Regional extraction of activation functions using anatomical MRI

Standard Operating Procedure

Updated 10/9/2022

Contact: Sky Jones, rsj44@case.edu

# Overview

This document details how to use a series of Python scripts (“the program”) to extract activation functions due to intracranial voltage distributions induced deep brain stimulation (DBS) along specific neural tracts.

Please note that the program is in a development state: there may be unknown bugs in its implementation, and there are certain limitations to its use. Most pressingly, the program analyzes data in MNI152 nonlinear space. This results in inaccuracies of lead position estimation due to nonlinear warping in addition to inaccurate tissue segmentation (atlas tissues are used for modeling rather than patient tissues). Additionally, anisotropic modeling of tissue conductivities is not yet implemented. Future implementations will analyze activations in native patient space and use anisotropic tissue conductivities derived from diffusion tensor imaging. Currently, the only DBS lead supported is the Medtronic B33005.

Execution of the program requires using several static assets as well as specifically formatted inputs. Most but not all of these static assets are contained within this repository. These additional assets and inputs as well as the outputs of the program are detailed below. Additionally, you will need to install Anaconda (a scientific installation of Python), *pip* install the Python package *nibabel*, and download Sim4Life.

Additional assets:

* 1. voltage\_modeling.smash – a Sim4Life template simulation that contains predefined brain tissues in MNI152 nonlinear space

Inputs:

1. LEAD-DBS patient folder (your/path/pt\_folder/) – this is the output folder produced by lead localization using DBS. You should also “export” the localized lead and anatomy as .ply files
2. your/path/pt\_folder/spectmean\_clean.xlsx – an excel file containing beta wave readings for each activation condition. Should contains the following columns:
   * active\_contacts – 0 indexed contacts active during that condition. If multiple, separate by commas
   * bounds – the voltage of the active contacts
   * recording contacts – the inactive contacts
   * high\_beta – the high beta band value
   * low\_beta – the low beta band value

An example file would be as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **active\_contacts** | **bounds** | **recording\_contacts** | **high\_beta** | **low\_beta** |
| 1 | 1.5 | 4,8 | 0.331 | 0.375 |
| 2 | 1.5 | 4,8 | 0.222 | 0.553 |
| 3 | 1.5 | 4,8 | 0.321 | 0.492 |
| 5 | 1.5 | 4,8 | 0.229 | 0.560 |
| 6 | 1.5 | 4,8 | 0.228 | 0.307 |
| 7 | 1.5 | 4,8 | 0.272 | 0.873 |
| 5,6,7 | 1.5 | 4,8 | 0.234 | 0.745 |

Output:

1. your/path/pt\_folder/voltage\_maps/ – a folder containing the voltage distribution for each contact as a .mat file and a regularized NiFTi
2. your/path/pt\_folder/voltage\_maps/tractographic\_activation/ – a folder containing the activation for each neural tract as an excel file as well as PDF visualizations of the activation
3. your/path/pt\_folder/voltage\_maps/tractographic\_activation/correlation/ – a folder containing visualizations of simple bivariate correlation analyses

# Full procedure for executing the program

Execution of the program is still a fairly manual process. It can be roughly broken into the following steps:

1. Setup –placing beta band Excel file and target neural tracts within your LEAD-DBS patient folder
2. Voltage distribution simulation – parameterize Sim4Life template simulation and extract voltage distributions
3. Extraction of neural activation – run Python scripts to evaluate neural activation along target neural tracts

Full details on executing the program are given below.

1. **Setup**
   1. Localize your lead using LEAD-DBS. Make note of where the resulting patient folder is (your/path/pt\_folder/)
   2. Export your lead using LEAD-DBS as a .ply file; this will create a subfolder in your patient folder called /export/ply
   3. Make a copy of voltage\_modeling.smash in your patient folder
   4. If it does not exist already, create a subfolder within your patient folder called /atlases/. Create a subfolder within this folder for your neural tracts of interest, e.g., /dbs\_tractography\_atlas\_middlebrooks\_2020/. Within this folder create another folder for each hemisphere containing the coordinates for each neural tract of interest as a Microsoft access table. Each table should be named after the tract it represents. The directory structure should look like this:

*your/path/pt\_folder/atlases/dbs\_tractography\_atlas\_middlebrooks\_2020/lh*

This repository contains the Middlebrooks tractography atlas for each hemisphere in the assets folder.

1. **Voltage distribution simulation – note that all Python scripts in this section will be executed in the Sim4Life scripting window**
   1. Open voltage\_modeling.smash. Import your .ply file representing the lead of interest using the import tool
   2. Select the lead in the model window. Use the Mesh Tools > Separate Meshes tool to explode the lead into its constituent part.
   3. Open the Python scripting window in Sim4Life (View > Scripter) and open *prep\_lead.py*. If needed, change lead\_name within the script to match the name of the unexploded lead. Execute this script. This will name and set the parameters for each component of the lead. Ensure that the lead positioning and parameterization is correct.
   4. Open *extract\_all\_voltage\_maps\_from\_sim.py* in the scripting window. Change the project\_folder parameter to be your *your/path/pt\_folder/*. Execute the script. This will create the folder *your/path/pt\_folder/voltage\_maps* and populate it with .mat files representing the voltage distribution for each contact if they were activated one at a time. This may take some time to run (~20 minutes).
2. **Extraction of neural activation– note that all Python scripts in this section will be executed using an Anaconda environment using your own IDE**
   1. Open the file *convert\_vmaps\_to\_nifti.py* in the IDE of your choice (Spyder is convenient). Change the variable vmap\_folder to *your/path/pt\_folder/voltage\_maps*. If necessary, change template\_image to point to the MNI nonlinear T1 NiFTi. This is available within the repository in the assets folder.
   2. Execute *convert\_vmaps\_to\_nifti.py*. This will convert each .mat voltage map to a NiFTi.
   3. Open *extract\_voltages\_along\_tract.py*. Change vmap\_folder to *your/path/pt\_folder/voltage\_maps*. Change tract\_folder to *your/path/pt\_folder/atlases/your\_tractography\_folder*.
   4. Execute *extract\_voltages\_along\_tract.py*. This will create a subfolder in the voltage maps folder called */tractographic\_activation/* and fill it with Excel sheets detailing the activation functions within each tract. This script takes several hours to run depending on how many tracts you are analyzing and the number of nodes within each tract.
   5. Open *visualize\_activations.py*. Change parent\_folder to *your/path/pt\_folder/voltage\_maps/tractographic\_activation.* Change conditions\_file to *your/path/pt\_folder/spectmean\_clean.xlsx.* Change anatomy file to your exported anatomy .ply file. Change lead\_file to your exported lead .ply file.
   6. Execute *visualize\_activations.py.* This will create visualizations of each tract’s activation with the *tractographic\_activation* subfolder.
   7. Open *collate\_correlation.py*. Change parent\_folder to *your/path/pt\_folder/voltage\_maps/tractographic\_activation.* Change conditions\_file to *your/path/pt\_folder/spectmean\_clean.xlsx*
   8. Execute *collate\_correlation.py.* This will create a subfolder in the tractographic\_*activation* folder called */correlation/* and fill it with plots beta oscillation intensities vs. measures of tractographic activation.