CARE-rCortex Tutorial

What is CARE-rCortex?

CARE-rCortex is an open source and interactive EEGLAB plugin for analyzing the electroencephalographic (EEG) basis of CArdio-REspiratory activities related to the Cortex in time-frequency domain.

The plugin is a user-friendly toolbox allowing users to automatically detect cardiac or respiratory events. It offers the possibility to the users to add, remove or modify the detected events by visual inspection. These events can be used to define adaptive baselines related to physiological cycles. These baselines can be also easily checked and modified by the user. By this way, CARE-rCortex can normalize EEG data by baselines time-locked to the detected cardiac or respiratory events and compute more accurate time-frequency maps. The plugin provides 4 types of baseline correction and is able to compute the significant parts of the time-frequency map with respect to the baseline. CARE-rCortex provides two types of time-frequency computation. The first one highlights the EEG activities of the epoch average in time-frequency domain (evoked time-frequency map). The other represents the average of the time-frequency maps across the epochs (induced time-frequency map).

For further statistical analysis, CARE-rCortex stores the computed time-frequency maps and their significant masks (when they are computed) in the existing EEGLAB structure.

Requirements

CARE-rCortex requires Matlab and EEGLAB.

Matlab version:
The toolbox works with all the version of Matlab older than R2013a release, until R2017b. The other Matlab versions were not tested yet.

EEGLAB version:
The eeglab_14_1_1b release of EEGLAB was used to developed CARE-rCortex. We advise you to use the same version of EEGLAB as the other versions were not tested.

Operating system:
The toolbox works without problem on all operating systems.

Compatible EEG data format:
Several data formats are available in EEGLAB. However, only Brain Vision Recording .vhdr files, .edf files and EEGLAB .set files were tested. If you use another format, please report if you succeed in using this other format or if you encounter some problems.

Warning: some data formats require to download other EEGLAB plugin(s).

Download

Click here to download the latest CARE-rCortex version. The archive contains a pdf tutorial to use CARE-rCortex.

Disclaimer: CARE-rCortex is a plugin developed for research purposes only with no guarantee of suitability for any particular purpose. This plugin should not under any circumstances be used for clinical purposes.

How install CARE-rCortex?

To install CARE-rCortex:

1. Unzip the CARE-rCortex zip file in the plugins folder of the EEGLAB folder you have.
2. Start Matlab
3. Change the Matlab path to the EEGLAB folder you have.
4. Type "eeglab" and press enter on the Matlab prompt. CARE-rCortex plugin will be automatically detected by EEGLAB with the other plugin(s) you have already installed.

Prepare the data with EEGLAB

In order to use CARE-rCortex, you have to start Matlab and EEGLAB as it is explained in the previous section.

Then, you need to **load data** in EEGLAB. To do that, you can choose to load a new dataset (File > Import data > …) or to load an existing EEGLAB dataset (File > Load existing dataset).

If it is not already provided by your dataset, you should **edit the channel locations** (Edit > Channels location).

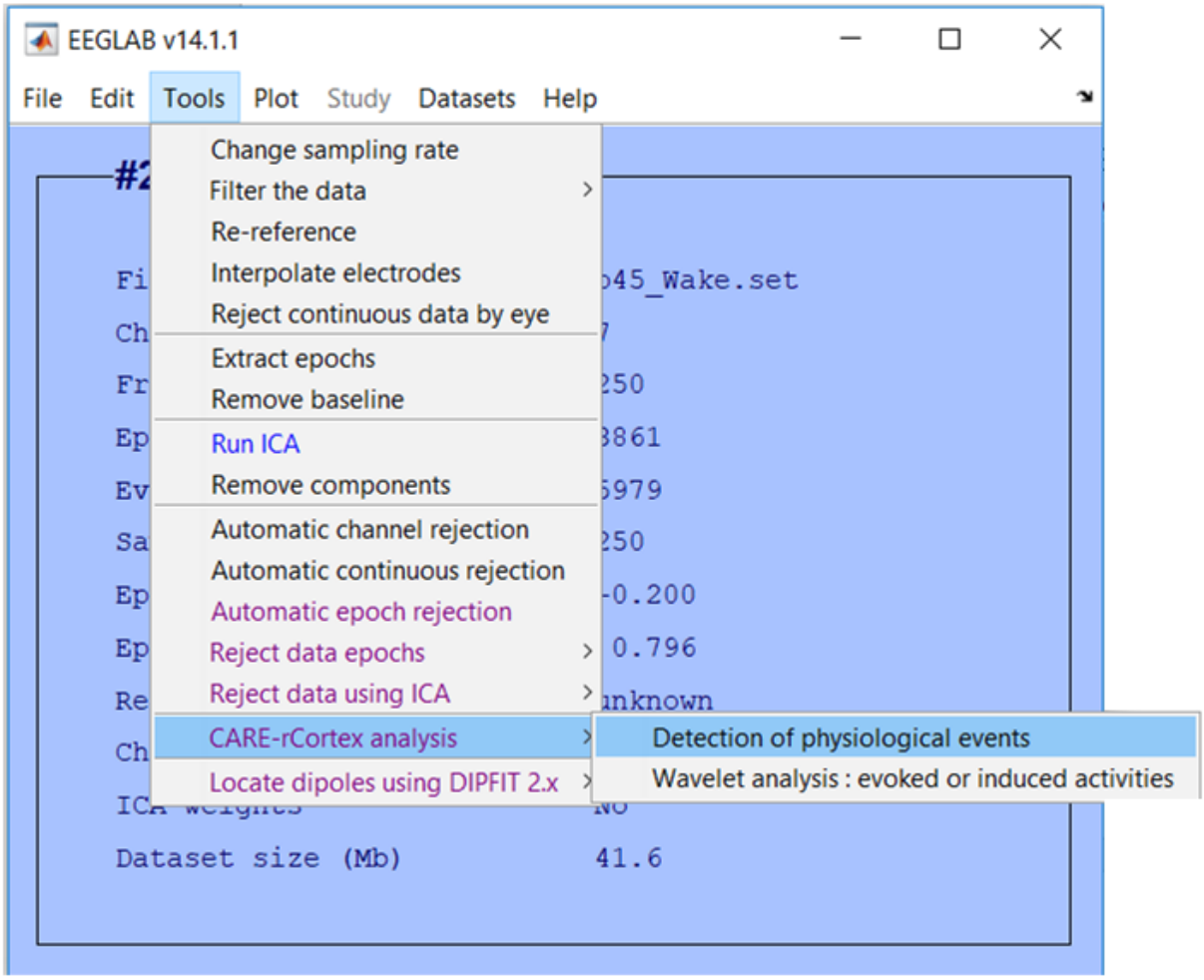
You can also **preprocess your data** with the tools provided by EEGLAB.

For these previous steps, please see the EEGLAB documentation.

Event detection

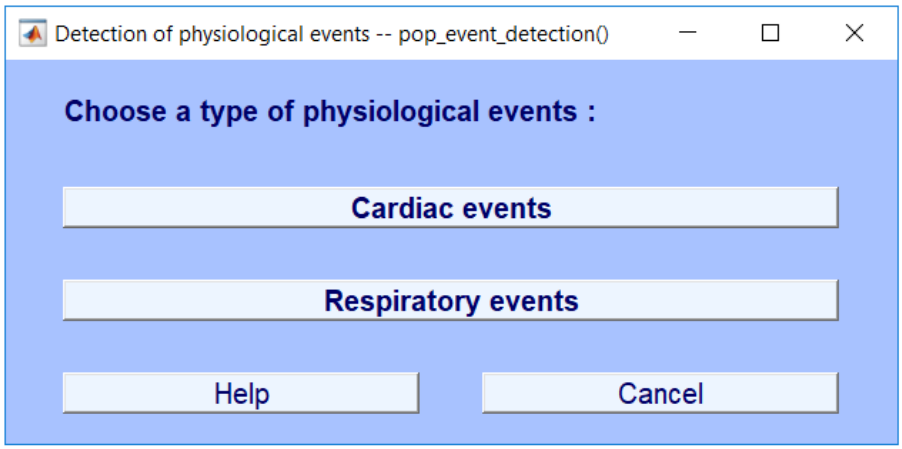
Now your data are prepared, you can use CARE-rCortex to **detect physiological events** with Tools > CARE-rCortex analysis > Detection of physiological events.

[NOTE: if you already have your physiological events, you have to skip this part and just extract epochs by doing Tools > Extract epochs. ***Warning***: you should not remove baseline at the end of the epoch extraction step, as it is provided later by the plugin. Then, you can go to **Time-frequency analysis** section.]



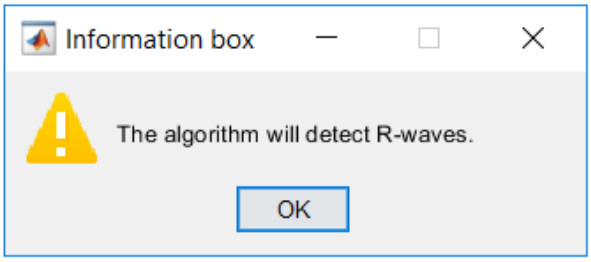
This selection called *pop_event_detection.m* function in order to allow you to choose one of the two type of physiological events:

- Cardiac events
  Or
- Respiratory events.

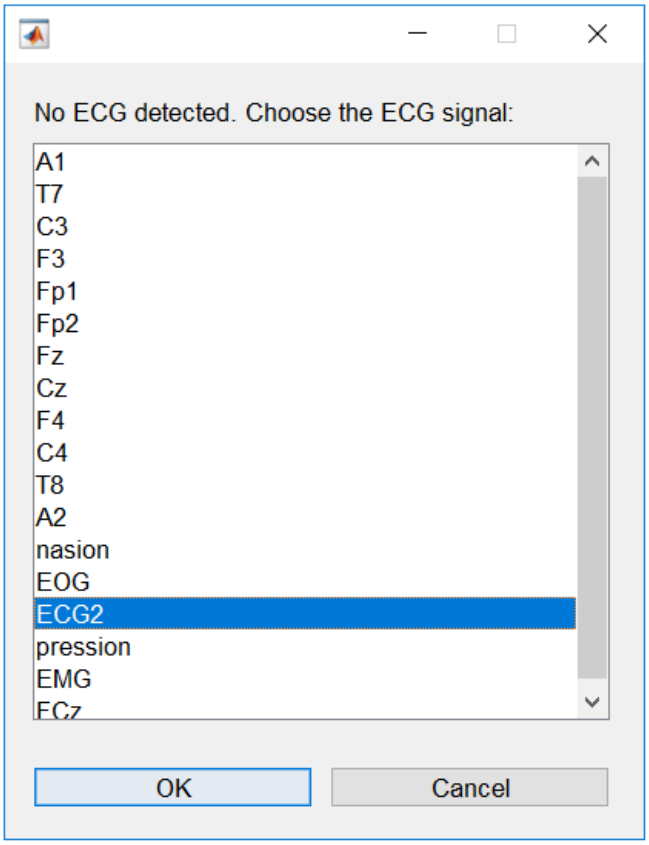


1. *Cardiac events*

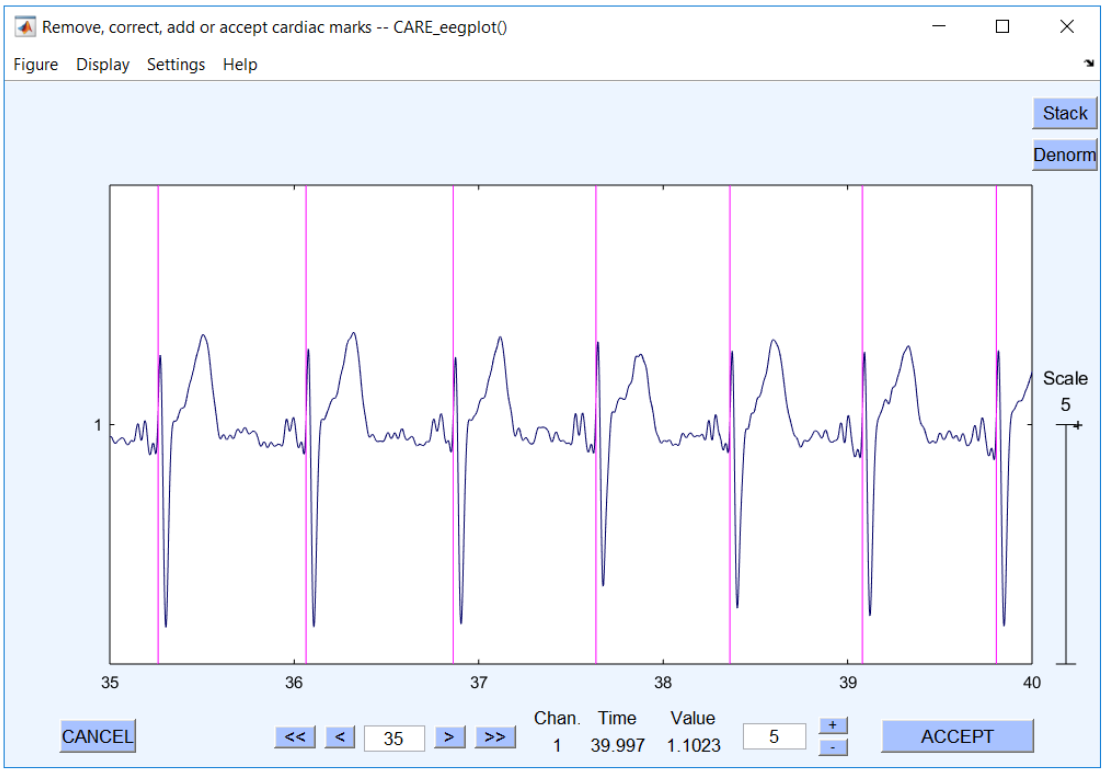
If you choose to detect cardiac events, a warning message appears in order to precise to you that CARE-rCortex will detect R-waves.



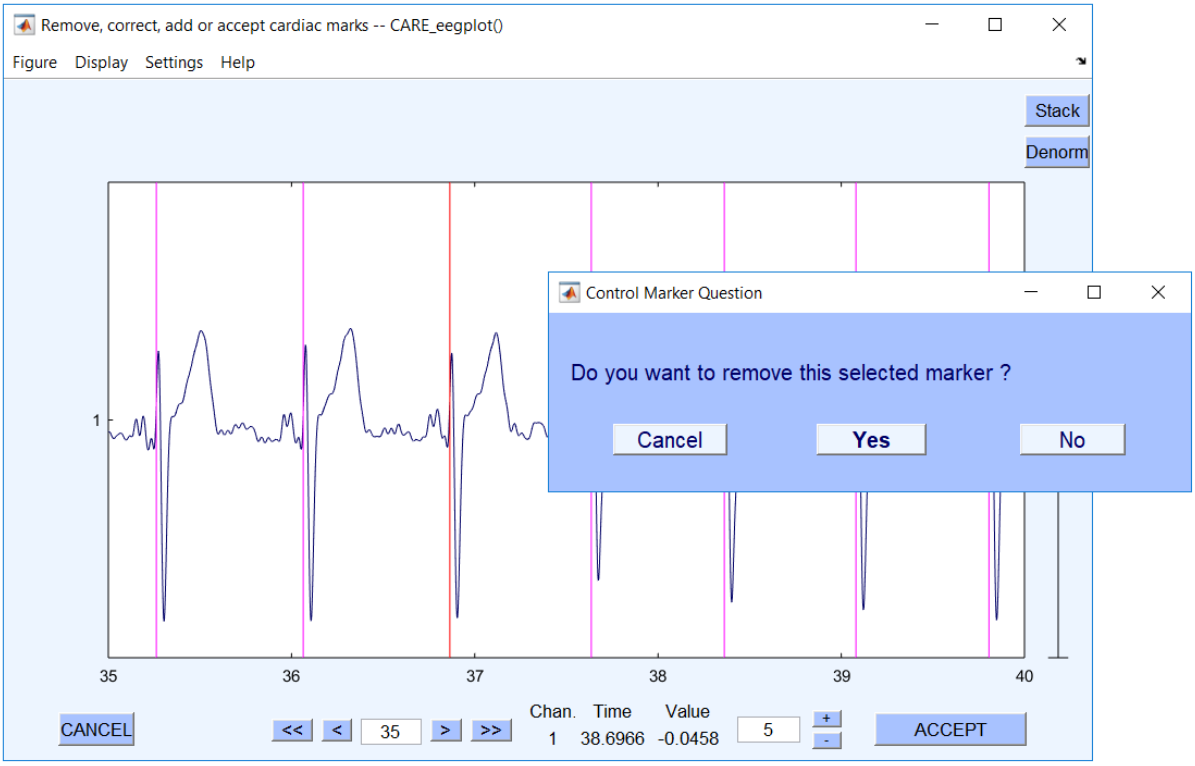
CARE-rCortex will try to find a channel which name can be one of the following: “CARDIAC”, “Cardiac”, “cardiac”, “COEUR”, “Coeur”, “Coeur”, “ECG”, “Ecg”, “ecg”, “CARDIAQUE”, “Cardiaque”, “cardiaque”. If none of the signals has one of these previous labels, a pop-up appears to let you choose the cardiac signal.



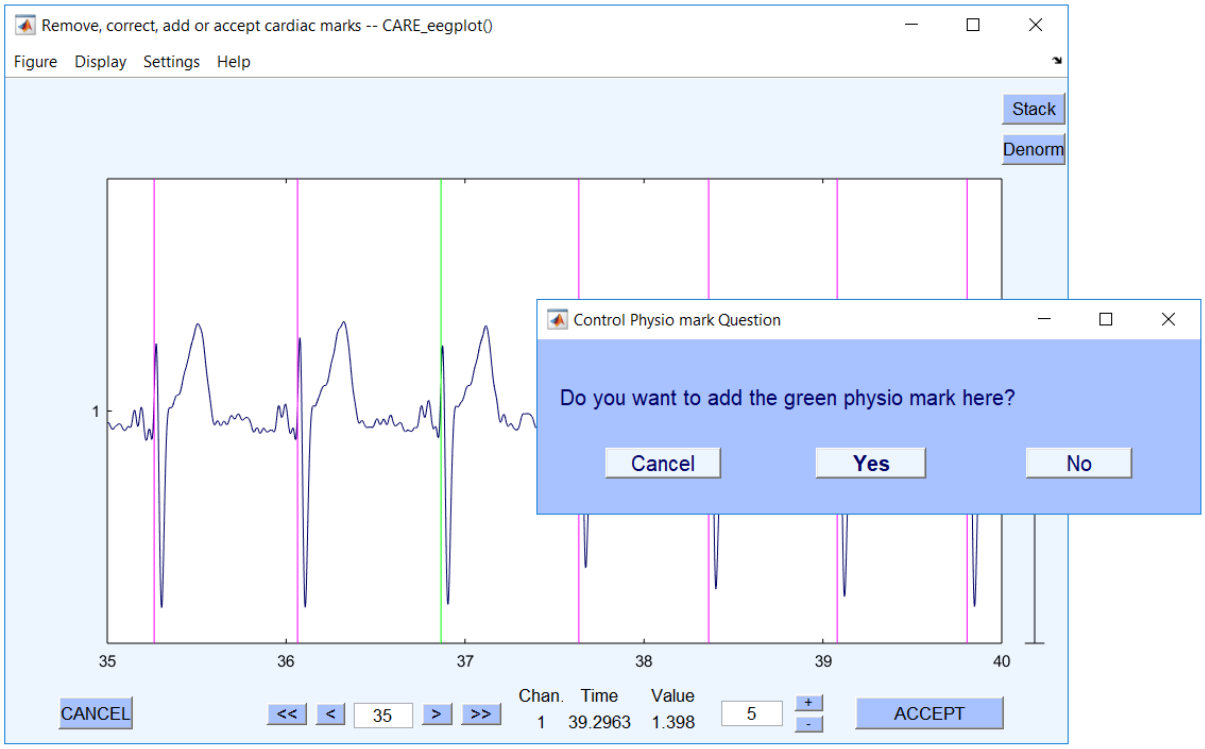
After cardiac signal selection, R-waves are detected by the *pan_tompkin.m* function written by Hooman Sedghamiz. *CARE_eegplot.m* function is automatically called by the plugin after the cardiac event detection, in order to display the R-wave detection.



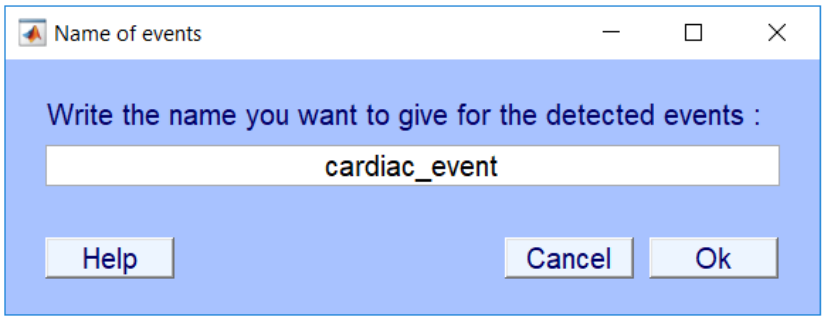
This function is the original *eegplot.m* function, adapted to CARE-rCortex to allow you to modify cardiac events. If you want to remove a cardiac event, you just click on the corresponding cardiac event materialized by a pink line. When a cardiac event is selected to be removed, the pink line becomes red and a pop-up message ask you if you really want to remove this event.



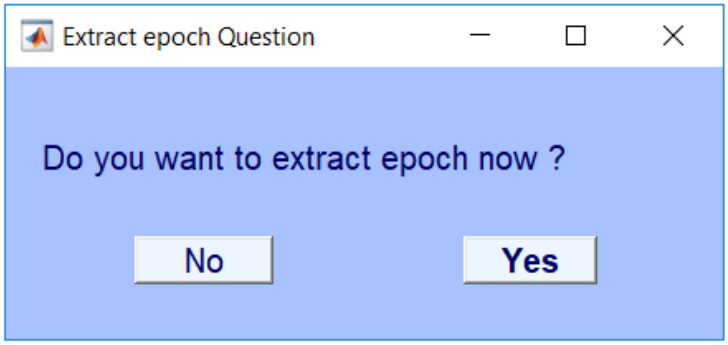
If you want to add a cardiac event, you can click on wherever you want on the display except on the lines which indicate the detected cardiac events. The location where you click to add a cardiac event is displayed by a vertical green line. A pop-up message appears to confirm that you want to add this cardiac event.



After you have checked the cardiac events, you have to press the “ACCEPT” button. Then, a pop-up window appears to ask you how you want to name the detected events. By default, the name is “cardiac_ event”. (The default name for respiratory events will be “resp_event”).

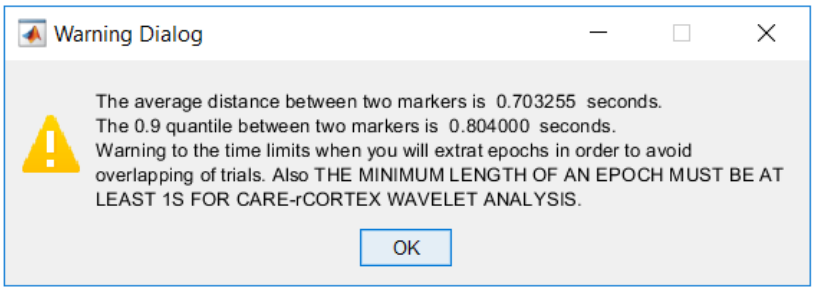


After pressing “Ok”, the EEGLAB structure is modified to save the time position of the new events in EEG.event.latency fields with the name saved in EEG.event.type fields. A new pop-up appears asking to extract the epochs now, based on the previous detected events.

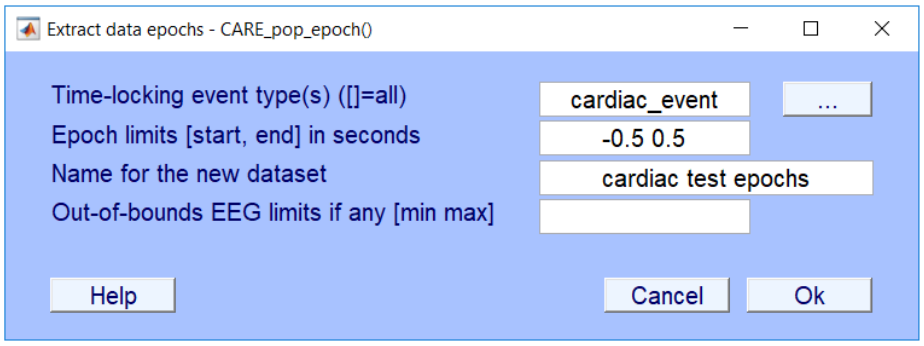


By pressing "Yes", you are going to extract the epoch now:

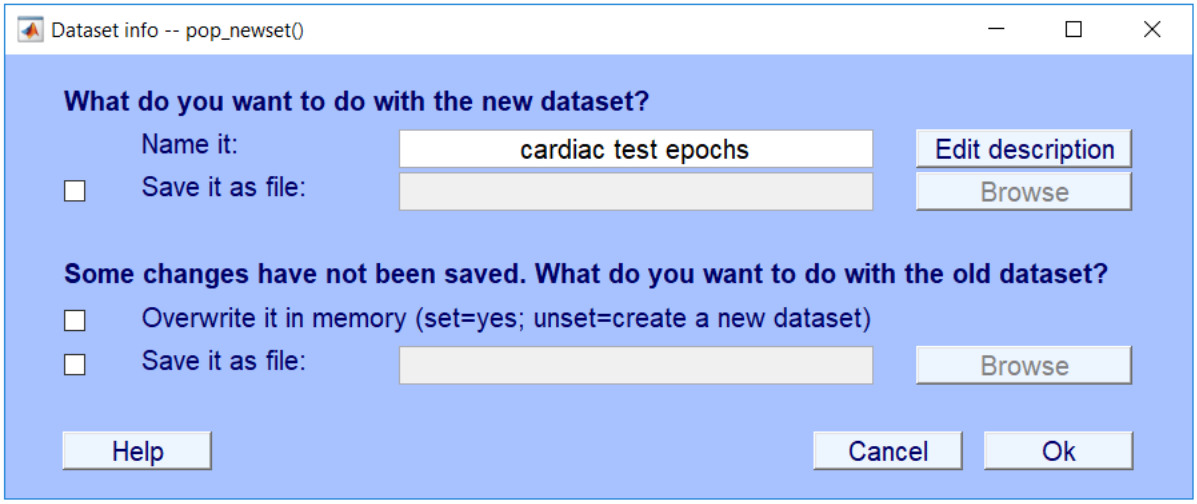
A warning dialog appears to let you know the average distance and the 0.9 quantile between two consecutive detected events. This information allows you to choose the epoch bounds avoiding overlapping of epochs. Also, a minimal time length is required for time-frequency analysis. This time length is specified in this warning dialog box and it depends on the sampling rate of your data.



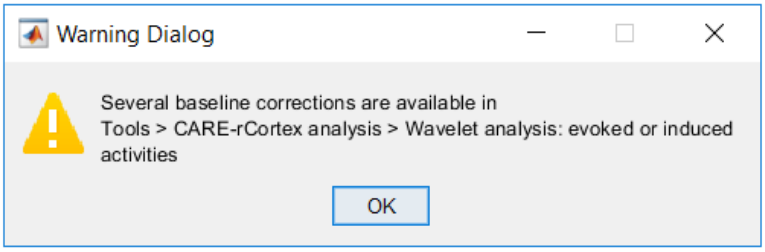
Then, *CARE_pop_epoch.m* function is called to extract the epochs, time-locked to the previous detected events. This function is similar to the original EEGLAB *pop_epoch.m* function but is slightly modified in order to propose by default the previous type of events and specified epoch limits. Indeed, instead of having "-1 2" as default epoch limits, we define the default lower bound equal to -0.5. The upper bound is defined as the averaged time length between two consecutive events minus 0.5 if this average is not lower than 1s. Else, the upper bound is set to 0.5.



After defining the type of the event, the epoch limits and the name of the new dataset, you can press "Ok" and the standard *Dataset info -- pop_newset()* figure appears to specify what you want to do with this new dataset.



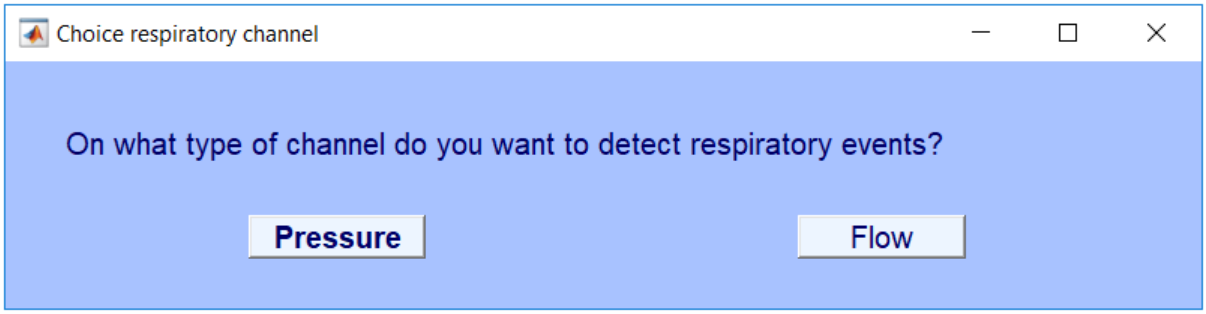
Contrary to the standard epoch extraction of EEGLAB, no baseline removal is proposed to you. Indeed, this is further provided by CARE-rCortex plugin. To avert you about this, a warning dialog box appears.



<u>If you press "No" at the pop-up box asking if you want to extract epochs now:</u>
You will be able to extract the epochs later by doing Tools > Extract epochs. However, pay attention to do not remove baseline after the epoch extraction as it is provided further by CARE-rCortex.

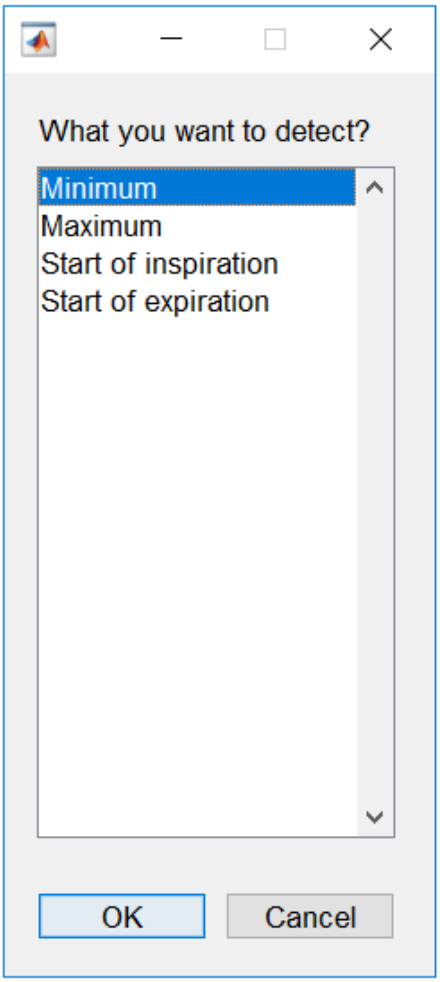
2. Respiratory events

If you choose to detect respiratory events in the *Detection of physiological events – pop_event_detection()* pop-up, a new pop-up appears to let you choose on which respiratory signal you want to detect the respiratory events. The choices are respiratory pressure signal and respiratory flow signal.



If you press “Pressure” button, CARE-rCortex will try to find a channel which name can be one of the following: “Pression”, “pression”, “PRESSION”, “Pressure”, “pressure”, “PRESSURE”. As for cardiac events, if none of the signals has one of these previous labels, a pop-up appears to let you choose the respiratory pressure signal by yourself. By the same way, if you press “Flow” button, CARE-rCortex will try to find a channel which name can be one of the following: “Debit”, “debit”, “D?bit”, “d?bit”, “DEBIT”, “Flow”, “flow”, “FLOW”, “Débit”, “débit”, “dÃ©bit'”, “DÃ©bit”. However, if none of the signals has one of these previous labels, a pop-up appears also to let you choose the respiratory flow signal.

After selecting the respiratory signal, a new pop-up window appears to let you choose the respiratory event you want to detect. There are 4 possibilities: the minimum or the maximum of each respiratory cycle, the start of the inspiration or the start of the expiration.

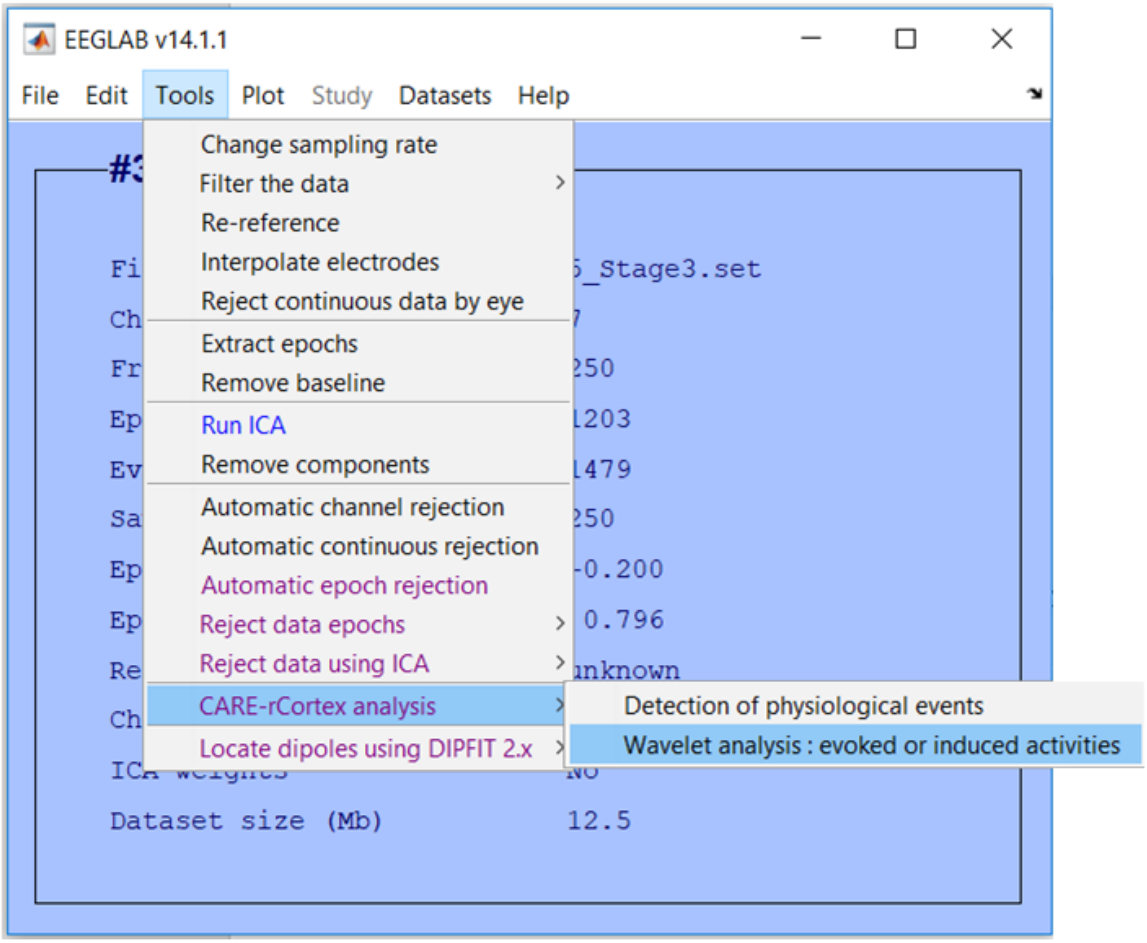


Whatever you choose to detect, *find_resp_marks.m* function is called to detect the chosen event. Then, as for cardiac events, *CARE_eegplot.m* function is automatically called by the plugin to display the detected events. At this step, all the steps are the same as for cardiac event detection, that is: add, remove or modify the time location of the detected events and extract epochs related to these events. To see more details, refer to **Cardiac events** section.

Time-frequency analysis

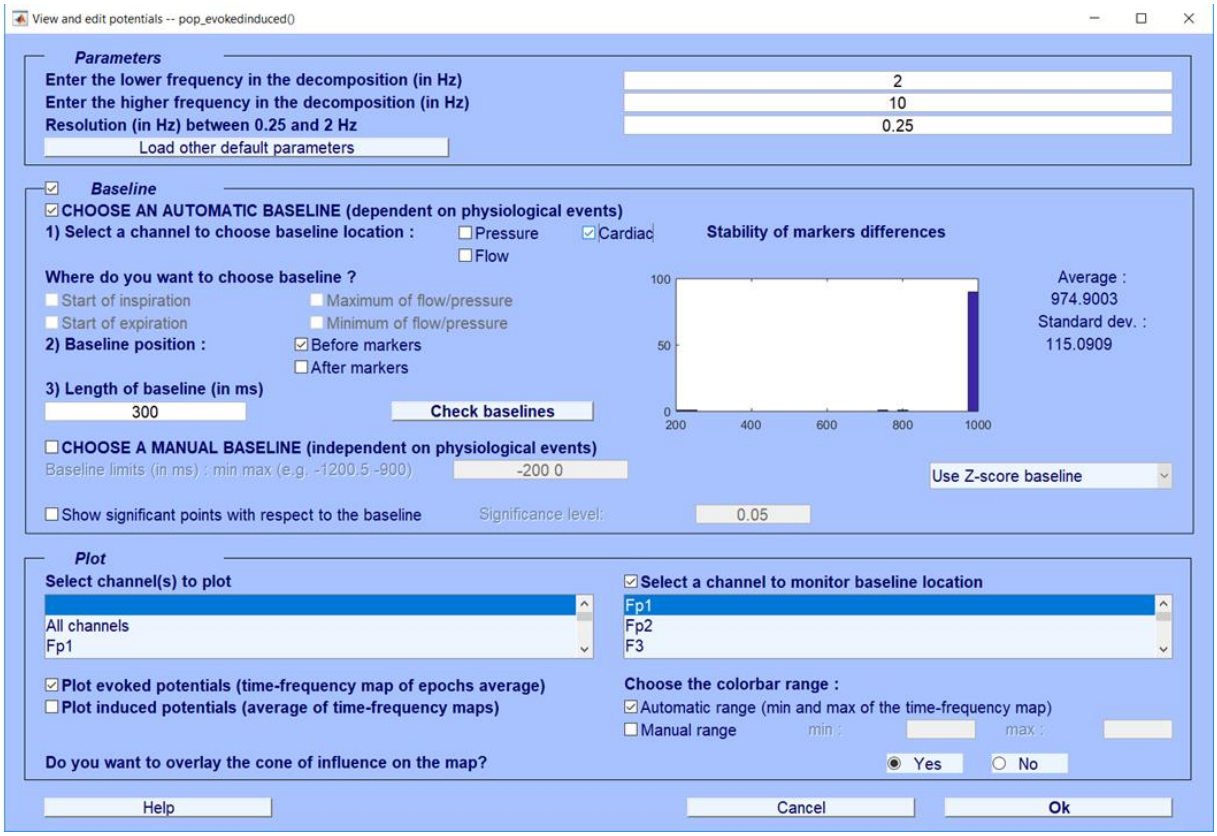
To continue, you need to save the data in EEGLAB format: File > Save current dataset as …

Now, you are ready to analyze your data in time-frequency domain with CARE-rCortex plugin. To do that, you have to select Tools > CARE-rCortex analysis > Wavelet analysis: evoked or induced activities.



This selection calls *pop_evokedinduced.m* function which generates a new figure composed of three panels:

- Parameters
- Baseline
- Plot



We will describe them in the following lines.

1. *Parameters panel*

Here you can define the frequency range of interest for your time-frequency analysis.

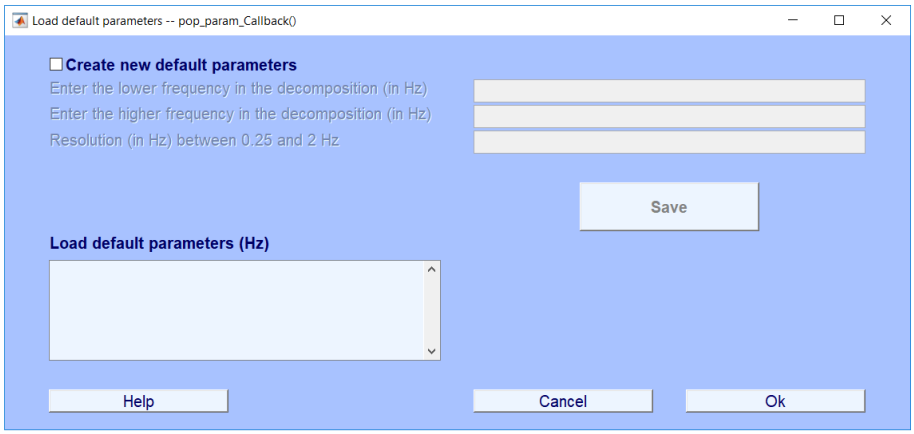
First, you can choose the **lower frequency** of the time-frequency analysis. We limited this value to 2Hz (default). If you enter a lower frequency, CARE-rCortex will start the analysis at 2Hz.

Then, the **higher range of the time-frequency analysis** can be specified. This value has to be higher than the previous frequency value but lower than the half of the sampling frequency (Nyquist frequency). The default is 30Hz.

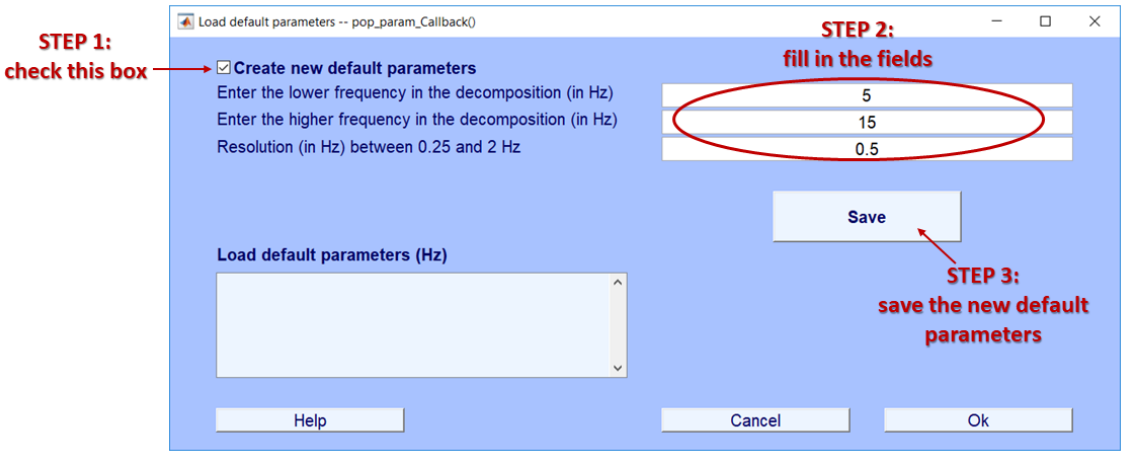
Finally, the **frequency bin resolution** can be chosen. This value has to be fixed between 0.25Hz and 2Hz. If it is lower than 0.25Hz, CARE-rCortex will compute the time-frequency analysis with a 0.25Hz frequency resolution. By default, this value is 0.25Hz.

The values you entered will be saved in a .txt file named *latest_parameters.txt*. The next time you will use CARE-rCortex analysis, these values will be automatically set in the interface.

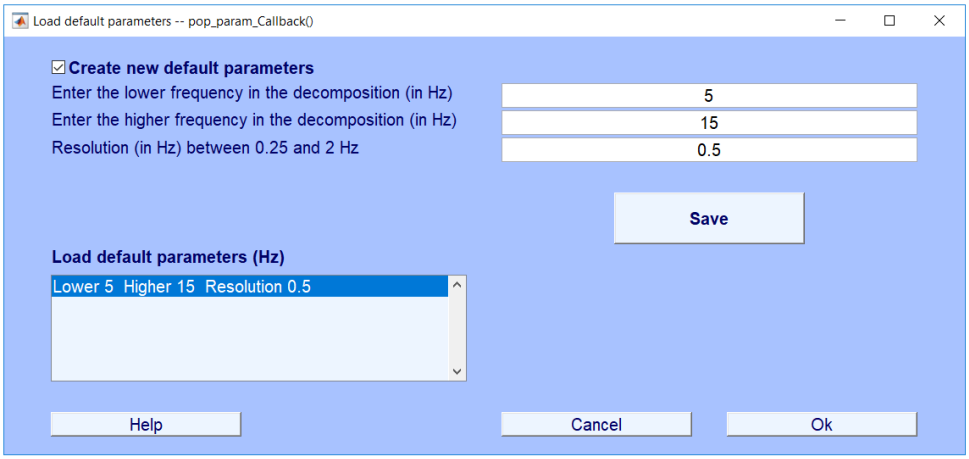
If you wish to create some default values for your analysis and your next analysis, you can push “Load other default parameters” button. This calls the *pop_param_Callback.m* function and a pop-up appears.



In this pop-up, you can create new default parameters by choosing the corresponding checkbox. Just type the lower and the higher frequency of the time-frequency analysis and the frequency bin resolution considering the previously described limits. Then, push the “Save” button the make the changes effective.



By pushing the “Save” button, your new default parameters are added in a .txt file entitled *default_parameters.txt*. These new parameters appear in the list with the caption “Load default parameters (Hz)”.



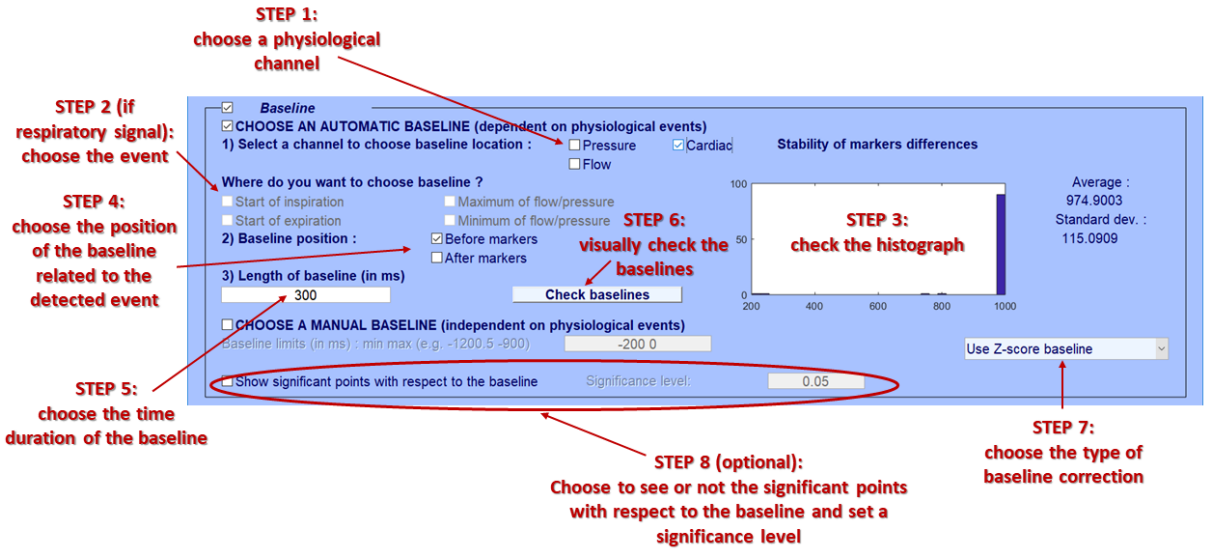
Then, you can choose one of the default parameters visible in the list (here there is only one set), and press “Ok”.

2. Baseline panel

In this part, you will define the baselines to normalize your data, i.e. to compare the influence of the physiological event in the EEG with respect to a reference period. You can choose either an automatic baseline or a manual baseline.

Choose an automatic baseline:

If you choose an automatic baseline, the epochs will be normalized by a baseline detected automatically that will be dependent on the physiological events of choice. This can be performed following these steps:



STEP 1: **Choose a physiological channel**. Like for the event detection, you can choose to define the baselines on respiratory pressure or flow events or cardiac events. “Cardiac” checkbox will be available if one of the channel has one of these following names: “CARDIAC”, “Cardiac”, “cardiac”, “COEUR”, “Coeur”, “Coeur”, “ECG”, “Ecg”, “ecg”, “CARDIAQUE”, “Cardiaque”, “cardiaque”. If this is the case, and you put a tick in this checkbox, a warning dialog box appears to inform you that CARE-rCortex will detect R-waves.

“Pressure” checkbox will be available if one of the channel has one of these following names: “Pression”, “pression”, “PRESSION”, “Pressure”, “pressure”, “PRESSURE”.

“Flow” checkbox will be available if one of the channel has one of these following names: “Debit”, “debit”, “D?bit”, “d?bit”, “DEBIT”, “Flow”, “flow”, “FLOW”, “Débit”, “débit”, “dÃ©bit'”, “DÃ©bit”.

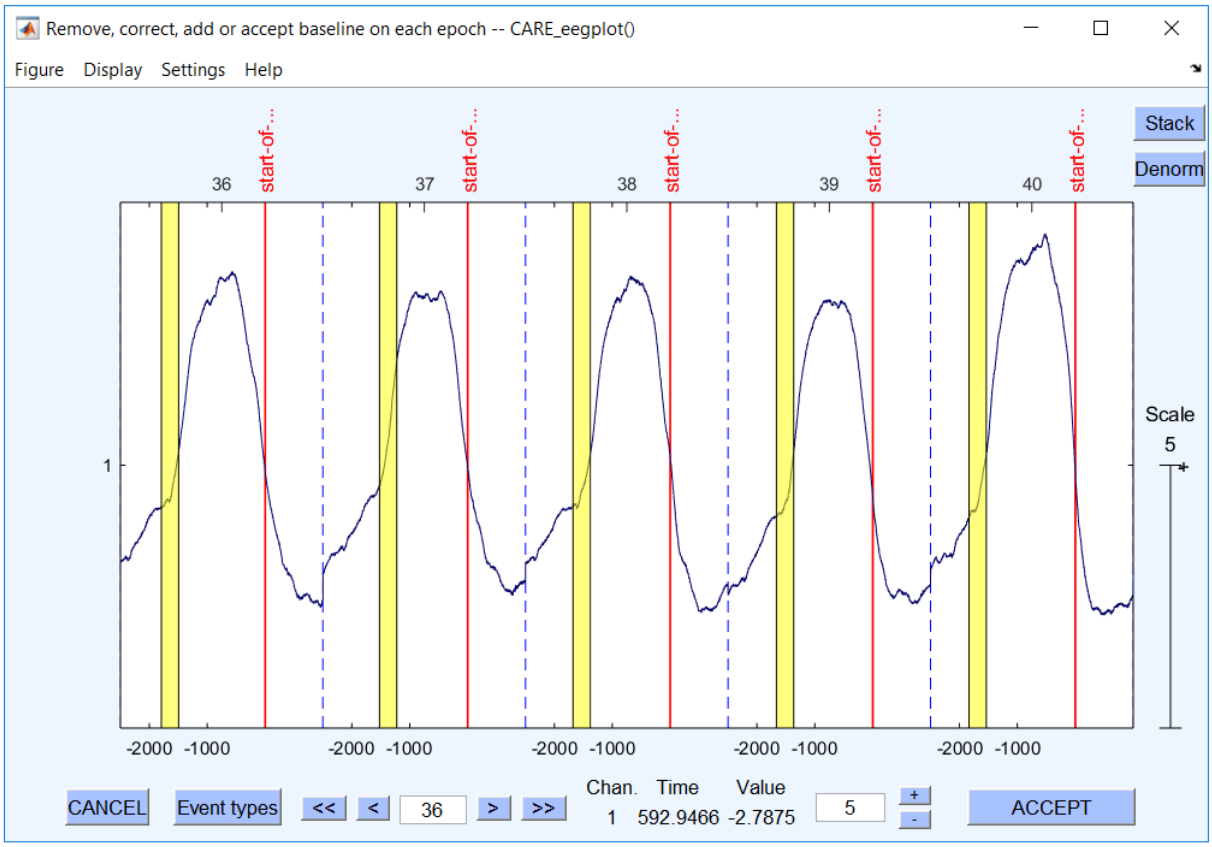
STEP 2 (if respiratory signal): **Choose the type of respiratory event** from which you want to define the baselines (minimum or maximum of each respiratory cycle or the inspiration/expiration start). These respiratory events are detected by *find_resp_marks.m* function.

STEP 3 (if respiratory signal): **Check the histogram** of time intervals between two consecutive detected events. On the right of the graph, the average *A* and the standard deviation *S* of these values are displayed. The goal of this tool is to help assessing the stability of the chosen event detection. As far as *A* is closer to the epoch length and *S* is closer to zero, the event detection becomes more stable. In this sense, we recommend to use the most stable respiratory event. This histogram is also presented for informative purpose when you decide to detect R-wave on a cardiac signal.

STEP 4: **Place your baseline before or after** the detected events.

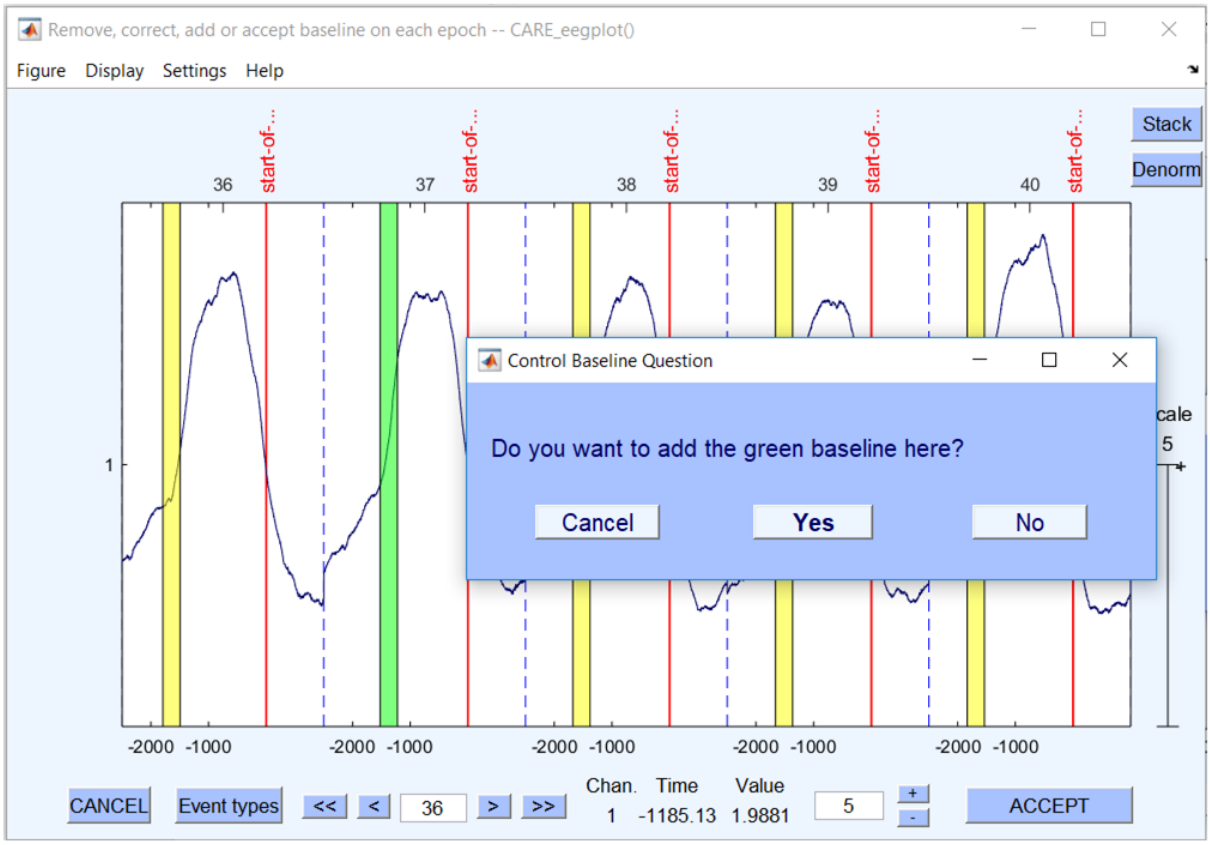
STEP 5: **Define the length of the baselines** in milliseconds. This length will be saved in a .txt file named *latest_automatic_baseline.txt* which will be read by CARE-rCortex in forthcoming analyses.

STEP 6: **Visually check** baselines by pushing the "Check baselines" button. This calls the *CARE_eegplot.m* function to show the baselines as you defined on the physiological signal of choice.



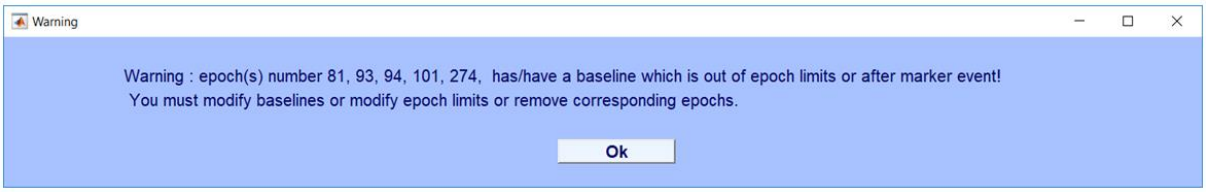
In this window, you can visually check the position of the baseline. If you want to remove a baseline, you just click on the corresponding baseline materialized by a yellow band. When a baseline is selected to be removed, the yellow band becomes orange and a pop-up message ask you if you really want to remove this baseline.

If you want to add a baseline, you can click on wherever you want on the display except on the already present baselines. The location where you click to add a baseline is displayed by a green band. A pop-up message appears to confirm that you want to add this baseline.

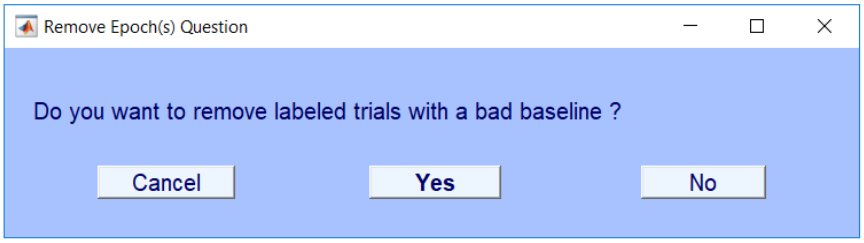


After visually checking the baselines, you can push the “ACCEPT” button. CARE-rCortex now will check if each baseline is located before the event which defines the 0 of the epoch (rule 1) and if each epoch has one and only one baseline (rule 2).

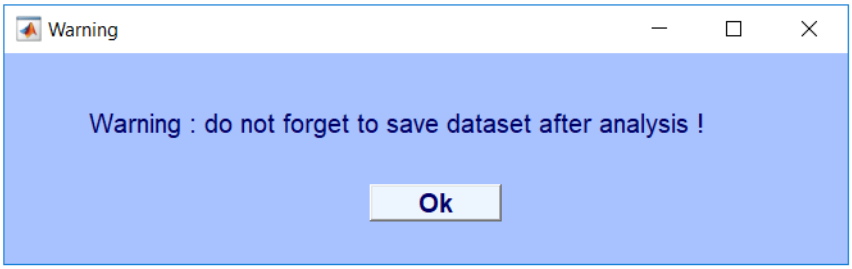
First, the rule 1 is checked. If it is not respected, a pop-up appears to specify to you what are the epochs that do not respect the rule and what are the options you have to fix this problem:



Then, another pop-up window asks to remove the epochs that do not respect the rule:

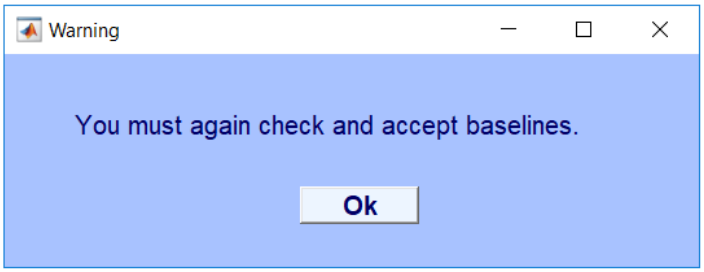


If you press “Yes”, the corresponding epochs will be deleted from the EEGLAB structure and a pop-up appears to remind you to save again the dataset.



Then, after removing the epochs that do not satisfy the first rule, a pop-up lets you know that the baselines need to be checked again. You need to check and accept again the baseline locations and the second rule is checked by CARE-rCortex.

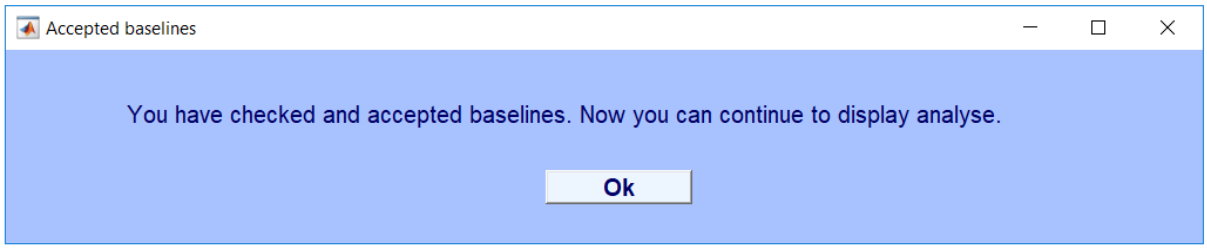
[NOTE: If the first rule is satisfied, the second rule is directly checked by CARE-rCortex without need to check and accept again the baseline locations.]



If the second rule is not respected, the same type of pop-up windows will appear to let you know the epochs that do not respect the second rule and the possibilities you have to fix the

issues. The same pop-up than for the first rule appears to propose you to remove or not the corresponding epochs.

After these two rules are checked, another pop-up window appears to let you know that the baseline locations are checked and you can now continue to parametrize CARE-rCortex to analyze your data.



[NOTE: This pop-up directly appears when both rules are satisfied.]

STEP 7: **Choose the type of normalization** you want to apply on your data. There are four possible baseline corrections:

- *Z-score baseline correction*: the average value (M_f) and the standard deviation value (STD_f) of the map in the baseline part is computed at each frequency bin *f*. Then, for each pixel of the time-frequency map, at each frequency bin *f*, CARE-rCortex subtracts the power values by M_f. The resulting values are divided by the corresponding STD_f. The unit of the time-frequency values is z-score or standard deviation of the baseline.
- *Absolute baseline correction* which is a close variant of the *z-score baseline correction*. If you choose this baseline correction, CARE-rCortex will only remove *Mf* from each wavelet power spectrum value at each frequency bin. The unit corresponds to the squared unit of the EEG power divided by Hertz.
- *Relative baseline correction*: in this normalization, instead of subtracting *Mf* from each power value at each frequency bin, we dividing by *Mf*. In this type of time-frequency map, the values are expressed in percentage.
- *dB baseline correction*: this latest approach is the log-transformed of the previous baseline correction and it is expressed in decibel (dB).

STEP 8 (optional): **Find significant parts** of the time-frequency maps with respect to the baseline. **Put a tick in the corresponding checkbox** and **choose a significance level (σ)**. By default, σ is set to 0.05.

To assess these significant parts, we use baseline values permutation to build a surrogate distribution of the mean values of the time-frequency maps at each frequency bin as it is proposed by R. Grandchamps and A. Delorme, “Single-trial normalization for event-related spectral decomposition reduces sensitivity to noisy trials”, *Frontiers in Psychology*, 2011. The way this surrogate distribution is build differs if evoked time-frequency or induced time-frequency maps are computed.

Note: if there are more than 100 epochs, we selected around 100 uniformly spaced epochs. This choice can be modified at the line 98 in *timefrequencyevoked.m* and

timefrequencyinduced.m functions, at the line 151 in *timefrequencyevokedBase.m* function, at the line 150 in *timefrequencyinducedBase.m* function ("`nbEpoch = 100;`").

- *For evoked time-frequency map*: First, we build *C* combinations each randomly composed of half of the selected epochs (E_i). Then, at each combination *i*, we average all the epochs in E_i and a time-frequency decomposition of this average epoch is computed. By this way, we obtain *C* time-frequency maps which are normalized with a dB baseline B_i. The baselines are chosen in a way that after the *C* combinations, all the data points are chosen as the baseline. If the baseline length contains at least data points, we provided an overlap of 75% between the baselines of two consecutive combinations. Otherwise, no overlap is allowed between the baselines of two consecutive combinations. If you want to change this rule, feel free to modify the lines 260 and 267 of *timefrequencyevokedBase.m* function. By this procedure, we are able to create one surrogate distribution of the mean values of the time-frequency maps at each frequency bin.

- *For induced time-frequency map*: we permute baselines values across time and epochs at each frequency bin on the time-frequency map (*rawTF*) which is not normalized by the baseline you have chosen. The number of permutation is limited at 200 for faster computation. However, if you want to change the number of permutations, you can change it at the line 149 in *timefrequencyinducedBase.m* function ("`R = 200;`"). For each permutation, a dB baseline correction is applied with the random baseline, on one of the selected epochs of *rawTF*, which is randomly selected by permutation. This procedure allows to create a surrogate distribution of the mean value of each time-frequency map at each frequency bin.

In parallel, the time-frequency map (*rawTF*) which is not normalized by the chosen baseline is normalized with the not-random dB baseline to obtain *normTF*. For each frequency bin, for each point of the time-frequency map, CARE-rCortex tests if *normTF* lied in the $\frac{\alpha}{2} * 100$ % or $1 - \frac{\alpha}{2} * 100$ % tail of the surrogate distribution at this given frequency. If it is the case, the time-frequency point is considered significant at $p < \sigma$. The parameter α is the Bonferroni corrected level σ, that takes into consideration the number of time points within an epoch. The obtained mask is displayed on the time-frequency map with a transparent grey color.

Choose a manual baseline:

If you choose a manual baseline, the epochs will be normalized by a baseline independent on physiological events. In this case, you just need to set the **time limits** (in milliseconds) of the baselines which will be the same for all the epochs. These time limits have to be between EEG.xmin and EEG.xmax (see the EEGLAB structure). These limits will be saved in a .txt file named *latest_manual_baseline.txt* which will be read by CARE-rCortex during next analysis. As for the automatic baseline, choosing the type of the baseline correction is up to the user. You have the choice between 4 baseline corrections: z-score baseline correction, absolute or relative baseline correction or dB baseline correction. See **STEP 7** of the "***Choose an automatic baseline***" part for more details.

In order to observe the **significant parts** of the time-frequency maps with respect to the manual baseline, **put a tick in the corresponding checkbox** and **choose a significance level**. By default, this significance level σ is set to 0.05. The way CARE-rCortex assesses the significance of the points in the time-frequency map is described previously in the **STEP 8** of the "***Choose an automatic baseline***" section.

3. Plot panel

Here, the channel(s) to be analyzed can be selected. This choice is done in the list of channels on the left. There is the possibility of **selecting one, several or "All channels"**.

In this section, there is the **choice between computing two time-frequency analysis**: evoked potentials or induced potentials. If you select "Plot evoked potentials (time-frequency map of epochs average)", CARE-rCortex will compute the average of epochs and then compute the corresponding time-frequency map. This type of analysis is relevant when the EEG potentials are perfectly time-locked with the event. If you select "Plot induced potentials (average of time-frequency maps)", CARE-rCortex will compute one time-frequency decomposition for each epoch and then will average the time-frequency maps (for further details about the differences about evoked/induced activity, refer to the article: C. Tallon-Baudry et al. "Induced γ-band activity during the delay of a visual short-term memory task in humans." *Journal of Neuroscience*, 1998).

The time-frequency decomposition is done by *waveletTransformFourierPAD.m* function which applies Morlet wavelet decomposition with 5 cycles. We chose Morlet wavelet with 5 cycles as it is convenient for most of EEG analysis. If you wish to change the number of cycles or the type of the wavelet decomposition, feel free to modify *waveletTransformFourierPAD.m*.
***Warning*: for faster computations, CARE-rCortex only keeps around 100 uniformly spaced epochs to perform the time-frequency analysis.**

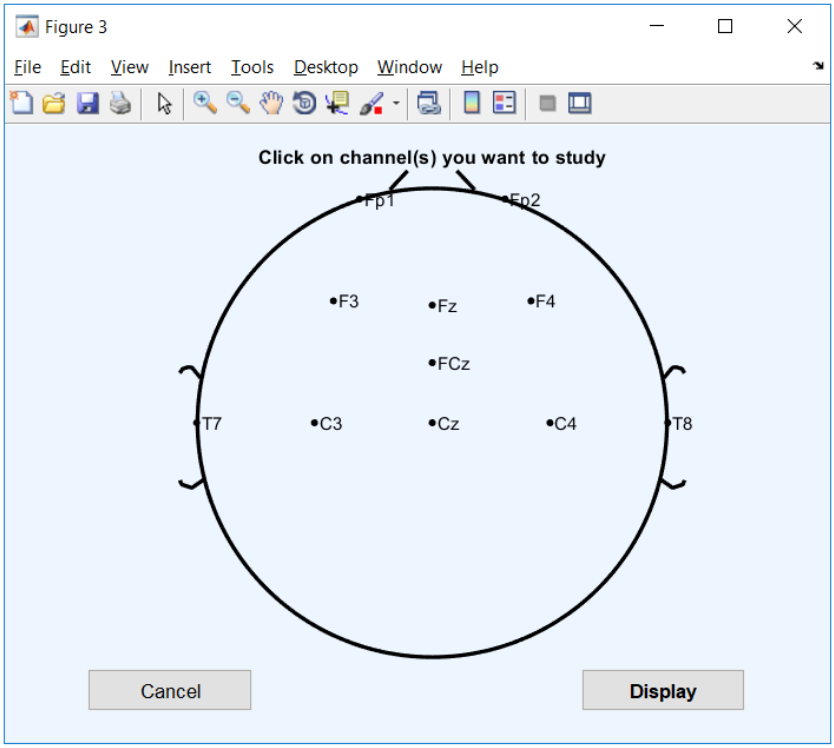
On these time-frequency maps, you can apply a "**cone of influence**" by choosing "Yes" at the corresponding question in the "Plot" panel. This cone of influence represents the zone of the time-frequency map where zero-padding data were introduced in order to compute the time-frequency decomposition. Hence, this zone should be interpreted with caution. The cone of influence will be superimposed on the time-frequency map in transparent grey color.

You can also choose the **colorbar range** of your time-frequency map. Either you let the automatic range or you choose the bounds of the colorbar. If you choose the automatic range, the lower bound will be the minimum through all the values in the time-frequency map, and the upper bound will be the maximum through all the values in the time-frequency map.

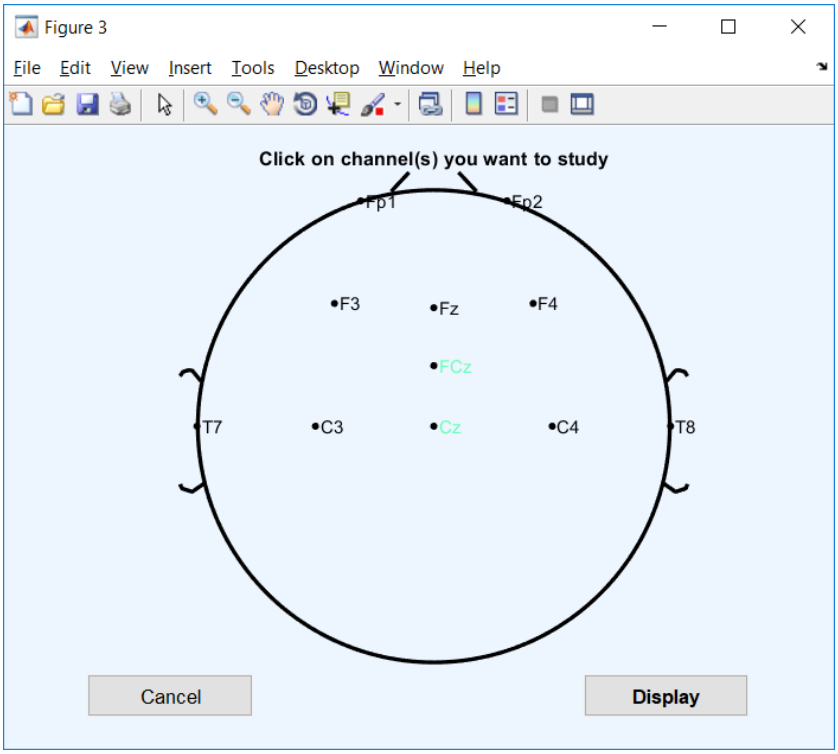
We provide the possibility to see the median baselines on the average of the same considered epochs than for the time-frequency decomposition of another channel, in particular physiological channel. To do that, **put a tick in the checkbox entitled "Select a channel to monitor baseline location"** and **choose one channel**.

Results of the analysis

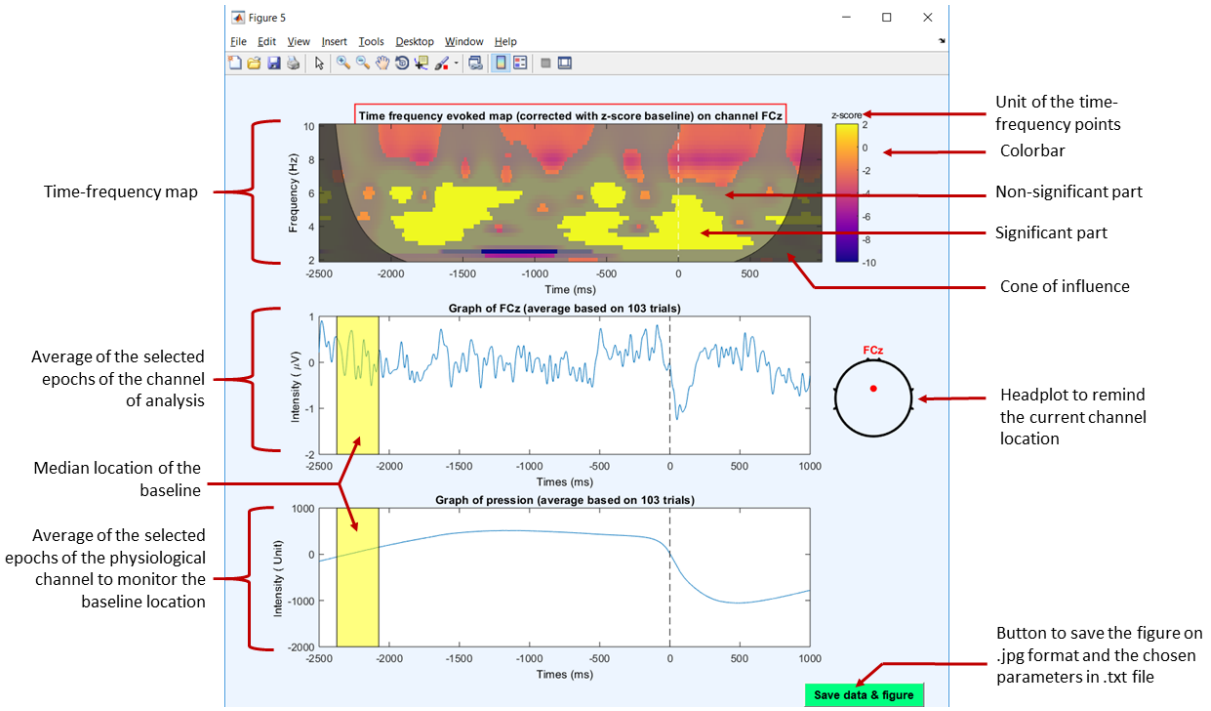
Now that the analysis is properly parametrized, you can push the “Ok” button.
If you have chosen several channel(s), *evokedinducedtopoplot.m* function is called and a head plot appears when channel locations are available, showing the previously selected channels in the CARE-rCortex interface.



Here you can select the channels to be analyzed. As shown in the figure below, the selected channel(s) (Cz and FCz in this case) are highlighted in green.



Once the channel(s) selection is done, results of the analysis will be displayed in the following figure after pushing the “Display” button.



The above figure is displayed for each selected channel(s). Here you will see the time-frequency map of the EEG activities of the corresponding channel. This map depends on the parameters entered by the user. As a reminder, the title of the time-frequency expresses these choices: evoked or induced time-frequency map, corrected or not with a baseline, z-score or absolute or relative or dB baseline correction, and the channel. If the EEGLAB structure contains the location of the current channel, a head plot is displayed with the channel location.

If you defined a significance level to display the significant zones of the time-frequency map with respect to the baseline location, the non-significant parts are displayed with a transparent grey color superimposed to the background values. If you have chosen to display the cone of influence, it will be superimposed as a supplementary layer in transparent gray.

The CARE-rCortex interface allows the inclusion of an additional channel to monitor the baseline location with the purpose of displaying the baseline on the average of the physiological signal. In case you select this additional a channel, it will be located at the bottom panel (see the figure above), representing its average across the epochs with the median baseline location highlighted in yellow. If the analysis concerns evoked time-frequency activity, the averaged EEG epochs, including the baseline highlighted in yellow, will be represented on the middle panel.

Finally, by pushing the “Save data & figure” button, the window depicting the results is saved in .jpg format. Moreover, all the chosen parameters during the analysis are written in a .txt file to reproduce easily the results. Of note, the EEGLAB structure is modified by the plugin, adding a “CARE-rCortex” field in the EEG structure. This field contains an array of time-

frequency map values (map and channels according to the user's choice) and the corresponding array of time-frequency mask if the analysis of significance was performed.

This tutorial was written by Fanny Grosselin.