

How to create a head model

This document describes the command line tools:

- **mri2mesh**: Central tool to reconstruct a head model from T1w and T2w data
- **dwi2cond**: Reconstruct conductivity tensors for brain GM and WM from diffusion MRI data; needed when anisotropic conductivities should be used in the FEM calculations

mri2mesh

This tool reconstructs a tetrahedral head mesh from T1w and T2w structural MR images. It runs with only a T1w image, but will create better skull segmentations when also a T2w image is available.

Usage:

1. Open a terminal and go to the directory of the SimNIBS example data set
2. Run the reconstruction:

```
mri2mesh --all almi5 org/almi5_T1fs.nii.gz org/almi5_T2.nii.gz
```

The argument **--all** tells mri2mesh to run all reconstruction steps including volume meshing. The **subject ID** “almi5” is given next. Mri2mesh will create a mesh named “almi5.msh”, a folder “fs_almi5” that contains the FreeSurfer results, and a folder “m2m_almi5” that contains the files that are needed for volume meshing. The **input images** are given as final arguments (first the T1w, then the T2w). When calling mri2mesh --all the first time for a dataset, it will run FreeSurfer on it using the fat-suppressed T1 as input. This is quite time-consuming. When re-running mri2mesh --all it will use the existing FreeSurfer results. Then it will need ~3-4 hours.

Alternatively, the reconstruction can be run with only the T1w image as input, but this will

result in a less accurate skull region:

```
mri2mesh --all almi5 org/almi5_T1fs.nii.gz
```

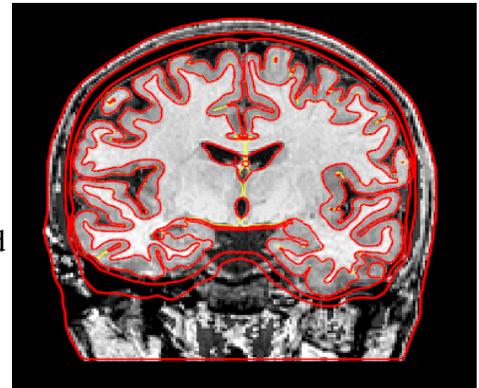
It is also possible to run it with 2 T1w images (one fat-suppressed, one normal) and 2 T2w images (again one fat-suppressed, one normal). This can bring a slight increase in the accuracy of the reconstructed skull:

```
mri2mesh --all almi5 org/almi5_T1.nii.gz org/almi5_T1fs.nii.gz  
org/almi5_T2.nii.gz org/almi5_T2fs.nii.gz
```

3. Check the results:

```
mri2mesh -c almi5
```

This will show the reconstructed surfaces overlaid over the MR images using freeview. The red lines indicate the final surfaces used for volume meshing, the yellow indicate the GM and WM surfaces created by FreeSurfer. In addition, you should have a look at the tetrahedral head mesh by loading it into gmsh.



4. Further notes

- A quick check can be performed by looking at the final volume masks overlaid over the structural images in fslview:

```
mri2mesh --qc almi5
```

- As a standard, mri2mesh uses 60000 triangles for each white matter surface, and the number of triangles for the other surfaces are scaled relative to this number. This results in a volume mesh of ~3 million tetrahedra. Alternatively, you can adjust the mesh resolution by setting `--numvertices=<mynumber>`
- After the head mesh creation, temporary files can be deleted to save disk space using:

```
mri2mesh --cleanup almi5
```

- When something goes wrong, you can check the `mri2mesh_log.html` which is saved in the `m2m_almi5` subdirectory.

dwi2cond

This optional tool reconstructs the conductivity tensors for brain GM and WM from diffusion MRI data. It is required when anisotropic conductivities will be used in the FEM calculations. Only single shell data (i.e., with a single b-value in addition to some b=0 images) with a single phase encoding direction for the EPI readout is supported. To run it in its simplest form:

```
dwi2cond --all almi5 org/almi5_DWI.nii.gz org/almi5_DWI_bvals  
org/almi5_DWI_bvecs
```

After the **subject ID**, the **diffusion dataset** is given, followed by text files for the **b-values** and the **b-vectors**. Please refer to the description of fsl dtifit for details on the file conventions. Dwi2cond performs eddy current correction on the data, registers it to the structural T1 image created by mri2mesh (using a non-linear registration based on fsl fnirt) and applies the diffusion-to-conductivity mapping. As a standard, it is ensured that the tensors are positive definite by setting eigenvalues $< 1e-6$ to $1e-6$. This is done to ensure that the FEM solver will robustly converge. In addition, a maximal mean conductivity of 1 S/m is enforced to guarantee a realistic conductivity range. Both options can be deactivated on the command line.

The results are stored in the directory “d2c_almi5”. The final **result files** are:

CTI_dir_tensor.nii.gz: Conductivity tensor image based on the direct mapping approach (Tuch et al., 2001; Rullmann et al., 2009).

CTI_dir_mc.nii.gz: Mean conductivities based on the direct mapping approach. In contrast to the full tensor image, the conductivities stored here are isotropic. They are inhomogeneous, i.e. they vary with position in accordance with the variations of the mean diffusivity (MD) of the diffusion MR data.

CTI_vn_tensor.nii.gz : Conductivity tensor image based on a volumetric normalized mapping approach (Güllmar et al., 2010).

To check the results:

```
dwi2cond -c almi5
```

Notes:

- `dwi2cond` also supports **distortion correction based on fieldmaps**:

```
dwi2cond --all --dwdwell=<dwel_time_in_ms>  
--udir=<warping_direction> <subjectID> <dMRI_dataset>  
<bvalue_file> <bvector_file> <field_map_magnitude_image>  
<field_map_phase_image>
```

The standard warping direction is set to “y-”. A standard Siemens gre fieldmap with a TE difference of 2.46 ms is assumed. For other fieldmaps, additional parameters have to be defined on the command line (see `dwi2cond -help`).

- **Eddy current correction based on fsl eddy** is supported and can be combined with **distortion correction based on fsl topup** (if a reversed phase b0 image is available):

```
dwi2cond --all --eddy=1 --readout=<readout time in ms>  
--phasedir=<x/-x/y/-y/z/-z> <subjectID> <dMRI_dataset>  
<bvalue_file> <bvector_file> <reversed_phase_b0_image>
```

The readout time of the dMRI sequence has to given in [s]. Phasedir refers to the phase encoding direction of the sequence. Both arguments are mandatory.

- Alternatively, you can do your own preprocessing of the dMRI data and feed the results into `dwi2cond`. The same kind of files as created by `fsl dtifit` have to be provided:

```
dwi2cond --all <subjectID> <DTI_basename>
```