

SimNIBS 2.1

command line tools

Guilherme B Saturnino, Axel Thielscher

Contact: support@simnibs.org

Table of Contents

1	Creating a head mesh.....	1
1.1	mri2mesh.....	2
1.2	headreco.....	4
1.3	dwi2cond.....	6
2	Running a FEM simulation.....	9
3	Auxiliary command line tools.....	11
3.1	eeg_positions.....	11
3.2	msh2cortex.....	12
3.3	msh2nii.....	12
3.4	nii2msh.....	13
3.5	mni2subject.....	13
3.6	mni2subject_coords.....	14
3.7	subject2mni.....	15
3.8	subject2mni_coords.....	15
4	References.....	17

1 Creating a head mesh

The automatic creation of individualized tetrahedral head meshes from structural MR data is supported by two command line programs to choose from, namely **mri2mesh** (Windhoff et al., 2013) or alternatively the new **headreco** (Nielsen et al., 2018). In addition, SimNIBS can model the conductivity of brain gray and white matter as being anisotropic. For this, the conductivity tensors are estimated from diffusion MR tensors. This requires that the diffusion MR images are pre-processed using **dwi2cond** (Opitz et al., 2011). Please cite the corresponding publications, as stated above, when using these programs.

1.1 mri2mesh

Description: mri2mesh reconstructs a tetrahedral head mesh from T1- and T2-weighted structural MR images. It runs also with only a T1w image, but will create better skull segmentations when also a T2w image is available.

Dependencies: mri2mesh needs FreeSurfer (5.3.0 or newer) and FSL (5.0.5 or newer) to be installed.

Usage example:

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

```
mri2mesh --all ernie org/ernie_T1.nii.gz org/ernie_T2.nii.gz
```

The argument **--all** tells mri2mesh to run all reconstruction steps including volume meshing. The **subject ID** “ernie” is given next. Mri2mesh will create a mesh named “ernie.msh”, a folder “fs_ernie” that contains the FreeSurfer results, and a folder “m2m_ernie” that contains the files that are needed for volume meshing. The **input images** are given as final arguments (first the T1, then the T2). When calling mri2mesh --all the first time for a dataset, it will run FreeSurfer on it using the 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.

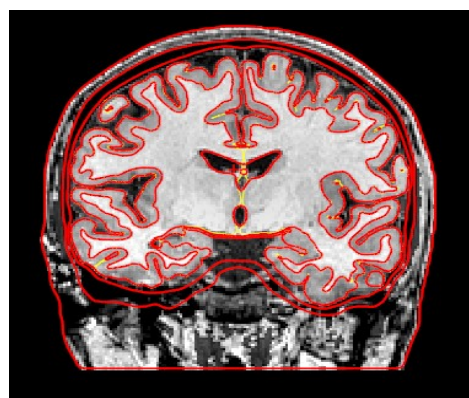
Alternatively, the reconstruction can be run with only the T1w image as input, but this will result in a less accurate skull region:

```
mri2mesh --all ernie org/ernie_T1fs.nii.gz
```

3. Check the results:

```
mri2mesh -c ernie
```

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. A second freeview will show the subject T1



overlaid on the MNI template for a visual check of the registration accuracy. In addition, you should have a look at the tetrahedral head mesh by loading it into gmsh.

Further notes

- A quick check can be performed by looking at the final volume masks overlaid over the structural images in fslview: **mri2mesh --qc ernie**
- 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 ~3.5 million tetrahedra. Alternatively, you can adjust the mesh resolution by setting **--numvertices=<mynumber>**
- After the head mesh creation, temporary files are deleted to save disk-space. Adding **--nocleanup** prevents this.
- When setting **--t2pial**, FreeSurfer will use the T2 image to improve the estimate of the pial surfaces (recommended only for high-res with T2 images 1mm iso voxel).
- **Manual editing:** For improving the GM and WM surfaces after the first run of mri2mesh, edit the FreeSurfer results as described on the FreeSurfer wiki. Then run mri2mesh again with the --all option, as stated above. Mri2mesh will use the edited FreeSurfer results to create a new head mesh. For improving the ventricles, cerebellum, csf, skull or skin surfaces, manually edit one or more of the binary masks stored in the mask_prep subdirectory of the m2m_{subID} folder. Then run mri2mesh again with the --all and --keep_masks options. The latter option will prevent mri2mesh from overwriting the edited masks.
- **Transformation from and to MNI space:** Both positions and results such as the electric field can be transformed between MNI and subject space. Please see below for a description of the corresponding command line programs. The transformation is based on a non-linear whole-head registration of the T1 of the subject to the MNI template, using FSL's fnirt command. Even though fnirt was developed for registering the brain, usually acceptable results are achieved by mri2mesh for the whole head. When a T1 without fat suppression is used as input, the bright skull might

be warped into the skin or brain. Using the option `-mnimaskskull` can prevent this. A skull mask will then be applied to down-weight the skull intensity. The transformations are stored in the `toMNI` subdirectory of the `m2m_{subID}` folder. Subject space is defined by the `qform` set in the `{subID}_T1fs_conform.nii.gz`, which can be found in the same folder as the head mesh.

- When something goes wrong, you can check the **`mri2mesh_log.html`** which is saved in the `m2m_ernie` subdirectory.

1.2 headreco

Description: headreco reconstructs a tetrahedral head mesh from T1- and T2-weighted structural MR images. It runs also with only a T1w image, but it will achieve more reliable skull segmentations when a T2w image is supplied.

Dependencies: headreco needs SPM12 and matlab to be installed. The SPM12 folder has to be in the search path of matlab (test by typing “spm” on the matlab command line). We also highly recommend the installation of the CAT12 toolbox to achieve more accurate reconstructions of the GM and WM surfaces. CAT12 is quite dynamically developed. Please make sure to use the CAT12 release version stated in the SimNIBS installation description in order to prevent undesired behavior. Headreco uses extended tissue probability masks that cover the neck region, as described in (Huang et al., 2013). These masks are supplied with the SimNIBS installation and do not have to be separately installed.

Usage example:

1. Open a terminal and go to the directory of the “Ernie” example data set.
2. Run the reconstruction:
`headreco all --cat ernie org/ernie_T1.nii.gz org/ernie_T2.nii.gz`

The argument **`all`** tells headreco to run all reconstruction steps including volume meshing. The **subject ID** “ernie” is given next. Headreco will create a mesh named “ernie.msh”, and a folder “m2m_ernie” that contains the segmentation results and the files needed for volume meshing. The **input images** are given as final arguments (first the T1, then the T2). The argument **`--cat`** tells headreco to use CAT12 for the segmentation of the WM

and GM surfaces. This is optional, but highly recommended.

Alternatively, the reconstruction can be run with only the T1w image as input, but this will result in a less accurate skull region:

headreco all --cat ernie org/ernie_T1.nii.gz

3. Check the results:

headreco check ernie

This will show the reconstructed surfaces overlaid over the MR images using freeview. A second freeview will show the subject T1 registered to the MNI template for visual inspection of the accuracy of the registration. In addition, you should have a look at the tetrahedral head mesh by loading it into gmsh. In case freeview is not available, the spm viewer will be opened to allow for a basic check of the results.

Further notes

- When setting **-d no-conform**, headreco will not resample the supplied T1 and keep its original qform.
- Mesh resolution can be controlled using the **-v** option, which allows setting the vertex density (nodes per mm²) of the surface meshes. As a standard, headreco uses 0.5 nodes per mm², resulting in head meshes with around 4 million tetrahedra.
- After the head mesh creation, temporary files are deleted to save disk space. Adding **--noclean** prevents this.
- **Manual editing:** Edit one or more of the binary masks stored in the mask_prep subdirectory of the m2m_{subID} folder. Then run **headreco surfacemesh subID** and **headreco volumemesh subID** to re-create the head mesh based on the edited masks. Add --cat to the surfacemesh step in case you used CAT12. Note: When using CAT12, surfaces instead of voxel masks will be stored for GM and WM in the mask_prep folder. For now, these surfaces cannot be manually improved.
- **Transformation from and to MNI space:** Both positions and results such as the electric field can be transformed between MNI and subject space. Please see below for a description of the corresponding command line programs. The transformation is based on a non-linear whole-head registration of the T1 of the subject to the MNI template that is determined

during the SPM12 segmentation procedure. The transformations are stored in the toMNI subdirectory of the m2m_{subID} folder. Subject space is defined by the qform set in the {subID}_T1fs_conform.nii.gz, which can be found in the same folder as the head mesh.

- When something goes wrong, you can check the **headreco_log.html** which is saved in the m2m_ernie subdirectory.

1.3 dwi2cond

Description: This optional tool prepares diffusion tensors for GM and WM from diffusion MRI (dMRI) data. The prepared tensors can then be used by SimNIBS to estimate anisotropic conductivities for GM and WM during 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. All processing steps are based on FSL tools.

Dependencies: dwi2cond needs FSL 5.0.9 or newer.

Usage example:

1. Open a terminal and go to the directory of the “Ernie” example data set.
2. Run:

```
dwi2cond --all --eddy --phasedir=-y --readout=0.05 ernie  
org/ernie_dMRI.nii.gz org/ernie_dMRI.bval org/ernie_dMRI.bvec  
org/ernie_dMRI_rev.nii.gz
```

- The argument **--all** tells dwi2cond to preprocess the diffusion MRI data, reconstruct the diffusion tensors, and register the tensors to the T1 of the subject. Preprocessing encompasses correction for subject motion and eddy-current-induced image distortions and, optionally, correction for static image distortions.
- Setting **--eddy** will prompt dwi2cond to use FSL eddy for motion and eddy-current correction. In this case, also the phase encoding direction of the dMRI sequence and its readout time (in [s]) has to be stated using **--phasedir** and **--readout**. Please refer to the description of FSL eddy on the FSL homepage for further information about these parameters.

- The **subject ID** is provided as first parameter after the arguments, followed by the name of the **diffusion dataset** and the text files for the **b-values** and the **b-vectors**. Please refer to the description of FSL dtifit for details on the file conventions.
- A dataset of b=0 images acquired with reversed phase encoding direction is given as last parameter. In this case, FSL topup will be used for static distortion correction.

3. Check the results:

dwi2cond -check ernie

This shows the fractional anisotropy (FA) image superimposed on the structural T1 of the subject in fslview or fsleyes for visual control of the coregistration accuracy. In a second viewer, FA images determined from the dMRI data before and after preprocessing are shown to allow for controlling the effects of the preprocessing. These FA images are registered to the T1 using a rigid body (6DoF) transformation, so that spatial distortions remain visible. Finally, also sum-of-squared-error (SSE) images of the tensor fitting procedure are shown. The SSE images are in arbitrary units, but are helpful to check the effectiveness of the preprocessing steps and to identify putative artifacts in the dMRI data. Finally, the first eigenvectors of the tensors are shown in gmsh for checking their orientation.

Further notes

- Dwi2cond uses a non-linear registration based on FSL fnirt to transform the diffusion tensors to the space of the structural T1 image. In case the distortion correction during preprocessing is good enough, 12 or 6 DoF affine registrations can be used instead (controlled by the argument **-regmthd**).
- dwi2cond also supports distortion correction based on **fieldmaps**:

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

The argument **--dwidwell** is used to specify the dwell-time (or echo-spacing) of the dMRI sequence in [ms]. The standard warping direction is set to “y-”, otherwise state it using **--udir**. 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`).

- Alternatively, you can use your own preprocessed DTI data as input

dwi2cond --all <subjectID> <DTI-tensor>

Dwi2cond will coregister the tensor image with the structural T1 of the subject for further use with SimNIBS.

- When something goes wrong, you can check the **dwi2cond_log.html** which is saved in the `d2c_ernie` subdirectory.
- SimNIBS will use the coregistered diffusion tensors (stored in `d2c_{subID}/dti_results_T1space/DTI_conf_tensor.nii.gz`) to estimate conductivity tensors for GM and WM. The following mappings from diffusion to conductivity tensors are supported:
 - Direct mapping based on linear rescaling of the diffusion tensors (Rullmann et al., 2009; Tuch et al., 2001).
 - Inhomogeneous, isotropic conductivities, calculated as the mean conductivities of tensors of the direct mapping approach.
 - Volume normalized mapping (Güllmar et al., 2010) that keeps the geometric mean of the eigenvalues identical to the standard isotropic conductivity.
- SimNIBS ensures that all conductivity tensors are positive definite. This is done to ensure that the FEM solver will robustly converge. In addition, a maximal conductivity of 2 S/m and a maximal ratio of 10 between the largest and smallest conductivity eigenvalues is enforced to guarantee a realistic conductivity range. The latter values can be changed in the `simnibs_gui`.

2 Running a FEM simulation

As alternative to start simulations from the `simnibs_gui`, they can be directly started from the command line using the `.mat`-file with the simulation settings. For example, open a terminal and go to the directory of the “Ernie” example data set. You can run a few pre-defined tDCS simulations by typing:

```
simnibs ernie_simu.mat
```

You can run multiple TMS simulations at once by setting the `--cpus` argument

```
simnibs tms_simu.mat --cpus 3
```

This will run 3 TMS simulations at once. Note that each takes approximately 4 GB memory.

Most settings are controlled by the information stored in the `.mat`-file. The easiest way to get familiar with this structure is to load an existing SimNIBS `.mat`-file into matlab and explore it there. We also provide some example matlab scripts to create SimNIBS `.mat`-files. Shortly, the structure is composed as follows:

- `SESSION`: Top structure

Input head mesh, output folder, anisotropy tensor name, list of TMS and tDCS simulations, in the format of a list of `POSLIST` structures

- `POSLISTS`:

Specify coils or electrodes, their positions and orientations, and conductivities

Can be of 2 types: `TMSLIST` and `TDCSLIST`

`TMSLIST`:

- Path to TMS coil file (either “.ccd” file or nifti file)
- List of conductivities, uses `COND` class
- Specification of tissue anisotropy
- List of TMS coil positions, orientations, and dl/dt , uses
 - `POSITION` class:

Coil position and orientation, given by a 4x4 matrix

Rate of change of current, di/dt

TDCSLIST:

- List of currents flowing through each channel. Negative values indicate anodes, positive values cathodes
- List of conductivities, uses COND class
- Specification of tissue anisotropy
- List of electrodes, uses
 - ELECTRODE class:
Electrode shape, dimensions, center, “Y” direction reference
Channel to which the electrode is connected

3 Auxiliary command line tools

3.1 eeg_positions

Description: Calculates the 10/10 EEG positions based on 4 fiducial points based on the UI 10/10 definition provided in (Jurcak et al., 2007).

Usage example:

1. Open a terminal and go to the directory of the “Ernie” example data set.
2. Open the mesh file in the GUI

simnibs_gui ernie.msh

3. Visually determine the positions of the nasion (Nz), left preauricular point (LPA), right preauricular point (RPA) and inion (Iz) on the skin surface and click on them to get their positions. Note down the coordinates displayed in the bottom of the window.
4. Run

eeg_positions -m ernie.msh -o ernie_10_10 -Nz 1.6 97.1 3.4 -LPA -82.2 -14.3 -22.2 -Iz -4.8 -108.8 -32.7 -RPA 79.7 -18.9 -24.6

(Usually, one would use the coordinates noted down in step 3 here)

5. Two files will be created: **ernie_10_10.csv** and **ernie_10_10.geo**
6. Check the positions by opening the mesh file in Gmsh

simnibs_gui ernie.msh

or double clicking in the case of MacOSX (see the installation instructions).

Go to **File**→ **Merge** and open the **ernie_10_10.geo** file for a visual check of the electrode positions

7. **ernie_10_10.csv** is a text file containing names and positions of the EEG electrodes. For now, you have to manually enter these coordinates when defining electrodes in `simnibs_gui`. Alternatively, you can write your own matlab- or python-script that uses the positions to create a .mat-file with the

simulation information.

3.2 msh2cortex

Description: Interpolates fields to a surface in the middle of the cortex. This only works for head models created with mri2mesh or headreco with the -cat option.

Not available for head models created using headreco with **SPM12 only**.

Usage example:

1. Open a simulation results folder, for example **ernie/simu** if you ran **ernie_simu.mat**
2. Run

```
msh2cortex -i ernie_TDCS_1_scalar.msh -m ../m2m_ernie/ -o  
subject_overlays -f fsavg_overlays
```

3. Freesurfer overlays for the subject files will be created in the **subject_overlays** folder and for the average template in the **fsavg_overlays** folder. This is equivalent to selecting the “interpolate to cortical surface” and the “transform to fsaverage space” options in simnibs_gui when setting up the simulation.

Further notes:

- Type **msh2cortex -h** for more information and options
- The **--load_freeview** option only works with FreeSurfer installed

3.3 msh2nii

Description: Transforms fields in mesh files to nifti files

Usage example:

1. Open a simulation results folder, for example **ernie/simu** if you ran **ernie_simu.mat**
2. Run

```
msh2nii ernie_TDCS_1_scalar.msh  
../m2m_ernie/T1fs_conform.nii.gz ernie_TDCS_1
```

3. This will create the **ernie_TDCS_1_E.nii.gz**, **ernie_TDCS_1_J.nii.gz**, **ernie_TDCS_1_normE.nii.gz**, **ernie_TDCS_1_normJ.nii.gz** files.

The files are in the same space as the **../m2m_ernie/T1fs_conform.nii.gz** file

Further notes:

- Type **msh2nii -h** for more information and options
- Can also be used to create tissue masks

3.4 nii2msh

Description: Interpolates NifTI volumes to mesh files. Mostly used internally by **dwi2cond**

Usage example:

1. Open a terminal and go to the directory of the “Ernie” example data set.
2. Run

```
nii2msh -ev d2c_ernie/dti_results_T1space/DTI_conf_tensor.nii.gz  
ernie.msh ernie_tensor.msh
```

3. This will create the **ernie_tensor.msh** file. This file will contain the eigenvectors associated with the largest eigenvalues of the conductivity tensors (for visual control).

Further notes:

- Type **nii2msh -h** for more information and options

3.5 mni2subject

Description: Transforms volumes from MNI space to subject space.

Usage example:

1. Open a terminal and go to the directory of the “Ernie” example data set.
2. Suppose you have a volume mask defined in MNI space, called **Mask_MNI.nii.gz**
3. Run

```
mni2subject -i Mask_MNI.nii.gz -m m2m_ernie/ -o  
Mask_ernie.nii.gz
```

4. This will create the **Mask_ernie.nii.gz** file, with the mask transformed to the subject space

Further notes:

- Type **mni2subject -h** for more information and options

3.6 mni2subject_coords

Description: Transforms coordinates from MNI space to subject space.

Usage example:

1. Open a terminal and go to the directory of the “Ernie” example data set.
2. Run

```
mni2subject_coords -m m2m_ernie/ -s  
$SIMNIBSDIR/resources/ElectrodeCaps_MNI/EEG10-  
10_UI_Jurak_2007.csv -o ernie_10_10.csv
```

This will transform a list of electrode positions that were defined in MNI space according to the EEG 10/10 system (using the “UI” convention as defined in Jurak et al 2007) to subject space. The transformed positions are stored in the file **ernie_10_10.csv**.

Further notes:

- Type **mni2subject_coords -h** for more information and options

- When transforming electrode positions, the results are always projected to the skin after the transformation.

3.7 subject2mni

Description: Transforms a mesh or volume to MNI space

Usage example:

1. Open a simulation results folder, for example **ernie/simu** if you ran **ernie_simu.mat**
2. Run

```
subject2mni -i ernie_TDCS_1_scalar.msh -m ../m2m_ernie/ -o ernie_TDCS_1
```

This will create the **ernie_TDCS_1_MNI_E.nii.gz**, **ernie_TDCS_1_MNI_J.nii.gz**, **ernie_TDCS_1_MNI_normE.nii.gz**, and **ernie_TDCS_1_MNI_normJ.nii.gz** files, with the electric fields transformed to MNI space.

Further notes:

- Type **subject2mni -h** for more information and options

3.8 subject2mni_coords

Description: Transforms coordinates from subject space to MNI space

Usage example:

1. Suppose you open a simulation and see a hotspot at the position (-43, -17, 74)
2. Run

```
subject2mni_coords -m m2m_ernie/ -c -43 -17 74
```

This will output the corresponding coordinates in MNI space.

Further notes:

- Type **subject2mni_coords -h** for more information

4 References

- Güllmar, D., Haueisen, J., Reichenbach, J.R., 2010. Influence of anisotropic electrical conductivity in white matter tissue on the EEG/MEG forward and inverse solution. A high-resolution whole head simulation study. *Neuroimage*. <https://doi.org/10.1016/j.neuroimage.2010.02.014>
- Huang, Y., Dmochowski, J.P., Su, Y., Datta, A., Rorden, C., Parra, L.C., 2013. Automated MRI segmentation for individualized modeling of current flow in the human head. *J Neural Eng.* 10.
- Jurcak, V., Tsuzuki, D., Dan, I., 2007. 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *Neuroimage* 34, 1600–1611. <https://doi.org/10.1016/j.neuroimage.2006.09.024>
- Nielsen, J.D., Madsen, K.H., Puonti, O., Siebner, H.R., Bauer, C., Madsen, C.G., Saturnino, G.B., Thielscher, A., 2018. Automatic skull segmentation from MR images for realistic volume conductor models of the head: Assessment of the state-of-the-art. *Neuroimage c*, 1–12. <https://doi.org/10.1016/j.neuroimage.2018.03.001>
- Opitz, A., Windhoff, M., Heidemann, R.M., Turner, R., Thielscher, A., 2011. How the brain tissue shapes the electric field induced by transcranial magnetic stimulation. *Neuroimage* 58, 849–859. <https://doi.org/10.1016/j.neuroimage.2011.06.069>
- Rullmann, M., Anwander, A., Dannhauer, M., Warfield, S.K., Duffy, F.H., Wolters, C.H., 2009. EEG source analysis of epileptiform activity using a 1 mm anisotropic hexahedra finite element head model. *Neuroimage* 44, 399–410. <https://doi.org/10.1016/j.neuroimage.2008.09.009>
- Tuch, D.S., Wedeen, V.J., Dale, a M., George, J.S., Belliveau, J.W., 2001. Conductivity tensor mapping of the human brain using diffusion tensor MRI. *Proc. Natl. Acad. Sci. U. S. A.* 98, 11697–701. <https://doi.org/10.1073/pnas.171473898>
- Windhoff, M., Opitz, A., Thielscher, A., 2013. Electric field calculations in brain stimulation based on finite elements: An optimized processing pipeline for the generation and usage of accurate individual head models. *Hum. Brain Mapp.* 34, 923–935. <https://doi.org/10.1002/hbm.21479>