evenly distributed directions (as generated by dirgen) into approximately uniformly distributed sets
	<i>dirstat</i>	Report statistics on a direction set
	<i>dwi2adc</i>	Convert mean dwi (trace-weighted) images to mean ADC maps
	<i>dwi2fod</i>	Estimate fibre orientation distributions from diffusion data using spherical deconvolution
	<i>dwi2mask</i>	Generates a whole brain mask from a DWI image
	<i>dwi2response</i>	Estimate response function(s) for spherical deconvolution
	<i>dwi2tensor</i>	Diffusion (kurtosis) tensor estimation
	<i>dwibiascorrect</i>	Perform B1 field inhomogeneity correction for a DWI volume series
	<i>dwicat</i>	Concatenating multiple DWI series accounting for differential intensity scaling
	<i>dwidenoise</i>	dMRI noise level estimation and denoising using Marchenko-Pastur PCA
	<i>dwiextract</i>	Extract diffusion-weighted volumes, b=0 volumes, or certain shells from a DWI dataset
	<i>dwifslpreproc</i>	Perform diffusion image pre-processing using FSL's eddy tool; including inhomogeneity distortion correction
	<i>dwigradcheck</i>	Check the orientation of the diffusion gradient table
	<i>dwinormalise</i>	Perform various forms of intensity normalisation of DWIs

Table 1 – continued from






























Lang	Command	Synopsis
	<i>dwishellmath</i>	Apply an mrmath operation to each b-value shell in a DWI series
	<i>fixel2peaks</i>	Convert data in the fixel directory format into a 4D image of 3-vectors
	<i>fixel2sh</i>	Convert a fixel-based sparse-data image into an spherical harmonic image
	<i>fixel2tsf</i>	Map fixel values to a track scalar file based on an input tractogram
	<i>fixel2voxel</i>	Convert a fixel-based sparse-data image into some form of scalar image
	<i>fixelcfstats</i>	Fixel-based analysis using connectivity-based fixel enhancement and non-parametric permutation tests
	<i>fixelconnectivity</i>	Generate a fixel-fixel connectivity matrix
	<i>fixelconvert</i>	Convert between the old format fixel image (.msf / .msh) and the new fixel directory format
	<i>fixelcorrespondence</i>	Obtain fixel-fixel correspondence between a subject fixel image and a template fixel mask
	<i>fixelcrop</i>	Crop/remove fixels from sparse fixel image using a binary fixel mask
	<i>fixelfilter</i>	Perform filtering operations on fixel-based data
	<i>fixelreorient</i>	Reorient fixel directions
	<i>fod2dec</i>	Generate FOD-based DEC maps, with optional panchromatic sharpening and/or luminance/perceptual sharpening
	<i>fod2fixel</i>	Perform segmentation of continuous Fibre Orientation Distributions (FODs) to produce discrete fixel images
	<i>for_each</i>	Perform some arbitrary processing step for each of a set of inputs
	<i>label2colour</i>	Convert a parcellated image (where values are node indices) into a colour image
	<i>label2mesh</i>	Generate meshes from a label image
	<i>labelconvert</i>	Convert a connectome node image from one lookup table to another
	<i>labelsgmfix</i>	In a FreeSurfer parcellation image, replace the sub-cortical grey matter structure delineations using FSL's <i>bet</i>
	<i>labelstats</i>	Compute statistics of parcels within a label image
	<i>maskdump</i>	Print out the locations of all non-zero voxels in a mask image
	<i>maskfilter</i>	Perform filtering operations on 3D / 4D mask images
	<i>mesh2voxel</i>	Convert a mesh surface to a partial volume estimation image
	<i>meshconvert</i>	Convert meshes between different formats, and apply transformations
	<i>meshfilter</i>	Apply filter operations to meshes
	<i>mraverageheader</i>	Calculate the average (unbiased) coordinate space of all input images
	<i>mrcalc</i>	Apply generic voxel-wise mathematical operations to images
	<i>mrconcat</i>	Concatenate several images into one
	<i>mrcentroid</i>	Determine the centre of mass / centre of gravity of an image
	<i>mrcheckerboardmask</i>	Create bitwise checkerboard image
	<i>mrclusterstats</i>	Voxel-based analysis using permutation testing and threshold-free cluster enhancement

Table 1 – continued from


























Lang	Command	Synopsis
	<i>mrcolour</i>	Apply a colour map to an image
	<i>mrconvert</i>	Perform conversion between different file types and optionally extract a subset of the input image
	<i>mrdegibbs</i>	Remove Gibbs Ringing Artifacts
	<i>mrdump</i>	Print out the values within an image
	<i>mredit</i>	Directly edit the intensities within an image from the command-line
	<i>mrfilter</i>	Perform filtering operations on 3D / 4D MR images
	<i>mrgrid</i>	Modify the grid of an image without interpolation (cropping or padding) or by regridding to an image
	<i>mrhistmatch</i>	Modify the intensities of one image to match the histogram of another
	<i>mrhistogram</i>	Generate a histogram of image intensities
	<i>mrinfo</i>	Display image header information, or extract specific information from the header
	<i>mrmath</i>	Compute summary statistic on image intensities either across images, or along a specified axis of a stack
	<i>mrmetric</i>	Computes a dissimilarity metric between two images
	<i>mrregister</i>	Register two images together using a symmetric rigid, affine or non-linear transformation model
	<i>mrstats</i>	Compute images statistics
	<i>mrthreshold</i>	Create bitwise image by thresholding image intensity
	<i>mrtransform</i>	Apply spatial transformations to an image
	<i>mrtrix_cleanup</i>	Clean up residual temporary files & scratch directories from MRtrix3 commands
	<i>mrview</i>	The MRtrix image viewer
	<i>mtnormalise</i>	Multi-tissue informed log-domain intensity normalisation
	<i>peaks2amp</i>	Extract amplitudes from a peak directions image
	<i>peaks2fixel</i>	Convert peak directions image to a fixel directory
	<i>population_template</i>	Generates an unbiased group-average template from a series of images
	<i>responsemean</i>	Calculate the mean response function from a set of text files
	<i>sh2amp</i>	Evaluate the amplitude of an image of spherical harmonic functions along specified directions
	<i>sh2peaks</i>	Extract the peaks of a spherical harmonic function in each voxel
	<i>sh2power</i>	Compute the total power of a spherical harmonics image
	<i>sh2response</i>	Generate an appropriate response function from the image data for spherical deconvolution
	<i>shbasis</i>	Examine the values in spherical harmonic images to estimate (and optionally change) the SH basis used
	<i>shconv</i>	Perform spherical convolution
	<i>shview</i>	View spherical harmonics surface plots
	<i>tck2connectome</i>	Generate a connectome matrix from a streamlines file and a node parcellation image

Table 1 – continued from

Lang	Command	Synopsis
	<i>tck2fixel</i>	Compute a fixel TDI map from a tractogram
	<i>tckconvert</i>	Convert between different track file formats
	<i>tckdfc</i>	Perform the Track-Weighted Dynamic Functional Connectivity (TW-dFC) method
	<i>tckedit</i>	Perform various editing operations on track files
	<i>tckgen</i>	Perform streamlines tractography
	<i>tckglobal</i>	Multi-Shell Multi-Tissue Global Tractography
	<i>tckinfo</i>	Print out information about a track file
	<i>tckmap</i>	Use track data as a form of contrast for producing a high-resolution image
	<i>tckresample</i>	Resample each streamline in a track file to a new set of vertices
	<i>tcksample</i>	Sample values of an associated image along tracks
	<i>tcksift</i>	Filter a whole-brain fibre-tracking data set such that the streamline densities match the FOD lobe into
	<i>tcksift2</i>	Optimise per-streamline cross-section multipliers to match a whole-brain tractogram to fixel-wise fib
	<i>tckstats</i>	Calculate statistics on streamlines lengths
	<i>tcktransform</i>	Apply a spatial transformation to a tracks file
	<i>tensor2metric</i>	Generate maps of tensor-derived parameters
	<i>transformcalc</i>	Perform calculations on linear transformation matrices
	<i>transformcompose</i>	Compose any number of linear transformations and/or warps into a single transformation
	<i>transformconvert</i>	Convert linear transformation matrices
	<i>tsfdivide</i>	Divide corresponding values in track scalar files
	<i>tsfinfo</i>	Print out information about a track scalar file
	<i>tsfmult</i>	Multiply corresponding values in track scalar files
	<i>tsfsmooth</i>	Gaussian filter a track scalar file
	<i>tsfthreshold</i>	Threshold and invert track scalar files
	<i>tsfvalidate</i>	Validate a track scalar file against the corresponding track data
	<i>vectorstats</i>	Statistical testing of vector data using non-parametric permutation testing
	<i>voxel2fixel</i>	Map the scalar value in each voxel to all fixels within that voxel
	<i>voxel2mesh</i>	Generate a surface mesh representation from a voxel image
	<i>warp2metric</i>	Compute fixel-wise or voxel-wise metrics from a 4D deformation field
	<i>warpconvert</i>	Convert between different representations of a non-linear warp
	<i>warpcorrect</i>	Replaces voxels in a deformation field that point to a specific out of bounds location with nan,nan,na
	<i>warpinit</i>	Create an initial warp image, representing an identity transformation
	<i>warpinvert</i>	Invert a non-linear warp field

5.40 List of MRtrix3 configuration file options

AmbientIntensity

default: 0.5

The default intensity for the ambient light in OpenGL renders.

AnalyseLeftToRight

default: 0 (false)

A boolean value to indicate whether images in Analyse format should be assumed to be in LAS orientation (default) or RAS (when this option is turned on).

BValueEpsilon

default: 80.0

Specifies the difference between b-values necessary for image volumes to be classified as belonging to different shells.

BZeroThreshold

default: 10.0

Specifies the b-value threshold for determining those image volumes that correspond to $b=0$.

BackgroundColor

default: 1.0,1.0,1.0

The default colour to use for the background in OpenGL panels, notably the SH viewer.

ConnectomeEdgeAssociatedAlphaMultiplier

default: 1.0

The multiplicative factor to apply to the transparency of edges connected to one selected node.

ConnectomeEdgeAssociatedColour

default: 0.0,0.0,0.0

The colour mixed in to edges connected to one currently selected node.

ConnectomeEdgeAssociatedColourFade

default: 0.5

The fraction of the colour of an edge connected to one selected node determined by the fixed colour.

ConnectomeEdgeAssociatedSizeMultiplier

default: 1.0

The multiplicative factor to apply to the size of edges connected to one selected node.

ConnectomeEdgeOtherAlphaMultiplier

default: 1.0

The multiplicative factor to apply to the transparency of edges not connected to any selected node.

ConnectomeEdgeOtherColour

default: 0.0,0.0,0.0

The colour mixed in to edges not connected to any currently selected node.

ConnectomeEdgeOtherColourFade

default: 0.75

The fraction of the colour of an edge not connected to any selected node determined by the fixed colour.

ConnectomeEdgeOtherSizeMultiplier*default: 1.0*

The multiplicative factor to apply to the size of edges not connected to any selected node.

ConnectomeEdgeOtherVisibilityOverride*default: true*

Whether or not to force invisibility of edges not connected to any selected node.

ConnectomeEdgeSelectedAlphaMultiplier*default: 1.0*

The multiplicative factor to apply to the transparency of edges connected to two selected nodes.

ConnectomeEdgeSelectedColour*default: 0.9,0.9,1.0*

The colour used to highlight the edges connected to two currently selected nodes.

ConnectomeEdgeSelectedColourFade*default: 0.5*

The fraction of the colour of an edge connected to two selected nodes determined by the fixed selection highlight colour.

ConnectomeEdgeSelectedSizeMultiplier*default: 1.0*

The multiplicative factor to apply to the size of edges connected to two selected nodes.

ConnectomeEdgeSelectedVisibilityOverride*default: false*

Whether or not to force visibility of edges connected to two selected nodes.

ConnectomeNodeAssociatedAlphaMultiplier*default: 1.0*

The multiplicative factor to apply to the transparency of nodes associated with a selected node.

ConnectomeNodeAssociatedColour*default: 0.0,0.0,0.0*

The colour mixed in to those nodes associated with any selected node.

ConnectomeNodeAssociatedColourFade*default: 0.5*

The fraction of the colour of an associated node determined by the fixed associated highlight colour.

ConnectomeNodeAssociatedSizeMultiplier*default: 1.0*

The multiplicative factor to apply to the size of nodes associated with a selected node.

ConnectomeNodeOtherAlphaMultiplier*default: 1.0*

The multiplicative factor to apply to the transparency of nodes not currently selected nor associated with a selected node.

ConnectomeNodeOtherColour*default: 0.0,0.0,0.0*

The colour mixed in to those nodes currently not selected nor associated with any selected node.

ConnectomeNodeOtherColourFade*default: 0.75*

The fraction of the colour of an unselected, non-associated node determined by the fixed not-selected highlight colour.

ConnectomeNodeOtherSizeMultiplier*default: 1.0*

The multiplicative factor to apply to the size of nodes not currently selected nor associated with a selected node.

ConnectomeNodeOtherVisibilityOverride*default: false*

Whether or not nodes are forced to be invisible when not selected or associated with any selected node.

ConnectomeNodeSelectedAlphaMultiplier*default: 1.0*

The multiplicative factor to apply to the transparency of selected nodes.

ConnectomeNodeSelectedColour*default: 1.0,1.0,1.0*

The colour used to highlight those nodes currently selected.

ConnectomeNodeSelectedColourFade*default: 0.75*

The fraction of the colour of a selected node determined by the fixed selection highlight colour.

ConnectomeNodeSelectedSizeMultiplier*default: 1.0*

The multiplicative factor to apply to the size of selected nodes.

ConnectomeNodeSelectedVisibilityOverride*default: true*

Whether or not nodes are forced to be visible when selected.

DiffuseIntensity*default: 0.5*

The default intensity for the diffuse light in OpenGL renders.

FailOnWarn*default: 0 (false)*

A boolean value specifying whether MRtrix applications should abort as soon as any (otherwise non-fatal) warning is issued.

FontSize*default: 10*

The size (in points) of the font to be used in OpenGL viewports (mrview and shview).

HelpCommand*default: less*

The command to use to display each command's help page (leave empty to send directly to the terminal).

IconSize*default: 30*

The size of the icons in the main MRView toolbar.

ImageInterpolation*default: true*

Define default interpolation setting for image and image overlay.

InitialToolBarPosition*default: top*

The starting position of the MRView toolbar. Valid values are: top, bottom, left, right.

LightPosition*default: 1.0,1.0,3.0*

The default position vector to use for the light in OpenGL renders.

MRViewColourBarHeight*default: 100*

The height of the colourbar in MRView, in pixels.

MRViewColourBarHorizontalPadding*default: 100*

The width in pixels between horizontally adjacent colour bars.

MRViewColourBarInset*default: 20*

How far away from the edge of the main window to place the colourbar in MRView, in pixels.

MRViewColourBarPosition*default: bottomright*

The position of the colourbar within the main window in MRView. Valid values are: bottomleft, bottomright, topleft, topright.

MRViewColourBarTextOffset*default: 10*

How far away from the colourbar to place the associated text, in pixels.

MRViewColourBarWidth*default: 20*

The width of the colourbar in MRView, in pixels.

MRViewDefaultTractGeomType*default: Pseudotubes*

The default geometry type used to render tractograms. Options are Pseudotubes, Lines or Points

MRViewDockFloating*default: 0 (false)*

Whether MRView tools should start docked in the main window, or floating (detached from the main window).

MRViewFocusModifierKey*default: meta (cmd on MacOSX)*

Modifier key to select focus mode in MRView. Valid choices include shift, alt, ctrl, meta (on MacOSX: shift, alt, ctrl, cmd).

MRViewImageBackgroundColour

default: 0,0,0 (black)

The default image background colour in the main MRView window.

MRViewInitWindowSize

default: 512,512

Initial window size of MRView in pixels.

MRViewMaxNumColourBarRows

default: 3

The maximal number of rows used to layout a collection of rendered colourbars Note, that all tool-specific colourbars will form a single collection.

MRViewMoveModifierKey

default: shift

Modifier key to select move mode in MRView. Valid choices include shift, alt, ctrl, meta (on MacOSX: shift, alt, ctrl, cmd).

MRViewOdfScale

default: 1.0

The factor by which the ODF overlay is scaled.

MRViewOrthoAsRow

default: false

Display the 3 orthogonal views of the Ortho mode in a row, rather than as a 2x2 montage

MRViewRoiAlpha

default: 0.5

The default alpha of a ROI overlay.

MRViewRotateModifierKey

default: ctrl

Modifier key to select rotate mode in MRView. Valid choices include shift, alt, ctrl, meta (on MacOSX: shift, alt, ctrl, cmd).

MRViewShowColourbar

default: true

Colourbar shown in main image overlay.

MRViewShowComments

default: true

Comments shown in main image overlay.

MRViewShowFocus

default: true

Focus cross hair shown in main image.

MRViewShowOrientationLabel

default: true

Anatomical orientation information shown in main image overlay.

MRViewShowVoxelInformation

default: true

Voxel information shown in main image overlay.

MRViewSyncFocus

default: false

Whether to sync the focus in mrview between other mrview processes.

MRViewToolFontSize

default: 2 points less than the standard system font

The point size for the font to use in MRView tools.

MRViewToolsColourBarPosition

default: topright

The position of all visible tool colourbars within the main window in MRView. Valid values are: bottomleft, bottomright, topleft, topright.

MRViewWrapVolumes

default: false

Wrap volumes around when cycling through

MSAA

default: 0 (false)

How many samples to use for multi-sample anti-aliasing (to improve display quality).

NiftIAllowBitwise

default: 0 (false)

A boolean value to indicate whether bitwise storage of binary data is permitted (most 3rd party software packages don't support bitwise data). If false (the default), data will be stored using more widely supported unsigned 8-bit integers.

NiftIAlwaysUseVer2

default: 0 (false)

A boolean value to indicate whether NiftI images should always be written in the new NiftI-2 format. If false, images will be written in the older NiftI-1 format by default, with the exception being files where the number of voxels along any axis exceeds the maximum permissible in that format (32767), in which case the output file will automatically switch to the NiftI-2 format.

NiftIAutoLoadJSON

default: 0 (false)

A boolean value to indicate whether, when opening NiftI images, any corresponding JSON file should be automatically loaded.

NiftIAutoSaveJSON

default: 0 (false)

A boolean value to indicate whether, when writing NiftI images, a corresponding JSON file should be automatically created in order to save any header entries that cannot be stored in the NiftI header.

NiftIUseSform

default: 1 (true)

A boolean value to control whether, in cases where both the sform and qform transformations are defined in an input NiftI image, but those transformations differ, the sform transformation should

be used in preference to the qform matrix. The default is to use the sform matrix; set to 0 / false to override and instead use the qform.

NeedOpenGLCoreProfile

default: 1 (true)

Whether the creation of an OpenGL 3.3 context requires it to be a core profile (needed on newer versions of the ATI drivers on Linux, for instance).

NumberOfThreads

default: number of threads provided by hardware

Set the default number of CPU threads to use for multi-threading.

NumberOfUndos

default: 16

The number of undo operations permitted in the MRView ROI editor tool.

ObjectColor

default: 1,1,0 (yellow)

The default colour to use for objects (i.e. SH glyphs) when not colouring by direction.

RealignTransform

default: 1 (true)

A boolean value to indicate whether all images should be realigned to an approximately axial orientation at load.

RegAnalyseDescent

default: 0 (false)

Linear registration: write comma separated gradient descent parameters and gradients to stdout and verbose gradient descent output to stderr.

RegCoherenceLen

default: 3.0

Linear registration: estimated spatial coherence length in voxels.

RegGdConvergenceDataSmooth

default: 0.8

Linear registration: control point trajectory smoothing value used in convergence check parameter range: [0...1].

RegGdConvergenceMinIter

default: 10

Linear registration: minimum number of iterations until convergence check is activated.

RegGdConvergenceSlopeSmooth

default: 0.1

Linear registration: control point trajectory slope smoothing value used in convergence check parameter range: [0...1].

RegGdConvergenceThresh

default: 5e-3

Linear registration: threshold for convergence check using the smoothed control point trajectories measured in fraction of a voxel.

RegGdWeightMatrix*default: 0.0003*

Linear registration: weight for optimisation of linear (3x3) matrix parameters.

RegGdWeightTranslation*default: 1*

Linear registration: weight for optimisation of translation parameters.

RegStopLen*default: 0.0001*

Linear registration: smallest gradient descent step measured in fraction of a voxel at which to stop registration.

ScriptScratchDir*default: '.'*

The location in which to generate the scratch directories to be used by MRtrix Python scripts. By default they will be generated in the working directory. Note that this setting does not influence the location in which piped images and other temporary files are created by MRtrix3; that is determined based on config file option *TmpFileDir*.

ScriptScratchPrefix*default: '<script>-tmp-'*

The prefix to use when generating a unique name for a Python script scratch directory. By default the name of the invoked script itself will be used, followed by *-tmp-* (six random characters are then appended to produce a unique name in cases where a script may be run multiple times in parallel).

SparseDataInitialSize*default: 16777216*

Initial buffer size for data in MRtrix sparse image format file (in bytes).

SpecularExponent*default: 5.0*

The default exponent for the specular light in OpenGL renders.

SpecularIntensity*default: 0.5*

The default intensity for the specular light in OpenGL renders.

TckgenEarlyExit*default: 0 (false)*

Specifies whether tckgen should be terminated prematurely in cases where it appears as though the target number of accepted streamlines is not going to be met.

TerminalColor*default: 1 (true)*

A boolean value to indicate whether colours should be used in the terminal.

TmpFileDir*default: '/tmp' (on Unix), '.' (on Windows)*

The prefix for temporary files (as used in pipelines). By default, these files get written to the current folder on Windows machines, which may cause performance issues, particularly when operating over distributed file systems. On Unix machines, the default is */tmp/*, which is typically a RAM file system

and should therefore be fast; but may cause issues on machines with little RAM capacity or where write-access to this location is not permitted.

Note that this location can also be manipulated using the `MRTRIX_TMPFILE_DIR` environment variable, without editing the config file. Note also that this setting does not influence the location in which Python scripts construct their scratch directories; that is determined based on config file option `ScriptScratchDir`.

TmpFilePrefix

default: 'mrtrix-tmp-'

The prefix to use for the basename of temporary files. This will be used to generate a unique filename for the temporary file, by adding random characters to this prefix, followed by a suitable suffix (depending on file type). Note that this prefix can also be manipulated using the `MRTRIX_TMPFILE_PREFIX` environment variable, without editing the config file.

ToolbarStyle

default: 2

The style of the main toolbar buttons in MRView. See Qt's documentation for `Qt::ToolButtonStyle`.

TrackWriterBufferSize

default: 16777216

The size of the write-back buffer (in bytes) to use when writing track files. MRtrix will store the output tracks in a relatively large buffer to limit the number of `write()` calls, avoid associated issues such as file fragmentation.

VSync

default: 0 (false)

Whether the screen update should synchronise with the monitor's vertical refresh (to avoid tearing artefacts).

5.41 List of MRtrix3 environment variables

DICOM_ID

when reading DICOM data, match the PatientID entry against the string provided

DICOM_PATIENT

when reading DICOM data, match the PatientName entry against the string provided

DICOM_SERIES

when reading DICOM data, match the SeriesName entry against the string provided

DICOM_STUDY

when reading DICOM data, match the StudyName entry against the string provided

MRTRIX_CONFIGFILE

This can be used to set the location of the system-wide configuration file. By default, this is `/etc/mrtrix.conf`. This can be useful for deployments where access to the system's `/etc` folder is problematic, or to allow different versions of the software to have different configurations, etc.

MRTRIX_NOSIGNALS

If this variable is set to any value, disable MRtrix3's custom signal handlers. This may sometimes be useful when debugging. Note however that this prevents the deletion of temporary files when the command terminates abnormally.

MRTRIX_NTHREADS

set the number of threads that MRtrix3 applications should use. This overrides the automatically determined

number, or the `NumberOfThreads` setting in the configuration file, but will be overridden by the ENVVAR `-nthreads` command-line option.

MRTRIX_QUIET

Do not display information messages or progress status. This has the same effect as the `-quiet` command-line option.

MRTRIX_RNG_SEED

Set the seed used for the random number generator. Ordinarily, MRtrix applications will use random seeds to ensure repeat runs of stochastic processes are never the same. However, when experimenting or debugging, it may be useful to explicitly set the RNG seed to ensure reproducible results across runs. To do this, set this variable to a fixed number prior to running the command(s).

Note that to obtain the same results from a multi-threaded command, you should also disable multi-threading (using the option `-nthread 0` or by setting the `MRTRIX_NTHREADS` environment variable to zero). Multi-threading introduces randomness in the order of execution, which will generally also affect the reproducibility of results.

MRTRIX_TMPFILE_DIR

This has the same effect as the `TmpFileDir` configuration file entry, and can be used to set the location of temporary files (as used in Unix pipes) for a single session, within a single script, or for a single command without modifying the configuration file.

MRTRIX_TMPFILE_PREFIX

This has the same effect as the `TmpFilePrefix` configuration file entry, and can be used to set the prefix for the name of temporary files (as used in Unix pipes) for a single session, within a single script, or for a single command without modifying the configuration file.

5.42 MRtrix 0.2 equivalent commands

For those users moving to *MRtrix3* from the old MRtrix 0.2.x software, this list provides the equivalent command names for the functionalities that they are accustomed to from the older version of the software. The new command naming scheme was carefully designed, so we hope people agree that it makes sense, and allows users to easily find the command functionalities available that are relevant for the data they are processing.

Further information can be found on these commands either through the [documentation](#), or by typing the binary name at the command-line with no arguments to access the help file for that command.

MRtrix 0.2.x	MRtrix3	Comments
<code>average</code>	<code>mrmath</code>	Use mean statistic as second argument, and <code>-axis</code> option
<code>cat_tracks</code>	<code>tckedit</code>	Simply provide multiple input track files to the command
<code>cleanup_ANTs_warp</code>	<i>no equivalent</i>	
<code>csdeconv</code>	<code>dwi2fod</code>	
<code>dicom_filename</code>	<i>no equivalent</i>	See <code>dcminfo</code>
<code>dir2amp</code>	<code>peaks2amp</code>	
<code>disp_profile</code>	<code>shview</code>	
<code>dwi2SH</code>	<code>amp2sh</code>	
<code>dwi2tensor</code>	<code>dwi2tensor</code>	
<code>erode</code>	<code>maskfilter</code>	Specify <code>erode</code> or <code>dilate</code> algorithm as second argument
<code>estimate_response</code>	<code>dwi2response</code> manual	The <code>dwi2response</code> script also has a number of algorithms for automatically selecting
<code>filter_tracks</code>	<code>tckedit</code>	
<code>find_SH_peaks</code>	<code>sh2peaks</code>	
<code>gen_ROI</code>	<i>no equivalent</i>	

Table 2 – con

MRtrix 0.2.x	MRtrix3	Comments
gen_WM_mask	<i>no equivalent</i>	Use of this command was discouraged and so it has been discontinued
gen_unit_warp	<i>warpinit</i>	
gendir	<i>dirgen</i>	The electrostatic repulsion algorithm now only uses an exponent of 2 by default,
import_tracks	<i>tckconvert</i>	
median3D	<i>mrfilter</i>	Specify median algorithm as second argument
mrabs	<i>mrcalc</i>	Use <code>-abs</code> operator
mradd	<i>mrcalc or mrmath</i>	E.g. <code>mrcalc A.mif B.mif -add out.mif</code> or <code>mrmath A.mif B.mif -add out.mif</code>
mrconcat	<i>mrconcat</i>	
mrconvert	<i>mrconvert</i>	
mrinfo	<i>mrinfo</i>	
mrmult	<i>mrcalc or mrmath</i>	E.g. <code>mrcalc A.mif B.mif -mult out.mif</code> or <code>mrmath A.mif B.mif -mult out.mif</code>
mrstats	<i>mrstats</i>	
mrtransform	<i>mrtransform</i>	
mrview	<i>mrview</i>	
normalise_tracks	<i>tcktransform</i>	
read_dicom	<i>dcminfo</i>	
read_ximg	<i>no equivalent</i>	
resample_tracks	<i>tcksample</i>	
sample_tracks	<i>tcksample</i>	
sdeconv	<i>dwi2fod</i>	
select_tracks	<i>tckedit</i>	
streamtrack	<i>tckgen</i>	
tensor2ADC	<i>tensor2metric</i>	Use <code>-adc</code> output option
tensor2FA	<i>tensor2metric</i>	Use <code>-fa</code> output option
tensor2vector	<i>tensor2metric</i>	Use <code>-vector</code> output option
tensor_metric	<i>tensor2metric</i>	
threshold	<i>mrthreshold</i>	Note that automatic threshold parameter determination (i.e. if you don't explicitly
track_info	<i>tckinfo</i>	
tracks2prob	<i>tckmap</i>	
tracks2vtk	<i>tckconvert</i>	
truncate_tracks	<i>tckedit</i>	

5.43 References

Bibliography

- [Christiaens2015] D. Christiaens, M. Reisert, T. Dhollander, S. Sunaert, P. Suetens, and F. Maes. *Global tractography of multi-shell diffusion-weighted imaging data using a multi-tissue model*. NeuroImage, 123 (2015), pp. 89–101. [[full text link](#)]
- [CorderoGrande2019] L. Cordero-Grande, D. Christiaens, J. Hutter, A.N. Price, J.V. Hajnal. *Complex diffusion-weighted image estimation via matrix recovery under general noise models*. NeuroImage 200 (2019), pp. 391–404 [[full text link](#)]
- [Dhollander2014] T. Dhollander, L. Emsell, W. Van Hecke, F. Maes, S. Sunaert, and P. Suetens. *Track Orientation Density Imaging (TODI) and Track Orientation Distribution (TOD) based tractography*. NeuroImage, 94 (2014), pp. 312–336. [[full text link](#)]
- [Dhollander2016a] T. Dhollander, D. Raffelt, and A. Connelly. *A novel iterative approach to reap the benefits of multi-tissue CSD from just single-shell ($+b=0$) diffusion MRI data*. Proceedings of the 24th annual meeting of the International Society of Magnetic Resonance in Medicine (2016), pp. 3010. [[full text link](#)]
- [Dhollander2016b] T. Dhollander, D. Raffelt, and A. Connelly. *Unsupervised 3-tissue response function estimation from single-shell or multi-shell diffusion MR data without a co-registered T1 image*. ISMRM Workshop on Breaking the Barriers of Diffusion MRI (2016), pp. 5. [[full text link](#)]
- [Dhollander2017] T. Dhollander, D. Raffelt, and A. Connelly. *Towards interpretation of 3-tissue constrained spherical deconvolution results in pathology*. Proceedings of the 25th annual meeting of the International Society of Magnetic Resonance in Medicine (2017), pp. 1815. [[full text link](#)]
- [Dhollander2018a] T. Dhollander, D. Raffelt, and A. Connelly. *Accuracy of response function estimation algorithms for 3-tissue spherical deconvolution of diverse quality diffusion MRI data*. Proceedings of the 26th annual meeting of the International Society of Magnetic Resonance in Medicine (2018), pp. 1569. [[full text link](#)]
- [Dhollander2018b] T. Dhollander, J. Zanin, B.A. Nayagam, G. Rance, and A. Connelly. *Feasibility and benefits of 3-tissue constrained spherical deconvolution for studying the brains of babies*. Proceedings of the 26th annual meeting of the International Society of Magnetic Resonance in Medicine (2018), pp. 3077. [[full text link](#)]
- [Dhollander2019] T. Dhollander, R. Mito, D. Raffelt, and A. Connelly. *Improved white matter response function estimation for 3-tissue constrained spherical deconvolution*. Proceedings of the 27th annual meeting of the International Society of Magnetic Resonance in Medicine (2019), pp. 555. [[full text link](#)]
- [Jeurissen2014] B. Jeurissen, J.-D. Tournier, T. Dhollander, A. Connelly, and J. Sijbers. *Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data*. NeuroImage, 103 (2014), pp. 411–426. [[full text link](#)]

- [Mito2018a] R. Mito, D. Raffelt, T. Dhollander, D.N. Vaughan, J.-D. Tournier, O. Salvado, A. Brodtmann, C.C. Rowe, V.L. Villemagne, and A. Connelly. *Fibre-specific white matter reductions in Alzheimer's disease and mild cognitive impairment*. Brain, 141(3) (2018), pp. 888–902. [\[full text link\]](#)
- [Mito2018b] R. Mito, T. Dhollander, D. Raffelt, Y. Xia, O. Salvado, A. Brodtmann, C.C. Rowe, V.L. Villemagne, and A. Connelly. *Investigating microstructural heterogeneity of white matter hyperintensities in Alzheimer's disease using single-shell 3-tissue constrained spherical deconvolution*. Proceedings of the 26th annual meeting of the International Society of Magnetic Resonance in Medicine (2018), pp. 135. [\[full text link\]](#)
- [Raffelt2011] D. Raffelt, J.-D. Tournier, J. Fripp, S Crozier, A. Connelly, O. Salvado. *Symmetric diffeomorphic registration of fibre orientation distributions*. NeuroImage 56 (2011), pp. 1171–1180. [\[full text link\]](#)
- [Raffelt2012] D. Raffelt, J.-D. Tournier, S. Rose, G.R. Ridgway, R. Henderson, S. Crozier, O. Salvado, A. Connelly. *Apparent Fibre Density: a novel measure for the analysis of diffusion-weighted magnetic resonance images*. NeuroImage 59 (2012), pp. 3976–3994. [\[full text link\]](#)
- [Raffelt2015] D.A. Raffelt, R.E. Smith, G.R. Ridgway, J.-D. Tournier, D.N. Vaughan, S. Rose, R. Henderson, A. Connelly. *Connectivity-Based Fixel Enhancement: Whole-Brain Statistical Analysis of Diffusion MRI Measures in the Presence of Crossing Fibres*. NeuroImage 117 (2015), pp. 40–55. [\[full text link\]](#)
- [Raffelt2017] D.A. Raffelt, J.-D. Tournier, R.E. Smith, D.N. Vaughan, G. Jackson, G.R. Ridgway, A. Connelly. *Investigating White Matter Fibre Density and Morphology using Fixel-Based Analysis*. NeuroImage, 144 (2017), pp. 58-73. [\[full text link\]](#)
- [Reisert2011] M. Reisert, I. Mader, C. Anastasopoulos, M. Weigel, S. Schnell, and V. Kiselev. *Global fiber reconstruction becomes practical*. NeuroImage, 54 (2011) pp. 955–962. [\[full text link\]](#)
- [Smith2012] R.E. Smith, J.-D. Tournier, F. Calamante, A. Connelly. *Anatomically-constrained tractography: improved diffusion MRI streamlines tractography through effective use of anatomical information*. NeuroImage 62 (2012), pp. 1924–1938. [\[full text link\]](#)
- [Smith2013] R.E. Smith, J.-D. Tournier, F. Calamante, A. Connelly. *SIFT: Spherical-deconvolution informed filtering of tractograms*. NeuroImage 67 (2013), pp. 298–312. [\[full text link\]](#)
- [Smith2015] R.E. Smith, J.-D. Tournier, F. Calamante, A. Connelly. *SIFT2: Enabling dense quantitative assessment of brain white matter connectivity using streamlines tractography*. NeuroImage 119 (2015), pp. 338-51. [\[full text link\]](#)
- [Smith2019a] R.E. Smith, D. Dimond, S. Bray, A. Connelly. *Mitigation of DWI brain cropping in Fixel-Based Analysis*. In Proc OHBM (2019), W765 [\[full text link\]](#)
- [Tax2014] C.M.W. Tax, B. Jeurissen, S.B.Vos, M.A. Viergever, and A. Leemans. *Recursive calibration of the fiber response function for spherical deconvolution of diffusion MRI data*. NeuroImage, 86 (2014), pp. 67–80. [\[full text link\]](#)
- [Tournier2004] J.-D. Tournier, F. Calamante, D.G. Gadian, and A. Connelly. *Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution*. NeuroImage, 23 (2004), pp. 1176–85. [\[full text link\]](#)
- [Tournier2007] J.-D. Tournier, F. Calamante, and A. Connelly. *Robust determination of the fibre orientation distribution in diffusion MRI: non-negativity constrained super-resolved spherical deconvolution*. Neuroimage, 35 (2007), pp. 1459–72. [\[full text link\]](#)
- [Tournier2012] J.-D. Tournier, F. Calamante, A. Connelly. *MRtrix: Diffusion tractography in crossing fiber regions*. INT J IMAG SYST TECH, 22 (2012), pp. 53-66. [\[full text link\]](#)
- [Tournier2013] J.-D. Tournier, F. Calamante, and A. Connelly. *Determination of the appropriate b value and number of gradient directions for high-angular-resolution diffusion-weighted imaging*. NMR Biomed., 26 (2013), pp. 1775–86. [\[full text link\]](#)

- [Tournier2019] J.-D. Tournier, R. E. Smith, D. Raffelt, R. Tabbara, T. Dhollander, M. Pietsch, D. Christiaens, B. Jeurissen, C.-H. Yeh, and A. Connelly. *MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation*. *NeuroImage*, 202 (2019), pp. 116–37. [[fulltext link](#)]
- [Veraart2016a] J. Veraart, E. Fieremans, and D.S. Novikov. *Diffusion MRI noise mapping using random matrix theory*. *Magn. Res. Med.* 76(5) (2016), pp. 1582–1593. [[full text link](#)]
- [Veraart2016b] J. Veraart, D.S. Novikov, D. Christiaens, B. Ades-aron, J. Sijbers, and E. Fieremans *Denoising of diffusion MRI using random matrix theory*. *NeuroImage* 142 (2016), pp. 394–406. [[full text link](#)]

A

AmbientIntensity
 command line option, 369
AnalyseLeftToRight
 command line option, 369

B

BackgroundColor
 command line option, 369
BValueEpsilon
 command line option, 369
BZeroThreshold
 command line option, 369

C

command line option
 AmbientIntensity, 369
 AnalyseLeftToRight, 369
 BackgroundColor, 369
 BValueEpsilon, 369
 BZeroThreshold, 369
 ConnectomeEdgeAssociatedAlphaMultiplier, 369
 ConnectomeEdgeAssociatedColour, 369
 ConnectomeEdgeAssociatedColourFade, 369
 ConnectomeEdgeAssociatedSizeMultiplier, 369
 ConnectomeEdgeOtherAlphaMultiplier, 369
 ConnectomeEdgeOtherColour, 369
 ConnectomeEdgeOtherColourFade, 369
 ConnectomeEdgeOtherSizeMultiplier, 369
 ConnectomeEdgeOtherVisibilityOverride, 370
 ConnectomeEdgeSelectedAlphaMultiplier, 370
 ConnectomeEdgeSelectedColour, 370
 ConnectomeEdgeSelectedColourFade, 370
 ConnectomeEdgeSelectedSizeMultiplier, 370
 ConnectomeEdgeSelectedVisibilityOverride, 370
 ConnectomeNodeAssociatedAlphaMultiplier, 370
 ConnectomeNodeAssociatedColour, 370
 ConnectomeNodeAssociatedColourFade, 370
 ConnectomeNodeAssociatedSizeMultiplier, 370
 ConnectomeNodeOtherAlphaMultiplier, 370
 ConnectomeNodeOtherColour, 370
 ConnectomeNodeOtherColourFade, 370
 ConnectomeNodeOtherSizeMultiplier, 371
 ConnectomeNodeOtherVisibilityOverride, 371
 ConnectomeNodeSelectedAlphaMultiplier, 371
 ConnectomeNodeSelectedColour, 371
 ConnectomeNodeSelectedColourFade, 371
 ConnectomeNodeSelectedSizeMultiplier, 371
 ConnectomeNodeSelectedVisibilityOverride, 371
 DiffuseIntensity, 371
 FailOnWarn, 371
 FontSize, 371
 HelpCommand, 371
 IconSize, 371
 ImageInterpolation, 372
 InitialToolBarPosition, 372
 LightPosition, 372
 MRViewColourBarHeight, 372
 MRViewColourBarHorizontalPadding,

- 372
- MRViewColourBarInset, 372
- MRViewColourBarPosition, 372
- MRViewColourBarTextOffset, 372
- MRViewColourBarWidth, 372
- MRViewDefaultTractGeomType, 372
- MRViewDockFloating, 372
- MRViewFocusModifierKey, 372
- MRViewImageBackgroundColour, 373
- MRViewInitWindowSize, 373
- MRViewMaxNumColourBarRows, 373
- MRViewMoveModifierKey, 373
- MRViewOdfScale, 373
- MRViewOrthoAsRow, 373
- MRViewRoiAlpha, 373
- MRViewRotateModifierKey, 373
- MRViewShowColourbar, 373
- MRViewShowComments, 373
- MRViewShowFocus, 373
- MRViewShowOrientationLabel, 373
- MRViewShowVoxelInformation, 373
- MRViewSyncFocus, 374
- MRViewToolFontSize, 374
- MRViewToolsColourBarPosition, 374
- MRViewWrapVolumes, 374
- MSAA, 374
- NeedOpenGLCoreProfile, 375
- NIfTIAAllowBitwise, 374
- NIfTIAAlwaysUseVer2, 374
- NIfTIAutoLoadJSON, 374
- NIfTIAutoSaveJSON, 374
- NIfTIUseSform, 374
- NumberOfThreads, 375
- NumberOfUndos, 375
- ObjectColor, 375
- RealignTransform, 375
- RegAnalyseDescent, 375
- RegCoherenceLen, 375
- RegGdConvergenceDataSmooth, 375
- RegGdConvergenceMinIter, 375
- RegGdConvergenceSlopeSmooth, 375
- RegGdConvergenceThresh, 375
- RegGdWeightMatrix, 375
- RegGdWeightTranslation, 376
- RegStopLen, 376
- ScriptScratchDir, 376
- ScriptScratchPrefix, 376
- SparseDataInitialSize, 376
- SpecularExponent, 376
- SpecularIntensity, 376
- TckgenEarlyExit, 376
- TerminalColor, 376
- TmpFileDir, 376
- TmpFilePrefix, 377
- ToolbarStyle, 377
- TrackWriterBufferSize, 377
- VSync, 377
- ConnectomeEdgeAssociatedAlphaMultiplier
 - command line option, 369
- ConnectomeEdgeAssociatedColour
 - command line option, 369
- ConnectomeEdgeAssociatedColourFade
 - command line option, 369
- ConnectomeEdgeAssociatedSizeMultiplier
 - command line option, 369
- ConnectomeEdgeOtherAlphaMultiplier
 - command line option, 369
- ConnectomeEdgeOtherColour
 - command line option, 369
- ConnectomeEdgeOtherColourFade
 - command line option, 369
- ConnectomeEdgeOtherSizeMultiplier
 - command line option, 369
- ConnectomeEdgeOtherVisibilityOverride
 - command line option, 370
- ConnectomeEdgeSelectedAlphaMultiplier
 - command line option, 370
- ConnectomeEdgeSelectedColour
 - command line option, 370
- ConnectomeEdgeSelectedColourFade
 - command line option, 370
- ConnectomeEdgeSelectedSizeMultiplier
 - command line option, 370
- ConnectomeEdgeSelectedVisibilityOverride
 - command line option, 370
- ConnectomeNodeAssociatedAlphaMultiplier
 - command line option, 370
- ConnectomeNodeAssociatedColour
 - command line option, 370
- ConnectomeNodeAssociatedColourFade
 - command line option, 370
- ConnectomeNodeAssociatedSizeMultiplier
 - command line option, 370
- ConnectomeNodeOtherAlphaMultiplier
 - command line option, 370
- ConnectomeNodeOtherColour
 - command line option, 370
- ConnectomeNodeOtherColourFade
 - command line option, 370
- ConnectomeNodeOtherSizeMultiplier
 - command line option, 371
- ConnectomeNodeOtherVisibilityOverride
 - command line option, 371
- ConnectomeNodeSelectedAlphaMultiplier
 - command line option, 371
- ConnectomeNodeSelectedColour
 - command line option, 371
- ConnectomeNodeSelectedColourFade

command line option, [371](#)
 ConnectomeNodeSelectedSizeMultiplier
 command line option, [371](#)
 ConnectomeNodeSelectedVisibilityOverride
 command line option, [371](#)

D

DiffuseIntensity
 command line option, [371](#)

E

environment variable
 DICOM_ID, [377](#)
 DICOM_PATIENT, [377](#)
 DICOM_SERIES, [377](#)
 DICOM_STUDY, [377](#)
 MRTRIX_CONFIGFILE, [377](#)
 MRTRIX_NOSIGNALS, [377](#)
 MRTRIX_NTHREADS, [377](#), [378](#)
 MRTRIX_QUIET, [378](#)
 MRTRIX_RNG_SEED, [378](#)
 MRTRIX_TMPFILE_DIR, [377](#), [378](#)
 MRTRIX_TMPFILE_PREFIX, [378](#)

F

FailOnWarn
 command line option, [371](#)
 FontSize
 command line option, [371](#)

H

HelpCommand
 command line option, [371](#)

I

IconSize
 command line option, [371](#)
 ImageInterpolation
 command line option, [372](#)
 InitialToolBarPosition
 command line option, [372](#)

L

LightPosition
 command line option, [372](#)

M

MRTRIX_NTHREADS, [378](#)
 MRTRIX_TMPFILE_DIR, [377](#)
 MRViewColourBarHeight
 command line option, [372](#)
 MRViewColourBarHorizontalPadding
 command line option, [372](#)
 MRViewColourBarInset

command line option, [372](#)
 MRViewColourBarPosition
 command line option, [372](#)
 MRViewColourBarTextOffset
 command line option, [372](#)
 MRViewColourBarWidth
 command line option, [372](#)
 MRViewDefaultTractGeomType
 command line option, [372](#)
 MRViewDockFloating
 command line option, [372](#)
 MRViewFocusModifierKey
 command line option, [372](#)
 MRViewImageBackgroundColour
 command line option, [373](#)
 MRViewInitWindowSize
 command line option, [373](#)
 MRViewMaxNumColourBarRows
 command line option, [373](#)
 MRViewMoveModifierKey
 command line option, [373](#)
 MRViewOdfScale
 command line option, [373](#)
 MRViewOrthoAsRow
 command line option, [373](#)
 MRViewRoiAlpha
 command line option, [373](#)
 MRViewRotateModifierKey
 command line option, [373](#)
 MRViewShowColourbar
 command line option, [373](#)
 MRViewShowComments
 command line option, [373](#)
 MRViewShowFocus
 command line option, [373](#)
 MRViewShowOrientationLabel
 command line option, [373](#)
 MRViewShowVoxelInformation
 command line option, [373](#)
 MRViewSyncFocus
 command line option, [374](#)
 MRViewToolFontSize
 command line option, [374](#)
 MRViewToolsColourBarPosition
 command line option, [374](#)
 MRViewWrapVolumes
 command line option, [374](#)
 MSAA
 command line option, [374](#)

N

NeedOpenGLCoreProfile
 command line option, [375](#)
 NIFTIAllowBitwise

- command line option, 374
- NIfTIAlwaysUseVer2
 - command line option, 374
- NIfTIAutoLoadJSON
 - command line option, 374
- NIfTIAutoSaveJSON
 - command line option, 374
- NIfTIUseSform
 - command line option, 374
- NumberOfThreads
 - command line option, 375
- NumberOfUndos
 - command line option, 375

O

- ObjectColor
 - command line option, 375

R

- RealignTransform
 - command line option, 375
- RegAnalyseDescent
 - command line option, 375
- RegCoherenceLen
 - command line option, 375
- RegGdConvergenceDataSmooth
 - command line option, 375
- RegGdConvergenceMinIter
 - command line option, 375
- RegGdConvergenceSlopeSmooth
 - command line option, 375
- RegGdConvergenceThresh
 - command line option, 375
- RegGdWeightMatrix
 - command line option, 375
- RegGdWeightTranslation
 - command line option, 376
- RegStopLen
 - command line option, 376

S

- ScriptScratchDir
 - command line option, 376
- ScriptScratchPrefix
 - command line option, 376
- SparseDataInitialSize
 - command line option, 376
- SpecularExponent
 - command line option, 376
- SpecularIntensity
 - command line option, 376

T

- TckgenEarlyExit

- command line option, 376
- TerminalColor
 - command line option, 376
- TmpFileDir
 - command line option, 376
- TmpFilePrefix
 - command line option, 377
- ToolbarStyle
 - command line option, 377
- TrackWriterBufferSize
 - command line option, 377

V

- VSync
 - command line option, 377