EEGpal: Main windows

Version 2.0, 23.08.2025

The EEGpal toolbox is developed by Michael DePretto (Michael.DePretto@gmail.com) and Michaël Mouthon (University of Fribourg, michael.mouthon@unifr.ch).

EEGpal is an open-source Matlab-based software for automated/semi-automated EEG data pre-processing and analyses. It proposes Graphical User Interfaces (GUIs) that allow EEG pre-processing to be batched across participants with a high degree of flexibility in processing parameters.

The purpose is to offer a complement to the free software Cartool developed by Denis Brunet (University of Geneva, reference: Brunet D., Murray M., Michel C. (2011) Spatiotemporal analysis of multichannel EEG: CARTOOL. *Computational intelligence and neuroscience*, vol. 2011, 813870. DOI : 10.1155/2011/813870). It is also an alternative to the original EEGLAB GUI (reference: Delorme, A., & Makeig, S. (2004). EEGLAB: an open-source toolbox for analysis of single-trial EEG dynamics. *Journal of Neuroscience Methods, 134*(1), 9-21. DOI: 10.1016/j.jneumeth.2003.10.009).

In summary, the authors propose the EEGpal graphical interface (GUI) as a more practical solution than the EEGLAB and Cartool GUIs for automating preprocessing. The authors recommend the use of Cartool software for the visualisation of EEG traces.

Minimum requirement

Matlab 2018b or later

License

GNU General Public License v4.0 or later

Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.

Disclamer

EEGpal is distributed in the hope that it will be useful for EEG data analysis and research purposes, but **without any warranty**. The authors and contributors of EEGpal **make no representations or guarantees regarding the accuracy, reliability, or validity of the results**

obtained using this software. Users are solely responsible for verifying the correctness of their analyses and for any interpretations, decisions, or consequences derived from the use of EEGpal. The software is provided **"as is"**, and the authors disclaim all liability for damages, data loss, or other issues arising from its use.

Before to start

This section reminds the import terminology and concept to know before to use this toolbox.

What is a Time Frame (TF)?

The abbreviation TF (or tf) used in the toolbox stands for 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.

File types

One of the advantages of the EEGpal toolbox is that it works with a serval EEG file format. For example:

- .bdf : Raw Biosemi files
- .eph : Text file format (with tab column as separator) used in Cartool. The first line is an header which contains info about the number of channel, number of time point and sampling rate
- .ep: Text file format (with tab as column separator) used in Cartool and RAGU software (Koenig, T., Kottlow, M., Stein, M., & Melie-García, L. (2011). Ragu: A Free Tool for the Analysis of EEG and MEG Event-Related Scalp Field Data Using Global Randomization Statistics. Computational Intelligence and Neuroscience, 2011, 1-14. DOI: 10.1155/2011/938925.). Contains only data without any temporal information in it
- .sef: Binary EEG file used in Cartool that contains the data and temporal information. It is the default option proposed in EEGpal because this solution uses the least disk space.
- .set/.fdt: File format used by EEGLAB
- .ris: Inverse solution file used in Cartool
- .freq: Frequency analysis file used in Cartool
- .mrk: marker file. It is a text file which include the events during recording (see FAQ at the bottom of this document for more details)
- .eeg/.vhdr: file format from BrainVision recorder. This file format is automatically converted to .set/.fdt during import. It will also create a *.Events.txt file with the new trigger/event name (text to number conversion required). However, you can only import this specific file format. You must work with one of the above formats in EEGpal.

The EEGpal toolbox offers the possibility to easily switch from one file format to the other, thanks to the export tool or the choice of output file format within each module.

What is the difference between Triggers and Markers?

Usually, we add a tag to the EEG recording to mark the position of an event. This tag could be a triggers or a markers. The difference is that the triggers are included in the EEG file (this is the case for .bdf and .set file formats), while the markers are recorded in a separate text file named eeg_file.mrk (this is the case for .eph, .ep, .sef file formats). EEGpal automatically detects these two cases and can handle them.

Processing Information File (PIF)

Each time you perform a processing step with an EEGpal tool or module, a Processing Information File (PIF) is created. This is a text file containing a summary of all input and output files and processing parameters. It allows you to keep track of the steps you have performed.

Example:

```
EEGpal ICA Processing Info File
======================================================================

Date: 23-Aug-2025 09:30:54

App name:              EEGpalCS_mod_ICA.mlapp
App folder:            D:\Software\EEGpal\EEGpal\modules
App last save:  20-août-2025 22:50:34
Current version of EEGpal:     1.510000
ICA decomposition/recomposition use EEGLAB tools and components labeling use ICLabel tool.
Current version of EEGLAB:     2023.100000
Current version of ICLabel:    1.600000


DATA

Input data root folder:       D:\FND_lab\temp_EEG_data\batch\output\02_Filtering
Output data root folder:      D:\FND_lab\temp_EEG_data\batch\output\03_ICApruned
Output suffix string:         ICApruned
Output files extension:       .sef
Batch Processing:             This processing was performed in Batch mode.
Batch Bad component selection:        Automatic Bad Components Rejection


ELECTRODE SETTING

Electrodes setting file:      D:\Software\EEGpal\EEGpal\Resources\biosemi64_officialAB_update.locs


ICA DECOMPOSITION

These INPUT files has been decomposed in the temporary folder:       D:\FND_lab\temp_EEG_data\batch\output\02_Filtering\ICAtemp\decomposed


INPUT FILES (File name  /  Sampling rate  /  N. channels  /  Folder)

P006_card_synch_FBI_VR_05052025_run3_filtered.sef        2048          64           ..


BAD CHANNELS IGNORE DURING ICA DECOMPOSITION

INPUT FILES (File name  /  Subject  /  Bad channels

P006_card_synch_FBI_VR_05052025_run3_filtered.sef        P006          8 9 20 21 26 27 29 31 39 40 57 62 63 64
```

What is a typical EEG processing pipeline in EEGpal?

Everyone has its own recipe, but the pipeline used by the Toolbox creator are detailed at the bottom of this document.

Main window

There is five important section in the EEGpal main window:

A. **Importation section**: Tool to import/Select the EEG file in the Data table B (see below for details)

B. **Data table**: Contains the list of files which will be used for the next processing step. It contains characteristic info about files like the number of channels, sampling rate, number of time frame, subject and condition

C. **Selection and update info about files listed in Data table**: Specify subject and condition information, sub-selection and electrode localization settings (see below for details)

D. **Saving options and Tools**: Permit to define the default format of file saving as well as export the file of the Data Table in another format. You can also find here the File Cut tool to cut your EEG files according to your trigger/marker

E. **Processing Modules**: List of modules to preprocess your data

F. **Analysis modules**: Modules which permit various type of analysis on your preprocessed data.

G. **Processing parameters and batch tool section**: Save or load all the parameters that you have set in the various EEGpal modules. You can then program a batch to automatically run all the preprocessing steps

Import Data
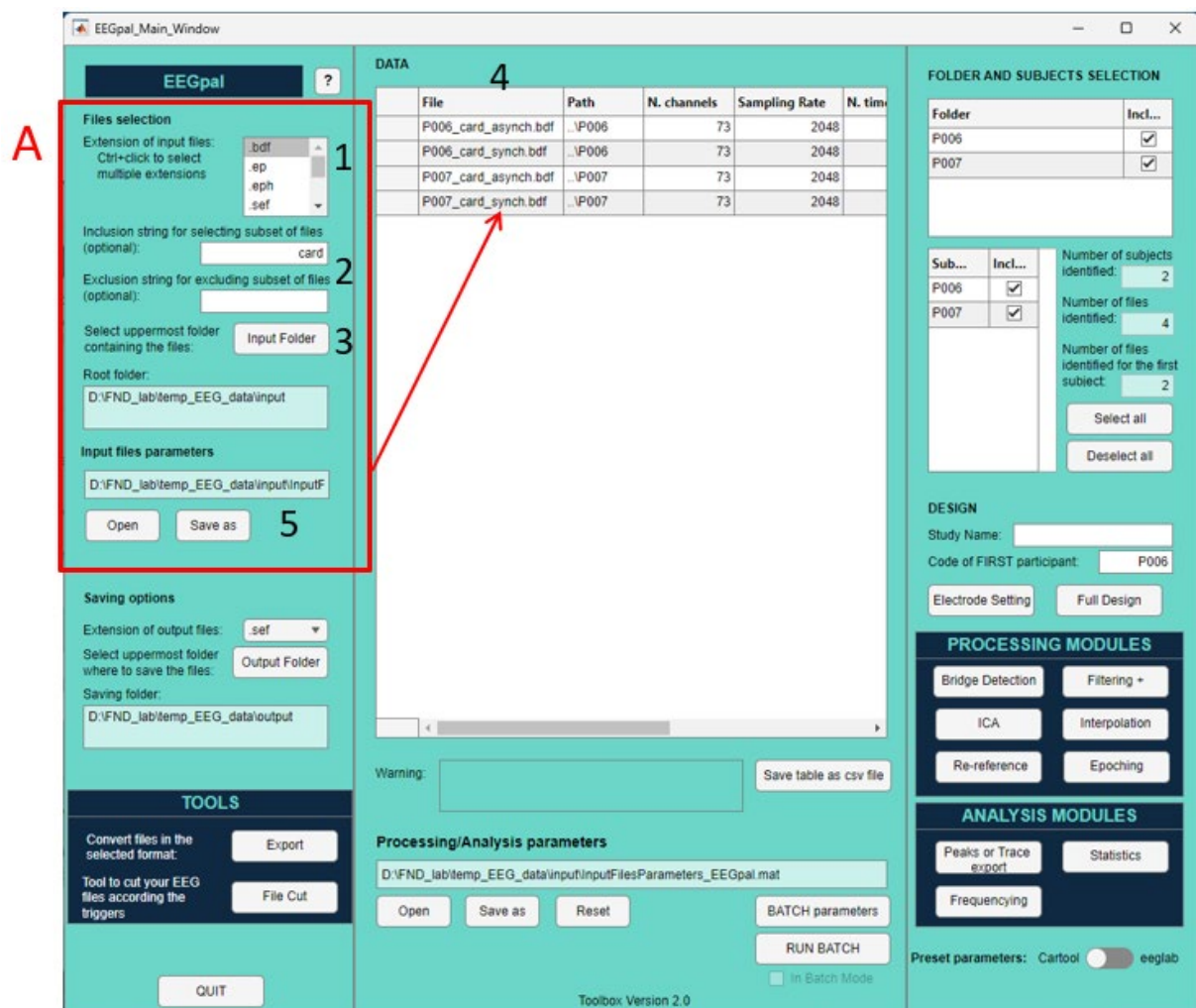
Select the raw data format (see **File types** section of this manual to have more details on the supported file format).

1. Optional: If you want to filter which files should be imported or not according to a string pattern in their file name, you can write in a character before to press the Input Folder button of the step 3 (for example, if you write 'synch' in Inclusion string field, then the toolbox will import only the files with has this pattern their name).
2. Press the Input Folder button and select the directory containing all the EEG files you wish to import. These files can be placed in sub-directories. The toolbox will select all files corresponding to the search criteria defined in 1 and 2, whether they are in the main folder or in sub-folders. It takes some time to fill the list in the central panel. Give Matlab time to complete this list before doing anything.
3. The main panel lists all the files that the toolbox has found. It informs us in order: file name, location on disk, sampling frequency, total number of electrodes recorded in the file, and the number of TFs contained in the file.
4. Optional: You can save the list of imported files. You can then load it later to automatically fill in the data table, so you won't need to perform steps 1-3 again.

Define participant and adjust the selection
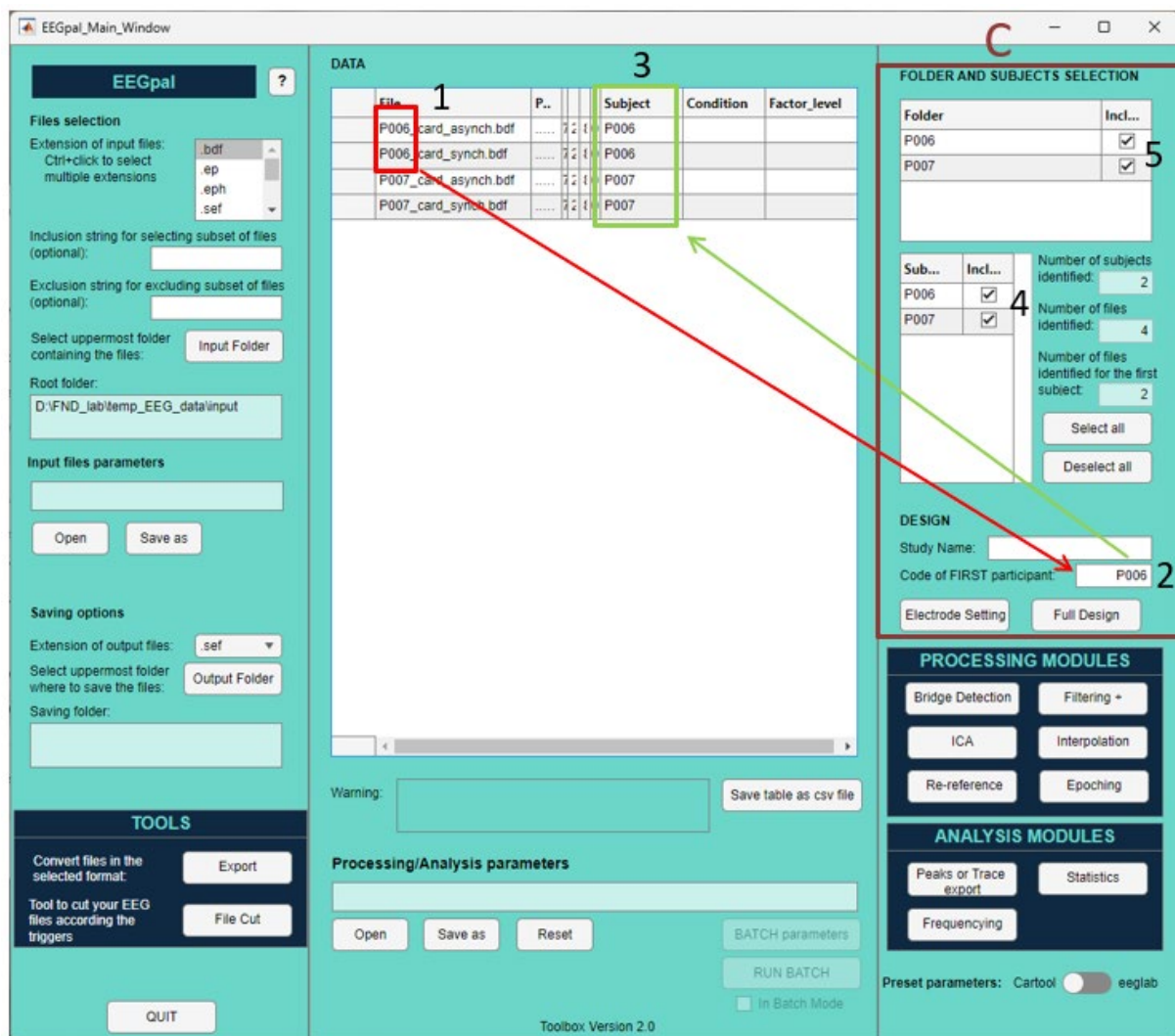
1. To specify different subjects in the EEGPal toolbox, identify the participant code in the EEG name of the first entry in the data table.
   WARNING: The participant code must have the same number of characters and the same position in the file name for every participant. Name your file according to this rule.
2. Enter the participant code that appears first in the data table (determine in step 1).
3. The toolbox will then automatically identify which participant the files belong to. The participant codes will be added to the 'Subject' column. Check that these values are correct.
4. You can then deselect specific participants to remove them from the data list.
5. You can also deselect files according to the sub-folder from the input folder

Note: defining the participants is especially important for the ICA, Epoching and Full Design +Statistics modules. These steps are indeed required to group all data from one subject. Otherwise, each file will be treated independently.

Loading the electrode coordinate file
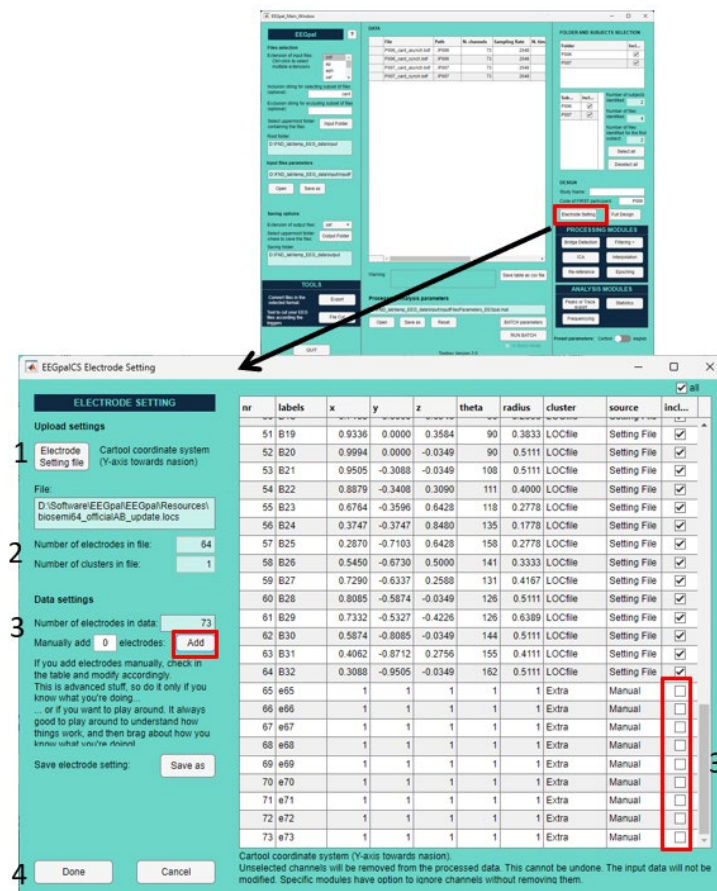
1. Load the .xyz, .els or .locs file containing the spatial coordinates for locating the electrode position on the EEG cap.
   In the current situation there are already two coordinate files for a 64 channel Biosemi configuration in the *Resource* sub-folder of the EEGpal. The *biosemi64_officialAB_update.locs* file contains the coordinates found on the official Biosemi website with a spherical distribution of electrodes. The file *64-Biosemi_OVAL_AB.xyz* is an oval distribution of the electrodes after coregistration with the MNI template in Cartool (more physiologically accurate). *BrainVision64elec.locs* contain a 64 configuration with a BrainVision system.
   You can add your own coordinate file directly in the Resource sub-folder of EEGpal to gain time.
2. The toolbox informs you of the number of electrodes present in this coordinate file, as well as the number of clusters. A cluster is a group of electrodes that can be specified within an .els file (e.g. to separate EEG electrodes from external channels).
3. Optional: To remove supplementary channel of non-interest: The toolbox compares with the number of electrodes in the EEG files.
   In our case, the EEG files contain 9 more electrodes than the coordinate file (ECG + Resp channels). Click on the Add button to manually add these 9 channels to the table on the right. Make sure that these new electrodes are deselected (the default setting) so that they are ignored/deleted (and not interpolated) in the analysis.
4. Click on Done

Export into another EEG file format
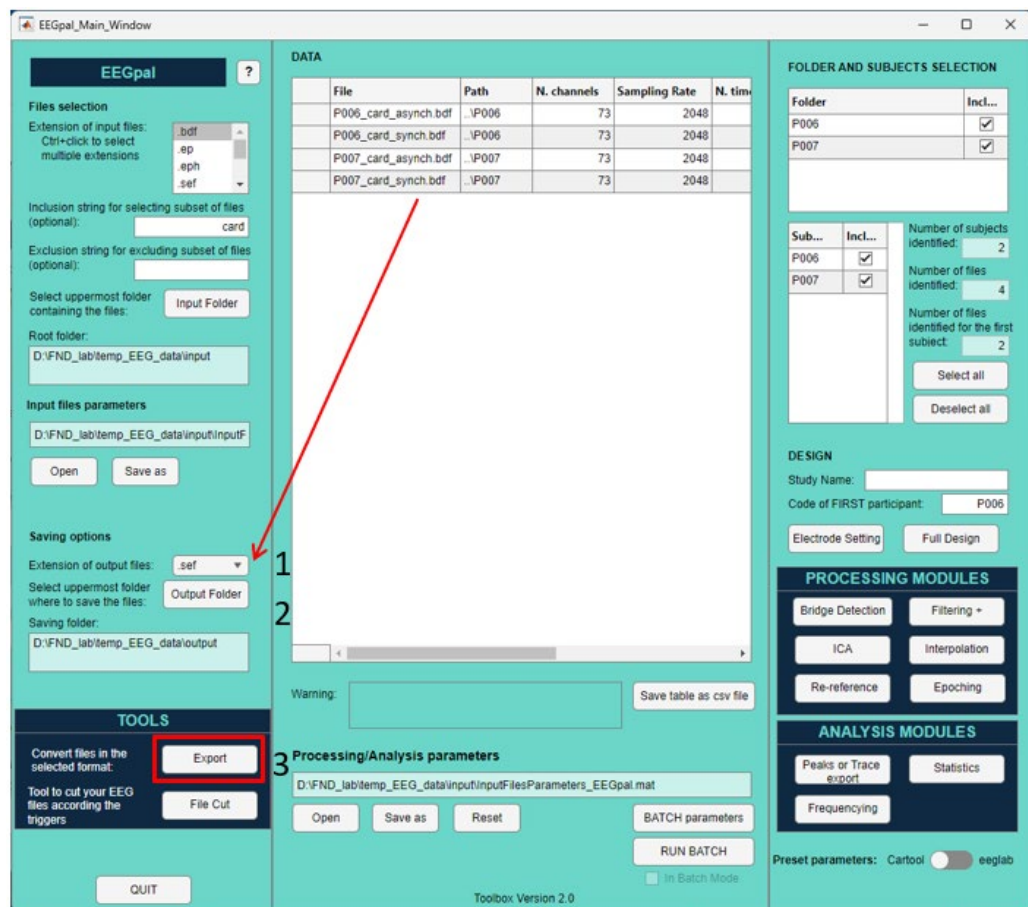
1. Select the output format (.sef, .eph, .ep for Cartool or .set for eeglab)
2. Choice the folder where the files will be stored
3. Click on the Export button

Bridge Detection
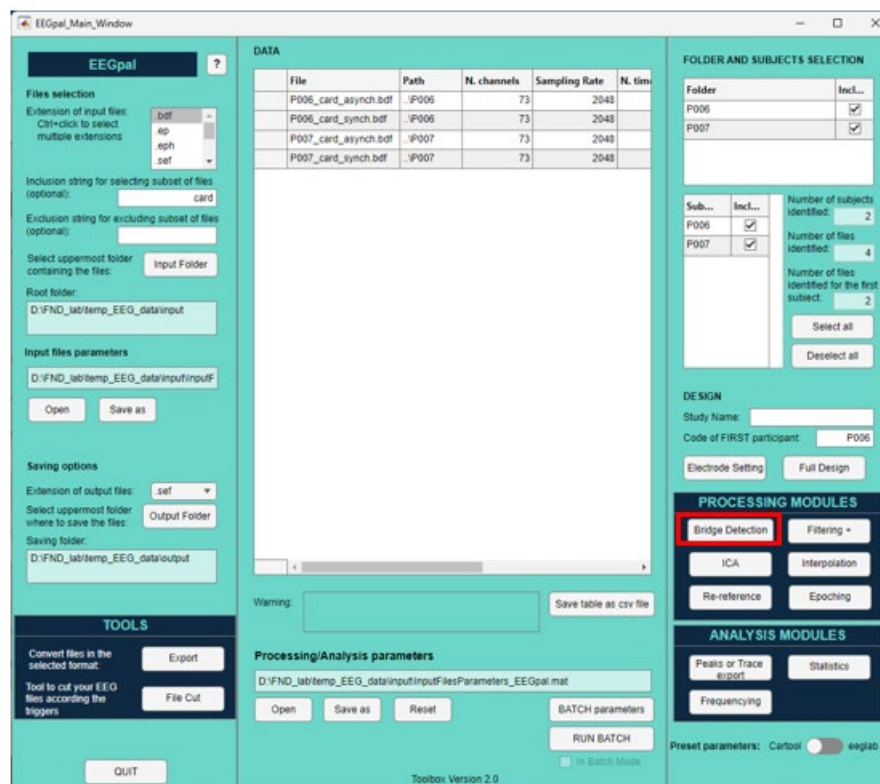
The first preprocessing step is to detect any bridges (where two or more electrodes are connected due to gel leakage, resulting in them recording the same signal). It is specially important to detect and remove these bridge if you perform ICA decomposition or Inverse solution analysis.

This **Bridge Detection** module calls the eeglab script eBridge (Alschuler et al (2014). Clin Neurophysiol, 125(3), 484-490.) which automatically detects them. The module will save a Excel sheet (.xlsx) with the results of this inspection inside of the **Input folder/Root folder** specified during the importation of the data. These results will also be stored in the processing parameter files, allowing the ICA and interpolation modules to access them.
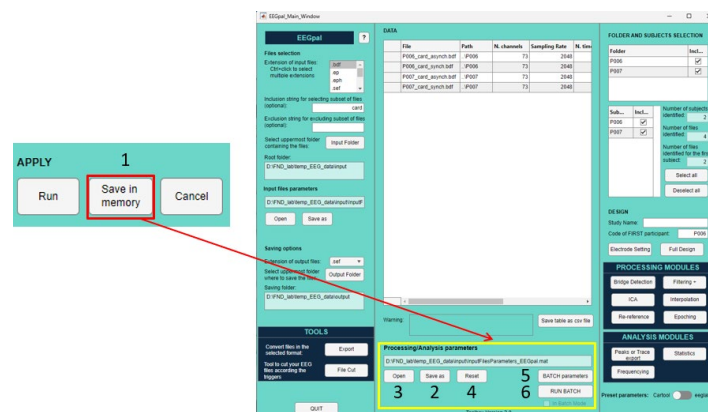
Other Procession and Analysis modules

Help files for each of these modules are available once you have opened them by clicking the Help button in the top-left corner:

EEGpal is a program that is easy to use, and these manuals will help new users to learn how to use it. They also provide information about the background knowledge required for using the program.

Processing/Analysis parameters and Batch system

The Processing/Analysis parameter plays an important role in how EEGpal functions. It stores all the parameters that you enter into the various processing and analysis modules. When you open a module and set the processing parameters (for example, adding a notch filter at 50 Hz in the Filtering+ module), clicking the **Save in memory button** (step 1) will save the changes to the parameter file in the main window. This means that when you open the Filtering+ module again, it will be restored to the exact state it was in when you left it. This is useful for restoring all your processing choices when you open EEGpal again later.

In addition, this system allows you to create a batch that will perform all the processing steps automatically. To do so, you will need to program the succession of processing steps using the **BATCH parameters** button. More details about this process can be found in the help section of this tool's window.

Processing/Analysis parameters

1. In the Modules window, you will find the **Save in memory** button. Click this to transfer your chosen parameters to the *SessionParameters* variable in the EEGpal main window. Please note that selecting the **Cancel** button or closing the window will not save your parameter choices.
2. Click on the **Save as** button to save all the processing parameters for all modules and tools to a *.mat file.
3. Click on **Open** button to restore all processing parameters from a file saved in step 2.
4. The **Reset** button erases all the processing parameters in memory, allowing you to restart the setup process.

Batch system

5. The **BATCH parameters** button is only available after you have created (step 2) or loaded (step 3) a processing parameter file. A separate window will open to allow you to configure the succession of each processing step (see the help section in this window for more information).
6. The **RUN BATCH** button is only available after you open and validate the **BATCH parameters** (step 5). It will run the batch in the order that you configured in Step 5. The small tick mark displayed behind the button indicates whether EEGpal is currently processing a batch.
Note: If a problem occurs or the wrong parameter is selected during batch execution, you can correct the problem and press this button again to resume the batch from where it stopped.

Typical EEG processing pipeline for an Evoked Related Potential (ERP) dataset

In this document, we distinguish the preprocessing, which permits to clean the data for the data analysis (the processing). The preprocessing is similar between the type of analysis but with some variation. Here I will first describe the case of Event Related Analysis (ERP).
This pipeline was proposed and validated by Michaël Mouthon and is used in the FND lab at the University of Fribourg, Switzerland.

Preprocessing

1. Import raw data into the main EEGpal windows and identify the subject as explained in the chapter *Define participant and adjust the selection* of this manual.
2. Specify the spatial position of the channels by loading an electrode coordinate file as explain in the chapter *Loading the electrode coordinate file* of this manual.
3. Perform a *Bridge Detection* in order to check if there is any unwanted connection between channels
4. Use the module *Filtering* for the first clean of the data. Filter to keep the signal between 0.5 and 40 Hz. Add a Line noise removal either by a Notch filter at 50 Hz or by using cleanline
5. Inspect visually the result file in Cartool to check if there are wrong channels (behave totally differently as the neighbors for some time)
6. Use the *ICA* module to perform an ICA decomposition by ignoring bad channels determine in point 3 and 5.
   Inspect manually the 24 first compounds of each file to removed compounds link to Eye artefacts
   Recomposed the signal without the previously selected compounds
7. Use the *Epoching* module to perform a temporary Epoching and inspects visually in Cartool the ERP files. Selected the bad electrodes which negatively impact the ERP to be interpolated
8. Use the *Interpolation* module to interpolated bad electrodes determined in 3,5 and 7.
9. Use the *Refef* module, to change the reference electrode to the average reference
10. Use the Epoching module to compute the final ERPs