

# Letswave 6

## Reference manual

01/03/2015

<http://www.nocions.org>

**TABLE OF CONTENTS**

<b>LETSWAVE 6 MAIN USER INTERFACE</b>	<b>6</b>
<b><u>FILE</u></b>	<b>8</b>
<b>IMPORT SIGNALS</b>	<b>9</b>
IMPORT ASCII TEXT FILES	9
IMPORT BIOSEMI BDF	10
IMPORT ASALAB CNT FILES	11
IMPORT DELTAMED ASCII/BIN FILES	12
IMPORT EDF FILES	13
IMPORT LETSWAVE 4 STRUCTURED MAT FILES	14
IMPORT LETSWAVE 5 DATAFILES	15
IMPORT MATLAB WORKSPACE VARIABLE	16
IMPORT MEGA DATAFILES	17
IMPORT EEGLAB SET FILES	18
IMPORT MICROMED TRC FILES	19
IMPORT BRAINVISION VHDR FILES	20
<b>EXPORT SIGNALS</b>	<b>21</b>
EXPORT MAT	21
EXPORT ASCII	22
<b>FILE MANAGEMENT</b>	<b>23</b>
CREATE FOLDER	23
RENAME FOLDER	23
DELETE FOLDER	23
RENAME DATASET	23
DELETE DATASET(S)	23
COPY DATASET(S)	23
MOVE DATASET(S)	23
<b><u>EDIT</u></b>	<b>24</b>
<b>ELECTRODES</b>	<b>25</b>
EDIT ELECTRODE COORDINATES	25
EDIT ELECTRODE INFO	26
EDIT ELECTRODE LABELS	27
COMPUTE SPLINE HEADMODEL	28
INTERPOLATE CHANNEL USING NEIGHBOURING ELECTRODES	29
<b>ADD OR EDIT TAGS</b>	<b>30</b>
<b>ADD OR EDIT EPOCH DATA</b>	<b>31</b>
<b>EDIT PROPERTIES</b>	<b>32</b>
<b><u>EVENTS</u></b>	<b>33</b>
<b>BROWSE AND EDIT EVENTS</b>	<b>34</b>
<b>DELETE DUPLICATE EVENTS</b>	<b>35</b>
<b>CREATE EVENTS FROM LEVEL TRIGGER</b>	<b>36</b>
<b>MERGE EVENT CODES AND LATENCIES</b>	<b>37</b>
<b><u>PREPROCESS</u></b>	<b>38</b>
<b>FREQUENCY FILTERS</b>	<b>39</b>
FFT FILTER	39
FFT NOTCH FILTER	40
BUTTERWORTH FILTER	41

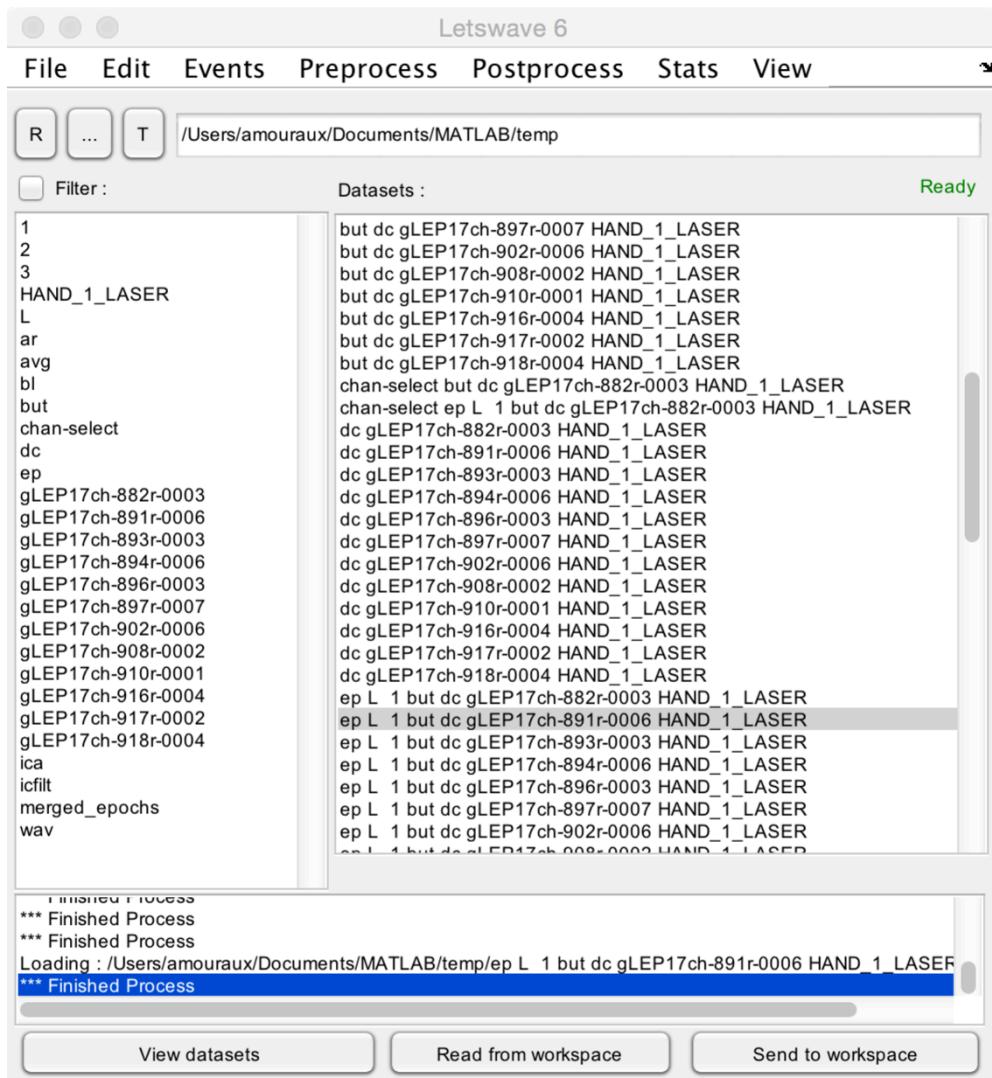
<b>SPATIAL FILTERS</b>	<b>42</b>
COMPUTE ICA MATRIX	42
COMPUTE ICA MATRIX (MERGED)	43
APPLY ICA FILTER	44
ASSIGN ICA MATRIX	45
ICA UNMIX SIGNALS	46
ICA REMIX SIGNALS	47
<b>SEGMENTATION</b>	<b>48</b>
SEGMENTATION RELATIVE TO EVENTS	48
SEGMENTATION RELATIVE TO EVENTS (ONE FILE PER EVENT CODE)	49
SEGMENTATION RELATIVE TO EVENTS (SS-EPs)	50
SEGMENT EPOCHS IN SUCCESSIVE CHUNKS	51
<b>BASELINE OPERATIONS</b>	<b>52</b>
BASELINE CORRECTION	52
FREQUENCY SPECTRUM SNR	53
<b>ARTIFACTS</b>	<b>54</b>
REJECT EPOCHS (AMPLITUDE CRITERION)	54
REJECT EPOCHS (VISUAL INSPECTION)	55
SUPPRESS ARTIFACT (LATENCY DEFINED)	56
SUPPRESS ARTIFACT (EVENT DEFINED)	57
<b>CURRENT SOURCE DENSITY (CSD)</b>	<b>58</b>
COMPUTE SCALP CSD	58
COMPUTE LINEAR CSD	59
<b>FREQUENCY TRANSFORMS</b>	<b>60</b>
CONTINUOUS WAVELET TRANSFORM (CWT)	60
FAST CWT	61
FAST FOURIER TRANSFORM (FFT)	63
SHORT-TERM FFT (ST-FFT)	64
SHORT-TERM FFT (ZHANG ET AL.)	66
HILBERT TRANSFORM	67
INVERSE FFT	68
<b>TIME-FREQUENCY FILTERS</b>	<b>69</b>
BUILD WAVELET FILTER (HU ET AL)	69
APPLY WAVELET FILTER (HU ET AL)	70
<b>RESAMPLE SIGNALS</b>	<b>71</b>
CROP SIGNALS	71
DOWNSAMPLE SIGNALS	72
NEW CHANNEL : LINEAR MIX	73
RESAMPLE SIGNALS	74
<b>ARRANGE SIGNALS</b>	<b>75</b>
REARRANGE/DELETE CHANNELS	75
REARRANGE/DELETE EPOCHS	76
REARRANGE/DELETE INDEX	77
EQUALIZE NUMBER OF EPOCHS ACROSS DATASETS	78
MERGE CHANNELS OF MULTIPLE DATASETS	79
MERGE EPOCHS OF MULTIPLE DATASETS	80
MERGE INDEXES OF MULTIPLE DATASETS	81
SELECT EPOCHS ACCORDING TO EPOCH DATA	82
SELECT EPOCHS ACCORDING TO EVENT LATENCIES	83
SORT EPOCHS ACCORDING TO EPOCH DATA	84
SORT EPOCHS ACCORDING TO EVENT LATENCIES	85
<b>DC REMOVAL AND LINEAR DETREND</b>	<b>86</b>
<b>REREFERENCE</b>	<b>87</b>
<b>REREFERENCE (CUSTOM MONTAGE)</b>	<b>88</b>

<b>POSTPROCESS</b>	<b>89</b>
<b>AVERAGE</b>	<b>90</b>
AVERAGE EPOCHS	90
SLIDING OPERATION ALONG DIMENSION	91
SLIDING AVERAGE ACROSS TRIALS	92
GRAND AVERAGE (WEIGHTED)	93
LINEAR CHANNEL MAP	94
POOL CHANNELS	95
WEIGHTED CHANNEL AVERAGE (CREATE TEMPLATE)	96
WEIGHTED CHANNEL AVERAGE (APPLY TEMPLATE)	97
<b>MATH</b>	<b>98</b>
COMPUTE SIGNAL DERIVATIVE	98
MATHEMATICAL OPERATION USING TWO DATASETS	99
MATHEMATICAL OPERATION USING A CONSTANT	100
RECTIFY OR SQUARE SIGNALS	101
THRESHOLD SIGNALS	102
<b>SOURCE ANALYSIS</b>	<b>103</b>
DIPFIT : LOAD CHANNEL COORDINATES	103
DIPFIT : LOAD HEAD MODEL	104
DIPFIT : LOAD MRI DATA	105
DIPFIT : FIT DIPOLE(S) AT A GIVEN LATENCY	106
DIPFIT : FIT DIPOLE(S) ONTO IC TOPOGRAPHIES	107
DIPFIT : FIT DIPOLE(S) AT EVENT LATENCIES	108
<b>SINGLE TRIAL ANALYSIS</b>	<b>109</b>
MULTIPLE LINEAR REGRESSION (MLR)	109
MULTIPLE LINEAR REGRESSION (MLR)	110
<b>FIND PEAKS IN WAVEFORMS</b>	<b>111</b>
<b>GLOBAL EXPLAINED VARIANCE</b>	<b>113</b>
<b>STATISTICS</b>	<b>114</b>
COMPARE MORE THAN TWO DATASETS (ANOVA)	115
COMPARE TWO DATASETS (T-TEST)	117
COMPARE TWO DATASETS (WILCOXON)	119
COMPARE SIGNALS AGAINST A CONSTANT	121
COMPARE SIGNAL AMPLITUDE AT THE EVENT LATENCIES IN TWO DATASETS	123
BOOTSTRAP TEST AGAINST A REFERENCE INTERVAL	125
<b>VIEW</b>	<b>126</b>
WAVEFORM MULTI-VIEWER	127
MAP MULTI-VIEWER	129
WAVEFORM AVERAGE MULTI-VIEWER	131
WAVEFORM CONTINUOUS DATA VIEWER	133
SEEG VIEWER	135
EVENT VIEWER: VIEW SIGNALS AT EVENT LATENCIES	136
GENERATE TABLE AND GRAPHS WITH ROI DATA	137
GENERATE TABLE AND GRAPHS WITH SS-EP DATA	138
EPOCH DATA VIEWER	139
<b>FIGURES</b>	<b>140</b>
LISSAJOUS FIGURE CREATOR	140
MAPS FIGURE CREATOR	141
WAVE SCALP ARRAY FIGURE CREATOR	142
SCALP MAP SS-EP SERIES FIGURE CREATOR	144

SCALP MAP TIME SERIES FIGURE CREATOR	145
WAVEFORMS FIGURE CREATOR	146
WAVEFORMS AVERAGE FIGURE CREATOR	147
<b><u>HISTORY</u></b>	<b><u>150</u></b>
<b><u>UPDATES</u></b>	<b><u>151</u></b>
<b><u>GUI OPTIONS</u></b>	<b><u>152</u></b>

## LETSWAVE 6 MAIN USER INTERFACE

The Letswave main user interface is started by typing 'letswave' in the Matlab command prompt.



- The **Datasets** listbox lists all the datasets found in the current folder, sorted by alphabetical order. To apply a function to a given dataset, simply select one or more dataset(s) from that listbox, and choose the function using the dropdown title menus.
- When running a function, information regarding the process is provided in the lower text box. In addition, the upper-right Ready/Busy label provides informs whether the function is still busy processing the data, or whether it has finished.
- To view dataset(s), you can either double-click the dataset(s) in the datasets listbox, click the **View datasets** button, or select a custom viewer interface in the **View** menu.
- If you wish to modify manually a dataset using your own Matlab code, you can send dataset(s) to the Matlab workspace using the **Send to workspace** button. This will create a new variable in the Matlab workspace, named **Iwdata**. The workspace variable can be imported back into Letswave using the **Read from workspace** button. This will replace the content of the selected dataset with the data contained in the **Iwdata** workspace variable.

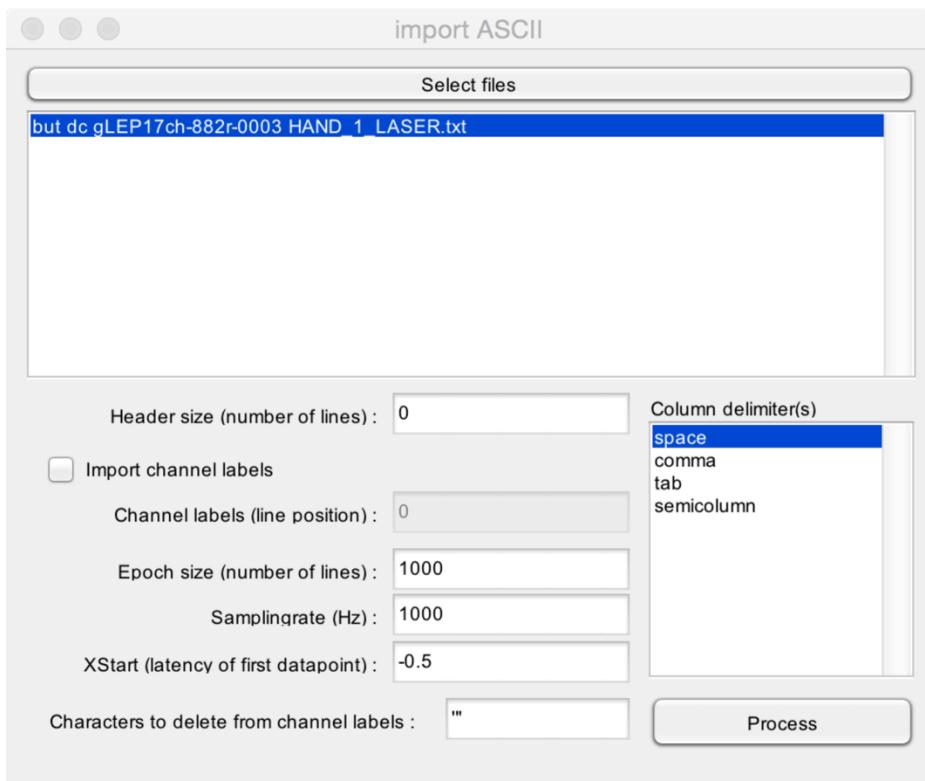
- The left **Filter** listbox lists all the different strings found in the dataset filenames. By selecting one or more strings, it is possible to filter the content of the datasets listbox, such as to display only datasets which have the corresponding string. The **Filter** checkbox is used to enable/disable filtering of the displayed datasets.
- The **R** button is used to refresh the content of the datasets listbox, for example, after manually copying files in the current folder.
- The **...** button is used to change the current folder.
- The **T** button is used to find datasets with matching tags.

**FILE**

## IMPORT SIGNALS

### IMPORT ASCII TEXT FILES

Import data stored in an ASCII (text) file. The data must be organized as follows. Samples are stored as lines. Channels are stored as columns. The data can contain multiple epochs of equal size (i.e. equal number of lines).



- **Select files.** Select one or more ASCII (text) files to import.
- **Header size.** Data exported as a text file often contain a header detailing the content of the text file, and providing information regarding the way the data is stored. Specify here the number of lines taken by the header.
- **Import channel labels.** It is possible to import channel labels, if these are stored in the text file. The channel labels must be stored on a single line, each column corresponding to a single channel label.
- **Channel labels (line position).** If you choose to import channel labels, you must mention the line number where channel labels are stored in the header.
- **Epoch size.** If the data contains several epochs, you should mention the size (number of lines) of each epoch.
- **Samplingrate.** The samplingrate of the data (samples/second).
- **XStart.** The timestamp of the first sample of each epoch (e.g. -0.5 s if the epochs extend from -0.5 to +1.0 s)
- **Characters to delete from channel labels.** Sometimes, bracket or hyphens are added around the channel labels. You can choose to delete these characters, for example, to recover standard International 10-5 labels.
- **Column delimiters.** character codes used to delimit columns in the data. It is possible to select multiple column delimiters, should that be the case.

## IMPORT BIOSEMI BDF

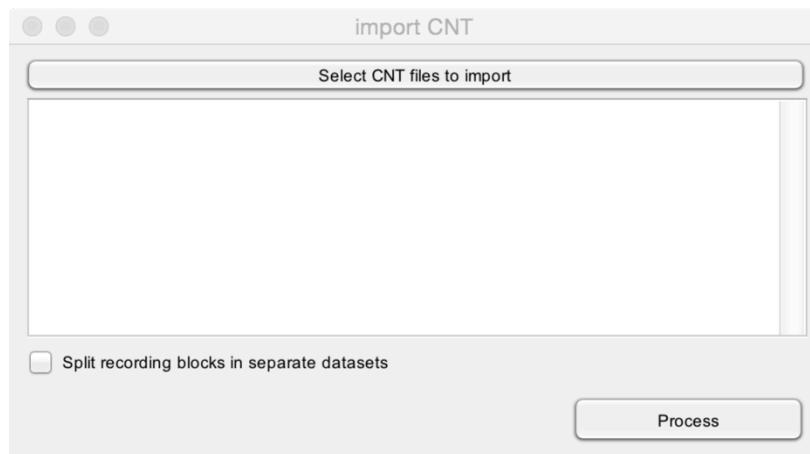
Import BDF datafiles (Biosemi).



- **Select files.** Select the datasets that you wish to import.

## IMPORT ASALAB CNT FILES

Import ASALAB CNT datafiles (ANT).



- **Select files.** Select the datasets that you wish to import.
- **Split recording blocks in separate datasets.** If the datafile contains multiple recording blocks, each block will be stored as a separate dataset.

## IMPORT DELTAMED ASCII/BIN FILES

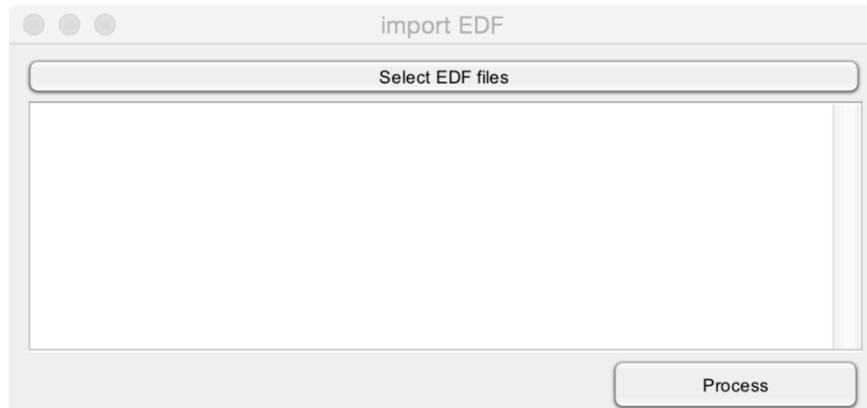
Import Deltamed ASCII/BIN files. These files can be generated in Coherence, using the export function. It is recommended to export the data as binary rather than text format, to reduce the time required to perform the export and import functions.



- **Select files.** Select the datasets that you wish to import. You should choose the text file which is created by the Coherence export function. The function will then automatically search for the associated BIN or ASCII file.

## IMPORT EDF FILES

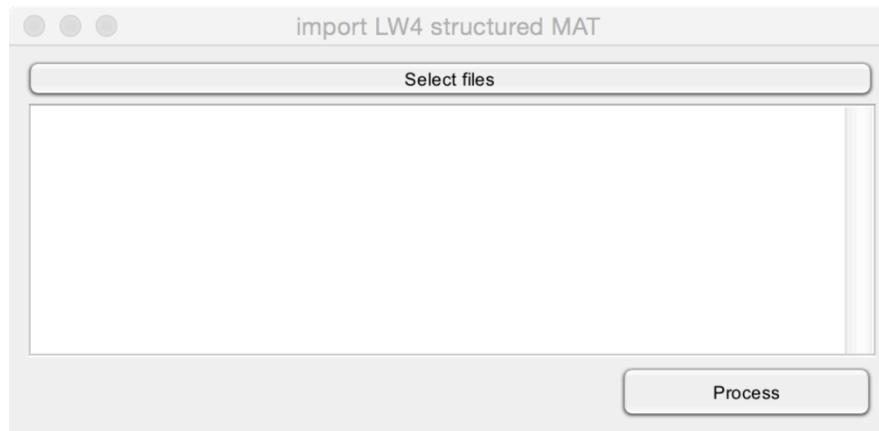
Import EDF datafiles (European Data Format).



- **Select files.** Select the datasets that you wish to import.

## IMPORT LETSWAVE 4 STRUCTURED MAT FILES

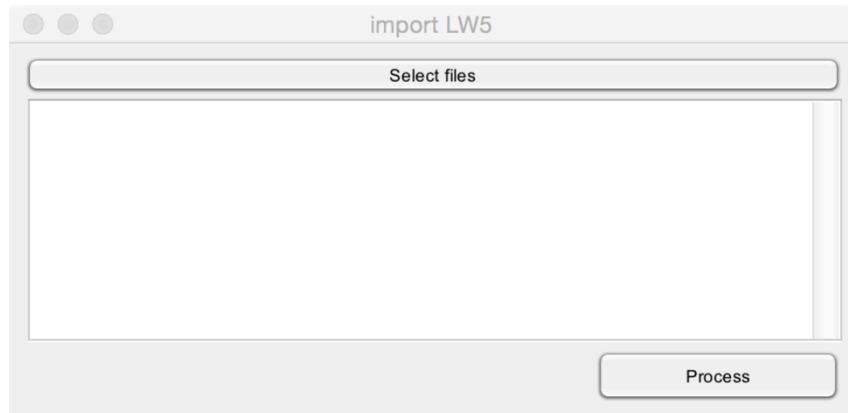
Import Letswave 4 datafiles. These files must be generated using the “Export as structured MAT file” in Letswave 4.



- **Select files.** Select the datasets that you wish to import.

## IMPORT LETSWAVE 5 DATAFILES

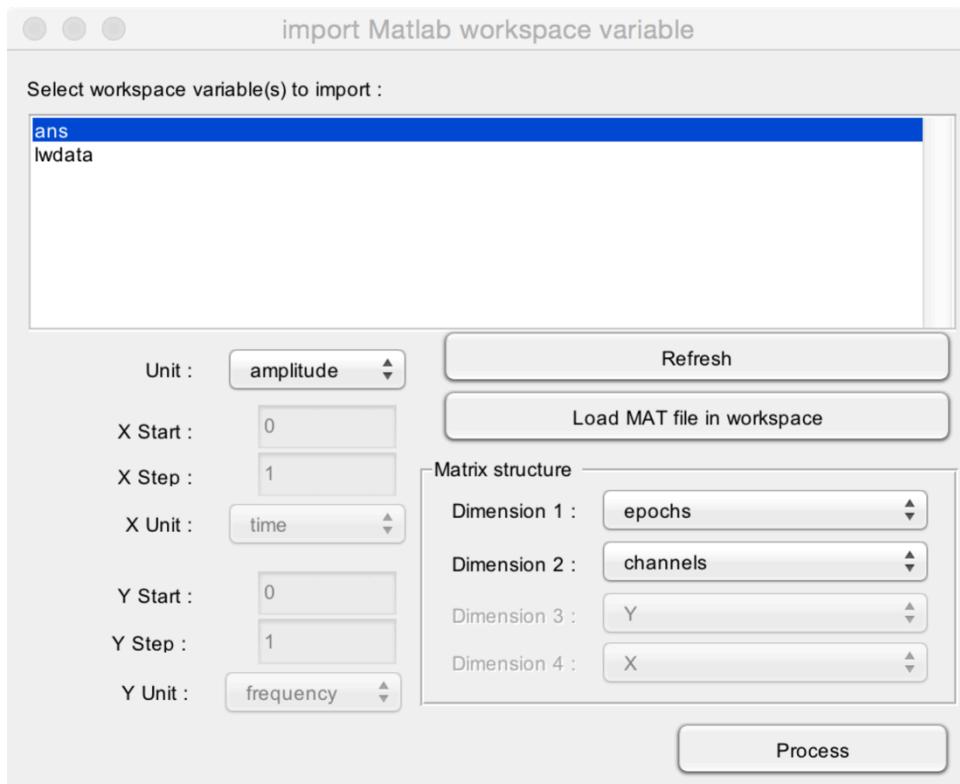
Import Letswave 5 datafiles.



- **Select files.** Select the datasets that you wish to import.

## IMPORT MATLAB WORKSPACE VARIABLE

Import data stored in a current variable of the Matlab workspace. The imported Matlab matrix can have up to four dimensions, storing epochs, channels, X, and Y dimensions.

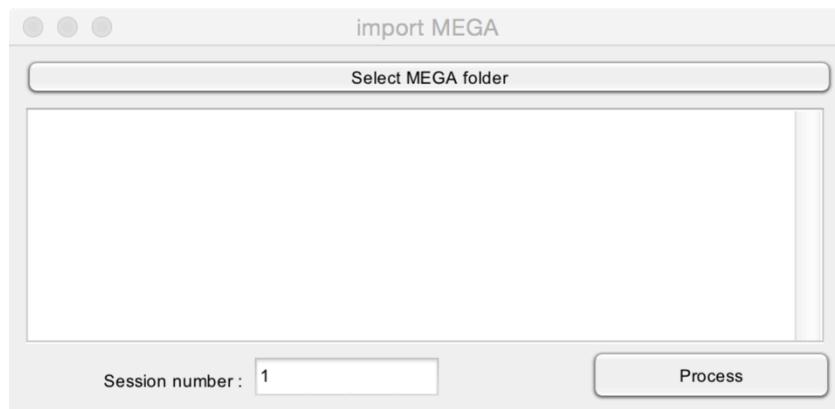


Example. A workspace variable containing EEG data with 10 epochs, 19 channels, and 1500 samples per epoch. Each epoch extends from -0.5 seconds to +1.0 seconds. Sampling rate is 1000 Hz. The size of the matrix is 10x19x1500.

- **Select workspace variable(s) to import.** Select one or more variables from the Matlab workspace.
- **Refresh.** Refresh the list of workspace variables.
- **Load MAT file in workspace.** Load a MAT file in the Matlab workspace.
- **Matrix structure.** Define the dimensions of the matrix. For the example, Dimension 1 = epochs, Dimension 2 = channels, Dimension 3 = X.
- **Unit.** The unit of the data stored in the dataset.
- **X Start.** The position of the first sample in each epoch along the X dimension. For the example : X Start = -0.5
- **X Step.** The interval between two consecutive samples along the X dimension. For the example, samplingrate is 1000 Hz, so X Step is  $1/1000 = 0.001$  s.
- **X Unit.** The unit of the X dimension. For the example : X Unit = time.
- **Y Start, YStep, Y Unit.** The values defining the Y dimension.

## IMPORT MEGA DATAFILES

Import MEGA datafiles (Neurone software).



- **Select MEGA folder.** Select the folder containing the MEGA dataset. Note that you can only import one dataset.
- **Session Number.** The number of the session to import.

## IMPORT EEGLAB SET FILES

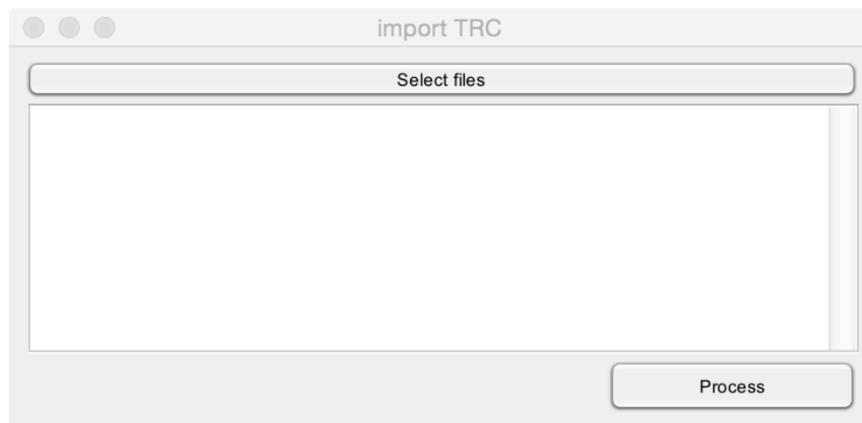
Import EEGLAB SET files.



- **Select files.** Select SET files to import.

## IMPORT MICROMED TRC FILES

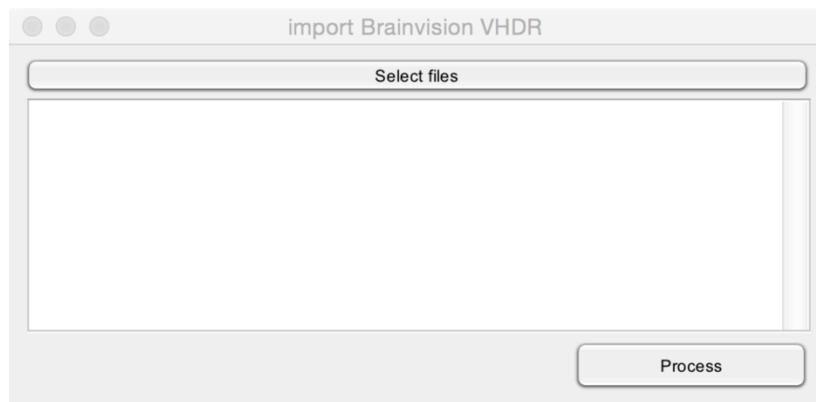
Import Micromed TRC files.



- **Select files.** Select the Micromed TRC files to import.

## IMPORT BRAINVISION VHDR FILES

Import Brainvision VHDR files.

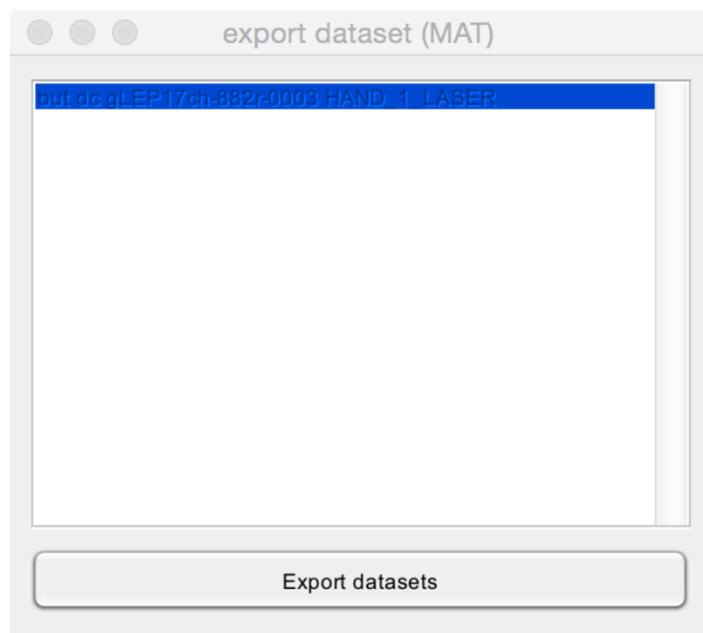


- **Select files.** Select the Brainvision VHDR files to import.

## EXPORT SIGNALS

### EXPORT MAT

Export dataset as a MAT file. The suffix '\_MAT' will be added to the filename.

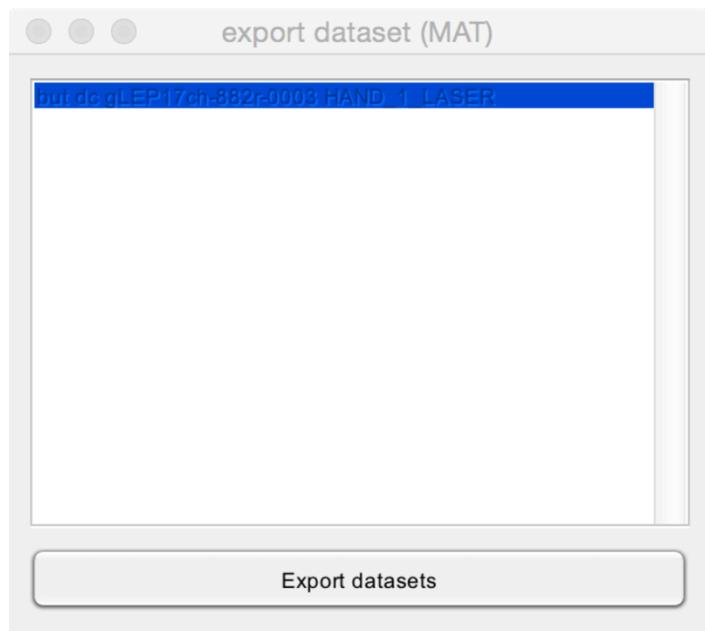


The exported dataset will be a matrix with all data organized according to the existing epoch/channel/index/z/y/x dimensions.

Note that you may also wish to use the 'Send to Workspace' button from the main interface, if you want to access the data directly from the Matlab command window.

## EXPORT ASCII

Export dataset as a text file.



## **FILE MANAGEMENT**

### **CREATE FOLDER**

Create a new folder.

### **RENAME FOLDER**

Rename an existing folder.

### **DELETE FOLDER**

Delete a folder.

### **RENAME DATASET**

Rename a dataset.

### **DELETE DATASET(S)**

Delete selected dataset(s).

### **COPY DATASET(S)**

Copy selected dataset(s).

### **MOVE DATASET(S)**

Move selected dataset(s).

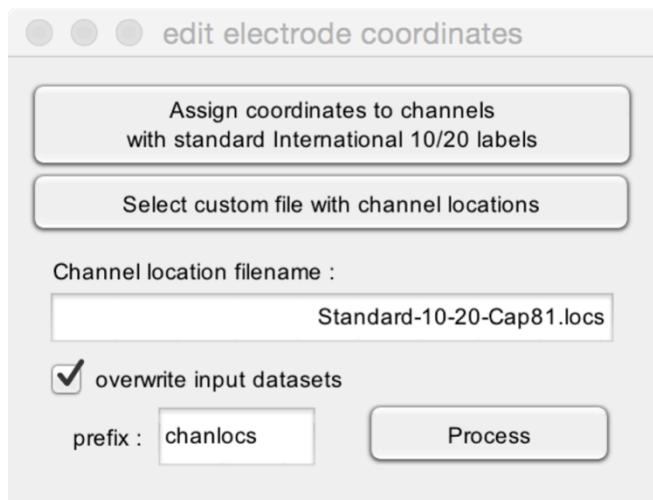
**EDIT**

## ELECTRODES

### EDIT ELECTRODE COORDINATES

Edit electrode coordinates (X/Y/Z). The coordinates must be read from a ‘chanlocs’ file. The function will assign coordinates to all channels whose labels match the labels in the ‘chanlocs’ file.

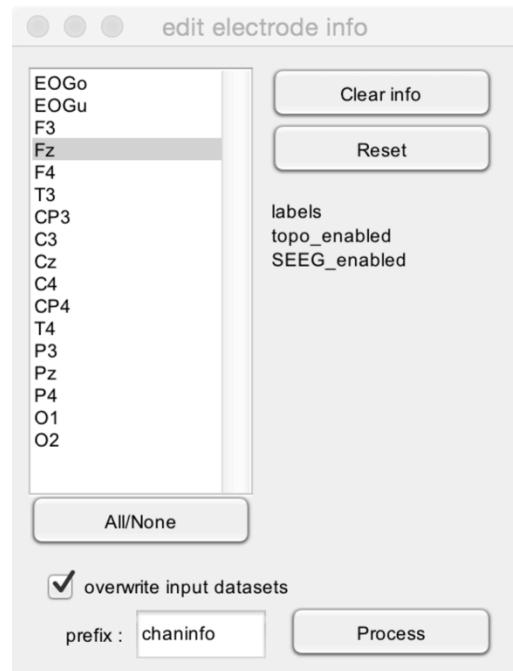
This step is required to view 2D scalp maps. For 3D scalp maps, you must also run the Compute headmodel function.



- **Assign coordinates to channels with standard international 10/20 labels.** All channels with a label corresponding to the International 10/20 system (e.g. Cz) will be assigned a standard coordinate.
- **Select custom file with channel locations.** Select a custom ‘locs’ file with channel locations and labels. All channels with a label corresponding to the label stored in the file will be assigned their corresponding coordinate.

## EDIT ELECTRODE INFO

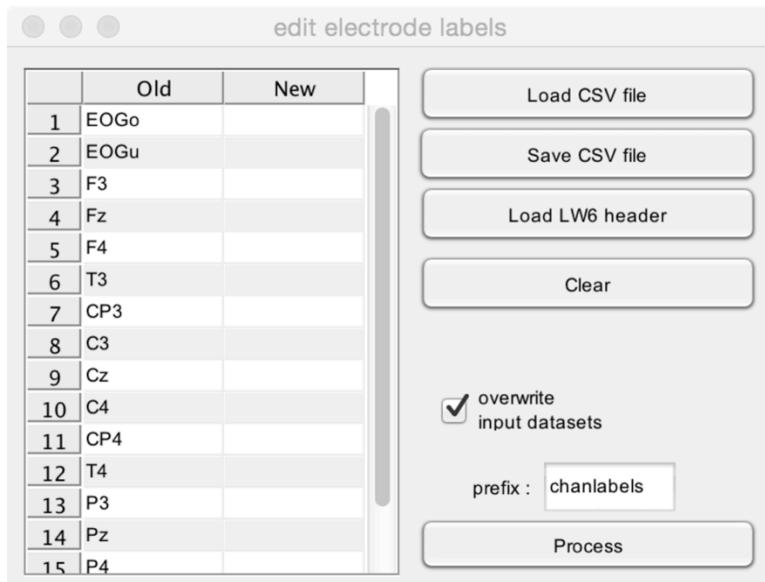
Edit electrode info. This function is used to visualize or clear electrode information.



- **Electrodes.** The list of electrodes in the dataset.
- **Clear info.** Clear all information stored for the currently selected channels.

## EDIT ELECTRODE LABELS

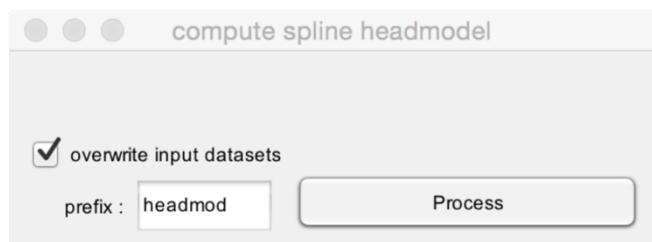
Edit electrode labels.



- **Old.** The original labels assigned to each channel of the dataset.
- **New.** The new labels assigned to each channel of the dataset. If the field is empty, the channel label will not be replaced.
- **Load CSV file.** Load labels from a CSV text file.
- **Save CSV file.** Save original labels to a CSV text file. You may then use this CSV file to assign new labels, and load the new labels using **Load CSV file**.
- **Load LW6 header.** Load the channel labels stored in another dataset.
- **Clear.** Clear all new labels.

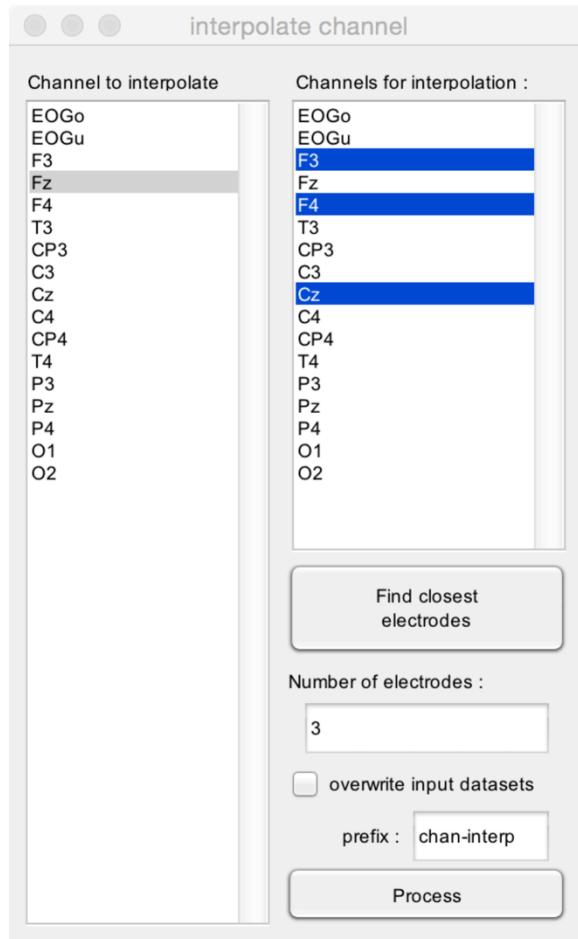
## COMPUTE SPLINE HEADMODEL

Compute spline head model. This is required to compute 3D scalp maps.  
Note that the spline headmodel requires that you first assign electrode coordinates.



## INTERPOLATE CHANNEL USING NEIGHBOURING ELECTRODES

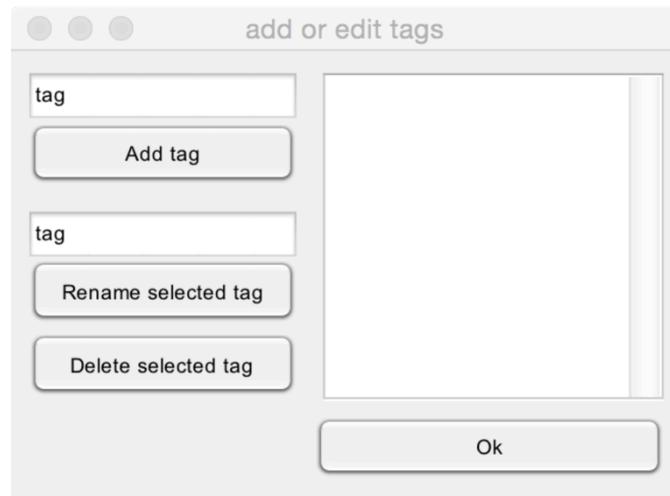
This function can be used to interpolate the data collected at a given electrode location using the average of the data collected at surrounding electrodes. This is useful to correct recordings in which the signal recorded at a specific channel contains a large number of artifacts.



- **Channel to interpolate.** The channel that must be interpolated (i.e. whose data must be replaced by the average of the data recorded from neighbouring electrodes).
- **Channels for interpolation.** The channels that must be used to reconstruct the signal at the channel to interpolate. Usually, it is suggested to use the average of the signals measured from three surrounding electrodes.
- **Find closest electrodes.** If your dataset has channel coordinates, this function will automatically select the electrodes most closely located in space to the location of the channel to interpolate. The number of electrodes to select is defined by **Number of electrodes**.

## ADD OR EDIT TAGS

It is possible to assign one or more tags to the datasets identifying, for example, the experimental condition to which they belong. Using the **T** button in the letswave interface, it is then possible to find all datasets having one or more combinations of tags.



All tags assigned to the selected dataset(s) are shown in the right listbox.

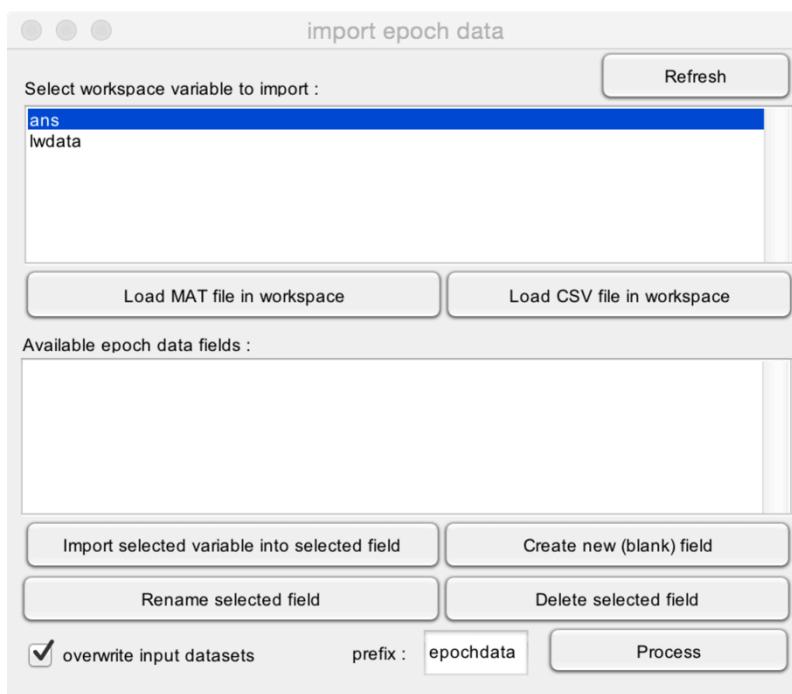
- **Add tag.** Add a new tag to the selected datasets.
- **Rename selected tag.** Rename the currently selected tag.
- **Delete selected tag.** Delete the currently selected tag.

## ADD OR EDIT EPOCH DATA

It is possible to assign user-defined variables to each epoch of the dataset. For example, this can be a variable with reaction times, ratings, quality of perception, or any other data that can be stored in a Matlab variable.

If some epochs are deleted at a later stage (e.g. artifact rejection), the epochdata variable will be automatically trimmed such as to keep only the data related to the remaining epochs.

The epochdata variable can also be used at a later stage to select epochs which satisfy a given condition (e.g. a reaction time < 500 ms; a rating >5, etc.).



To add epochdata to a dataset, the variable must be loaded/created in the Matlab workspace. All variables currently present in the workspace are shown in the upper listbox. Use the **Refresh** button to update its content.

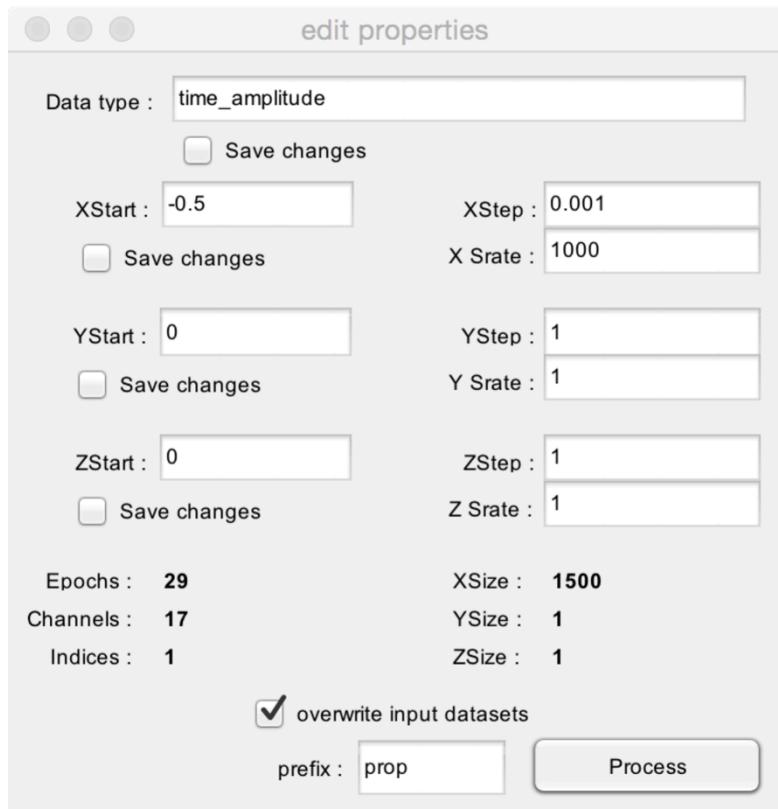
- **Load MAT file in workspace.** This opens a file dialog box, prompting you to load a MAT file in the Matlab workspace.
- **Load CSV file in workspace.** This opens a file dialog box, prompting you to load a CSV file in the Matlab workspace.

The lower listbox displays all the fields present in the current epochdata structure.

- **Import selected variable into selected field.** The variable selected from the Matlab workspace will be stored in the currently selected epochdata field.
- **Create new (blank) field.** Create a new field in the epochdata structure.
- **Rename selected field.** Rename the currently selected field in the epochdata structure.
- **Delete selected field.** Delete the currently selected field in the epochdata structure.

## EDIT PROPERTIES

Edit dataset basic properties.

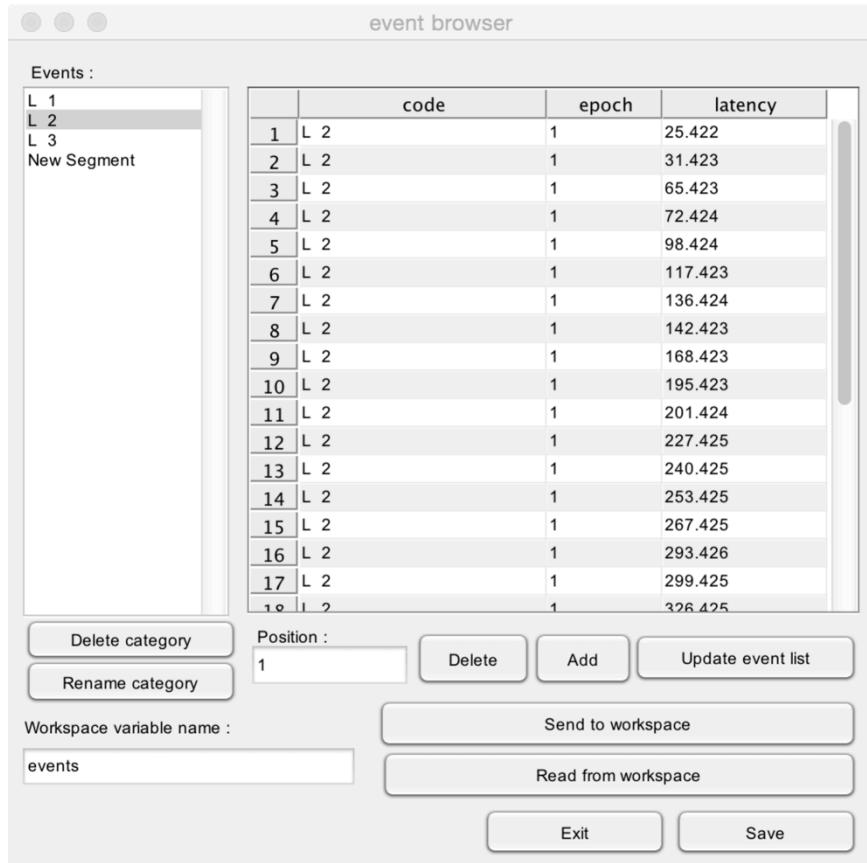


- **Data type.** Change the data type (e.g. time\_amplitude, frequency\_amplitude, frequency\_time\_amplitude).
- **XStart.** Change the value of the first sample along the X dimension (e.g. -0.5 s for epochs which extend between -0.5 s and +1.0 s).
- **XStep.** Change the interval between two samples along the X dimension (e.g. 0.001 s for epochs sampled at 1000 Hz).
- **X Rate.** Change the samplingrate of the X dimension.
- **YStart, YStep, Y Rate.** Change the values defining the Y dimension.
- **ZStart, ZStep, Z Rate.** Change the values defining the Z dimension.
- **Save Changes.** Values defining the X, Y and Z dimensions will be updated if and only if the Save Changes checkbox is checked.

## EVENTS

## BROWSE AND EDIT EVENTS

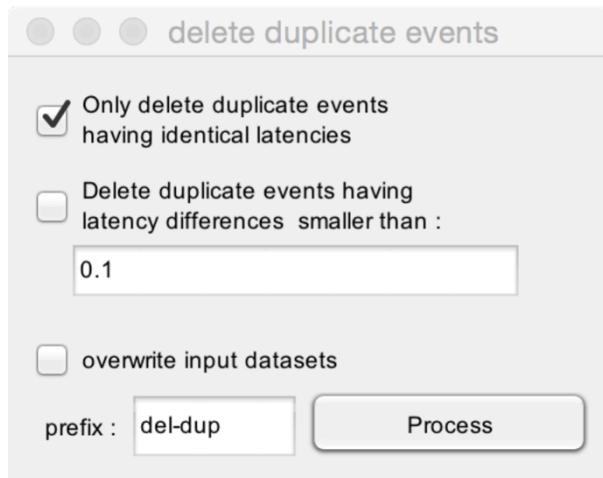
Browse and edit events stored in a dataset.



- **Events.** The list of available event codes are shown in the left listbox. If you select a given event code, all occurrences will be shown in the right table, including the epoch and latency of each occurrence.
- **Event table.** You can manually edit the event codes, epoch position and latencies by changing the values in the table. However, you must click **Update event list** to update the events of the dataset using the data displayed in the table.
- **Add.** Add an event at the current position.
- **Delete.** Delete the currently selected event.
- **Send to workspace.** Send all events as a variable to the Matlab workspace. This makes it easy to edit events using a script, or using Excel. The variable will be a cell array, with column 1 = code, column 2 = epoch and column 3 = latency.
- **Read from workspace.** Read all events from a variable stored in the Matlab workspace. The variable must be of the format : column 1 = code, column 2 = epoch, column 3 = latency.
- **Workspace variable name.** The name of the variable in the Matlab workspace used to **Send to workspace** or to **Read from workspace**.

## DELETE DUPLICATE EVENTS

Delete duplicate events.



If you select to **Only delete duplicate events having identical latencies**, the function will only delete duplicate events that have exactly the same latency.

If you select to **Delete duplicate events having latency differences smaller than** a given a value, the function will delete events that have the same event code, and occur close in time. The maximum latency difference is defined by the value entered in the corresponding edit box (e.g. 0.1 s). If matching events are found, the event occurring first is kept, and the event occurring last is discarded (e.g. if two events with the same event code occur respectively at 14.56 s and 14.59 s, the event occurring at 14.59 s will be discarded).

## CREATE EVENTS FROM LEVEL TRIGGER

Create events using a trigger sampled using the ascending or descending slope of an analog channel.

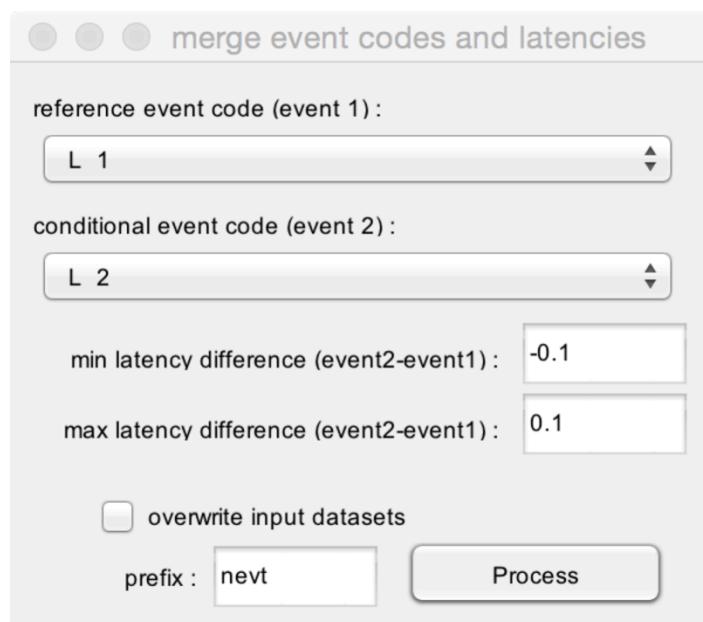


- **Threshold.** The value that must be reached in the signal to create an event.
- **Min ISI (s).** The minimum time interval between two triggers. This is to avoid creating more than one event for a single trigger.
- **Direction.** The direction of the analog trigger (detect an ascending or a descending slope).
- **Channel.** The label of the channel containing the analog trigger data.
- **Event code.** The event code label to assign to the events.

## MERGE EVENT CODES AND LATENCIES

In many experimental setups, it is useful to assign specific event codes to different types of events, for example, to distinguish between different experimental conditions. However, the method used to generate these different codes (e.g. a byte sent through the parallel port or the serial port to the EEG system) often have poor temporal precision. Therefore, it is sometimes also useful to record an additional uncoded trigger marking the actual onset of the event (e.g. a trigger obtained from an output connector of a sensory stimulator).

This function allows merging the latencies of a given series of events with the codes of another series of events.



- **Reference event code (event1).** The code of the events that must be used to store latency.
- **Conditional event code (event2).** The code of the events that must be used to assign the event code label.
- **Minimum latency difference (event2-event1).** The minimum value of the difference in latency between the conditional event code and the reference event code (e.g. -0.1 s means that the events will be merged if and only if the conditional event does not precede the reference event by more than 0.1 s).
- **Maximum latency difference (event2-event1).** The maximum value of the difference in latency between the conditional event code and the reference event code (e.g. +0.1 s means that the events will be merged if and only if the conditional event does not follow the reference event by more than 0.1 s).

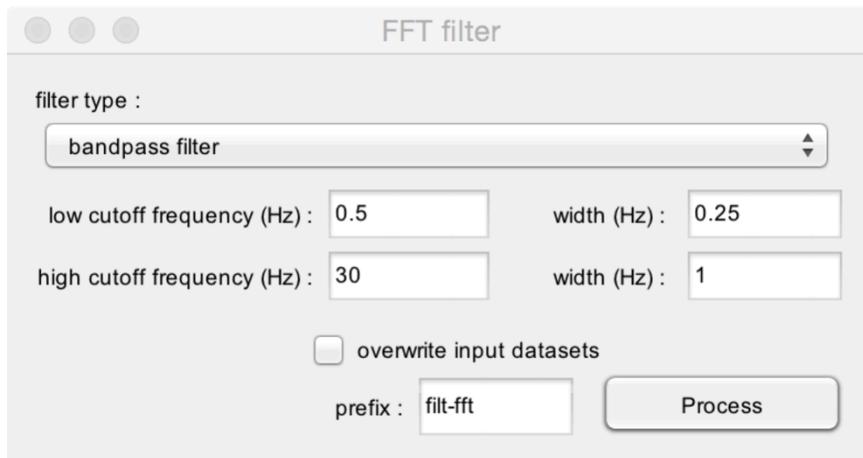
In this example, events with event codes corresponding to 'L 1' will be changed to event codes 'L 2' if and only if the latency of 'L 1' events equals the latency of an 'L 2' event  $\pm 0.1$  s.

## PREPROCESS

## FREQUENCY FILTERS

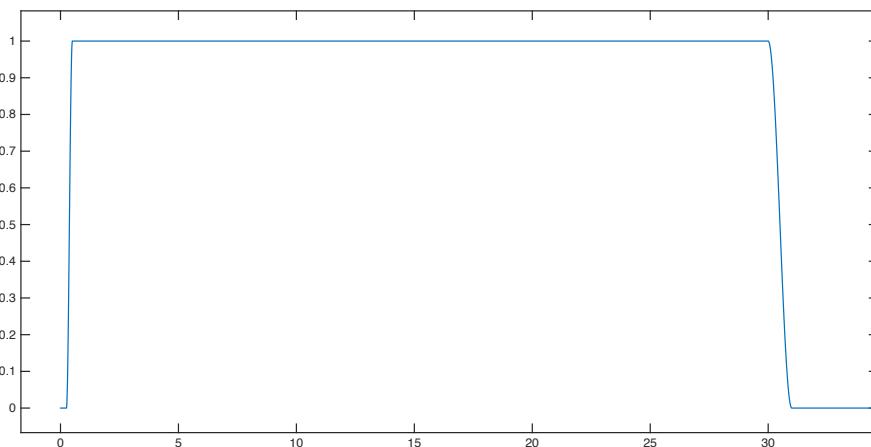
### FFT FILTER

Apply an FFT filter to the dataset(s).



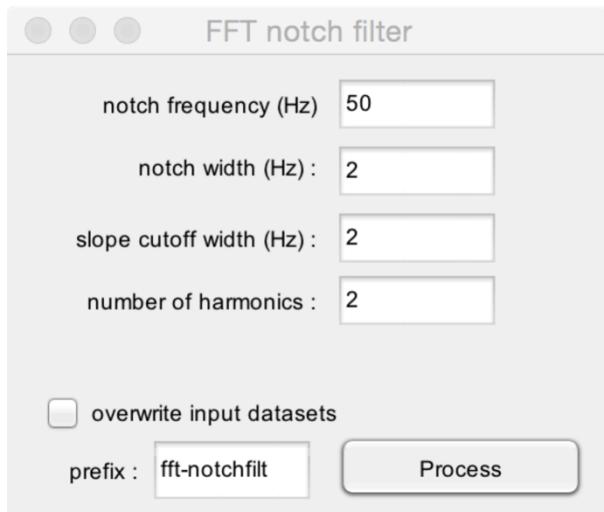
- **Filter type.** Choose between a bandpass filter, a low pass filter, a high pass filter, or a notch filter.
- **Low cutoff frequency (Hz).** The low frequency cutoff of the filter.
- **Low cutoff width (Hz).** The width of the low cutoff frequency. A Hanning window is used to design the cutoff transition.
- **High cutoff frequency (Hz).** The high frequency cutoff of the filter.
- **High cutoff width (Hz).** The width of the high cutoff frequency. A Hanning window is used to design the cutoff transition.

In this example, we apply a bandpass filter which will remove frequencies below 0.5 Hz and frequencies above 30 Hz. The width of the transition is 0.25 Hz for the low cutoff frequency and 1 Hz for the high cutoff frequency.



## FFT NOTCH FILTER

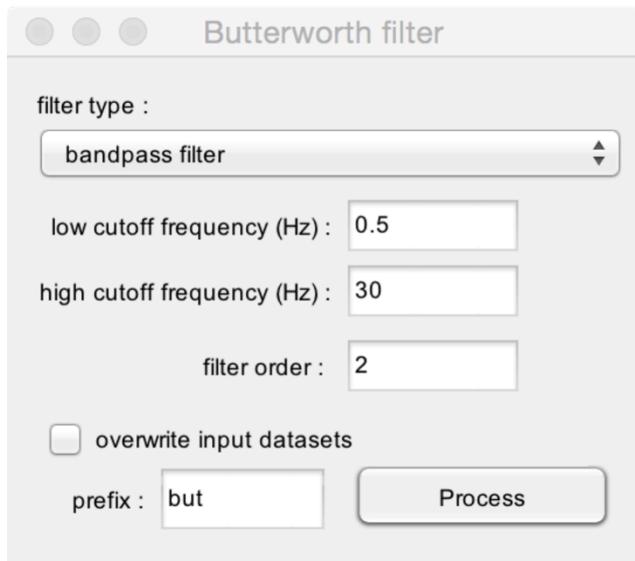
Apply a notch FFT filter to the dataset(s). In addition to removing the main frequency, the filter can also be used to remove N harmonics of that frequency.



- **Notch frequency.** The frequency to remove from the dataset (e.g. 50 Hz).
- **Notch width.** The width of the notch filter. (e.g. with a Notch Frequency = 50 Hz and a notch width = 2, the filter will remove all frequencies between 49-51 Hz).
- **Slope cutoff width.** The width of the transition cutoff, designed using a Hanning function.
- **Number of harmonics.** The number of harmonics to remove. (e.g. with a Notch Frequency = 2 and Number of Harmonics = 2, the filter will remove two frequencies: 50 Hz and 100 Hz).

## BUTTERWORTH FILTER

Apply a Butterworth filter to the dataset. Except for specific applications, it is recommended to use this filter for the preprocessing of continuous EEG datasets.

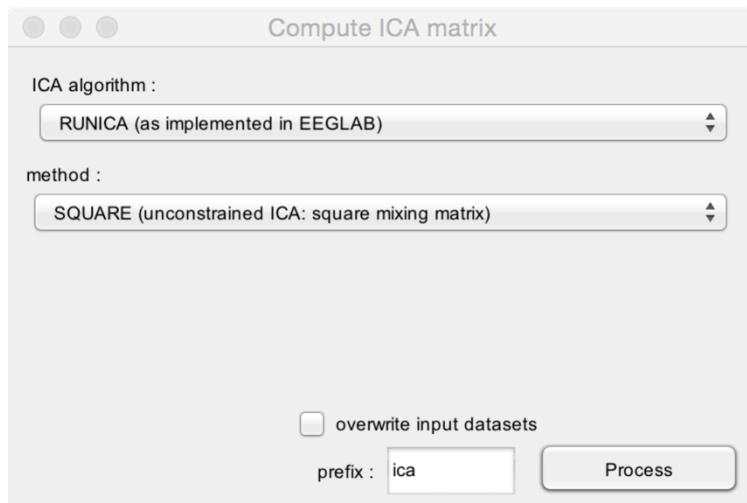


- **Filter type.** Choose between a bandpass filter, a low pass filter, a high pass filter, or a notch filter.
- **Low cutoff frequency (Hz).** The low frequency cutoff of the filter.
- **Low cutoff width (Hz).** The width of the low cutoff frequency. A Hanning window is used to design the cutoff transition.
- **High cutoff frequency (Hz).** The high frequency cutoff of the filter.
- **Filter order.** The order of the filter. Higher order will produce sharper filters, but may also introduce distortions in the signals.

## SPATIAL FILTERS

### COMPUTE ICA MATRIX

Compute an Independent Component Analysis (ICA). The ICA decomposition can be performed using the runica algorithm (as implemented in EEGLAB), or using the JADER algorithm. The mixing and unmixing matrices are stored in the dataset header.



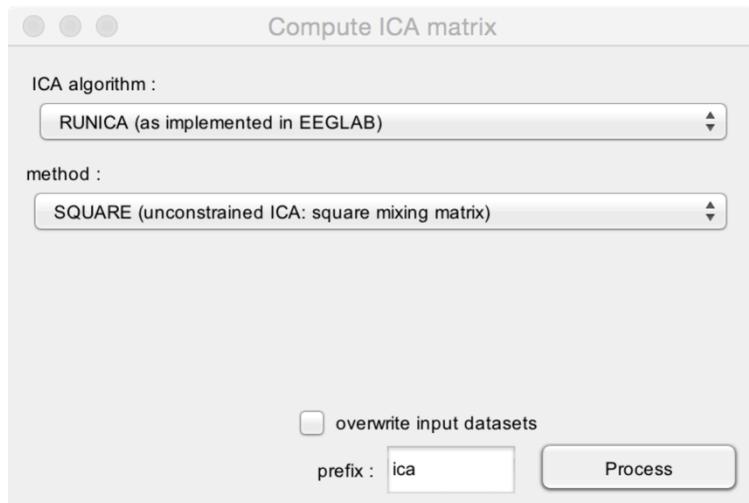
- **ICA algorithm.** Choose between the RUNICA algorithm (as implemented in EEGLAB; <http://sccn.ucsd.edu/eeglab/>) or the JADER algorithm (<http://perso.telecom-paristech.fr/~cardoso/guidesepsou.html>).
- **Method.** The following methods can be used to constrain ICA:
  - **SQUARE (unconstrained ICA).** ICA will be performed using a square matrix. The number of ICs will equal the number of channels. This is recommended when ICA is used to remove eyeblink artifacts. Note that, in some cases where the number of independent sources is far lower than the number of channels, this can lead to overfitting and the creation of spurious ICs. In these cases, it is recommended to reduce the number of ICs (see also <http://www.ncbi.nlm.nih.gov/pubmed/14964560>).
  - **USER-DEFINED constrained ICA.** ICA will be performed using a rectangular matrix. The number of ICs is defined by the user.
  - **PROBABILISTIC-ICA.** Several methods are implemented to automatically estimate the number of independent sources in the data, and to perform ICA using this estimate. This two-step method is referred to as Probabilistic ICA (PICA). Available methods are Laplacian Estimate (LAP), Bayesian Information Criterion (BIC), Rajan & Rayner (RRN), AIC and MDL (see also <http://www.ncbi.nlm.nih.gov/pubmed/14964560>). Note that in our experience, good results have been obtained using the RRN approach.
- **Percentage of PICA estimate.** If you are using PROBABILISTIC ICA, it is possible to increase or decrease the number of estimated ICs, expressed as a percentage of the estimate provided by the algorithm. This is useful to check that the obtained results are not critically dependent on the number of estimated ICs.

## COMPUTE ICA MATRIX (MERGED)

This function is identical to the COMPUTE ICA MATRIX function except for the fact that ICA is performed after merging multiple datasets. The obtained ICA matrix is then assigned to each original dataset.

This is useful if you wish to obtain a single matrix for multiple datasets that have all been obtained during the same recording.

Note that this is valid if and only if it is reasonable to assume that all the datasets can be explained by a single set of independent components, each projecting identically to the different channels across datasets.

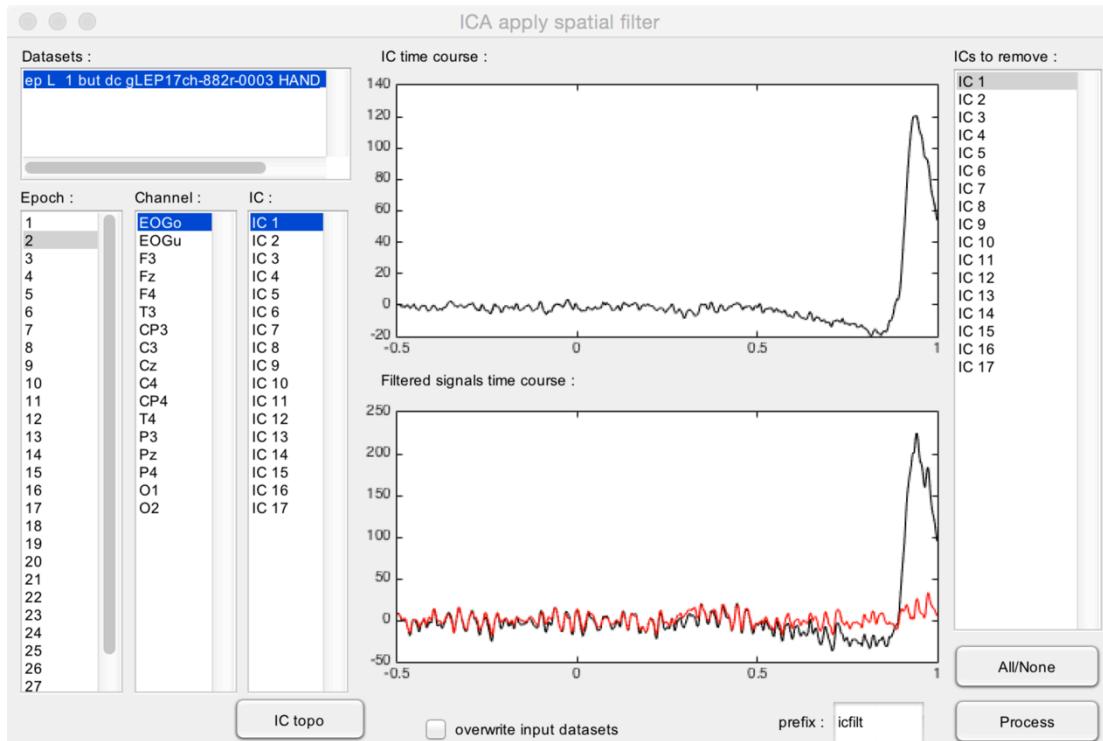


- **ICA algorithm.** See above.
- **Method.** See above.
- **Percentage of PICa estimate.** See above.

## APPLY ICA FILTER

Interactive GUI to remove independent components, e.g. independent components capturing artifacts such as eye blinks or muscle activity.

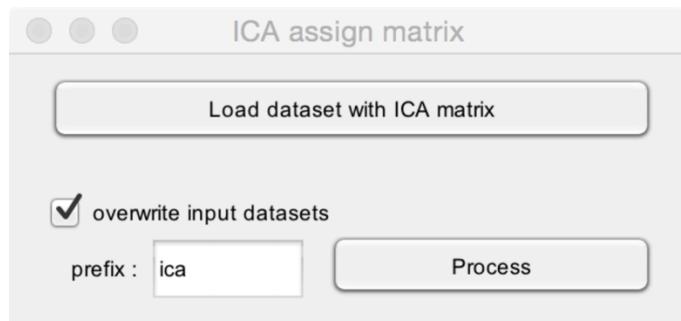
Note that this function requires computing the ICA first (see Compute ICA matrix).



- **Datasets.** It is possible to process several datasets at the same time, as long as they have the same associated ICA matrix.
- The **upper graph** shows the IC time course for the selected dataset and epoch. Use the **IC listbox** to change the ICs shown in this graph. In this example, we can see that IC1 captures eye movement artifacts, as evidenced by the large eye blink artifact occurring approximately 0.8 s after stimulus onset.
- The **lower graph** shows the EEG time course for the selected dataset and epoch. The black waveform corresponds to the unfiltered signal. The red waveform corresponds to the filtered signal (i.e. the signal obtained after removing the ICs selected in the **ICs to remove** listbox). In this example, you can see that removing IC1 efficiently removes the eye blink artifact. You can change the displayed channel and epoch using the **Channel** and **Epoch** listbox.
- **IC topo.** Plot the scalp topography of the selected ICs.
- **All/None.** Select or unselect all ICs in the **ICs to remove** listbox.

## ASSIGN ICA MATRIX

Assign an existing matrix of a given dataset to another dataset.



- **Load dataset with ICA matrix.** Select a dataset which already has an associated ICA matrix.

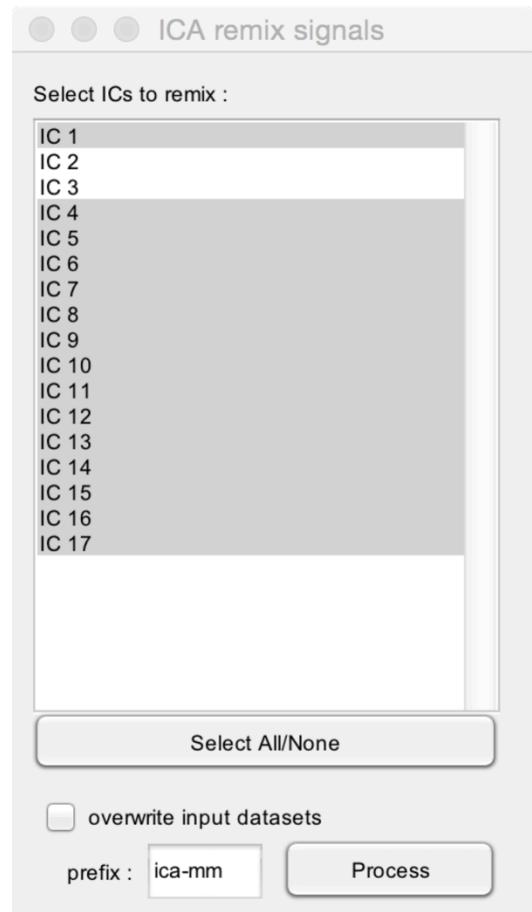
## ICA UNMIX SIGNALS

Unmix original signals into a set of independent components (ICs).  
This requires a dataset with an associated ICA matrix.



## ICA REMIX SIGNALS

Remix independent components (ICs) into mixed signals.

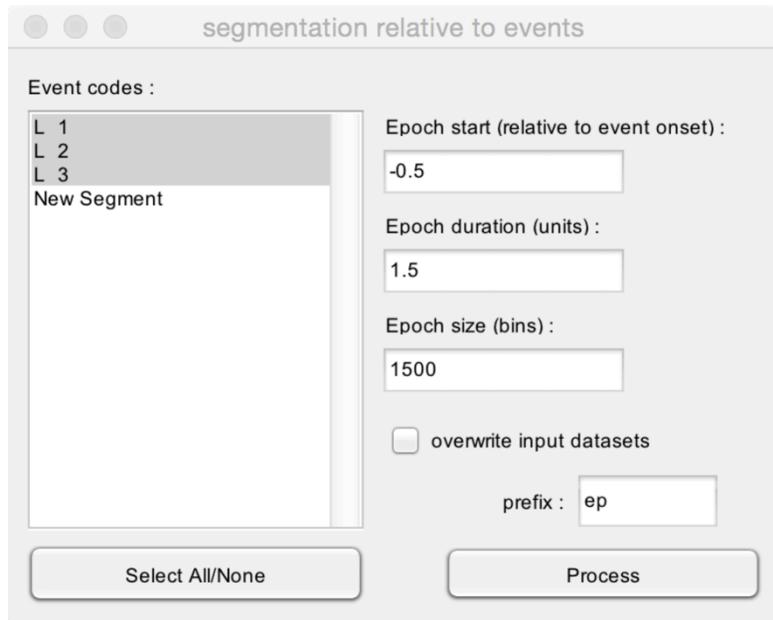


- **Select ICs to remix.** It is possible to remix only a selection of independent components (ICs), for example, to only keep ICs which do not contain an artifact. In this example, the remixed signals will include the signals contained in all ICs except IC2 and IC3.

## SEGMENTATION

### SEGMENTATION RELATIVE TO EVENTS

Segment a continuous dataset into a series of epochs of a given length, relative to the latencies of events.

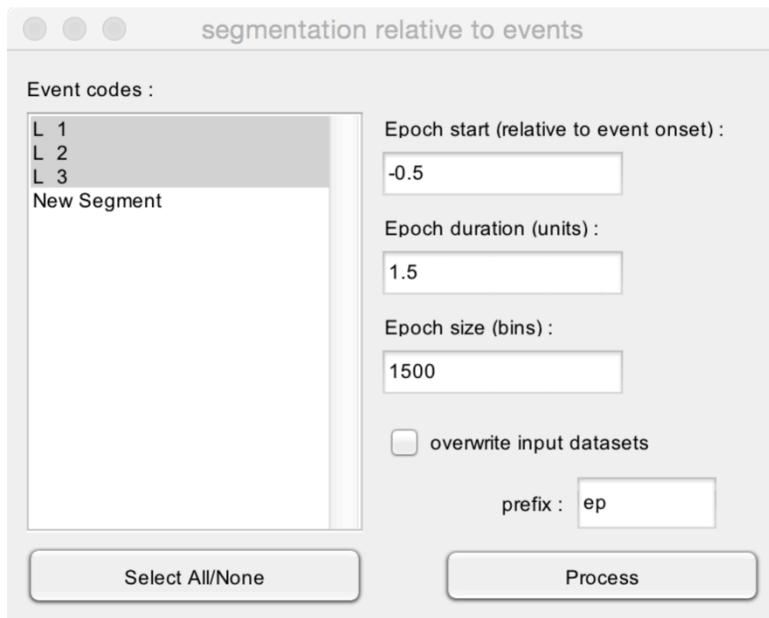


- **Event codes.** Choose the event codes to include. Epochs will be segmented relative to all events having matching event codes. If you select more than one event code, the output will be a single dataset including all epochs occurring around the selected event codes.
- **Epoch start.** The time of the first sample of each epoch, expressed relative to the event onset.
- **Epoch duration.** The duration of each epoch.
- **Epoch size.** The number of samples in each epoch. This value is dependent on the samplingrate and epoch duration.

In this example, the continuous dataset has 30 events corresponding to 'L 1', 30 events corresponding to 'L 2' and 30 events corresponding to 'L 3'. The output dataset will thus have a total of 90 epochs (30 epochs with L\_1, 30 epochs with L\_2 and 30 epochs with L\_3). Each epoch will have 1500 samples, extending from -0.5 s to +1 s relative to the onset of each event.

## SEGMENTATION RELATIVE TO EVENTS (ONE FILE PER EVENT CODE)

Segment a continuous dataset into a series of epochs of a given length, relative to the latencies of events. If multiple event codes are selected, the function will generate a separate dataset for each event code.

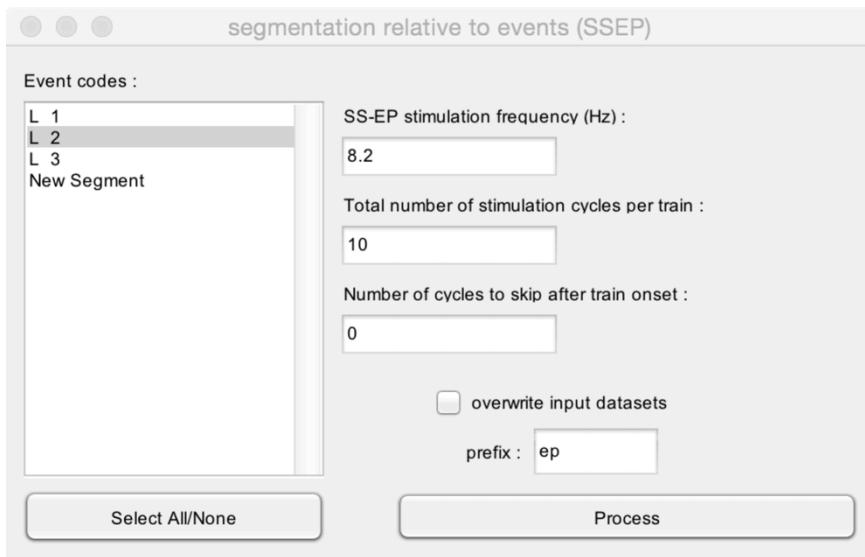


- **Event codes.** Choose the event codes to include. Epochs will be segmented relative to all events having matching event codes. If you select more than one event code, the output will be multiple datasets: one dataset including all epochs occurring around each selected event code).
- **Epoch start.** The time of the first sample of each epoch, expressed relative to the event onset.
- **Epoch duration.** The duration of each epoch.
- **Epoch size.** The number of samples in each epoch. This value is dependent on the samplingrate and epoch duration.

In this example, the continuous dataset has 30 events corresponding to 'L 1', 30 events corresponding to 'L 2' and 30 events corresponding to 'L 3'. The output will be three datasets, each having 30 epochs. Each epoch will have 1500 samples, extending from -0.5 s to +1 s relative to the onset of each event.

## SEGMENTATION RELATIVE TO EVENTS (SS-EPS)

This function is used to segment continuous data. It is specifically designed to segment data obtained in experiments recording steady-state evoked potentials (SS-EPs).

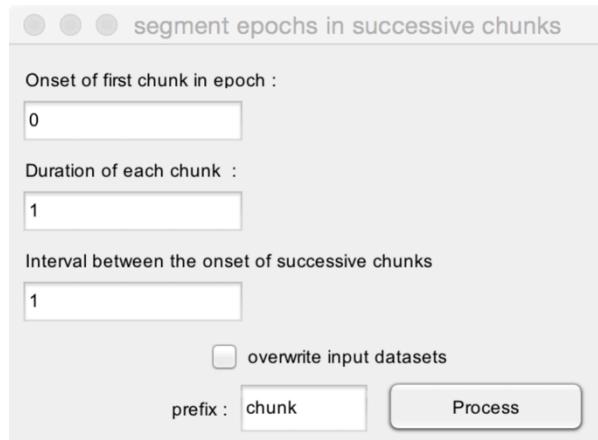


- **Event codes.** Choose the event codes to include. Epochs will be segmented relative to all events having matching event codes. If you select more than one event code, the output will be multiple datasets: one dataset including all epochs occurring around each selected event code).
- **SS-EP stimulation frequency.** The frequency of SS-EP stimulation.
- **Total number of stimulation cycles per train.** The number of times the stimulus is repeated in each stimulation train.
- **Number of cycles to skip after train onset.** In some experiments, it is interesting to discard the signal recorded during the first cycles of the stimulation train, such as to avoid a contribution of activity related to the train onset. This field allows removing these initial cycles.

In this example, a visual stimulus is flashed at a frequency of 8.2 Hz. The total number of repetitions is 10. Hence, a train of stimulation lasts  $10/8.2=1.22$  s. Because we have chosen to not exclude initial cycles, the function will segment the data into epochs extending between 0 and 1.22 s relative to the onsets of event L\_2.

## SEGMENT EPOCHS IN SUCCESSIVE CHUNKS

This function is used to segment epochs into successive chunks of data. This is useful to segment EEG epochs during which a stimulus is repeated periodically (steady-state evoked potentials) into epochs circumscribing each event of a train.

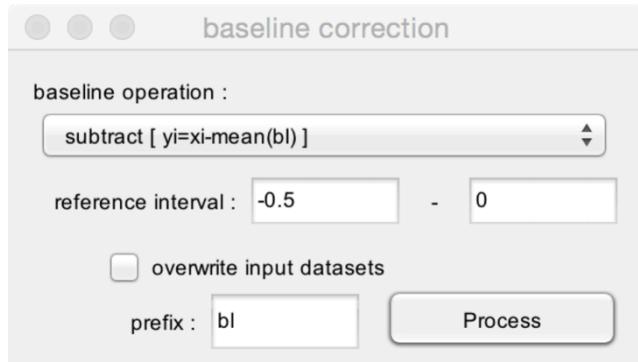


- **Onset of first chunk in epoch.** The latency of the first repeated event in the epoch. For example, if the train starts at latency = 0 s, the onset of the first chunk in the epoch should be set to 0 s.
- **Duration of each chunk.** The duration of each chunk. For example, if your stimulus is repeated periodically at a frequency of 1 Hz, you may want to cut the epoch into chunks lasting 1 s.
- **Interval between the onset of successive chunks.** The time interval between two events in the train. For example, if your stimulus is repeated periodically at a frequency of 1 Hz, you will want to set the time interval to 1 s.

## BASELINE OPERATIONS

### BASELINE CORRECTION

This function is used to apply a baseline correction to epoched EEG data. This allows expressing signals relative to a reference interval (e.g. a time interval before the onset of the stimulus).

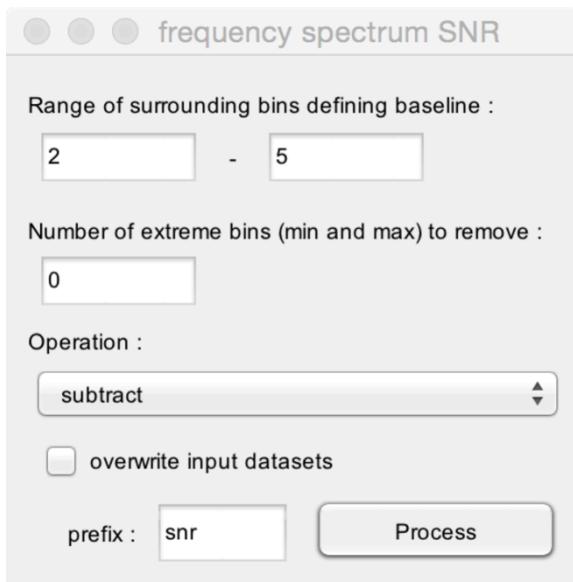


- **Baseline operation.** Several operations are available : **subtract** (remove the mean of the signal amplitude measured during the reference interval), **ER%** (express signal amplitude as the percentage of change relative to the mean of the signal amplitude measured during the reference interval), **divide** (express the ratio between signal amplitude and the mean of the signal amplitude measured during the reference interval) and **zscore** (express the signal as standard deviation relative to the distribution of the signal measured during the reference interval).
- **Reference interval.** The begin and end of the reference interval. For example, the reference interval can be defined as the 0.5 s preceding the onset of the stimulus (begin : -0.5 S, end: 0.0 s).

## FREQUENCY SPECTRUM SNR

This function is used to express the magnitude of steady-state evoked potentials (SS-EPs) identified in frequency spectra relative to the amplitude of the spectra obtained at neighbouring frequency bins. The function can be viewed as a method to apply a baseline correction in the frequency domain.

See also : <http://www.ncbi.nlm.nih.gov/pubmed/21753000>

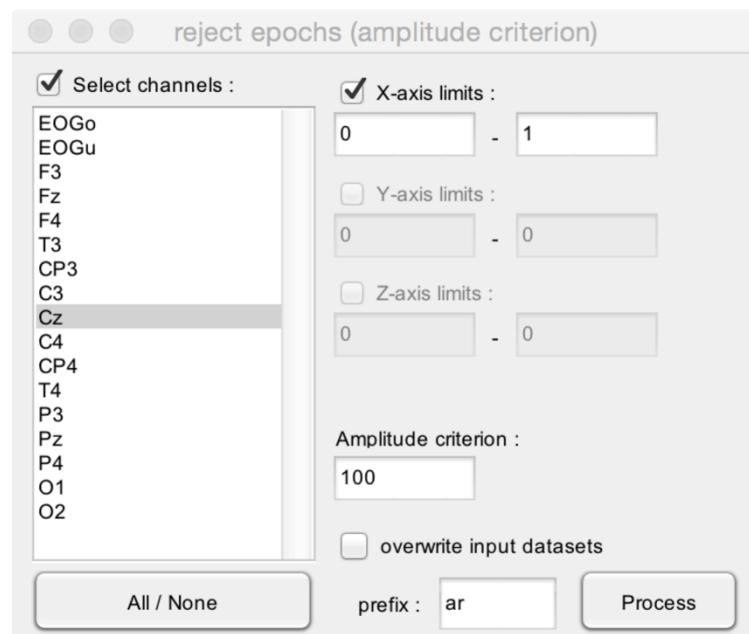


- **Range of surrounding bins defining baseline.** Each bin of the frequency spectrum will be expressed relative to the amplitude of the frequency spectrum obtained at surrounding bins. This parameter defines the range of surrounding frequency bins that should be used to compute the baseline operation. In this example, each bin ( $i$ ) will be expressed relative to bins ( $i-2$  to  $i-5$  and  $i+2$  to  $i+5$ ). When defining this interval, it is important to take into consideration the frequency resolution of your FFT, as well as the spread of your response (SS-EP) in the frequency domain (Is it restricted to a single frequency bin? does it spread to several frequency bins?)
- **Number of extreme bins to remove (min and max).** This option allows discarding extreme values in the reference interval (defined by the range of surrounding bins). This can be useful to avoid effects due to the presence of artifacts in the surrounding frequency bins.
- **Operation.** Several operations can be applied: **subtract** (subtract the mean of the reference interval), **SNR** (express as the ratio of the reference interval), **zscore** (express as the standard deviation relative to the distribution of the reference interval), **percent** (express as percentage of difference relative to the mean of the reference interval).

## ARTIFACTS

### REJECT EPOCHS (AMPLITUDE CRITERION)

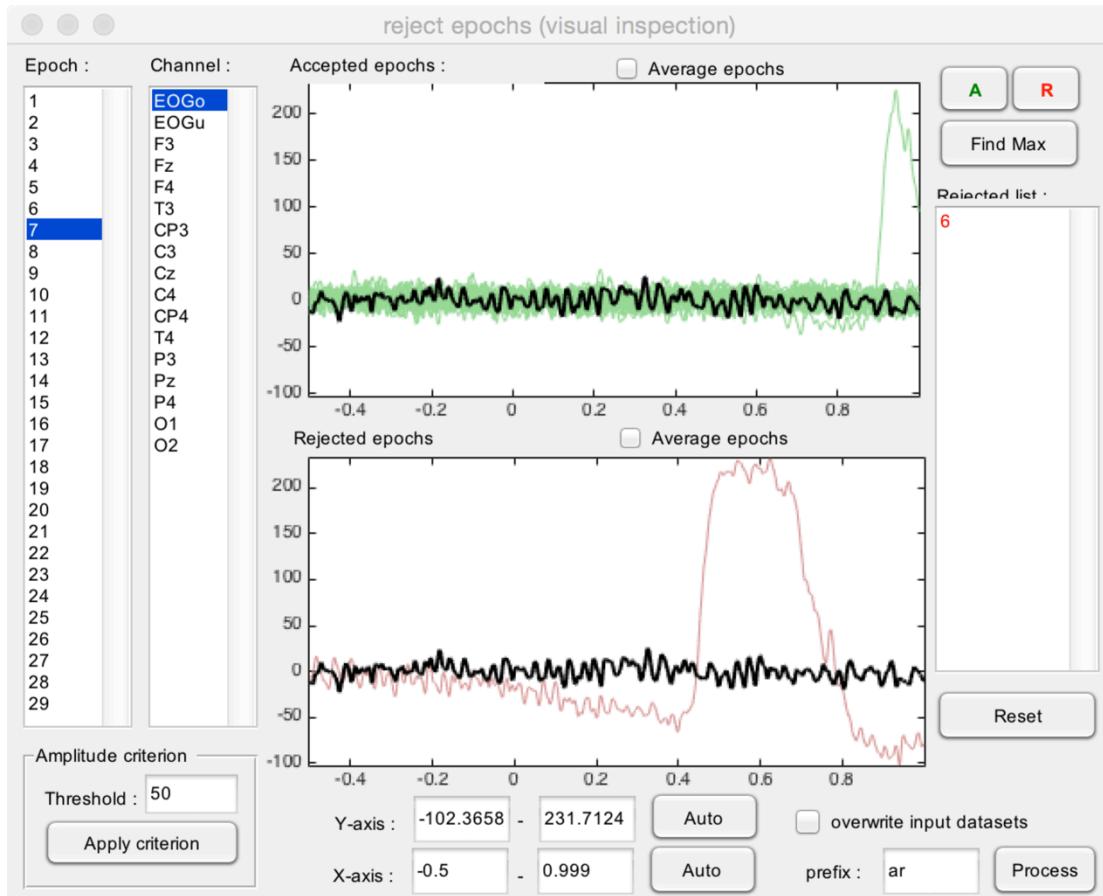
This function is used to reject artifacted epochs based on an amplitude criterion.



- **Amplitude criterion.** The amplitude criterion. If, in a given epoch, the absolute value of the signal exceeds the amplitude criterion, the epoch will be rejected.
- **Select channels.** When checked, rejection of epochs based on amplitude criterion will be only performed using the signals measured at the channels selected in the Channels listbox. In this example, the amplitude criterion will only be applied to the signals measured at electrode Cz.
- **X-axis limits.** When X-axis limits is checked, the amplitude criterion will only be applied to the signal measured during the defined time interval. In this example, the amplitude criterion will only be applied to the signal measured between 0 s and 1 s relative to stimulus onset.
- **Y-axis limits, Z-axis limits.** For 2D (e.g. time-frequency maps) or 3D data.

## REJECT EPOCHS (VISUAL INSPECTION)

This interactive GUI allows rejecting epochs using visual inspection.



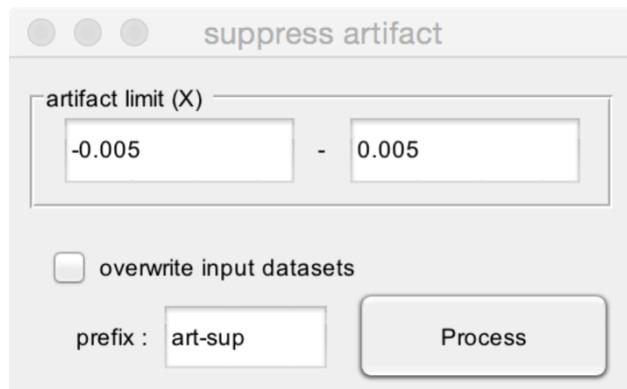
The upper graph displays all **accepted** epochs superimposed in green. The lower graph displays all **rejected** epochs superimposed in red. If **Average epochs** is checked, instead of displaying superimposed single-trial waveforms, the graphs display the average of accepted epochs and the average of rejected epochs.

The left **Epoch** listbox selects a given epoch. The selected epoch is displayed as a thick black waveform in both graphs. The **Channel** listbox selects the channel to be displayed in the graphs.

- **A, R.** The **A** button defines the currently selected epochs as ‘Accepted’. The **R** button defines the currently selected epochs as ‘Rejected’. Selected epochs appear in the **Rejected list**.
- **Find Max.** Automatically selects the epoch displaying the greatest amplitude (i.e., in most cases, the epoch most likely to contain an artifact).
- **Reset.** Clears the list of rejected epochs.
- **Amplitude criterion.** This can be used to automatically select epochs with absolute values exceeding a given amplitude criterion. The criterion will be applied to the signals measured at the selected channel.
- **X-axis and Y-axis.** The scales of the X-axis and Y-axis can be defined manually by editing the X- and Y-axis range. It can also be adjusted automatically using the **Auto** buttons.

## SUPPRESS ARTIFACT (LATENCY DEFINED)

This function is used to suppress artifacts always occurring at the same latency (e.g. a stimulation artifact occurring at latency = 0 s).



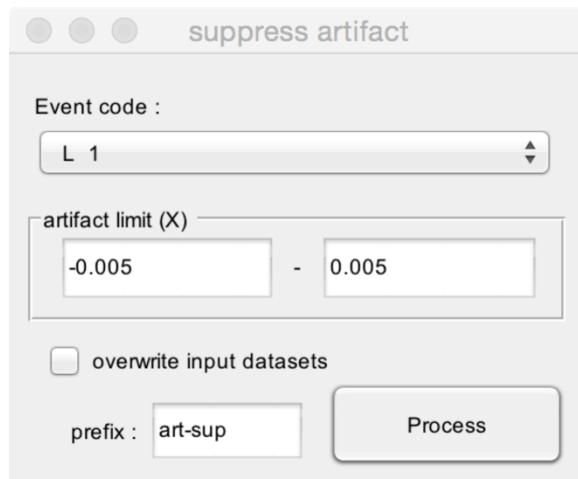
The function will interpolate the values defined by the artifact limits, using the values of the signal measured just before the beginning of the artifact limit, and the signal measured just after the end of the artifact limit.

The interpolation performed is a linear interpolation.

- **Artifact limit.** The time interval during which the artifact occurs.

## SUPPRESS ARTIFACT (EVENT DEFINED)

This function is used to suppress artifacts always occurring at the same latency relative to an event in continuous data (e.g. a stimulation artifact occurring at latency = 0 s relative to a given event).



The function will interpolate the values defined by the artifact limits, using the values of the signal measured just before the beginning of the artifact limit, and the signal measured just after the end of the artifact limit.

The interpolation performed is a linear interpolation.

- **Event code.** The code of the events defining the latency of the artifacts.
- **Artifact limit.** The time interval during which the artifact occurs.

## CURRENT SOURCE DENSITY (CSD)

### COMPUTE SCALP CSD

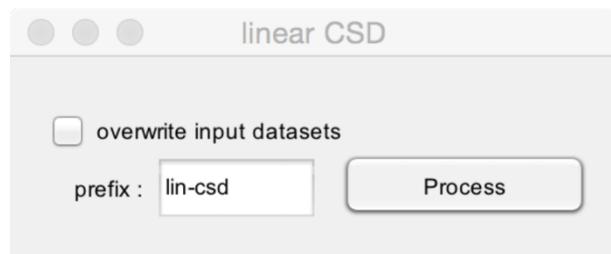
Compute Scalp CSD (spherical spline algorithm to compute scalp surface Laplacian or current source density (CSD) estimates for surface potentials.

See : <http://psychophysiology.cpmc.columbia.edu/Software/CSDtoolbox/>



## COMPUTE LINEAR CSD

This function is used to compute a linear CSD.



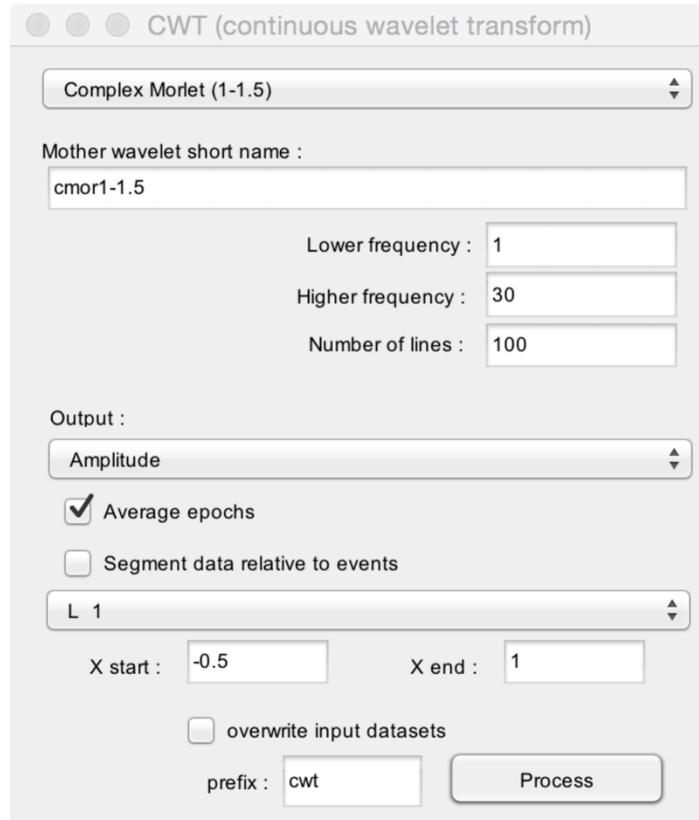
The signal of each channel is expressed as the difference between the signal at that channel and the mean of the signal measured at the two neighbouring channels. For example, the signal at channel 2 becomes the signal at channel 2 – the mean of the signals at channels 1 and 3.

Note that the first and last channels of the dataset are discarded.

## FREQUENCY TRANSFORMS

### CONTINUOUS WAVELET TRANSFORM (CWT)

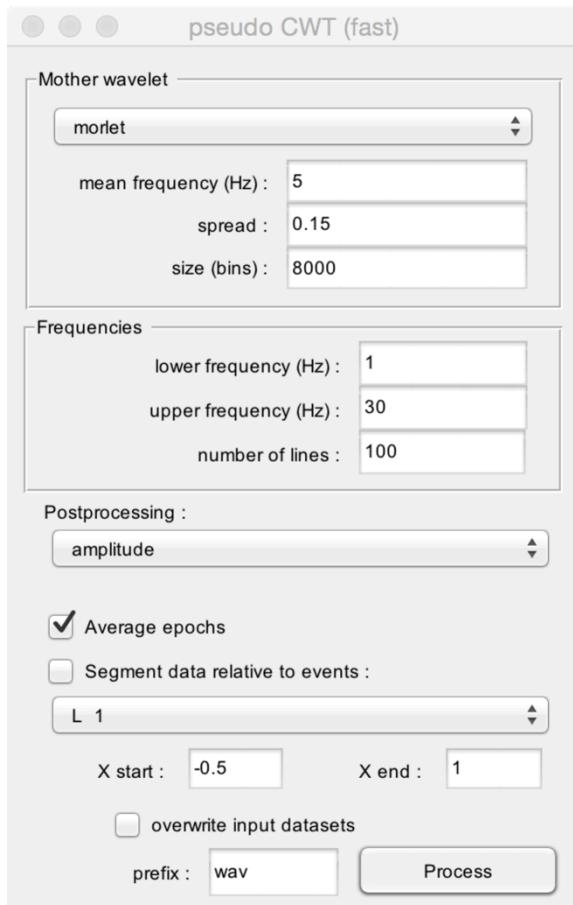
Compute a time-frequency transform using the Continuous Wavelet Transform as implemented in Matlab.



- **Mother wavelet short name.** The name of the Mother wavelet. See Matlab for details. A selection of wavelet names can be chosen from the dropdown list.
- **Lower frequency.** The lowest frequency line to estimate.
- **Higher frequency.** The highest frequency line to estimate.
- **Number of lines.** The number of frequency lines to estimate. In this example, the time-frequency map will have 100 lines, extending between 1 and 30 Hz.
- **Output.** Output the **amplitude**, **power**, **phase**, or **complex** value of the CWT.
- **Average epochs.** If you are applying the time-frequency transform to single trials such as to obtain a time-frequency map of the average power or amplitude as a function of time and frequency, you may choose to **average epochs** such as to avoid storing a very large dataset containing the time-frequency maps of each single trial. Also, if you are applying the time-frequency transform to single trials such as to obtain a time-frequency map of **phase locking values**, you should choose **phase** as output and choose to **average epochs**.
- **Segment data relative to events.** If you are applying the CWT to a continuous EEG dataset, you can choose to segment the obtained time-frequency maps relative to the onset of a given event and, eventually, to average these epochs to obtain average time-frequency maps. The **X start** and **Y start** fields define the onset and offset of the epoch segmentation. Applying the CWT to continuous data is interesting because it avoids border effects due to zero-padding. However, this will be possible if and only if your system is equipped with a large amount of memory.

## FAST CWT

Compute a time-frequency transform using a custom Continuous Wavelet Transform algorithm. The time required to compute the transform is shorter than the time required to compute the transform using the Matlab CWT.



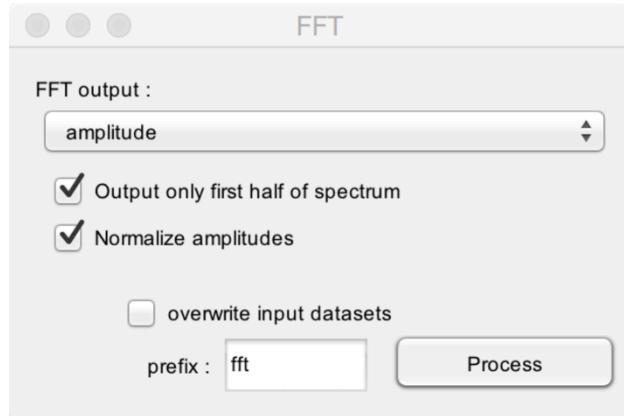
- **Mother wavelet.** The CWT can be computed using a complex **Morlet** wavelet, or a complex **Hanning** wavelet.
- **Mean frequency.** The mean frequency of the mother wavelet.
- **Spread.** The spread in time of the mother wavelet. The **mean frequency** and **spread** parameters are linked, and define the number of cycles included in the wavelet localizer. Increasing the mean frequency and/or increasing the spread will increase the overall frequency resolution but decrease the temporal resolution of the time-frequency transform. Decreasing the mean frequency and/or decreasing the spread will decrease the overall frequency resolution, but increase the overall time resolution.
- **Size.** The number of bins used to generate the Mother wavelet. It is not recommended to change this parameter, unless you understand how it may impact your results (see the RLW\_fastCWT.m function).
- **Lower frequency.** The lowest frequency line to estimate.
- **Higher frequency.** The highest frequency line to estimate.
- **Number of lines.** The number of frequency lines to estimate. In this example, the time-frequency map will have 100 lines, extending between 1 and 30 Hz.
- **Postprocessing.** Output the **amplitude**, **power**, **phase**, or **complex** value of the CWT.
- **Average epochs.** If you are applying the time-frequency transform to single trials such as to obtain a time-frequency map of the average power or amplitude as a function of time and frequency, you may choose to **average epochs** such as to avoid storing a very large dataset

containing the time-frequency maps of each single trial. Also, if you are applying the time-frequency transform to single trials such as to obtain a time-frequency map of **phase locking values**, you should choose **phase** as output and choose to **average epochs**.

- **Segment data relative to events.** If you are applying the CWT to a continuous EEG dataset, you can choose to segment the obtained time-frequency maps relative to the onset of a given event and, eventually, to average these epochs to obtain average time-frequency maps. The **X start** and **Y start** fields define the onset and offset of the epoch segmentation. Applying the CWT to continuous data is interesting because it avoids border effects due to zero-padding. However, this will be possible if and only if your system is equipped with a large amount of memory.

## FAST FOURIER TRANSFORM (FFT)

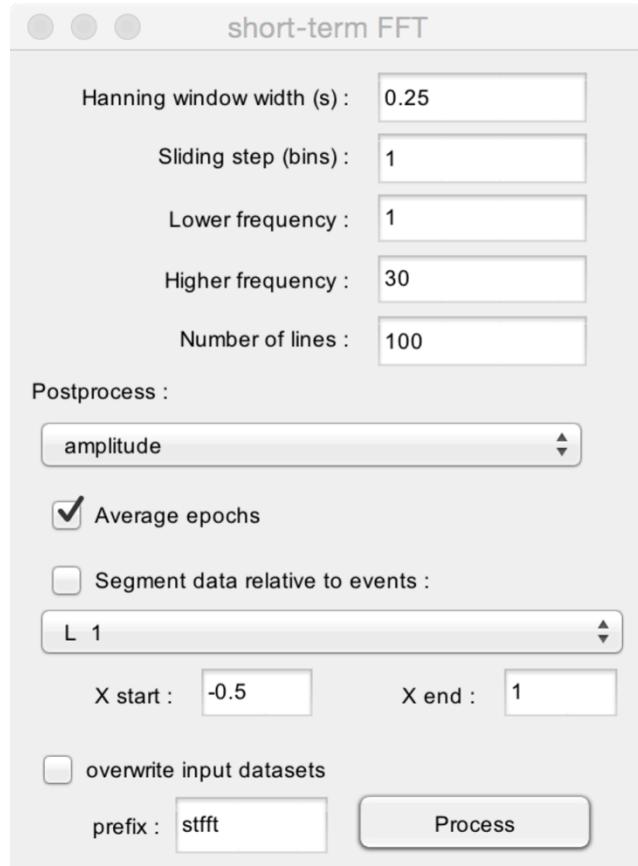
Compute a Fast Fourier Transform (FFT).



- **FFT output.** After computing the FFT, the result can be used to express the frequency spectrum of the **amplitude**, **phase**, **real part**, **imaginary part**, or **complex number**.
- **Output only first half of spectrum.** If you are using the FFT to assess the amplitude or power spectrum of your signal, only the first half of the spectrum is needed, as the second half is a mirror image of the first half. Indeed, for a real signal, the FFT spectrum will be conjugate symmetric which means that the second half of the spectrum is a mirror image of the first half centred around the Nyquist frequency, (sampling rate / 2).
- **Normalize amplitudes.** The results of fft must be divided by N/2, where N is the length of data. This is required to match the amplitudes of the FFT spectrum with those of the original signal.

## SHORT-TERM FFT (ST-FFT)

Compute a time-frequency transform using a short-term FFT, as implemented in Matlab.

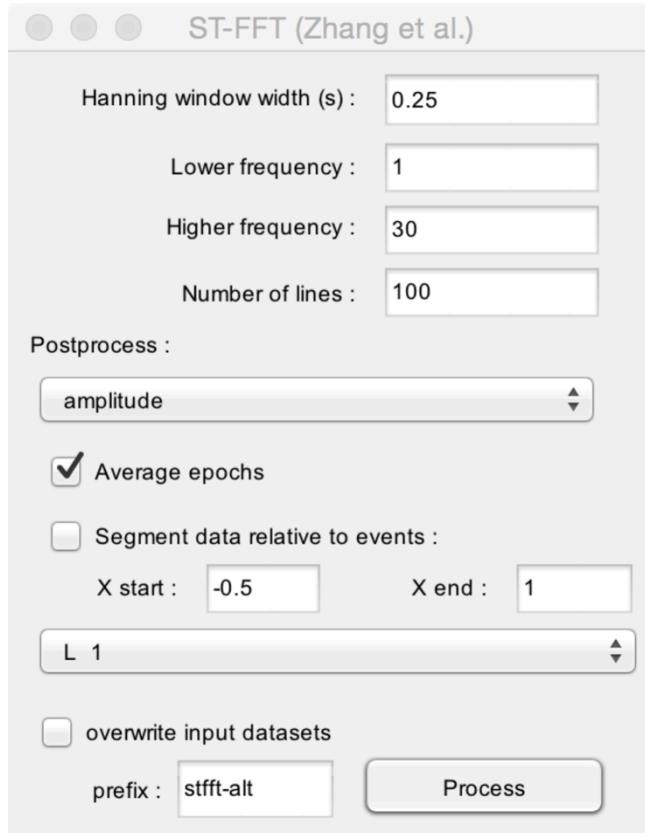


- **Hanning window width (s).** The width of the Hanning window used to compute the ST-FFT. Increasing the window width will increase the frequency resolution and decrease the temporal resolution. Decreasing the window width will decrease the frequency resolution and increase the temporal resolution.
- **Sliding step.** The default value is 1. In that case, the FFT will be computed for each bin (sample) of the time series. If you increase this value, the FFT will be computed every N bins. For example, if your time series has 1000 samples, the time-frequency map will have 1000 time samples if the sliding step = 1, and will have 500 samples if the sliding step = 2.
- **Lower frequency.** The lower frequency line to estimate.
- **Higher frequency.** The highest frequency line to estimate.
- **Number of lines.** The number of frequency lines to estimate. In this example, the time-frequency map will have 100 lines, extending between 1 and 30 Hz.
- **Postprocess.** Output the **amplitude**, **power**, **phase**, or **complex** value of the CWT.
- **Average epochs.** If you are applying the time-frequency transform to single trials such as to obtain a time-frequency map of the average power or amplitude as a function of time and frequency, you may choose to **average epochs** such as to avoid storing a very large dataset containing the time-frequency maps of each single trial. Also, if you are applying the time-frequency transform to single trials such as to obtain a time-frequency map of **phase locking values**, you should choose **phase** as output and choose to **average epochs**.
- **Segment data relative to events.** If you are applying the CWT to a continuous EEG dataset, you can choose to segment the obtained time-frequency maps relative to the onset of a given event and, eventually, to average these epochs to obtain average time-frequency

maps. The **X start** and **Y start** fields define the onset and offset of the epoch segmentation. Applying the CWT to continuous data is interesting because it avoids border effects due to zero-padding. However, this will be possible if and only if your system is equipped with a large amount of memory.

## SHORT-TERM FFT (ZHANG ET AL.)

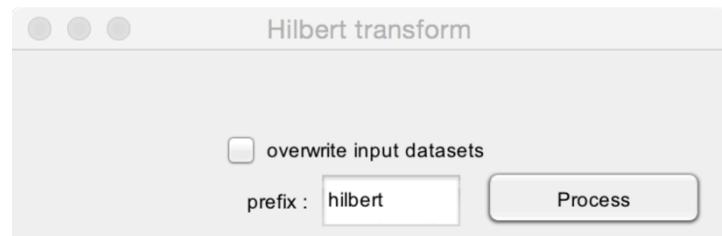
Compute a time-frequency transform using the short-term FFT implemented in Zhang et al. (2012). See <http://www.ncbi.nlm.nih.gov/pubmed/22649223>.



- **Hanning window width (s).** The width of the Hanning window used to compute the ST-FFT. Increasing the window width will increase the frequency resolution and decrease the temporal resolution. Decreasing the window width will decrease the frequency resolution and increase the temporal resolution.
- **Lower frequency.** The lower frequency line to estimate.
- **Higher frequency.** The highest frequency line to estimate.
- **Number of lines.** The number of frequency lines to estimate. In this example, the time-frequency map will have 100 lines, extending between 1 and 30 Hz.
- **Postprocess.** Output the **amplitude**, **power**, **phase**, or **complex** value of the CWT.
- **Average epochs.** If you are applying the time-frequency transform to single trials such as to obtain a time-frequency map of the average power or amplitude as a function of time and frequency, you may choose to **average epochs** such as to avoid storing a very large dataset containing the time-frequency maps of each single trial. Also, if you are applying the time-frequency transform to single trials such as to obtain a time-frequency map of **phase locking values**, you should choose **phase** as output and choose to **average epochs**.
- **Segment data relative to events.** If you are applying the CWT to a continuous EEG dataset, you can choose to segment the obtained time-frequency maps relative to the onset of a given event and, eventually, to average these epochs to obtain average time-frequency maps. The **X start** and **Y start** fields define the onset and offset of the epoch segmentation. Applying the CWT to continuous data is interesting because it avoids border effects due to zero-padding. However, this will be possible if and only if your system is equipped with a large amount of memory.

## HILBERT TRANSFORM

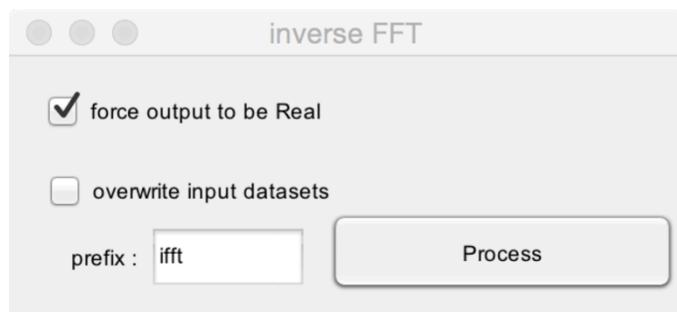
Compute the Hilbert transform.



## INVERSE FFT

Compute the inverse FFT transform.

This requires a dataset with **complex** FFT values.

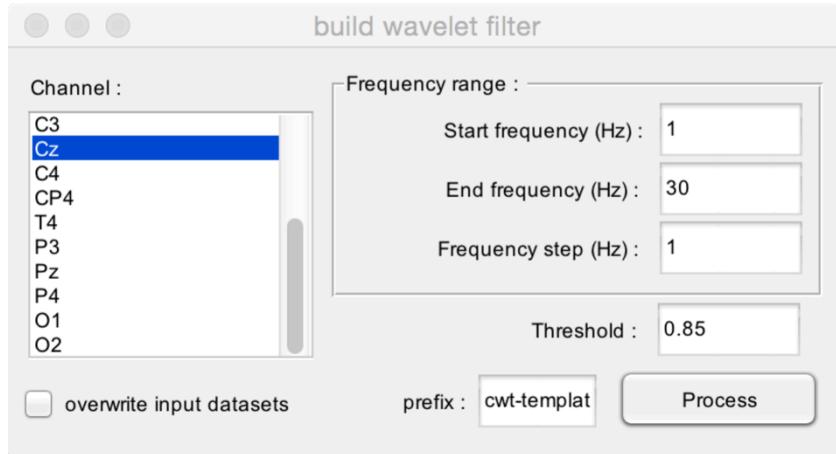


- **Force the output to be real.** Due to slight imprecision due to rounding, the inverse FFT of the FFT of a real signal may, in some cases, not return real values. To ensure that the output are real values, check this option.

## TIME-FREQUENCY FILTERS

### BUILD WAVELET FILTER (HU ET AL)

Build a time-frequency filter using a continuous wavelet transform, as implemented in Hu et al. 2012. See <http://www.ncbi.nlm.nih.gov/pubmed/20004255>.

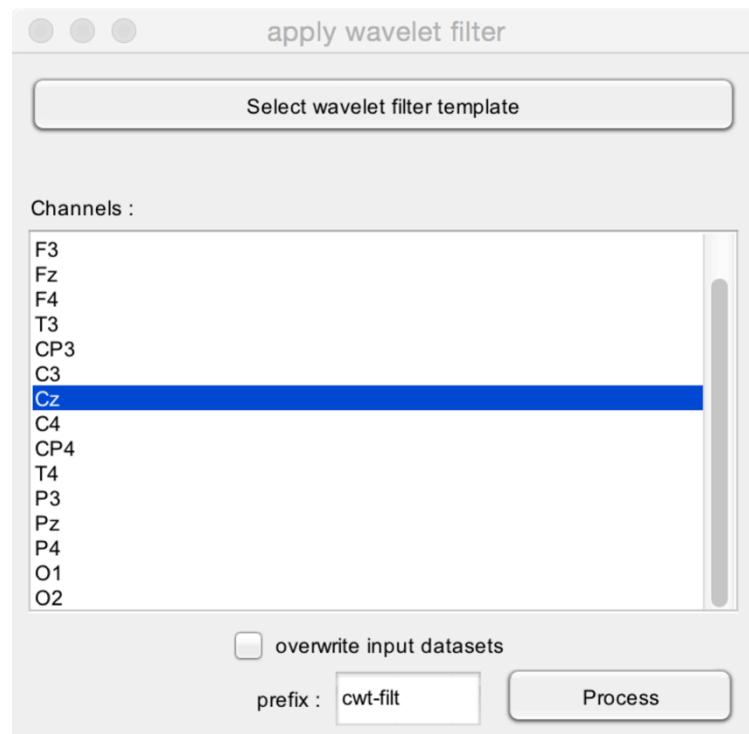


- **Channel.** Separate time-frequency filters can be built for a selection of channels. Select one or more channels for which a time-frequency filter template should be computed. Note that applying the filter is very time consuming. Therefore, it is recommended to restrict the filter procedure to a single channel.
- **Frequency range.** The frequency range to include in the time-frequency filter. The frequency range is defined by setting the **Start frequency** and **End frequency** (i.e. the lower and upper limits of the time-frequency transform). The number of lines of the time-frequency transform is defined by adjusting the **Frequency step**. Increasing the frequency step will reduce the number of frequency lines to compute and, hence, reduce computing time, but also reduce the quality of the filter. Increasing the frequency step will increase the number of frequency lines, increase computing time, but also increase the quality of the filter
- **Threshold.** The threshold used to define which bins of the time-frequency maps should be kept, and which should be rejected (considered as noise). Increasing this value will exclude more bins and, possibly, remove more noise (but may also remove some signal). Reducing this value will exclude less bins and, possibly, remove less noise (but also retain more signal). See <http://www.ncbi.nlm.nih.gov/pubmed/20004255> for details.

## APPLY WAVELET FILTER (HU ET AL)

Apply a time-frequency filter computed using the Build Wavelet Filter function, as implemented in Hu et al. 2012.

See <http://www.ncbi.nlm.nih.gov/pubmed/20004255>.

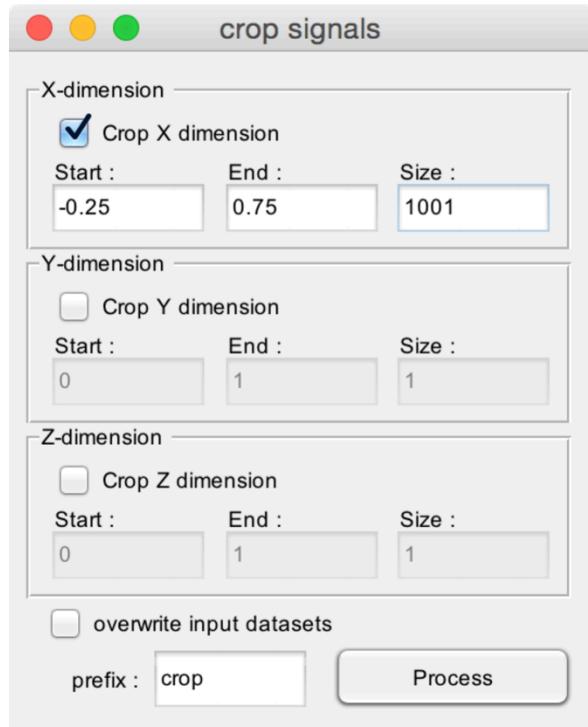


- **Select wavelet filter template.** Select the dataset containing the wavelet filter template.
- **Channels.** Select the channel(s) onto which the time-frequency filter template should be applied.

## RESAMPLE SIGNALS

### CROP SIGNALS

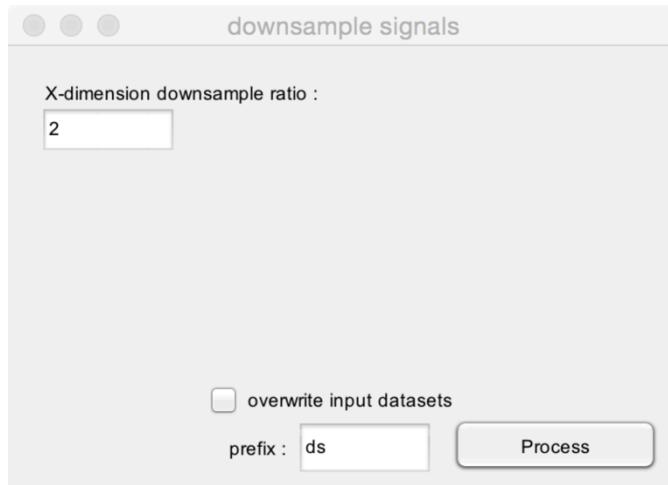
Crop signals along the X, Y and/or Z dimension(s).



- **Crop X-dimension.** Crop signals along the X-dimension.
  - **Start.** The lower limit of the cropped signals along the X-dimension.
  - **End.** The upper limit of the cropped signals along the X-dimension. For example, setting Start to -0.25 and End to +0.75 will generate a new dataset with epochs extending between -0.25 and +0.75 s
  - **Size.** The number of bins (samples) in the X-dimension.
- **Crop Y-dimension.** Crop signals along the Y dimension.
- **Crop Z-dimension.** Crop signals along the Z dimension.

## DOWNSAMPLE SIGNALS

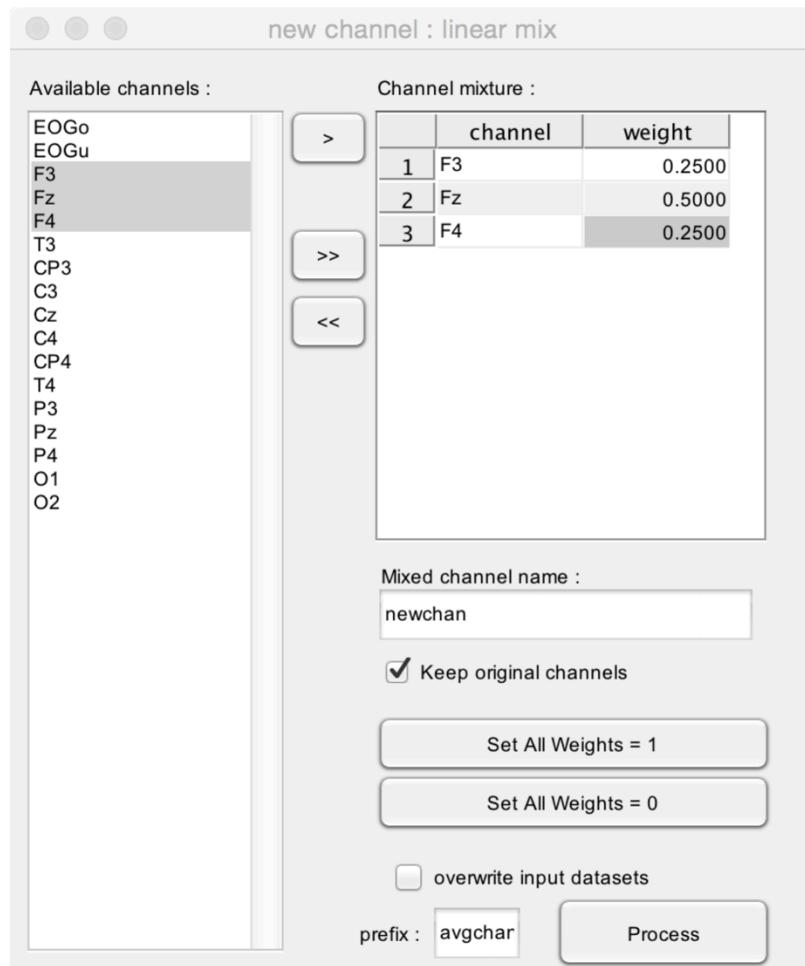
Downsample signals along the X, Y and/or Z dimension(s).



- **X-dimension downsample ratio.** The integer value by which the X-dimension sampling rate should be downsampled. For example, if the samplingrate is 1000 Hz and the downsample ratio is 2, the output dataset will have a sample rate of  $1000/2=500$  Hz.
- **Y-dimension downsample ratio.** If your dataset has a Y-dimension, this defines the downsample ratio for the Y dimension. Note that a downsample ratio of 1 will leave the samplingrate unchanged.
- **Z-dimension downsample ratio.** If your dataset has a Z-dimension, this defines the downsample ratio for the Z dimension. Note that a downsample ratio of 1 will leave the samplingrate unchanged.

## NEW CHANNEL : LINEAR MIX

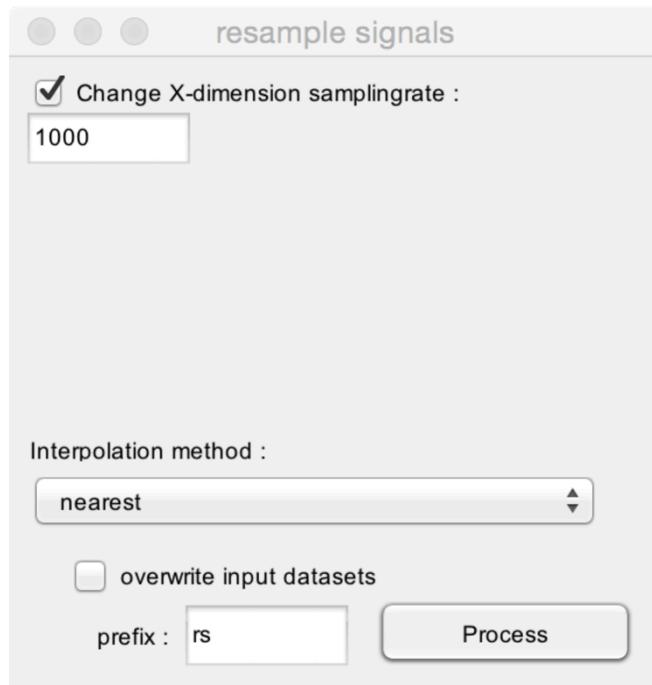
Create a new channel, based on a linear mixture of existing channels.



- **Channel mixture.** The new channel is defined by setting this table. The > button allows adding selected channels to the channel mixture. The >> button adds all available channels to the mixture. The << button removes all channels from the mixture. You can then set individual weights for each channel. In this example, the new channel will be a mixture of the signals measured at channels F3, FZ and F4 : NEWCHAN = 0.25\*F3 + 0.5\*FZ + 0.25\*F4.
- **Mixed channel name.** The name of the new mixed channel.
- **Keep original channels.** When checked, the new channel will be appended to the existing channels. When unchecked, the output dataset will only contain the new channel.
- **Set all Weights = 1.** Set all weights to 1.
- **Set all Weights = 0.** Set all weights to 0.

## RESAMPLE SIGNALS

Resample signals using a user-defined samplingrate. The signals will be interpolated using one of several available interpolation methods.

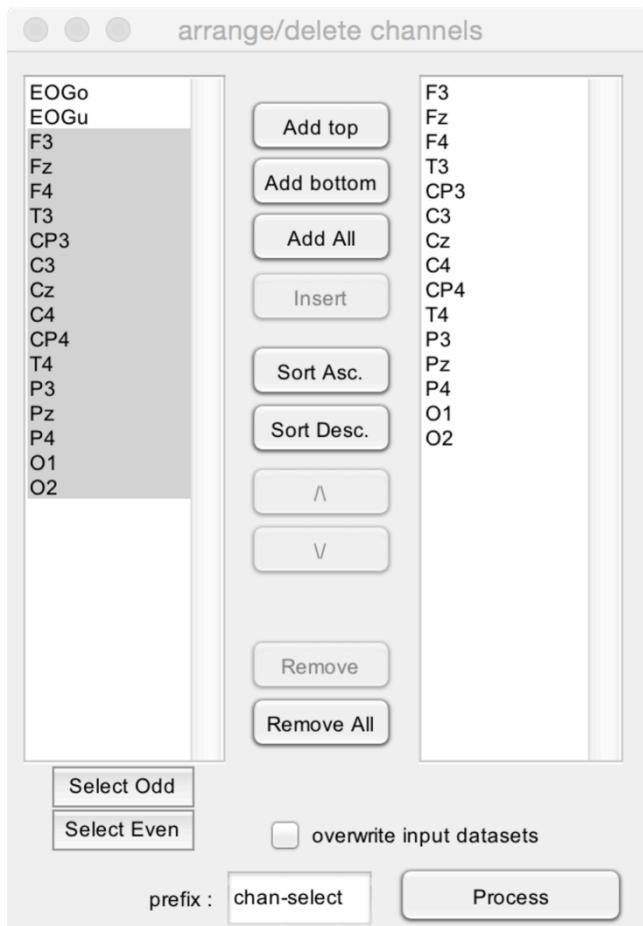


- **Change X-dimension samplingrate.** If checked, the function will resample the X-dimension, such that the samplingrate of the output dataset will correspond to the value entered in the samplerate field. In this example, the dataset will be resampled to reach a sampling rate of 1000 Hz.
- **Change Y-dimension samplingrate.** This option is available if the dataset contains a Y dimension.
- **Change Z-dimension samplingrate.** This option is available if the dataset contains a Z dimension.
- **Interpolation method.** Several methods can be used for interpolation. See <http://www.mathworks.com/help/vision/ug/interpolation-methods.html> for details.

## ARRANGE SIGNALS

### REARRANGE/DELETE CHANNELS

Arrange the order or delete channels in a dataset.

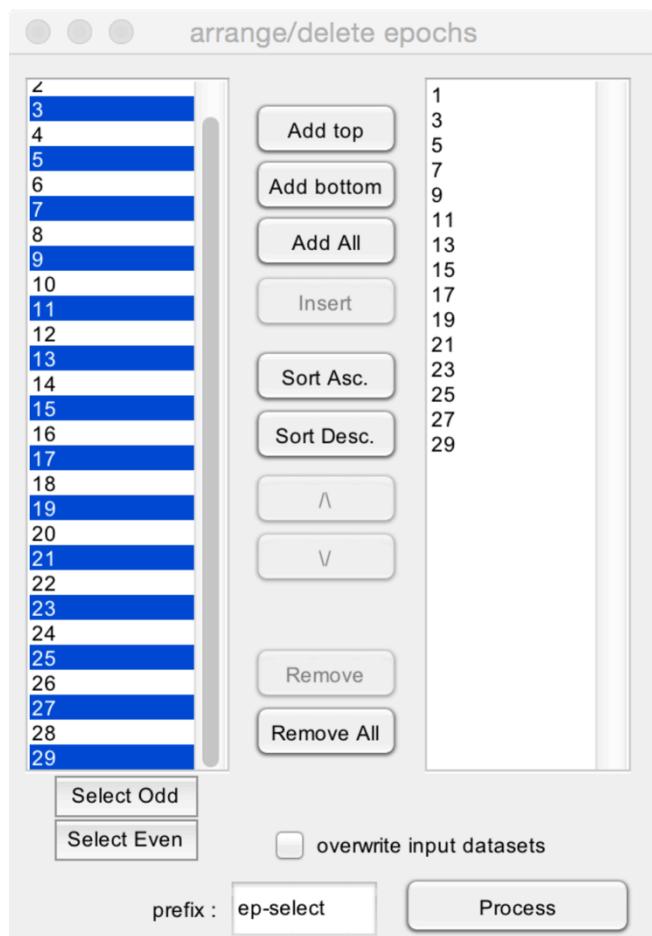


The left listbox shows all available channels. The right listbox shows the channels of the output dataset.

- **Add top.** Add the channels selected in the list of available channels to the top of the list of output channels.
- **Add bottom.** Add the channels selected in the list of available channels to the bottom (end) of the list of output channels.
- **Add all.** Add all available channels to the list of output channels.
- **Insert.** Insert the channels selected in the list of available channels to the position selected in the list of output channels.
- **Sort ascending.** Sort the channels in alphabetical order (ascending).
- **Sort descending.** Sort the channels in alphabetical order (descending).
- The **Up** and **Down** arrows can be used to change the order of the output channels.
- **Remove.** Remove the selected channels from the output listbox.
- **Remove all.** Remove all channels from the output listbox.
- **Select odd.** Select all odd-numbered channels.
- **Select even.** Select all even-numbered channels.

## REARRANGE/DELETE EPOCHS

Arrange the order or delete epochs in a dataset.

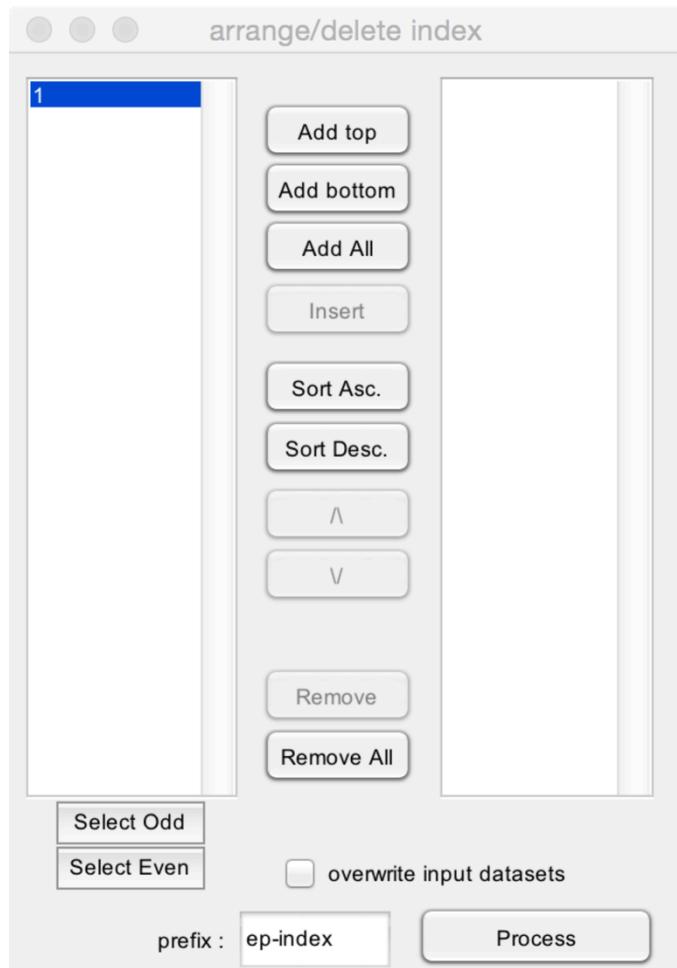


The left listbox shows all available epochs. The right listbox shows the epochs of the output dataset.

- **Add top.** Add the epochs selected in the list of available channels to the top of the list of output epochs.
- **Add bottom.** Add the epochs selected in the list of available epochs to the bottom (end) of the list of output epochs.
- **Add all.** Add all available epochs to the list of output epochs.
- **Insert.** Insert the epochs selected in the list of available epochs to the position selected in the list of output epochs.
- **Sort ascending.** Sort the epochs in alphabetical order (ascending).
- **Sort descending.** Sort the epochs in alphabetical order (descending).
- The **Up** and **Down** arrows can be used to change the order of the output epochs.
- **Remove.** Remove the selected epochs from the output listbox.
- **Remove all.** Remove all epochs from the output listbox.
- **Select odd.** Select all odd-numbered epochs.
- **Select even.** Select all even-numbered epochs.

## REARRANGE/DELETE INDEX

Arrange the order or delete indexes in a dataset.

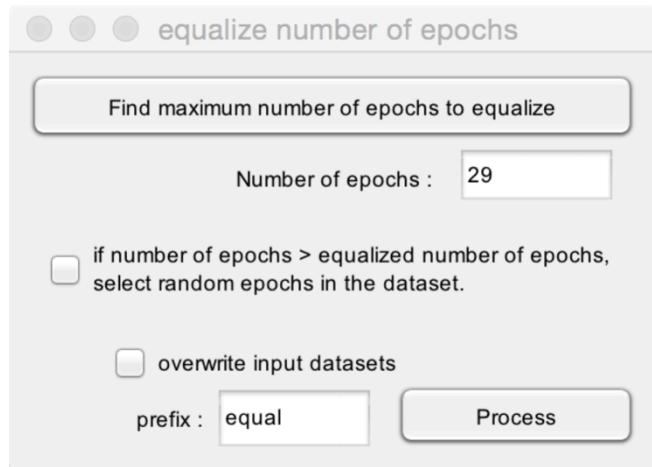


The left listbox shows all available indexes. The right listbox shows the indexes of the output dataset.

- **Add top.** Add the indexes selected in the list of available indexes to the top of the list of output indexes.
- **Add bottom.** Add the indexes selected in the list of available indexes to the bottom (end) of the list of output indexes.
- **Add all.** Add all available indexes to the list of output indexes.
- **Insert.** Insert the index selected in the list of available indexes to the position selected in the list of output indexes.
- **Sort ascending.** Sort the indexes in alphabetical order (ascending).
- **Sort descending.** Sort the indexes in alphabetical order (descending).
- The **Up** and **Down** arrows can be used to change the order of the output indexes.
- **Remove.** Remove the selected indexes from the output listbox.
- **Remove all.** Remove all indexes from the output listbox.
- **Select odd.** Select all odd-numbered indexes.
- **Select even.** Select all even-numbered indexes.

## EQUALIZE NUMBER OF EPOCHS ACROSS DATASETS

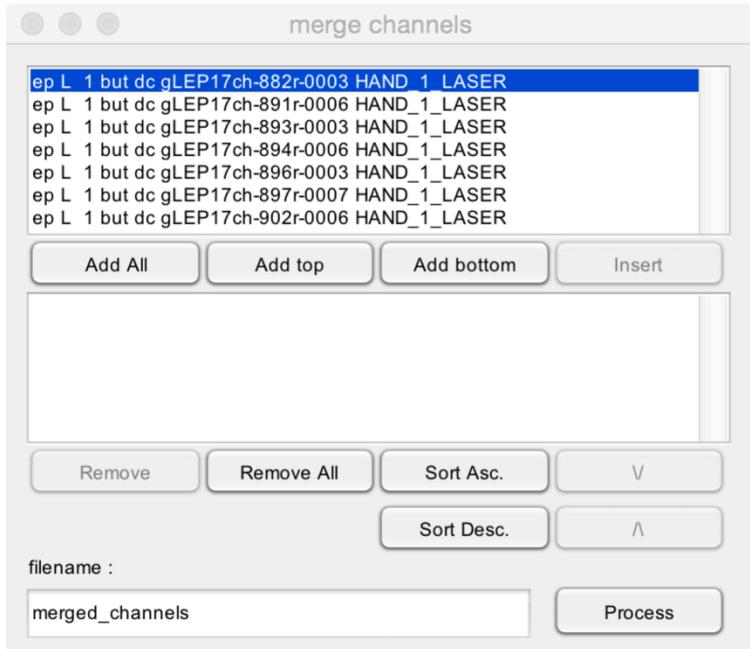
Equalize the number of epochs across multiple datasets.



- **Number of epochs.** The number of epochs to include in each output dataset.
- **Find maximum number of epochs to equalize.** Finds the maximum number of epochs that can be used to perform the equalize function, this will correspond to the number of epochs in the dataset having the smallest number of epochs. For example, if you selected three datasets with 31, 33 and 29 epochs, respectively, the maximum number of epochs will be 29.
- **If number of epochs > equalized number of epochs, select random epochs in the dataset.** When checked, if the number of epochs to equalize is smaller than the number of epochs in the dataset, the function will randomly select these epochs from the list of available epochs. If unchecked, the function will select the N first epochs of the dataset.

## MERGE CHANNELS OF MULTIPLE DATASETS

Merge the channels of multiple datasets into a single dataset.

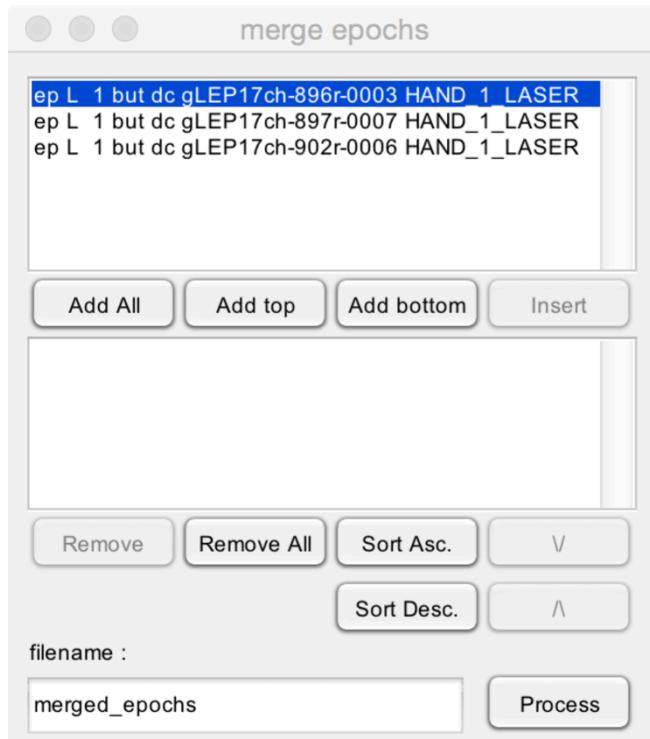


This function will merge the channels of multiple datasets into a single dataset. This is possible if and only if the datasets have corresponding numbers of epochs, indexes. In addition, the datasets must have identical X, Y and Z dimensions.

- **Add All.** Add all datasets to the merged dataset.
- **Add top.** Add the selected datasets to the top of the merged dataset.
- **Add bottom.** Add the selected datasets to the bottom of the merged dataset.
- **Insert.** Insert the selected dataset at the position selected in the merged dataset.
- **Remove.** Remove the selected datasets from the merged dataset.
- **Remove all.** Remove all datasets from the merged dataset.
- **Sort Ascending.** Sort the datasets of the merged dataset in alphabetical order (ascending).
- **Sort descending.** Sort the datasets of the merged dataset in alphabetical order (descending).
- **Up and Down** arrows. Change the order of the datasets.

## MERGE EPOCHS OF MULTIPLE DATASETS

Merge the epochs of multiple datasets into a single dataset.

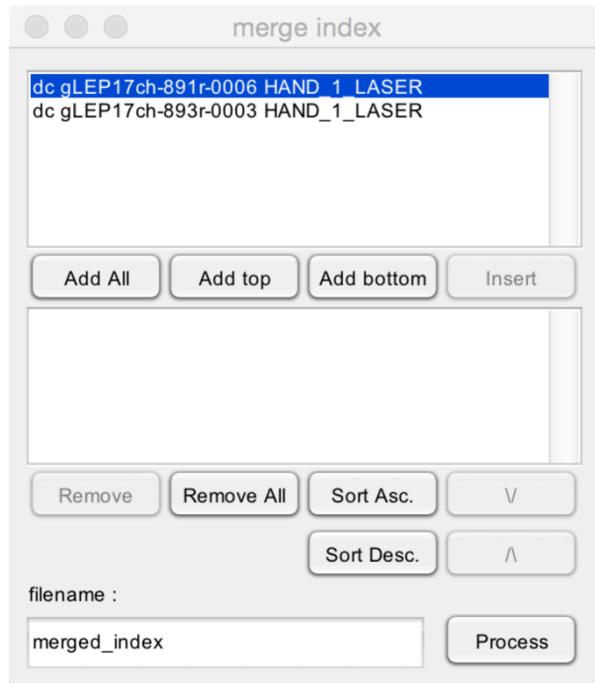


This function will merge the epochs of multiple datasets into a single dataset. This is possible if and only if the datasets have corresponding numbers of channels, indexes. In addition, the datasets must have identical X, Y and Z dimensions.

- **Add All.** Add all datasets to the merged dataset.
- **Add top.** Add the selected datasets to the top of the merged dataset.
- **Add bottom.** Add the selected datasets to the bottom of the merged dataset.
- **Insert.** Insert the selected dataset at the position selected in the merged dataset.
- **Remove.** Remove the selected datasets from the merged dataset.
- **Remove all.** Remove all datasets from the merged dataset.
- **Sort Ascending.** Sort the datasets of the merged dataset in alphabetical order (ascending).
- **Sort descending.** Sort the datasets of the merged dataset in alphabetical order (descending).
- **Up and Down** arrows. Change the order of the datasets.

## MERGE INDEXES OF MULTIPLE DATASETS

Merge the indexes of multiple datasets into a single dataset.

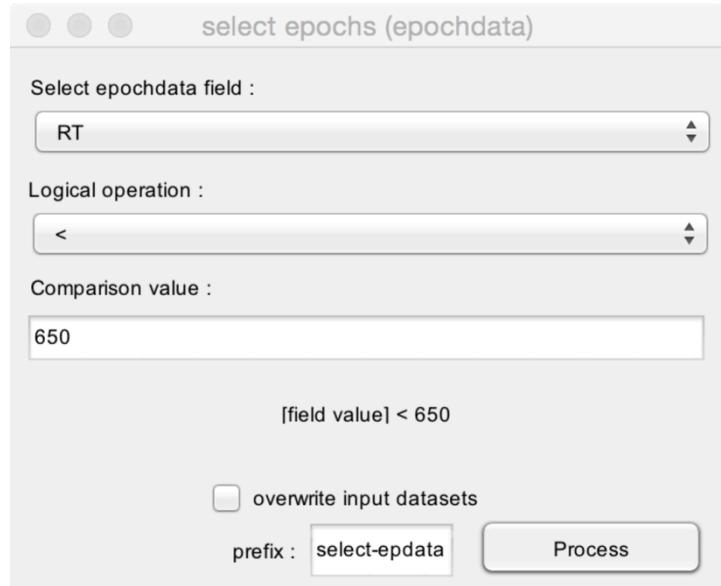


This function will merge the indexes of multiple datasets into a single dataset. This is possible if and only if the datasets have corresponding numbers of epochs, and channels. In addition, the datasets must have identical X, Y and Z dimensions.

- **Add All.** Add all datasets to the merged dataset.
- **Add top.** Add the selected datasets to the top of the merged dataset.
- **Add bottom.** Add the selected datasets to the bottom of the merged dataset.
- **Insert.** Insert the selected dataset at the position selected in the merged dataset.
- **Remove.** Remove the selected datasets from the merged dataset.
- **Remove all.** Remove all datasets from the merged dataset.
- **Sort Ascending.** Sort the datasets of the merged dataset in alphabetical order (ascending).
- **Sort descending.** Sort the datasets of the merged dataset in alphabetical order (descending).
- **Up and Down** arrows. Change the order of the datasets.

## SELECT EPOCHS ACCORDING TO EPOCH DATA

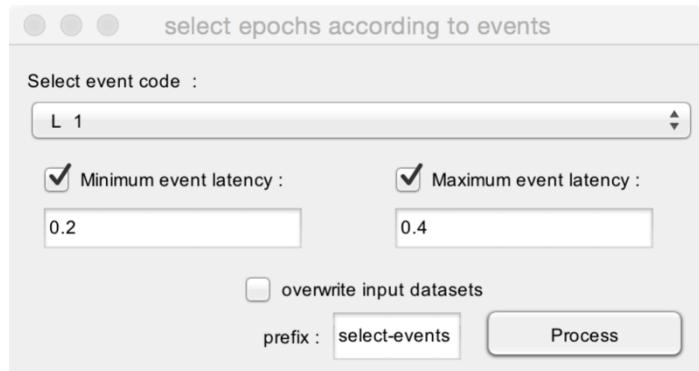
Select epochs of a dataset according to epoch data. This requires dataset(s) with epoch data fields.



- **Select epochdata field.** Select the epochdata field to be used as criterion to select epochs. In this example, the field is 'RT', corresponding to reaction times associated with each epoch of the datasets (in milliseconds).
- **Logical operation and Comparison value.** The logical operation to apply as criterion. In this example, epochs will be selected if and only the value stored in the field RT (i.e. the reaction time) is shorter than 650 ms.

## SELECT EPOCHS ACCORDING TO EVENT LATENCIES

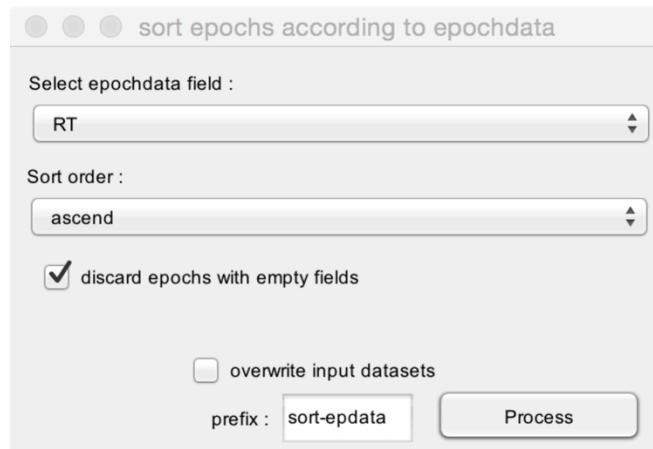
Select epochs of a dataset according to the latencies of events.



- **Select event code.** Select the event code to be used as criterion to select epochs. In this example, epochs will be selected based on the latencies of the event 'L\_1'.
- If both **minimum event latency** and **maximum event latency** are unchecked, epochs will be selected if they contain the selected event (in this example, if the event 'L\_1' occurs during the epoch), regardless of the latency of the event.
- **Minimum event latency.** If checked, epochs will be selected if and only if they contain the selected event and if the latency of that event is greater than the defined minimum latency. In this example, epochs will be selected if and only if they contain an event 'L\_1' with a latency >0.2 s.
- **Maximum event latency.** If checked, epochs will be selected if and only if they contain the selected event and if the latency of that event is shorter than the defined maximum latency. In this example, epochs will be selected if and only if they contain an event 'L\_1' with a latency <0.4 s.

## SORT EPOCHS ACCORDING TO EPOCH DATA

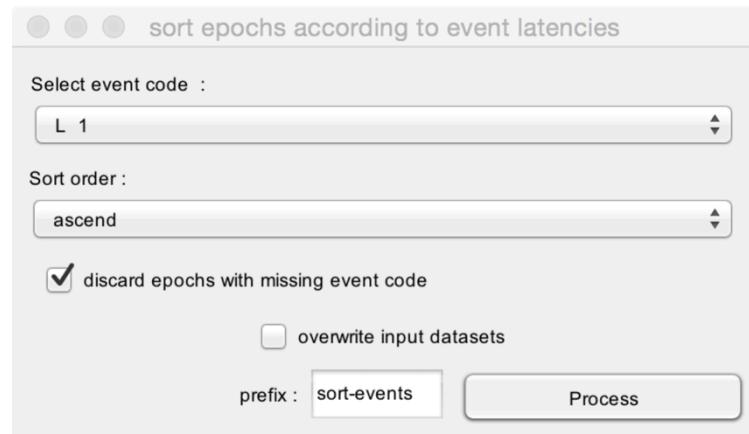
Sort epochs of a dataset according to the values stored in an epoch data field. This requires dataset(s) with epoch data fields.



- **Select epochdata field.** Select the epochdata field to be used for sorting. In this example, the field is 'RT', corresponding to reaction times associated with each epoch of the datasets (in milliseconds).
- **Sort order.** Epochs can be sorted in ascending or descending order.
- **Discard epochs with empty fields.** If epochs do not have a value associated to the selected field (e.g. epochs with no reaction times), these epochs can be either discarded, or appended to the end of the dataset.

## SORT EPOCHS ACCORDING TO EVENT LATENCIES

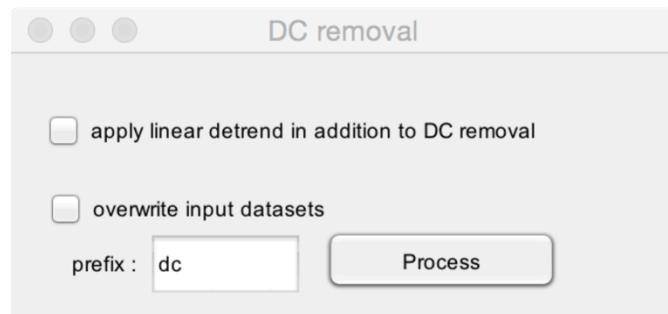
Sort epochs of a dataset according to the latencies of events.



- **Select event code.** Select the event code to be used as criterion to select epochs. In this example, epochs will be selected based on the latencies of the event 'L 1'.
- **Sort order.** Epochs can be sorted in ascending or descending order.  
**Discard epochs with empty fields.** If epochs do not have a value associated to the selected field (e.g. epochs with no reaction times), these epochs can be either discarded, or appended to the end of the dataset.

## DC REMOVAL AND LINEAR DETREND

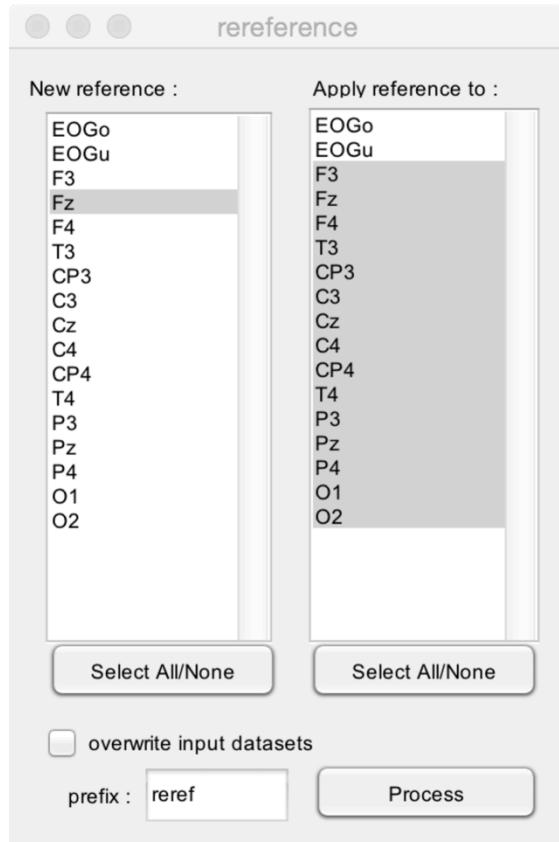
Remove DC from signals (i.e. for each channel, remove the mean of the signals). It is also possible to apply a linear detrend, such as to remove drifts in signals.



- **Apply linear detrend in addition to DC removal.** In addition to demeaning the signals (DC removal), also apply a linear detrend to remove drifts in the signals.

## REREFERENCE

Express signals using a new reference.



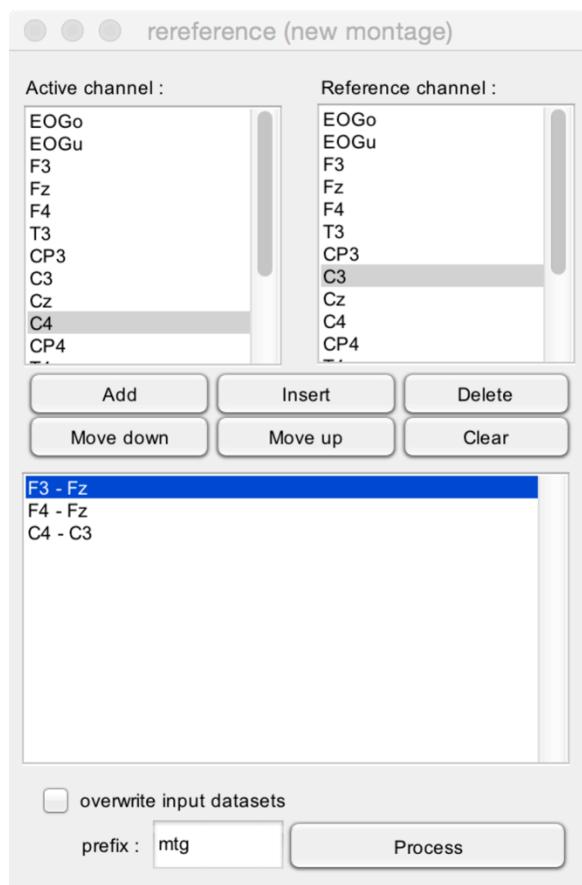
- **New reference.** Select one or more channels to be used as a new reference channel. In this example, the new reference will be Fz.
- **Apply reference to.** Select one or more channels onto which the new reference should be applied. In this example, the new reference (Fz) will be applied to all scalp channels (channels F3 to O2).

Note. For the rereferencing to be valid, the channels used to build the new reference and the channels onto which the new reference will be applied must have the same reference.

## REREFERENCE (CUSTOM MONTAGE)

Express signals using a new reference.

This function allows to design a custom montage.



The bottom listbox contains the list of channels of the output dataset. To add a new pair of signals (active channel – reference channel), you must select one **active channel** and one **reference channel**.

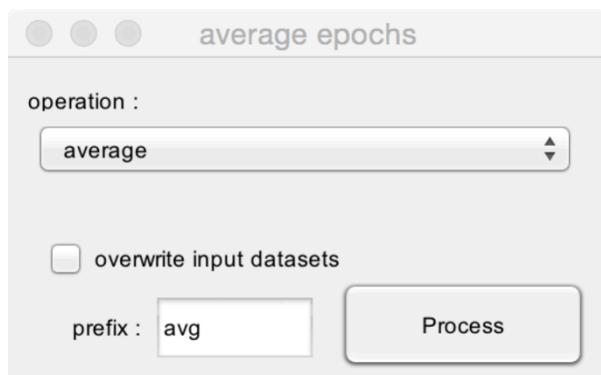
- **Add.** Add the selected channel pair (active – reference channel) to the output dataset.
- **Insert.** Insert the selected channel pair at the position selected in the output dataset.
- **Delete.** Delete the selected channel pair of the output dataset.
- **Move down** and **Move up.** Move the selected channel pair up or down in the output dataset.
- **Clear.** Clear all channel pairs of the output dataset.

## **POSTPROCESS**

## AVERAGE

### AVERAGE EPOCHS

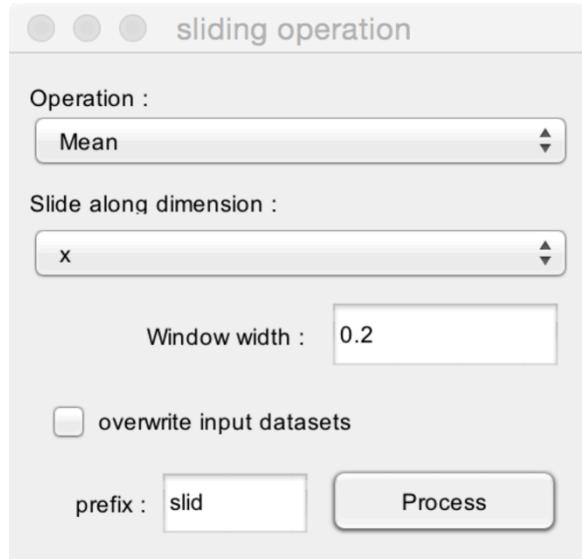
Average epochs across trials.



- **Operation.** In addition to computing the **average**, it is also possible to compute the **median** and the **standard deviation** across trials.

## SLIDING OPERATION ALONG DIMENSION

Apply a sliding average (or other operation) along a given dimension (X, Y or Z).



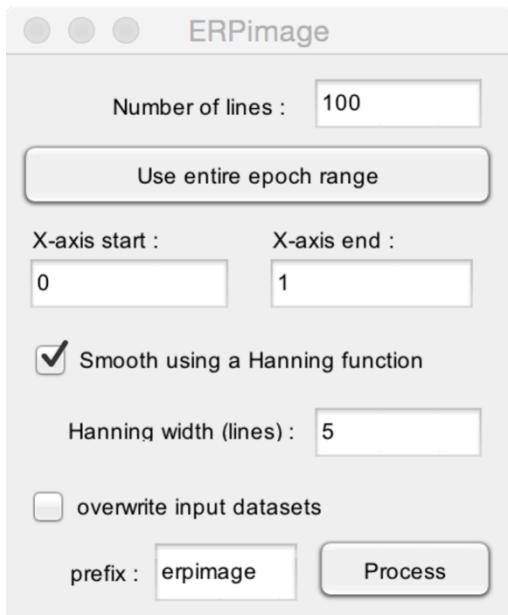
This function will perform a sliding operation along a given dimension. The width of the window is defined by the window width. For each sample of the dataset along the chosen dimension, the operation will be computed using all samples within the window, centered around that sample.

For example, if one selects to compute a sliding average along the X dimension with a window width of 0.2, the value of the output dataset at latency  $x_i$  will correspond to the mean of the values obtained at  $x_i-0.1$  s and  $x_i+0.1$  s

- **Operation.** In addition to applying a **sliding average** along a given dimension, it is also possible to compute the **standard deviation**, **maximum**, **minimum**, **75<sup>th</sup> percentile** and **25<sup>th</sup> percentile** along a given dimension.
- **Slide along dimension.** The dimension along which the sliding operation should be performed.
- **Window width.** The width of the window used to compute the sliding operation.

## SLIDING AVERAGE ACROSS TRIALS

Apply a sliding average across trials. This can be used to compute maps expressing signal amplitude as a function of time (x-axis) and trial (y-axis). See [http://sccn.ucsd.edu/wiki/Chapter\\_08:\\_Plotting\\_ERP\\_images](http://sccn.ucsd.edu/wiki/Chapter_08:_Plotting_ERP_images) for an explanation of these maps and their usefulness.

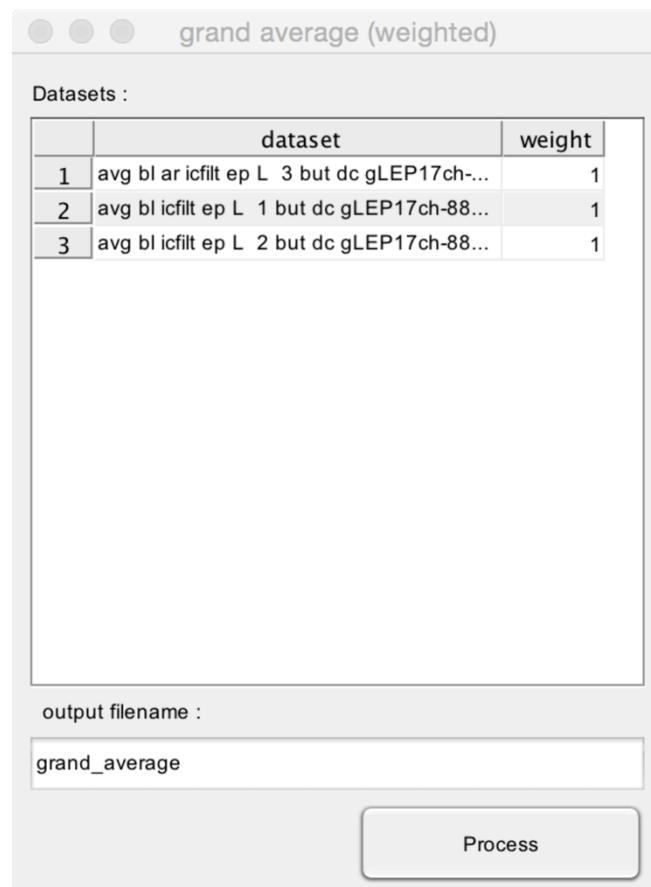


- **Number of lines.** The number of lines (y-axis) of the output ERPimage.
- **Use entire epoch range.** This will compute the ERPimage across the entire duration of the epochs.
- **X-axis start** and **X-axis end** define the beginning and end of the X-axis of the output ERP image. In this example, the ERPimage will be computed using the data sampled between 0 and 1.
- **Smooth using a Hanning function.** If checked, each line of the ERPimage will not correspond to the data measured at a single epoch, but to a weighted average of the data measured at that epoch and surrounding epoch. The weights are defined using a Hanning function. The width of the Hanning function is defined by the **Hanning width** field.

## GRAND AVERAGE (WEIGHTED)

Compute a grand average of the signals contained in multiple datasets.

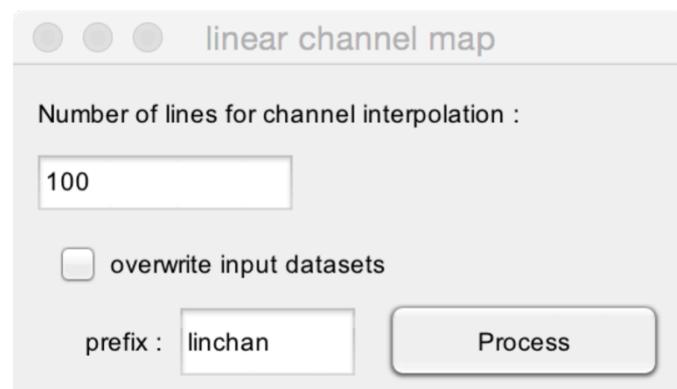
This function also makes it possible to compute a weighted grand average, such as to give different weights to each dataset (for example, to take into consideration that the averages obtained in each dataset were not obtained using the same number of trials).



- **Datasets.** The selected datasets.
- **Weight.** The weight to assign to each dataset (default = 1).

## LINEAR CHANNEL MAP

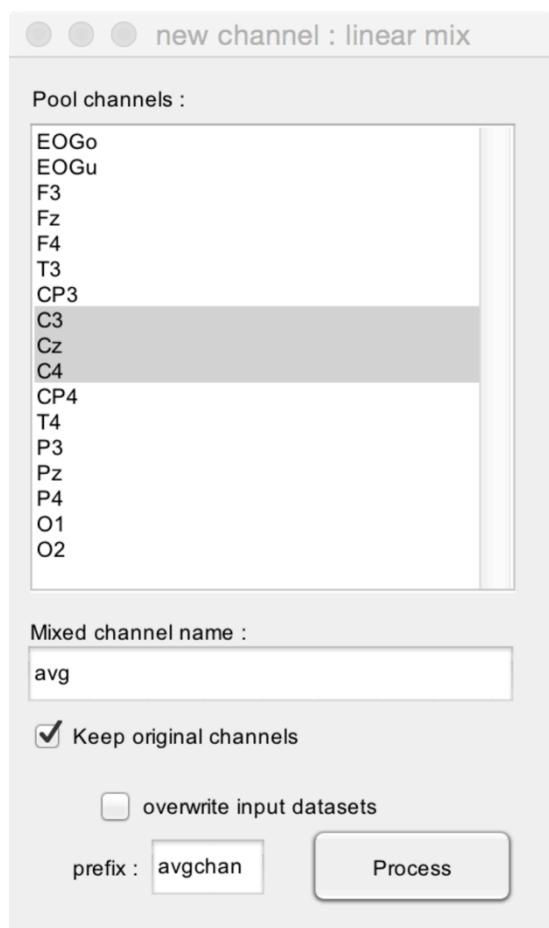
Compute a linear channel map. This will generate a map interpolating the values obtained at the different channels along the y dimension. This is useful to explore signals obtained from a linear configuration of electrodes, for example, to assess the occurrence of polarity reversals along the different contacts of an intracerebral electrode tract. In most cases, you will want to apply this function to the signals obtained after applying a **linear CSD**.



- **Number of lines for channel interpolation.** The number of lines to interpolate when creating the linear channel map.

## POOL CHANNELS

Pool (average) channels of a dataset.

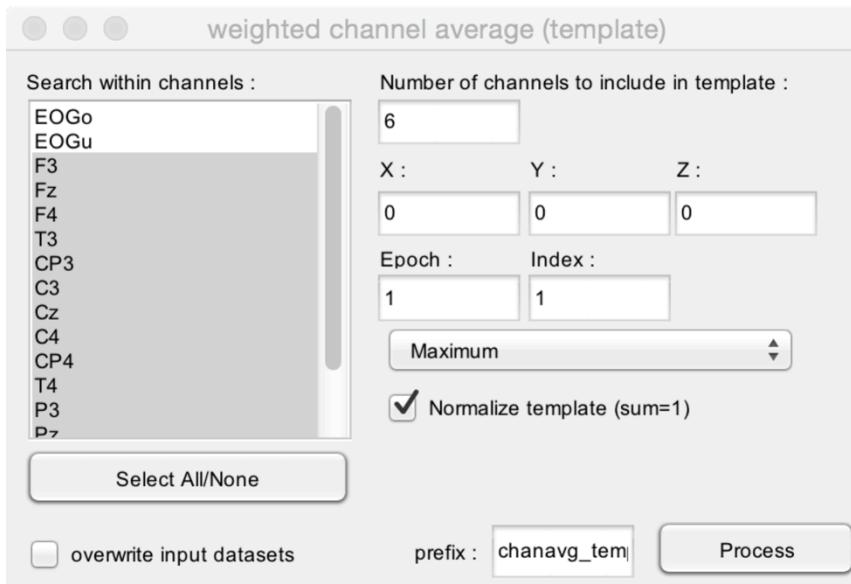


- **Pool channels.** The channel(s) to pool (average).
- **Mixed channel name.** The name of the new (pooled) channel.

## WEIGHTED CHANNEL AVERAGE (CREATE TEMPLATE)

Create a template to compute a weighted channel average. This is useful if you wish to express the magnitude of a given response measured at different electrodes.

This function is still under development.



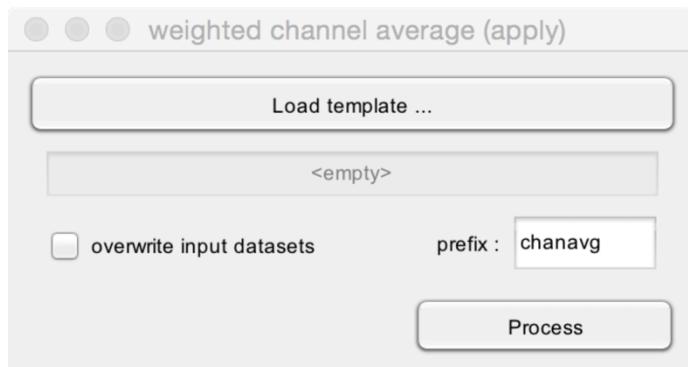
The function will search for the channels displaying maximum or minimum amplitude at a given position of the X/Y/Z axis. A template will then be created by selecting N channels exhibiting the strongest response. Each channel will be assigned a weight proportional to the amplitude of the response at the given position.

- **Search within channels.** The channels within which channels exhibiting the strongest response magnitude should be searched.
- **Number of channels to include in the template.** The number of channels to be included in the template.
- **X/Y/Z.** The location of the response along the X/Y/Z dimension (e.g. the latency of a peak, or the latency and frequency of a response identified in the time-frequency domain).
- **Maximum/Minimum.** Choose maximum if the response is positive (e.g. a positive peak, event-related synchronization). Choose minimum if the response is negative (e.g. a negative peak, event-related desynchronization).
- **Normalize template.** Normalize the weights such that their sum equals 1.

## WEIGHTED CHANNEL AVERAGE (APPLY TEMPLATE)

Apply a weighted channel average template to one or more datasets.

This function is still under development.



- **Load template.** Load a dataset containing a weighted channel average template (computed using the *weighted channel average (create template)* function).

## MATH

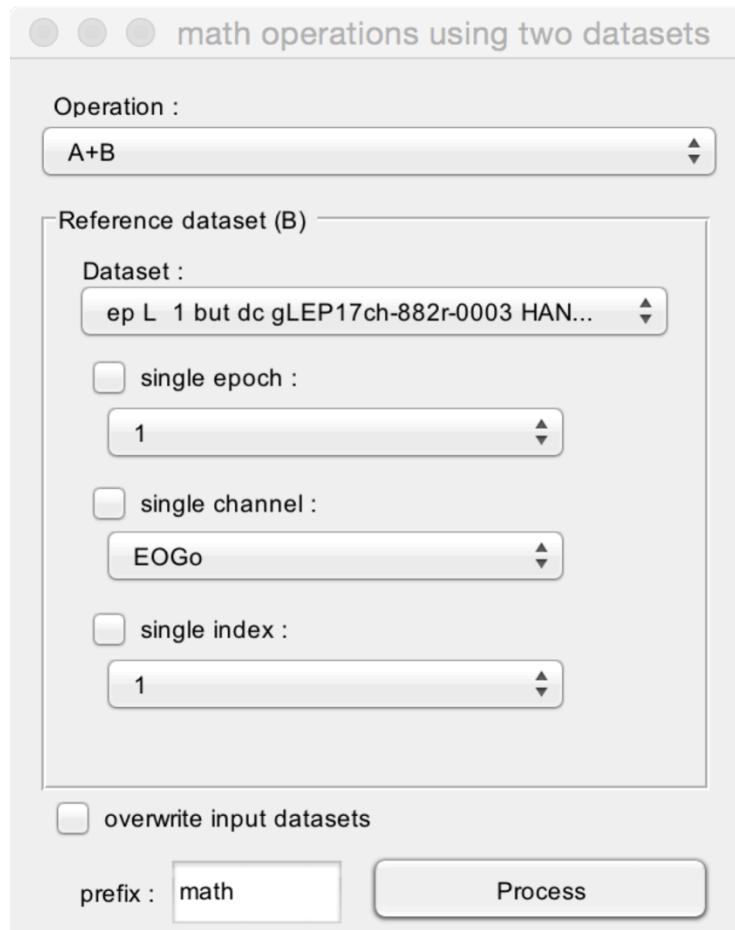
### COMPUTE SIGNAL DERIVATIVE

Compute the signal derivative. The derivative is computed by subtracting from each sample of the dataset the value measured at the preceding sample of the dataset ( $y_i = y_i - y_{i-1}$ )



## MATHEMATICAL OPERATION USING TWO DATASETS

Compute simple mathematical operations between two datasets.



- **Operation.** The operation to perform (add, subtract, multiple, divide).
- **Reference dataset.** The operation will be performed on pairs of datasets, referred to as **A** and **B**. You must select one dataset as the reference dataset (**B**).
- **Single epoch.** If checked, the operation will be performed using a single epoch of the reference dataset (i.e. the operation will be performed using all epochs of A and the selected epoch of B). If unchecked, the operation will be performed using all epochs of A and B (i.e. the operation will be performed using each epoch of A against each corresponding epoch of B).
- **Single channel.** If checked, the operation will be performed using a single channel of the reference dataset (i.e. the operation will be performed using all channels of A and the selected channel of B). If unchecked, the operation will be performed using all channels of A and B (i.e. the operation will be performed using each channel of A against each corresponding channel of B).
- **Single index.** If checked, the operation will be performed using a single index of the reference dataset (i.e. the operation will be performed using all indexes of A and the selected index of B). If unchecked, the operation will be performed using all indexes of A and B (i.e. the operation will be performed using each index of A against each corresponding index of B).

## MATHEMATICAL OPERATION USING A CONSTANT

Compute simple mathematical operations using one dataset and a constant value.

math operations using a constant

Operation :  
Add (A+constant)

Constant :  
10

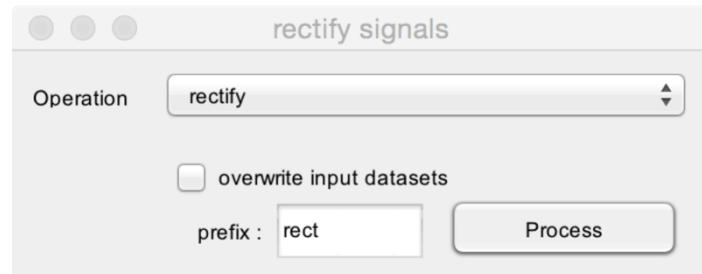
overwrite input datasets

prefix : math      Process

- **Operation.** The operation to perform (add, subtract, multiple, divide).
- **Constant.** The value of the constant. In this example, a constant value of 10 will be added to the signals of all datasets.

## RECTIFY OR SQUARE SIGNALS

Rectify or square signals.

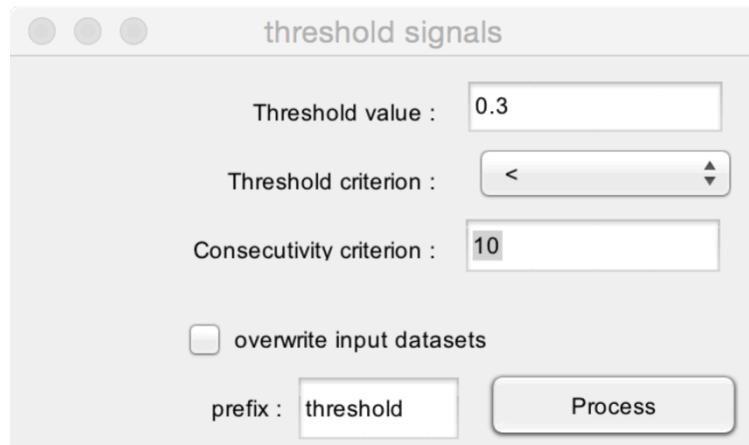


- **Operation.** You can either **rectify** signals (compute the absolute value of all samples in the dataset) or **square** signals (compute the square of all samples in the dataset).

## THRESHOLD SIGNALS

Threshold signals. Apply a threshold criterion to the datasets. This will output a binary dataset, where 0 defines sample positions which did not satisfy the threshold criterion, and 1 defines sample positions which satisfy the threshold criterion.

This binary dataset can then be used as a mask using the function **Mathematical operations using a constant**.



- **Threshold value.** The value to be used as threshold criterion.
- **Threshold criterion.** The threshold criterion. In this example, sample values  $<0.3$  will satisfy the threshold criterion (1), and sample values  $\geq 0.3$  will not satisfy the threshold criterion (0).
- **Consecutivity threshold.** It is possible to consider samples as satisfying the threshold criterion if and only if these samples are surrounded by a number of samples also satisfying the threshold criterion. In this examples, samples will be considered as satisfying the threshold criterion if and only if they are surrounded by at least 10 samples also satisfying the threshold criterion.

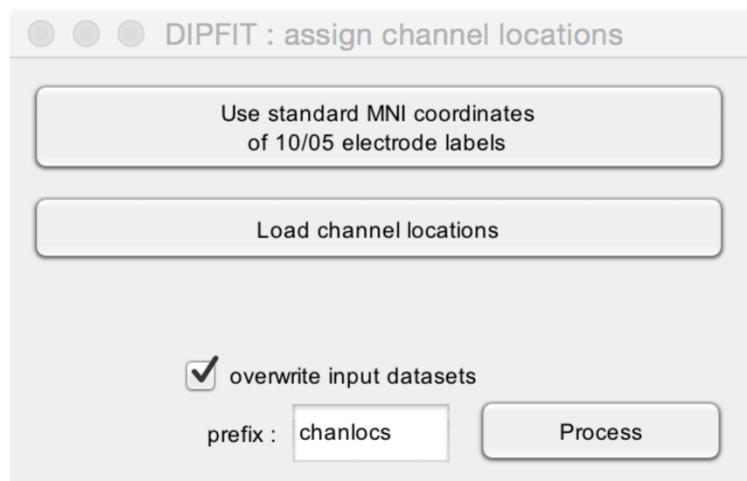
## SOURCE ANALYSIS

### DIPFIT : LOAD CHANNEL COORDINATES

DIPFIT can be used to perform dipolar source analysis.

See <http://fieldtrip.fcdonders.nl/tutorial/natmeg/dipolefitting> for details.

This function is used to define the channel coordinates for DIPFIT.



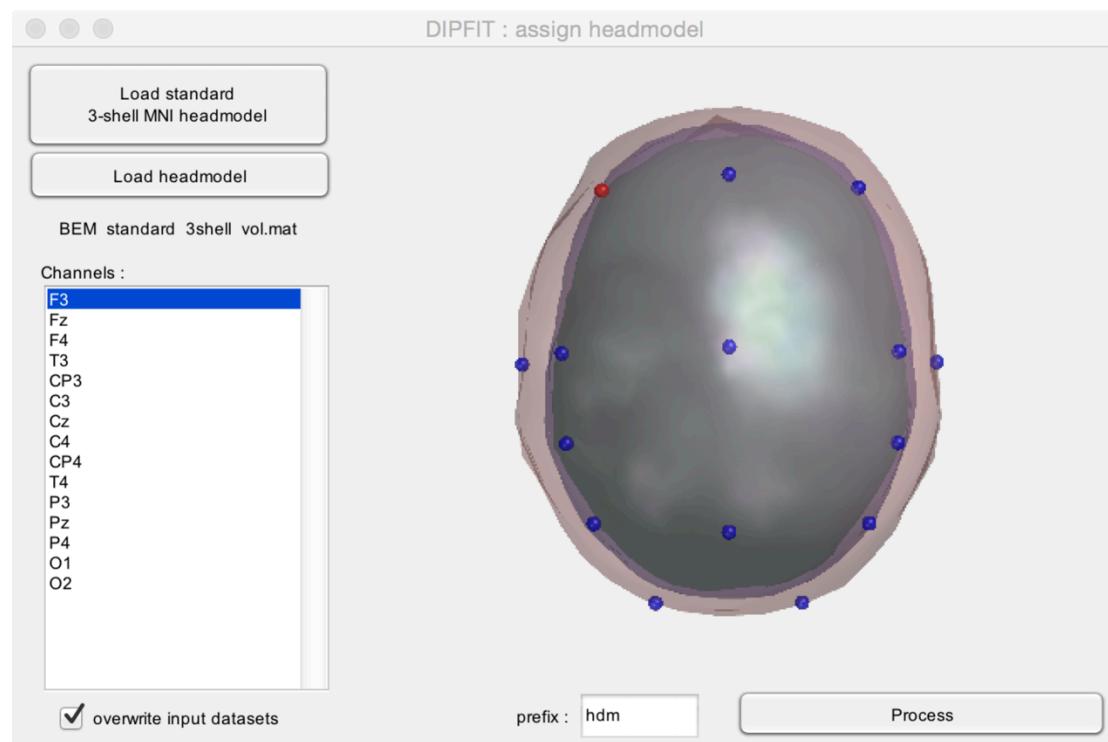
- **Use standard MNI coordinates of the 10/05 electrode labels.** Assign standard MNI coordinates to channels having standard International 10/05 labels.
- **Load channel locations.** Load a custom file with channel locations. This can be a file with individually-sampled channel coordinates.

## DIPFIT : LOAD HEAD MODEL

DIPFIT can be used to perform dipolar source analysis.

See <http://fieldtrip.fcdonders.nl/tutorial/natmeg/dipolefitting> for details.

This function is used to define the head model.



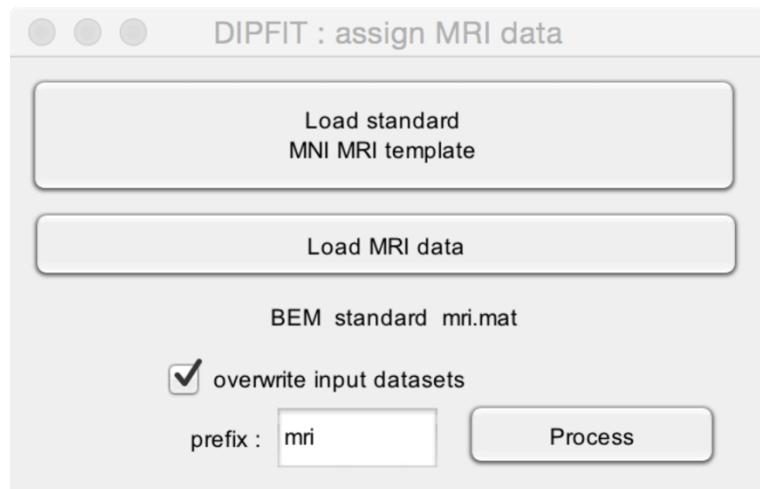
- **Load standard 3-shell MNI headmodel.** Assign a standard 3-shell MNI head model.
- **Load head model.** Load a custom headmodel.

## DIPFIT : LOAD MRI DATA

DIPFIT can be used to perform dipolar source analysis.

See <http://fieldtrip.fcdonders.nl/tutorial/natmeg/dipolefitting> for details.

This function is used to define the associated MRI data (for visualization).



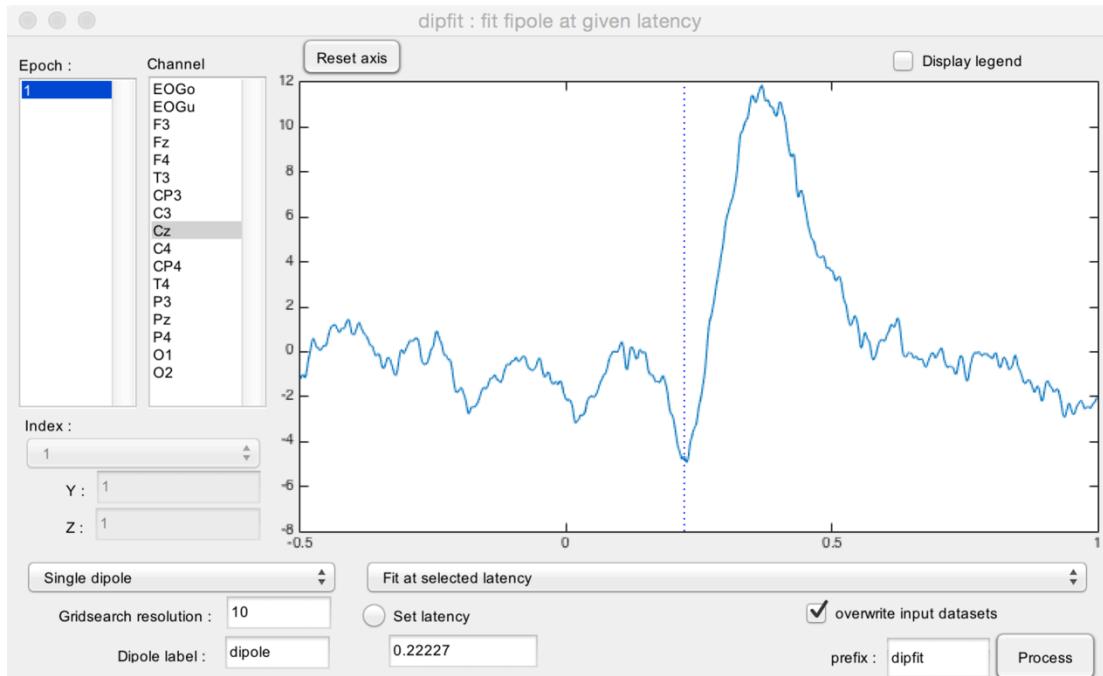
- **Load standard MNI MRI template.** Assign a standard MNI MRI template.
- **Load MRI data.** Load a custom MRI data.

## DIPFIT : FIT DIPOLE(S) AT A GIVEN LATENCY

DIPFIT can be used to perform dipolar source analysis.

See <http://fieldtrip.fcdonders.nl/tutorial/natmeg/dipolefitting> for details.

Fit dipole(s) at a user-defined latency.



- **Epoch.** Select one or more epochs to display.
- **Channel.** Select one or more channels to display
- **Index.** Select the index to display (if >1).
- **Y.** Select the Y position to display.
- **Z.** Select the Z position to display.
- The dipole model can be either a **Single dipole** or **Two symmetrical dipoles** (symmetry in the X, Y or Z dimension), or **Two unconstrained dipoles**.
- The dipole fitting involves two steps, a coarse search using a grid, followed by a fine search. The **Gridsearch resolution** defines the resolution of the coarse grid search.
- **Dipole label.** The label to assign to the fitted dipole.

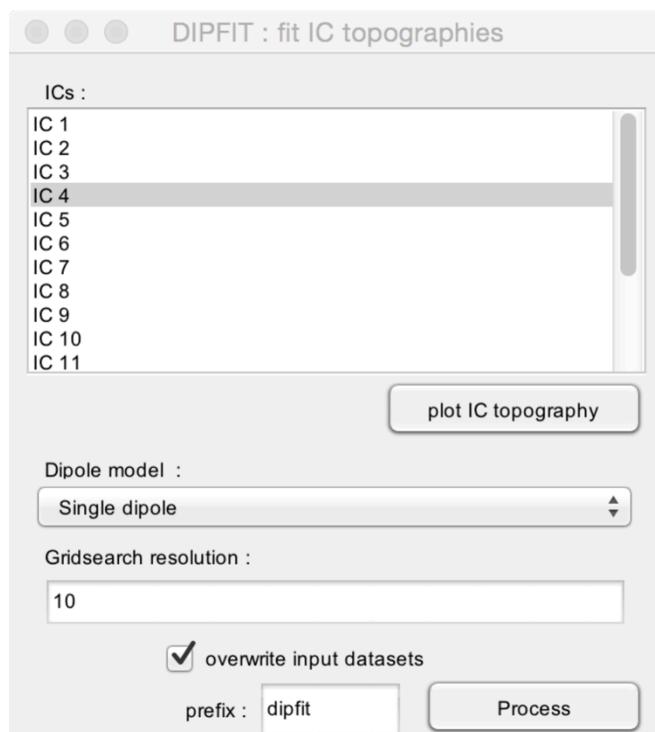
You may either **fit at a selected latency**, and use **Set latency** to define the latency. Alternatively, you can **fit at a maximum within a range** or **fit at a minimum within a range**, and set that latency range using **Set lower latency limit** and **Set upper latency limit**. The latencies can be defined by entering values in the corresponding fields, or clicking on the graph after clicking the **Set latency** radio button.

## DIPFIT : FIT DIPOLE(S) ONTO IC TOPOGRAPHIES

DIPFIT can be used to perform dipolar source analysis.

See <http://fieldtrip.fcdonders.nl/tutorial/natmeg/dipolefitting> for details.

Fit dipole(s) onto IC topographies. This requires a dataset with an associated ICA matrix.



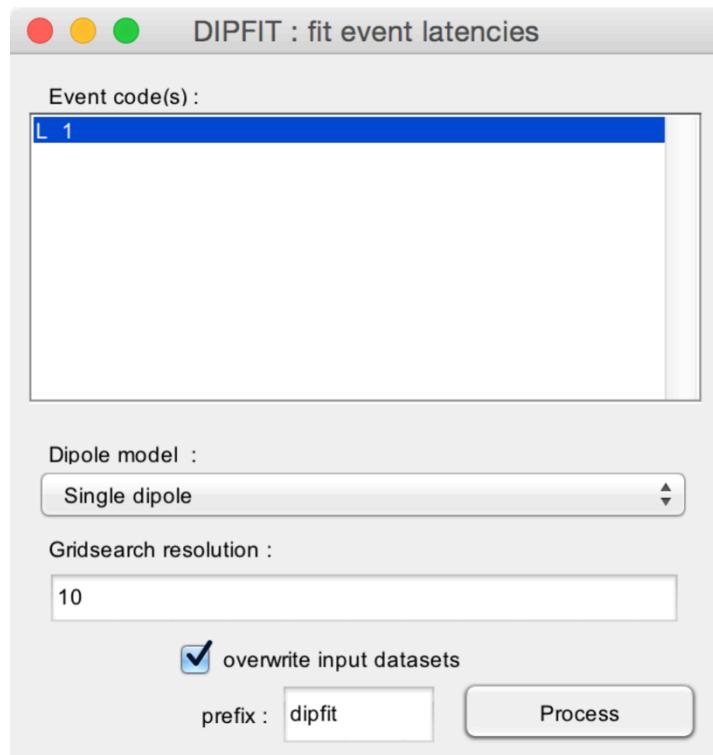
- **ICs.** Select the IC topography onto which the dipole model must be fitted.
- **Plot IC topography.** Plot the scalp topography of the selected IC.
- **Dipole model.** The dipole model can be either a **Single dipole** or **Two symmetrical dipoles** (symmetry in the X, Y or Z dimension), or **Two unconstrained dipoles**.
- The dipole fitting involves two steps, a coarse search using a grid, followed by a fine search. The **Gridsearch resolution** defines the resolution of the coarse grid search.

## DIPFIT : FIT DIPOLE(S) AT EVENT LATENCIES

DIPFIT can be used to perform dipolar source analysis.

See <http://fieldtrip.fcdonders.nl/tutorial/natmeg/dipolefitting> for details.

Fit dipole(s) at event latencies. This is particularly useful if you have identified the latency of EEG responses (e.g. ERP peaks) using the **Find peaks in waveforms** function.



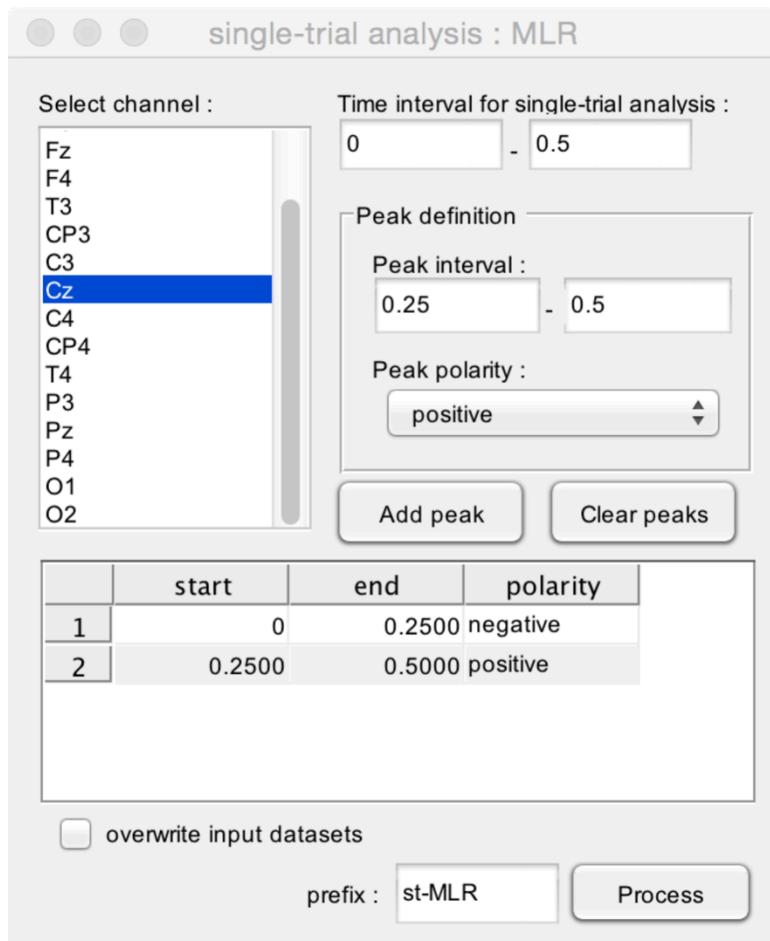
- **Event code(s).** Select the event code(s) onto which the dipole model should be fitted. The function will fit the dipolar model to the signals at each occurrence of the selected event code(s).
- **Dipole model.** The dipole model can be either a **Single dipole** or **Two symmetrical dipoles** (symmetry in the X, Y or Z dimension), or **Two unconstrained dipoles**.
- The dipole fitting involves two steps, a coarse search using a grid, followed by a fine search. The **Gridsearch resolution** defines the resolution of the coarse grid search.

## SINGLE TRIAL ANALYSIS

### MULTIPLE LINEAR REGRESSION (MLR)

Find and estimate peak latencies and amplitudes at the level of single trials.

See <http://www.ncbi.nlm.nih.gov/pubmed/20004255>.



- **Select channel.** Select the channel within which the function should estimate peak latencies and amplitudes.
- **Time interval for single-trial analysis.** The time interval to be used for the single trial analysis. This time interval should at least include the peak intervals.

The function can be used to estimate any given configuration of peaks. The **Peak definition** box is used to define and add peaks to the model.

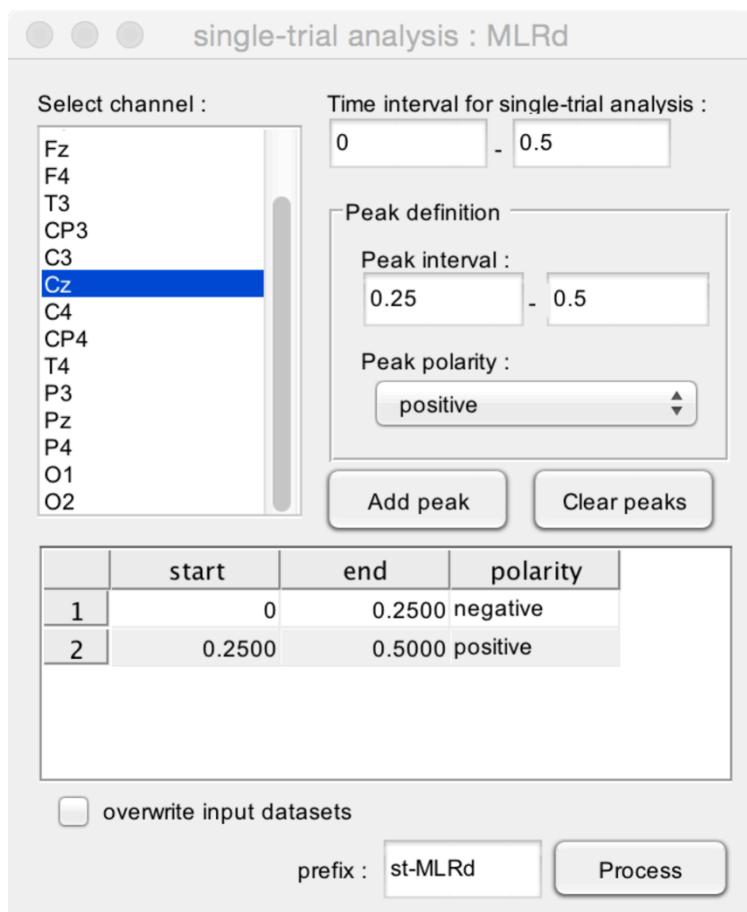
- **Peak interval.** The time interval within which the peak should occur.
- **Peak polarity.** Defines whether the peak is positive or negative.
- **Add peak.** Add the defined peak configuration to the list of peaks.
- **Clear peaks.** Delete all peak definitions.

In this example, the function will estimate two peaks using the single-trial signals measured at electrode Cz. The first peak is a negative peak whose latency must occur between 0 and 0.25 s after stimulus onset. The second is a positive peak whose latency must occur between 0.25 and 0.50 s after stimulus onset.

## MULTIPLE LINEAR REGRESSION (MLR)

Find and estimate peak latencies and amplitudes at the level of single trials. As compared to the previous function, this introduces an additional ‘distortion’ parameter to account for variability in peak width across trials.

See <http://www.ncbi.nlm.nih.gov/pubmed/21880936>.



- **Select channel.** Select the channel within which the function should estimate peak latencies and amplitudes.
- **Time interval for single-trial analysis.** The time interval to be used for the single trial analysis. This time interval should at least include the peak intervals.

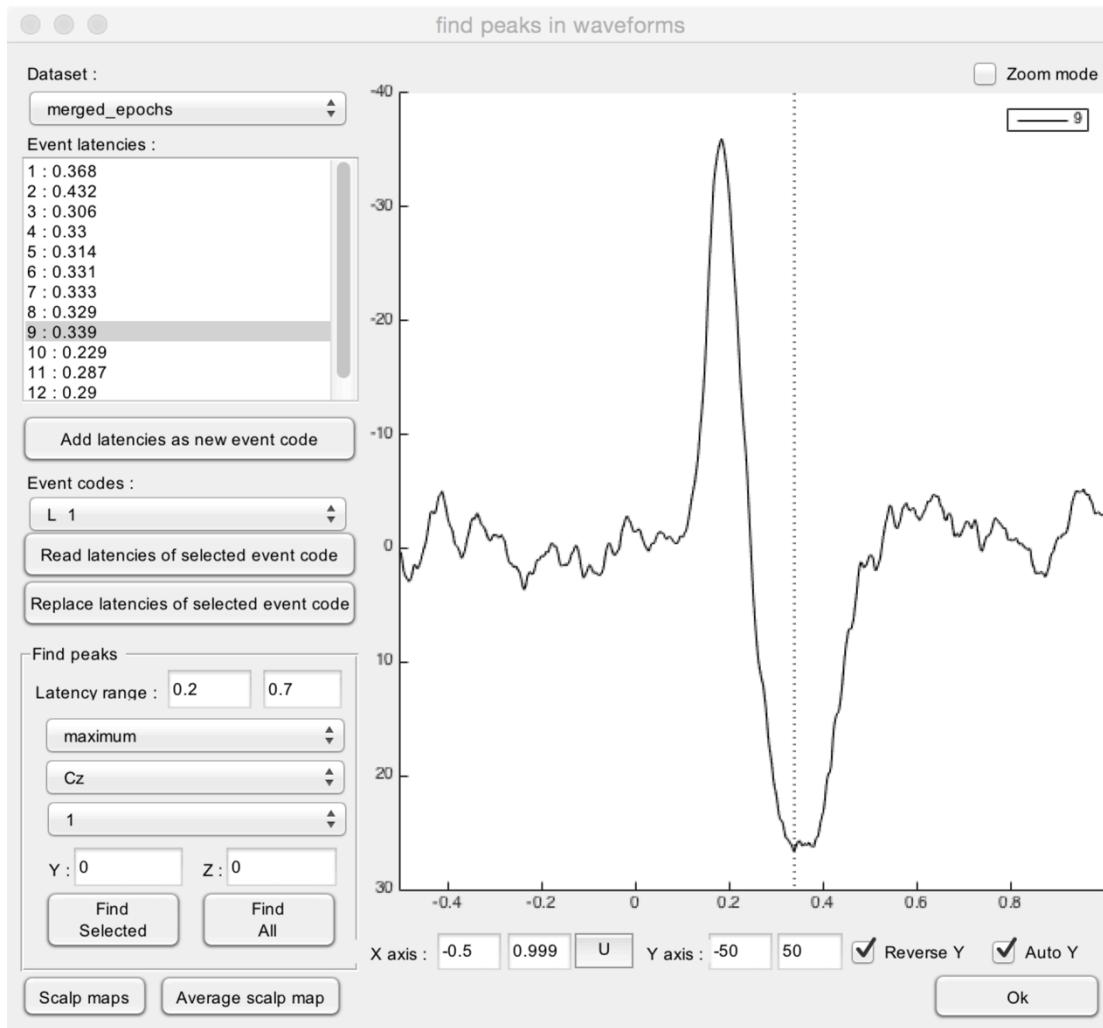
The function can be used to estimate any given configuration of peaks. The **Peak definition** box is used to define and add peaks to the model.

- **Peak interval.** The time interval within which the peak should occur.
- **Peak polarity.** Defines whether the peak is positive or negative.
- **Add peak.** Add the defined peak configuration to the list of peaks.
- **Clear peaks.** Delete all peak definitions.

In this example, the function will estimate two peaks using the single-trial signals measured at electrode Cz. The first peak is a negative peak whose latency must occur between 0 and 0.25 s after stimulus onset. The second is a positive peak whose latency must occur between 0.25 and 0.50 s after stimulus onset.

## FIND PEAKS IN WAVEFORMS

This function is used to identify peaks in waveforms.  
The peak latencies are stored as events.



- **Dataset.** Select the dataset to display.
- **Event latencies.** This lists all epochs of the selected dataset, and the event latencies associated to each dataset.
- **Find peaks.** Peaks are identified by finding a maximum (positive peak) or minimum (negative peak) within a user-defined time interval, and at a user-defined channel.
  - **Latency range.** The user-defined time interval within which the peak should be identified.
  - **Maximum/Minimum.** Defines whether the function should identify a maximum (positive peak) or a minimum (negative peak) within the user-defined latency range.
  - **Channel listbox.** Defines the channel at which the peak should be defined. Also defines the channel that is displayed in the right graph.
  - **Index listbox.** Defines the index at which the peak should be defined. Also defines the index that is displayed in the right graph. This is relevant if and only if your dataset contains multiple indexes.
  - **Y/Z.** Defines the Y and Z position at which the peak should be defined. Also defines the Y and Z position of the data displayed in the right graph. This is relevant if and only if your dataset contains Y or Z dimensions.

- **Find selected.** Find peak latencies within the epochs selected in the **Event latencies** listbox.
- **Find All.** Find peak latencies within all epochs of the selected dataset.

Once you have defined all peak latencies of all datasets, you can add these peak latencies as a new event code in the dataset.

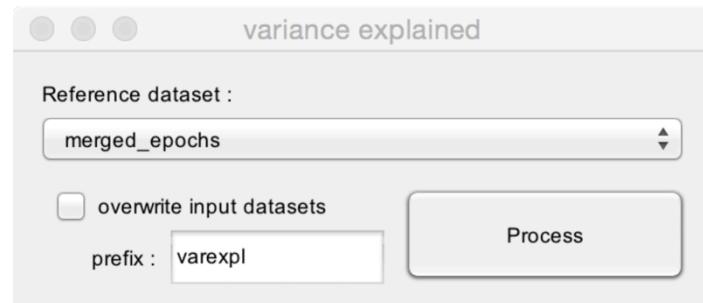
- **Add latencies as a new event code.** Adds all the defined event latencies. as a new event. You will be prompted for a new event\_code defining this peak. This applies to all datasets.
- **Read latencies of selected event code.** Reads the latencies of the event code selected in the **Event codes** listbox. This will replace all latencies in the **Event latencies** listbox.
- **Replace latencies of selected event code.** This will replace all latencies of the event code selected in the **Event codes** listbox with the latencies stored in the **Event latencies** listbox.

To decide whether or not a peak has been correctly identified, you can use the following visualization functions:

- **Scalp maps.** Plot a scalp map of the topographical distribution of the signal at the selected latency.
- **Average scalp map.** Plot s scalp map of the topographical distribution of the average of the signals measured at each peak latency in each single trial.
- **X axis** and **Y axis.** Change the X-axis and Y-axis range used to display the waveforms. The **U** button is used to update the X and Y axes.
- **Reverse Y.** Change the direction of the Y axis.
- **Auto Y.** Auto-adjust the Y axis range.
- **Zoom mode.**

## GLOBAL EXPLAINED VARIANCE

Assesses how the variance of one dataset can be explained by the signal contained in another dataset.



- **Reference dataset.** The reference dataset is the dataset to be used to assess how well it can explain the variance of the other datasets.

## **STATISTICS**

## COMPARE MORE THAN TWO DATASETS (ANOVA)

Compute a point-by-point ANOVA using multiple datasets. The model can include any combination of within and between-subject factors. The function also implements permutation testing to identify significant clusters in the obtained statistical time-courses or maps.

To use this function, you must create merged datasets, where each epoch corresponds to a measure (e.g. the average waveform of a given subject). For within-subject factors, each epoch of the different datasets must correspond to the same subject.

The screenshot shows the ANOVA software interface. At the top, there is a 'Factor name' input field containing 'time'. Below it are three buttons: 'Add as within factor', 'Add as between factor', and 'Clear All'. A 'Groups' section contains a table with four rows of data:

	W:intensity	W:time
gLEP17ch-882r-0003 HAND_1 LASER	1	1
gLEP17ch-891r-0006 HAND_1 LASER	2	1
gLEP17ch-893r-0003 HAND_1 LASER	1	2
gLEP17ch-894r-0006 HAND_1 LASER	2	2

Below the groups table are several configuration fields:

- 'Alpha level': 0.05
- 'filename (F values)': anova\_F\_values
- 'filename (p values)': anova\_p\_values
- A checkbox for 'Clustersize-based permutation testing' is checked.
- 'Number of permutations': 250
- A checkbox for 'Enable parallel computing' is checked.
- A dropdown menu for 'Percentile of mean cluster sum' is set to '95'.
- 'Cluster threshold': 95
- A large 'Process' button at the bottom right.

- **Factor name.** The name of the factor that you wish to add to the ANOVA model.
- **Add as within factor.** Add the factor as a within-subject factor.
- **Add as between factor.** Add the factor as a between-subject factor.
- **Clear All.** Clear the ANOVA model.

The **Groups** table is used to assign each dataset to the corresponding levels of the different factors. In this example, the model includes two within-subject factors, labeled 'intensity' and 'time'. For the factor 'intensity', the 1<sup>st</sup> and 3<sup>rd</sup> files belong to level 1, and the 2<sup>nd</sup> and 4<sup>th</sup> files belong to level 2. For the factor 'time', the 1<sup>st</sup> and 2<sup>nd</sup> files belong to level 1, and the 3<sup>rd</sup> and 4<sup>th</sup> files belong to level 2.

- **Filename (F values).** The name of the output dataset containing the point-by-point F values.
- **Filename (p values).** The name of the output dataset containing the point-by-point p values.

- **Enable parallel computing.** Enable Matlab parallel computing. This may speedup the analysis, especially, if you are running permutation testing.
- **Alpha level.** The minimum p-value to consider the result of the statistical test as "significant".

The main issue of point-by-point statistical testing is the correction for multiple comparisons. Permutation testing can be a mean to address this issue. See <http://www.ncbi.nlm.nih.gov/pubmed/17517438>. Permutation testing computes a distribution of the clusters of significant p values obtained when computing the statistical tests after permuting the datasets. The "strength" of each cluster is estimated using the sum of the test statistic within each significant cluster obtained using the permuted datasets. This distribution of clusters is then used to define a cluster threshold. The cluster threshold is applied to the result of the statistical test applied on the non-permuted datasets, such as to discard all clusters which are below the cluster threshold.

- **Clustersize-based permutation testing.** Perform a clustersize-based permutation testing. This can be very time consuming.
- **Number of permutations.** The number of permutations to be used to build the clustersize distribution.

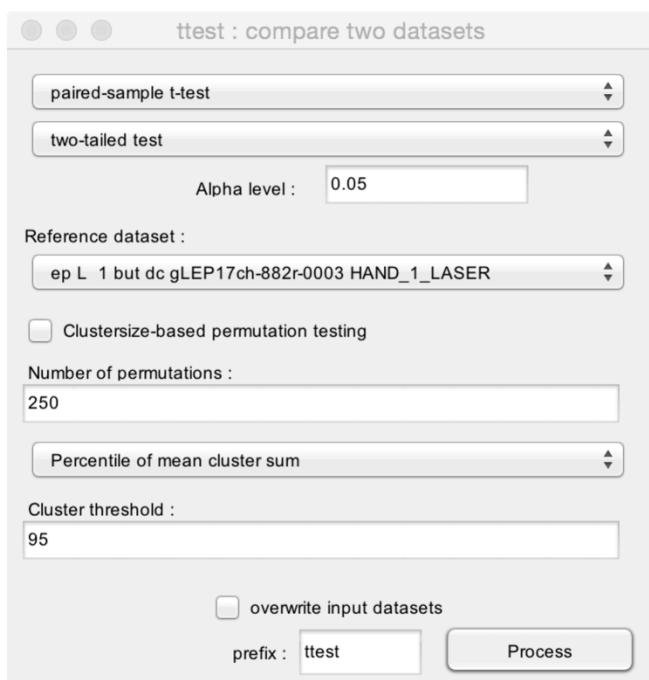
Several methods have been proposed to define the threshold distinguishing between "significant" and "non significant" clusters:

- **Standard deviation of mean cluster sum.** The threshold is defined as a z-score using the mean and standard deviation of the distribution of all significant clusters obtained using permutation testing (e.g. Z>2).
- **Standard deviation of maximum cluster sum.** The threshold is defined as a z-score using the mean and standard deviation of the distribution of the most significant cluster obtained for each permutation testing (e.g. Z>2).
- **Percentile of mean cluster sum.** The threshold is defined as a percentile using the distribution z of the distribution of all significant clusters obtained using permutation testing (e.g. >95%; this is equivalent to setting the cluster threshold to p <0.05).
- **Percentile of mean cluster sum.** The threshold is defined as a percentile using the distribution z of the distribution of the most significant cluster obtained for each permutation testing (e.g. >95%; this is equivalent to setting the cluster threshold to p <0.05).

## COMPARE TWO DATASETS (T-TEST)

Compute a point-by-point t-test using two datasets. The function also implements permutation testing to identify significant clusters in the obtained statistical time-courses or maps.

To use this function, you must create two merged datasets, where each epoch corresponds to a measure (e.g. the average waveform of a given subject). For a paired-sample t-test, each epoch of the two datasets must correspond to the same subject.



- **Test type.** The function can be used to perform a **paired-sample t-test**, or a **two samples t-test**.
- Select whether you want to perform a **two-tailed test**, a **right-tailed test**, or a **left-tailed test**. Add the factor as a within-subject factor.
- **Reference dataset.** Choose one reference dataset, to which the other dataset(s) should be compared.
- **Alpha level.** The minimum p-value to consider the result of the statistical test as "significant".

The main issue of point-by-point statistical testing is the correction for multiple comparisons. Permutation testing can be a mean to address this issue. See <http://www.ncbi.nlm.nih.gov/pubmed/17517438>. Permutation testing computes a distribution of the clusters of significant p values obtained when computing the statistical tests after permuting the datasets. The "strength" of each cluster is estimated using the sum of the test statistic within each significant cluster obtained using the permuted datasets. This distribution of clusters is then used to define a cluster threshold. The cluster threshold is applied to the result of the statistical test

applied on the non-permuted datasets, such as to discard all clusters which are below the cluster threshold.

- **Clustersize-based permutation testing.** Perform a clustersize-based permutation testing. This can be very time consuming.
- **Number of permutations.** The number of permutations to be used to build the clustersize distribution.

Several methods have been proposed to define the threshold distinguishing between “significant” and “non significant” clusters:

- **Standard deviation of mean cluster sum.** The threshold is defined as a z-score using the mean and standard deviation of the distribution of all significant clusters obtained using permutation testing (e.g.  $Z>2$ ).
- **Standard deviation of maximum cluster sum.** The threshold is defined as a z-score using the mean and standard deviation of the distribution of the most significant cluster obtained for each permutation testing (e.g.  $Z>2$ ).
- **Percentile of mean cluster sum.** The threshold is defined as a percentile using the distribution  $z$  of the distribution of all significant clusters obtained using permutation testing (e.g. >95%; this is equivalent to setting the cluster threshold to  $p < 0.05$ ).
- **Percentile of mean cluster sum.** The threshold is defined as a percentile using the distribution  $z$  of the distribution of the most significant cluster obtained for each permutation testing (e.g. >95%; this is equivalent to setting the cluster threshold to  $p < 0.05$ ).

## COMPARE TWO DATASETS (WILCOXON)

Compute a point-by-point Wilcoxon test using two datasets. The function also implements permutation testing to identify significant clusters in the obtained statistical time-courses or maps.

To use this function, you must create two merged datasets, where each epoch corresponds to a measure (e.g. the average waveform of a given subject). For a paired-sample t-test, each epoch of the two datasets must correspond to the same subject.



- **Test type.** The function can be used to perform a **signed rank test**, a **rank sum test**, or a **sign test**.
- Select whether you want to perform a **two-tailed test**, a **right-tailed test**, or a **left-tailed test**. Add the factor as a within-subject factor.
- **Reference dataset.** Choose one reference dataset, to which the other dataset(s) should be compared.
- **Alpha level.** The minimum p-value to consider the result of the statistical test as "significant".

The main issue of point-by-point statistical testing is the correction for multiple comparisons. Permutation testing can be a mean to address this issue. See <http://www.ncbi.nlm.nih.gov/pubmed/17517438>. Permutation testing computes a distribution of the clusters of significant p values obtained when computing the statistical tests after permuting the datasets. The "strength" of each cluster is estimated using the sum of the test statistic within each significant cluster obtained using the permuted datasets. This distribution of clusters is then used to define a

cluster threshold. The cluster threshold is applied to the result of the statistical test applied on the non-permuted datasets, such as to discard all clusters which are below the cluster threshold.

- **Clustersize-based permutation testing.** Perform a clustersize-based permutation testing. This can be very time consuming.
- **Number of permutations.** The number of permutations to be used to build the clustersize distribution.

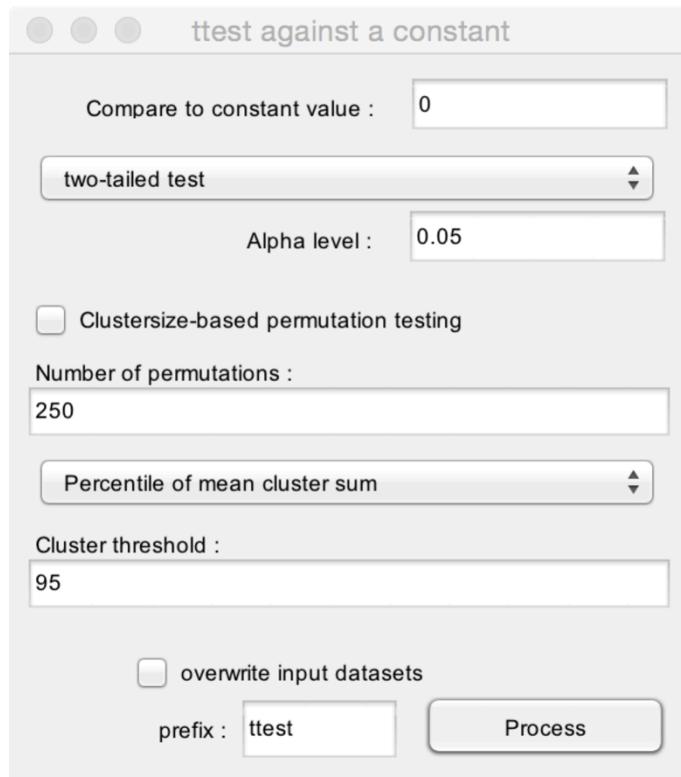
Several methods have been proposed to define the threshold distinguishing between “significant” and “non significant” clusters:

- **Standard deviation of mean cluster sum.** The threshold is defined as a z-score using the mean and standard deviation of the distribution of all significant clusters obtained using permutation testing (e.g. Z>2).
- **Standard deviation of maximum cluster sum.** The threshold is defined as a z-score using the mean and standard deviation of the distribution of the most significant cluster obtained for each permutation testing (e.g. Z>2).
- **Percentile of mean cluster sum.** The threshold is defined as a percentile using the distribution  $z$  of the distribution of all significant clusters obtained using permutation testing (e.g. >95%; this is equivalent to setting the cluster threshold to  $p < 0.05$ ).
- **Percentile of mean cluster sum.** The threshold is defined as a percentile using the distribution  $z$  of the distribution of the most significant cluster obtained for each permutation testing (e.g. >95%; this is equivalent to setting the cluster threshold to  $p < 0.05$ ).

## COMPARE SIGNALS AGAINST A CONSTANT

Compute a point-by-point t-test to assess whether a signal is significantly different from a constant (e.g. test whether a signal is significantly different from zero). The function also implements permutation testing to identify significant clusters in the obtained statistical time-courses or maps.

To use this function, you must create a merged dataset, where each epoch corresponds to a measure (e.g. the average waveform of a given subject).



- **Constant value.** The value of the constant against which the dataset should be tested.
- Select whether you want to perform a **two-tailed test**, a **right-tailed test**, or a **left-tailed test**. Add the factor as a within-subject factor.
- **Alpha level.** The minimum p-value to consider the result of the statistical test as "significant".

The main issue of point-by-point statistical testing is the correction for multiple comparisons. Permutation testing can be a mean to address this issue. See <http://www.ncbi.nlm.nih.gov/pubmed/17517438>. Permutation testing computes a distribution of the clusters of significant p values obtained when computing the statistical tests after permuting the datasets. The “strength” of each cluster is estimated using the sum of the test statistic within each significant cluster obtained using the permuted datasets. This distribution of clusters is then used to define a cluster threshold. The cluster threshold is applied to the result of the statistical test applied on the non-permuted datasets, such as to discard all clusters which are below the cluster threshold.

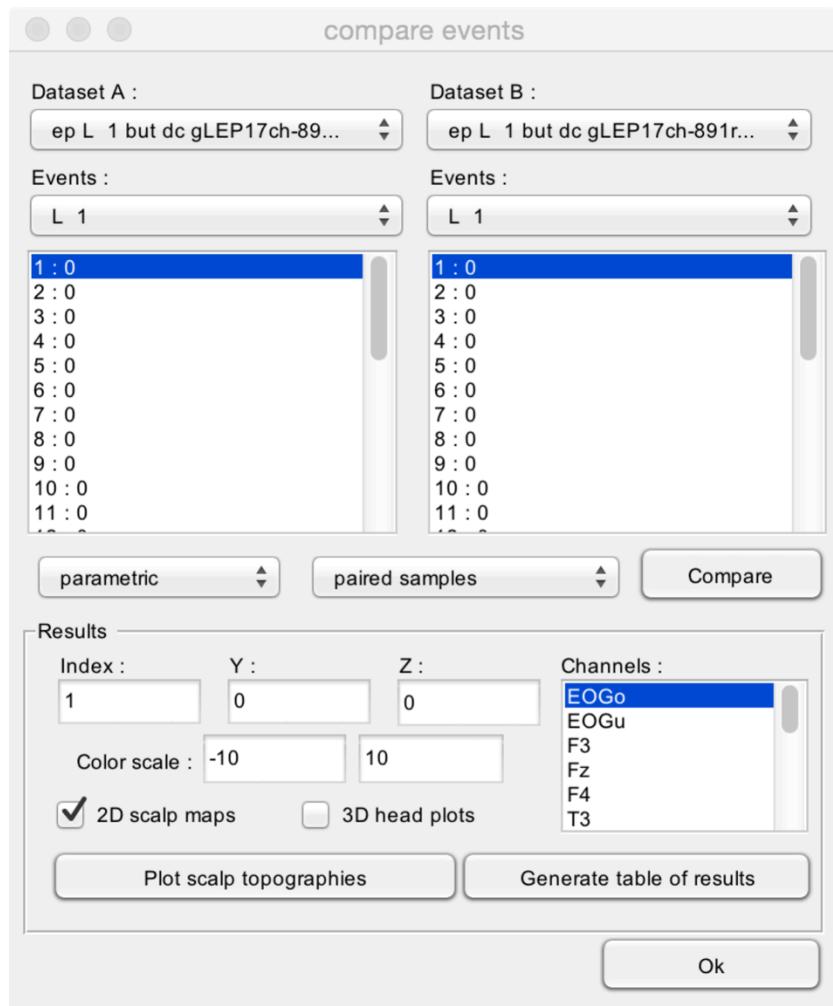
- **Clustersize-based permutation testing.** Perform a clustersize-based permutation testing. This can be very time consuming.
- **Number of permutations.** The number of permutations to be used to build the clustersize distribution.

Several methods have been proposed to define the threshold distinguishing between “significant” and “non significant” clusters:

- **Standard deviation of mean cluster sum.** The threshold is defined as a z-score using the mean and standard deviation of the distribution of all significant clusters obtained using permutation testing (e.g.  $Z>2$ ).
- **Standard deviation of maximum cluster sum.** The threshold is defined as a z-score using the mean and standard deviation of the distribution of the most significant cluster obtained for each permutation testing (e.g.  $Z>2$ ).
- **Percentile of mean cluster sum.** The threshold is defined as a percentile using the distribution  $z$  of the distribution of all significant clusters obtained using permutation testing (e.g. >95%; this is equivalent to setting the cluster threshold to  $p < 0.05$ ).
- **Percentile of mean cluster sum.** The threshold is defined as a percentile using the distribution  $z$  of the distribution of the most significant cluster obtained for each permutation testing (e.g. >95%; this is equivalent to setting the cluster threshold to  $p < 0.05$ ).

## COMPARE SIGNAL AMPLITUDE AT THE EVENT LATENCIES IN TWO DATASETS

This function can be used to perform a statistical comparison of signal amplitudes and latencies measured at the event latencies in two datasets (e.g. the event latencies identifying a given ERP peak in two datasets).



- **Dataset A** and **Dataset B**. The datasets that should be compared.
- **Events**. The event codes that should be compared in each dataset. The latencies of the selected event codes are shown in the lower listboxes.
- **Test type**. The comparisons can be performed using a **parametric test** (t-test) or a **non-parametric test** (Wilcoxon test), for **paired samples** or **two independent samples**. If you want to perform a paired-sample comparison, each epoch of each dataset must correspond to the same subject.
- **Compare** computes the statistical comparison.

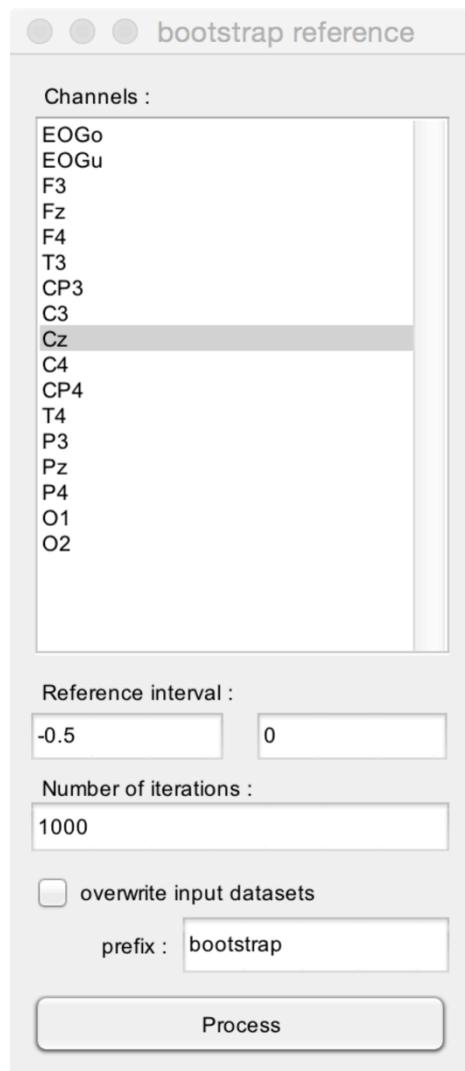
After performing the test, the **Results** table can be used to display or export results.

- **Index**. If your dataset has more than one index, you can choose which index to display.
- **Y/Z**. If your dataset includes Y- and/or Z-dimensions, you can choose which position along the Y- and Z-dimensions should be displayed.
- **Channels**. Choose one or more channels to display.

- **Plot scalp topographies.** Plot statistical scalp maps showing the results of the statistical test obtained at all scalp channels.
- **Generate table of results.** Generate a table of results. This table can then be exported into Excel or another software for further analysis.

## BOOTSTRAP TEST AGAINST A REFERENCE INTERVAL

Compute a bootstrap test against a reference interval.

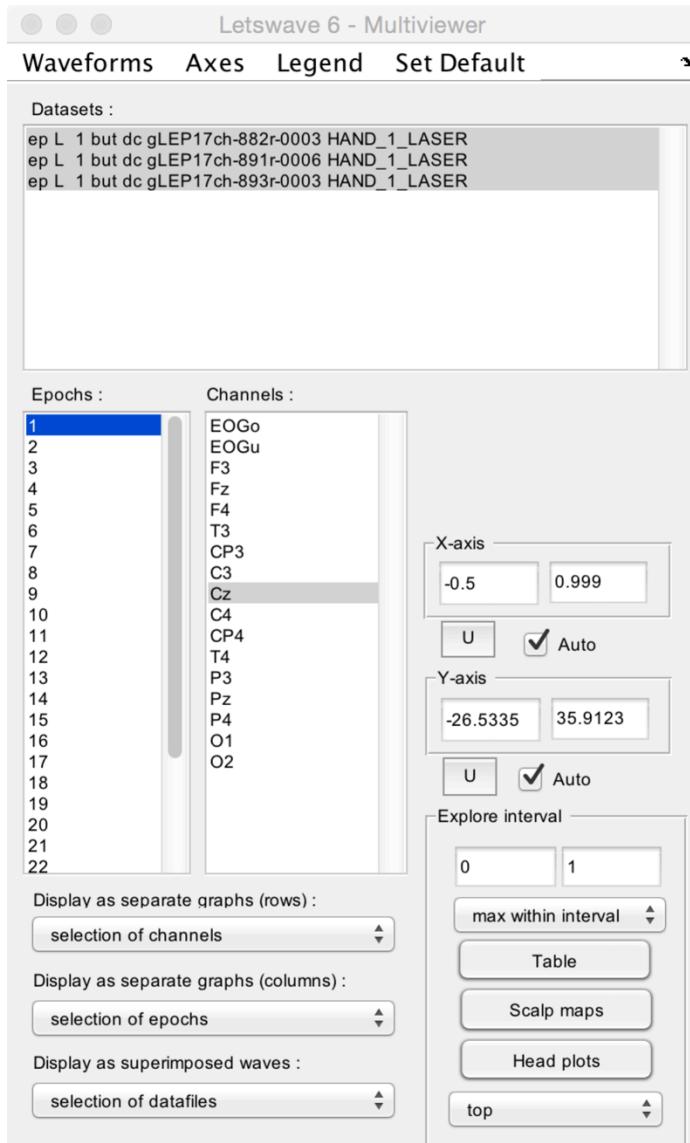


- **Channels.** The channels onto which the bootstrap test should be computed.
- **Reference interval.** The reference interval for the bootstrap test.
- **Number of iterations** to perform the bootstrap.

**VIEW**

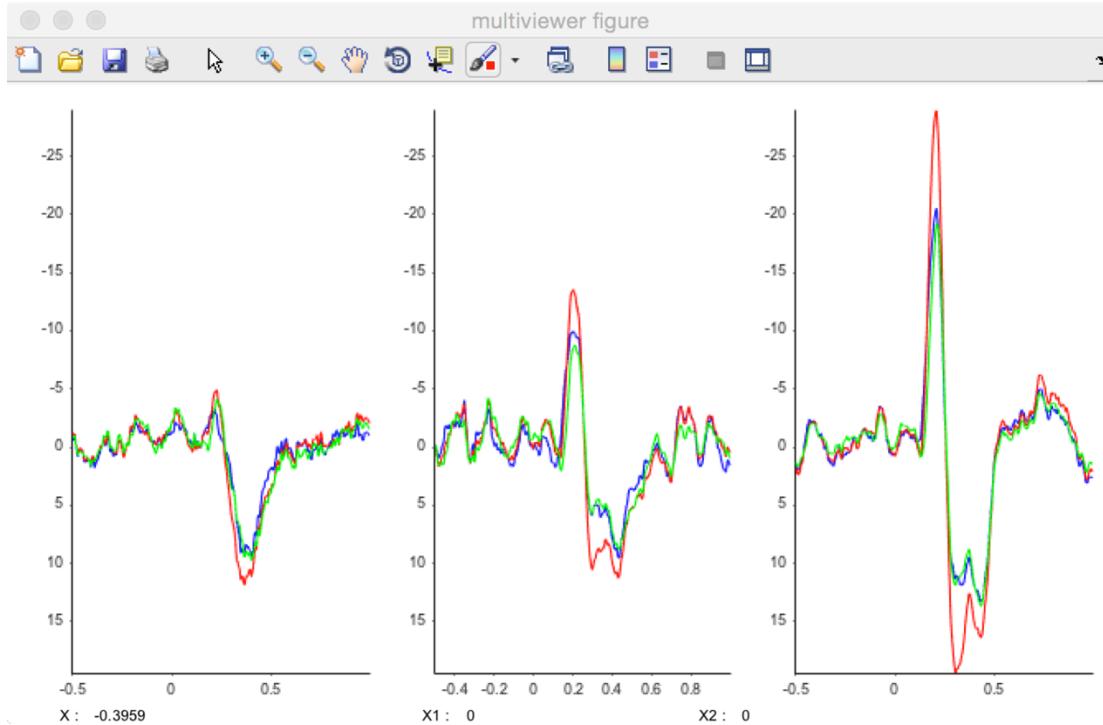
## WAVEFORM MULTI-VIEWER

This is the default viewer for epoched data or averaged waveforms.



- **Datasets.** Select one or more datasets to display.
- **Epochs.** Select one or more epochs to display.
- **Channels.** Select one or more channels to display.
- **Display as separate graphs (rows).** Choose what should be displayed in separate graphs (organized as rows). This can be the selected datasets, epochs, or channels.
- **Display as separate graphs (columns).** Choose what should be displayed in separate graphs (organized as columns). This can be the selected datasets, epochs, or channels.
- **Display as superimposed waves.** Choose what should be displayed as superimposed waves across graphs. This can be the selected datasets, epochs, or channels.
- **X-axis and Y-axis.** Manually set the X- and Y-axis range, or set the range to **Auto** for automatic adjustment. Because the X- and Y-axis range can also be modified using the zoom and navigation tools of the Graph window, the **U** button can be used to update X- and Y-axis settings using the currently displayed range.

- **Explore interval.** A time interval can be defined by editing the lower and upper limits, or by selecting a range in the Graph window. After defining the interval, the **Table** button can be used to generate a table of descriptive statistics of the data within the interval. The **Scalp maps** button can be used to generate 2D scalp maps of the maximum (e.g. a positive ERP peak) or minimum (e.g. a negative ERP peak) values in the interval (defined using the **Maximum within interval** or **Minimum within interval** dropdown list). The **Head plots** button can be used to generate a 3D head plot of the signals.

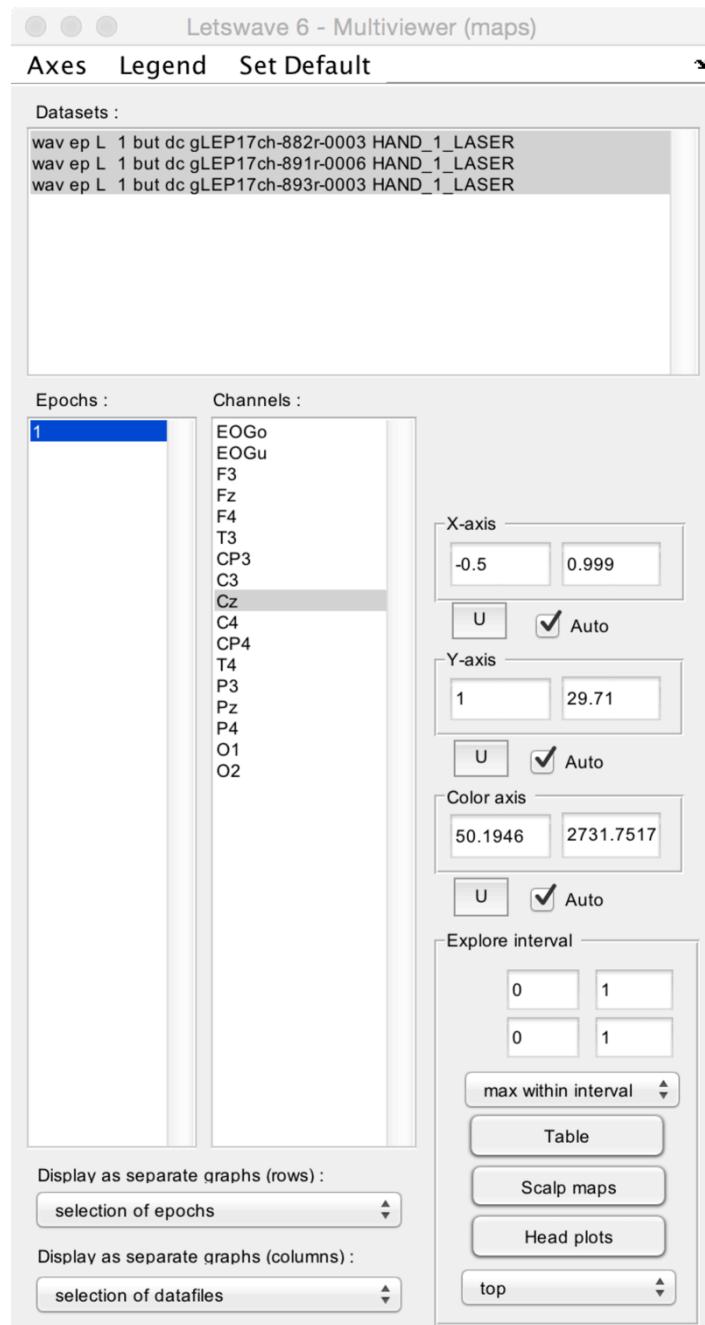


Several display options can be adjusted using the dropdown title menu:

- **Waveforms.** The waveforms can be displayed as a **plot** graph, a **stem** graph, or a **stair** graph.
- **Axes.** The direction of the Y-axis can be set to **normal** (positive upwards), or **reverse** (negative upwards).
- **Legend.** A legend can be added to the graphs, displaying the name of the dataset, channel, or epoch.
- **Set default.** Set the current settings as default.

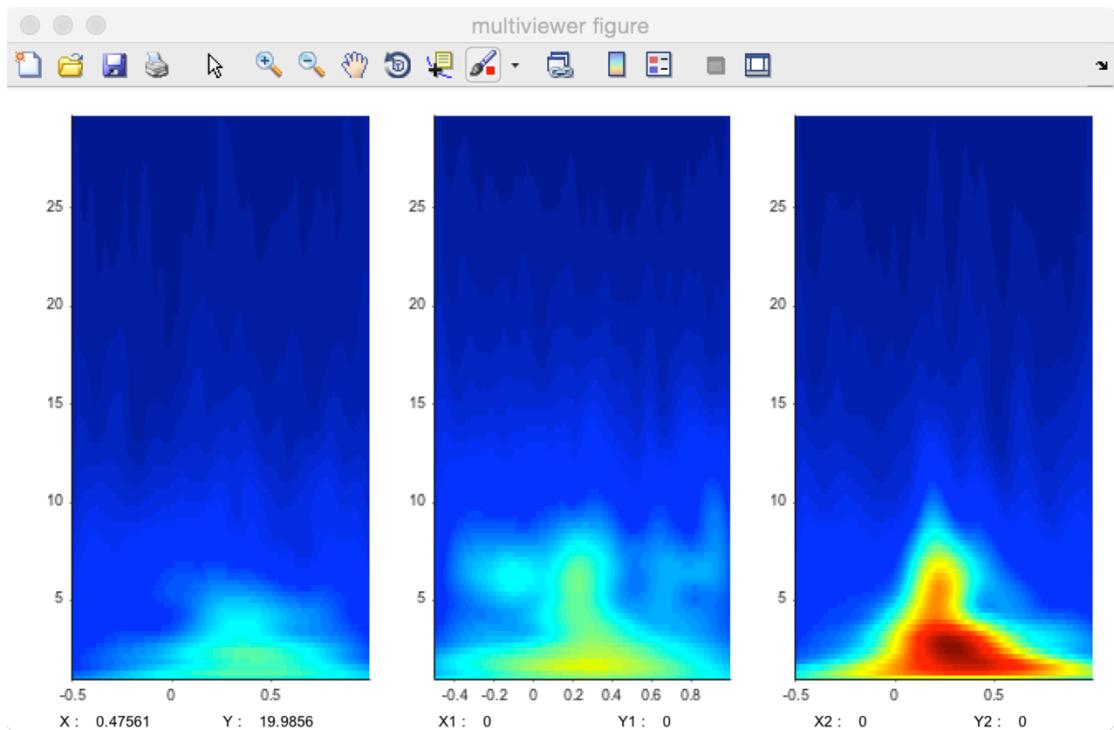
## MAP MULTI-VIEWER

This is the default viewer for time-frequency maps.



- **Datasets.** Select one or more datasets to display.
- **Epochs.** Select one or more epochs to display.
- **Channels.** Select one or more channels to display.
- **Display as separate graphs (rows).** Choose what should be displayed in separate graphs (organized as rows). This can be the selected datasets, epochs, or channels.
- **Display as separate graphs (columns).** Choose what should be displayed in separate graphs (organized as columns). This can be the selected datasets, epochs, or channels.

- **X-axis and Y-axis.** Manually set the X- and Y-axis range, or set the range to **Auto** for automatic adjustment. Because the X- and Y-axis range can also be modified using the zoom and navigation tools of the Graph window, the **U** button can be used to update X- and Y-axis settings using the currently displayed range.
- **Explore interval.** A X-axis and a Y-axis interval can be defined by editing the lower and upper limits, or by selecting a range in the Graph window. After defining the interval, the **Table** button can be used to generate a table of descriptive statistics of the data within the interval. The **Scalp maps** button can be used to generate 2D scalp maps of the maximum (e.g. event-related synchronization) or minimum (e.g. event-related desynchronization) values in the interval (defined using the **Maximum within interval** or **Minimum within interval** dropdown list). The **Head plots** button can be used to generate a 3D head plot of the signals.

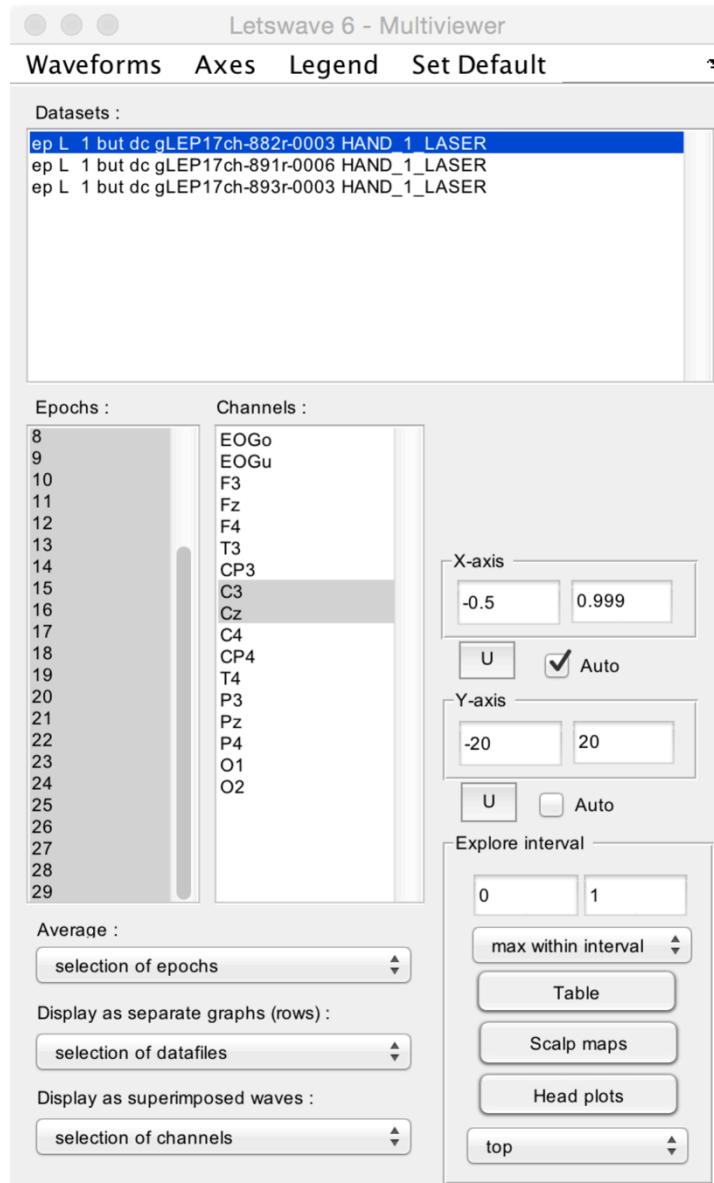


Several display options can be adjusted using the dropdown title menu:

- **Axes.** A grid can be added onto the 2D maps.
- **Legend.** A legend can be added to the graphs, displaying the name of the dataset, channel, or epoch.
- **Set default.** Set the current settings as default.

## WAVEFORM AVERAGE MULTI-VIEWER

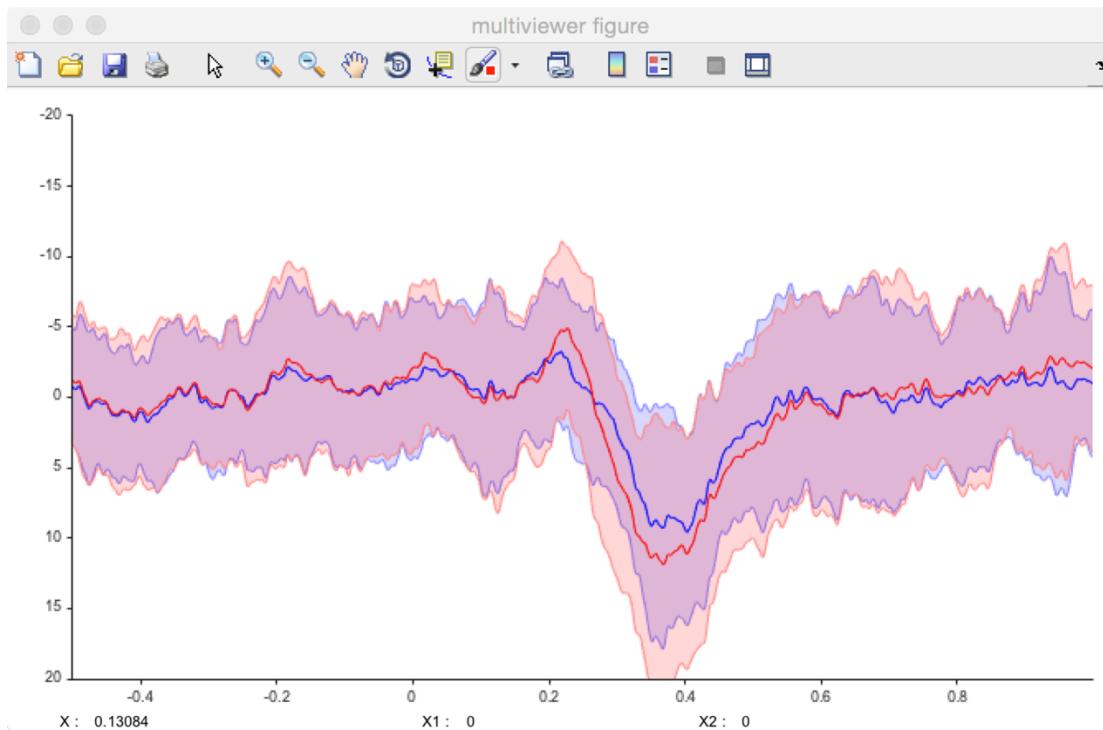
This viewer is designed to view average waveforms of epoched data.



- **Datasets.** Select one or more datasets to display.
- **Epochs.** Select one or more epochs to display.
- **Channels.** Select one or more channels to display.
- **Average.** Choose what should be averaged. This can be the selected datasets, epochs, or channels. In the present example, we have chosen to average the selected epochs.
- **Display as separate graphs (rows).** Choose what should be displayed in separate graphs (organized as rows). This can be the selected datasets, epochs, or channels.
- **Display as superimposed waves.** Choose what should be displayed as superimposed waves across graphs. This can be the selected datasets, epochs, or channels.
- **X-axis and Y-axis.** Manually set the X- and Y-axis range, or set the range to **Auto** for automatic adjustment. Because the X- and Y-axis range can also be modified using the zoom

and navigation tools of the Graph window, the **U** button can be used to update X- and Y-axis settings using the currently displayed range.

- **Explore interval.** A time interval can be defined by editing the lower and upper limits, or by selecting a range in the Graph window. After defining the interval, the **Table** button can be used to generate a table of descriptive statistics of the data within the interval. The **Scalp maps** button can be used to generate 2D scalp maps of the maximum (e.g. a positive ERP peak) or minimum (e.g. a negative ERP peak) values in the interval (defined using the **Maximum within interval** or **Minimum within interval** dropdown list). The **Head plots** button can be used to generate a 3D head plot of the signals.

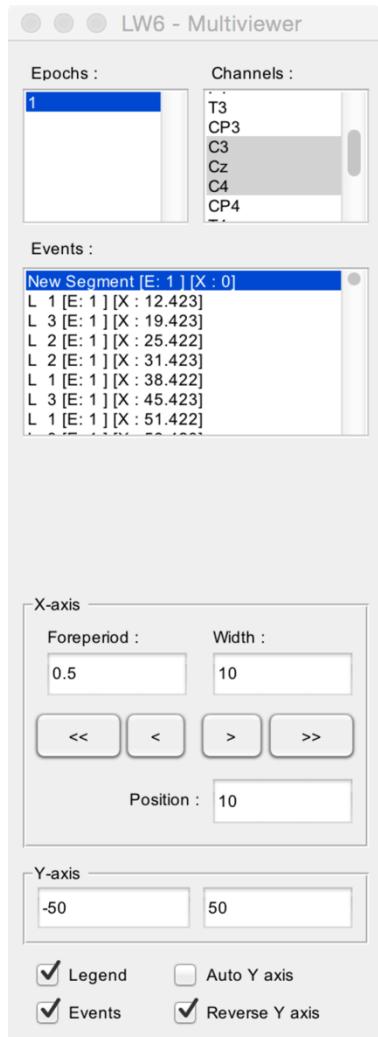


Several display options can be adjusted using the dropdown title menu:

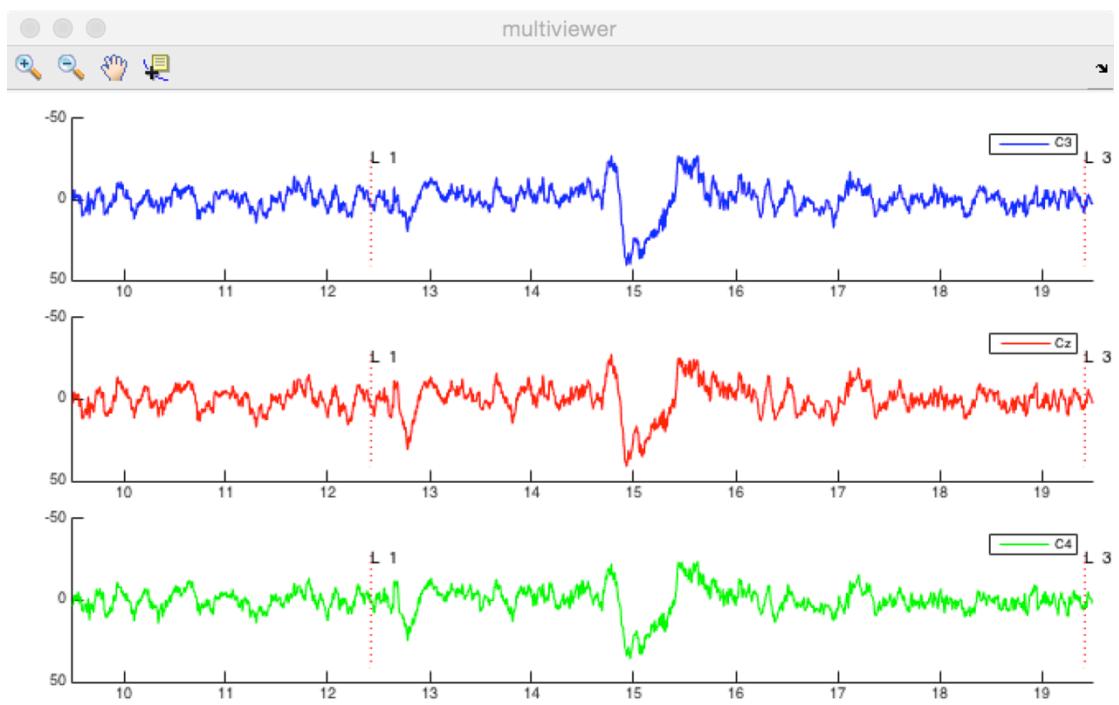
- **Waveforms.**
  - Display mean.** Only display average waveform.
  - Display mean ± SD.** Display average waveforms and standard deviation from the mean as a light area.
  - Display median.** Only display median waveforms.
  - Display median ± IQR.** Display median waveforms and interquartile ranges as a light area.
- **Axes.** The direction of the Y-axis can be set to **normal** (positive upwards), or **reverse** (negative upwards).
- **Legend.** A legend can be added to the graphs, displaying the name of the dataset, channel, or epoch.
- **Set default.** Set the current settings as default.

## WAVEFORM CONTINUOUS DATA VIEWER

This viewer is designed to view continuous (EEG) data.

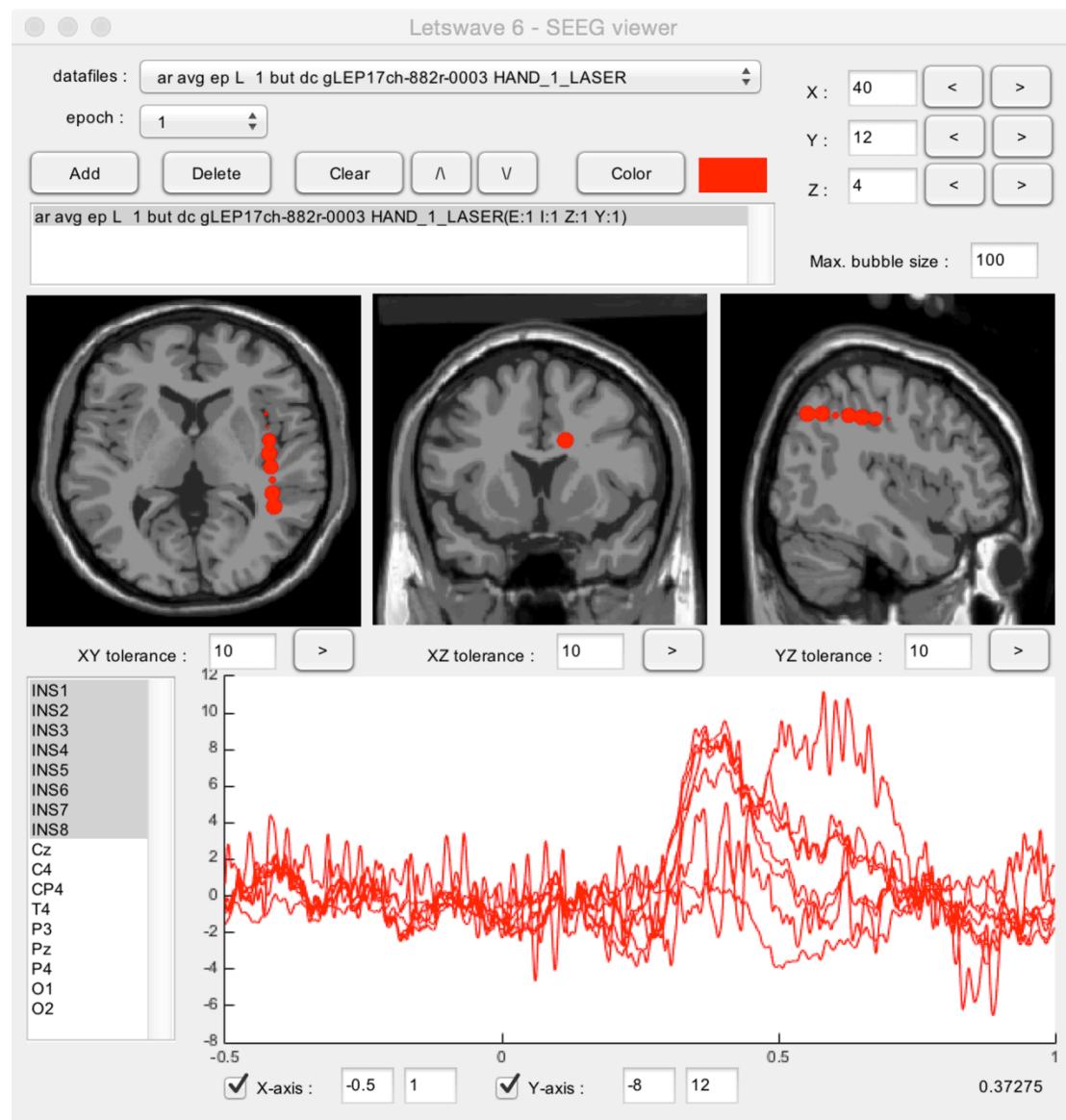


- **Epochs.** Select one or more epochs to display.
- **Channels.** Select one or more channels to display.
- **Events.** All events of the dataset are displayed sequentially in the **Events** listbox. The signal measured after the occurrence of an event can be viewed by selecting that event. The listbox also provides information as to the epoch associated with the event ([E]), and the latency of the event ([X]).
- **X-axis.** Define the time interval displayed in the graphs. The **Foreperiod** defines the time-interval to display before the onset of an event. The **Width** defines the width of the displayed signal window. The < and > buttons can be used to move backward and forward in the signal. The << and >> buttons can be used to move to the beginning or end of the recording. The actual position in the dataset is defined by the **Position** field.
- **Y-axis.** Manually set the Y-axis range, or set the range to **Auto Y-axis** for automatic adjustment.
- **Legend.** Display a legend on the graphs.
- **Events.** Display the events in the graphs.
- **Reverse Y axis.**

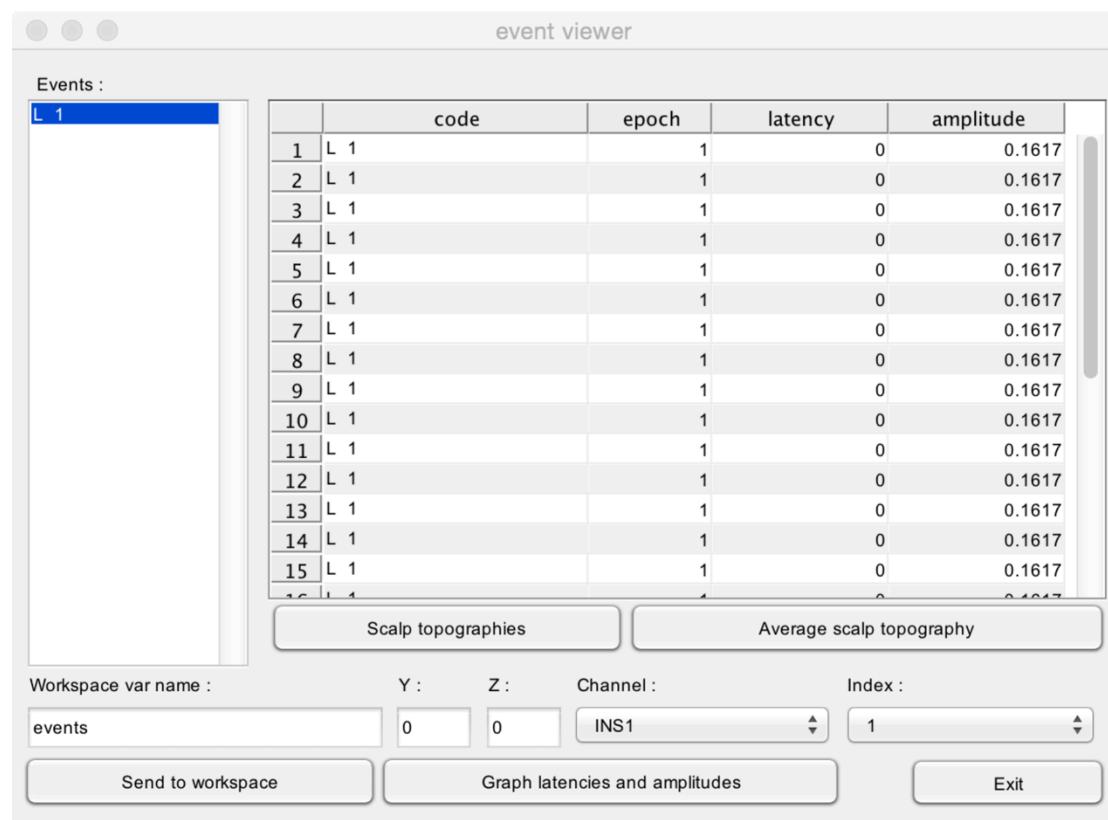


## SEEG VIEWER

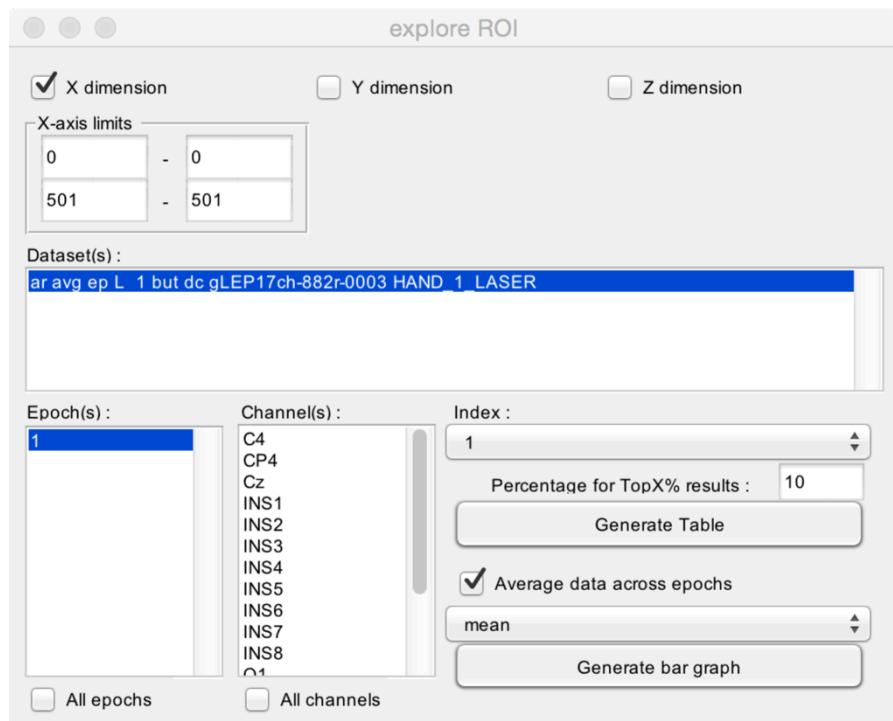
This viewer is designed to view SEEG data. This function is still under development.



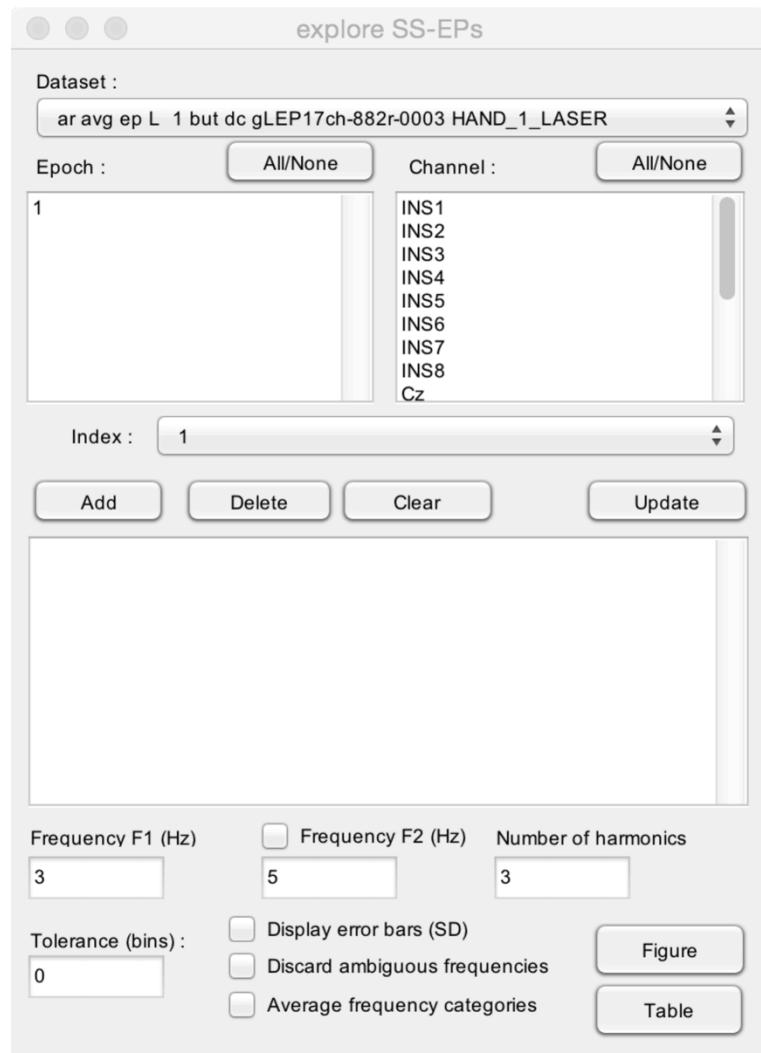
## EVENT VIEWER: VIEW SIGNALS AT EVENT LATENCIES



## GENERATE TABLE AND GRAPHS WITH ROI DATA



## GENERATE TABLE AND GRAPHS WITH SS-EP DATA



## EPOCH DATA VIEWER

view epochdata

Dataset :  
ar avg ep L 1 but dc gLEP17ch-882r-0003 HAND\_1\_L...

Epoch :  
1

Epoch data fields :

	field name	value
1	RT	0.8147

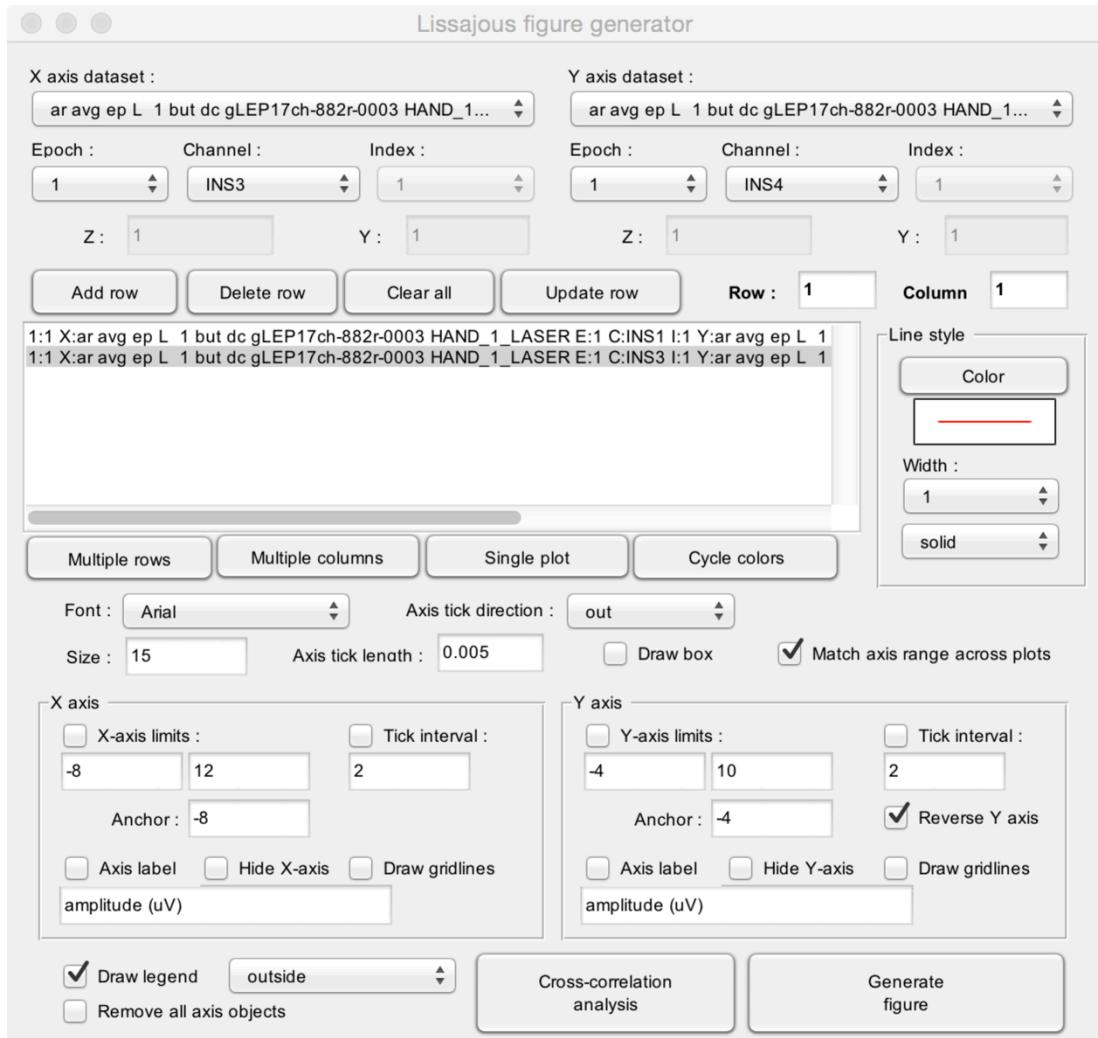
RT

Send to workspace      Graph

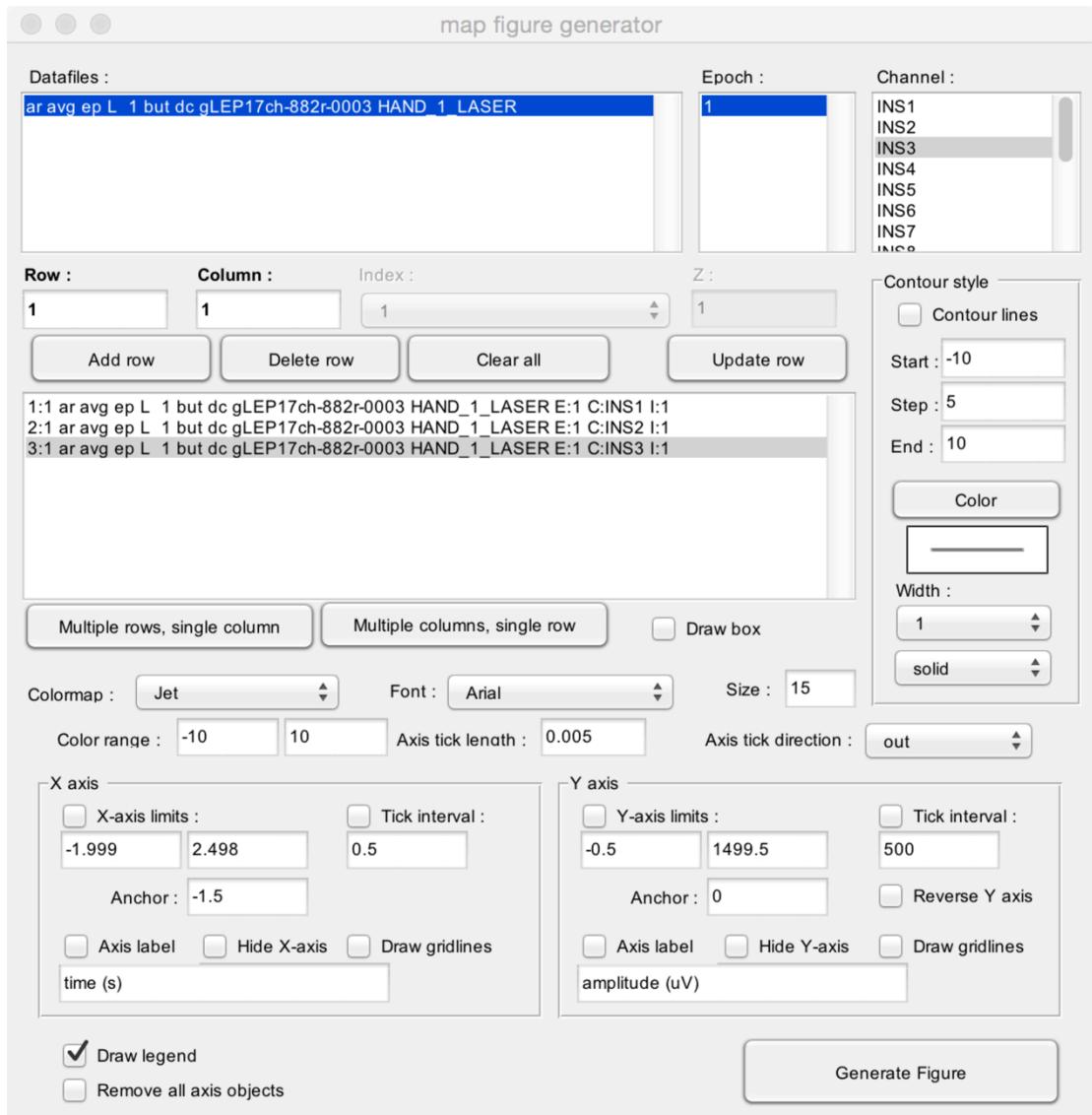
The screenshot shows a software interface titled "view epochdata". At the top, there are three dropdown menus: "Dataset" containing the text "ar avg ep L 1 but dc gLEP17ch-882r-0003 HAND\_1\_L...", "Epoch" containing the number "1", and "Epoch data fields". Below these is a table with one row, showing a field named "RT" with a value of "0.8147". At the bottom of the window are two buttons: "Send to workspace" and "Graph".

## FIGURES

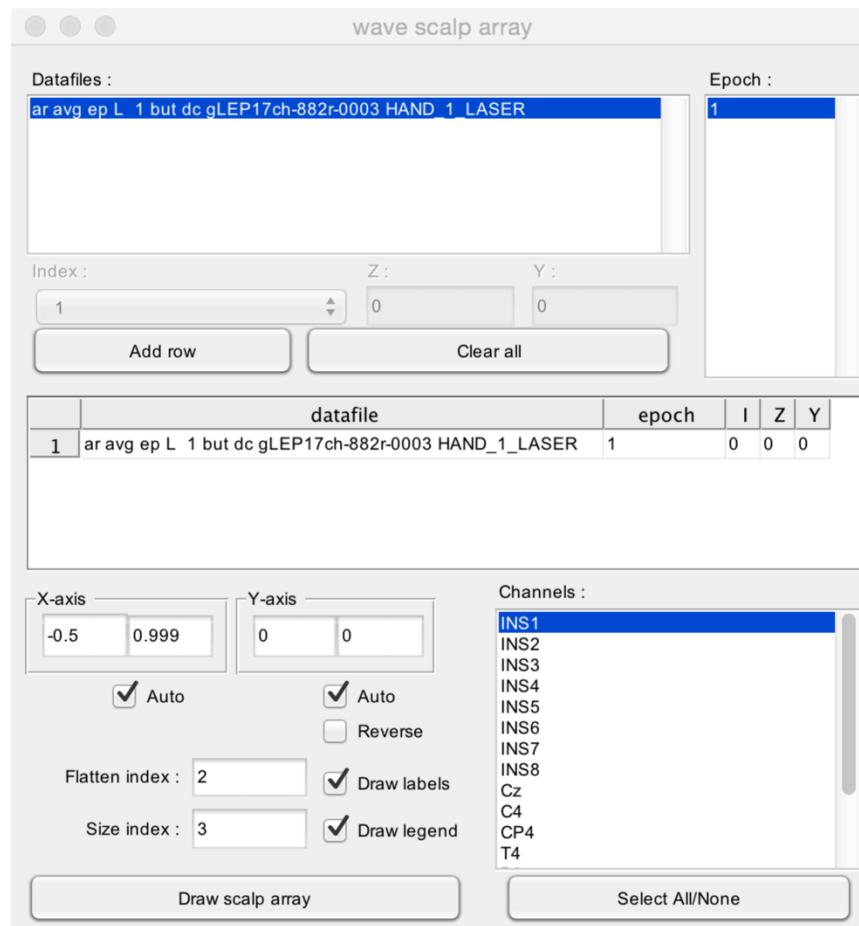
### LISSAJOUS FIGURE CREATOR

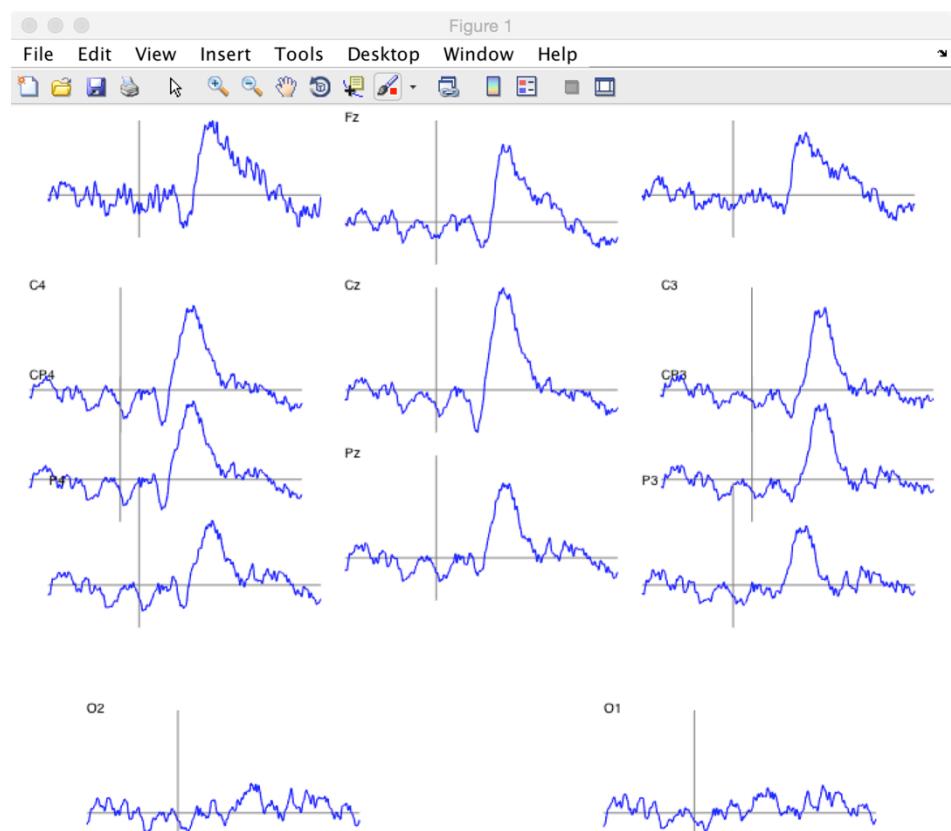


## MAPS FIGURE CREATOR



## WAVE SCALP ARRAY FIGURE CREATOR





## SCALP MAP SS-EP SERIES FIGURE CREATOR

scalp map SSEP series

Datafiles : avg ep L\_1 but dc gLEP17ch-882r-0003 HAND\_1\_LASER

Epoch : 1

Index : 1 Z : 0 Y : 0

Add row Clear all

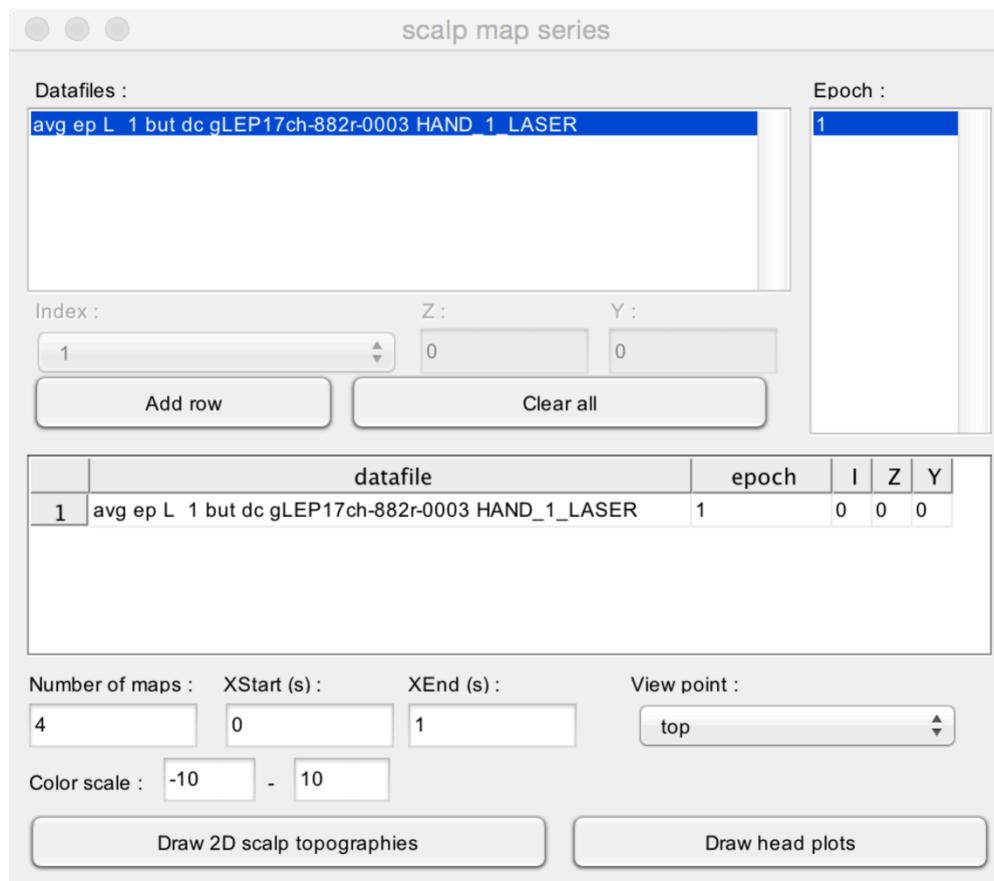
	datafile	epoch	I	Z	Y
1	avg ep L_1 but dc gLEP17ch-882r-0003 HAND_1_LASER	1	0	0	0

Frequency (Hz) : 10 Number of harmonics : 4 Tolerance (bins) : 2 View point : top

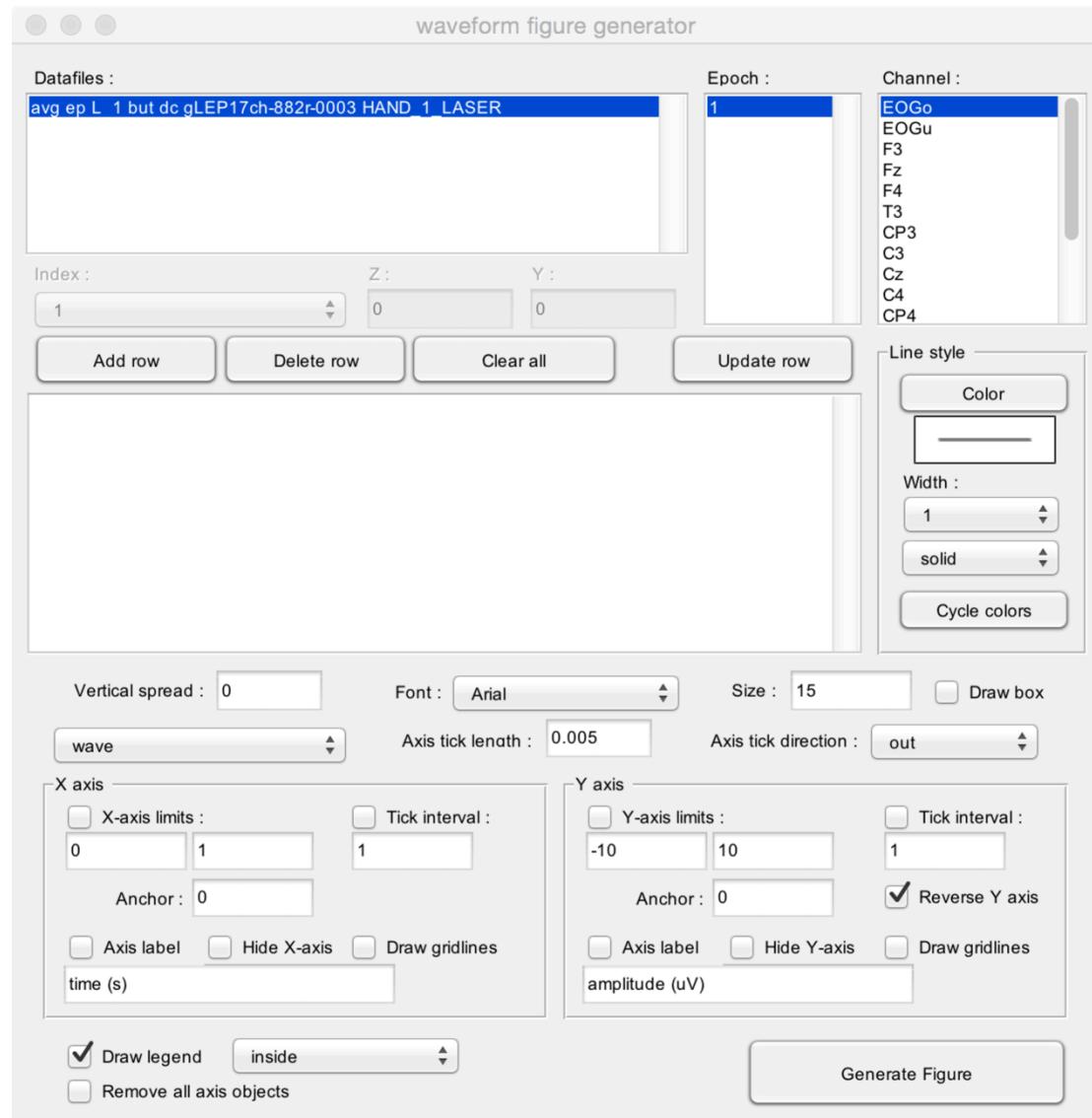
Color scale : -10 - 10

Draw 2D scalp maps Draw 3D head plots

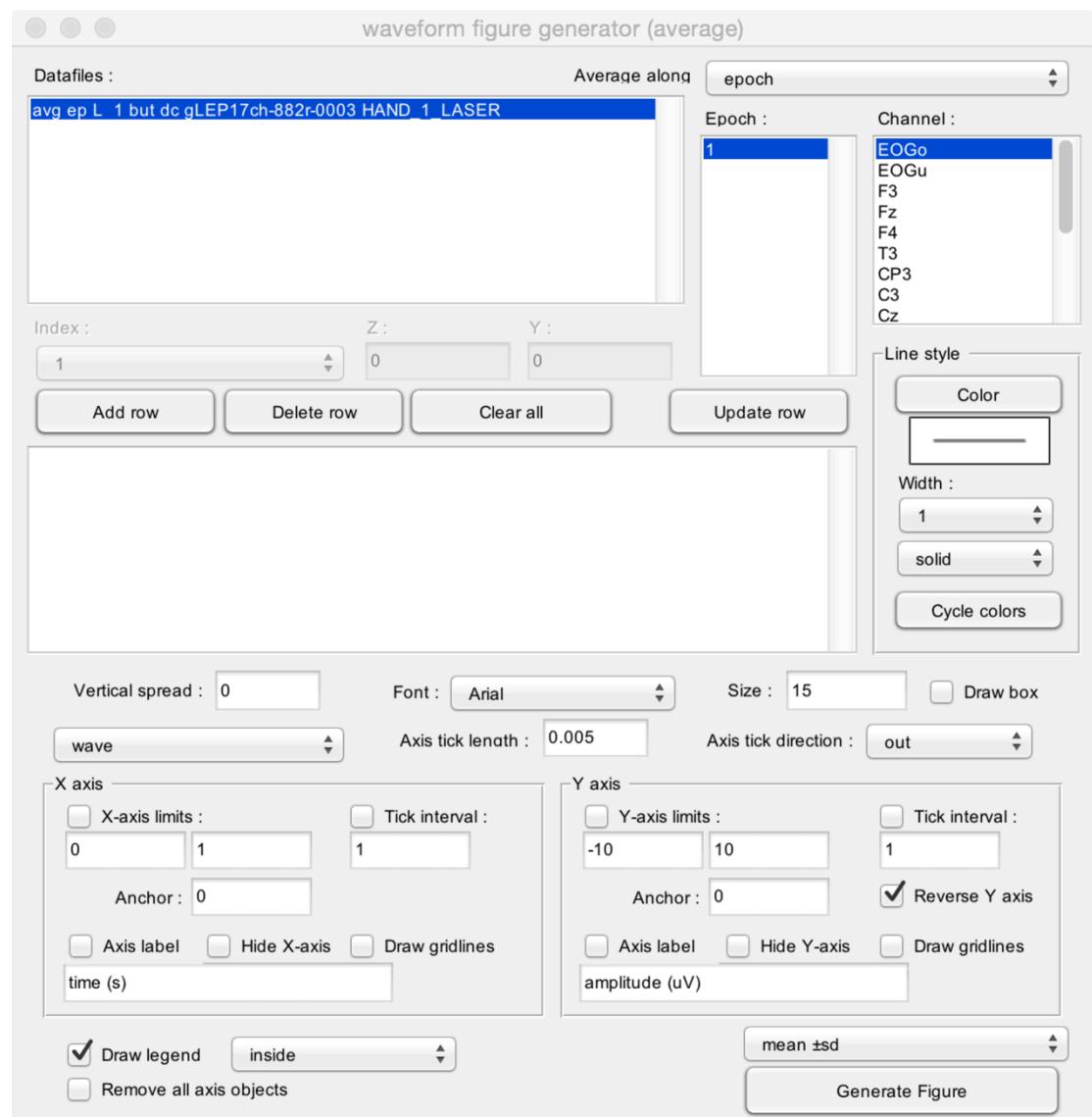
## SCALP MAP TIME SERIES FIGURE CREATOR



## WAVEFORMS FIGURE CREATOR

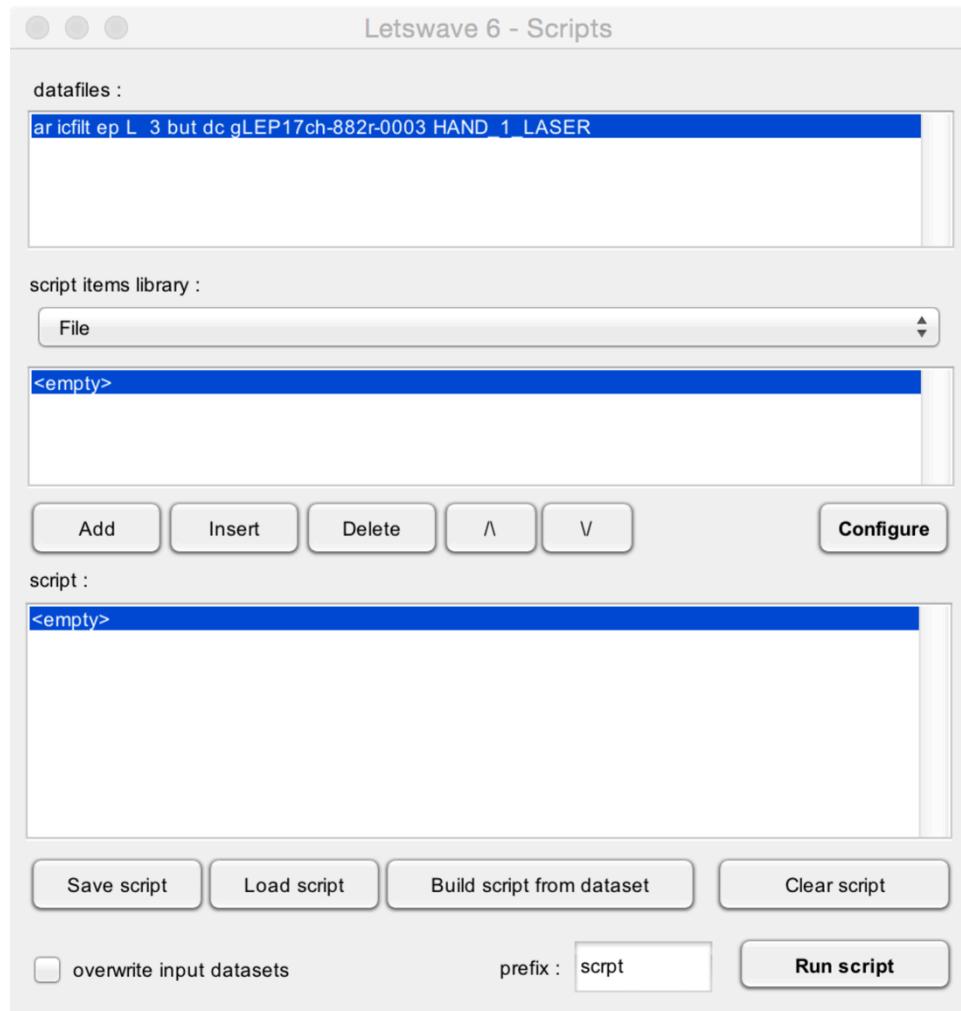


## WAVEFORMS AVERAGE FIGURE CREATOR



## SCRIPTS

Scripts make it possible to apply a succession of operations to one or more datasets.



- **Datafiles.** The datasets that onto which the operations will be performed.

The first step is to add functions to your script. This can be done using the Add, Insert, Delete, Up and Down buttons:

- **Script items library.** This listbox contains all the categories of scriptable functions. When you choose a given category, the lower listbox displays all available functions for that category.
- **Add.** Add the selected function to the script.
- **Insert.** Add the selected function to the script, at the currently selected position.

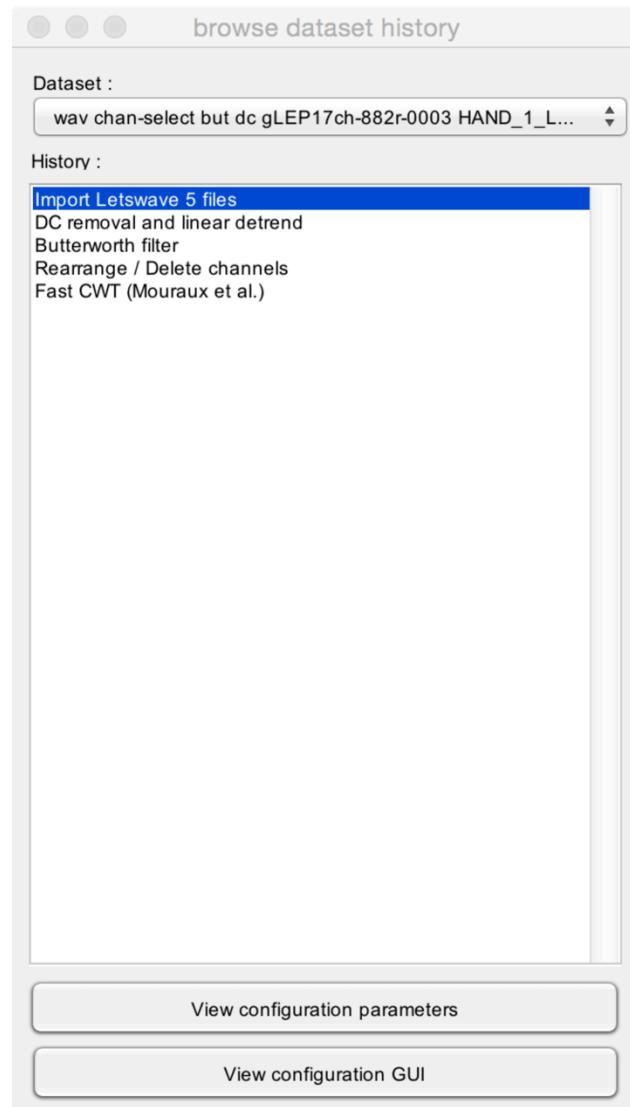
- **Delete.** Delete the selected function from the script.
- **Up and Down.** Move the selected function up or down in the list.

The next step is to configure each function. This can be done by selecting each function of the script, and clicking the ‘Configure’ button.

- **Save script.** Save the script as a script file.
- **Load script.** Load a script file.
- **Build script from dataset.** If you have already processed a dataset, you can create a script using the history of that dataset. The script will include all the functions and configuration that have been applied to the dataset.
- **Clear script.** Clear the script.
- **Run script.** Run the script.

## HISTORY

Browse the history of a dataset.

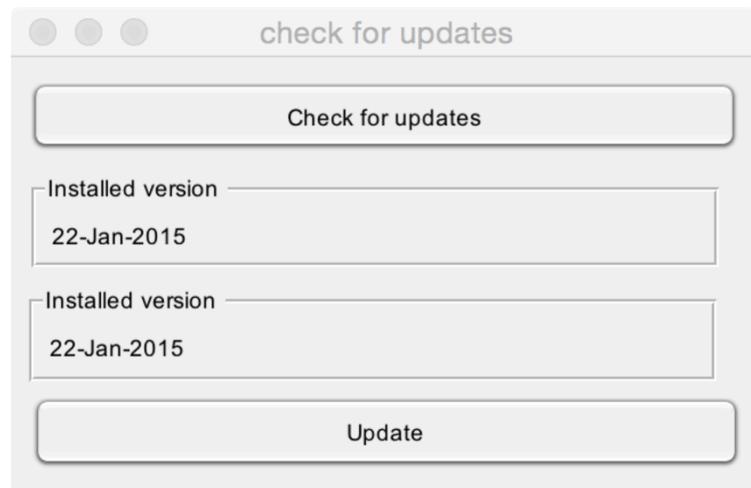


- **History.** The list of all operations that have been applied to the current dataset.
- **View configuration parameters.** View the parameters of the selected operation.
- **View configuration GUI.** View the parameters of the selected operation (displayed using the GUI used to define these parameters).

## UPDATES

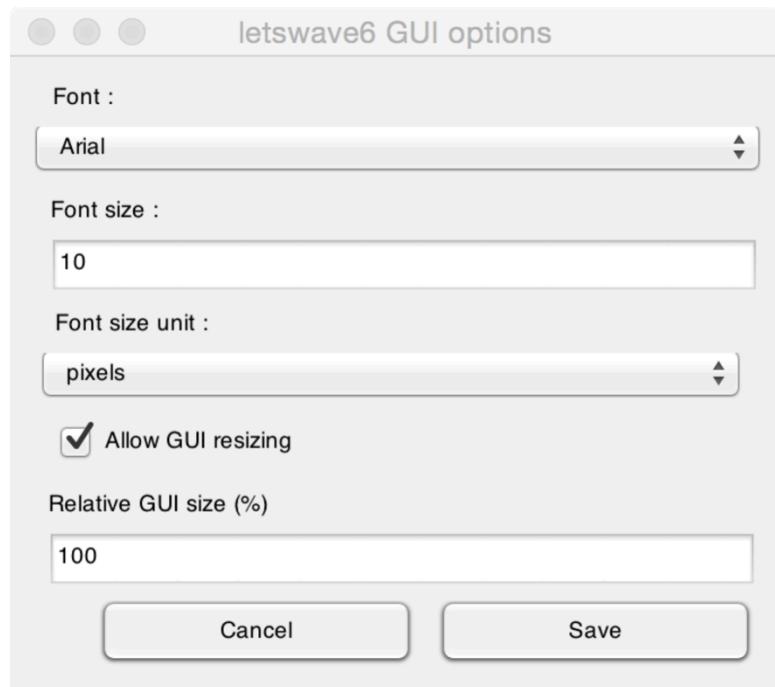
Check for updates.

Note that your firewall configuration must allow Matlab to connect to the internet.



- **Check for updates.** Connect to the Letswave server and check for updates.
- **Installed version.** The timestamp of the Letswave version currently installed.
- **Available version.** The timestamp of the Letswave version available on the server.
- **Update.** Download the latest Letswave version and update.

## GUI OPTIONS



In Letswave 6, it is now possible to set GUI display options. These options will modify the display of all the GUI interfaces. This makes it possible to optimize display according to your OS, screen resolution and screen real-estate.

- **Font.** The font used for all GUIs.
- **Font size.** The font size used for all GUIs.
- **Font size unit.** See Matlab for details.
- **Allow GUI resizing.** Make it possible to resize all GUIs.
- **Relative GUI size.** Increase or decrease the size of all GUIs.

