Utilizing the FLIP Algorithm

As mentioned previously, the FLIP algorithm has practical applications for analysis of both the quality of the data collected and the spatial location of the probe. The following examples will explore the different functionalities of FLIP and explain the interpretation of the results.

Before Calling the FLIP Algorithm: Required Inputs

Non-Normalized Power Matrix

The first parameter required to call the FLIP algorithm is the non-normalized power matrix the user would like to analyze. In our experimental analysis, we utilized the FieldTrip MATLAB toolbox to compile FFT analysis information across different trials. Regardless of the method used to produce the matrix, there are basic requirements to be followed for successful implementation of the FLIP algorithm. The first dimension of the matrix must represent the spatial location of the contacts, starting from position 0. Position 0 on the first dimension should correspond to the contact furthest from the tip of the probe used during experimentation. Additionally, the second dimension of the matrix must represent the frequencies of interest used in the FFT analysis. There is no requirement for the ranges of frequencies that can be utilized or the frequency bin size used for analysis, as these will be passed to the FLIP algorithm separately. An example power matrix for 32 probe contacts with frequencies of interest 1-250 Hz (frequency bin size 1 Hz) would be a 32 x 250 matrix.

Laminar Axis

After effectively obtaining the non-normalized power matrix to be analyzed, the next parameter required for using the FLIP algorithm details the laminar axis of the data in mm. This should be passed as a row vector to the function. Requirements for successful implementation are that index 1 of the vector is 0, the values are evenly spaced, and the length matches the length of the second dimension of the power matrix. Additionally, this vector should be at minimum 0.6 mm long to fulfill the minimum length requirement of the FLIP algorithm. An example vector for a 32 channel probe with 100 micron spacing: 0:0.1:3.

Frequency Axis

The last user generated data structure required for the FLIP algorithm is the frequency axis of the power matrix. There are less restrictions on the frequency axis than the laminar axis to

allow for more flexibility for the user to analyze their data. However, basic requirements of the frequency axis are that the frequency bins are equally spaced, the frequencies are increasing, and the length of the frequency axis corresponds to the length of the first dimension of the power matrix. An example vector for a power matrix using frequencies 1-250 Hz with frequency bins of 1 Hz: 1:250.

Calling the FLIP Algorithm: 3 Options

Using Default Values:

The first option when utilizing the FLIP algorithm is to use the default frequency values empirically found in the manuscript to maximize goodness of fit value: 10-19 Hz for the alpha-beta range and 75-150 Hz for the gamma range. If the user desires to use these frequency windows in their analysis, the fourth parameter (boolean, 1 for FLIP, 0 for vFLIP) should be set to 1 to indicate that the FLIP algorithm should be used with default frequency bands.

No other user inputs are strictly required. The FLIP algorithm will then perform linear regressions on the alpha-beta and gamma traces for each possible range of channels, determining the best fit to the data.

Using User-Defined Values:

The second option when utilizing the FLIP algorithm is to use user-defined frequency windows for the algorithm. If the user would like to use frequency windows that differ from the default values and would still like to bypass the more computationally expensive vFLIP algorithm, the fourth parameter should still be set to 1. In addition to the fourth parameter, two 1x2 row vectors must be passed as the fifth and sixth parameters. The elements of the vectors represent the lower and upper bounds of the alpha-beta and gamma, respectively. For example, [1 30] for the fifth parameter and [100 140] for the sixth parameter.

The FLIP algorithm will run exactly as before, but instead evaluating the relative power of the alpha-beta and gamma traces within the user-defined frequency windows.

Finding Optimal Frequency Values:

The final option when utilizing the FLIP algorithm is to take full advantage of its functionality and employ the “vFLIP” algorithm. To use the vFLIP algorithm, the first three parameters remain the same. To specify the vFLIP algorithm, the fourth and final parameter should be set to 0. In this case, the algorithm looks at every possible combination of alpha-beta

and gamma frequency windows, performs a FLIP analysis on each, and pools the results in a table. After it has checked every possibility, it finds the combination that produced the highest goodness of fit value and returns the results to the user.

While this leads to the optimal alpha-beta and gamma windows and returns the best fit data, the major drawback of vFLIP is efficiency. Checking every frequency window and performing calculations on the data is a time-intensive process and the runtime increases drastically as the size of the matrix increases.

Generating the spectrolaminar pattern from example LFP data

The online data supplement includes example data from two laminar electrodes (data.example1_vlPFC_lfp and data.example2_7A_lfp). Examples 4 and 5 utilize the Fieldtrip toolbox (www.fieldtriptoolbox.org), which needs to be downloaded to run this part of the script. We utilize Fieldtrip functions in our example script (main_FLIP_script.m) to generate power from example data. In the script main_FLIP_script, thwo examples are provided (Examples 4 and 5). The provided function 'relpow_from_rawLFP' will use this provided LFP data to provide a normalized power 2-D array (a.k.a. relative power map or "relpow" map). This normalized power map can be plotted using the FLIPAnalysis function.

Plotting Sample Normalized Power Data

The online data supplement includes example datasets for normalized power (data.relpow). The first dimension of this fields represent individual probes. I.e., there are 942 probes in this example dataset. The second dimension represents the laminar axis (different electrode channels) and the third dimension represents frequencies (in this case 1 to 250 Hz). Using the provided script, any probe can be selected (e.g., probe #752) and plotted using FLIPAnalysis.

Generate Area-average Spectrolaminar Patterns

In Example 7, average spectrolaminar patterns can be generated. In the meta data (data.meta struct), each probe has an area label (data.meta.brain_area_num). For example, area MST has label "2". Using this as an index, all MST probes can be identified and an average spectrolaminar pattern can be created.

Plot Current Source Density (CSD)

The online data supplement also include example normalized current source density data for each probe (data.CSD). This field has dimensions probes x laminar axis x time (in ms). Any probe (e.g., probe #752), can be plotted using the Example 8 script.