

# A Comprehensive Head Model Pipeline

Alexandra Warner, Chris Johnson

University of Utah, Salt Lake City UT

December 9, 2017

## Abstract

Here is some abstract text

## 1 Introduction

Head models are built and used in many research areas such as imaging, mesh generation, and any bioelectric research topic that conducts simulations such as forward and inverse problems (ref). Imaging is necessary for building any head model, and finding a dataset that includes all imaging modalities is rare since most research projects will only include the modalities necessary to their specific project. In each case, a new head model must be built and this is a time consuming task.

In this project, a comprehensive pipeline to build a complete, high-resolution head model is described in great detail. Steps include image acquisition, preprocessing and registration, image segmentation, finite element mesh generation, and simulations and visualization. Included in this paper, there are two 3-dimensional tetrahedral finite element meshes that are of different resolutions. These are used to solve the forward problem. There are visualizations of the forward problem, with isotropic and anisotropic systems, functional MR data onto a tetrahedral mesh, electroencephalography (EEG) onto electrodes, and of diffusion tensor data. There is a complete, high-resolution brain segmentation that can be used to create 3-dimensional meshes.

Along with outlining the pipeline, this project will be released an open-source dataset. This project took roughly a year to complete due to the many options in software/techniques and the complexity of the data. Segmentation took roughly 100 hours because the imaging was done at the highest resolution available, and needed a lot of manual work. Meshing took several weeks to find an appropriate size without holes. Since the segmentation was complex, and high-resolution, tetrahedral meshes become extremely large which make for very slow simulations. Further simplification methods failed because of the number of different tissues in the head. Registration of diffusion tensor data and functional MR data was difficult because transformations from other software packages were not compatible with SCIRun and functional MR data has previously not been used in SCIRun. This dataset could provide a starting point for other researchers to begin their projects without the hassle of building an entire model.

## 2 Methods

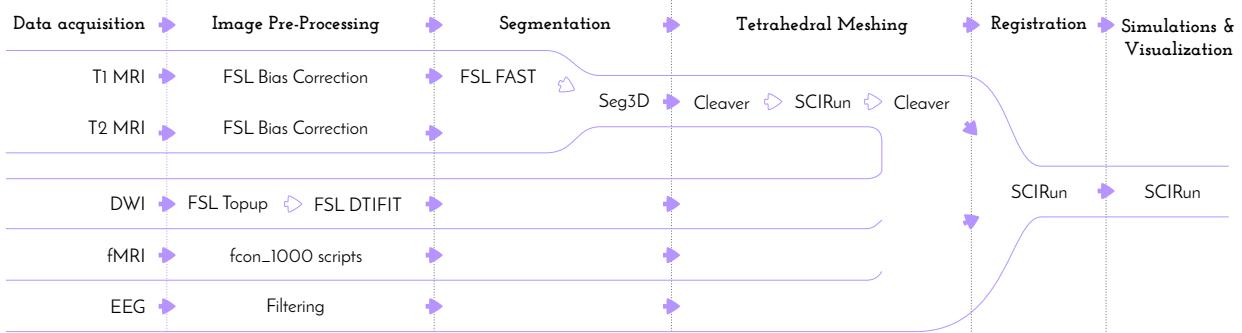


Figure 1: Head/Brain Model Pipeline

### 2.1 Data Acquisition

To construct a high-resolution, personalized, anisotropic volume conductor whole-head model,  $T_1$ -,  $T_2$ - weighted, diffusion weighted, and functional magnetic resonance (MRI) scans were acquired on a healthy, female volunteer who is 23 years of age on a Skyra 3T full-body scanner (Siemens Medical Solutions).

The  $T_1$ -weighted scan was performed with a 3D magnetization prepared rapid gradient echo (MPRAGE) sequence (Mugler and Brookman, 1990) – get this reference. The parameters used were as follows: echo time: 3.41ms, repetition time: 2500ms, flip angle: 7 °, resolution matrix size: 256x256 pixels, field of view: 256mm, 208 sagittal slices with a slice thickness of 1mm. Acquisition time was 10:42 minutes.

The  $T_2$ -weighted scan was performed with a sampling perfection with application-optimized contrast using different flip angle evolutions (check ref) (SPACE) sequence (Lichy et. al., 2005; Mugler et al., 2000). The parameters used were as follows: echo time: 406ms, repetition time: 3200ms, resolution matrix size: 256x256 pixels, field of view: 256mm, 208 sagittal slices with a slice thickness of 1mm. Acquisition time was 5:34 minutes. The subject did not move in between the two scans so the scans did not need to be registered.

The diffusion weighted images (DWI) were acquired with multiband two-dimensional echo-planar imaging (EPI). Both phase encoding directions were performed (anterior to posterior and posterior to anterior) with 64 diffusion directions each. Further sequence parameters for each scan was as follows: echo time: 76.8ms, repetition time: 4070ms, flip angle: 90 °, resolution matrix size: 104x104 pixels, field of view: 208mm, 60 slices with 2.5mm slice thickness. Acquisition time was 5:05 minutes each.

The function MRI (fMRI) scans were acquired with blood oxygenation level dependent contrast (BOLD). The following parameters were used: echo time: 76.8ms, repetition time: 780ms, flip angle: 55 °, resolution matrix size: 104x104 pixels, field of view: 210mm, 72 slices with 2mm slice thickness. Acquisition time was 10:32 minutes.

A continuous electroencephalogram (EEG) was recorded using a 256-channel HydroCel Geodesic Sensor Net that was connected to a NetAmps 400 amplifier and referenced online to a single vertex electrode shown in Figure 2. Channel impedances were kept at or below 50 kOhms and signals

were sampled at 250Hz. The EEG was recorded while the subject sat quietly in a chair, alternating two minute epochs of eyes open and eyes closed for a total of 12 minutes.

All acquisition reports will be included with the dataset.



Figure 2: 256-channel HydroCel Geodesic Sensor Net on Subject

## 2.2 Preprocessing of Images

### 2.2.1 MRI Correction

Bias field signal is a low-frequency, smooth signal that corrupts MRI images due to inhomogeneities in the magnetic fields of the MRI machine by blurring images reducing the high frequencies of the images such as edges and contours. It changes the intensity values of image pixels so that the same tissue has a different distribution of grayscale intensities across the image. (ref) An estimated bias field correction on the  $T_1$  and  $T_2$  MRI's was done using FMRIB Software Library (FSL) FAST, and will be further described in the Section 2.3. Figure 3 shows FSL's basic user interface, which will be used multiple times throughout this pipeline.

### 2.2.2 DWI Distortion Correction

Diffusion weighted images performed with EPI sequences are prone to distortions from rapid switching of diffusion weighting gradients, movement from the scanning table, and movement from the subject. The diffusion data was collected with reversed phase-encode blips (anterior to posterior (AP) and posterior to anterior(PA)), resulting in pairs of images with distortions going in opposite directions. From these pairs the susceptibility-induced off-resonance

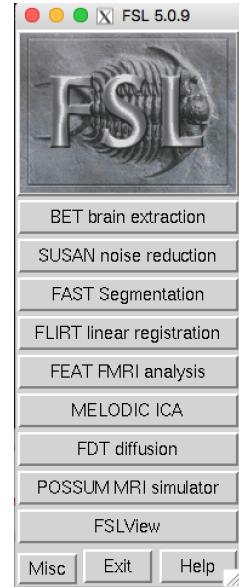


Figure 3: FMRIB Software Library User Interface

field was estimated using a method similar to that described in [Andersson 2003] as implemented in FSL [Smith 2004] and the two images were combined into a single corrected one. This was accounted for using FSL's topup and eddy command line tools.

Before running these tools, an acquisition parameters text file needs to be created with the FSL-defined total readout time.

```
0 -1 0 0.0345
0 1 0 0.0345
```

Figure 4: Acquisition Parameters Text File

There are two parameters that are frequently needed when calculating and applying field maps: the effective echo spacing and the total readout time for an EPI sequence. "Effective" echo spacing was used, rather than the actual echo spacing, in order to include the effects of parallel imaging, phase oversampling, etc. Multiplying this by the size of the reconstructed image in the phase direction will give you the reciprocal of the effective echo spacing:

$$\text{Effective Echo Spacing (s)} = 1 / (\text{BandwidthPerPixelPhaseEncode} * \text{MatrixSizePhase})$$

The total readout time (FSL definition) is:

$$\text{Total readout time (FSL)} = (\text{MatrixSizePhase} - 1) * \text{EffectiveEchoSpacing}$$

MRIConvert (ref) is a software package that provides all of the acquisition information about a dicom series, as well as converts to a NiFTI format, including effective echo spacing and total readout time. To obtain this information, first load in the dicom series for either DWI acquisition. Choose "Options" to ensure that the DWI will be saved as a NiFTI, and click "Convert All." This will save all the files into the output directory specified upon opening MRIConvert. The text file will include the FSL-defined total readout time which should be contained in the acquisition parameter file in seconds. MRIConvert also outputs the b-values and b-vectors files, and they should be the same for both the DWI AP and DWI PA scans. The last input file needed is an "index.txt" file. This text file contains one column with 65 rows (for 64 directions plus the b0 image) of 1's.

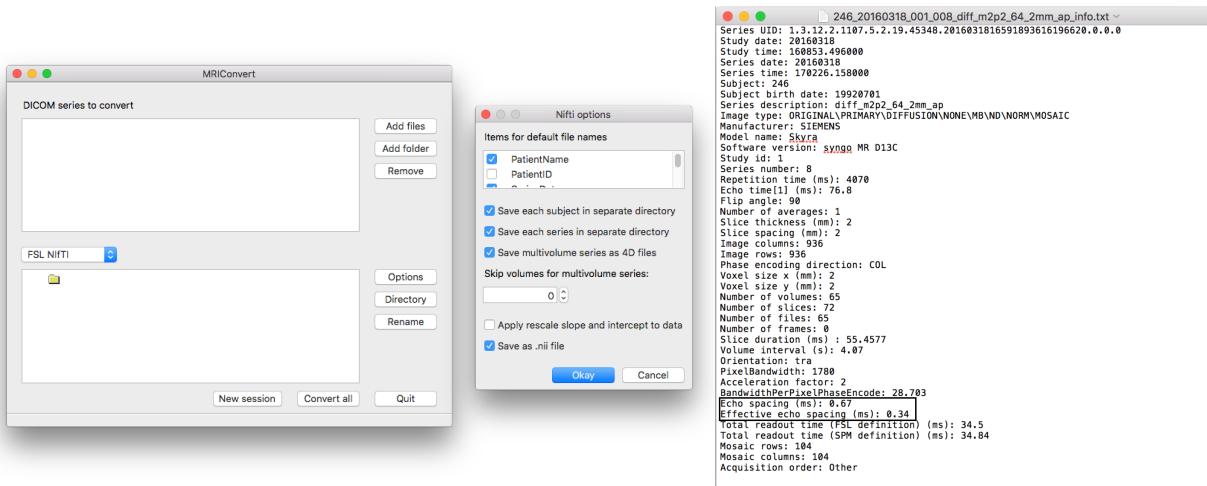


Figure 5: MRIConvert (*left*) with Options (*middle*) & Output (*right*)

Create a separate folder for topup results and include the following files: the acquisition parameters file, the index file, b-values, b-vectors, and the DWI AP and DWI PA files. In the following instructions, DWI AP was renamed as DWI\_up and DWIAP was renamed to DWI\_down. The b-values and b-vectors were renamed to dwi.bval and dwi.bvec, respectively. After all of these files are in place, run the following command line commands.

```
fslroi DWI\_up b0\_up 0 1
fslroi DWI\_down b0\_down 0 1

fslmerge -t both\_b0 b0\_up b0\_down

topup --imain=both\_b0 --datain=acq_params.txt --config=mine.cnf --out=topup\_results
applytopup --imain=b0\_up,b0\_down --inindex=1,2 --datain=acq_params.txt
--topup=topup\_results --out=b0\_hifi

bet b0\_hifi b0\_hifi\_brain -m -f 0.2
eddy --imain=DWI\_up --mask=b0\_hifi\_brain\_mask --index=index.txt --acqp=acq_params.txt
--bvecs=dwi.bvec --bvals=dwi.bval --fwhm=0 --topup=topup\_results --flm=quadratic
--out=eddy\_unwarped
```

These commands first obtain the b0 image, which is the baseline image used for calculating field maps, for both encoding directions. Then the two b0's are merged together into one file. Topup and eddy are applied for distortion correction. Bet is applied for brain extraction. The distortion corrected file is named "eddy\_unwarped.nii."

### 2.2.3 Diffusion Tensor Images

After the DWI images have been corrected, diffusion tensor images (DTI) are calculated using FSL's DTIFIT. Upon opening FSL choose "FDT Diffusion." Then choose "DTIFIT Reconstruct diffusion tensors" in the drop down menu, select to input files manually, and put in the files in Table 1. (reference and figure of DTIFIT)

Table 1: My caption

DTIFIT Input Files	
Diffusion weighted data:	eddy_unwarped.nii
BET binary brain mask:	b0_hifi_brain_mask.nii
Output basename	desired output location
Gradient directions:	dwi.bvec
b values:	dwi.bval

DTIFIT will output the eigenvalues (named L1, L2, and L3) and the eigenvectors (named V1, V2, and V3) for the diffusion tensor field. The files were converted from NiFTI format to nrrd format using ITK-SNAP (reference) although there is a loss of precision. The files were then input into SCIRun to build the tensor field using the eigenvalue and eigenvectors. The SCIRun CalculateFieldData module only requires two eigenvectors as input because it calculates the third eigenvector automatically since it should be orthogonal to the first two. This process can be done in either SCIRun 4 or SCIRun 5 and the same results are produced.

The tensor field was built in SCIRun rather than in 3D Slicer (reference) or FSL DTIFIT because the output data would be "backwards" and couldn't be registered with the mesh.

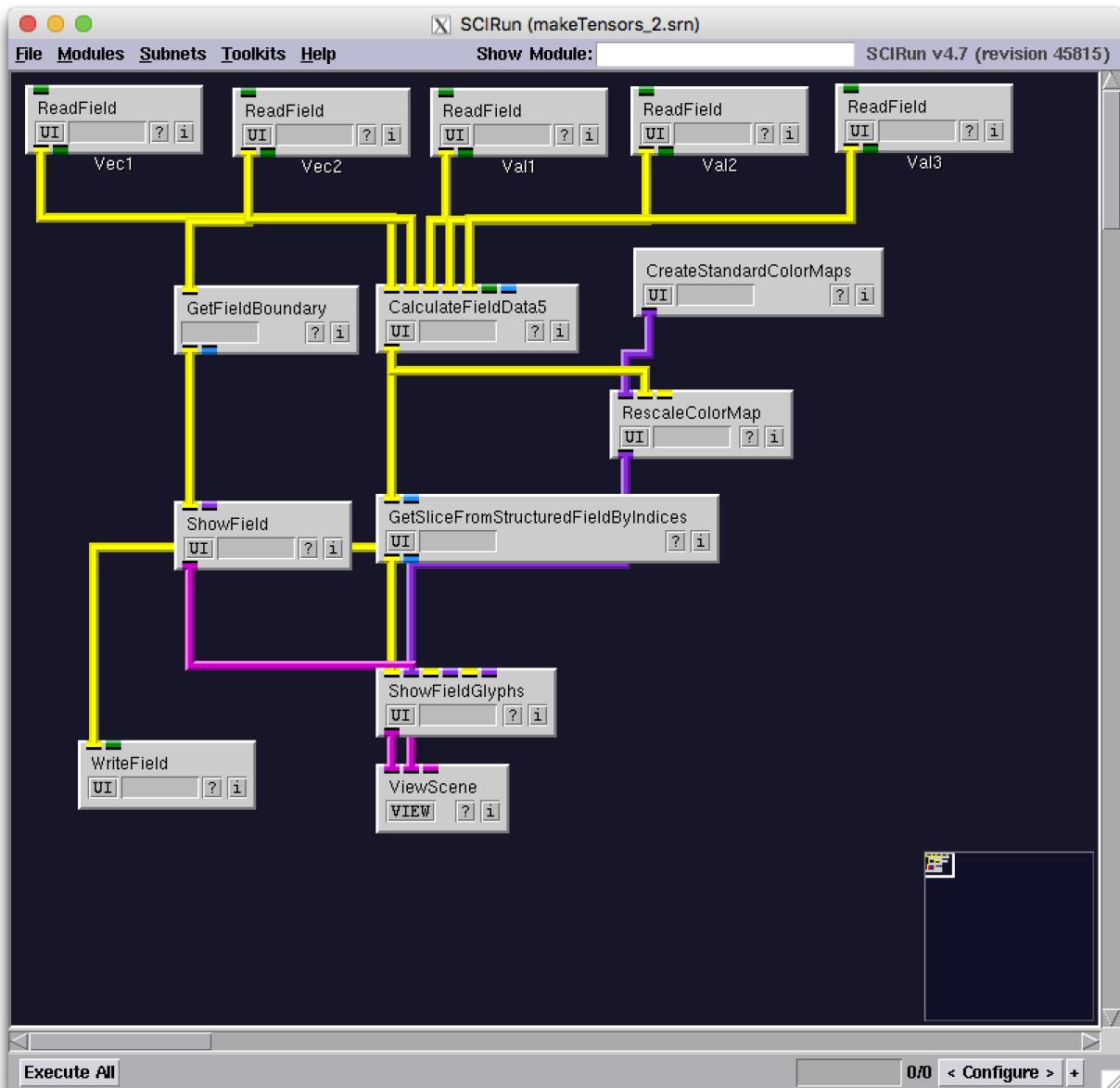


Figure 6: default

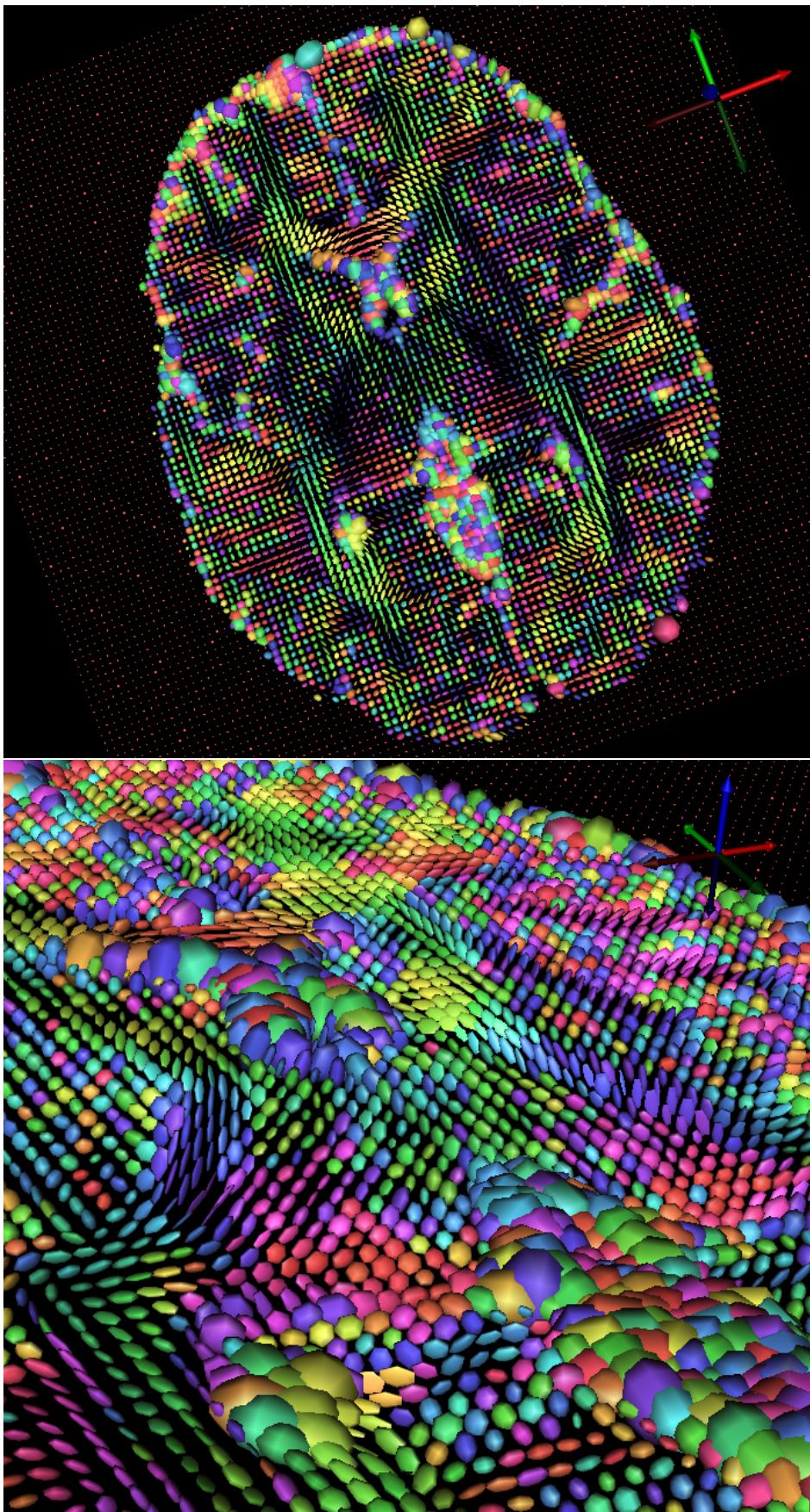


Figure 7: default

#### **2.2.4 fMRI**

fMRI data was preprocessed using the 1000 Functional Connectomes Project pipeline scripts (reference) which do anatomical preprocessing, functional preprocessing, registration to the  $T_1$  MRI, segmentation, and nuisance signal regress. The outline pipeline used on this fMRI dataset, specific to the University of Utah, can be found at [https://bitbucket.org/UtahBrainNetworks/base\\_prep](https://bitbucket.org/UtahBrainNetworks/base_prep) and includes specific instructions for installation, compilation, and usage.

After running fMRI data through the pipeline “rest.nii”, the preprocessed fMRI file, was opened in Matlab using the “load\_nii(‘rest.nii’)” function within the NiFTI toolbox. The 4D “img” variable ( $x, y, z, t$ ) is then resized to a 2D variable ( $x*y*z,t$ ) and saved to use in SCIRun. (reference for NiFTI toolbox?)

#### **2.2.5 EEG**

The filetype of the EEG recordings in an .edf file after it had a 60Hz notch filter and its harmonics. The EEG signal matrix was obtained using a Matlab script called “edfRead.m.” To run this script use the following command “[hdr, record] = edfread(fname).” The variable ‘record’ will contain the signals. The last two rows of the matrix were removed because they did not correspond to EEG electrodes. Also the beginning and end of the experiment were cut out of the matrix when the EEG net is being put on and taken off.

#### **2.2.6 Registration**

Since the subject did not move in between the  $T_1$  and  $T_2$  MRI, no registration was necessary before segmentation and meshing. The tetrahedral mesh was generated in its own coordinate space from the segmentation, and was registered to the DTI coordinate space with a rigid registration using SCIRun. The fMRI data was registered to the mesh coordinate space with a rigid registration using SCIRun. The fMRI data can use same transform to register to DTI later if desired. The SCIRun networks for registration are included in Section 2.6.

### **2.3 MRI Segmentation of Tissues**

Segmentation of the head tissues proved to be the most time consuming section of the pipeline. The head volume was segmented into air, cerebral spinal fluid (CSF), white matter, grey matter, skull, sinus, eyes, and scalp. Segmentation of the brain can be difficult due to the similar intensities of the different tissues, making merely applying a median filter and thresholding the image not enough.

The brain was initially segmented by inputting a skull stripped  $T_1$  MRI into FSL FAST Segmentation. This outputs CSF, white matter, and grey matter layers as well as a bias-corrected  $T_1$  MRI. This method, compared with Freesurfer, Statistical Parametric Mapping through Matlab (SPM), Atlas Based Classification through 3D Slicer, and Seg3D methods alone, produced the best initial brain segmentation results for this data due to how well it filled in each tissue.

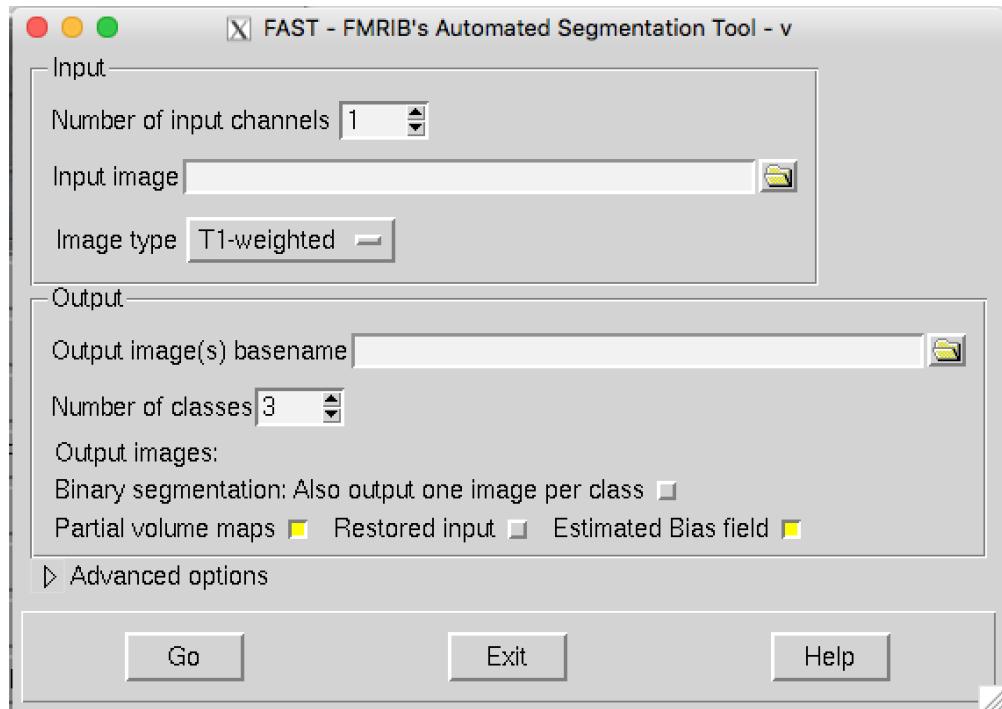


Figure 8: FSL FAST User Interface

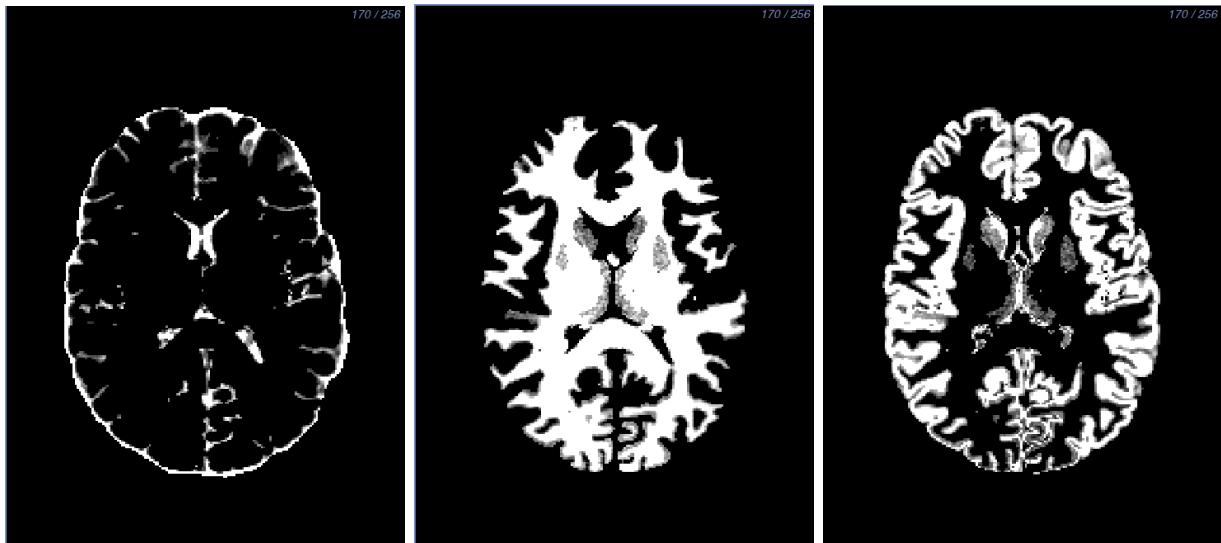


Figure 9: default

Although the FSL Fast results were a great improvement compared to the other segmentation trials, manual segmentation still needed to be done on those layers to add more detail and take out any cross over between the layers. Since white matter is the innermost layer, it was worked on first. All manual editing was done using Seg3D software. (ref) First a threshold layer was created from

the FSL Fast output. Every slice in every direction was inspected and manually edited whether that was adding more detail that could be seen with the naked eye or cleaning up noise from FSL Fast. This manual editing of the white matter took roughly 40 hours of work.

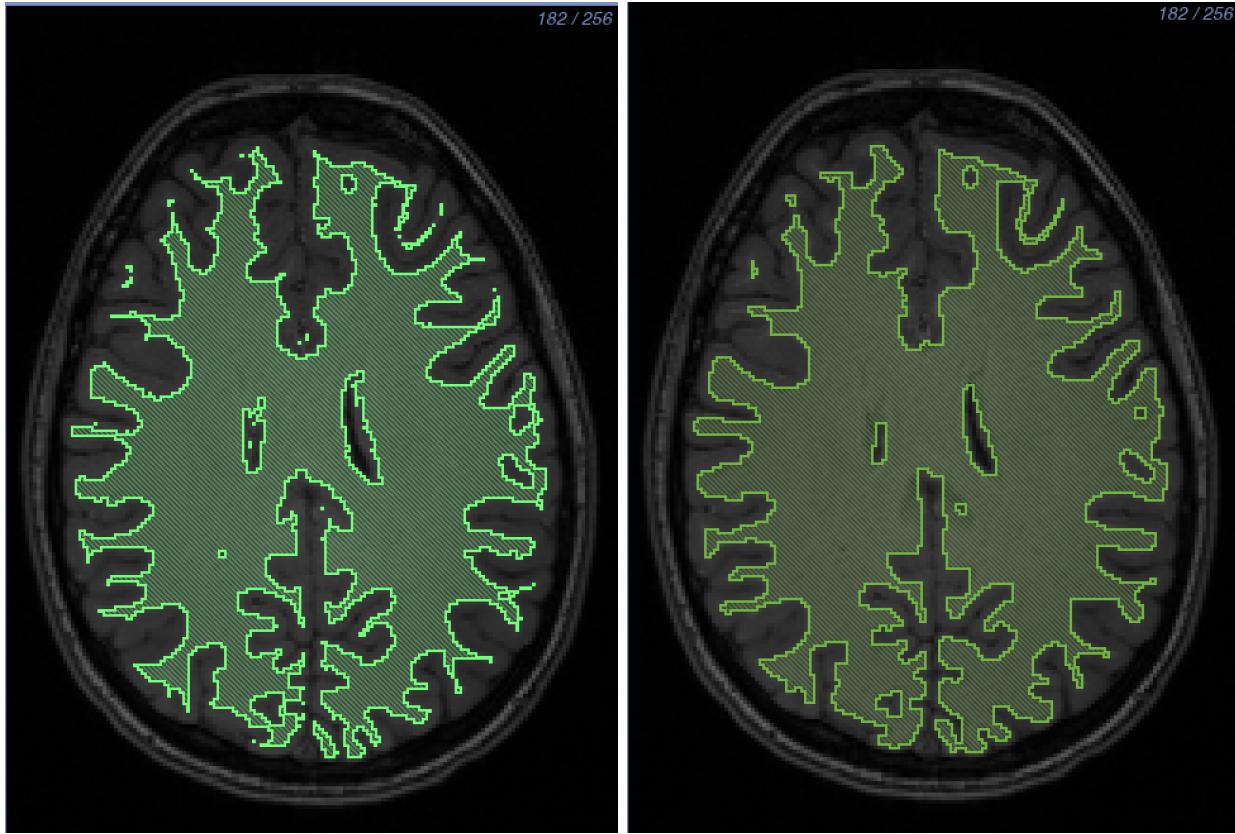


Figure 10: default

After the white matter was completed, a threshold layer for grey matter was created from the FSL Fast output. Each slice in every direction of the grey matter was inspected as well. The white matter layer was removed from the grey matter using a boolean remove mask filter. Any holes between the two layers were decided manually. More detail was added to the grey matter folds to add to the CSF layer as well. The last part of editing the grey matter was decided to add a grey matter nucleus to the layer. The thresholding algorithms generated a lot of noise around these nuclei so they were segmented by hand, using the paintbrush tool, and added to the grey matter layer with a boolean or mask filter. The nuclei were also removed from the white matter layer using a boolean remove mask filter. The manual editing of the grey matter took roughly 20 hours of work. (figures of grey matter and nuclei)

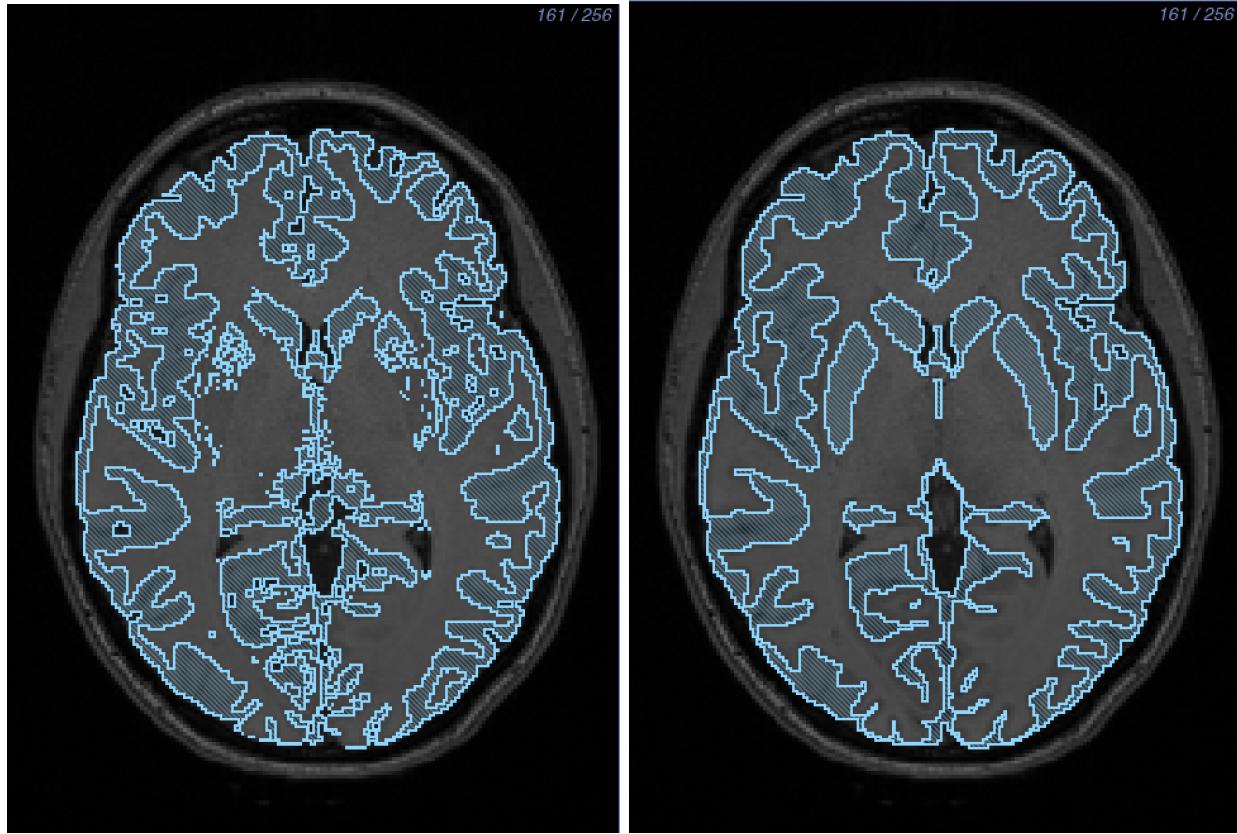


Figure 11: default

After the grey and white matter layers were completed, the CSF layer was made by creating a solid threshold layer for the entire brain and removing the white and grey matter layers using a boolean remove mask filter. The white matter, grey matter, and CSF layers were then checked for holes, whether on the surface or the inside of the segmentation. Also a quality check on the layers was performed to ensure that the layers were at least two pixels wide. This is an important note for creating a hole-less tetrahedral mesh. The creation of the CSF layer and the manual editing and hole checking took roughly 4 hours of work.

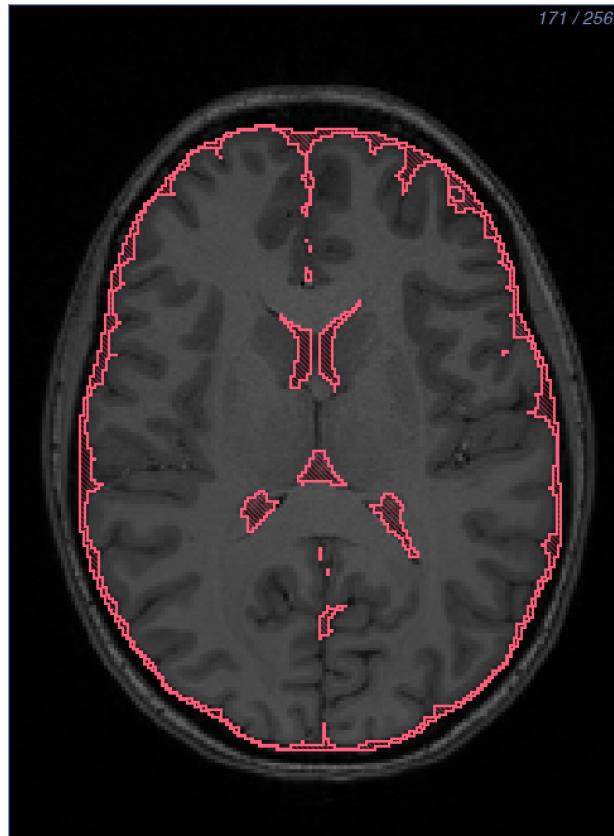


Figure 12: default

The skull and the sinus layers are the most difficult to segment using only an MRI because they both appear black in the image, and the volunteer did not have a CT scan. (define?) The first attempt to create a bone layer was first to use FSL's skull stripping using the BET2 tool to create a skull. Then to threshold the remainder of the bones in Seg3D from the  $T_1$  MRI and connect it to the skull from FSL. Although this gave a decent skull for only having an MRI layer, the method to segment sinus layer was still to be determined. As a second method, the skull was estimated from an MR-based synthetic pseudo-CT. An improved iterative version of the patch-based method was used described by Torrado-Carvajal et al. [ref] that takes the  $T_1$  and  $T_2$  images as input, and synthesize the pseudo-CT based on both images providing more refined and accurate bone boundaries.

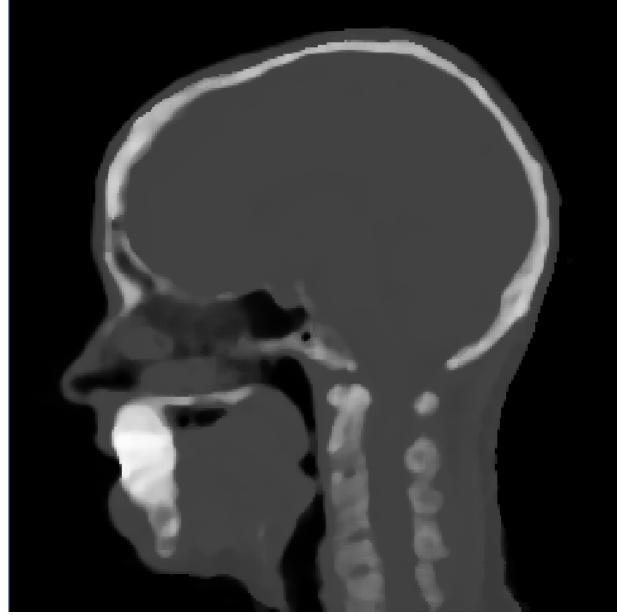


Figure 13: default

This method gave a good starting place for skull segmentation, but still needed manual editing. After using a median filter with one pixel radius and thresholding, each slice in each direction was manually edited by hand. Since the volunteer has a permanent retainer in their mouth, the mouth was segmented as solid bone for now. This is not concerning because the EEG cap used did not cover the volunteer's mouth. The psuedo-CT image also provided a segmentation of the sinuses and esophagus by thresholding the black pixels. After the thresholding, the sinus layer was also manually edited. Quality checks were done on both layers to ensure that there were no holes and that the layers were at least two pixels thick. Manually editing the second skull and the sinuses took roughly 30-40 hours of work.

The eyes, skin, and air layers were simple, in comparison, to segment. The eyes were easily segmented by thresholding the  $T_2$  MRI. The skin layer was segmented by thresholding the entire volume and removing all of the previous layers using a boolean remove mask filter. A quality check was performed on the skin layer to ensure that it was at least two pixels thick. The important places to check for this is at the bridge of the nose and the bottom of the chin. Last, the air was segmented by thresholding the entire image and removing the solid skin layer. There was a check to ensure that the segmentation did not contain any holes between layers after they were removed. To create these three layers took roughly 8 hours of work, most of which was ensuring there were no holes in the segmentation. This is imperative to creating a quality mesh.

Table 2: My caption

Segmented Tissue	Amount of Work (hrs)
White Matter:	40
Grey Matter	20
CSF	4
Skull and Sinus	35
Eyes, Scalp, & Air	8

## 2.4 Finite Element Mesh Generation

Segmentations were used to generate realistic 3D geometries for use in subsequent finite element simulations. A smooth, linear, subject-specific, boundary-conforming, tetrahedral finite element mesh was generated using Cleaver software (ref and figure) on a Late 2013 Mac Pro with a 2.7 Ghz 12 Core Intel Xeon E5 processor, 64 GB of RAM, and an AMD FirePro graphics card. Cleaver is a multimaterial meshing package that produces structured meshes of tetrahedral elements with guaranteed minimum element angles, resulting in quality meshes that require fewer computational resources. To make a very high resolution mesh with no holes the following parameters were used:

Table 3: My caption

Cleaver Settings (high resolution)	
scaling factor:	0.6
size multiplier:	1.0
lipschitz:	0.2
padding:	0
element sizing method:	adaptive

Along with these parameters, indicator functions must be input to Cleaver. These are made by creating distance maps in Seg3D. Ensure the map is inverted before making the distance map. To reduce the size of the mesh, a mesh was made with a scaling factor of 1.0 with the remainder of the parameters as described before. The computing sizing field was exported from Cleaver and manipulated using SCIRun4 (ref and figure of network) by changing how quickly the elements increase in size. This was done by multiplying the scaling by a factor of 27. The changed sizing field was then input into Cleaver with the same indicator functions and cleaved a new mesh.

## 2.5 Mathematical Modeling

The head mesh, with associated inhomogeneous and anisotropic regions, was used as a volume conductor to solve the following boundary value problem:

$$\nabla \cdot \sigma \nabla \Phi = -I_V \quad \text{in } \Omega, \tag{1}$$

where  $\Phi$  is the electrostatic potential,  $\sigma$  is the electrical conductivity tensor, and  $I_V$  is the current per unit volume defined within the solution domain,  $\Omega$ . Equation (1) is solved for  $\Phi$  with a known description of  $I_V$  and the Neumann boundary condition:

$$\sigma \nabla \Phi \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_T, \tag{2}$$

Table 4: My caption

Tissue Type	Isotropic Conductivity ( $S/m$ )
White matter	0.1429
Grey Matter	0.3333
Cerebrospinal Fluid (CSF)	1.79
Skull	0.001
Skin	0.4346
Sinus	1e-6
Eyes	0.5051

which says that the normal component of the electric field is zero on the surface interfacing with air (here denoted by  $\Gamma_T$ ). The brain and surrounding tissue and skull were discretized, and using dipoles for current source the electrical field was calculated within the brain and then projected onto the surface of the scalp.

### 2.5.1 Electrical Conductivity Preparation

All electrical conductivities were homogeneous for each tissue with the exception of white matter when using tensor data. The isotropic conductivities are as follows (Rullman):

When DTI tensor data is added, there are two approaches to converting the tensor data to conductivities. The first is scaling the data (ref), and the second is giving the white matter a fixed ratio of conductivity.

$$\sigma_{aniso} = \frac{\sigma_{iso}}{\sqrt[3]{d_1 d_2 d_3}} D \quad (3)$$

where  $D$  is the diffusion data,  $d_i$  is the  $i$ th eigenvalue of  $D$ , and  $\sigma_{iso}$  is the white matter isotropic conductivity.

$$\sigma_{aniso} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ W \end{bmatrix}, W = \begin{bmatrix} \sigma_{iso} \\ \frac{\sigma_{iso}}{10} \\ \frac{\sigma_{iso}}{10} \end{bmatrix} \quad (4)$$

where  $v_i$  is the  $i$ th eigenvector of  $D$  and  $W$  is the white matter ratio vector and the ratio is 10 : 1.

### 2.5.2 Numerical Methods

Solutions to Equation 1 were computed using finite element methods. By applying Green's divergence theorem to Equation 1, the following weak formulation is generated

$$\int ((\bar{\sigma}_e + \bar{\sigma}_i) \nabla \phi_e) \cdot \nabla \psi(\bar{x}) d\bar{x} = - \int (\bar{\sigma}_i \nabla V_m) \cdot \nabla \psi(\bar{x}) d\bar{x}, \quad \forall \psi \in \Omega \quad (5)$$

where,  $\Omega$  is the linear, finite element mesh,  $\psi$  represents the finite element basis functions characterized by local hat functions associated with mesh nodes. By applying this formulation to the finite dimensional mesh, we can reduce Equation 5 to a system of linear equations

$$A\phi_e = -RV_m \quad (6)$$

where  $A$  and  $R$  represent stiffness matrices defined by  $A_{j,k} = \langle \nabla \psi_j, (\bar{\sigma}_e + \bar{\sigma}_i) \nabla \psi_k \rangle_\Omega$  and  $R_{j,k} = \langle \nabla \psi_j, \bar{\sigma}_i \nabla \psi_k \rangle_\Omega$ , while  $\phi_e$  and  $V_m$  represent extracellular and transmembrane potentials, respectively.<sup>7</sup>

We used SCIRun, the open-source problem solving environment, to apply parameters and to solve Equation 6 numerically. Within the SCIRun environment, isotropic and anisotropic conductivity tensors were applied to the mesh as well as inhomogeneous regions, initial and boundary conditions were defined, and border regions were generated in order to compute potentials by way of a conjugate gradient method with a Jacobi preconditioner.

## 2.6 Simulations & Visualization

All simulations and visualization were done in the SCIRun problem solving environment.

### 2.6.1 Forward Problem

When solving systems described in Section 2.5 from a known source to the EEG electrodes, it is called a forward problem. To solve the opposite is to solve an inverse problem and would usually be the case for finding sources for subjects with neurological disorders. In this project, we have set up SCIRun networks to solve forward problems with known sources and to write a lead field matrix for inverse problem networks made in the future. We solve the forward problem with an isotropic system and an anisotropic system, which includes DTI data.

For the isotropic case necessary inputs are the tetrahedral head mesh, isotropic conductivities, the head segmentation, the physical electrode locations, and dipole sources. The physical electrode locations and dipole sources were part of the EEG dataset. The electrodes have the fiducials removed and the only dipole sources that are used are those that the user chooses. There are 4800 dipoles in the data set and there should be 256 electrodes after the fiducials are removed. The head mesh was registered to the head segmentation with a rigid registration and after the flat tets are cut out the conductivities are mapped to their respective tissues. The electrodes and dipoles are registered to the head mesh with the same transform. The system is solved with the mapped data and the chosen dipole sources. The solution is then mapped onto the mesh and the electrodes for visualization. Streamlines and isolines are also included in the visualization.

For the anisotropic case, the network is going to be largely the same with the exception of the diffusion tensor data set used as white matter conductivities after it was scaled as explained in Section 2.5.1 and the DTI to mesh transform is needed as input. The head mesh, electrodes, and dipoles were registered to the DTI space with a rigid registration and the head segmentation wasn't needed.

### 2.6.2 fMRI

The fMRI data is visualized one step at a time using the 2D matrix described in Section 2.2.4 as input and viewing one column at a time. Each column is set onto a lattice volume, which is then rotated 180 degrees, smoothed, thresholded and clipped for easier rigid registration. The fMRI is registered to the tetrahedral mesh using the bounding boxes and manual registration. After registration, the smoothed fMRI data is mapped onto the mesh using a mapping matrix with a linear interpolation basis.

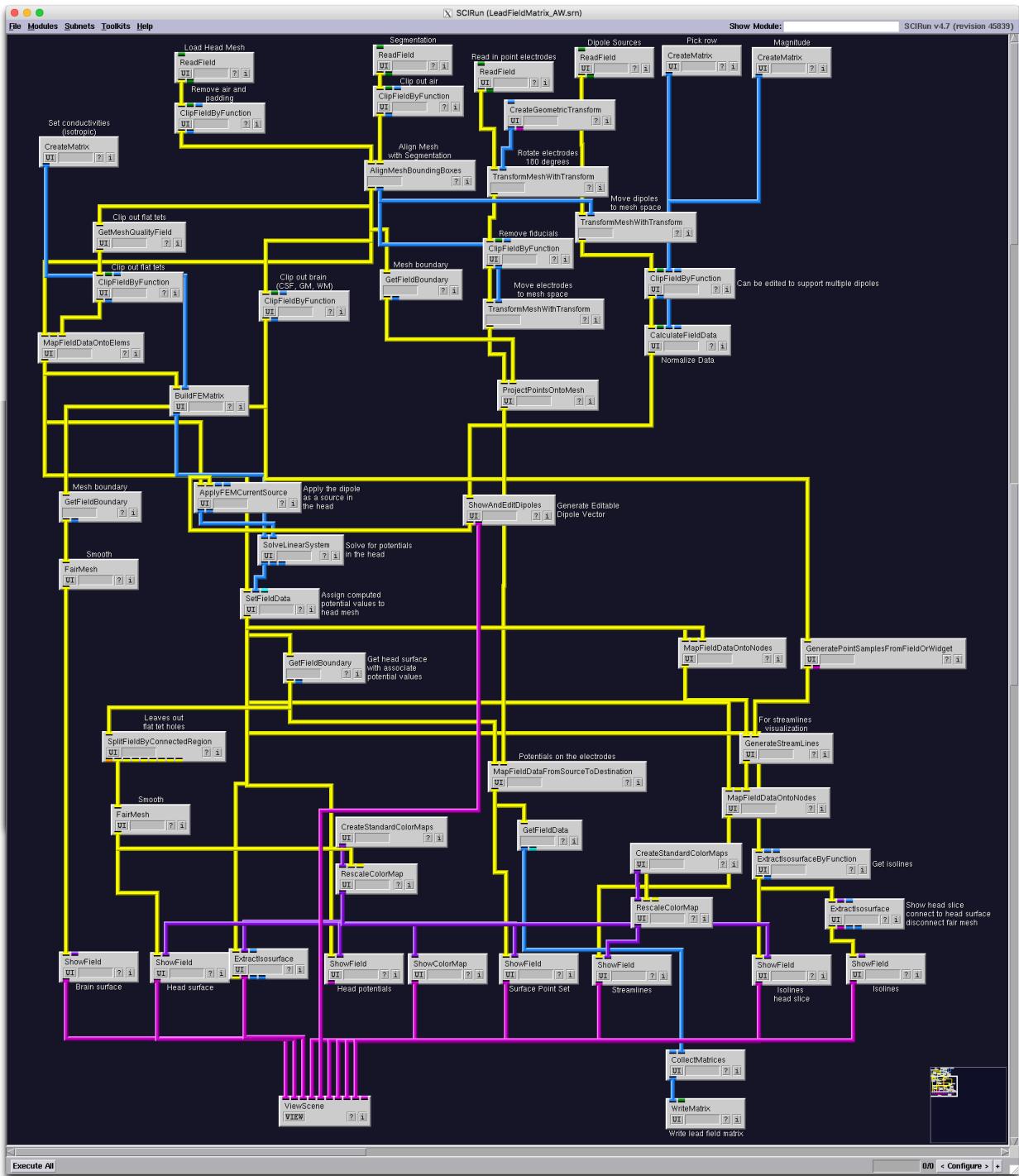


Figure 14: default

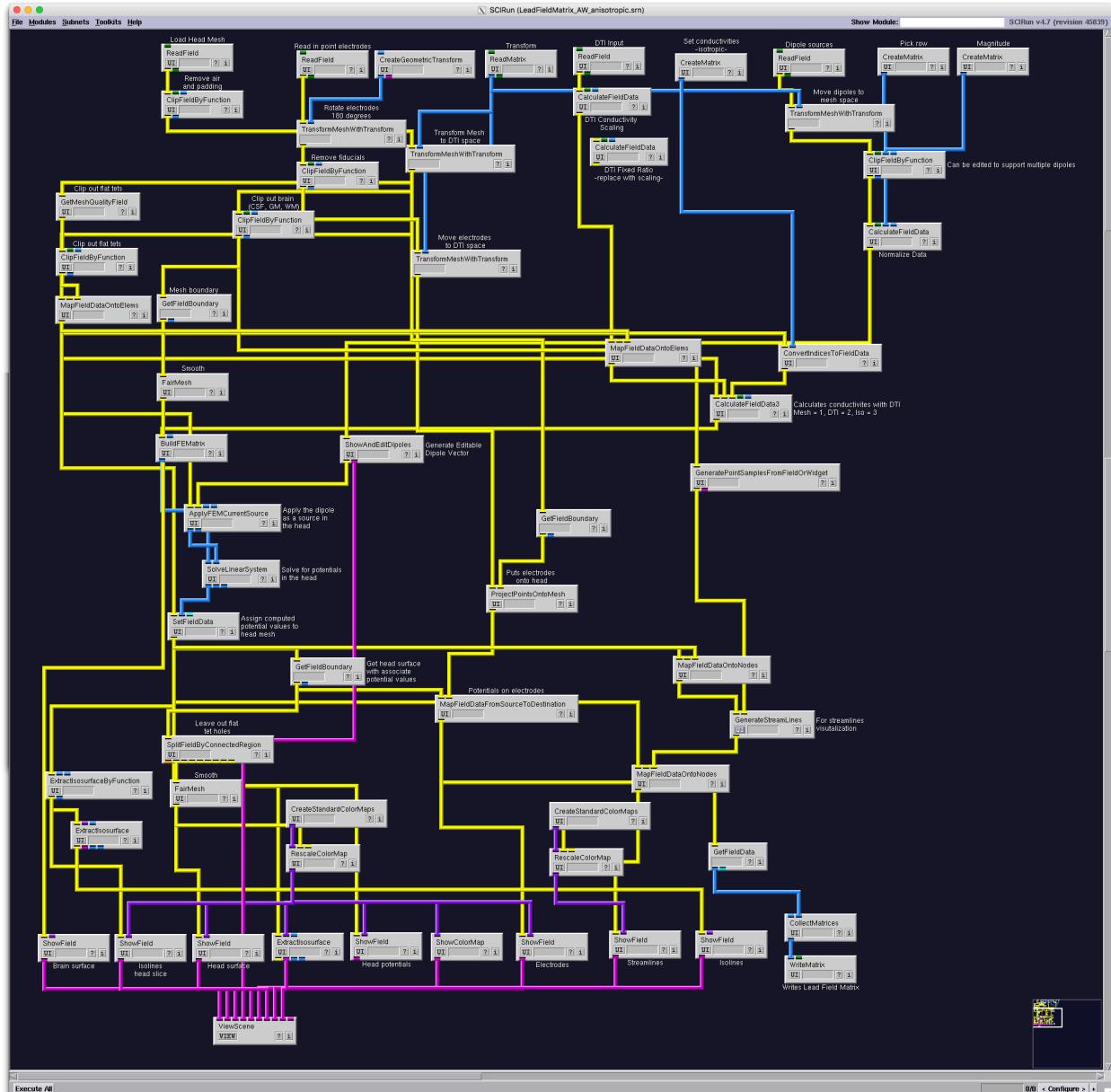


Figure 15: default

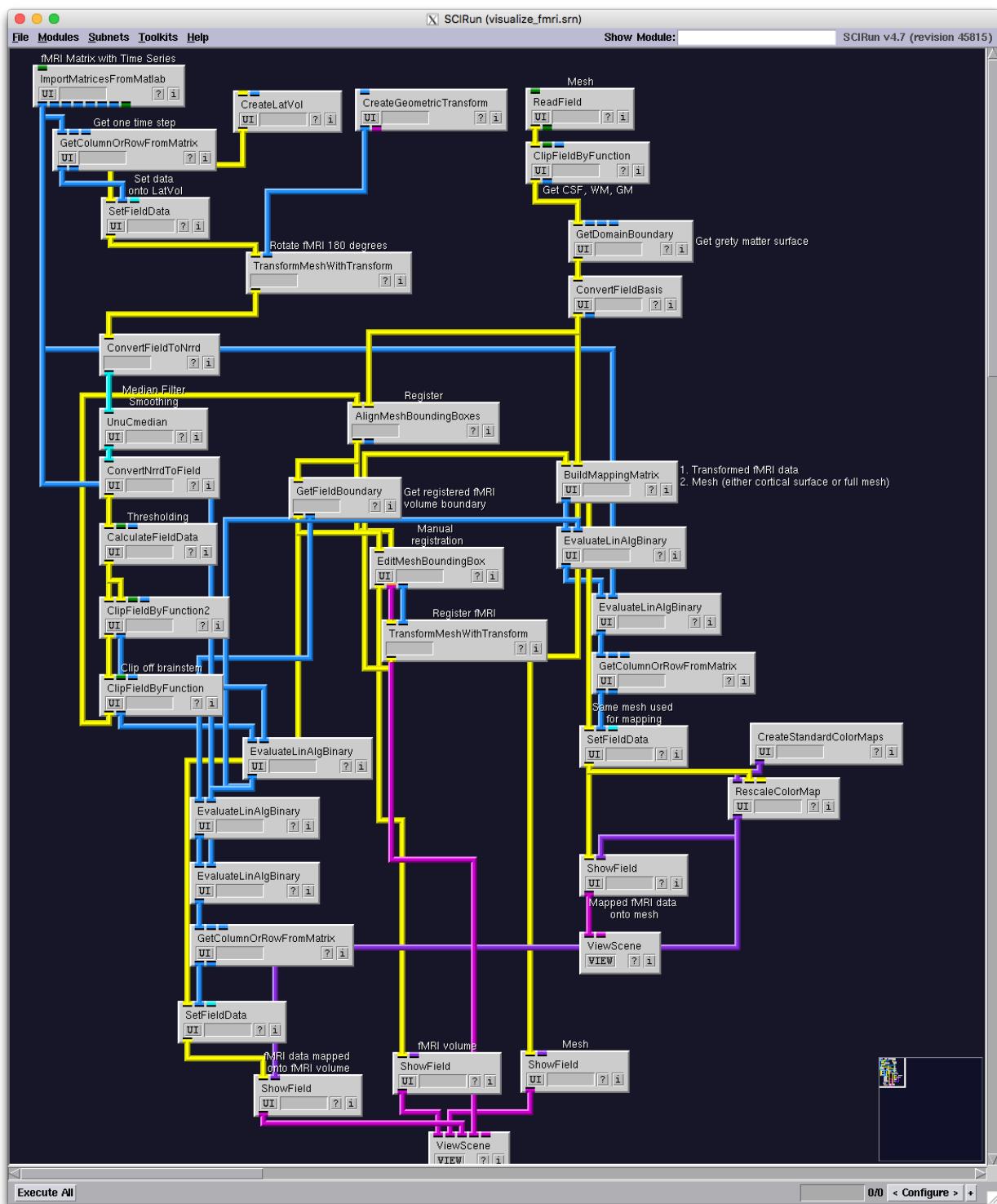


Figure 16: default

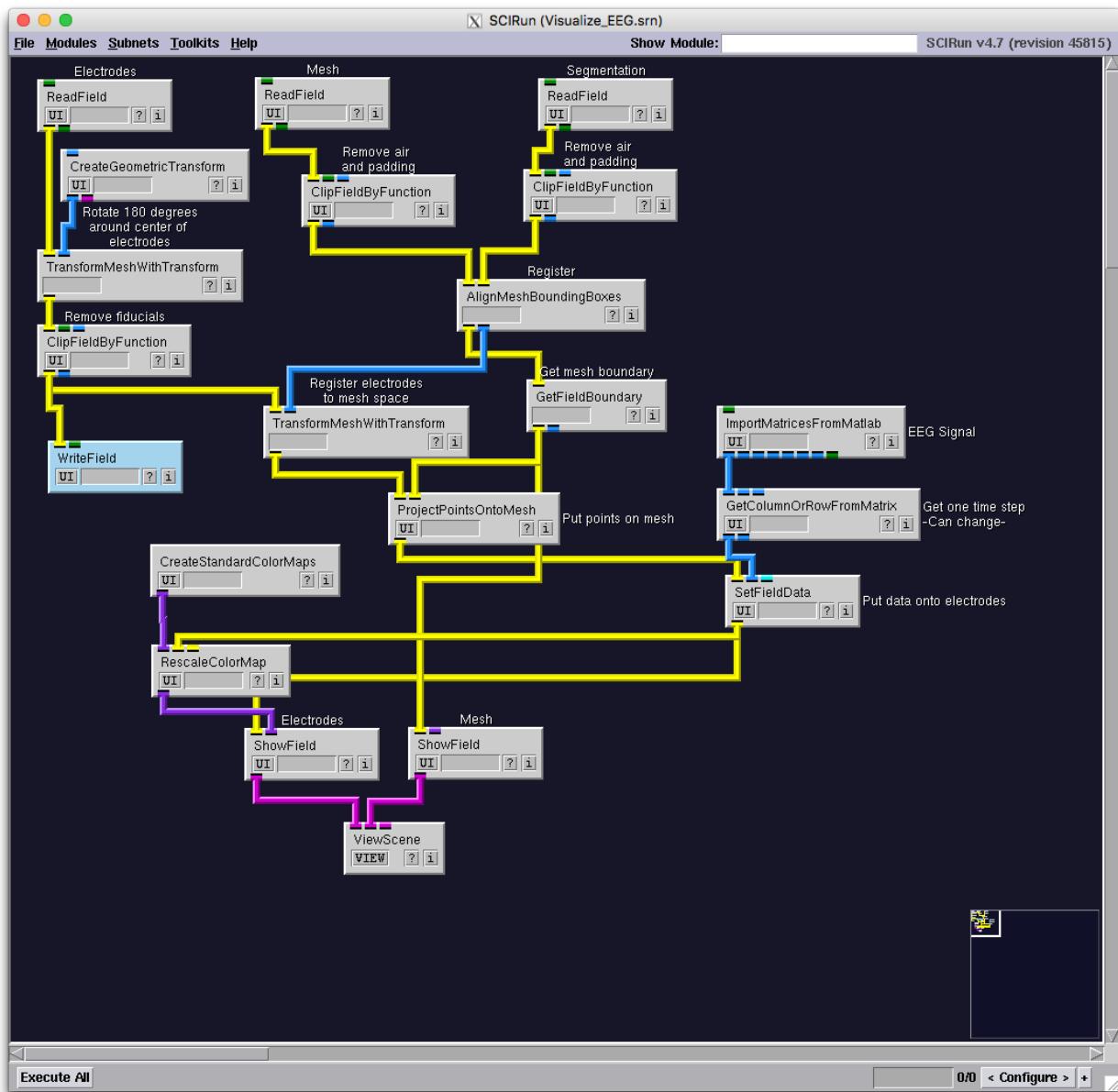


Figure 17: default

### 2.6.3 EEG

We visualized the EEG data on the physical electrode locations. These were registered to the mesh space with a rigid registration after the fiducials were removed from the electrode data set. The filtered and cut EEG matrix was used as input and each time step was represented as a column. The electrodes were then placed onto the mesh, and EEG data mapped onto the electrodes one time step at a time.

### 3 Results

All SCIRun networks used to generate results will be included in the open-source data set for future use and editing.

#### 3.1 Segmentation

In this project a full, detailed head is segmented into different tissue layers to be able to make an inhomogeneous 3D mesh.

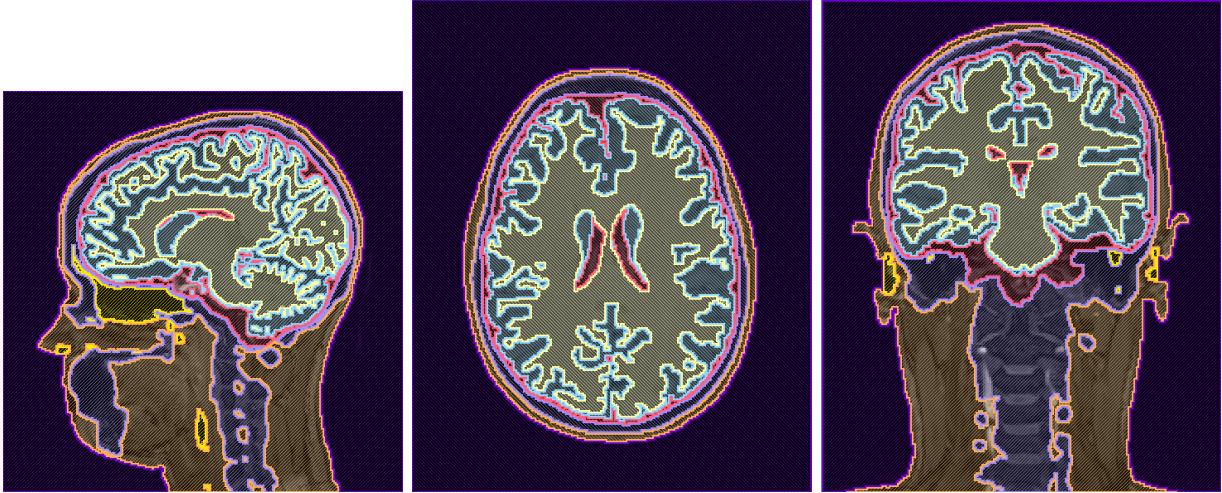


Figure 18: default

Since there was no computed tomography (CT) scan with this data set, the task of segmenting the skull and especially the sinus layers seemed daunting. A basic skull and bone layer was created using FSL skull stripping feature in the brain extraction tool (ref?) and then concatenated it with a rough bone segmentation using thresholding in Seg3D of black pixels and manually correcting the segmentation. Although a skull and bone segmentation was pieced together, the sinus segmentation seemed extremely difficult and time consuming. After obtaining the pseudo-CT scan, this did not seem like such a large feat anymore. The skull/bone and sinus segmentations made from it fit the brain segmentation extremely well and had no artifacts from combining two different layers together. Although the teeth are solid bone from the subject's permanent retainer.

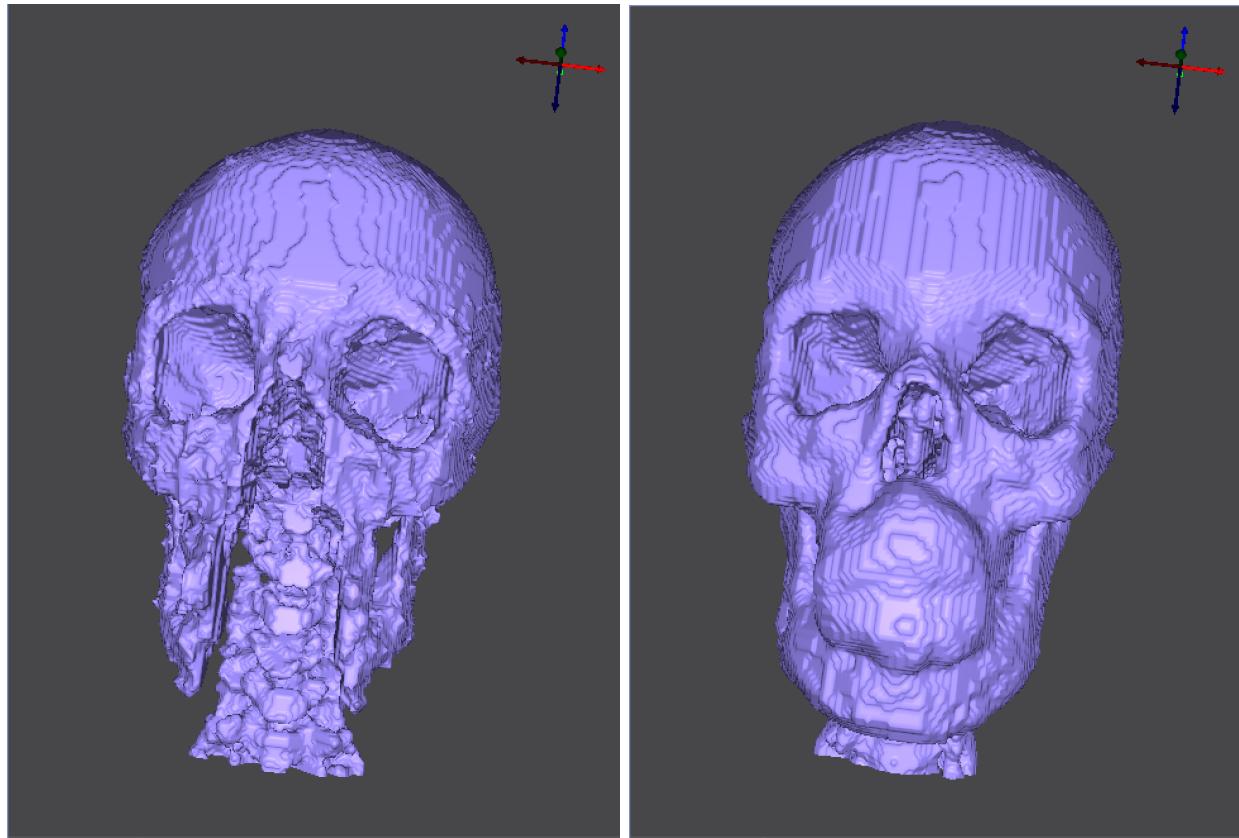


Figure 19: default

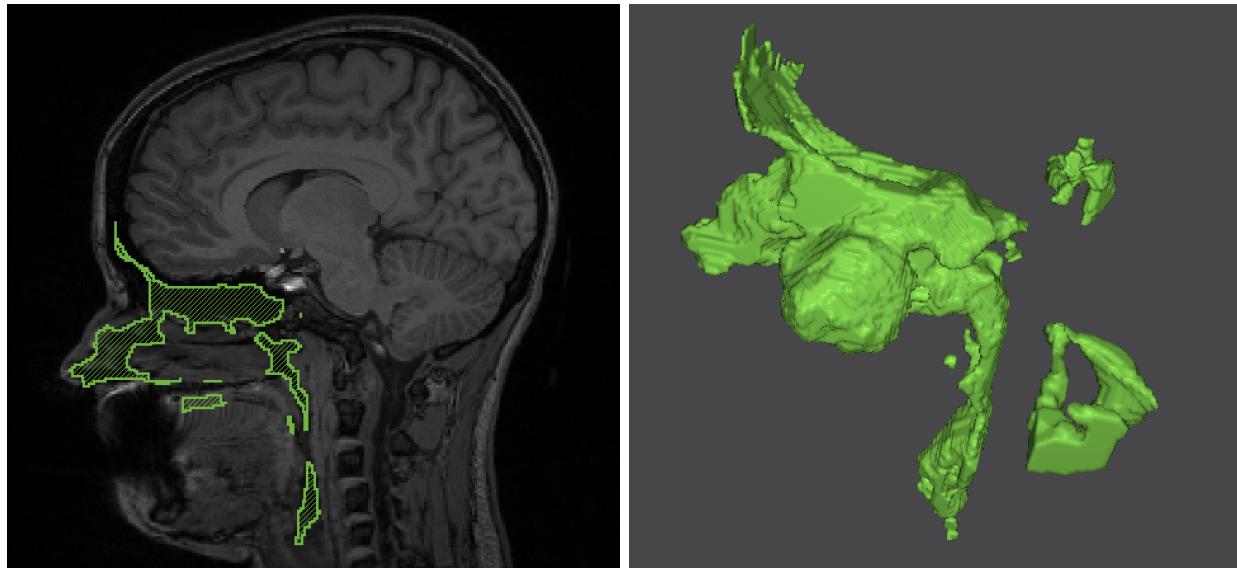


Figure 20: default

When imaging, the subject is laying on their back which shifts the brain to the back of the head resulting in thin segmented layers in the back of the head. There were also thin layers on the side of the subjects head, the bridge of the nose and the bottom of the chin. These thin layers were made to be at least two pixels thick manually to ensure that a mesh could be made without holes.



Figure 21: default

## 3.2 Finite Element Meshes

The highest resolution mesh, which was made with settings in Section 2.4, has 60.2 million elements and 10.3 million nodes. This mesh was so large due to the complexity of the segmentation. Simulations run very slow when using this mesh due to its size, and require at least 32GB of RAM in order to not crash SCIRun.

The goal to make smaller meshes was made in order to be able to run simulations more efficiently. After manually changing the sizing field as described in Section 2.4, a mesh was produced with 15.7 million elements and 2.7 million nodes with no holes. However, this mesh does contain one flat tetrahedra. It is later removed in a SCIRun network, and is currently being investigated by Cleaver software developers.

## 3.3 Forward Problem

### 3.3.1 Isotropic

An isotropic, inhomogeneous head model would be expected to have fairly spherical results in propagation of signal. 3-dimensional streamlines were generated to show this. Isolines are another way to view this, but in each slice as to better show the differences between isotropic and anisotropic conductivity. Here we can see that the electrodes and dipoles are registered well to the mesh. (figures of each)

### 3.3.2 Anisotropic

An anisotropic, inhomogeneous head model would be expected to not have spherical results. This can be seen with streamlines, and better so with isolines. As discussed in 2.5.1, there are two methods of scaling diffusion tensor data. The network has the option to choose one or the other. We can see that the electrodes, the dipoles, and the mesh registered well to the diffusion tensor space. (figures)

## 3.4 fMRI Visualization

With the rigid and manual registration of the fMRI data to the mesh space, fMRI data was mapped properly onto the mesh (for the first time). This allows for future use of fMRI using the SCIRun package.

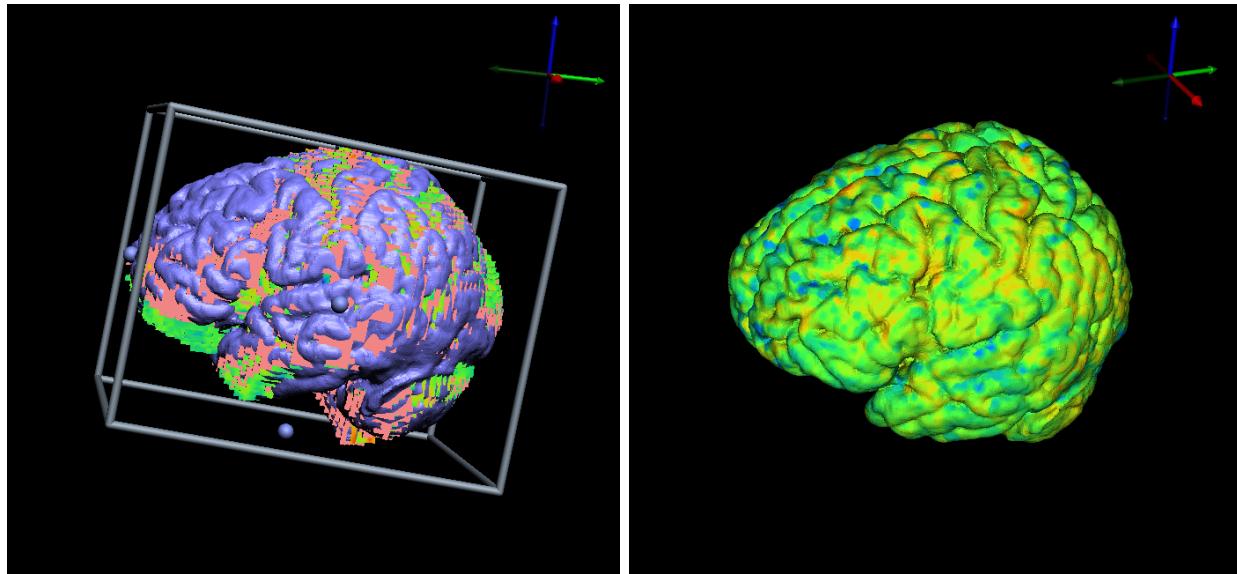


Figure 22: default

### 3.5 EEG Visualization

When using EEG data, the particular application dictates what kind of additional processing, filtering, and cutting of the data needs to be done. In these visualizations it can be seen with this basic preprocessing there are bad leads, specifically around the eyes which could be due to blinking or eyes rolling, but will be improve with further specific processing.

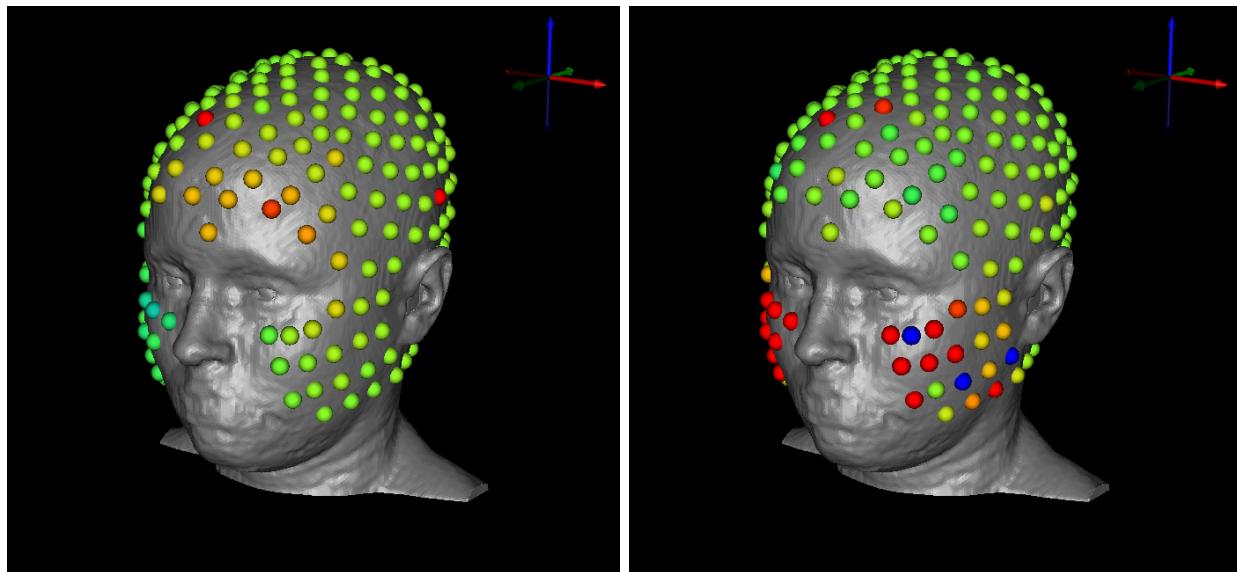


Figure 23: default

## 4 Conclusion

In this paper we have outlined a comprehensive pipeline to build an inhomogeneous, anisotropic head and brain model based on human data of all image modalities for use in electroencephalography with an emphasis in forward and inverse problem research as well as visualizations of function MRI data and EEG data. Along with the pipeline, the human data will be released as open-source to enable other scientists to have a starting point to continue further research. Building a model can be time consuming and hinder further important research, and this model will allow scientists to have a straight-forward path to building their own model and/or using the model that I have built because it is a high-resolution, multi-image data set.

Further investigations from building this pipeline include finding better decimation algorithms for 3D tetrahedral finite element meshes. Due to the number of materials included in the mesh, current decimation algorithms have not been able to further simplify the mesh. More exact sinus and skull segmentation methods would improve the appearance and accuracy of the layers. More exact registration techniques that will provide a better transformation matrix for moving images to DTI space, especially fMRI data. More inclusive methods for functional MRI data into a source localization head model. More specific processing of EEG data for different applications which will make for better visualization.

## 5 Acknowledgements

Imaging and electroencephalography for this project was done at Electrical Geodesics Inc. at the University of Oregon. This work could not have been completed without the help of Jess Tate and Brett Burton. This project was supported by the National Institute of General Medical Sciences of the National Institutes of Health under grant number P41 GM103545-18.