GridLoc Documentation

M. Leibbrand and M.P. Branco

M.Leibbrand@umcutrecht.nl and m.pedrosobranco@umcutrecht.nl

25th May 2018

1. The GridLoc Toolbox

A toolbox was constructed in MATLAB® (tested in Linux, Windows and Mac) for the localization of high density (HD) electrode grids. The *GridLoc Toolbox* allows the user to set custom grid specifications and will give a predicted location based on an activity model constructed by cortical depth and presence of blood vessels within a certain Region-of-Interest (ROI). The inputs and the detailed functionality of this toolbox will be described below. Use the main script "Main_Script.m" to run the toolbox (see details in section 2).

1.1 Unpack the code in a directory

Add folder to Matlab path. Be sure to also have SPM12 in your path.

1.2 The input

The input to the *GridLoc* toolbox consists of the subject information presented in a structure format with multiple fields and variables. The structure is called *subj_info* and, per subject, contains:

1) The location of the file that contains the gamma power of the resting state data (min 3 min, in *.mat, [power x time]);
2) The channels of interest within the data file that represent the HD grid;
3) The bad channels (taken from clinical logs or calculated with an in-house function);
4) The shape and dimensions of the electrode grid that is being localized and the dimensions of the ROI, which determines the cortical search area.
5) The surface model of the subject, which is a triangular mesh. The surface model, and supplementary files, are constructed from the free surfer representation of the cortex (*.nii);
6) Several settings related to the neuralAct *viewBrain* function.
7) The hemisphere of the subject where the grid resides (**l**eft or **r**ight);
8) The side of the grid where the cables reside (**u**p, **d**own, **l**eft or **r**ight);

9) The location of a surface file (balloon) of the cortex of the subject and the location of the angiogram T-file (*.nii).

A detailed description of the contents and specifications of the *subj_info* structure and additional details about input variables can be found in **Table 1**. Also, in order to index the electrodes correctly a function has been added to allow to enter the layout of the electrode grids (see **Figure 1b**).

Table 1. The fields within the *subj_info* structure that is used by the *GridLoc Toolbox*.

Subj_info.gamma_mean	File that contains the gamma power of the resting state data.
Subj_info.chansel	Channel selection regarding the channels of interest.
Subj_info.bad_channels	The bad channels of the grid, calculated beforehand with an in-house function or taken from the clinical log.
Subj_info.dims	Dimensions of the electrode grid in the X and Y-directions, formatted as (X,Y). X-axis of the electrode grid is where the cables reside (see **figure 1a**).
Subj_info.intElec	Inter-electrode distance in mm formatted as (X,Y) (see **figure 1a**).
Subj_info.ROIsize	The ROI coverage formatted as (X,Y). Around every ROI point (center-point) a grid is constructed and between every ROI point there is 1 mm spacing. Requires a minimum of [3,3], which gives an approximate 9 square mm cortex coverage. Increasing these values exponentially increases the time the script takes to run.
Subj_info.stepSize	The step size between translations define in the ROI size. For example translate the grid search space of 3x3 mm with step size of 0.5 mm between center-points.
Subj_info.neuralAct	Location of the file that contains the cortical surface model and supplementary models needed for the NeuralAct toolbox such as the hull cortex and a coarser model.
Subj_info.hemiVect.hemi	The hemisphere of interest (‘l’ or ‘r’).
Subj_info.hemiVect.side	This field represents the side (‘l’,’r’,’u’ or ‘d’) of the grid

	where the leads are located after they have been inserted under the cortex by the surgeon. For the left hemisphere, this is left (anterior), right (posterior), up (dorsal) or down (ventral) respectively (see **figure 1a**). For the right hemisphere, the left side is posterior and right side is anterior.
Subj_info.sfile	Location of the cortex surface file.
Subj_info.tfile	Location of the angiogram file.
Subj_info.Tthreshold	Threshold for the angiogram file with vessel information. I based upon a T-statistic.
Subj_info.VoxelDepth	Search depth for co-registration of blood vessels to surface

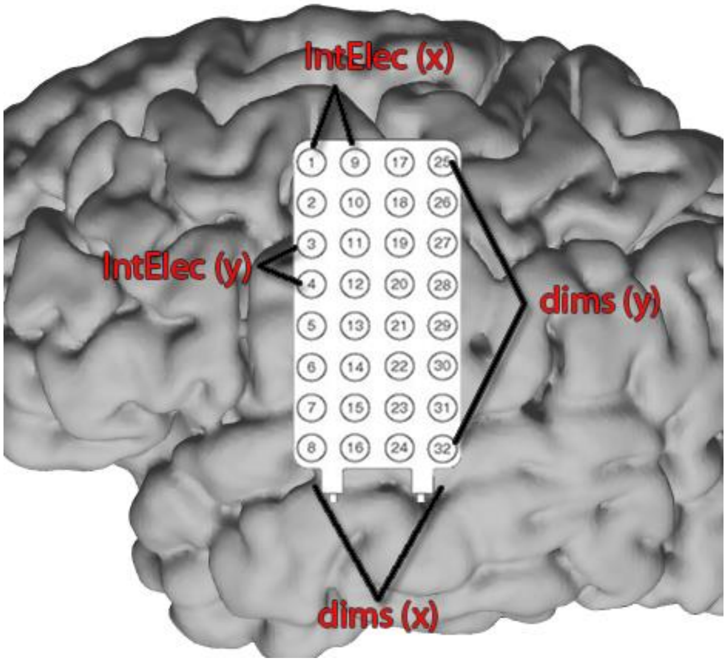
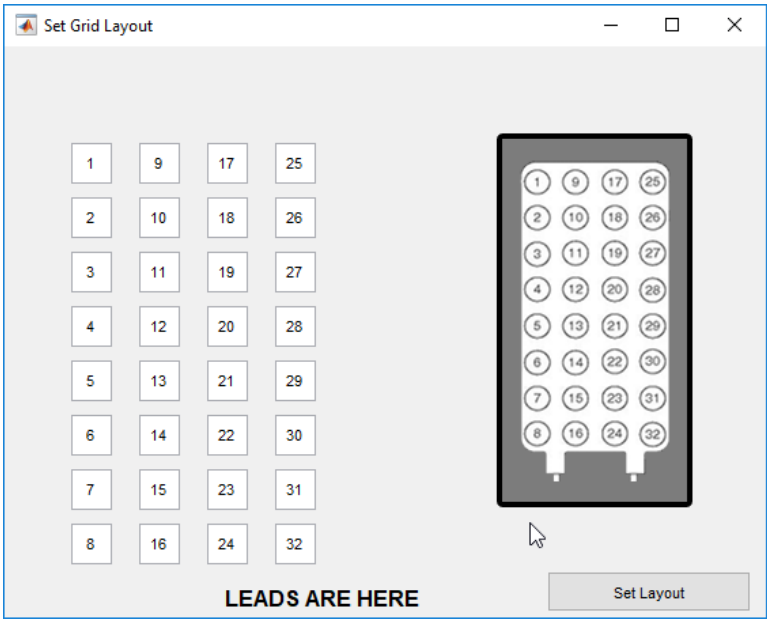


Figure 1a Grid representation.
The figure above elaborates which settings to use in this hypothetical situation. Seeing as the leads are located at the bottom, *subj_info.side* is set to ‘d’. *Subj_info.dims* is set to [4,8] to account for the amount of electrodes on the grid (X,Y), horizontally and vertically, respectively. From the technical specifications of the grid we found that the inter-electrode spacing is 3 mm in both directions, therefore setting *subj_info.intElec* to [3,3].



1	9	17	25
2	10	18	26
3	11	19	27
4	12	20	28
5	13	21	29
6	14	22	30
7	15	23	31
8	16	24	32

Figure 2b Grid layout.
The figure above shows how to set the grid layout by using the function *gridLayout*. The user starts entering the numbers of the electrodes from the top left corner. It is important to note that the electrode leads must be located on the bottom, with the recording electrodes faced downwards (to the brain), in order to index the grids correctly.

1.3 The search-space: Region-of-Interest

The next aspect of the pipeline is to define the search-space, i.e., the cortical region-of-interest (ROI) where the model will search for the position that best predicts the measured frequency band activity. To set the ROI, a graphical user interface (GUI) is launched by the *getROI* function (see **Figure 2**). This GUI enables the user to select the middle point of the ROI on the cortex model of the subject. After clicking continue, a red circle will be displayed which estimates the cortical coverage of the ROI (**Figure 2**). Additionally, the side of the grid containing the leads (set in the *subj_info* structure) will be displayed in yellow. The output to this function is a variable with ROI points (X,Y,Z) spaced, for example, 1 mm apart, which are fixed in a 2D-plane that is perpendicular to the selected middle point of the ROI (to define other steps use subj_info.stepSize). The dimensions of the ROI are set in the *subj_info.ROIsize* variable.

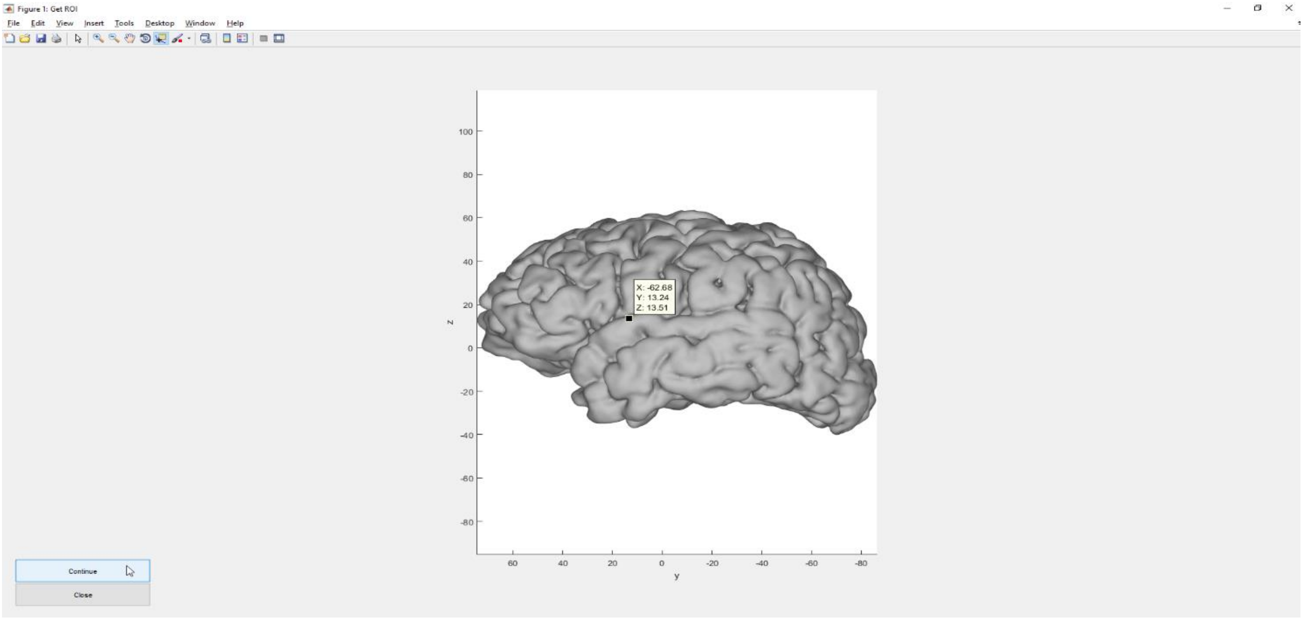
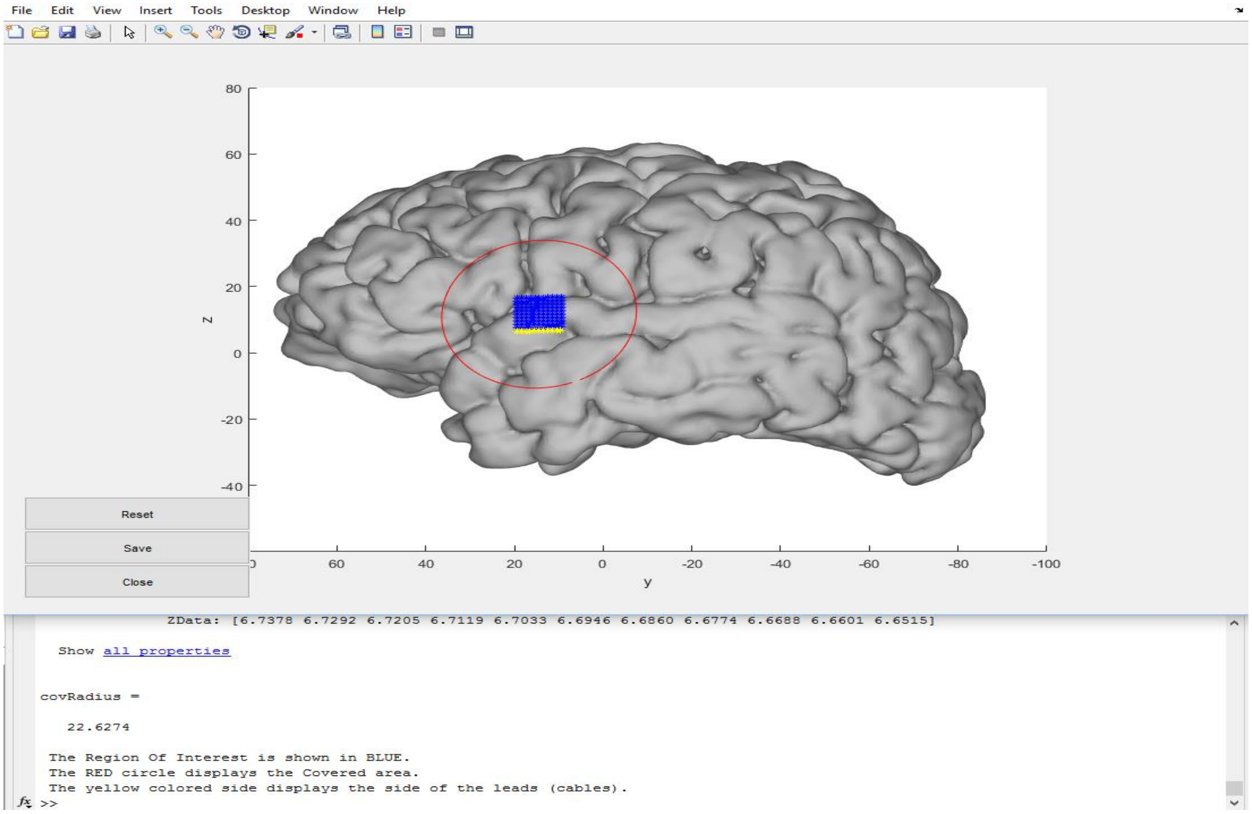


Figure 3a. **The *getROI* Graphical User Interface**
The user gets to select the middle point of the ROI by clicking on the cortical surface model. After selecting a point, they click on continue (see **fig. 2b**).



```
           ZData: [6.7378 6.7292 6.7205 6.7119 6.7033 6.6946 6.6860 6.6774 6.6688 6.6601 6.6515]

  Show all properties


covRadius =

   22.6274

 The Region Of Interest is shown in BLUE.
 The RED circle displays the Covered area.
 The yellow colored side displays the side of the leads (cables).
>>
```

Figure 4b. **The *getROI* Graphical User Interface**
The user gets a preview of the estimated covered cortical area which is displayed by the red circle. The blue square indicates the ROI points and the yellow side indicates the side of the leads. Press save to save the current ROI.

1.4 Translation, rotation and projection of the grids

The next part of the script constructs grids around every ROI point calculated by the *getROI* function to model the predicted gamma band activation for that specific location. However, one of the factors that the toolbox must also account for is that an electrode grid can rotate when the neurosurgeon slides it underneath the skull during the intraoperative procedure. To solve this problem, every grid is rotated over a 90º angle with one-degree increments to account for a change of orientation. This means that the grids will rotate from -45º to +45º with steps of 1º. This adds up to 90 possible grid orientations, and thus locations, for a grid constructed from a single ROI point (**Figure 3**).

The toolbox constructs grids around the ROI points by applying a difference matrix with dimensions set in the *subj_info* structure. The ROI points and the grids are all constructed by use of the *meshgrid* function from Matlab, which creates a mesh with specified dimensions. This function is limited to specifying a set of points in a 2D plane. This means that every ROI point and electrode that has a fixed 3rd dimension needs to be projected onto the 3D cortical surface model. This is achieved via the *NeuralAct Toolbox* (Kubanek & Schalk, 2015). This toolbox projects a point onto a convex hull (*hullcortex*) surface model by finding the intersection of a

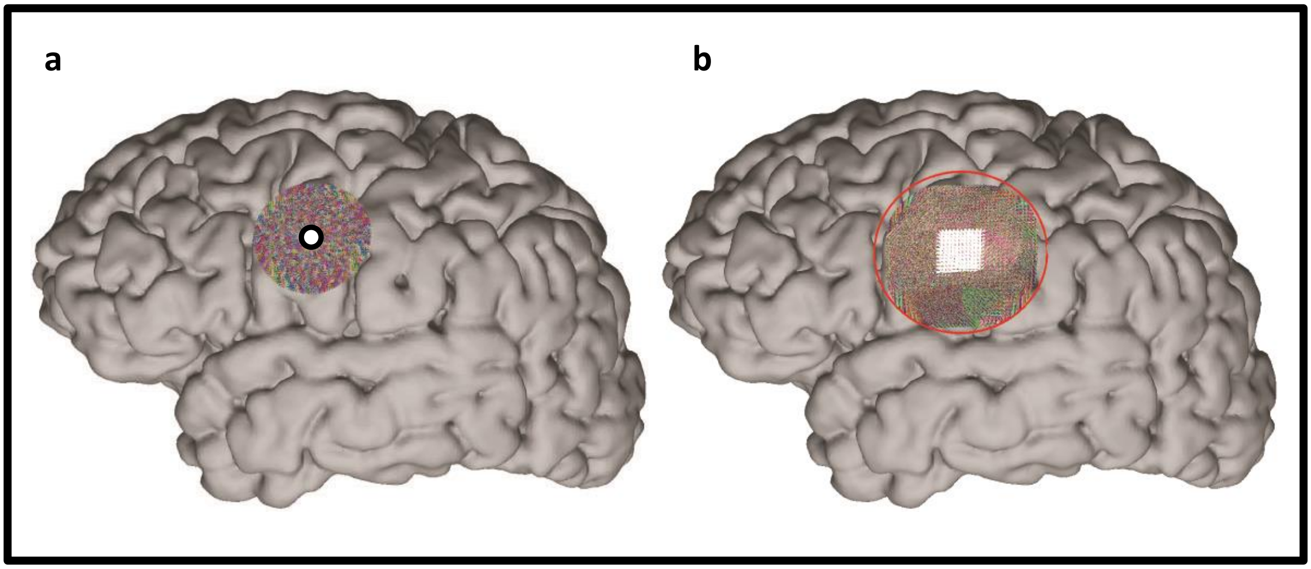


Figure 3a,b. Translation and rotation of electrode grids
A grid is constructed around a single ROI point (white dot; see **a**) and rotated by 1º increments over a 90º angle. This gives a near circular cortical coverage per ROI point for squared electrode grids. The actual cortical coverage for all ROI points is estimated by the red circle in the *getROI* function (see **b**). The square in the middle of the circle represents the ROI points calculated via the *getROI* function.

vector calculated from the average normal vectors of vertices within a certain diameter (25 mm). However, the *NeuralAct Toolbox* can introduce projection errors for points located too far from the surface model. This poses a problem for the *GridLoc Toolbox b*ecause the points constructed from a mesh are all fixed in a 2D plane. To counter this problem, the mesh grids have to be rotated into the tangent plane to their middle point before projecting the points onto the surface. Moving the points into a tangent plane of the middle point of the grid ensures that the average distance from the points to the cortical surface is at at a minimum and is an attempt to minimize the projection error introduced by using the *NeuralAct Toolbox.* The tangent plane is calculated in the function *calcTangent* and uses a plane equation that calculates the tangent plane from a point (the middle point of the mesh) and a normal (the normal calculated via the *NeuralAct Toolbox*). The *calcTangent* function then finds the angle between the 2D plane and the tangent plane and uses a rotation matrix to rotate all the points from the 2D mesh into the tangent plane.

1.6 The output

The output of this function consists of a figure with the predicted location of the grid (Figure 4), together with four image scale matrices at the bottom. These matrices represent the normalized mean gamma response of the measured signal, the normalized angiogram and depth model for the predicted location and the combined weighted values of the angiogram and the depth model. Both models are weighted equally before their combined values are correlated to the measured gamma power to find the predicted location. The predicted correlation is shown in the command window, as well as it being set in the variable *corrPred*. The values of the individual matrices can be found in the *AngioSq*, *DepthSq, CombinedSq* and *GammaSq* variables.

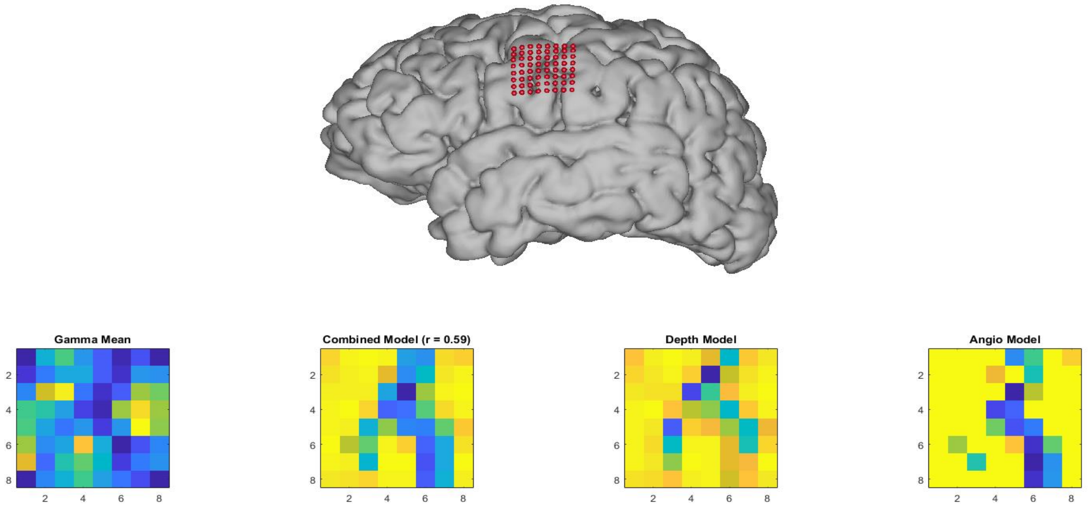


Figure 4. The output figure.
This figure shows the output of the *GridLoc Toolbox*. It shows the predicted location of the electrodes in red and shows the image scale matrices of the different models at the bottom. The correlation between the normalized gamma mean and the combined model is also shown in the title of the combined model image scale matrix.

2. Run the main_script.m

Section 1 – create the subject info structure and set location of input files:

1. Check the fields in subj_info described in Table 1.
2. Create NeuralAct display options ('viewstruct.lighpos’, ‘what2view’, ‘transp’, ‘colix’ and ‘viewvect’).
3. Import freesurfer cortex file in *.nii format (convert the ribbon.mgz into *.nii). The function gen_cortex_click_from_freesurfer asks to input the *nii file, also referred to as t1_class.nii.
4. Create and save the cortex models.

Section 2 - process the angiogram files:

1. Add path to the sfile (*.img/*.hdr or *nii), that is the surface balloon of the brain, to which the vessels are projected. This balloon (see Figure 5) can be, for instance,

computed using the function get_mask_from_Freesurfer.m inside the folder ‘Adapted Hermes2010’:

```
get_mask_from_FreeSurfer(‘name’,… % subject name
                                   './data/FreeSurfer/t1_class.nii',... % freesurfer class file
                                   './folder_name/',... % where you want to safe the file
                                   hemi,... % 'l' for left 'r' for right
                                   11,0.3); % settings for smoothing and threshold
```

2. Add path to the tfile, that is the angiogram file (*.img/*.hdr or *nii) masked to remove the skull (see Figure 5). You can use SPM12 for this procedure by defining a brainmask and image calculator.
3. Define the thresholds for the angiogram, such that you see only the values (voxels) representing the vessels of interest.
4. Define the voxel depth for projection of the vessels into the surface (e.g., 8).
5. Compute and save the angiomap.

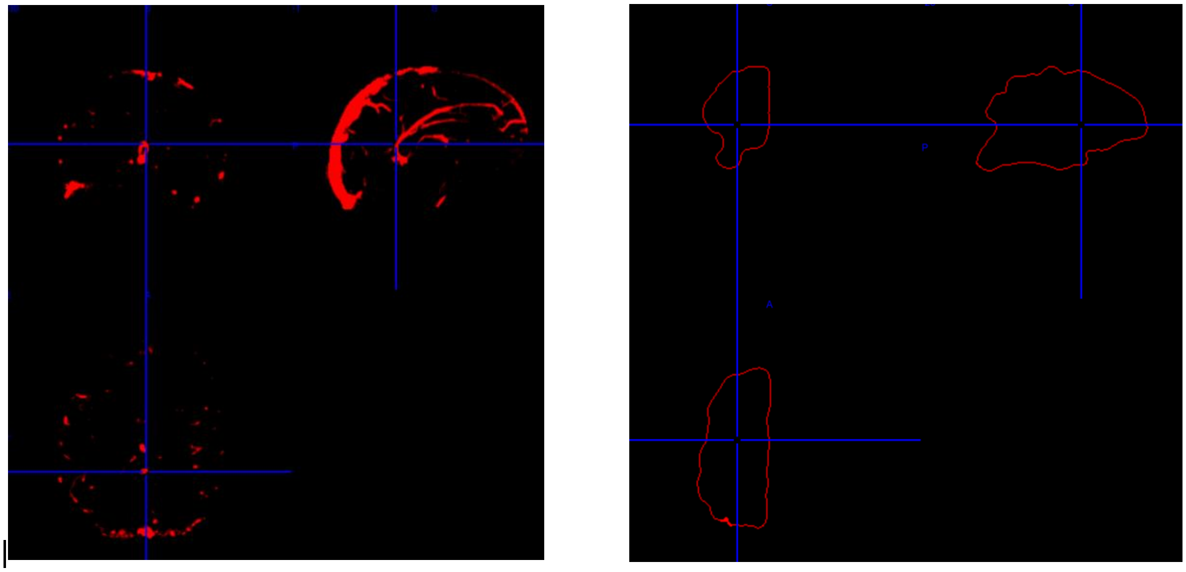


Figure 5 – sfile and tfile.
Left) tfile: angiogram masked to remove everything outside the brain (skull stripped) and thresholded for visualization of the main (target) vessels. Right) Cortical surface balloon extracted from the freesurfer t1_class.nii file.

Section 3 - Calculate the ROI and grid properties:

1. Load the previously saved files.
2. Run this section to define the region-of-interest (ROI) for the search-space (see section 1.3 for more details about the interface).

Section 4 - Project ROI and grids into brain and calculate the model:

1. Run this section and wait till it is completed.

Section 5 - Run model correlation to HFB spatial pattern and plot results:

1. Run this section to compute the correlation between the model and the HFB patterns.

Section 6 - Display electrode prediction:

1. Run this section to plot the predicted electrode position (see Figure 4).

Section 7 - Display the attenuation patterns:

1. Run this section to plot the attenuation patterns (see Figure 4).