2017

EEGapp Documentation

EEG ANALYSIS AND PROCESSING PIPELINE

DR. BLAIN-MORAES AND YACINE MAHDID

MCGILL UNIVERSITY | Montreal

EEG Analysis and Processing Pipeline

Table of content

EEG Analysis and Processing Pipeline

Overview

Goal of this program: Accurate analysis of electroencephalography (EEG) data requires detailed training and domain expertise. Some programs (e.g. eeglab, Brainstorm, BrainNet Viewer) have been built to facilitate basic EEG analysis for non-experts, but recent advances in analysis techniques are not included in these software packages. The goal of this pipeline is to automatize the analysis of EEG data in order to reduce the workload for experts and the possible errors that could arise while self-coding the analysis techniques.

Installation: To install the EEG pipeline simply double click on the Matlab package named EEG pipeline (fig. 1) and select install (fig. 2). In your Matlab App tab the EEG pipeline should now have appeared. Simply click on it to launch it.

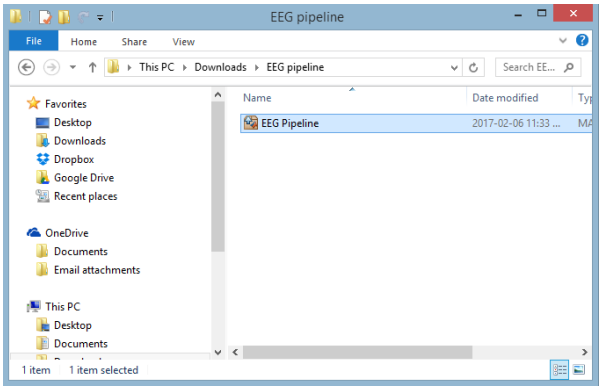

Figure 1: Double click on the Matlab package named EEG pipeline.

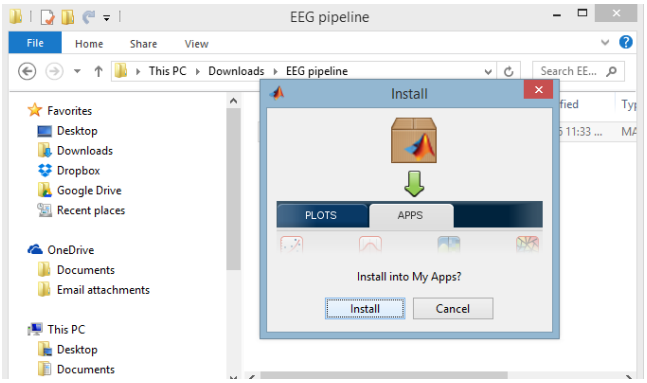

Figure 2: Select install to install the pipeline as a Matlab app.

How to use: This program require only minimal knowledge of Matlab. The basic manipulations the user needs to do are the following: feed the cleaned EEG data into the pipeline, select a saving directory, select the analysis techniques he wish to undertake and then enter the required variables for each techniques.

1. **Cleaning of EEG data in eeglab:** The cleaning of the EEG data need to be done in eeglab because the pipeline uses the EEG structure file made by exporting data from eeglab. For more information about eeglab follow the link provided below. Clean the EEG data until it is ready to be analyzed and then proceed to step 2.
2. **Export data as a Matlab structure file:** When your raw EEG has been cleaned enough to be analyzed go back to the main window in Matlab with eeglab still opened (i.e. see figure 3 below). In the workspace right click on the EEG structure file and select Save As… (fig. 4) . Select where you want to save the EEG structure file, give the data a name and select save (fig. 5). You are now ready to analyze your EEG data.

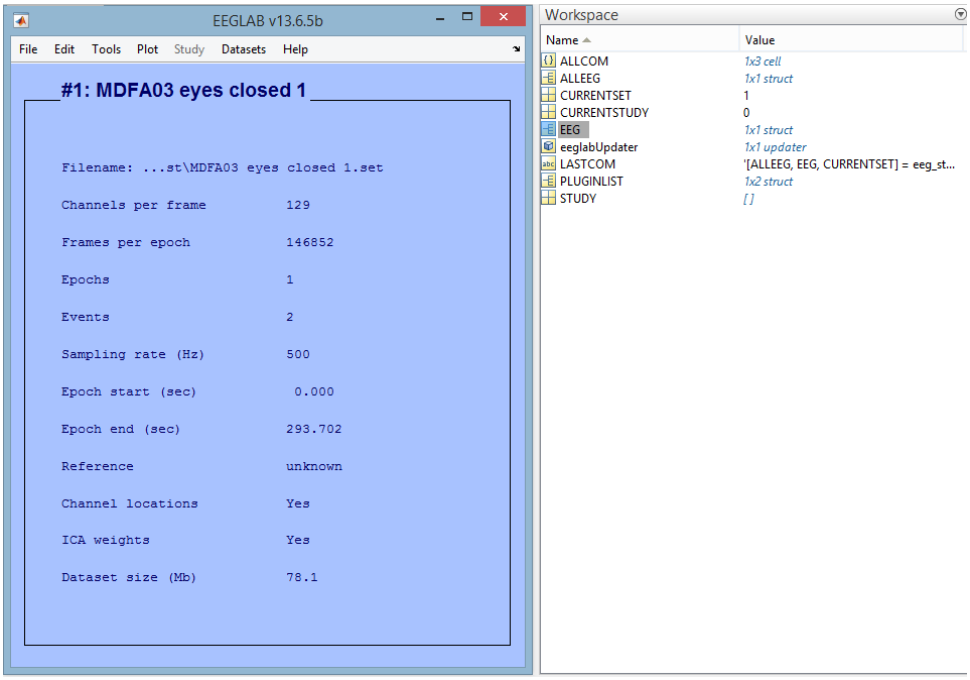

Figure 3: On the left is the eeglab window still open with the loaded cleaned EEG data set. On the left is the Matlab Workspace when eeglab is open.

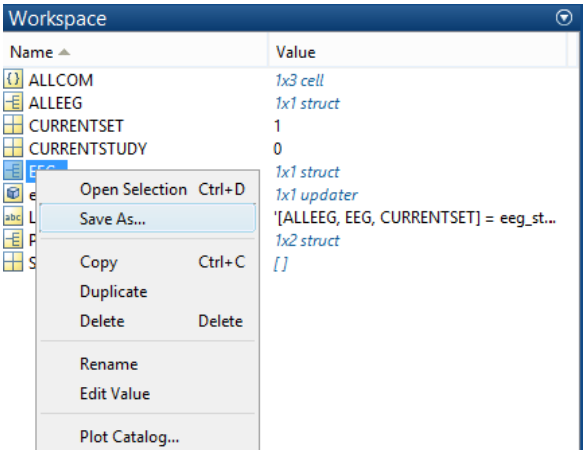

Figure 4: Right click on the EEG structure file and select Save As...

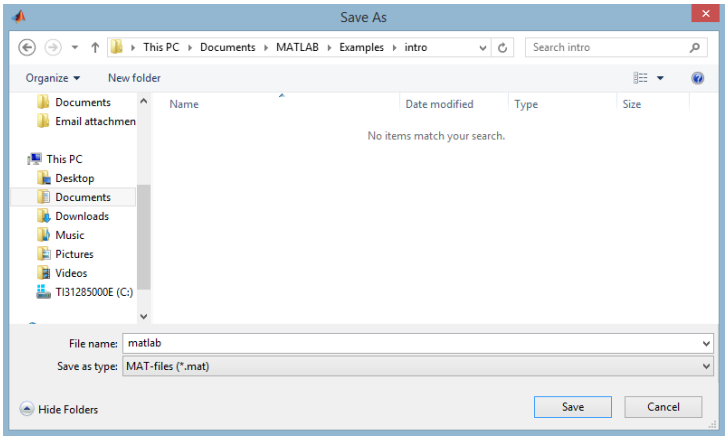

Figure 5: Save the EEG structure file somewhere you can access it easily. Give it a meaningful name.

eeglab Documentation:

Swartz Center for Computational Neuroscience eeglab: https://sccn.ucsd.edu/eeglab/

eeglab tutorials : https://sccn.ucsd.edu/wiki/Getting_Started

Contact Information:

By email at: Yacine.mahdid@mail.mcgill.ca

Main Window

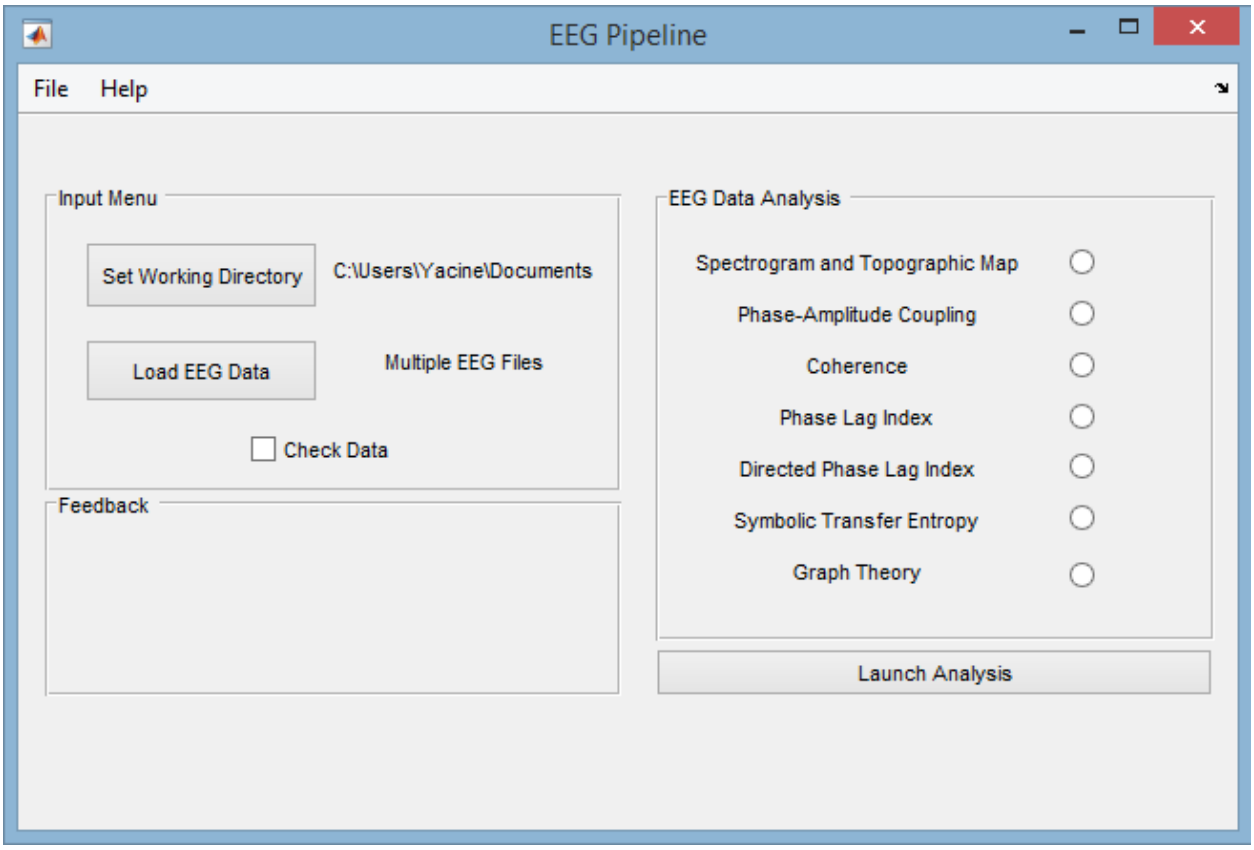

Figure 6: Main window of the EEG pipeline.

The Main window of the EEGapp Processing can be seen in figure 6. It is subdivided into three panels: Input Menu, Feedback and EEG Data Analysis Menu. The Main window is where the user will set its saving directory, load the cleaned EEG structure file from eeglab and select the analysis technique he wish to undertake.

Input Menu:

1. **Set working directory:** Before doing any kind of analysis you must first select where the data will be saved. To do so the user need to press the "Set Working Directory" button and select in which directory he wish to save its data. The result of the different analysis techniques will be saved in a main folder that correspond to the name given to the EEG sets in eeglab used to export the EEG structure files. Each EEG techniques will have its own folder inside the main folder (fig. 7).
2. **Load EEG data:** Then you need to select which EEG structure files to work with during an analysis session. You will need to click on the "Load EEG Data" button and load the EEG structure file that was previously saved from egglab (i.e. see page 2).

3. **Check data check box:** This check box must be select if you want to visualize your Spectrogram and topographic map before continuing the analysis with the other analysis techniques. When selected EEGapp will stop and a window will appear to ask the user he want to continue or abort the analysis (fig. 8). If continue is selected the pipeline will run each analysis without stopping. If abort is selected the pipeline will stop and return to the Main window.

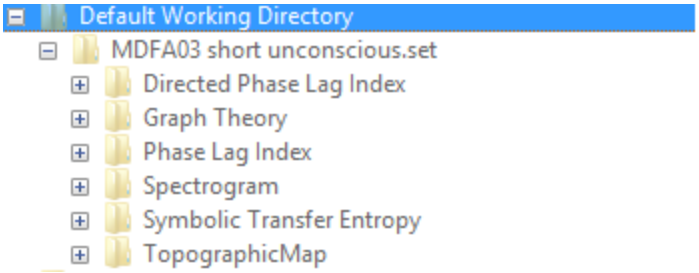

Figure 7: Here the working directory is called Default Working Directory. The pipeline created a folder inside that directory named after the name of the EEG data set in egglab. Inside that directory the analysis techniques are saved in different directories.

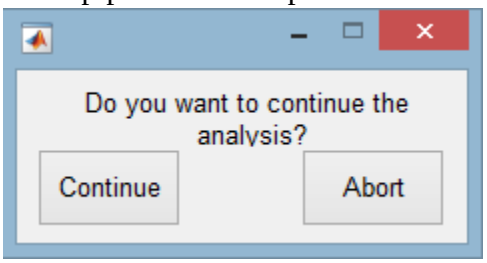

Figure 8: Windows that pops up when check Data is selected while the pipeline is running. This let you inspect the Spectrogram and the Topographic map before deciding to continue or not the rest of the analysis.

File tab:

The same functionality as in step 1 and 2 of Input Menu are available in the file tab.

Help tab:

The documentation is available under the help tab.

Feedback:

This is where EEGapp will give you feedback on what to do or give warnings if there is missing information.

EEG Data Analysis:

When you have selected in which directory you want to save your analysis result and which EEG structure file you want to analysis, you will need to select which analysis techniques to use. All the analysis currently supported by the pipeline are listed there, if you want to contribute to the pipeline and wish to include an analysis technique of your own please contact Yacine Mahdid at Yacine.mahdid@mail.mcgill.ca.

Launch Analysis:

After having selected every analysis you wish to run click on “Launch Analysis” and wait while the pipeline is running.

Spectrogram and Topographic Map

Spectrogram Menu:

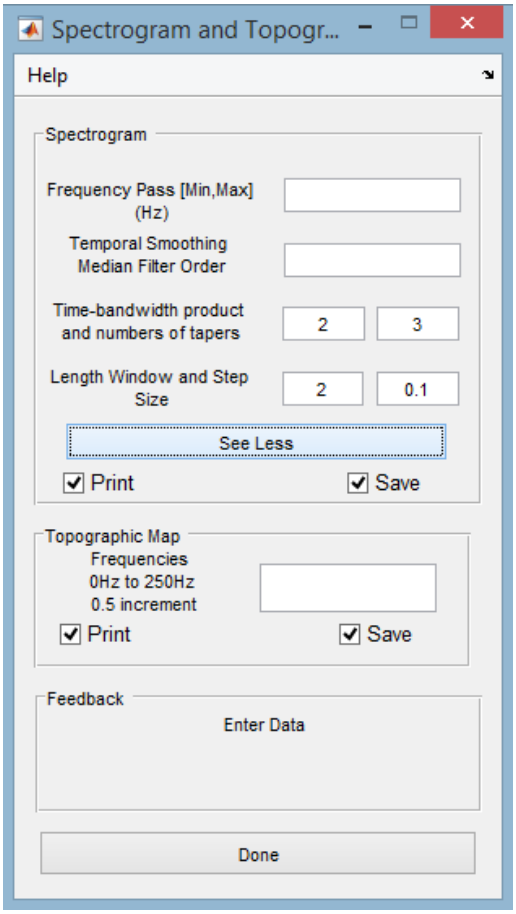

Figure 10: Spectrogram and Topographic map windows. This open when the radio button is selected in the Main window.

1. **Frequency Pass:** Select at which frequency you wish to filter your EEG data by giving two number one next to the other. For example : 10 30 , would filter the EEG data by letting only frequencies greater than 10Hz and frequencies lower than 30 Hz in. The input must be greater than 0 and two number one next to the other are required.
2. **Temporal Smoothing Median Filter Order:** Select the order of your median filter in order to reduce the noise in the spectrogram.
3. **Time-Bandwidth product and Numbers of Tapers:** You should generally set the number of tapers to be 2 times the time-bandwidth product − 1, as using more tapers will include tapers with poor concentration in the specified frequency bandwidth (see the Chronux documentation for a more information[1]).
4. **Length of Window:** The length of the windows over which the spectrum will be calculated.
5. **Step Size:** The step at which to move the previously defined window to calculate the spectrogram.
6. **Print check box:** By selecting print, when the pipeline will have finished computing the spectrogram it will output it as a figure to the screen (fig. 11).
7. **Save check box:** By selecting save the pipeline will save the file in the Spectrogram and Topographic map folder under a name that reflect the time at which the analysis was performed (fig. 12). Also, a text file containing the input will be saved using the same name as the figure.

Example of input:

Figure 11 was made by inputting the following:
Frequency Pass = [10 30], Temporal Smoothing Median Filter Order = 10, Time-Bandwidth product, Numbers of Tapers, Length of Window and Step Size were left at default.

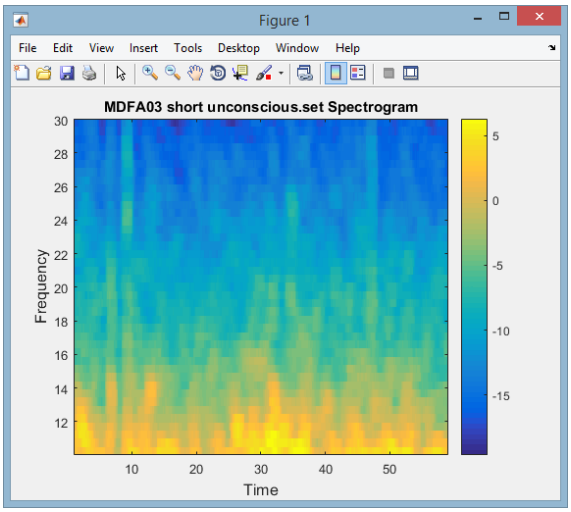

Figure 11: Example of the output of the spectrogram analysis when print is selected.

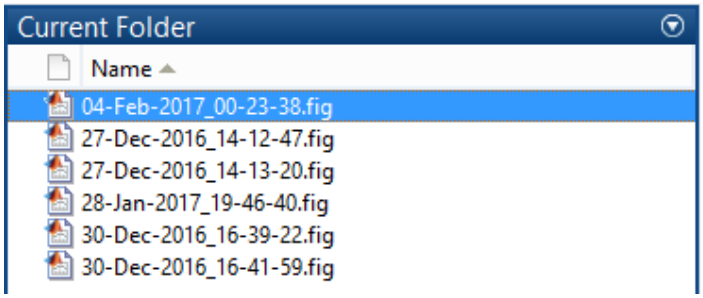

Figure 12: If the saved option was selected then the spectrogram is saved under the workingDirectory/EEGName/Spectrogram . The name is generate as follows : day/month/year_time.fig . Feel free to edit the name of the spectrogram to a more meaningful one.

Topographic Map:

1. **Frequencies:** For the topographic map, you need to enter up to 4 frequencies at which you want to visualize the topographic map. If more frequencies are inputted only the first 4 will be outputted. Only 0.5 increment are allowed, the frequency will be rounded if not written properly. The frequencies allowed range from 0Hz to the sampling rate at which the EEG signal were collected divided by 2 (i.e. Nyquist frequency). Example of correct input: 10Hz 24.5Hz 245Hz 123.5Hz.
2. **Print check box:** If selected the pipeline will output the topographic map as soon as it is done computing it (fig. 13).
3. **Save check box:** If selected it will save it in the topographic map folder in the same format as mentioned in figure 12.

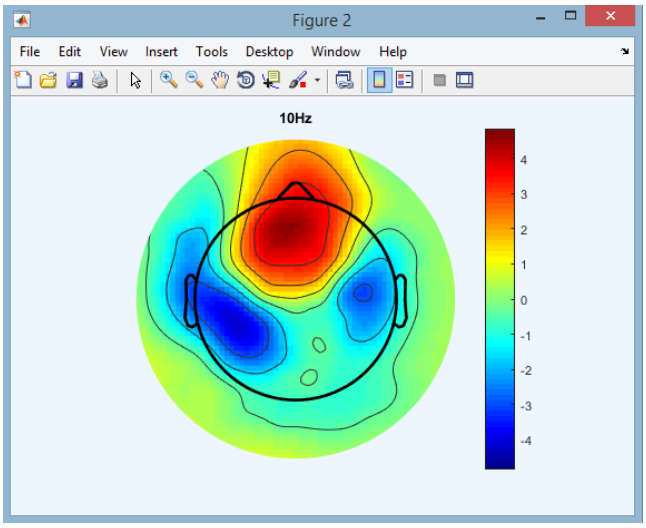

Figure 13: Example of a topographic map outputted when the print option was selected. The frequency of the topographic map is in the title right above the map.

Feedback:

Like in the main window the feedback panel will output warnings if illegal inputs are given to the pipeline.

Done Button:

When every variable has been entered click the “Done” button to go back to the main window.

Phase-Amplitude Coupling

Phase-Amplitude Coupling Menu:

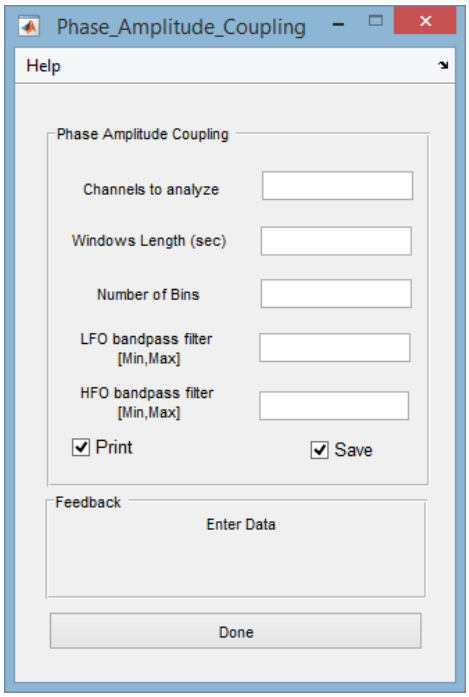

Figure 14: Phase amplitude coupling pop up window that appear when this analysis technique is selected.

1. **Channels to analyze:** Select which sets of channels you which to do the PAC analysis on. The channels can be inputted in sequence like this: 1 2 34 5 32.
   Or you can use the Matlab shorthand to create a vector,
   for example: 1:129. This is equivalent to listing all the channels from 1 to 129.
2. **Windows length:** The windows length is used to calculate the number of segment in the PAC plot. For example, in an EEG with 60seconds of data, if we choose windows length to be equal to 10 we will have 6 segments.
3. **Number of bins:** The number of bins is used to sort the amplitude per phase.
4. **LFO bandpass filter:** The bandpass filter is used to filter to data to obtain the low frequency component.
5. **HFO bandpass filter:** This will filter the data to obtain the high frequency component. This bandpass filter need to be greater than the LFO bandpass filter.
6. **Print check box:** By selecting print, when the pipeline will have finished computing the PAC plot will output as a figure to the screen (fig. 15).
7. **Save check box:** By selecting save the PAC plot will be saved in the Phase Amplitude Coupling folder under a name that reflect the time at which the analysis was performed. Also, a text file containing the input will be saved using the same name as the figure.

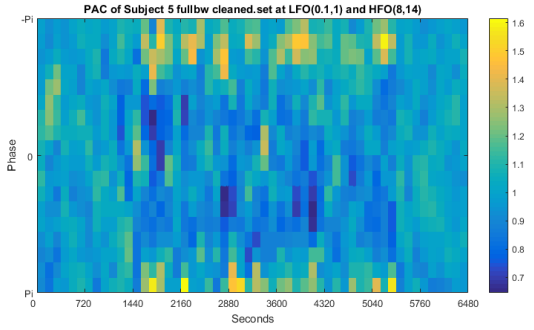

Figure 15: Example of an output for phase amplitude coupling with LFO bandpass = (0.1Hz,1Hz) and HFO bandpass = (8Hz,14Hz).

Example of Input:

Figure 15 was made by inputting the following:
Channels to analyze: 10 5 6 8 2 3, Windows length: 120, Number of bins: 18, LFO bandpass filter: 0.1 1, HFO bandpass filter: 8 14.

Coherence

Coherence Menu:

1. **Channels set 1:** One channels set to do the coherence with. You can input each channel you want manually like this: 1 3 45 67. Or you can use the Matlab shorthand: 1:129. This is equivalent to manually inputting every channel from 1 to 129.
2. **Channels set 2:** The other set of channels with which you want to do the coherence analysis with.
3. **Print Checkbox:** Will print the figure to the screen.
4. **Save Checkbox:** Will save the figure in the saving directory under the coherence sub folder. Also, a text file containing the input will be saved using the same name as the figure.

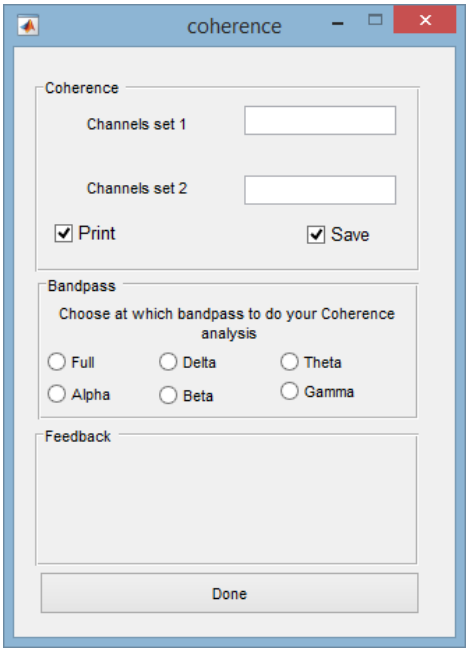

Figure 16: The pop up window that will appear when coherence is selected.

Bandpass Menu:

In the bandpass menu, you can select at which frequency you want to filter your EEG data to do the coherence analysis. If you select more than one frequency pass, the analysis will be repeated for each bandpass,

1. **Full:** Full band goes from 1Hz to 50Hz.
2. **Delta:** Delta band goes from 1Hz to 4Hz.
3. **Theta:** Theta band goes from 4Hz to 8Hz.
4. **Alpha:** Alpha band goes from 8Hz to 13Hz.
5. **Beta:** Beta band goes from 13Hz to 30Hz.
6. **Gamma:** Gamma band goes from 30Hz to 50Hz.

Example of Input:

Figure 17 was made by inputting the following: Channels set 1: 1:20, Channels set 2: 1:20, Bandpass: Alpha.

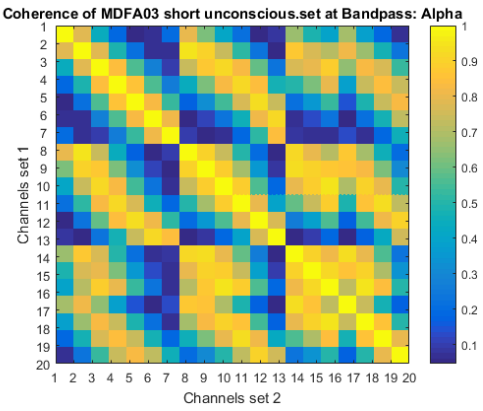

Figure 17: An example of an output for the coherence with channels 1 to 20 for set 1 and 2.

Phase Lag Index

Phase Lag Index Menu:

1. **Length of Analysis Segment:** The whole length of the EEG data will be cut into segments and the PLI will be calculated for each segment before getting averaged. The length in seconds need to be between 0 and the length of the EEG data.
2. **Number of Permutations:** The number of permutation correspond to the number of time the surrogate will be calculated. The number of permutations need to be greater than 0.
3. **p Value for Surrogate Analysis:** The p value will be used in the surrogate analysis to assess whether a given PLI is significant or no.
4. **Print check box:** If this option is check the pipeline will output a PLI plot at the end of the analysis (fig. 19).
5. **Save check box:** If select the PLI will be saved in the Phase Lag Index directory. Also, a text file containing the input will be saved using the same name as the figure.
6. **Reorder EEG data with custom order check box:** If selected a pop-up window will appear where you will be able to enter the order in which you want each channel to appear (fig. 20). The default order is from channel 1 to the last channel. Make sure that there is no duplicate and that there is the same number of channels as before.

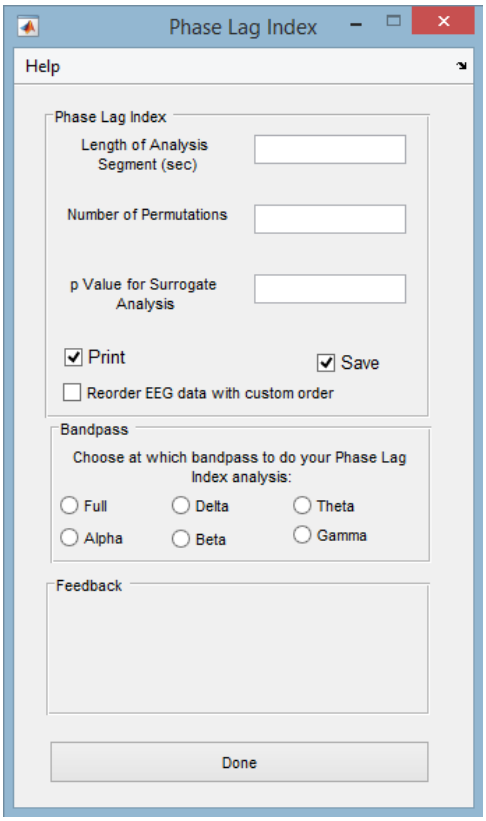

Figure 18: Phase Lag Index windows that appears when the corresponding radio button is pressed in the main window.

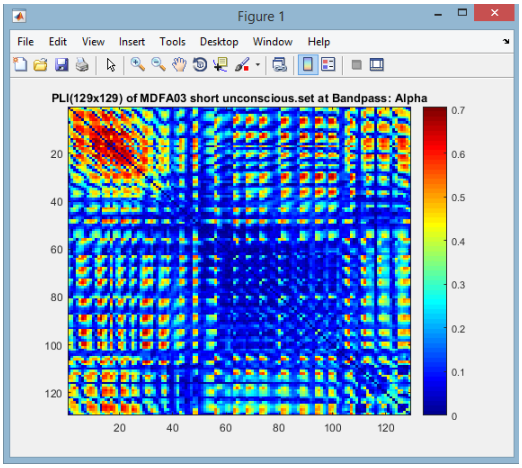

Figure 19: Example of a PLI plot made with an alpha bandpass.

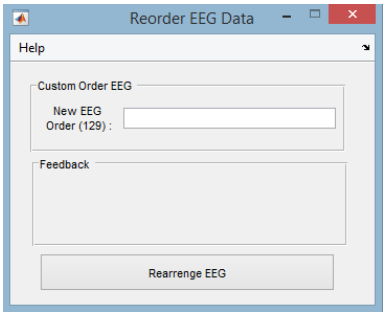

Figure 20: Pop up window that let you reorder the channels order.

Bandpass Menu:

In the bandpass menu you can select at which frequency you want to filter your EEG data in order to do the Phase Lag Index analysis. If you select more than one frequency pass, the analysis will be repeated for each bandpass,

7. **Full:** Full band goes from 1Hz to 50Hz.
8. **Delta:** Delta band goes from 1Hz to 4Hz.
9. **Theta:** Theta band goes from 4Hz to 8Hz.
10. **Alpha:** Alpha band goes from 8Hz to 13Hz.
11. **Beta:** Beta band goes from 13Hz to 30Hz.
12. **Gamma:** Gamma band goes from 30Hz to 50Hz.

Example of Input:

Figure 19 was made by inputting the following:
Length of Analysis Segment: 10, Number of Permutation: 20, p value: 0.05,bandpass: alpha.

Directed Phase Lag Index

Directed Phase Lag Index Menu:

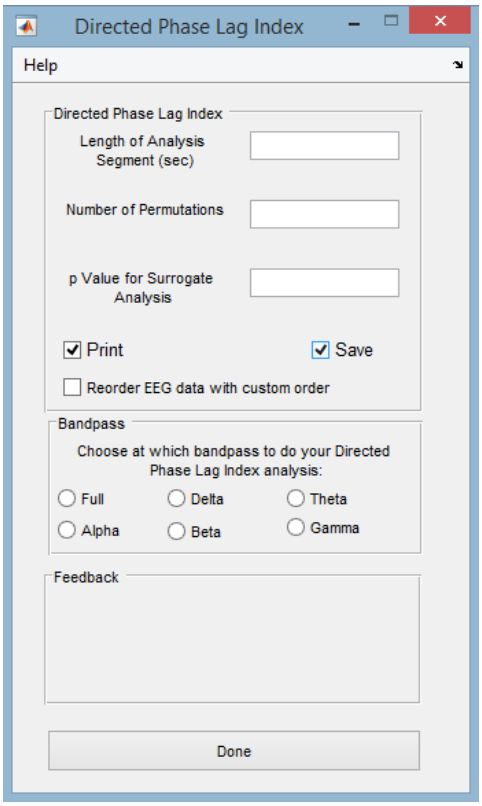

Figure 21: Windows that open when Directed Phase Lag Index is selected in the main window.

1. **Length of Analysis Segment:** The whole length of the EEG data will be cut into segments and the dPLI will be calculated for each segment before getting averaged. The length in seconds need to be between 0 and the length of the EEG data.
2. **Number of Permutations:** The number of permutation correspond to the number of time the surrogate will be calculated. The number of permutations need to be greater than 0.
3. **p Value for Surrogate Analysis:** The p value will be used in the surrogate analysis to assess whether a given dPLI is significant or no.
4. **Print check box:** If this option is checked the pipeline will output a dPLI plot at the end of the analysis (fig. 22).
5. **Save check box:** If select the dPLI will be saved in the Directed Phase Lag Index directory. Also, a text file containing the input will be saved using the same name as the figure.
6. **Reorder EEG data with custom order check box:** If selected a pop-up window will appear where you will be able to enter the order in which you want each channel to appear (fig. 20). The default order is from channel 1 to the last channel. Make sure that there is no duplicate and that there is the same number of channels as before.

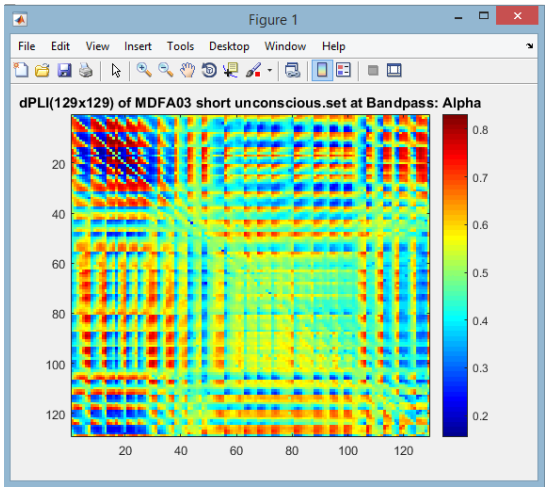

Figure 22: Example of a dPLI analysis done with an alpha bandpass.

Bandpass Menu:

In the bandpass menu you can select at which frequency you want to filter your EEG data in order to do the Directed Phase Lag Index analysis. If you select more than one frequency pass, the analysis will be repeated for each bandpass,

1. **Full:** Full band goes from 1Hz to 50Hz.
2. **Delta:** Delta band goes from 1Hz to 4Hz.
3. **Theta:** Theta band goes from 4Hz to 8Hz.
4. **Alpha:** Alpha band goes from 8Hz to 13Hz.

5. **Beta:** Beta band goes from 13Hz to 30Hz.
6. **Gamma:** Gamma band goes from 30Hz to 50Hz.

Feedback:

Like in the main window the feedback panel will output warnings if illegal input are given to the pipeline.

Done Button:

When every variable has been entered click the “Done” button to go back to the main window.

Example of Input:

Figure 22 was made by inputting the following:
Length of Analysis Segment: 10, Number of Permutation: 20, p value: 0.05,bandpass: alpha.

Symbolic Transfer Entropy

Symbolic Transfer Entropy Menu:

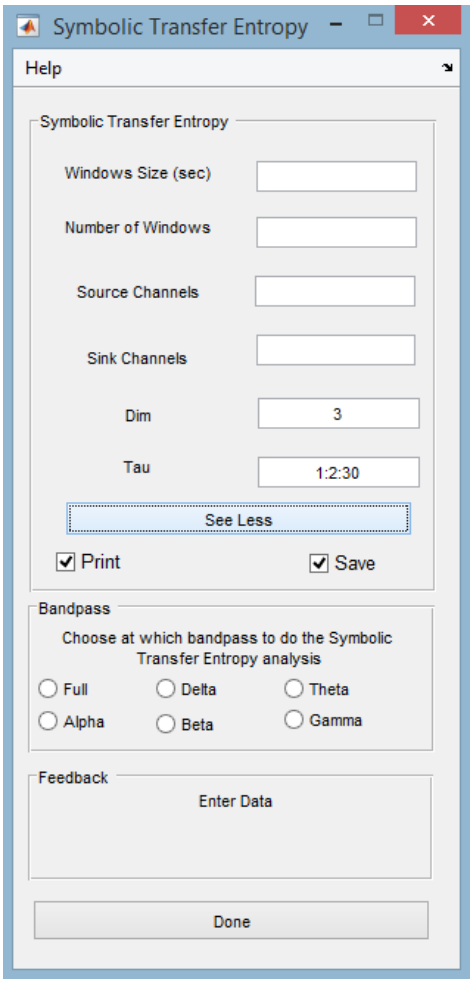

Figure 23: Symbolic Transfer Entropy window that appear after the corresponding radio button is selected in the main window.

1. **Windows Size:** Windows size used to calculate each segment for the symbolic transfer entropy analysis.
2. **Number of Windows:** Number of windows of the given size inputted above. Make sure the numbers of windows times the windows size is not greater than the length of the EEG.
3. **Source Channels:** The channels from which a signal could originate.
4. **Sink Channels:** The channels that receive could receive a signal from the source channels.
5. **Dim:** Dim represent the embedding dimension[2].
6. **Tau:** Tau represent the time delay[2]. Both variables are used to convert the transfer entropy into its symbolic form.
7. **Print check box:** If this is selected when the pipeline will be done computing the symbolic transfer entropy it will output it on the screen in the command windows. Because the output of this analysis technique is a structure file the output might get messy and span a great length of the command windows.
8. **Save check box:** When done, the pipeline will save the Symbolic Transfer Entropy data as a structure file in two form (fig. 24). The first form is a long storage form looking like this BandpassDay-month-year_time_ste.mat and another one that is short term and that will be overwritten each time a STE analysis with a same bandpass will be done. It will look like this: Bandpassste.mat. Also, a text file containing the input will be saved using the same name as the structure.

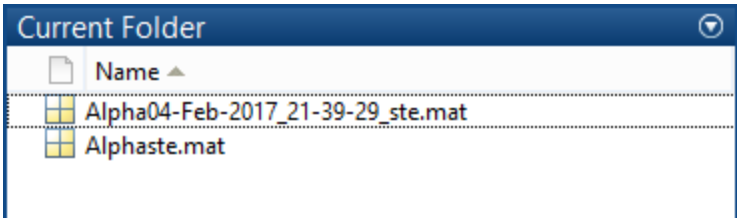

Figure 24: Saved Symbolic transfer entropy structure file. At top is the long storage form and at the bottom is the short storage form that will be overwritten.

Bandpass Menu:

In the bandpass menu you can select at which frequency you want to filter your EEG data in order to do the Symbolic Transfer Entropy analysis. If you select more than one frequency pass, the analysis will be repeated for each bandpass,

1. **Full:** Full band goes from 1Hz to 50Hz.
2. **Delta:** Delta band goes from 1Hz to 4Hz.
3. **Theta:** Theta band goes from 4Hz to 8Hz.
4. **Alpha:** Alpha band goes from 8Hz to 13Hz.
5. **Beta:** Beta band goes from 13Hz to 30Hz.
6. **Gamma:** Gamma band goes from 30Hz to 50Hz.

Feedback:

Like in the main window the feedback panel will output warnings if illegal input are given to the pipeline.

Done Button:

When every variable has been entered click the “Done” button to go back to the main window.

Example of Input:

The structure showed at figure 24 were made by inputting the following:
Windows Size: 10, Number of Windows: 6, Source Channels: 1:10, Sink Channels: 1:10,Dim and Tau were left at default and bandpass: alpha.

Graph Theory

Graph Theory Menu:

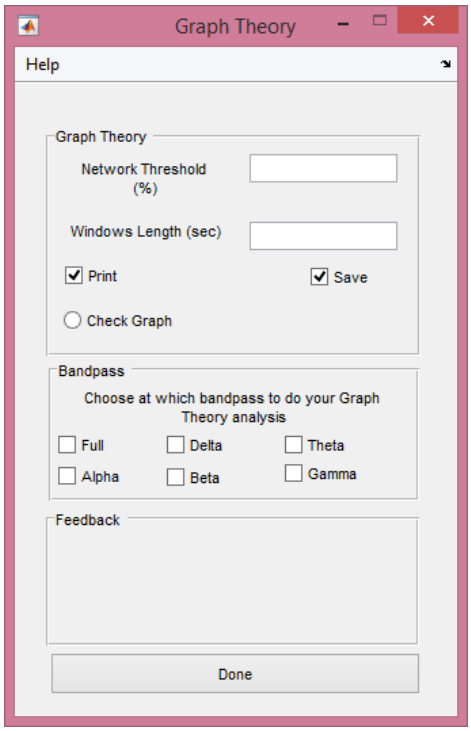

Figure 25: Windows that open when the Graph Theory radio button is selected in the main window.

1. **Network Threshold:** This value ranging from 0 to 1 will be used to construct the binary matrix of the connection between one channel and another.
2. **Windows Length:** The length in seconds of the each segments that will be used to calculate the relevant component of the graph theory analysis.
3. **Print check box:** If this option is selected, when the pipeline will be done calculating the component of the analysis it will output them on the screen.
4. **Save check box:** When done, the pipeline will save the Graph Theory data as a structure file in two form (like in fig. 24). The first form is a long storage form looking like this BandpassDay-month-year_time_graphTheoryData.mat and another one that is short term and that will be overwritten each time a Graph Theory analysis with a same bandpass will be done. It will look like this: BandpassgraphData.mat. The short format is more easily writable into script than the long format, but in either case you can change the name to something more meaningful. Also, a text file containing the input will be saved using the same name as the structure.
5. **Check Graph:** If this option is selected the pipeline will stop in the middle of the graph theory and will output a 3D rendering of what the node and the edges looks like with BrainNet Viewer[3] (fig 26). A pop up will then appear which will ask you if you wish to continue with this analysis or if you wish to abort. If continue is selected the pipeline will continue calculating the graph theory component. If abort is selected the pipeline will exit the graph theory analysis and nothing will be saved.

Bandpass Menu:

In the bandpass menu you can select at which frequency you want to filter your EEG data in order to do the Graph Theory analysis. If you select more than one frequency pass, the analysis will be repeated for each bandpass,

1. **Full:** Full band goes from 1Hz to 50Hz.
2. **Delta:** Delta band goes from 1Hz to 4Hz.
3. **Theta:** Theta band goes from 4Hz to 8Hz.

4. **Alpha:** Alpha band goes from 8Hz to 13Hz.
5. **Beta:** Beta band goes from 13Hz to 30Hz.
6. **Gamma:** Gamma band goes from 30Hz to 50Hz.

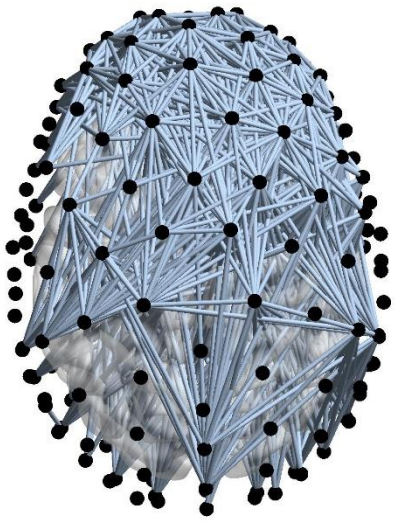

Figure 26: Example of a graph obtained by selecting check graph in the user interface. This graph was generated with the BrainNet Viewer program.

Feedback:

Like in the main window the feedback panel will output warnings if illegal input are given to the pipeline.

Done Button:

When every variable has been entered click the "Done" button to go back to the main window.

Example of Input:

Figure 26 & the structure that contain its parameters was made by inputting the following: Network Threshold: 5, Windows Length: 10 and bandpass: alpha.

Output:

A structure containing the binary matrix used in calculating the other property at each window, the degree of the channels at each window, binary clustering coefficient, characteristic path length, binary global efficiency, binary smallworldness and the modularity.

Command Line Functions:

If for some reasons you wish to use the analysis techniques without having to deal with the graphical user interface you can. To do so you only need to install EEGapp and call functions that are bundled with the package. Beware that no input checking is done before executing the analysis techniques.

Spectrogram and Topographic Map:

errors = spectopo_eegapp(EEG,fp,tso,frequencies,timeBandwidth,numberTaper,windowLength,stepSize)

Input:
EEG = eeg structure file coming from eeglab
fp = frequency pass, two numbers corresponding to a low pass and an high pass given like this [a b]
tso = temporal smoothing median filter order, used to reduce the noise in the spectrogram
frequencies = used to plot the topographic map, will plot up to 4 frequencies.
timeBandwith and numberTaper = see documentation
windowLength = length of the window over which the spectrum will be calculated
stepSize = the step length over which to move thw window.

Output:
errors = 0 if an error occurred and 1 if not.
Will also print a plot of the spectrogram and of the topographic map.

Phase Amplitude Coupling:

errors = pac_eegapp(EEG,channels,window_length,numberOfBin,LFO_bp,HFO_bp)

Input:
EEG = eeg structure file coming from eeglab
channels = a vector containing all channels to analyze
windows_length = length of the windows of data to calculate the pac.
numberOfBin = number of bins to categorize the different phases.
LFO_bp = Low Frequencies Oscillation bandpass. Two numbers a low pass and an high pass given like this [a b]
HFO_bp = High Frequencies Oscillation bandpass. Two numbers a low pass and an high pass given like this [a b]

Output:
errors = 0 if an error occurred and 1 if not.
Will also print a plot of the phase amplitude coupling using imagesc.

Coherence:

errors = coherence_eegapp(EEG,from,to,bandpass)

Input:
EEG = eeg structure file coming from eeglab
from = a vector containing channels number
to = a vector containing channels number
bandpass = a string: full, alpha, beta, gamma, theta or delta.

Output:
errors = 0 if an error occurred and 1 if not.
Will also print a plot of the coherence between channels from and to.

Phase Lag Index:

errors = pli_eegapp(EEG,data_length,permutation,p_value,bandpass,custom_order)

Input:
EEG = eeg structure file coming from eeglab
data_length = length of the windows of data to calculate the pli.
permutation = number of permutation to do the surrogate data analysis.
p_value = ranging from 0 to 1.
bandpass = a string: full, alpha, beta, gamma, theta or delta.
custom_order = a vector containing all channels in a particular ordering

Output:
errors = 0 if an error occurred and 1 if not.
Will also print a plot of the pli.

Directed Phase Lag Index:

errors = dpli_eegapp(EEG,data_length,permutation,p_value,bandpass,custom_order)

Input:
EEG = eeg structure file coming from eeglab
data_length = length of the windows of data to calculate the dpli.
permutation = number of permutation to do the surrogate data analysis.
p_value = ranging from 0 to 1.
bandpass = a string: full, alpha, beta, gamma, theta or delta.
custom_order = a vector containing all channels in a particular ordering

Output:
errors = 0 if an error occurred and 1 if not error
Will also print a plot of the dpli.

Symbolic Transfer Entropy:

ste_struct = ste_eegapp(EEG,winsize,NumWin,from,to,bandpass,dim,tau)

Input:
EEG = eeg structure file coming from eeglab
winsize = size of the windows to calculate ste.
NumWin = number of windows to calculate ste.
from = a vector containing channels number
to = a vector containing channels number
bandpass = a string: full, alpha, beta, gamma, theta or delta.
dim and tau = see documentation.

Output:
ste_struct = structure containing the ste data. Will be equal to 0 if there was an error.

Graph Theory:

graph_struct = graph_eegapp(EEG,network_thresh,win,bandpass)

Input:
EEG = eeg structure file coming from eeglab
network_thresh = number ranging from 0 to 100, used to construct a binary matrix

win = The length in seconds of the each segments that will be used in the analysis.
bandpass = a string : full, alpha, beta, gamma, theta or delta.

Output:
graph_struct = a structure containing 5 properties of the graph. Will return a 0 if there was an error.

For more information type help *name of the function* in MATLAB.

Reference:

1. Chronux Manual (2008) http://chronux.org/chronuxFiles/filesReleases/manual.pdf
2. Anesthesiology 2013; 188:1264-75 in the manual
3. Xia M, Wang J, He Y (2013) BrainNet Viewer: A Network Visualization Tool for Human Brain Connectomics. PLoS ONE 8: e68910.