NeuroMeasure v4.4



User Manual

Michael Gerber, Alasdair Mclean, Samuel Stephen, Alex Chalco, Usman Arshad

Summary

NeuroMeasure is an open source interactive software application for the analysis of motor map data collected from the Nexstim and BrainSight neuronavigation platforms. It is designed to guide the user through a standardized workflow for analyzing neuronavigated Transcranial Magnetic Stimulation (TMS) induced Motor Evoked Potentials (MEP) recorded from a target muscle while stimulating over the the motor cortex. The TMS coil is typically positioned and fired serially across the motor cortex sampling at regular spatial intervals to generate a “motor map” of MEP response in which spatial position denotes the peak of the induced magnetic field, and MEP can have a number of defined characteristics including peak-to-peak amplitude of paired electromyography (EMG) recordings measured in units of voltage, or signal latency measured in units of milliseconds. This application is a platform for generating characteristic measurements of a motor map, such as center of gravity (COG) position, peak value position, surface area and volume integral and comparing those characteristics spatially and temporally within a subject.

Background

## Purpose

Transcranial Magnetic Stimulation (TMS) is a non-invasive means of inducing ionic current flow in neurons with magnetic waves. This method has recently become popular as a tool in characterizing the motor cortex. TMS applied over this region creates a response in a target muscle known as a motor evoked potential (MEP) and is typically measured by surface electromyography (EMG). A collection of MEP values collected in this manner can be processed to produce a heat map. Mapping motor cortex is of interest in Neurosurgery for clinical decision making, as well as in studies of motor cortex plasticity in health and disease.

TMS hardware has evolved significantly since its debut to make it the versatile and robust stimulation mode today, with notable advancements in magnetic coil type, magnetic field strength, resolution and stimulation technique (repetitive TMS, paired pulse TMS, etc.). With neuronavigation, it has become possible to track position (and direction) of the magnetic field distribution registered to the patients MRI and anatomical landmarks, to within several millimeters. However, despite these advances, there is no comprehensive software to use this data to generate meaningful characteristic measurements of the results of these mapping procedures. Thus, the purpose of this design project is to create such a software platform, offering researchers and clinicians an easy to use workflow for characterizing and comparing motor maps generated with neuronavigated TMS.

## Intrinsic Data Instability

Motor evoked potential (MEP) amplitudes are highly variable due to the fluctuations in motor cortex excitability, which is normal in the brain. The excitability of cortical and spinal neurons, or generally their tendency to depolarize and initiate an action potential, depends on the magnitude of the graded potentials traveling within the neurons of interest and the timing of their appearances. Graded potentials can either initiate action potentials (suprathreshold potential), excite the interior of the neuron without producing an action potential (subthreshold potential) or depress the neuron interior and make depolarization more difficult (inhibitory potential). Coupled with the refractory period intrinsic to neurons, the summation of these graded potentials in both time and space leads to fluctuations in excitability across the cortex (Kiers et al. 1993, Thickbroom et al. 1998). Microscale fluctuations of individual neurons sum to macroscale fluctuations of the entire motor threshold of the cortex, leading to graded variance of the MEPs themselves even when repeatedly measured in rapid succession (Mills et al. 1996). TMS itself may be a factor in this variation of excitability thresholds among the neural population or it may be wholly innate to the tissue or both, thus making motor maps created with MEP amplitudes difficult to accurately reproduce. Methods employed to reduce MEP variance across subjects include: having subjects partially contract the target muscle, incorporating MEP latency into maps or alternate forms of TMS (repetitive TMS, theta burst stimulation, paired-pulse TMS) (Darling et al. 2006, Fitzgerald et al. 2006, Kallioniemi et al. 2015).

Despite the difficulty in reproduction, measurements of reproducibility in TMS-generated MEP motor maps showed a variance of 1-2 mm in center of gravity over 30 repetitions in a sample size of 5 healthy subjects, which may be acceptable in most circumstances (Thickbroom et al. 1998). TMS motor mapping has been useful in glioma resection neurosurgery as a means of locating the boundary of a glioma within 1-2 mm of precision (Kreig et al. 2012, Saito et. al. 2015) or for studying neural plasticity in human subjects (Pascual-Leone et al. 1999) so even without correcting variance of MEP amplitudes, current motor maps are still viable for a broad range of usage in clinical and research fields.

Still, if one were to consider possible post-acquisition processing solutions the first step would be to rectify the type of data that MEPs are: continuous or categorical. MEPs can be viewed as the sum of a host of neurons firing in response to a TMS pulse, and the peak-to-peak amplitude of the response would correspond to the number of activated corticospinal neurons. Since the activation thresholds of these neurons fluctuate by small and variable amounts between pulses, MEP amplitudes can be considered continuous data (in this model a small MEP and a large MEP are distinct events). Alternatively, MEPs can be viewed from the context of the motor threshold as binary categorical in the sense that the muscle either contracts if enough neurons activate to elicit even the smallest response, or it does not (in this model a small MEP and a large MEP are the same event as long as both are above the threshold). When taken as continuous, multiple readings taken from the same location can be averaged together, or the greatest of the repeated values can be used for the mapping. When taken as categorical, a site can be considered to be either above a chosen threshold if at least half or more stimulations lead to an MEP response, or below threshold otherwise (Mills et al. 1996). The resulting map from categorical data would be probabilistic as opposed to a regression.

One way to account for the natural variance in MEP amplitude is to average the responses across the set of stimuli. If the data is clustered in some fashion between simulations, then a time-averaged MEP amplitude would yield a useful motor map. However, the exact probability distribution of the MEP values over time is currently unknown and the mean only provides the most basic information of the shape of the distribution. Variance of MEPs may not be well modeled with a normal distribution, so the expected value of the amplitudes would not catch the physiologically relevant amplitudes detected per trial of TMS. For example, if the MEP values for each trial vacillated between a thresholded minimum and maximum, the average MEP would take an intermediate constant that would not provide relevant information on the gross excitability of the region of the MEP origin. This precedent, while extreme, illustrates the restrictions of a time-averaging method on data with a vaguely defined probability distribution and density.

Another method for resolving the data instability, as discussed in relevant literature, is to take a repeated measure approach for determining a hotspot within a set area. As further explained by Arya et al., one could stimulate a region of the brain 20 times and observing an MEP amplitude above threshold 10 times would define that area as a hotspot. Iterated TMS pulses would yield the full variability of the MEP amplitudes; with enough cycles and sufficient inter-pulse interval so as to avoid cumulative effects that might modulate amplitude, a trend could be observed and a variance can be detected. That said variance could then be circumvented by producing a map by defining hotspots based on the number of responses evoked. The brain is dynamic and MEP maps can shift space because of the change in excitability of the underlying neurons, but this thresholding through repeated measures process can still detect the hotspots themselves and help construct physiologically relevant maps.

## State of the Art

The neuroscience field has developed the practice of sampling the motor cortex and representing motor evoked potentials over a gridspace where the TMS pulses are applied (Littmann et al. 2014, Kleim et al. 2007). Comparisons are then made between maps by directly comparing the values of likewise points on the grid. While effective, the standardization of the grid space makes data comparison between maps that were not collected under those standards impossible. Few attempts have been made in the field to employ mathematical model fitting to estimate between the collected data points and generate a function that can be sampled anywhere for a predicted value (Arya et al. 2010). As such, we developed a workflow based on model fitting for motor map comparison and made this software to facilitate research into its applicability for investigational and clinical use. The advantages of this approach are that making comparisons between maps are substantially simpler as the motor map function can be sampled at any location and compared with its likewise point from another motor map. The fitted models also lend themselves more easily to computations of surface area and volume integral. Furthermore, the particular kind of model fitting algorithms included in this package are non-parametric algorithms developed for creating smooth interpolations with high goodness-of-fit. This is convenient for motor mapping data as the underlying distribution of TMS-evoked responses from the motor cortex are not fully understood, so a non-parametric model that makes no such assumptions is ideal.

Installation

The software can be easily downloaded on our github page: <https://github.com/EdwardsLabNeuroSci/NeuroMeasure>

To download the application installer, navigate to the Releases tab and download the latest release (NeuroMeasure v4.4) for either mac or windows depending on what operating system you are using.

If you are a developer and wish to add to the NeuroMeasure project, clone the repository from the master branch, or download the source code from the latest release. To make edits to the code, you will need to own a MATLAB license and should have downloaded Matlab v 2017b.

## Requirements

NeuroMeasure is compiled for use on 64-bit editions of Windows 7, Windows 10, and Mac OSX. NeuroMeasure may work on other versions of these operating systems, but has not been tested on any other operating system or architecture.

It is recommended that you use a computer with at least 4GB of RAM when installing this software. Lower RAM may cause processes to run extremely slowly, or become unable to continue.

Import Scan

When initially launching the application, the graphic user interface (GUI) window will appear with only the *Import Scan* button enabled as shown in figure 1. Both Nexstim and BrainSight systems register their neuronavigated data points to imaging data of the patient’s head. The first step to beginning a new analysis is to import the scan to which the TMS data points are aligned. The application currently accepts either T1 or T2 modalities as valid imaging entries. This is a required part of the process because of the necessity of locating the head’s centroid, which plays an important role in the surface fitting and subsequent surface area and volume integral calculations (See the Surface Fitting section for more information). Clicking on *Import Scan* will launch an analysis window where the user will be prompted to select a folder. Navigate to the folder containing the desired stack of dicom images and confirm the selection to begin the scan import process.

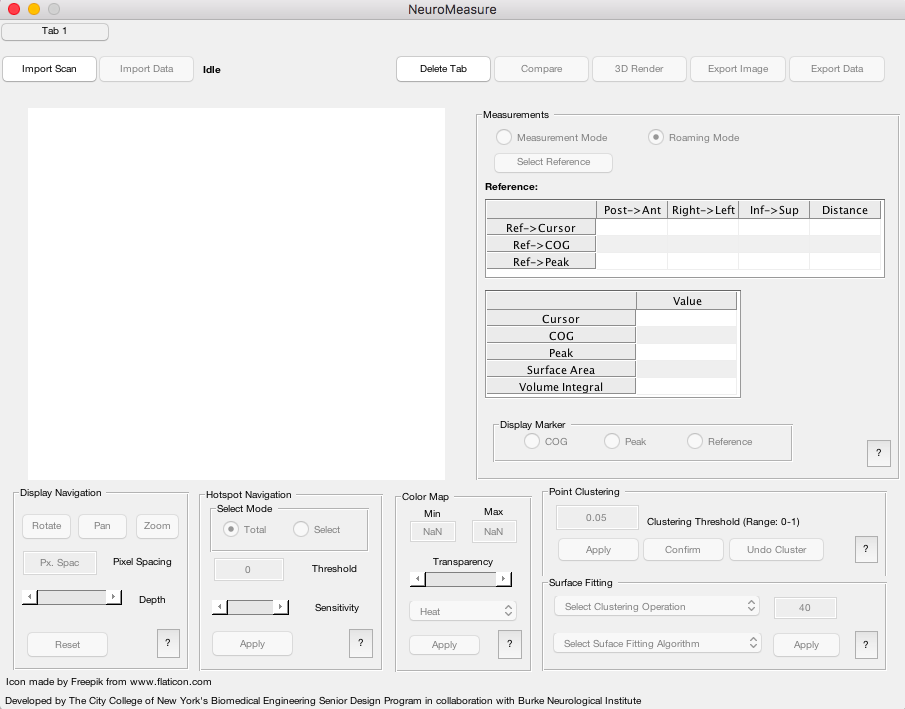


Figure 1: The NeuroMeasure GUI upon initial launch.

## Dicom Format Requirements

After confirming the scan directory, the application will begin reading dicom files immediately. Dicom files are first sorted away from non dicom files using MATLAB’s “isdicom” function (see the Mathworks website for further documentation). Files determined to be of the dicom format are then further sorted into those that contain the metadata tags “ImagePositionPatient”, “ImageOrientationPatient”, ”PixelSpacing”, ”SliceThickness”, “Height”, “Width”, and “InstanceNumber”, from those that do not. Dicoms that meet those criteria are then grouped with those that have the same “ImageOrientationPatient”, ”PixelSpacing”, ”SliceThickness”, “Height”, and “Width” into sets. These grouped dicom sets are than organized in ascending order based on their “InstanceNumber”. Sets that either have more than 10 discontinuities in their ordering (defined as increases in the “InstanceNumber” greater than 1) or contain less than 10 valid dicom slices overall, are discarded. If there is only one remaining set of valid dicoms, the system will automatically progress into the alignment phase. If not, a dialog will be launched that asks the user to select among the remaining valid sets, pictured in figure 2. The dialog will display some characteristics of the chosen scan to help the user identify it.

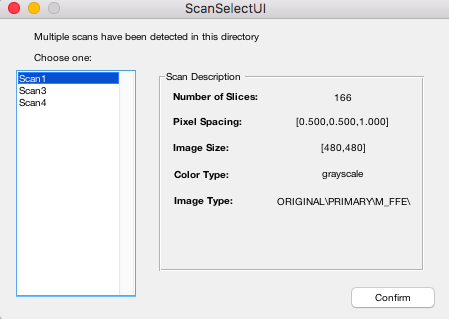


Figure 2: The scan selection dialog launched when more than one valid set of dicoms exists in a folder. The Number of Slices entry shows how many valid image slices are grouped in that set. The first two entries of Pixel Spacing are the in-plane dimensions of pixel spacing and the last entry is the out-of-plane pixel spacing (slice thickness). The image size shows the Height and Width metadata entries. The Color Type and Image Type show the ColorType and ImageType entries from the scan metadata if they exist and will not appear if those tags do not exist.

In summary, the dicom formatting requirements for a scan to be imported to NeuroMeasure include:

* Must be classified as dicom by MATLAB’s “isdicom” function
* Must contain the tags “ImagePositionPatient”, “ImageOrientationPatient”, ”PixelSpacing”, ”SliceThickness”, “Height”, “Width”, and “InstanceNumber”
* Must contain more than 10 slices with the same orientation as determined by having the same “ImageOrientationPatient” vectors, pixel spacing and slice thickness as determined by having the same “PixelSpacing” and “SliceThickness” entries, and image dimensions as determined by having the same “Height” and “Width” entries.
* Must use the “InstanceNumber” tag as an index for slicing order and that index should have fewer than 10 discontinuities.

Scan Alignment

Following dicom set upload, the application will move into the Scan Alignment phase. The scan alignment window will be launched as shown in figure 3. The left *Input* image displays the uploaded scan and the right *Standard* image displays a sample image volume showing the standard head orientation. The user may use the slider to flip through the image volume and make sure that the *Input* scan is properly loaded. If the scan is not properly loaded, and *Abort* button is available. At this stage, the user must use the rotation and flip tools on the bottom left of the GUI to position the *Input* scan so that it is oriented like the *Standard*. The rotate buttons each impart a counterclockwise turn on the labeled axis and flip imparts a reflection over the labeled axis. As evident by the change in figure window size, the aspect ratio of the image is not altered by each manipulation; the orientation alone is affected. It is not recommended to use the flip buttons as reflecting the scan will affect the chirality of the head. Use only the rotate tools unless the scan is taken under radiological conventions such that the right hemisphere is in the position of the left. Before clicking *Done*, the user may select the segmentation mode, which will be discussed in the next section.

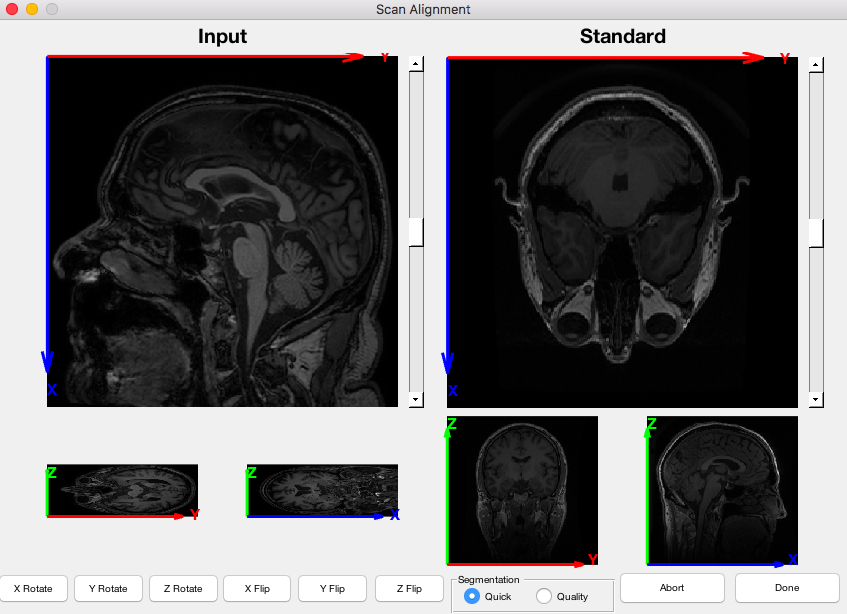


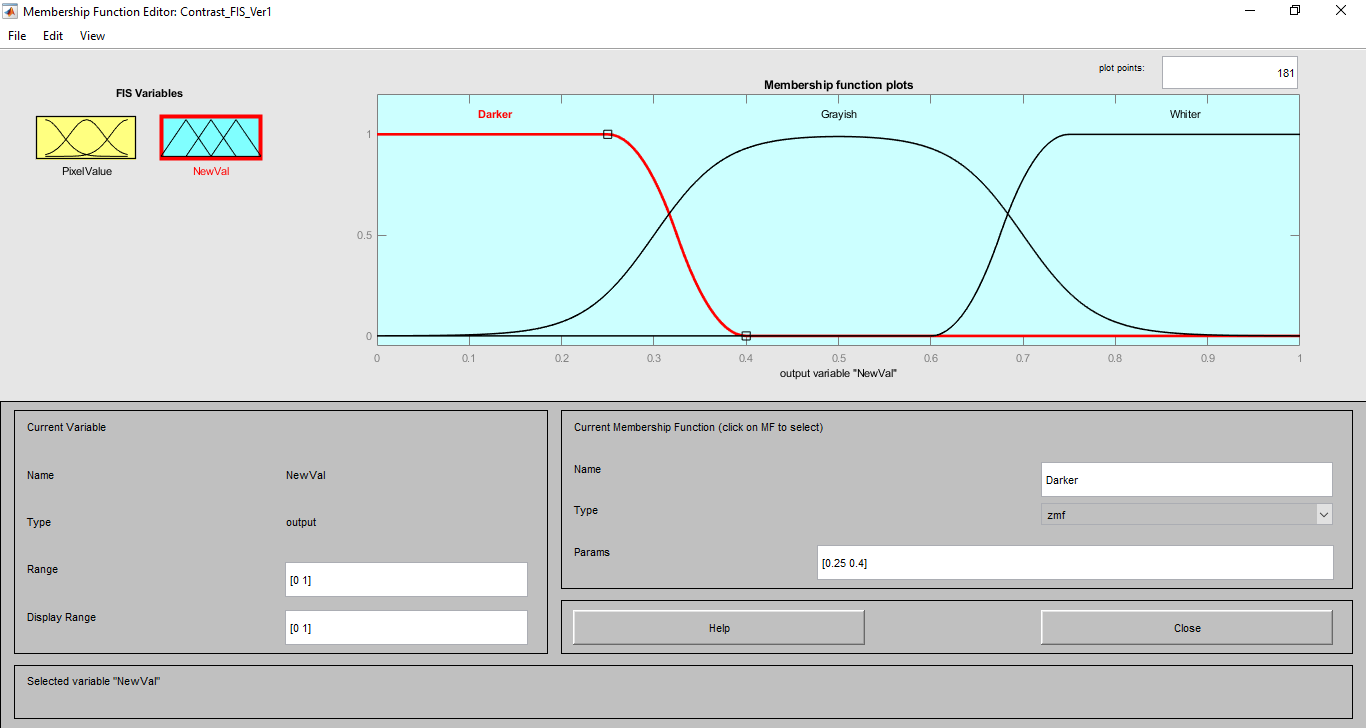
Figure 3: The scan alignment GUI upon initial launch showing a typical scan upload. The Input on the left shows the imported scan. The Standard on the right shows a standard scan in standard alignment. In this particular case, the *Input* scan would need to be rotated once over X and once over Y to reach standard alignment.

## Segmentation

The “*Quick* option segments the scans with a multi-level Otsu’s thresholding method. This algorithm reduces the grayscale images to several different color levels by binning the pixel colors to discrete histograms. Consequently, some scans and 3D rendered images may appear slightly inaccurate with minor distortions to brain anatomy since a Boolean system is being applied to a non-Boolean data set. However, this option is less computationally onerous than the latter. The *Quality* segmentation employs fuzzy logic to enhance image contrast and more accurately segment scans relative to the *Quick* option. Fuzzy logic deals with the fuzzy object class set; objects that have arbitrary parameters with varying degrees of membership to each parameter. MRI scans fall under this class because while a grayscale image does consist of three colors (black, gray and white), each of these colors are arbitrary denotations we apply to every pixel that do not encompass all color variants in the image. Some pixels appear darker or brighter than others, but the degree of darkness/brightness has no clear definition outside of what human eyes can perceive. Thus, a grayscale MRI image has 3 membership functions for each definite color and all pixels have varying assignment to these functions (i.e. some pixels look more gray than black, some pixels look more gray than white, some pixels look more white than gray, but to varying degrees). This explains why Boolean approaches to MRI segmentation are occasionally false in reproduction; the variance of these three specified colors for all pixels is lost in Boolean categorization. Fuzzy logic also utilizes a Boolean-like rule system to link membership functions from the input to output with specified weights. Rules are processed through an implication method that swings input to some output while considering respective weights. All assigned functions are then aggregated and “defuzzified” to return the output value. The contrast of the scans are enhanced by the change in membership functions in Figures 4a and 4b. Each pixel value is assigned to a membership function and depending on where the value is located on the function(s), it is pushed to an output after being processed by implication. The exact choice of membership function type was purely experimental, so the current fuzzy inference system is not as powerful as it can be. The type of inference system (Mamdani type 1) was chosen due to how suited it is for human input, although it is less computationally efficient than Sugeno types.

Figure 4: a) Membership Functions for Fuzzy Logic Input. Each primary grayscale color has a correlated membership function and any pixel value will reside on any one of these functions. These functions do not define the actual color distribution of MRI pixels, but only estimate it.

b) Membership Functions for Fuzzy Logic Output. These functions have less intrinsic variance than their input counterparts, which help enhance contrast. This produces a larger threshold for pixels to be considered one of the three colors or some degree of it.



**a**

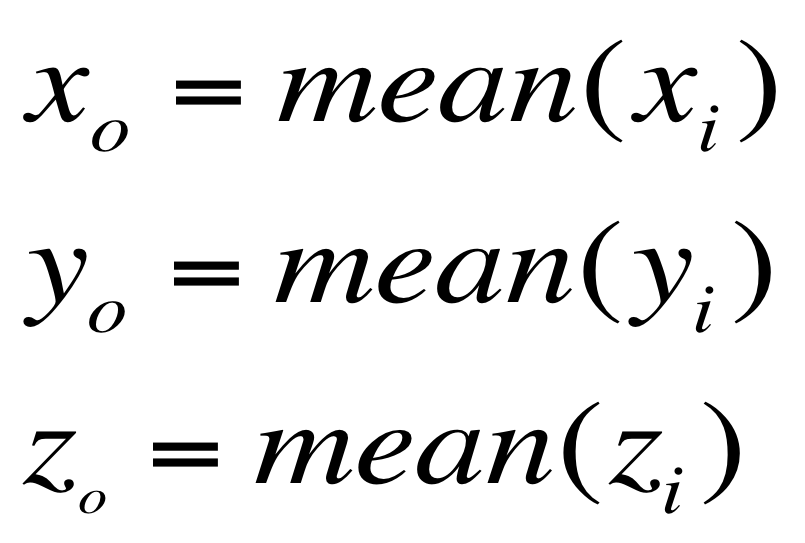
**b**

## Display Navigation

Once data upload is completed, the scan will appear in a 2D plane. There are options to rotate, pan or zoom into the MRI from any position. Buttons *Rotate* and *Pan* are enabled in *Roaming Mode* where the user has freedom to view the MRI scan without clustering and fitting MEP amplitudes. The creation of these two modes came from a consequence of the method for MRI display in 2D from 3D. Returning to *Roaming Mode* after fitting MEP amplitudes in *Measurement Mode* will remove any applied fit since all MEP-based calculations in *Roaming Mode* would be incorrect (see Surface Fitting section for more details). There is a depth slider within this panel that allows for peeling further into the scan and *Reset* reverses the effects of *Zoom*. Pixel spacing can also be altered however the user must be careful with this setting as changing it has several implications: a higher resolution image takes increasingly greater processing and will therefore slow down the application significantly, however lowering the resolution will lower the accuracy of the *Surface Area* and *Volume Integral* approximation measurements (See the Measurements section for details)

Topographic Display

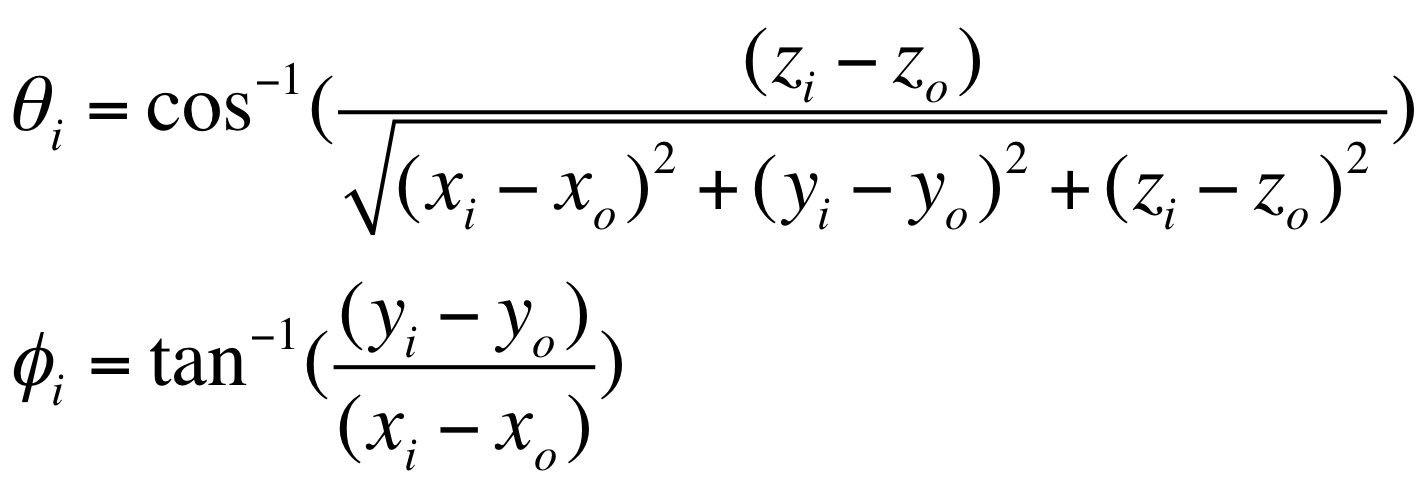
This section will describe the mathematics of generating the topographic map of the cerebral cortex that is displayed upon successful upload of an MRI scan as shown in figure 5f.



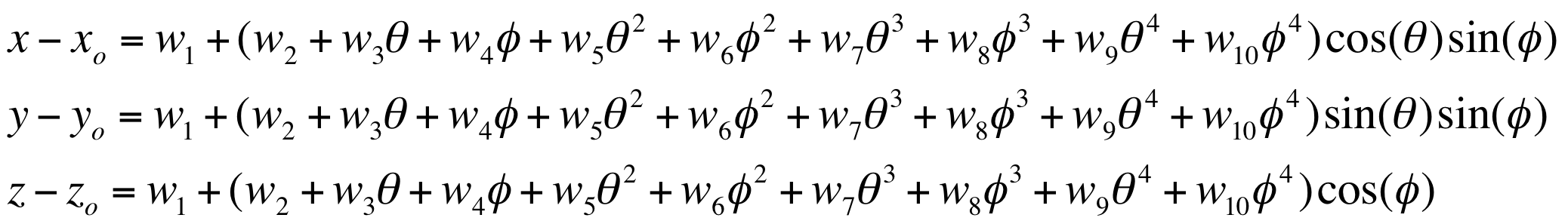
Equation 1: xo, yo, zo are the coordinates of the centroid. xi, yi, zi are the coordinates of all of the points in the point cloud.

NeuroMeasure’s segmentation algorithm outputs a binary mask of the head which is then converted into a point cloud by sampling the 3D coordinate positions of every 25th “white” voxel in the binarized image volume shown in figure 5a. The centroid of this point cloud is than computed with via equation 1. The point cloud is then converted from 3D cartesian coordinates into angular coordinates from the centroid using equation 2.

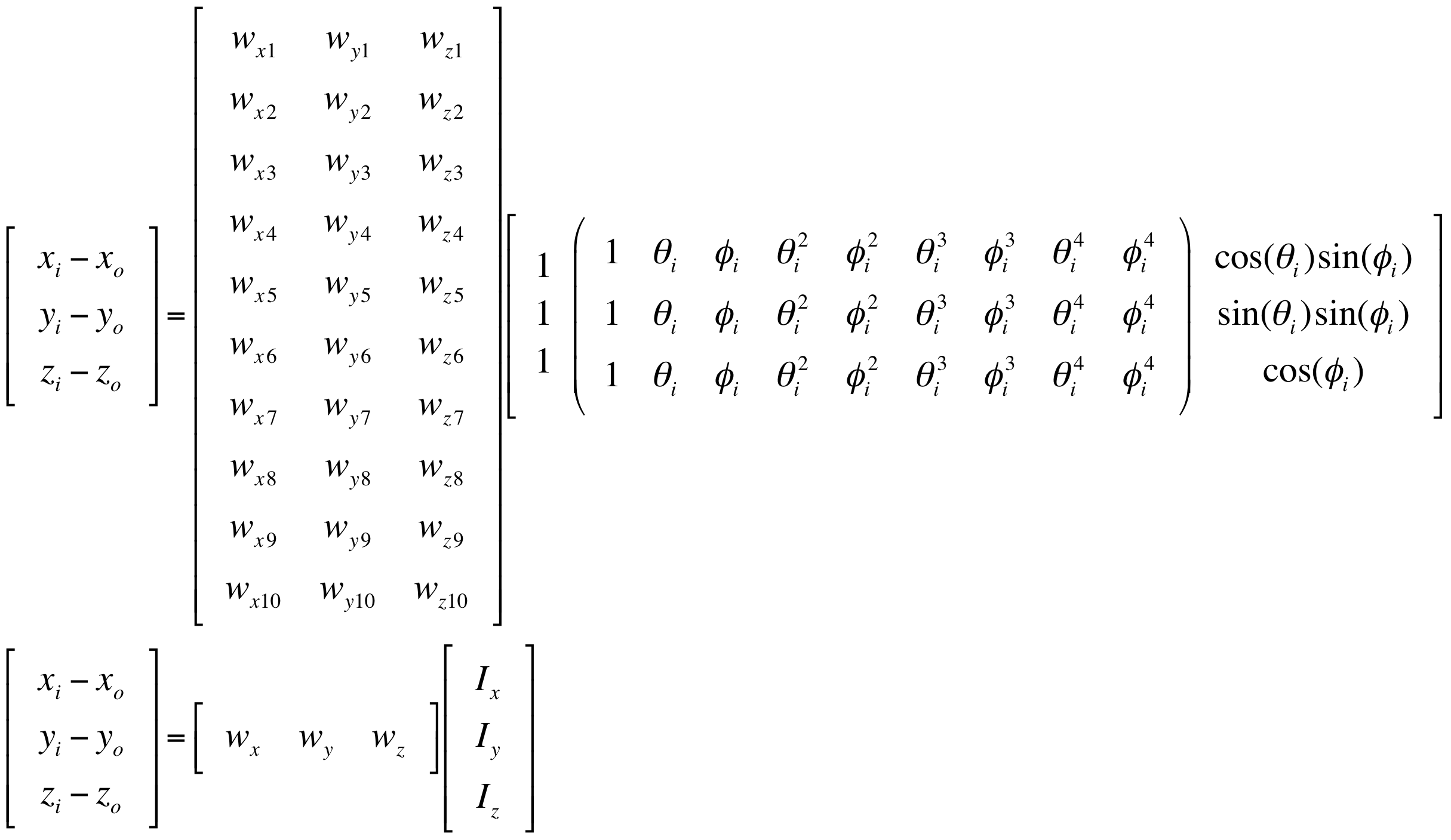
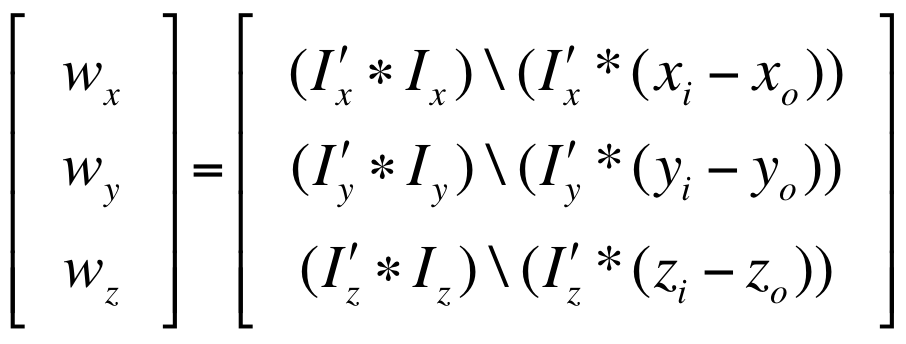
Equation 2: and are the azimuth and inclination angle respectively. xi,yi,zi are the coordinates of the point cloud and xo, yo, zo are the coordinates of the centroid.



We now use least squares regression to fit a function that relates angular coordinates, expressed as (θ,ϕ), to 3D cartesian coordinates expressed as (x,y,z). The function was chosen to be that of an ellipsoid with a radial term represented by a fourth order polynomial, shown in equation 3, converted into matrix format in equation 4 and the weights solved for in equation 5.

The solved weights from equation 5 can then be substituted back into equation 3 to compute the equivalent 3D cartesian coordinates from any set of angular coordinates. This establishes a means by which 3D spatial position can be registered onto a 2D gridspace and projected into an image. This process is shown visually in figure 5, where a point cloud (figure 5a) representing the segmentation of a head is fitted to an ellipsoid figure 5b/c) whose surface is characterized by an angular coordinate system (figure 5d). The MRI scan can then be sampled by reading the grayscale voxel position at every angular position on the ellipsoid’s surface, which can then be represented as a 2D topographic map shown in figure 5e. This process of converting 3D Cartesian coordinates to 2D angular coordinates and using a fitted surface function to convert between 2D and 3D space is a common motif employed within our software and we use a similar technique for processing imported motor mapping data. More on the can be found in the Surface Fitting section.

Equation 3: and are the azimuth and inclination angle respectively. xi,yi,zi are the coordinates of the point cloud and xo, yo, zo are the coordinates of the centroid.



Equation 5: Solving for the parametric weights using the standard form of the least squares regression formula.

Equation 4: a) The same equation from equation 3 substituted for (xi,yi,zi) the 3D cartesian coordinates of the point cloud and their angular equivalents (,) and converted into matrix

format, b) the same equation from (a) abbreviated with substitutions.

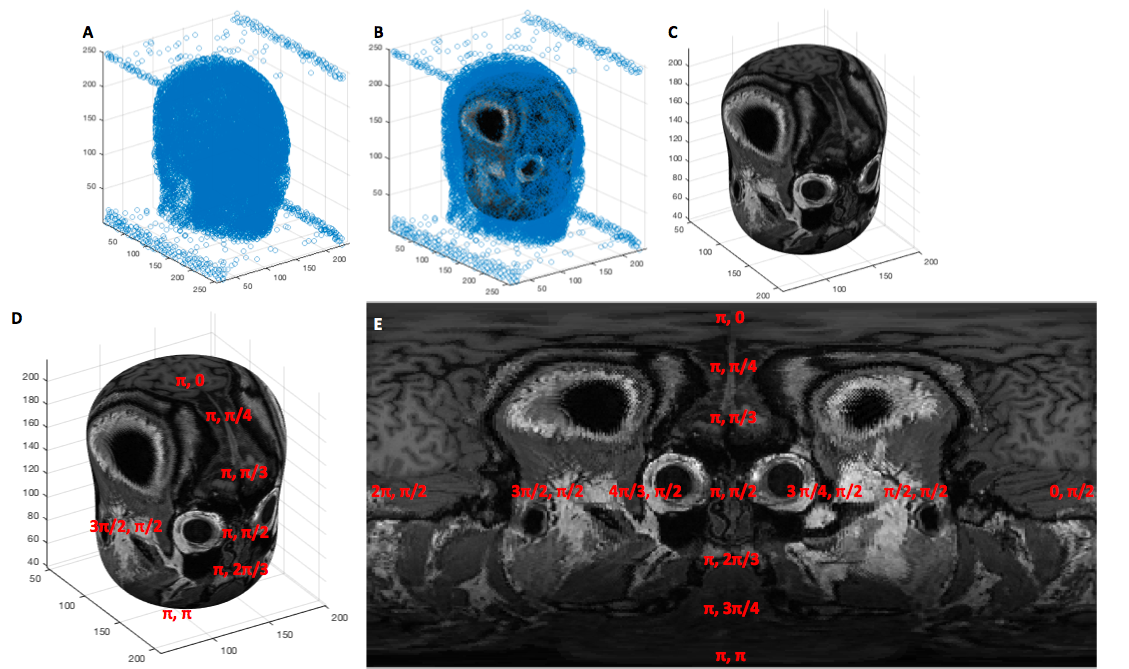
One important note about the topographic display is that it is not a perfect system. The topographic display can be thought of as a projection of the surface of the fitted ellipsoid unwrapped into a map. It is desirable that, if we were to choose a point on our 2D map and calculate its straight distance to another 2D point, this distance would be proportional to the length of an arc moving along the surface of the 3D ellipsoid between the two equivalent points. This, however, is not the case because an ellipsoid is a geodesic object and its surface area can never be perfectly represented on a 2D map. For example, consider the more simplified geometry of a sphere shown in figure 6. The gridlines show how angular coordinates describe every position on the sphere’s surface. The longitude and latitude lines represent increments of θ and ϕ respectively. Note how as the latitude increases towards the pole, the arc length between two longitude lines gets smaller until they eventually converge onto a singularity at the pole. At this singularity, the coordinates (π,0), (2π/3,0), (π/4,0) all represent different positions on the 2D topographic map, but the same 3D point on the ellipsoid’s surface. In fact, this is true everywhere on the map to some degree; the proportionality between angular dimensions and 3D arc length is not conserved throughout the map and thus there will always be some misrepresentation of distance depending on what proportion is used as the reference (in our case, the equator). This is the reason why the topographic map appears “distorted” towards its extremes. Although this effect is purely a visual issue in this case, because surface fitting also relies on this conversion, it has implications regarding the interpolated motor map and measurements that rely on it like *Surface Area* and *Volume Integral*. See the surface fitting section to learn more about these implications.

Figure 5: a) A point cloud generated from sampling the segmentation of the head every 25th voxel, b) An ellipsoid fit to the point cloud via least squares regression, c) the ellipsoid shown without the point cloud surrounding it, d/e) the elliptic surface position is described by an angular coordinate system that, when projected onto an image, creates the topographic display.

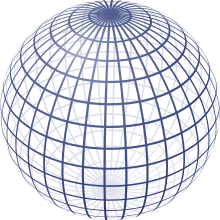
 Note that in figure 5e, sections of the frontal cortex appear to be occluded by skull and derma. This is a result of a poor segmentation causing the ellipsoid to “bulge out” near those regions, thus slicing MRI scan outside of the cortex. In this particular case, using the depth slide to peel deeper into the brain, or the rotate option to turn the ellipsoid so that it gets a better cut of the cortex are all viable strategies to compensate for poor segmentation. The quality of the MRI cut, can however, vary depending on the input scan. One particular example of this is how the algorithm processes scans with anisotropic pixel spacing. Scans that are “thinner” in one direction than they are in other directions can sometimes result in cuts where the ellipsoid bulges out of the cortex in the thinner orientation, as shown in figure 7. **We stress that this is a visual bug only and not one that will impact measurements in any way.**

Figure 6: A representation of how the angular coordinate system is mapped onto the surface of a sphere. Retrieved from: <https://en.wikipedia.org/wiki/Sphere>

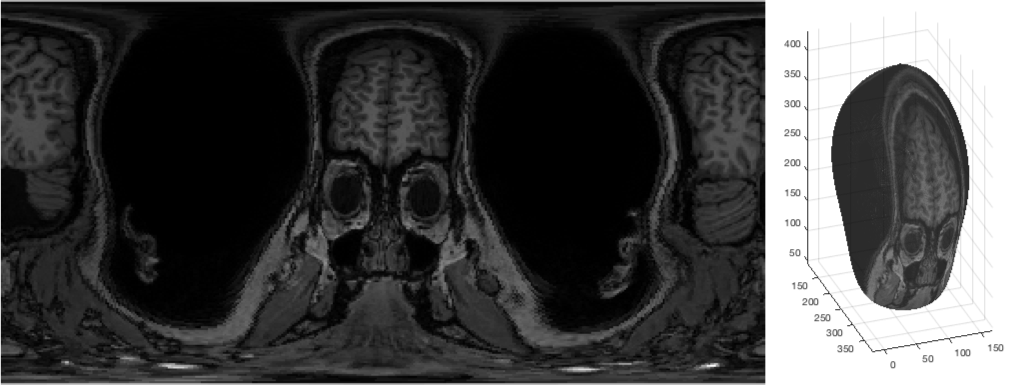
 Import Data

Figure 7: An example of a topographic map generated from a scan that was sliced thicker sagittally than it was sampled coronally and transversely. The ellipsoid bulges out of the head sagittally and slices through the skull instead of the cortex. Currently this is a known visual bug, but it can be compensated for using the rotate tool and depth slider.

## Data Format Requirements

MEP data must be formatted in an xls or xlsx (Microsoft Excel) file. Data must be formatted into four columns: The first three columns must contain the X, Y, and Z coordinates of each TMS stimulation point, and the fourth column must contain the MEP amplitude value of interest corresponding to that point. The intensity can be anything, although the software was designed with the intention of using the peak-to-peak amplitudes of EMG readings. If you attempt to upload data using a different format, a warning will display and the main window will reappear.

## Import Reference Point

If you are using Nexstim Head Coordinates, Then the *data import* window will display text boxes for corresponding X, Y, and Z coordinates for left and right inner ears, and tip of the nose. Simply insert the correct coordinates into the text boxes, and NeuroMeasure will align the coordinate axes into Nexstim Head Coordinates. The origin 0,0,0 point will be automatically calculated in the center of the head. This will be the default reference point if no reference coordinates are entered at the bottom of the window.

If you are not using Nexstim Head Coordinates, then the coordinate system will be automatically aligned based on your coordinate selection. Additionally, the reference point can be manually entered at the bottom of the window.

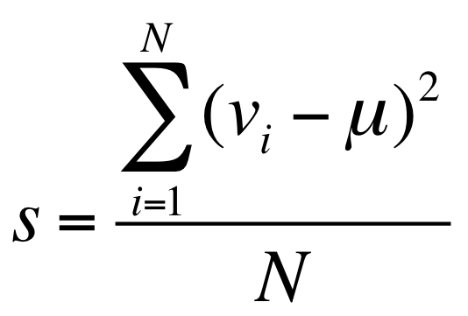
Quality Threshold Clustering

For repeated-measures experiments, NeuroMeasure uses a quality-threshold clustering algorithm that offers a significant amount of freedom in the analysis and interpretation of multiple data points. This algorithm uses a distance threshold to group points that are close together (measured by Euclidean distance). This distance threshold is normalized from 0 to 1, where the highest value of 1 represents the largest Euclidean distance between any two points in the dataset (at 1, every point in the dataset will be grouped into one cluster), and the lowest value of 0 represents a Euclidean distance of 0 (at 0, every point in the dataset will be its own cluster regardless of how close they are together unless two points have exactly the same position coordinates). To perform a clustering after data has been imported, simply enter a value between zero and one into the clustering input box on the lower right of the screen, and click the *Apply* button. Data points on the screen will be displayed in different colors, with each ‘node site’ shown in the same color. We recommend starting with the default of 0.05 and then making adjustments until all of the points that are intended to be repeated measurements are grouped together. Once all sites have been clustered properly, click the *Confirm* button to finalize the clustering and the data will be returned to its previous color mapping. The default color map uses black as a minimum intensity, so some points may be difficult to see over the grayscale MRI image. No data is strictly lost during clustering, although it may visually appear to be the case. The type of color map can be changed in the *Color Map* panel to the immediate left of the *Point Clustering* panel. Once data has been clustered, proceed to fit the data in *Measurement Mode*. In *Measurement Mode*, MEP points and MRI image are fixed, so the *Roaming* and *Pan* buttons are disabled.

Surface Fitting

NeuroMeasure is equipped with the capability of fitting a predictive model to the imported motor mapping data in order to spatially interpolate values. Surface fitting is only available after the data has been clustered, as it was found that in motor mapping studies with repeated measurements, that predictive models fit onto unclustered data generated low stabillity and choppy fits due to highly variant data points in close proximity with one another. If the user is conducting a single measurement experiment, set the clustering threshold to 0 as described in the Quality Threshold Clustering section. Then, proceed as normal.

## Clustering Operations

 The first step to generating a motor map of the TMS experiment data is to select the operation that will be done on point clusters. The term *Clustering Operation* in this manual refers to the processing of grouped, “repeated measurement” values to generate a single value that will represent the position of those repeated measurements.

Equation 6: The variance formula, where s is the variance, N is the total number of points in the cluster, vi is the ith value of the points in the cluster and μ is the mean of the cluster. Variance represents the average of squared differences between each individual value of the grouped points and the overall mean.

|  |  |  |
| --- | --- | --- |
| Operation | Summary | Output Data Type |
| Average | Average of grouped values assigned to cluster | Continuous; Units of input values |
| Maximum | Maximum value assigned to cluster | Continuous; Units of input values |
| Minimum | Minimum value assigned to cluster | Continuous; Units of input values |
| Variance | Variance as computed by the formula below, on the data within the cluster. | Continuous;  Units of input values squared |
| Probability | Chosen threshold value binarizes all values in cluster. Probability is than defined as the average of the binary values (such that in a cluster with three total entries, two above threshold and one below, the probability would be 66%) thus representing the probability that a value above the chosen threshold will manifest at that location | Categorical; Units of probability (%) |

There are five supported operations that can be performed on each node within the clustered set to process the multiple points. The current operations are averaging, maximum, minimum, variance and probability. Selecting average will simply set the mean of the grouped values to each of the positions of the clusters from which the final map will be generated. Maximum and minimum will similarly set the value of the cluster position to the maximum or the minimum of the grouped values respectively. The units of the motor map constructed from average, maximum and minimum clustering operations is the unit of the imported data (if imported values are in units of μV, the map is in units of μV). Selecting variance will use the variance formula shown in equation 6, to compute the variance of the grouped points. The units of variance are the units of the imported values squared (if imported values are in units of μV, the map is in units of μV2).

Table 1: Summary of all available clustering operations and the units in which their measurements are computed.

Finally, probability is a special case that is different from the four clustering operations mentioned previously; unlike the last four which kept the data in its raw, continuous numerical format, probability will binarize the values so that they are categorical. When selecting probability, an edit box to the side of the popup menu will become enabled. The value in this edit box (40 by default) is a binarization threshold. The raw dataset is binarized with this threshold so that all values above it are 1, and all values below it are 0. Then, a probability score is computed for each cluster depending on how many values within the group are above or below the threshold. For example, if a cluster contains four values, and two of those values are above the threshold and therefor 1, and two values are below the threshold and therefor 0, the overall probability value assigned to the cluster is 50%. Alternatively if there are three values in a cluster and two are 1’s and one is a 0, the overall probability is 66.67%. In single measurement experiments where there is only one value per cluster, the data points will always be either 0% or 100%. The units of the motor map when generated from the probability operation are in unitless %. Measurements like *peak position*, *center of gravity*, and *surface area* and *volume integral* are computed with % values so much of their meaning is lost. The value of using the probability option is in comparison with other motor maps, which is explained in more detail in the Categorical subpart of the Compare section. The chosen threshold will greatly affect the results of the model fitting. Currently, there are no recommendation for setting the binarization threshold as the inclusion of this feature into NeuroMeasure is highly experimental. Table 1 summarizes the different clustering operations and their units.

## Surface Fitting Algorithms

Surface fitting is the process of fitting a continuous function of the form z = S(x,y), to a set of known data points (x,y,z), that uses two parameters x and y to predict values of z. The brain, however, is a three dimensional object and its positions are described by three spatial coordinates (x,y,z) and a fourth parameter that represents the intensity of the motor map. Thus, in order to fit motor mapping data described by three position parameters and one intensity parameter, we must choose to either convert our dataset into a format that works with surface fitting, or use a different predictive modeling approach that is capable of fitting data described by 4 parameters. In order to make this decision, we must first consider the nature of neuronavigated motor mapping data. The position of a point describes the peak magnetic field strength and its intensity is usually denoted as the peak to peak amplitude of the EMG response when the magnetic field is applied. The depth of the peak of the magnetic field is fixed by the physical limitations of the system so the motor map is characteristically planar. Since there is no value information to predict along the depth orientation of the head, a 2D model is sufficient. Thus, we begin by conforming the 3D positional coordinates into a 2D system that can than be applied towards surface fitting.

The methodology for converting 3D position coordinates into 2D coordinates has already been established in the Topographic Display section. In brief, we convert the 3D Cartesian coordinates to angular coordinates centered around the centroid of the head[[1]](#footnote-1). The same ellipsoid fitting process used for slicing the MRI is used for the motor mapping data to establish a way to convert from angular coordinates back into 3D Cartesian coordinates. The data, now in the form of angular coordinates, is then fed into a surface fitting algorithm that computes a function predicting intensity values at any given position. Notably, conversion into angular coordinates is a lossy process. The proportionality between angular distance and 3D Cartesian distance is not maintained throughout the map as a result of two discrete effects. The first, which is discussed in the Topographic Display section, happens due to the geodesic nature of spherical shapes and their increasing misrepresentation of distance proportionality as one approaches the poles of the coordinate system. The second is a problem of projection; radial position is lost upon conversion to angular coordinates which has the effect of “morphing” the positions of data points onto the surface of a sphere with a center of curvature at the centroid of the head. The closer the shape of the sampling plane is to that of the morphed shape, the less of an effect this distortion will have, but this is uncommon. As a result, the surface fitting algorithm fits data points whose positions with respect to one another are only approximate to the data that is imported. The approximation of the sampling plane to a spherical shape is a better approximation than it being planar, which is why this approach was chosen over z-squashing.

|  |  |
| --- | --- |
| Algorithm | Summary |
| Piecewise Cubic Spline | 3rd Order 2D polynomial fit to dataset in patches of 3 data points each, fit in a piecewise fashion directly without least squares. Edges between patches are smoothed by splines. |
| Local Linear Weighted Scatter Smoothing; a.k.a Lowess | 1st Order 2D polynomial fit to dataset via least squares regression combined with a ratio that splits the dataset into local parts and the polynomial is fit in a piecewise fashion. Ratio controls the degree of smoothing. Here, the ratio is fixed to 0.25 (0 = least smooth, 1 = most smooth) |
| Biharmonic (v4) | The same algorithm used in MATLAB’s griddata function. See the comprehensive explanation on the Mathworks website griddata documentation: https://www.mathworks.com/help/matlab/ref/griddata |

The application currently supports four diverse non-parametric interpolation methods to yield a heat map from the post-clustered MEP amplitudes. The algorithms, being non-parametric, mean they do not assume anything about the distribution of the data and provide only provide a high goodness-of-fit; this was deemed ideal for motor mapping data where the expected distribution is not well understood. Piecewise Cubic uses a bi-cubic function as the basis of interpolation. The cubic algorithm fits a different function between every three data points and connects all the patches to produce a single curve/surface. Piecewise Linear utilizes a similar algorithm but uses a bi-linear function as opposed to bi-cubic. Biharmonic (v4) is a fourth-order partial differential equation using the bilaplacian (biharmonic) operator, which is the square of the Laplacian operator. Unlike the piecewise cubic and piecewise linear algorithms that can produce either curves or surfaces depending on the data, biharmonic is only designed to yield a surface. Locally weighted scatterplot smoothing (lowess) method can smooth data through locally weighted linear regression. The piecewise cubic algorithm is recommended for most cases as it generates the best goodness of fit, however see the methods paper in our Github for a delineation of the pros and cons between different fit algorithms based on experimentation. The surface fitting algorithms are summarized in table 2. Also, see the Mathworks website curve fitting section for a detailed description of the different surface fitting algorithms available in NeuroMeasure: <https://www.mathworks.com/help/curvefit/interpolation-methods.html#bsz6baz>

Table 2: Summary of surface fitting algorithms available in NeuroMeasure.

## Color Mapping

NeuroMeasure has multiple color map choices for data visualization whose transparency and relative minimum/maximum can be manipulated by the user. These features can be found on the *Color Map* panel immediately left to the *Point Clustering* and *Surface Fitting* panels. Default *Heat* color map linearly raises in brightness from black (minimum) to orange (maximum) with shades of red representing intermittent intensities. The *Rainbow* color map utilizes a typical jet palette where dark blue represents the minima and light orange represents the maxima of the data set. Alternate options *Pink, Copper,* and *Bone* follow a similar minima to maxima gradient with complimentary colors for each map.To make map comparison more quantifiable, the voltage minimum and maximum can be altered to scale the color map and truncate/extend the gradient between the minimum and maximum. The transparency slider changes the titular feature of the colors within the map, which can be beneficial for better understanding the map topography or viewing the underlying brain anatomy.

Measurements

Upon fitting a surface function to the data points, the measured characteristics of the data will be available in the measurement panel. Fundamentally, the system measures five parameters: the center of gravity (COG), the peak value, the cursor position, the map’s surface area, and its volume integral. The first three are points with a position and a value at that position. Thus, they have entries in the position table where their position is given by their distance from the selected reference point in the three cardinal directions of the MRI scan, as well as in Euclidean distance as pictured in Figure 8B. They also have entries in the value table where the value of the interpolated map at those positions are given.

## Reference Point

All measurements of position in NeuroMeasure are reported with respect to a reference point chosen by the user. This allows for between-patient measurement comparisons from a standardized anatomical reference point and serves as a normalized means of reporting data. The reference point can be imported along with the dataset from Nexstim or BrainSight neuronavigation systems as described in the Data Import section. If left blank, the reference point will be set to the centroid of the head segmentation by default.

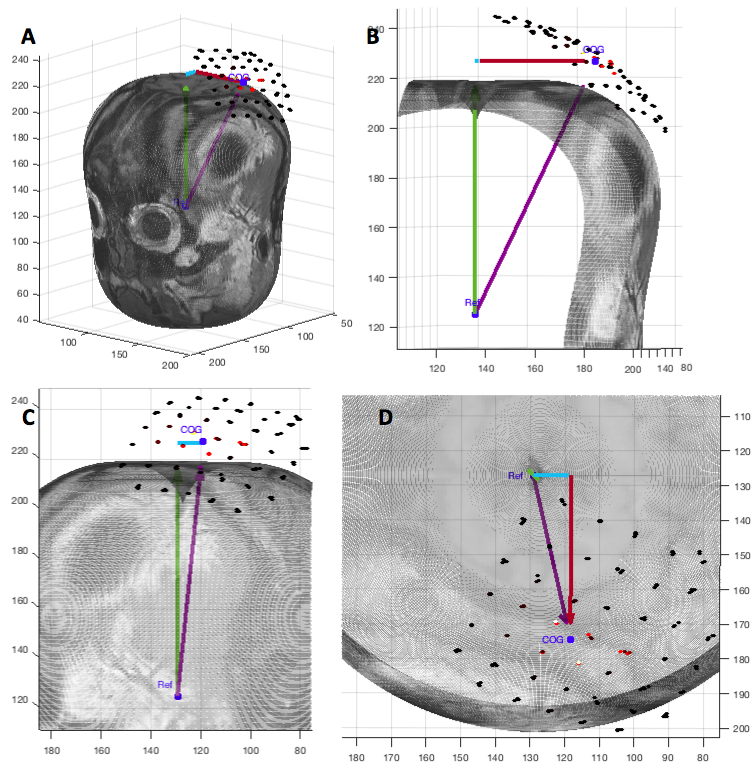
The reference point can also be changed interactively within the app. When in *Roaming Mode* the *Select Reference* button will be enabled. When pressed, hover the cursor over the topographic map of the brain and the reference entry in the position table will begin live tracking of cursor position coordinates over the scalp. Click anywhere on the topographic map of the brain to set the reference point to that location. Enable the Reference radio button within the *Display Markers* panel to place a marker over the current position of the reference point. Notably, the reference marker will not appear while the reference is in its default position at the centroid of the head for reasons clarified in the Topographic Display section of this manual.

Figure 8A-8D: A graphic display of the four position measurements reported in NeuroMeasure’s position table. The cyan arrow represents posterior->anterior distance. The red arrow represents right->left distance. The green arrow represents inferior->superior distance. The purple arrow represents Euclidean distance. The four measurements are reported in units of the imported scan’s resolution.

## Cursor Tracking

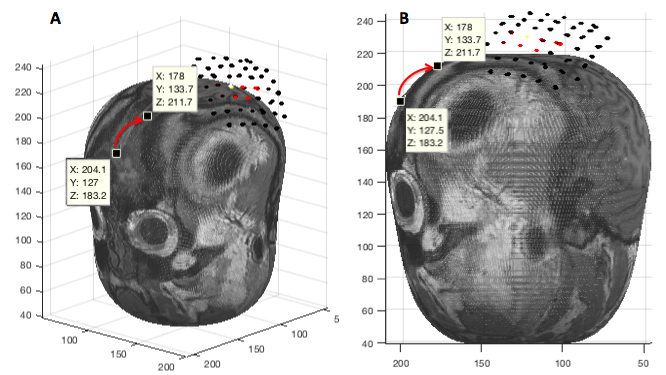
The COG and peak entries are fixed, however the cursor position and value tracks the movement of the mouse over the topographic map of the head and update their values accordingly. Notably, the topographic map of the brain is constructed from a fitted ellipsoid to the geometry of the head (as discussed in the Topographic Display section of this manual) as pictured in figure 9A, so the coordinates of the cursor correspond to the positions of the points on the ellipsoid as if the cursor’s movement was bound to its surface. Therefor, the position of the cursor is reported from the surface of the ellipsoid, but the value of the cursor is reported from the fitted data function whose position in space is within the plane of the imported data points as shown in figures 9A and 9B. This is why, when hovering the cursor over the marker for COG, the value reported in the cursor entry is the same as that in COG, but the positions are not the same. The radius of the fitted ellipsoid can be modulated by the depth slider, as mentioned in the Display Navigation section, which will impact the position reported by the cursor by moving deep or superficial to the cortex. Naturally, the depth slider has no effect on the cursor value, which probes the fitted data function within its fixed plane.

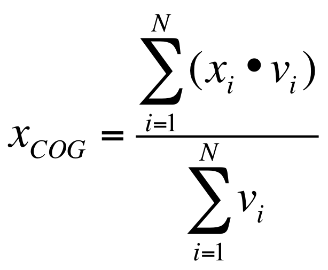
Figure 9: An example of how moving the cursor from one position to another on the 2D topographic map is akin to moving a cursor rigidly bound to the surface of an ellipsoid. The reported position values in the cursor position table are bound to the ellipse used to slice the image volume.

## 

Figure 10: One way to understand the meaning of the cursor position and cursor value measurements is to consider a ray passing from the centroid of the head to the current position of the cursor hovering over the topographic map. The ray intersects the ellipsoid used to slice through the image volume and the imported data points in 3D space. The position entry thus reports the position of the point made by intersecting the ray with the ellipsoid. The value entry reports the value of the fitted data function at the position of the point made by intersecting the ray with the plane of the interpolated map.

## Center of Gravity

The center of gravity is a well-established point of interest (Thickbroom et al. 1999) for characterizing a motor map because of its convenient condensation of the values and positions of many data points into one. Although referred to as center of gravity in the neuroscience literature, it should be noted that the engineering and mathematics literature refers to this measurement as center of mass due to it being a characteristic measurement of a scalar field as opposed to a vector field. Here, we will continue to call it center of gravity in accordance with neuroscience convention.



Equation 7: Center of Gravity Formula for three dimensions where xi are the x positions of the dataset, yi are the y positions of the dataset, zi are the z positions of the dataset, vi are the values of the dataset, and N is the total number of points in the dataset.

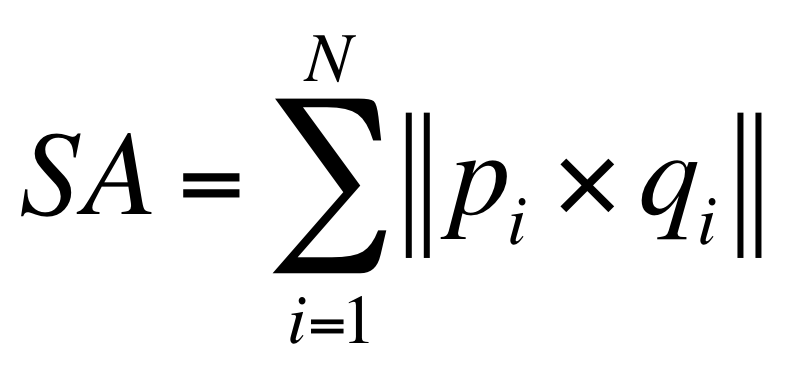
The position of the center of gravity is given by equation 7. The COG’s position is reported as distance, in units of the scan resolution, from the reference point as mentioned in the preface of this chapter. The value table also contains an entry for COG where the value of the fitted data function at that position is reported. Enable the COG radio button in the *Display Marker* panel to place a marker showing the computed COG position in both 2D and 3D views.

It is important to note for those who attempt to check the results of the computation by doing their own calculation on the raw data that the results will differ if the points have been clustered. The COG is computed based on the clustered points and will vary depending on whether you had selected “average” or “maximum” as this changes the values assigned to the points used for the COG calculation.

## Peak

The position of the peak value of the fitted data function is also reported. Although not as represented in the neuroscience literature, its use as a measurement may be of interest to some users. The peak position is given as the point in 3D space at which the highest value of the fitted data function occurs. Like the other measurements of position it is reported as distance, in units of the scan resolution, from the reference point. The value table also contains an entry for peak where the peak value is reported.

## Surface Area

 Surface area is computed using a discrete integration method shown in equation 8. The surface area is dependant on the threshold value entered into the *Hotspot Navigation* panel (0 by default).

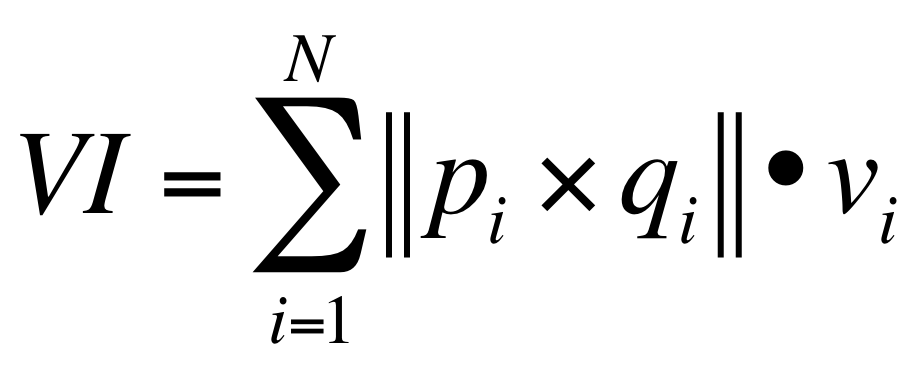
Equation 8: SA is the surface area value. pi and qi are the vectors pointing in the and . N represents the total number of positions whose values are above the threshold.

Consider one position on the motor map whose value is above the threshold. This position is described by two angular coordinates (,). Now, we define three points: (,), (+s,), (,+s) where s is the value of the pixel spacing of the topographic map (in radians). These three points are converted to 3D cartesian coordinates using the fitted ellipsoid function described in the Surface Fitting section. Now, we define a vector *p* between two 3D coordinates corresponding to (,) and (+s,). Then, we define a vector *q* between two 3D coordinates corresponding to (,) and (,+s). The norm of the cross product between these two vectors represents the surface area of the patch defined by that position on the topographic map. This process is then repeated for all positions on the topographic map whose values are above the threshold.

## Volume Integral

Volume integral is computed similarly to surface area as shown in equation 9.

Volume integral gives the volume under the curve of the motor map function by summing the volumes of patches whose intensity is higher than the threshold. Its units are (map unit)\*(scan resolution unit)2.



Equation 9: VI is the volume integral value. pi and qi are the vectors pointing in the and . vi represents the intensity of the motor map. N represents the total number of positions whose values are above the threshold.

## Image Export

In order to save images produced by NeuroMeasure, there is an *Image Export* button on the top right of the main window, and on the bottom right of the comparison window. Clicking the *Image Export* button will save a portable network graphics (png) image file to your computer containing the image currently displayed on the screen.

## Data Export

To save data produced by NeuroMeasure, there is a *Data Export* button on the top right of the main window, and on the bottom right of the comparison window. Clicking the *Data Export* button will save the relevant data calculated by NeuroMeasure on the screen to a text file. The resulting text file is illustrated in Figure 11.

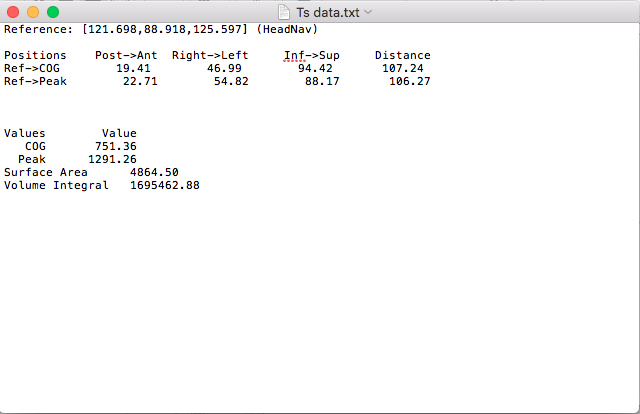


Figure 11: Content and Format of Output Text File

## 3D Render

*3D Render* generates a 3D reconstruction of the scanned head alone or with the projected MEP amplitudes if data was imported prior to clicking the button. The accuracy of the render depends on the segmentation type as mentioned previously. Changing the depth with the appropriate slider alters the apparent depth of the MEP amplitudes within the 3D render figure.

HotSpot Navigation

In some circumstances, such as in patients with cerebral palsy, motor maps may contain many individual peaks. Due to our sponsor having an interest in separating a whole motor map into individual peaks and measuring each separately, this feature was included in the final NeuroMeasure package, however no standardized workflow from the literature was used as its basis.

## Total Mode

This is the default mode that is enabled upon completion of surface fitting. In this mode, the *Measurements* panel displays values computed using all of the data in the data set. The only editable parameter is the *Threshold* which influences the *Surface Area* and *Volume Integral* measurements (See Measurements section for details).

## Select Mode

After completion of surface fitting, the user can switch into *Select Mode* from the default, *Total Mode*. Upon switching, an algorithm will separate the peaks of the motor map so that they can be selected and analyzed seperately. Clicking on the map will display a white border for the selected peak region and the *Measurement* panel will display values that were computed from data only within the white border. Notice that when COG and peak display markers are enabled, their position changes upon selection of a hotspot. The measurement readout adjusts automatically upon selection of a new hotspot. Switching back to total mode will recalculate all of the values characterizing the whole map.

The algorithm that seperates peaks begins by defining “local maxima” as points in which the surrounding motor map is decreasing in value. Each of these local maxima become a node that acts as the focal point of a region growing operation. Each region growing step takes turns between the different nodes for adding a pixel to it’s region. For example, take a map with three local maxima that become starting nodes. The first node adds a pixel to its region, then the second node adds a pixel, then the third node adds a pixel, and then it is the first node’s turn again. This ensures that each of the nodes have a fair claim to available space of the motor map. The region growing stops when a region attempts to annex a pixel that has already been claimed by a different region, or it reaches the map boundaries (set by the threshold in the *Hotspot Navigation* panel). Sometimes, two maxima will be near another one leading to a single hotspot being split into two regions. The sensitivity slider can be used to merge regions whose maxima are close together in order to make the hotspot “whole”.

Compare

A key feature of NeuroMeasure is the abillity to compare values between multiple maps. NeuroMeasure was designed to handle both temporal and spatial analysis of maps only within the same patient, however through the placement of a standardized reference point, comparisons between different patients is also possible, but not recommended, as the system does not currently have a setting for normalizing distance with respect to the scalp. There are two ways to compare motor maps within a patient; the first is through direct comparison of numerical values computed from continuous data type motor maps fitted from points treated with the Average, Minimum, Maximum or Variance *Clustering Operations*, and the second is through predictive analysis with a motor map that acts as a temporally predictive model of categorical data. The second is much more limited in scope than the first, but both are discussed below.

## Continuous Data Comparison

For any data set clustered with a continuous clustering mode discussed above, NeuroMeasure will fit both data with one selected fitting algorithm and display both maps and their difference, illustrated in Figure 12. The *Measurements* panel features not only single map information present in the main GUI window, but also metrics related to the difference, such as the change in center of gravity location, change in peak MEP location and surface area/volume integral difference. Color maps can still be altered in this window and all images and data can be exported. Continuous motor map comparison is recommended for comparing motor maps taken over time and between the left and right hemisphere. Currently, making comparisons between two different patients is not possible within the application, however if the user has a common point of reference established in BrainSight or Nexstim, and that common reference point is used when computing measurements between two different patients than COG and peak positions can be directly compared. Surface area, volume integral and peak/COG values can be compared regardless of common reference point as these measurements are independent of location.

One new metric available in the comparison window is the root mean square error (RMSE), displayed at the bottom of the *Measurements* panel in figure 12. The root mean square error is computed as shown in equation 10.

Equation 10: RMSE is the root mean square error value, n represents the total number of datapoints, Oi represents the ground truth value of one raw data point from Data Set 1, and Pi represents the predicted value from data set 2’s motor map function at the same location as Oi.

RMSE can be interpreted as the average difference between two likewise points on the two motor maps being compared. Note that RMSE is relevant only when the two maps being compared are sampling the same region of the brain so it is only applicable to temporal comparison on the same patient. In all other cases, the metric should be disregarded.

## Categorical Data Comparison

Categorical data comparison is launched when the clustering operation is set to probability and has only one specific usage: to quantify the stabillity of a motor map reading over time. When set to probabillity, the Compare button will compute the receiver operator curve (ROC) in a procedure outlined below.

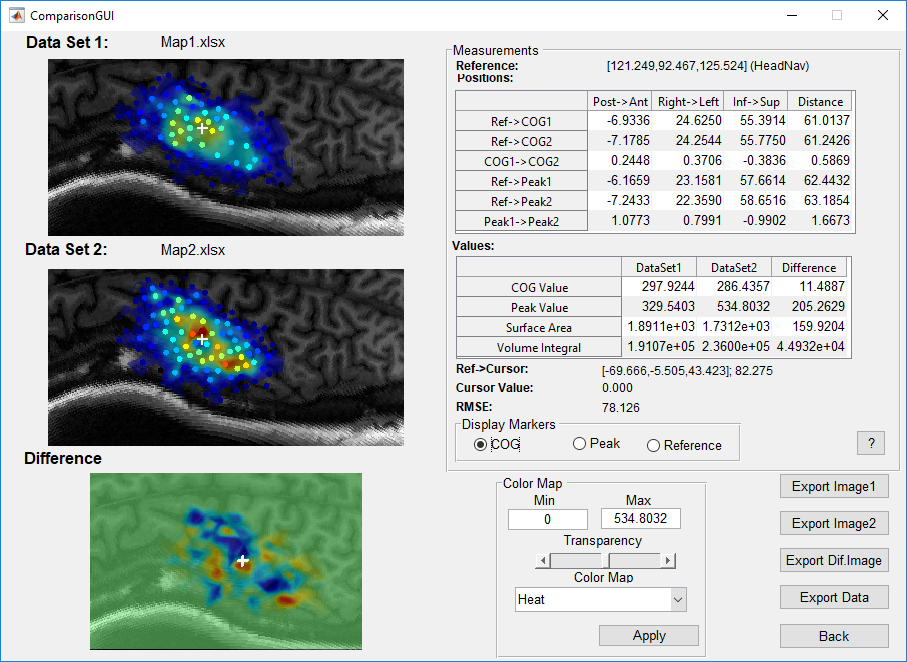
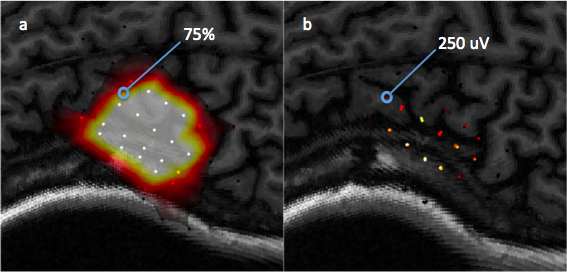
 A probabilistic model is first generated by fitting probabilistic clustered data points (computed as described in the Clustering Operations section) to a surface function as described in the Surface Fitting section. The result is a continuous surface function that predicts at every location a probability (from 0% to 100%) that a value above the set binarization threshold will be elicited by the TMS pulse (note that the term “value” is most often referring to the peak to peak amplitude of an MEP response, but it is kept general here as the model makes no assumptions). This leads to a motor map shown in in figure 13a. The dataset used to generate this predictive model will be referred to as the “training set”.

Figure 12: An example of the comparison window launched in continuous comparison mode. The reference point coordinates (in their respective coordinate system) is displayed on top and all position measurements are reported as distance between combinations of two points. COG and Peak value represent the value of the respective motor maps at the COG and peak position, and surface area and volume integral are as described in the Measurement section. Generated motor maps of dataset 1 and 2 are displayed on the left and the Difference map shows the error between the two maps at likewise positions. The Difference map’s color scheme is kept constant (blue negative, red positive, green zero) but the color maps of the other maps can be changed with the Color Map panel.

Now, the predictions of the model are tested using a “testing set” of data selected by the user when the Compare button is pressed, illustrated in figure 13b. Consider one data point from the testing set (for example, take the value of that point to be 250 uV as in figure 13b), and find the model’s prediction at that same location (for example, take the prediction to be 75% as in figure 13a). Now, lets say that the binarization threshold used to binarize our training data was 100uV. Thus, 250uV is higher than 100uV so this testing point has a value of 1. Now, we define an arbitrary map threshold that binarizes 75% to either a 0 or 1 and the model’s prediction is checked with the testing point. There are four possible outcomes illustrated in figure 14: the model and testing point are both 1 (true positive), the model predicts 0 but the testing point is 1 (false negative), the model predicts 1 but the testing point is 0 (false positive), and both the model and testing point are 0 (true negative). This is repeated for all points in the testing set. Now, we can compute the detection rate and the false positive rate using equations 11a and 11b.

Equation 11a/b: Formula for true positive rate (TPR) and false positive rate (FPR)

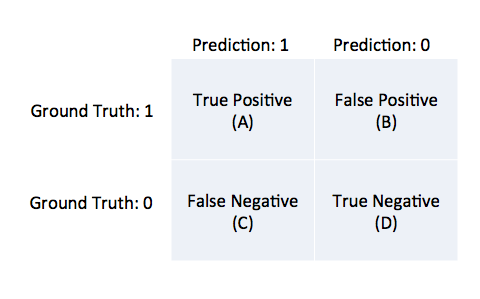
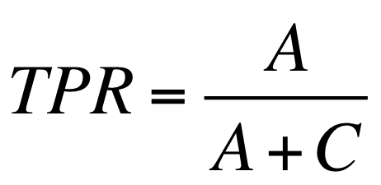
Now, we compute the true positive rate (also known as detection rate) and false positive rate for a sequence of arbitrary map thresholds varying from 0% to 100%. We can then plot the false positive rate against the detection rate for each of the map thresholds in the sequence to generate the curve shown in in figure 15. This is the ROC and the area underneath the curve (AUC) is a metric used for quantifying the performance of the predictive model. A model that has an AUC of 1 always predicts correctly with respect to the testing set (this is the best score). A model with an AUC of 0 is one that always predicts the opposite of the testing set (this is still perfect performance, just inverted). A model with an AUC of 0.5 is randomly guessing with respect to the testing set (this is the worst score).

Figure 14: A diagram of the four possible outcomes of a model’s prediction tested against a ground truth.

Figure 13: A visual aid for computing ROC between a predictive model and testing data, a) a predictive model computed from a training set, b) the testing set used to evaluate the model’s predictions

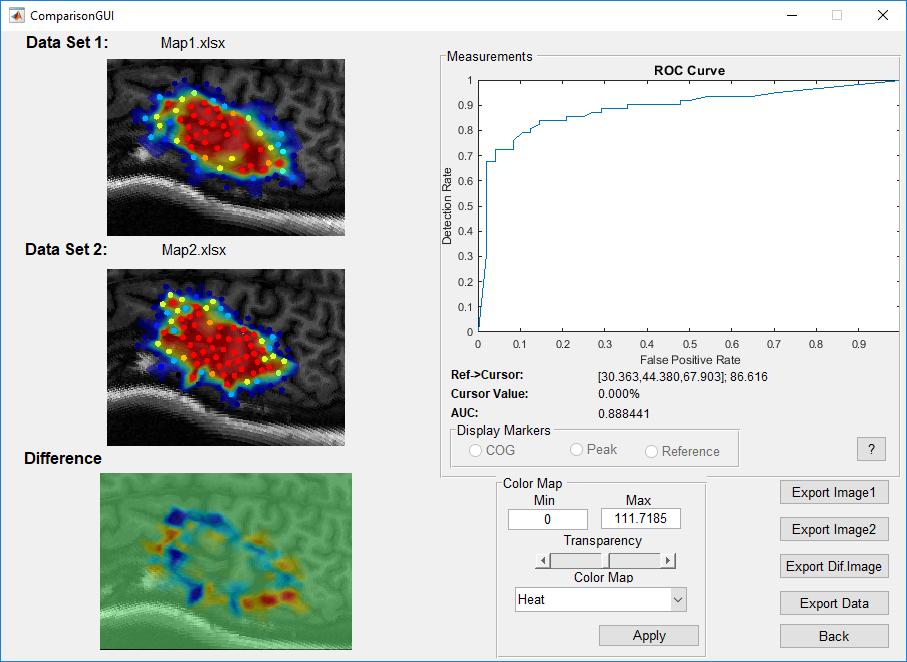
 In the case of motor mapping, the AUC is a good way of testing the stability of a motor map over time. By computing the AUC, we test the following hypothesis: How well does a motor map that predicts the likelihood of a TMS response above a specified value threshold perform when evaluated on test data collected at a different time point under different experimental conditions?[[2]](#footnote-2) The more consistent the testing set is with the training set, the closer the AUC will be to 1. Conversely, the more inconsistent the measurements are, the closer the AUC will be to 0.5. See the *Examples* section for a demonstration on how this functionality of NeuroMeasure can be used. Notably, the binarization threshold is an important part of this analysis, and there are currently no recommendations on how to use it. This functionality of NeuroMeasure at the time of publication is experimental and has no precedence in the literature, unlike the continuous comparison measurements.

Figure 15: An example of the comparison window launched in categorical comparison mode. Motor maps for Dataset 1 (training set) and Data set 2 (testing set) are displayed on the left, but the training map is the more important one as that is the predictive model being tested. The difference map is also displayed similar to continuous mode but it is not as useful and should largely be ignored. In this mode, the ROC replaces the measurement table. As described in the Compare section, the ROC is a plot of the true positive rate and false positive rate at computed at varying map thresholds (not to be confused for binarization threshold) from 0% to 100%. The ROC is useful for selecting a threshold at which the predictive model performs optimally and this is usually chosen to be the point with the highest acceptable false positive rate for the highest possible true positive rate. The AUC is an indicator of overall model performance and is a good metric for quantifying the predictive power of the first dataset for the second data set. The value of the motor maps will not exceed 100% so the colormap min and max should be set to 0 and 100 to get the full range of color representation.

Examples

## First-Sampling vs. Second-Sampling Comparison

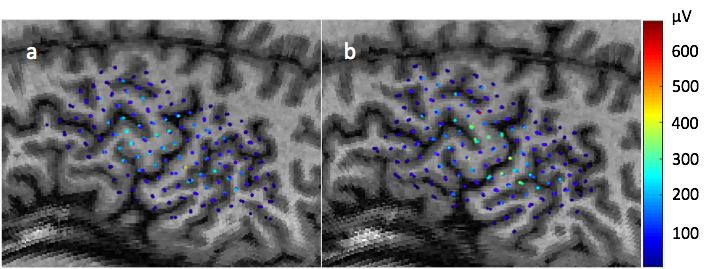
In this example, we test the predictive performance of different surface fitting algorithms fitted onto training data with a testing set that was collected immediately after the collection of the training set. The datasets used for this example are shown in figure 16. The data was collected with the Nexstim neuronavigation hardware/software. A subject’s motor evoked potentials were recorded following TMS stimulation in a 0.5x0.5 cm grid centered around a hotspot. Immediately afterwards, the same region was sampled again between the grid spaces. Three samples per location were collected.

Figure 16: Individual motor evoked potential amplitudes sampled three times per location in a 0.5x0.5cm grid. Color represents peak-to-peak MEP amplitude in μV. Recorded from the FDI muscle with TMS stimuli delivered over the dominant (left) hemisphere of a healthy female adult. Data registered to MRI using Nexstim neuronavigation system. a) first sampling of the left hemisphere, b) second sampling of the left hemisphere immediately after completion of the first.

The data used for this can be downloaded from our Github at the specified link. In the Data folder, find the scan labeled “Fig9 MRI” and upload it to NeuroMeasure. Then in the same folder containing the MRI slices, find the data files called Fig9-N-Square.xlsx and Fig9-N-Intersection.xlsx and upload both to NeuroMeasure. In both cases, select the Nexstim option in the data import window and select MRI from the dropdown menu (See Data Import and Scan Import sections for further details). Now, use the clustering panel to cluster both sets of data; 0.02 is a good threshold to group the repeated measurements. Once both have been clustered, go to the Fig9-N-Square tab, enter *Measurement Mode*, and use the Surface Fitting panel to generate a motor map.

First, choose the “average” clustering operation, and the “Piecewise Cubic” surface fitting algorithm, and then click *Apply*. This will generate a motor map much like the one illustrated in figure 20. Now, click on the *Compare* button. Since both data tabs have been clustered, a popup will appear asking the user to select which dataset to compare; select the only available option, which should be the Fig9-N-Intersection tab. The comparison window will then be launched as shown in figure 12 in the continuous data comparison mode due to the clustering operation being “average” (had the user chosen probability, the comparison window would be launched in categorical mode). In this window, the Fig9-N-Square is shown in the Data Set 1 window, Fig9-N-Intersection is shown in the Data Set 2 window and the difference between the two is shown in the Difference window. Measurements of COG and peak position and value can be found in the tables and can be exported as a text file using the Data export button. Finally, the root mean square error between the map of Data Set 1 and the raw data from Data Set 2 is reported. Recall that the two datasets were collected from the same patient with little latency between the two procedures, so we expect the maps to be similar. Thus, the RMSE is expected to be low.

Now exit the comparison window by clicking on the *Back* button or closing the window to return to the main window. From here, select the clustering operation to be “probability” and set the binary threshold to 75μV. Then, reapply the surface fit with using the “Piecewise Cubic” fitting algorithm. Now, click on *Compare* to re-launch the comparison window, this time in categorical mode. Notice that the measurements table has been replaced with a graph of the ROC and the RMSE now reports AUC instead. Exporting data will now export the true positive and false positive rata of the ROC at every threshold into a text file. Since the two maps under comparison are expected to be similar, we expect that the predictions of a motor map fitted onto the first dataset will correspond with the values of the second dataset, so our AUC value should indicate good performance (close to 1). The ROC will be close to a square curve because there is a (or are several) thresholds that create optimal performance.

Now exit back to the main window and try changing the fitting algorithm or clustering operation and see how they change the results. A detailed description of the results of this experiment can be found in the Methods paper accessible on our Github.

## Pre-Fatigue vs. Post-Fatigue Comparison

This example is very similar to the previous example (First Sampling vs. Second Sampling Comparison), with the exception that we will use data from a different experiment. In this experiment, the motor cortex was sampled in a 1x1 cm grid using the BrainSight system, prior to a hand fatigue exercise (pre), then sampled again immediately afterwards (post), and then sampled again 60 minutes after that (post+60). The data is displayed in figure 17. This technique is well documented in the literature to change the excitability of the motor cortex and we expect that the pre vs post motor maps will be less similar than the pre vs post+60 motor maps.

The data used for this example can be downloaded from our Github at the specified link. In the Data folder, find the scan labeled “Fig10 MRI” and upload it to NeuroMeasure. Then in the same folder containing the MRI slices, find the data files called Fig10-B-Pre.xlsx, Fig10-B-Post.xlsx, and Fig10-B-Post+60.xlsx and upload them to NeuroMeasure by selecting the BrainSight option in the Data Import window, followed by “BrainSight” from the dropdown menu. Now, use the clustering panel to cluster all sets of data; 0.02 is a good threshold to group the repeated measurements. Once both have been clustered, go to the Fig10-B-Pre tab, enter *Measurement Mode*, and use the Surface Fitting panel to generate a motor map as demonstrated in the previous example.

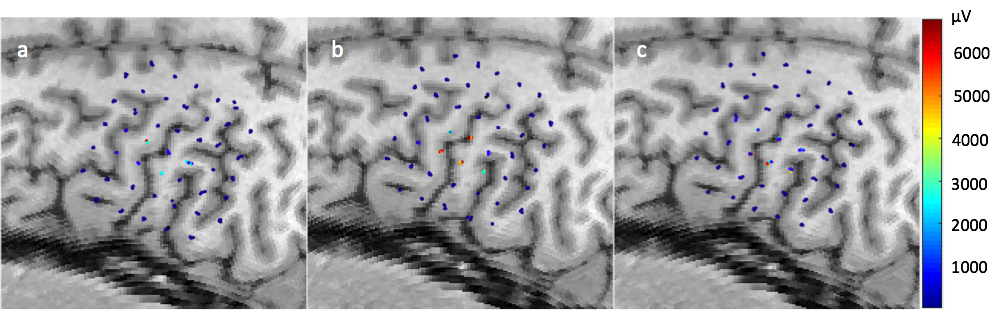
Click on *Compare* and first select the Fig10-B-Post tab for comparison. From the comparison window export the data and save it to any directory. Repeat this now for the DE-B-Post+60 tab. Open both exported documents and note the differences: The COG and peak positions are closer together in pre vs post+60 then in pre vs post and the RMSE for pre vs post+60 is lower than that of pre vs post indicating that the similarity between pre vs post+60 is higher. Now repeat this for the probability setting using a binarization threshold of 2700μV. Notice that the AUC for pre vs post+60 is closer to 1 than pre vs post, indicating that “pre” predicts the post+60 data better than post, as expected. Repeat this experiment using different surface fitting algorithms to see how they change the outcome, or see the methods paper available in the Github download to see the results of this experiment and a discussion of the data.

Figure 17: Individual motor evoked potential amplitudes sampled four times per location in a 1x1 cm grid. Color represents peak-to-peak MEP amplitude in μV. Recorded from the FDI muscle with TMS stimuli delivered over the dominant (left) hemisphere of a healthy male adult. Data registered to MRI using BrainSight neuronavigation system. a) Sampling prior to hand fatigue exercise, v) sampling immediately after hand fatigue exercise, c) sampling 60 minutes after hand fatigue exercise.

## Left vs. Right Hemisphere Comparison

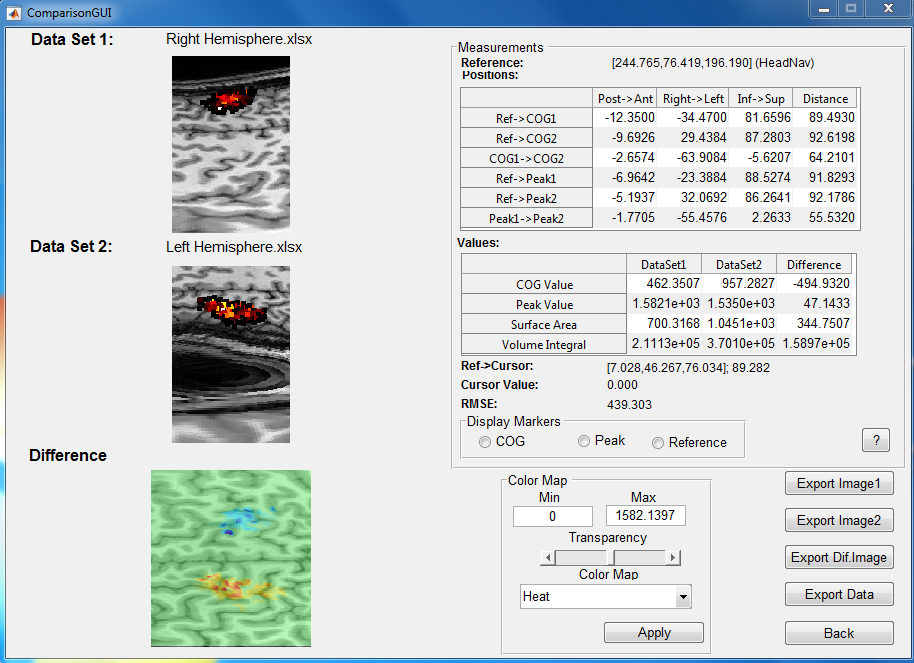
 In this example we demonstrate NeuroMeasure’s capability of comparing motor maps collected on the right and left hemisphere of the same subject and comparing the map characteristics between them. The data used in this example can be found in the Github within the Data folder. Begin by uploading “Fig11 MRI” by following the standard procedure outline in the previous sections and importing both datasets within the JSScan folder labeled Fig11-N-Left.xlsx and Fig11-N-Right.xlsx respectively. In the data import window, select the Nexstim option and “MRI” from the dropdown menu for both excel files. Now cluster both datasets; since this is a single-measurement-per-location experiment, simply enter 0 as the threshold and confirm the clustering. From either tab, set the *Clustering Operation* to “average” and *Fitting Algorithm* to “Piecewise Cubic” and generate a motor map. Now click on the *Compare* button and select the other tab to launch the comparison window as described in earlier examples or in the Compare section of this manual.

Figure 18: An example of the comparison window when launched to compare data sets on the left and right hemisphere. The scan used here is an anisotropic scan whose MRI is poorly sliced for reasons discussed in the Topographic Display section of this manual. When the motor maps are not overlapping the difference map subtracts non-zero values from a field of zeros to produce the exampled display. RMSE, in the absence of likewise values, simply represents the average value of data set 1 (in equation 10, set Oi to 0 and see how the equation transforms into the formula for mean). The data table will display useful comparative measurements. Some example observations for this dataset is that the right hemisphere’s COG is 1mm posterior of the left hemisphere’s COG w.r.t the reference point and the surface area of the left hemisphere is 141mm^2 larger than the right hemisphere.

The resulting comparison window is shown in figure 18. Here we can compare the positions of the peak and COG in the three anatomical axis as well as the value of the motor map at those points. We can also compare the surface area and volume integral between the left and right hemisphere. **Because RMSE compares values at corresponding positions and the maps of the left and right hemisphere have no points in common, RMSE has no value in this kind of comparison and thus should be disregarded. On a similar note, probabilistic comparison using an ROC curve is also meaningless when the two maps have no common points as well.** **The only useful metrics in this analysis are those in the data table and the cursor tracked values.**

## Cerebral Palsy Peak Discretization

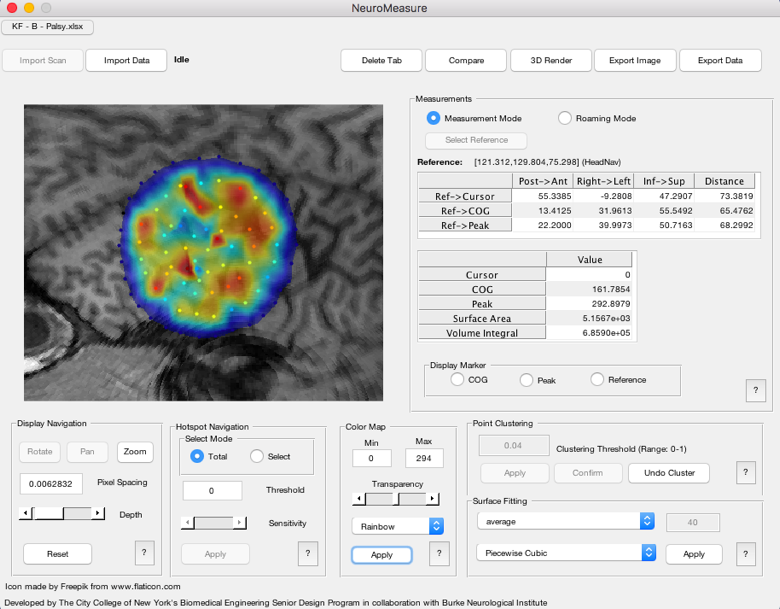
 Sometimes, motor mapping data from subjects with neurologic conditions can yield uncharacteristic data like that shown in figure 19. This dataset comes from a cerebral palsy patient. Note that this dataset contains many individual “hotspot” peaks. In some experimental protocols or data analysis regimes, it may be useful to seperate, or discretize the motor map into individual peaks and calculate characteristic measurements for those portions alone. With this in mind, NeuroMeasure comes with an automated peak discretization package capable of doing so.

Figure 19: The NeuroMeasure application window showing an example of a cerebral palsy motor map that has been clustered and surface fitted. Note in the *Hotspot Navigation* panel, that the *Select Mode* is currently set to *Total* in which the displayed measurements are computed using all of the available motor mapping data.

The data from this example can be found in the Github under the Data folder. Find the scan labeled “Fig12 MRI” and upload it to NeuroMeasure. Then, upload the data spreadsheet “Fig12-B-Palsy.xlsx” in the same folder by selecting the BrainSight option in Data Import window and selecting Brainsight from the dropdown menu. Now, enter *Measurement Mode*, cluster the data with a threshold of 0.04 and generate a motor map with the “average” *Clustering Operation* and “Piecewise Cubic” fitting algorithm.

Note that *Select Mode* in the *Hotspot Navigation* panel is currently set to *Total*. This means that the measurements displayed in the *Measurements* panel were computed from all of the available motor mapping data. However, notice in figure 19 how the cerebral palsy map shows many more discrete peaks than is characteristic for a normal motor cortex. NeuroMeasure is equipped with the functionality to separate and measure those peaks individually, for this we switch from *Total* mode to *Select* mode. Upon entering *Select* mode, the *Measurements* panel will be cleared of all values and a flood fill algorithm (See the Select Mode section for further details) will separate the peaks. Now, simply click on the display of the motor map and the selected peak will be highlighted by a white border as shown in figure 20. Notably, the values in the *Measurements* panel will update to those computed only from the data within the white border. For example, the COG in figure 20 is computed only from the 7 data points within the white selection border of the selected hotspot. Enabling markers in the *Display Markers* panel will show the COG and peak position, which will update live with hotspot selection. The hotspot selection can be adjusted with the sensitivity slider at the bottom of the *Hotspot Navigation* panel. Reducing the sensitivity of the selection reduces the distance threshold for which two hotspot peaks would be considered part of the same hotspot. Thus, the sensitivity slider can be used to merge hotspots that are close to each other and refine the hotspot selection. Change the slider settings and click *Apply* to lock in the setting.

Figure 20: *Select* mode is now enabled allowing for the selection and measurement of individual peaks. Clicking on the display will show a white border that highlights the selected peak and the *Measurements* panel shows values computed only from the data points within the white border. If either COG or peak are enabled in the *Display Marker* panel, the marker will update appropriately.

Acknowledgements

This project was made possible by the collaboration between The City College of New York’s Biomedical Engineering Senior Design Program and Burke Neurological Institute. We would like to thank our City College mentors Dr. Luis Cardoso, Dr. Sihong Wang and Dr. Steven Nicoll for project supervision. We would also like to thank the Burke team Dr. Dylan Edwards, Dr. Kathleen Friel, Dr. Taiza Grespan Santos, and Josh Silverstein for guidance, data and testing.

References

Arya, B. K., M, M., Kishore, A., Velayudhan, B. (2010). Technique of Topographical Mapping of FDI Muscle Response after Single Pulse TMS Stimulation. Proceedings of 2010 International Conference on Systems in Medicine and Biology, 1-6.

Darling, W. G., Wolf, S. L., & Butler, A. J. (2006). Variability of motor potentials evoked by transcranial magnetic stimulation depends on muscle activation. *Experimental Brain Research*, 376-385.

Fitzgerald, P. B., Fountain, S., & Daskalakis, Z. J. (2006). A comprehensive review of the effects of rTMS on motor cortical excitability and inhibition. *Clinical Neurophysiology*, 2584-2596.

Kallioniemi, E., Pitkanen, M., Saisanen, L., & Julkunen, P. (2015). Onset Latency of Motor Evoked Potentials in Motor Cortical Mapping with Neuronavigated Transcranial Magnetic Stimulation. *The Open Neurology Journal*, 62-69.

Kiers, L., Cros, D., Chiappa, K., & Fang, J. (1993). Variability of motor potentials evoked by transcranial magnetic stimulation. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 415-423.

Kreig, M. S., et al. Predictive Value and Safety of Intraoperative Neurophysiological Monitoring with Motor Evoked Potentials in Glioma Surgery. Neurosurgery (2012) 70:1060-1071

Kleim J. A., Kleim E. D., Cramer S. C., Systematic assessment of training-induced changes in corticospinal output to hand using frameless stereotaxic transcranial magnetic stimulation. Nature Protocols (2007) 2: 1675-1684

Littmann A. E., McHenry C. L., Shields R. K., Variability of motor cortex excitability using a novel mapping procedure. J Neurosci Methods (2014) 214(2): 137-143

Mills, R., K., Nithi, A., K., (1996). Corticomotor Threshold to Magnetic Stimulation: Normal Values and Repeatability. *Clinical Neurophysiology*, 570-576

Pascual-Leone A., Tarazona F., Keenan J., Tormos J., M., Hamilton R., Catala M., D., (1999). Transcranial magnetic stimulation and neuroplasticity. *Neuropsychologia*, 37(2): 207-217

Saito, T., Muragaki, Y., Maruyama, T., Tamura, M., Nitta, M., & Okada, Y. (2015). Intraoperative Functional Mapping and Monitoring during Glioma Surgery. Neurologia Medico-Chirurgica, 1-13.

Thickbroom, G., Byrnes, M., & Mastaglia, F. (1998). A model of the effect of MEP amplitude variation on the accuracy. *Clinical Neurophysiology*, 941-943.

1. As a side note, importing a scan prior to importing motor mapping data is mandatory for this reason. The head centroid is a critical step in our 3D to 2D conversion process and it would be undesirable to have different cases depending on whether the scan was available or not as that would un-standardize the surface fitting as well as any measurements that depend on it like *surface area* and *volume integral*. [↑](#footnote-ref-1)
2. The term “different experimental conditions” is non-specific to imply that this workflow can be applied to any analysis where the stability of the TMS evoked readings from the motor cortex is a measurement of interest. [↑](#footnote-ref-2)