How to Make a Custom Head Model from MRI

Author: Noelle Jacobsen, University of Florida, 6/24/21. Last updated 9/15/21

This document contains instructions on how to create a custom head model from a subject-specific MRI using Fieldtrip (FT) in Matlab. This guide follows most steps from the fieldtrip tutorials linked below, so II will only comment on the modifications made to the tutorial instead of each step.

[](https://github.com/jacobsen-noelle)

GitHub

Code for making a custom FEM head model is available at Noelle Jacobsen’s GitHub:

[**jacobsen-noelle**](https://github.com/jacobsen-noelle/EEG_Processing/commits?author=jacobsen-noelle)

[**EEG\_Processing**](https://github.com/jacobsen-noelle/EEG_Processing)**/Custom Head Model**/ **makeheadmodel.m**

<https://www.fieldtriptoolbox.org/tutorial/headmodel_eeg_fem/>

<https://www.fieldtriptoolbox.org/workshop/baci2017/forwardproblem/#9a-compute-the-leadfield>

\*Note that the conductivities for the gray and white matter in step **6B Create head model** in the second linked tutorial are WRONG. The correct values should be gray = 0.33, white = 0.14. See [Conductivities](#_Conductivities) for table of values.

# Setup Fieldtrip

First [download the FieldTrip toolbox](https://www.fieldtriptoolbox.org/download/) and add to any folder EXCEPT your …\Program Files\ MATLAB folder. You should never manually add folders to your Matlab folder or it will cause issues. The Fieldtrip version found in the EEGLab plugin will not suffice. To avoid confusing Matlab, you should:

%remove the path to your EEGlab FieldTrip plugin folder

rmpath'YOURPATH\eeglab2021.0\plugins\Fieldtrip-lite20210601

% add the path to the full fieldtrip folder

addpath 'YOURPATH’

%Let FT add default paths

ft\_defaults;

# Preprocess MRI

The MRI does not need to be normalized to an MNI template to create a custom head model. However, I’ve found that my raw images don’t work (images get flipped and oriented, errors with various functions), so I recommend normalizing the MRIs anyways. To do this, you can normalize your MRI using one of the following two methods:

1 ) SPM (see instructions here *Ferris-Lab\share\HNL\_Documentation\Lab Manual\MRI\Subject-specific MRI Files Processing*)

2) fieldtrip function ft\_volumenormalise() – recommended (quicker, less additional steps)

## SPM normalization

Reorient images in SPM is to roughly align structural images with standard space prior to tissue class segmentation. If you’re using MNI template, roughly reorient your image to be centered at anterior commissure (center [0 0 0] on ac). See [how to reorient using SPM display tool](http://jpeelle.net/mri/misc/display.html).

## Fieldtrip normalization

%Read MRI

mri = ft\_read\_mri(mrifile);

cfg = [];

mri\_norm = ft\_volumenormalise(cfg,mri);

mri\_norm = ft\_determine\_coordsys(mri\_norm); %verify coordinate system is labeled correctly

cfg = [];

ft\_sourceplot(cfg, mri\_norm); %when you click on anterior commissure, coordinates should be [0 0 0] (or close)

## Issue: MRI images are cropped after normalizing

Change size of bounding box so that it is homogeneous (i.e. same total size in all directions XYZ)

Because the anterior->posterior length is the longest, which is in the Y direction, that will be the new minimum size of the bounding box.

For this image, the largest dimension

-104 -112 -96

104 76 112

I added some padding to the Y dim (middle column), then made the x and z size match same total num of voxels

-104 -122 -96

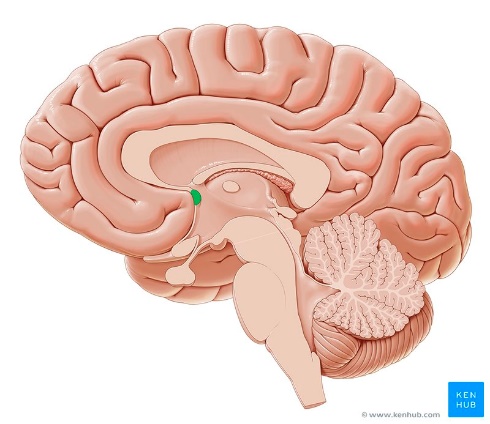
104 86 112

e.g the total number of voxels in the Y dim is 86-(-122)= 208. I then added padding to X and Z in both directions so that the total number of voxels equals 208. X vox = 104-(-104)= 208, Z vox = 122-(-96)

\*(could be a tiny bit bigger)

# Making Head Model

## Load MRI

* Use fieldtrip function ft\_read\_mri to load MRI .nii file.

## Normalize MRI

* If the MRI image is raw (not processed), and you selected inputMRItype = ’raw’ , then the MRI will be normalized for you using ft\_volumenormalise as mentioned previously.
* **Verify that your MRI has been normalized and with the origin at the anterior commissure (ac).** After loading the image and plotting using ft\_sourceplot(), click on the location of the anterior commissure (ac), which looks like a little circle. The ac is a white matter tract (a bundle of axons) connecting the two temporal lobes of the cerebral hemispheres across the midline. The coordinates of the ac on the MRI should be [0 0 0] (or very close) (Fig. 2. Verify that your MRI has been normalized properly and that the ac is at the origin of the coordinate system.

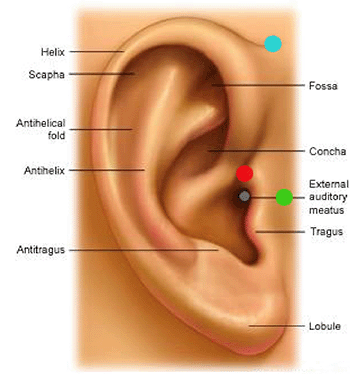
Figure 1 Location of anterior commissure (green)

200 
voxel 3554569, indices [91 109 
spm coordinates [O O O) 
atlas label: NA 
Isa 

Figure 2 Crosshairs located at anterior commissure [0 0 0]

## Locate fiducials in MRI

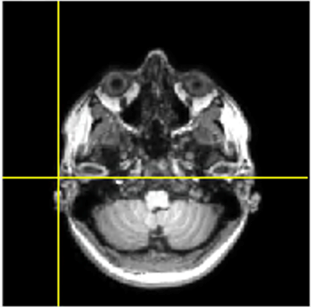
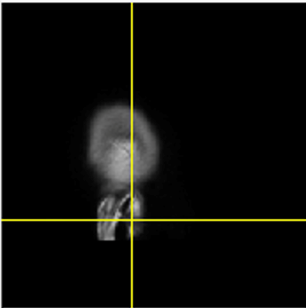
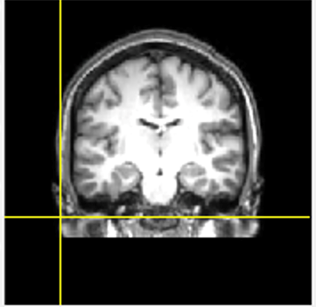
Use ft\_volumerealign to mark fiducials (nas, lhj, rhj), but don't actually use the output MRI created with this function, or else your MRI will be realigned so that that fiducials create a new coordinate system (CTF coordsys) instead of keeping the MNI coordinate system. We are only using this function because it is an interactive way to mark fiducials. **IMPORTANT –** mark the same fiducial locations that are marked when determining EEG electrode locations in EEGlab. You must be consistent in locations or your head model won’t be properly aligned with your electrode locations. Typically, this means you need to find lhj/rhj instead of lpa/rpa (Fig. 4). Store these fiducial coordinates you marked so that you can use them to quickly align your electrodes to your source model later.



* LPA/RPA - Left and Right Pre Auricular
* LEC/REC - Left and Right Ear canal
* LHS/RHS - Left and Right Helix-Scalp junction
* **LHJ/RHJ - Left and Right Helix-Tragus Junction**

Figure 3 Fiducal points nasion (nas) and left helix-tragus junction (LHJ) identified on MRI





**LHJ**

Figure 4 Fiducial points

### Transform fiducial coordinates to new coordinate system

Your MRI anatomy isn’t changed through normalization but instead now has a transform matrix associated with the MRI. This transform matrix can be used to transform coordinates obtained from the MRI space and move them to a new coordinate system—the MNI coordinate system. To do this, use the ft\_warp\_apply function() along with the transformation matrix associated with your normalized MRI.

cfg = [];

cfg.method = 'interactive';

cfg.coordsys = 'ctf'; %spm doesn't work bc won't let me mark nas, need to mark in this coodsys

mri\_realigned = ft\_volumerealign(cfg, mri\_norm); %you won't actually use this mri, just getting fiducial locations

%grab fiducial locations you just marked using ft\_volume realign

nas = mri\_realigned.cfg.fiducial.nas;

lhj = mri\_realigned.cfg.fiducial.lpa;

rhj = mri\_realigned.cfg.fiducial.rpa;

% use original transformation from original MRI to MNI to bring fid to

%normalize coord space

vox2head = mri\_norm.transform; %same transform as mr\_realign.transformprig

nas = ft\_warp\_apply(vox2head, nas, 'homogenous');

lhj = ft\_warp\_apply(vox2head, lhj, 'homogenous');

rhj = ft\_warp\_apply(vox2head, rhj, 'homogenous');

### Verify fiducials

You can plot the transformed fiducial coordinates on the MRI using ft\_sourplot() to verify they have been marked correctly. The crosshairs on the plot should be at the location you identified.

2: ft_sourcepl 
at: mrl norm 
File Edit View 
Insert Tools Desktop Window 
100 ISO 
Help 
FieldTrip 
100 ISO 
200 
voxel 1137677, •ndices [92210 291 
spm coordinates [1 83-441 
atlas label: NA 

Figure 5 Nasion location verified at crosshair position

## Segment MRI

Use FEM to segment the MRI in to 5 layers. I recommend saving this segmented MRI in a folder because you can later use it as an atlas to remove dipoles outside of the brain.

## Create the mesh

Head model 
EEG electrodes 
Fiducials in MRI NOTE: you might have to copy simbio folder and add it to ...\fieldtrip-20190920\external if simbio is not being recognized and you get this error "Error using ft\_hastoolbox (line 496) the SIMBIO toolbox is not installed"

Figure 6 EEG electrodes aligned to source model using alignment of fiducials

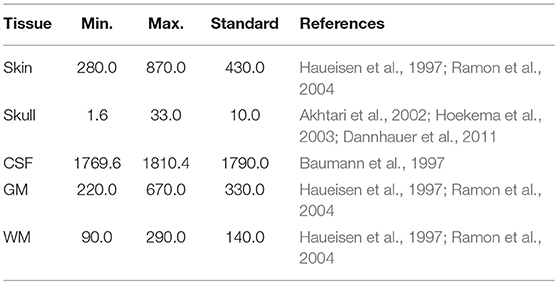
## Align electrodes

Co-register source model and EEG electrodes based on fiducials (nasion and left and right pre-auricular points). Must already have EEG electrode locations saved in .txt file, which should include fiducial locations (nas, lhj,rhj). There are other optional methods to align electrode locations to the mesh (alignment with head shape method, interactive electrode alignment), but using fiducial location method is quick and does not require manual tuning. Electrode locations will later be projected onto the surface of the head model using ft\_prepare\_leadfield so do not worry if electrodes do not exactly lie on surface of head model.

## Create the headmodel

This step takes (1-2 mins). I used the following table to reference conductivity values and hardcode them so that it automatically checks the tissue order and finds the matching conductivity value for that tissue.

### Conductivities

  
*Vorwerk et al 2019*

## Create the sourcemodel

Steps 8 & 9 are done in DIPFIT, so don't need to compute here---

BUT I recommend doing here in Fieldtrip because dipfit is dumb and recomputes things it doesn't need to. It would take about 30days/subject at the rate dipfit goes currently FYI. Had to modify functions within pop\_multifit to allow for other inputs like sourcemodel and electrode file .mat files

cfg = [];

cfg.resolution = 7.5;

cfg.threshold = 0.1;

cfg.smooth = 5;

cfg.headmodel = headmodel\_fem;

cfg.inwardshift = 1; %shifts dipoles away from surfaces

sourcemodel = ft\_prepare\_sourcemodel(cfg);

## Compute leadfield

This step takes a while! When I ran 128 channels, ft\_prepare\_leadfield took ~8.5 hrs compute the transfer matrix. When using parfor, 2 workers, took ~5 hrs. Makes my computer freeze for a couple minutes, but always comes around :). If you want to modify ft\_prepare\_leadfield to process things in parallel, see [this section](#_Implementing_parallel_processing).

[headmodel\_fem\_tr, elec]=ft\_prepare\_vol\_sens(headmodel\_fem,elec\_aligned);

cfg = [];

cfg.sourcemodel = sourcemodel; %NJ changed cfg.grid to cfg.sourcemodel

cfg.headmodel= headmodel\_fem\_tr;

cfg.elec = elec\_aligned;

cfg.reducerank = 3;%modify the leadfields by reducing the rank (i.e. %remove the weakest orientation),default = 3 for EEG

leadfield = ft\_prepare\_leadfield(cfg);

# Loading Custom Head Model in Dipfit

Modifications first must be made to dipfit to get it to work with this head model. [See section on modifications](#_Dipfit_4.0_modifications).

In the DIPFIT settings, select:

* Model🡪 Custom template
* Model file 🡪 Browse🡪 select *headmodel.mat* filecreated in [Step 9](#_Compute_leadfield) with ft\_prepare\_vol\_sens()
* Output coordinates 🡪 CTF
* MRI 🡪 Browse🡪 select MRI used to make the head model (normalized to MNI)
  + Template channel locs 🡪 Browse🡪 *select .mat file with electrode locations realigned to head model* (e.g. elec\_aligned.mat) created in [Step 6](#_Align_Electrodes).

'hdmfile' - [string] file containing a head model compatible with

the Fieldtrip dipolefitting() function ("vol" entry)

dipolefitting()

DIPPLOT does not actually allow you to align your electrode locations to the head model itself, but rather allows you to align your electrode locations to matching template electrode locations associated with the head model.

**Note about fiducials:** Your channel structure may contain standard fiducial locations (nasion and pre-auricular points). If you import a channel file with fiducial locations into the channel editor, in EEGLAB v4.6- give them the standard 'fiducial' channel type "**FID**" and they will be stored in the channel information structure, **EEG.chaninfo**. This will also be done automatically if your fiducials have the standard names, "**Nz**" (nasion), "**LPA**" (left pre-auricular point), and "**RPA**" (right pre-auricular point ). Note that fiducial locations are stored outside the standard channel location structure, **EEG.chanlocs**, for compatibility with other EEGLAB plotting functions.  
  
Thereafter, fiducial locations will appear in the channel co-registration window (above) and may be used (in place of location-matched scalp channels) to align your electrode montage to the template locations associated with the head model. Use the "**Align fiducials**" button to do this. Press "**OK**" to update the DIPFIT settings window. This will display the resulting talairach transformation matrix, a vector comprised of nine fields named **[shiftx shifty shiftz pitch roll yaw scalex scaley scalez]**. Then press "**OK**" in the DIPFIT settings window and proceed to localization.

## Computation costs & potential for parallel processing

 ‘The most time-consuming steps were the computation of the transfer matrix (in *ft\_prepare\_vol\_sens*) and the leadfield matrix (*ft\_prepare\_leadfield*), with a time effort of about 6 h 29 min and 43 min, respectively. However, both steps can be easily parallelized within MATLAB with an optimal speed-up by using parallel loops (*parfor*). Several lines of the transfer matrix and several forward solutions can thereby be computed in parallel. For a fully parallel implementation, an overall computation time of less than one hour can already be achieved with an eight-core CPU, which can nowadays even be found in portable computers.” [– J. Vorwerk et al, 2018](https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-018-0463-y)

## Implementing parallel processing

Here are instructions on how to speed up ft\_prepare\_vol\_sens and ft\_prepare\_leadfield using parallel processing. This was written using Matlab v R2018a. There are several lines of the transfer matrix that can be computed in parallel. Modified: **fieldtrip-20210614\ft\_prepare\_leadfield.m line 316**

%% NJ commented out for loop

%for i=1:length(insideindx)

% % compute the leadfield on all sourcemodel positions inside the brain

% ft\_progress(i/length(insideindx), 'computing leadfield %d/%d\n', i, length(insideindx));

% thisindx = insideindx(i);

% sourcemodel.leadfield{thisindx} = ft\_compute\_leadfield(sourcemodel.pos(thisindx,:), sens, headmodel, leadfieldopt{:});

% end % for all sourcemodel locations inside the brain

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%NJ - implement parallel computation

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%setup parallel environment

parpool('local',3) %NJ, local workspace, X workers

parfor i=1:length(insideindx)

% compute the leadfield on all sourcemodel positions inside the brain

fprintf('Computing leadfield %i/%i)', i, length(insideindx)); %NJ added progress update

%ft\_progress(i/length(insideindx), 'computing leadfield %d/%d\n', i, length(insideindx)); %NJ commented out bc of error

thisindx = insideindx(i);

%sourcemodel.leadfield{thisindx} = ft\_compute\_leadfield(sourcemodel.pos(thisindx,:), …

%sens, headmodel, leadfieldopt{:});

grid\_leadfield{i} = ft\_compute\_leadfield(sourcemodel.pos(thisindx,:), …

sens, headmodel, leadfieldopt{:});

end % for all sourcemodel locations inside the brain

sourcemodel.leadfield(insideindx) = grid\_leadfield;

delete(gcp('nocreate')) %NJ, close parallel pool

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Dipfit 4.0 modifications

Fail attempts:

Adding EEG.dipfit.elecfile.cfg.senstyp = ‘eeg’ or EEG.dipfit.elecfile.senstype = ‘eeg’ to dipfit 🡪 did not fix the issue.

'Error using ft\_notification (line 345)

Cannot determine which sensors you want to work on. Specify cfg.senstype as 'meg', 'eeg' or 'nirs''

**Solution: manually hardcode sensor type in dipfit function (temporary fix)**

dipfit\_gridsearch() –

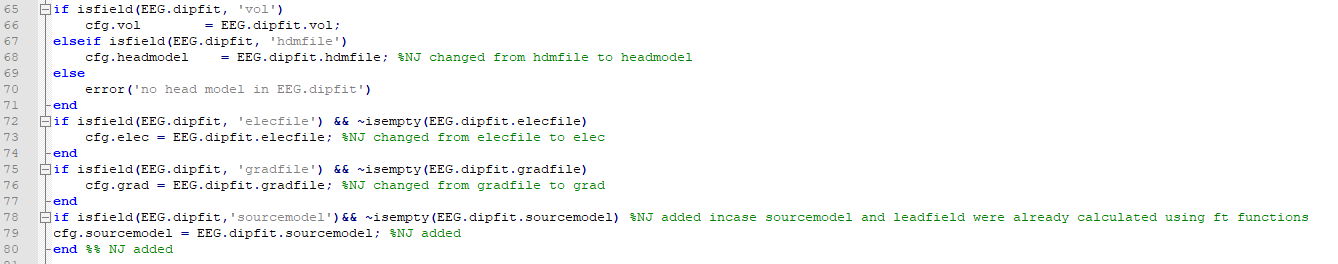
* cfg.senstype = 'eeg'; %NJ add, to make sure it knew to use eeg sensors

Other modifications:

* L 72 - cfg.elec = EEG.dipfit.elecfile; %NJ changed from elecfile to elec (FT will also fix this automatically after giving you a warning)
* L 75 - cfg.grad = EEG.dipfit.gradfile; %NJ changed from gradfile to grad (FT will also fix this automatically after giving you a warning)
* L93 %cfg.gradfile = EEG.dipfit.chanfile; %NJ commented out because it kept thinking I had meg sensors
* if isfield(EEG.dipfit,'sourcemodel')&& ~isempty(EEG.dipfit.sourcemodel)

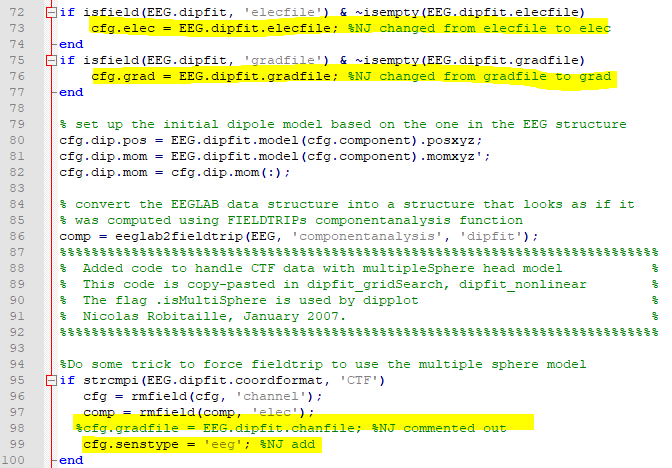
cfg.sourcemodel = EEG.dipfit.sourcemodel;

end %% added incase sourcemodel and leadfield were already calculated, SAVES TIME



**dipfit\_nonlinear () – *note: similar changes as dipfit\_gridsearch to setup right configuration.***

* L93 %cfg.gradfile = EEG.dipfit.chanfile; %NJ commented out because it kept thinking I had meg sensors
* cfg.senstype = 'eeg'; %NJ add, to make sure it knew to use eeg sensors



**C:\Users\jacobsen.noelle\Desktop\eeglab2021.0\plugins\Fieldtrip-lite20210601\forward\ft\_prepare\_vol\_sens**

# Trouble shooting

* Make sure headmodel file has mesh too??

1. **Issue with MEX files**

Steps I took

* <https://www.mathworks.com/matlabcentral/answers/98351-how-can-i-set-up-microsoft-visual-studio-2008-express-edition-for-use-with-matlab-7-7-r2008b-on-64>
* Installed Microsoft Visual Studio 2008 Express , C++ and SDK from Microsoft website
  + This is required because x64 platform doesn't have a compiler..?
* Check for Microsoft/Windows updates
* Setup mex on matlab
  + 'mex -setup'
  + Chose one that said C+
  + Test compiler using example given in mathworks link
* The ft\_compile\_mex function is used to compile the mex files and the synchronize-private.sh Bash script is used to copy the updated mex files to all required (private) directories.

Things still weren't working, so I installed Microsoft Visual C++ 2008 Service Pack 1 Redistributable Package MFC Security Update (not sure what the difference is between that and Microsoft Visual Studio 2008 Express. Seems like the ft\_compile\_mex function didn't compile the mex files that I was having errors with

* I found the missing .dll files in other folders on my PC and copied them over to the fieldtrip subfolder 'C:\Users\jacobsen.noelle\Desktop\fieldtrip-20210614\external\simbio.
* I was then getting an error:

"Invalid MEX-file '...\MexFileName.mexw64': The specified module could not be found"

* After changing my Matlab working directory to the same folder containing the mex file, the function ran perfectly fine.

1. **Error using ft\_notification (line 345)**

**no electrodes, gradiometers or optodes specified.**

**Solution:** manually add electrode locations that were realigned to the MRI coordinate system. Elec\_realigned variable is a structure with rows [ 1: Num of electrodes] and columns: [ labels, X,Y, Z]

EEG.dipfit.elec = elec\_realigned;

1. **Maximum number of iterations reached. Fitting failed**

**“***Edit 'dipole\_fit' line 117 and 119 'maxiter'. These numbers could be x10*

>> *and it usually solves the problem.* [*-Makoto*](https://sccn.ucsd.edu/pipermail/eeglablist/2015/009529.html)

**[FieldTrip] issue with FieldTrip-SimBio pipeline for EEG FEM forward solution**

https://mailman.science.ru.nl/pipermail/fieldtrip/2019-February/025697.html