# EEGpal: **Epoching module**

Version 2.0, 31.08.2025

Abbreviation:
[tf] = Time Frame (It corresponds to one sampling point the EEG files. This is the smallest unit of time. Its duration depends on the sampling frequency. For example, if the sampling frequency is 2048 Hz, then 1 TF=1/2048= 0.000488s = 0.488 ms.)
[ms] = duration in milliseconds

The module 'Epoching' has three purposes:

A. As the name suggests, this tool enables you to create epochs from a pre-processed EEG file. In other words, it enables the signal to be divided into several equal-length segments. The segment can originate from a trigger/marker or the entire file.

B. The module also allows artefact rejection. In the *'Artefact rejection'* tab, you can specify a rejection criterion, after which any epochs that do not respect it will be suppressed or ignored.

C. Finally, the module enables you to calculate the average of all clean epochs in order to create an event-related file (ERP), which is the most common type of EEG analysis. However, it also enables you to save all clean epochs (segments of your input signal) into a single file, which is typically employed for frequency analysis.
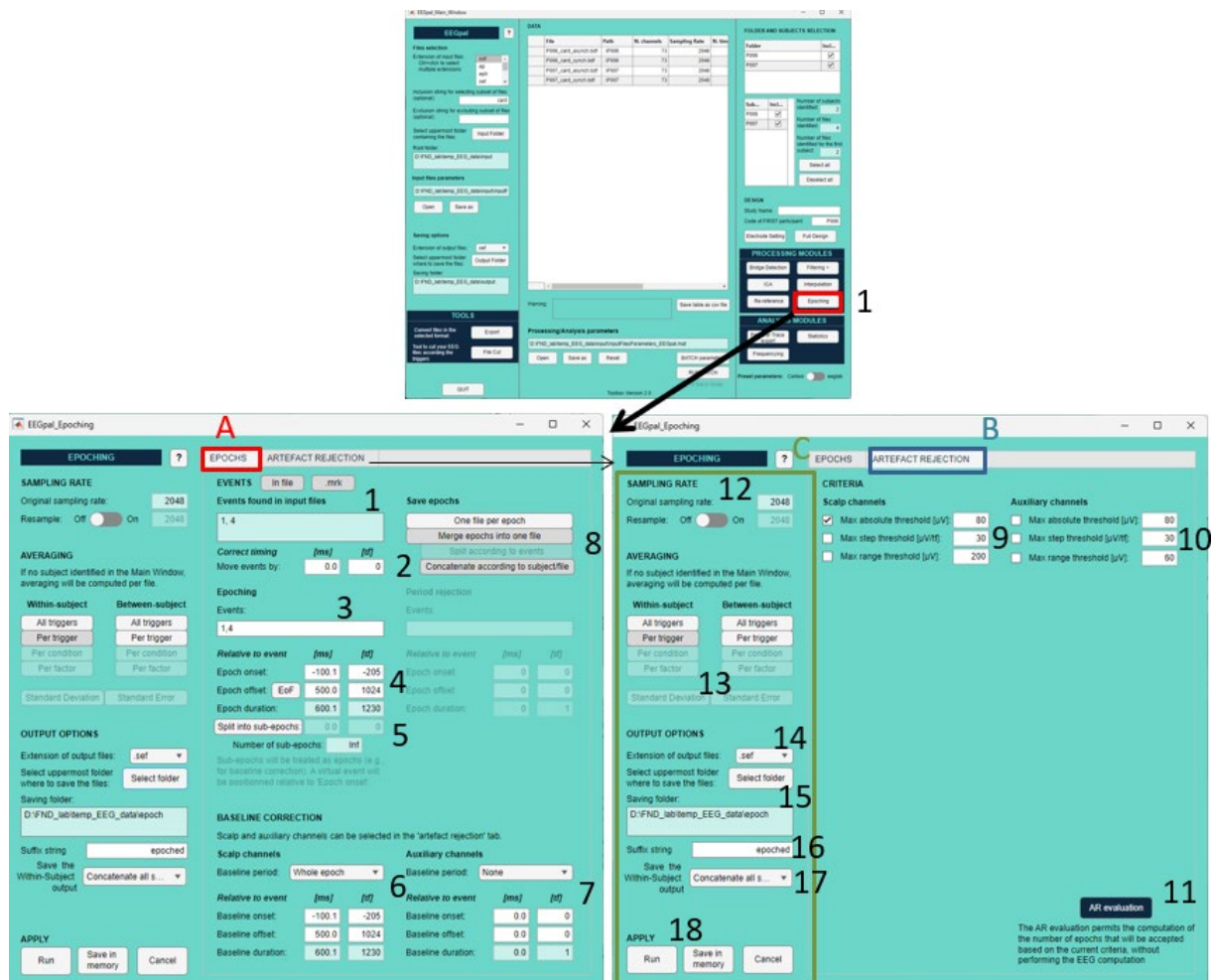
Pannel A: EPOCHS

1. In EEGPal, we distinguish between triggers (tags within the EEG file) and markers (tags recorded in a separate .mrk file). You can choose to work with either triggers (button *'In file'*), markers (button *'.mrk'*), or both. EEGPal will list all of these tags in the section '*Events found in input files'*.

2. If you wish, you have the option of adjusting the position of the event (e.g. to account for a systematic timing error between the events and the stimulation due to technical reasons such as display delays measured with a photodiode sensor). You can specify either a negative delay to move the event backwards or a positive delay to move it forwards. Enter the delay in either milliseconds or [tf].

3. Specify the events you wish to examine, separating them with commas or spaces. Values must be listed in the section *Events found in input files*.

4. Specify the duration of the epoch. Note that this time is relative to the event. If you want to include a period before the event, enter a negative value for the onset. In this example, the epoch will last 100 ms before and 500 ms after the event. The onset and offset can be entered in either milliseconds [ms] or time frame [tf]. For the offset, use the '*EoF'* button (=end of file) if you want your epochs to last until the end of the file.

5. The *Split into sub-epochs* button permits to create sub-epochs. This option is usually selected when you want to divide all your EEG files into epochs without considering events (this is typically used in frequency analysis, where the signal needs to be divided into small, equal parts). See the FAQ for more details.
Note: The field Number of '*Sub-epochs'* as well as the '*Epoch offset value'* when 'EoF' is enabled, provide values in relation to the first file in the list. However, if you have multiple files of different lengths, the values will be adjusted accordingly.

6. The **baseline correction** involves subtracting the mean amplitude of a specified period of the epoch (the baseline) from the entire epoch. See the FAQ for a more detailed explanation of how to use it. You can choose from several different baselines:
    a. None: No baseline correction applied
    b. Whole epoch: Compute the mean amplitude on the whole epochs as baseline
    c. Pre-trigger period: Compute the mean amplitude before the event as baseline
    d. Custom selection: specify yourself the period of the epoch to but considered as baseline (Warning: this period must be inside of the interval specified in the Epoch in point **4**).

7. The same as point **6** but only for the Auxiliary channels if you have specified any in the **Electrode Setting** option of the main EEGpal windows.

8. This is the saving option related to the epoch computation:
    a. *'One file per epoch'*: record a separate file for each clean epoch
    b. *'Merge epochs into one file'*: concatenate all clean epochs into one file
    c. *Concatenare according to subject*: concatenate all the clean epochs for each participant

    Note: Clean epoch refers to epochs that survived the *artefact rejection*

Pannel B: ARTEFACT REJECTION

9. There are three possible ways to detect a bad signal within an epoch:

a. *'Max absolute threshold'*: Define a maximum threshold amplitude. If this amplitude is reached by one or more electrodes during one or more [tf], the epoch will be discarded.
b. *Max step threshold*: This allows amplitude jumps in the signal to be detected. If the difference in amplitude between two time points reaches the specified threshold, the epoch is discarded.
c. *Max range threshold*: The epoch will be discarded if the difference between its highest and lowest amplitude reaches the specified threshold.

An 'Epochs_Summary.xlsx' file will be created containing details of how many epochs have been rejected by each criterion. Please refer to the FAQ for more information.

10. Same as point **9** but only for the Auxiliary channels if you have specified any in the **Electrode Setting** option of the main EEGpal windows.
11. The AR Evaluation tool is optional and allows you to test artefact rejection parameters without saving the EEG average or epoching files. Two Excel output files will be created:
    a. *'ARtesting_Summary.xlsx'*: Summary of number of epochs accepted and rejected (similar as the standard *Epoching_Summary.xlsx*)
    b. *'BadChannels_Summary.xlsx'*: Provide the number of epochs that reach the artefact rejection threshold in each separate channel, expressed as a percentage of the total number of epochs. It allows you to see if one or more channels is responsible for most of the rejection. If so, it is worth interpolating it/them.
    The file contains several sheets, depending on the artefact rejection option you selected:
        i. *SCALP_badamp*: Results for the *'Max absolute threshold'* criterium
        ii. *SCALP_badstep*: Results for the *'Max step threshold'* criterium
        iii. *SCALP_badrange*: Results for the *'Max range threshold'* criterium
        iv. *SCALP_badall*: Results for the combination of all previous criteriums

Pannel C: Averaging (ERP creation) as well as saving parameters

12. You can resample your data at a different frequency, which must be specified in Hz. This is usually done if you want to reduce the sampling rate.
13. **Averaging**: Here you can define how you want to average your data:
    a. **Within-subject**: Activate one of these two options to display one or more averages per participant. This is the default option if you have performed an ERP analysis.:
        i. *'All triggers'*: Average all epochs of all events specified in point **3**.
        ii. *'Per trigger'*: Calculate a separate average for each of the events specified in point **3** (the most used in ERP research).
    b. **Between-subject**: Calculate the average across participants. This is then used to calculate the grant average in ERP analysis.
        i. *'All triggers'*: Average all epochs of all events specified in point **3**.
        ii. *'Per trigger'*: Calculate a separate average for each of the events specified in point **3** (the most used in ERP research).
14. Select the format of the output files.
15. Select the destination folder where the results files will be saved. According to the option chosen, it will create two or three subfolders:

a. *Epochs*: Will contains outputs related to each epochs according the options specified in point **8.** In any case, you will find an Excel spreadsheet containing a summary of each epoch's acceptance or rejection.
b. *Avg_within*: Create output files according to the **Within-subject** option specified in point **13**. If you specify a subject code in the main EEGpal window, the sub-folder will be named according to this code. Otherwise, it will be named 'file01', 'file02', and so on.
c. *Avg_between*: Create output files according to the **Between-subject** option specified in point **13**.

   Please refer to the FAQ for more information.
16. The suffix added to the input file name to create the output file name.
17. For the within-subject result, you can saved them either in separate folder per subject or concatenate all the output file in a single folder (which is more practical when you work with Cartool or Ragu). You can choice between these options.
18. There are three validation buttons:
    a. The **Run** button will carry out the processing parameterized in the Epoching module.
    b. The **Save in memory** button will store all the parameters in memory and close the Epoching module without performing the processing.
    c. The button **Cancel** closes the module without processing and without keeping the entered parameters in memory. The same effect will be achieved by closing the Epoching module window.

FAQ

What does the output look like?
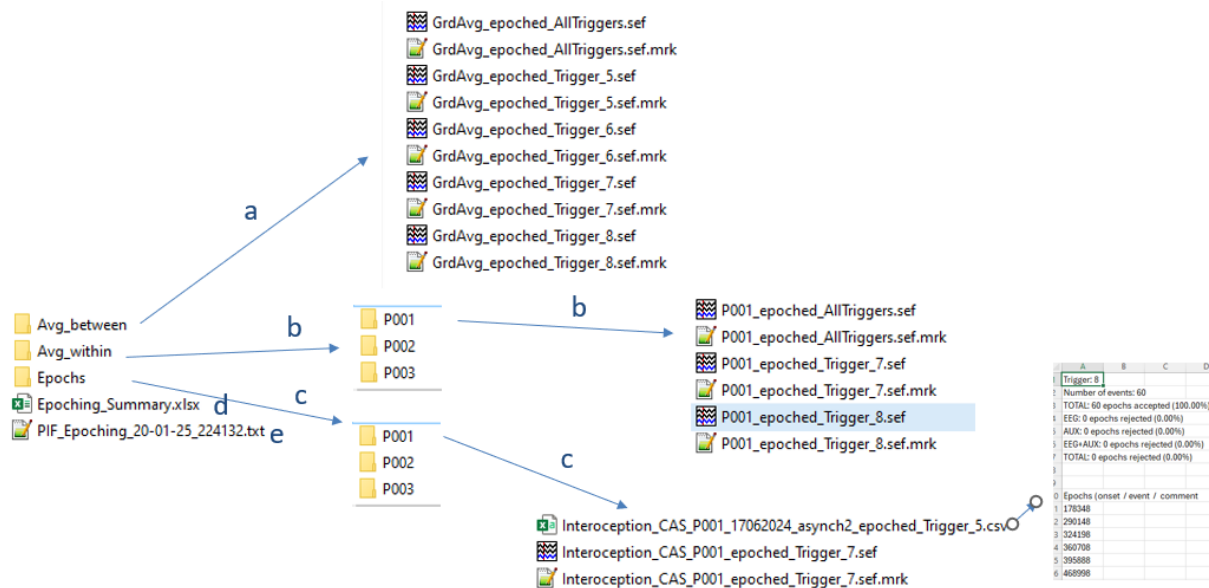
a. Between-subject averaging output (specified in **13**). A folder containing all the grant averages between subjects. In this example the 'All Triggers' option has been selected (file *GrdAvg_epoched_AllTriggers.sef*) as well as the 'Per Trigger' option (file *GrdAvg_epoched_Trigger_*.sef*).

b. Within-subject averaging output (specified in **13**). The folder contains one sub-folder per subject (or per file if no subject has been specified in the EEGpal main window). In this example the 'All Triggers' option has been selected (file *P001_epoched_AllTriggers.sef*) as well as the 'Per Trigger' option (file *P001_epoched_Trigger_*.sef*).

c. Output for the epoch saving option (specified in **8**). The folder has the same structure as the input file. In this example, the input file has been saved in various sub-folders (P001, ...). In this example, the 'Merge epochs into one file' option has been selected. It creates a .csv file with the details of the start of each epoch, as well as the information of whether they were rejected and by which criterion (*Interoception_CAS_P001_17062024_asynch2_epoched_Trigger_5.csv*). The .sef file contains the EEG signal for all concatenated epochs (*Interoception_CAS_P001_epoched_Trigger_7.sef*) and the associated marker file (Interoception_CAS_P001_epoched_Trigger_7.sef).

d. The Epoching_Summary.xlsx file: please see next FAQ question for more information

e. The Processing Information File (PIF) with a summary of all the parameters selected before pressing on the Run button.

What information does the Epoch_Summary.xlsx file contain, and how should it be used?

This file summaries the number of epochs accepted and rejected by the ARTEFACT REJECTION. Here is the meaning of its columns name:

- *Trigger*: the events processed (as specified in **3**)
- *TotalTriggers*: the number of events detected for each file
- *nEEGrejected*: number of epochs rejected by the specified criterium in **9**
- *nAUXrejected*: number of epochs rejected by the specified criterium in **10**

- *nTOTALrejected*: total number of epochs rejected (addition of nEEGrejected and nAUXrejected)
- *nTOTALaccepted*: number of epochs accepted
- *pcEEGrejected*: percentage of number of epochs rejection according criterium specified in 9
- *pcAUXrejected*: percentage of number of epochs rejection according criterium specified in 9
- *pcTOTALrejected*: percentage of the total number of epochs rejected
- *pcTOTALaccepted*: percentage of the total number of epochs accepted

This file can be used to identify participants who are outliers. Generally, participants with fewer than 66% accepted epochs could be outliers. However, this is not a hard and fast rule, so you should check this type of data carefully and consider discarding it.

How can I use the Epoching module to perform frequency analysis or reject periods of poor signal quality?

For the frequency analysis, the signal had to be split into small, equal parts (epochs), with the poor-quality ones removed. In this case, leave the *'Events'* field empty (point **3**), select the entire file with an *'Epoch onset'* of 0 and an *'Epoch offset'* of *'EoF'* (point **4**), activate the *'Split into sub-epochs'* option (point **5**) and select the length of these sub-epochs (for example, 2000 ms). The entire EEG file will then be split into 2000 ms epochs, each of which will be tested for artefact rejection (**B**). To create a new file containing all the clean epochs, select the *'Merge epochs into one file'* option (point **8**). You will then use this file in the **Frequency** module with an epoch length equal to the sub-epoch duration specified in this module.

When should I use baseline correction?

As explained above, baseline correction involves subtracting the mean amplitude of a specified period of the epoch (the baseline) from the entire epoch.

The most common option is to use the entire epoch as the baseline, which compensates for general amplitude differences throughout the EEG recording (mainly due to signal drift). This allows for a more comparable amplitude between epochs (correction by this relative baseline).

Some authors prefer not to use a baseline in order to avoid losing significant differences between conditions. While a baseline correction does make the results more reliable, it also makes them more conservative.

In some cases, a pre-trigger baseline is used. This ensures that the amplitude at the event is 0, but this is only valid if it is hypothesised that there is no anticipation or signal related to the event beforehand, which is rarely the case. This is why this option is usually not used.

Which artefact rejection method should I use?

The three methods proposed by EEGpal have been used in literature. The '*Max absolute threshold'* is the default method use by Cartool. In our laboratory (using a Biosemi system), we usually use a threshold between 80 to 100 µV. In practice, the '*Max range'* does a similar job.
There is no hard and fast rule as to which criterion to use. Look at the summary file generated as output to see how many epochs were rejected by each criterion.