

# FAST manual

---

FAST developpers at the Cyclotron Research Centre

Jessica Schrouff  
Dorothee Coppieters  
Remy Lehembre  
Yves Leclercq  
Pierre Maquet  
Quentin Noirhomme  
Christian Degueldre  
Christophe Phillips

Cyclotron Research Centre  
University of Liège, B30  
8 Allée du Six Août  
4000 Liège, BELGIUM

<http://www.montefiore.ulg.ac.be/~phillips/FASST.html>

August 2, 2013

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Installing &amp; launching the toolbox</b>	<b>5</b>
2.1	Installation . . . . .	6
2.2	Launching and batching . . . . .	6
<b>3</b>	<b>Data format &amp; channels setup</b>	<b>6</b>
3.1	Data format conversion with FAST . . . . .	8
3.1.1	Brain Products data format conversion . . . . .	8
3.1.2	Other data format conversion . . . . .	8
3.2	Channel definition . . . . .	9
<b>4</b>	<b>Main functions of the toolbox</b>	<b>9</b>
4.1	General tools . . . . .	10
4.1.1	Displaying one M/EEG data file . . . . .	10
4.1.2	Comparing multiples M/EEG data files . . . . .	13
4.1.3	Appending M/EEG data files . . . . .	14
4.1.4	Chunking a time window from an M/EEG data file . . . . .	15
4.1.5	Computing the spectrogram of one M/EEG data file . . . . .	15
4.1.6	Displaying the spectrogram or power spectrum of an M/EEG data file . . . . .	16
4.2	EEG-MRI artefact rejection tools . . . . .	17
4.2.1	Gradient artefact rejection . . . . .	17
4.2.2	Pulse artefact rejection . . . . .	19
4.3	Sleep specific tools . . . . .	21
4.3.1	Manual sleep scoring . . . . .	21
4.3.2	Spectral power calculation and statistics . . . . .	23
4.4	Automatic wave detection tools . . . . .	23
4.4.1	Slow Wave detection . . . . .	24
4.4.2	Sleep Spindles detection ( <i>BETA version!</i> ) . . . . .	27

<i>CONTENTS</i>	3
<b>5 Advanced Users</b>	<b>31</b>
5.1 Addition to the SPM8 structure . . . . .	31
5.1.1 The <code>D.info</code> field . . . . .	31
5.1.2 The <code>D.CRC</code> field . . . . .	31
5.2 “Hidden” functions . . . . .	33
<b>6 Acknowledgments</b>	<b>33</b>

# 1 Introduction

“FAST” stands for “fMRI Artefact rejection and Sleep Scoring Toolbox”. This M/EEG toolbox is developed by researchers from the Cyclotron Research Centre, University of Liège, Belgium, with the financial support of the Fonds de la Recherche Scientifique-FNRS, the Queen Elizabeth’s funding, and the University of Liège. On Dr. Pierre Maquet’s impulse we started writing these tools to analyze our sleep EEG-fMRI data and tackle four crucial issues:

**Continuous M/EEG.** Long multi-channel recording of M/EEG data can be enormous. These data are cumbersome to handle as it usually involves displaying, exploring, comparing, chunking, appending data sets, etc.

**EEG-fMRI.** When recording EEG and fMRI data simultaneously, the EEG signal acquired contains, on top of the usual neural and ocular activity, artefacts induced by the gradient switching and high static field of an MR scanner. The rejection of these artefacts is not easy especially when dealing with brain spontaneous activity.

**Scoring M/EEG.** Reviewing and scoring continuous M/EEG recordings, such as is common with sleep recordings, is a tedious task as the scorer has to manually browse through the entire data set and give a “score” to each time-window displayed.

**Waves detection.** Continuous and triggerless recordings of M/EEG data show specific wave patterns, characteristic of the subject’s state (e.g., sleep spindles or slow waves). Their automatic detection is thus important to assess those states.

Currently the toolbox can efficiently deal with these issues, as described in section 4. Eventhough the toolbox was originally divided for EEG data, it also handles MEG and LFP data without any problem, if in the right format (see section 3). FAST is in effect an add-on toolbox for SPM8[15, 13] and relies on SPM8 data format and functions to perform some operations.

Troubles usually come from the data format and channel setup but see section 3 for FAST requirements and solutions. The “eye-balling” and manipulation of continuous multi-channel recordings is eased by a series of tools (see section 4.1). Specifically, for EEG-fMRI acquisitions (see section 4.2), FAST can operate directly on the raw data acquired with a “BrainAmp MR” system (Brain Products GmbH, Gilching, Germany) and includes the well-known AAS method[1] for the “gradient artefact” rejection, as well as the recently published “constrained ICA” method[10] for the rejection of the “pulse artefact”. Other “classic” methods for the “pulse artefact” rejection are also available. And an easy GUI is available for the manual scoring of sleep EEG by different users (see section 4.3). Some statistics and sleep specific features can then be automatically extracted.

FAST is available for free, under the GNU-GPL and, obviously, it comes without any warranty: you should use it at your own risk. This is an ongoing research project and we plan to add new features and functionalities, according to our own needs and the feedback from users. If you are happy with the results obtained with FAST and publish them, please use the following references:

- for the toolbox in general and the sleep specific tools[11, 12]

J. Schrouff, D. Coppieters, R. Lehembre, Y. Leclercq, P. Maquet,  
Q. Noirhomme, C. Degueldre, E. Balteau, and C. Phillips,  
fMRI Artefact rejection and Sleep Scoring Toolbox (FASST).  
Cyclotron Research Centre, University of Liège, Belgium, 2009.  
<http://www.montefiore.ulg.ac.be/~phillips/FASST.html>

and

Y. Leclercq, J. Schrouff, Q. Noirhomme, P. Maquet, C. Phillips,  
 ‘‘fMRI Artefact rejection and Sleep Scoring Toolbox’’,  
 Computational Intelligence and Neuroscience, 2011.  
<http://www.hindawi.com/journals/cin/2011/598206/>

- for the constrained ICA ‘‘Pulse artefact’’ rejection method[10] in particular

Y. Leclercq, E. Balteau, T. Dang-Vu, M. Schabus, A. Luxen, P. Maquet, and  
 C. Phillips.  
 ‘‘Rejection of pulse related artefact (PRA) from continuous electroencephalo-  
 graphic (EEG) time series recorded during functional magnetic resonance  
 imaging (fMRI) using constraint independent component analysis (cICA)’’,  
 NeuroImage, 44(3):679–691, 2009.

Note that FAST includes a few routines from 3 other freely available toolboxes:

- EEGLAB[3, 4], mainly for a few functions used by the constrained ICA ‘‘pulse artefact’’ rejection, available from <http://sccn.ucsd.edu/eeglab/>;
- the FMrib plug-in for EEGLAB[18], for the ECG peak detection and classic PCA/Gaussian mean ‘‘pulse artefact’’ rejection, available from <http://users.fmrib.ox.ac.uk/~rami/fmribplugin/>;
- the ‘‘Mutual Information computation’’ package[21], for some the selection of correction matrices during cICA ‘‘pulse artefact’’ rejection, available from <http://penglab.janelia.org/proj/mRMR/>.

We thank their authors for letting us use their work.

We would also be thankful to hear from you if you find bugs in the code. Feel also free to submit add-ons to this toolbox but please note that support will not be provided for ‘‘home-made’’ changes of the distributed code.

## 2 Installing & launching the toolbox

In order to work properly, FAST needs to have those 2 softwares installed:

- a recent version of MATLAB , we used version 7.5 (R2007b) to develop FAST. Any later MATLAB version should work, *in theory*.
- the latest version of the ‘‘Statistical Parametric Mapping Software’’, i.e. SPM8 [15, 13]. You can download it from: <http://www.fil.ion.ucl.ac.uk/spm/software/>. Install it in a suitable directory, for example C:\SPM8\, and make sure that this directory is on the MATLAB path<sup>1</sup>.

**NOTE:** Originally FAST relied on routines from MATLAB ‘‘Signal Processing Toolbox’’ for some filtering functions. In this version we have tried to lift this requirement. If you do not have the toolbox, filtering and time-frequency operations are executed by freely available equivalents<sup>2</sup> distributed under the GNU General Public License .

<sup>1</sup>No need to include the subdirectories!

<sup>2</sup>see in the subdirectory \FAST\SPTfunctions.

## 2.1 Installation

After registering and downloading the zipped file containing **FAST**, the installation proceeds as follow:

1. Uncompress the zipped file in your favourite directory, for example `C:\FASST\`;
2. Launch MATLAB ;
3. Go to the “File” menu → “Set path”;
4. Click on the “Add folder” button and select the **FAST** folder, i.e. `C:\FASST\` if you followed the example;
5. Click on save.

A few routines from the “Mutual Information computation” package<sup>[21]</sup> are written in C++ (.cpp files) and should thus be compiled for your specific operating system. We are providing these compiled routines for the usual OS’s such as: Windows XP (32 bits), Windows 7 (64 bits), Mac OS 10, Linux (32 bits). If your OS is not listed (or routines do not work properly), you can compile the routines with the `makeosmex.m` script available in the **MInfo** subdirectory.

## 2.2 Launching and batching

Once installed, there are three ways to call up **FAST** functionalities:

- to launch the toolbox GUI, just type `crc_main` or `fasst` at the MATLAB prompt and the main GUI figure will pop up, see Fig. 1. From there on simply click on the tool you want to use;
- some functions of **FAST** have been integrated into SPM8 `matlabbatch` batching system<sup>[5]</sup>, the batching GUI is launched by typing `crc_batch` at the MATLAB prompt<sup>3</sup>;
- most tools can also be called individually by calling them directly from the MATLAB prompt, or for scripting in a .m file.

## 3 Data format & channels setup

Data format is always an issue in EEG and MEG since each company enjoys his own specific (and usually proprietary) format. So we decided to use the MATLAB based and open SPM8 `meeg` data format. Therefore **FAST** internally works with SPM8 `meeg` data format:

- a .mat “header” file, which contains all the information about the data (channel names and types, sampling rate, stimuli, data format, etc.) as well as room for user-specific data; and

---

<sup>3</sup>check SPM8’s manual for more information about `matlabbatch`

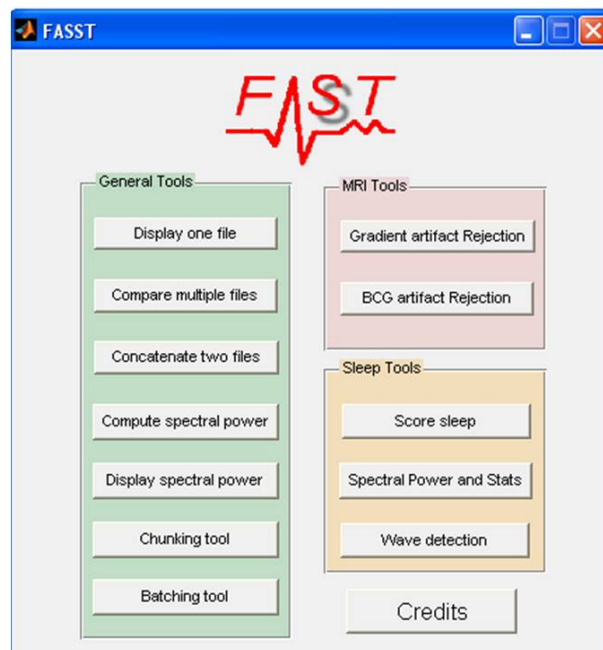


Figure 1: Main GUI window. The 3 families of tools (General, EEG-fMRI, and sleep) are clearly separated.

- a `.dat` binary file<sup>4</sup>, which stores the recorded data themselves as a raw list of numbers<sup>5</sup>.

The resulting data files are also stored in that format. Informations, like the sleep score encoded by users, are simply added to the header data structure and do not interfere with SPM8 machinery. Advanced users are advised to have a look at SPM8’s manual for a description of the SPM8 M/EEG tools and data format.

Data can thus be imported in two ways, either via SPM8 “convert” function or directly via FAST:

- SPM8 relies on the “FieldTrip” (FT) toolbox [20, 19] to read in and convert a very large number of EEG/MEG data formats (pretty much any format that exists). SPM8 generally reads in the header and creates the `.mat` “header” file, then goes through the data file and creates the `.dat` binary file. The 2 files created have the same name as the original data but prepended with “spm8\_”.
- FAST directly handles far fewer data format than SPM8-FT. So far only Brain Products (Brain Products GmbH, Gilching, Germany), raw-exported EGI (Electrical Geodesics Inc., Eugene, OR, USA), EDF [7, 9, 8] and BCI2000 (<http://www.bci2000.org/>) data formats are directly supported. Those types of data can be directly selected via FAST GUI and are converted ‘on the fly’. There is therefore no “import” button in FAST.

FAST import can be faster for the supported format but SPM8/FT approach is probably the safest for any other format, as both SPM8/FT are constantly improved. Nevertheless depending on the way raw original data are stored on disk, SPM8/FT can more than double the size of your data on disk<sup>6</sup>. In both cases, before any ‘advanced processing’, i.e. anything more than just having a look at the signal, it is advisable to check the channel informations (type, scale, units, name, 2D location, etc.) using SPM8 `meeg` reviewing function.

<sup>4</sup>it’s only a binary finale, so any extension could be used actually, and that’s what FAST does sometimes.

<sup>5</sup>By default SPM8 saves the data as `FLOAT` but other format, e.g. `INT16` can be used.

<sup>6</sup>Sometimes raw EEG data are saved in `INT16` format, i.e. 2 bytes per sample, but SPM8/FT converts them to `FLOAT`, i.e. 4 bytes per sample.

### 3.1 Data format conversion with FAST

When converting data with **FAST**, there is a bit more information that can be retrieved than with SPM8/FT. If the recording time of the data, i.e. clock time of the beginning of the recording, is available, then that information is extracted and saved in the `meeg` object. This information can be useful for

- single file display. For example with night long recording, it's easier to be able to look at the 'real world' recording time instead of the time elapsed since the beginning of the recording
- multiple file display. The display of multiple data files (raw vs. chunked and filtered for example) is then synchronized on the 'real world' recording time.
- data handling. When chunking or appending data set(s), the 'real world' recording time is accounted for.

#### 3.1.1 Brain Products data format conversion

**FAST** can *directly* read and import raw data from the Brain Products (Brain Products GmbH, Gilching, Germany) EEG amplifiers: BrainAmp, QuickAmp and V-Amp<sup>7</sup>. EEG recorded for several hours at the row (like in a sleep study) or during fMRI acquisition usually generates massive data files (up to several Gb for fMRI-sleep recordings!) due to recording duration and the high sampling rate (typically 5kHz for EEG-fMRI). Being able to read and process these raw data without a lengthy exportation/importation procedure or the need to make copies of this huge data file, was thus particularly important for us. Since we own BrainAmp MR amplifiers for EEG-fMRI recording (as many other laboratories do), we focused on their data format.

The raw data from BP amps consist in three files, for example `fname.eeg`, `fname.vhdr`, `fname.vmrk`, which are respectively the binary data file, the header file and the marker file. The header and marker files are simple text files. These two are directly read in by **FAST** and translated into SPM8 data structure<sup>8</sup>, which is then directly linked to the binary data file. Thus be aware that the binary data file containing the data is **NOT** copied at this "importation" stage! The following processing step(s), for example the "gradient artefact removal" (see section 4.2.1), will thus access the *original* binary data file. Even though the original binary data should never be overwritten by **FAST**, it is still advisable to keep a safe copy of your original "raw data" in a safe place.

#### 3.1.2 Other data format conversion

Raw continuous EDF, EGI and BCI2000 data formats are now supported by **FAST** but some restrictions apply:

- EDF data set sometimes have channels with different sampling rates. Channels with lower sampling rates, usually physiological measures (breathing, CO2, etc.) are simply upsampled to the highest sampling rate, usually the EEG channel, in order to have the same sampling rate for all channels. Note also that .edf data are assumed to be in a INT16 format, which is usually the case but not necessarily so.

<sup>7</sup>V-Amp special edition have crypted headers (with a `.ahdr` extension), you need to export the data to "binary data format" in order to obtain the `*.vhdr` and `*.dat` files.

<sup>8</sup>This includes the triggers recorded along side the EEG.



- EGI data should be continuous “raw exported” data. For the moment, the data file is read in and re-written on disk after removing the channels associated with the triggers/inputs<sup>9</sup>. These trigger/inputs are turned into events in the SPM8 `meeg` object.
- BCI2000 data importation is only a *beta* version!

As stated above, other (raw) EEG/MEG data formats should be imported via SPM8 importation routine. In this case the raw binary file is copied and the 2 files created (`*.mat` and `*.dat`) are prepended with “`spm8_`”.

### 3.2 Channel definition

Channel definition (name, type, sacele, unit, 2D-location, etc.) is in line with that of SPM8. Therefore it is strongly advisable to use SPM8’s review function to set up the channel informations properly and, possibly, apply a montage to re-reference all or some channels. Still within the toolbox we use an enhanced “channel setup” file (`CRC_electrodes.mat`) specific to EEG recordings, and introduce an extra bit of information “`crc_types`” for the channels. This allows the on-the-fly display of bipolar montage along EEG channels and the use of different channel scalings, e.g. for EEG and ECG channels, for the display. Most common channel names are already available within the toolbox. If you used a lab-specific setup (and most everyone does...) or weird channel names, you should then add those channels to `CRC_electrodes.mat`. For more information on how to adjust the channel definitions to your specific channel setup, please read and edit the function `crc_electrodes.m`. A new `CRC_electrodes.mat` is then generated by executing `crc_electrodes.m`.

## 4 Main functions of the toolbox

The following tools and features are available in the current version of FAST:

- General tools
  - Displaying one M/EEG data file
  - Comparing multiple M/EEG data files
  - Appending M/EEG data files into a single data file
  - Chunking a time window from an M/EEG data file
  - Computing the spectrogram of one M/EEG data file
  - Displaying the spectrogram or power spectrum of an M/EEG data file
- EEG-fMRI artefact rejection tools
  - Gradient artefact rejection: AAS method
  - Pulse artefact rejection: AAS, ‘Gaussian mean’ and cICA methods

---

<sup>9</sup>It should be possible though to avoid reading-and-writing the big data file on disk and directly access the original raw data. Maybe in a future update...

- Sleep specific tools
  - Manual sleep scoring
  - Spectral power calculations and statistics
- Automatic wave detection tools
  - Slow waves (SW)
  - sleep spindles (SP, *beta version*)

## 4.1 General tools

These are tools to display, review, compare, process, manipulate and handle (long) multichannel continuous data files.

### 4.1.1 Displaying one M/EEG data file

This tool is designed to display multiple channels of one continuous M/EEG recording and browse throughout the data. The display routine includes built-in “on-screen” filtering and re-referencing function. There is therefore no need to perform these pre-processing steps before visualizing the data. In fact you can use these options to see what would be the effect of filtering or re-referencing on your data.

When you launch the display routine, the selection channel interface appears first. At this point you have to choose the list of channels you want to look at. To select a channel, you just have to click on its name in the left hand column (Fig. 2-A). The name of this channel will disappear from the left column and appear in the right column (Fig. 2-B). The channels recognized by the toolbox, i.e. with 2D-localisation information, will appear in the lower right white box (Fig. 2-C) according to their position on the scalp. Some buttons allow you to automatically select a set of electrodes (Fig. 2-D). When you have selected all the channels of interest, click on the ‘PLOT’ button to launch the display interface (Fig. 2-E). The ‘SCORE’ button (Fig. 2-F) leads to the same display but with the scoring tools and menu enabled, see section 4.3.1.

A display window appears with the channel display in the main central box (Fig. 3-A). The scaling, time window settings, number of channels displayed, re-referencing and filtering option are on the right part of the window (Fig. 3-B). There are two scrolling bars: one to quickly browse throughout the data over time, below the channel display (Fig. 3-C), and another one to browse through sub-sets of selected channels, on the right of the channel display (Fig. 3-D).

In the main display window (Fig. 3-A), standard unipolar channels are displayed in dark blue, electrocardiographic (EKG or ECG) in red and the others in lighter blue. The scaling of the various channels will depend on their type, as defined in the SPM8 `meeg` object. If this information is not available (poor importation or the data have not been “properly” prepared) then FAST uses the `crc_types` variable of the channel setup file `crc_channels.mat` to find the channels that are automatically rescaled and the bipolar montages<sup>10</sup>.

<sup>10</sup>In this case, it will not be possible to scale correctly EEG and MEG channels simultaneously!

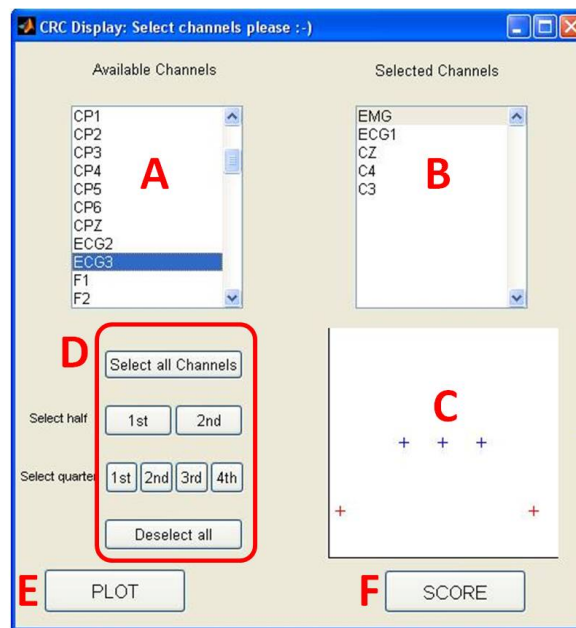


Figure 2: Channel selection GUI: A. channel list, B. selected channels, C. selected channels 2D-location, D. pre-specified automatic channel selection, E. plot selected channels, F. plot selected channels and score the signal.

The ‘Window Size’ (Fig. 3-B1) fixes the length of the window displayed (in seconds). The ‘Scale’ setting fixes the relative distance in  $\mu V$  between two channels in the main window (Fig. 3-B2). The number between two channel labels stands either for minus half the scale for the above channel and for plus the half scale for the lower channel. This scale indicates a signal of reference in the order of  $150\mu V$ . Channels with other units will be rescaled according to their type (see the ‘Scales’ pull-down menu (Fig. 3-K), while the ‘Scale’ value (Fig. 3-B2) will act on the signal of all channels at once. For example if you fixed the scale at  $150\mu V$ , 75 will appear between label channels and the scale of each EEG channel goes from  $-75\mu V$  to  $+75\mu V$ . The ‘Number of displayed channels’ (Fig. 3-B3) fixes the number of channels per screen (to see the other channels, scroll up or down).

The ‘Reference’ panel (Fig. 3-B4) allows you to re-reference the data to a chosen reference. For the EEG channels, this reference may be any other channels (even if you did not select it in the channel selection interface), the mean of all the EEG channels or the mean of the two mastoids (called M1 and M2). For the EOG and EMG channels you have an additional ‘Bipolar’ choice for the reference. Pairs of EOG (and EMG) channels are usually designed to work as a bipolar montage, let’s say that EMG1 and EMG2 are specified to be referenced to each other. If you only display EMG1 and choose the bipolar reference, then EMG1 will be displayed referenced to EMG2. Note that this re-referencing relies on the channel names and is performed only ‘on-display’ (the original data stored on disk are left untouched).

The ‘Filter’ panel (Fig. 3-B5) allows you to bandpass filter the various types of channels differently. The cut-off frequencies are indicated in the boxes next to the channel type. Like the re-referencing, this filtering is performed only ‘on-display’ and the original data stored on disk are left untouched.

The ‘Save Position’ button (Fig. 3-F) allows you to store the time your time cursor is currently at and, when opening the same file later, your time cursor will be replaced at that position. Useful when reviewing waves or when scoring very long files. The ‘Score Sleep’ button (Fig. 3-G) allows you to launch the scoring utility to score the file you are currently displaying. The ‘View another

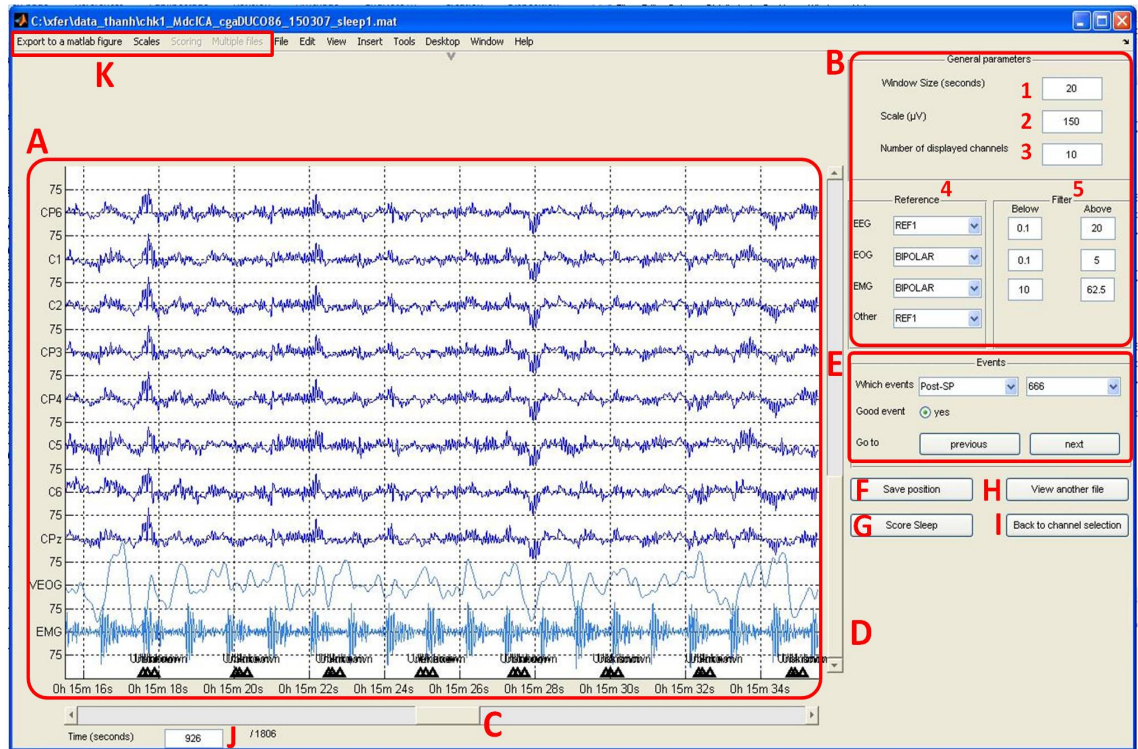


Figure 3: Channels display window: A. main display, B. display options (see text for details), C. time scrolling bar, D. go to sleep scoring, E. view another file, F. back to channel selection, G. time in seconds at the beginning of display, H. export display to a MATLAB figure.

file' button (Fig. 3-H) allows you to select another file to display and, eventually, the 'Back to channel selection' button (Fig. 3-I) lets you go back to the channel selection interface.

The box below the time-slider shows the current time (Fig. 3-J), in the middle of the display window, expressed in seconds, from the beginning of the recording; the other number is the overall length of the recording also expressed in seconds. You can type in a value to jump directly to any moment of interest in the recording.

Events comprised in the data set are displayed as small triangles on the display. The 'events' panel (Fig. 3-E) allows to scroll through these events, by first choosing which events type and value to display (respectively the left and right pull-down menus) and then using the 'previous' and 'next' buttons to move to the nearest previous and next event. The events types and values are those saved in the `events` field of the data.

For the case where waves (e.g. slow waves or sleep spindles) were automatically detected (see section 4.4), some specific codes are used:

- for spindles, the event type would be 'spindles', 'Post-SP' or 'Ant-SP', with code values of 777, 666 or 555 respectively. 'spindles' are marked when no wavelet analysis is used during detection, otherwise they are differentiated into posterior or anterior spindles ('Post-SP' and 'Ant-SP' respectively).
- For slow waves, the event types are either 'SW' (codes starting with 3) or 'delta' (code starting with 4).

The ‘events number’ option from the right pull-down menu lets you display the values of the selected events instead of their type<sup>11</sup>. Using the ‘good event’ radio button, individual events can be marked as good (by default, value 1) or bad (0). Events marked as ‘bad’ appear in red on the display. The good/bad values of all events are saved in a new ‘goodevents’ field specific to **FAST** in the **meeg** object. For spindles and slow waves particularly, the values are also saved in the ‘spindles’ and ‘SW’ fields, created when detecting the waves. Note that any change in the event numbering or any new wave detection will overwrite this information<sup>12</sup>. Finally, when displaying a slow wave, a ‘map of delay’ radio-button will be visible. When checked, a propagation map of the current SW (based on the SW delay) is displayed. The 2D channel location either comes from default CRC channels positions (‘auto’) or the positions of your channels (‘file’, then select your file).

A series of pull-down menus are available in the upper part of the main display window (Fig. 3-K). There is a button allowing you to export the current plot to a new MATLAB figure. This may be handy if you plan to save the displayed signal as an image file. Next to this menu, a ‘Scales’ menu is also available, letting you enter directly the scale of certain types of channels: defined as  $10^x$ , with  $x$  representing the factor of your signal of interest compared to your original data. For example, if you have EEG channels recorded in Volts (V), the scale should be  $10^{-6}$ , since the signal of interest is in  $\mu V$ .

Finally you can compute the power spectrum on one channel of the signal you are currently seeing. To do so, right click next to the signal of interest in the main window, select ‘Compute power spectrum on this channel’ and a new MATLAB figure will appear (Fig. 4) showing you the power spectrum of the signal on the window displayed and on the designated channel.

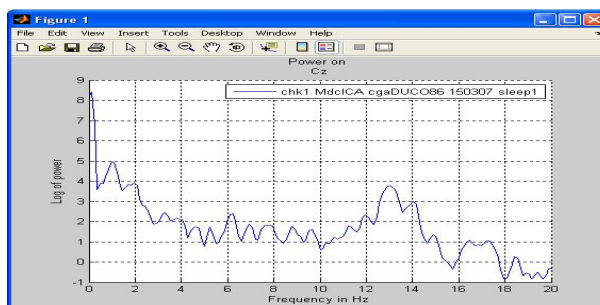


Figure 4: Power spectrum of the signal displayed from the designated channel.

#### 4.1.2 Comparing multiples M/EEG data files

This tool is designed to display the same channel from multiple M/EEG files. It can obviously be used to compare the results of different artefact correction methods applied on the same data set or to visually check the effect of a series of “filtering” steps applied on a data set (Fig. 5). This routine is the same as the main display, except that some functionalities are hidden and others added. Instead of displaying multiple channels from one file, it will display one common channel from multiple files. The ‘Multiple files’ pulldown menu is enabled, which allows you to display the filenames, change the selected channel to display, and exit the display.

When the tool is launched, you simply select multiple EEG files. The program will automatically compare the channels recorded in each file. The channels common to all files will be

<sup>11</sup>For example, this is useful when you’re dealing with EEG-fMRI data and scanner triggers are numbered, then you display scan numbers along the EEG signal

<sup>12</sup>Therefore, we would advise to perform this event marking after all analyses, especially wave detections, are done.



available for display. Only one channel can be displayed at one time. The displayed channel can be modified via the ‘Multiple files’ menu.

The routine also checks the beginning time of each recording (if available) and aligns the different EEG time series in consequence. If the beginning time was not imported from the raw data (or was manually deleted at some point), it is assumed that all files begin at the same time: their first time bin. Otherwise the “true” beginning time of each file is used: if for some time window, some signal is missing from one data set (e.g., you chunked out the 1st few seconds of a data set before filtering it) then that data channel is left empty.

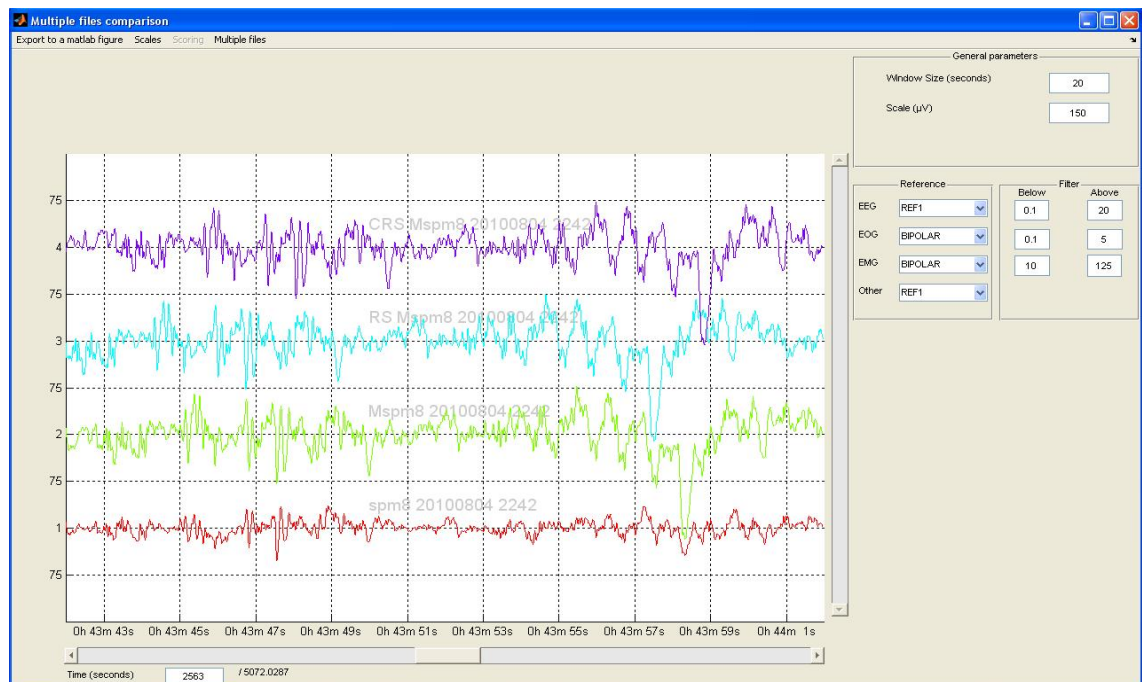


Figure 5: Multiple files comparison GUI, with some features from the main display GUI and others specific to this data comparison GUI.

Contrary to the colour scheme of the main display tool, here the different files are simply displayed in different colours to differentiate them. If you right-click on the displayed signal, you may display the power spectrum of the displayed channel for one or all of the signals currently displayed.

#### 4.1.3 Appending M/EEG data files

Sometimes during long recording session, the operator may need to stop the recording to fix some problem, or recording may be accidentally interrupted. So for one “recording session”, the user ends up with two (or sometimes more...) EEG files. This tool is designed to append those separate EEG files into a single one.

The routine works with two files at a time. So when you have more than two files, you should append the first two files which creates a new file that you append with the third, and so on:

- If the ‘real world’ recording time of each file is available, the routine first checks the ‘real world’ beginning time of each file, and automatically determines the file order. Then it fills the gap between the end of the first data set and the beginning of the second by zero’s, and eventually appends the second the file.

- If no 'real worl' time information is available in the file (after using the SPM8/FT conversion for example), the routine will ask if you want to operate a 'blind concatenation'. In this case, the routine supposes that there is no missing samples between the consecutive file and that you entered the files in the correct order.

In both cases, the resulting file has the same name as the first file selected prefixed with `cc_`.

#### 4.1.4 Chunking a time window from an M/EEG data file

This tool allows the user to cut out an episode out of a large EEG file and save it as a different EEG file. This can be useful if one wants to study specific activity such as sleep stages, epileptic discharge,... The beginning and end of the new file can be defined by a marker or by time (Fig. 6).

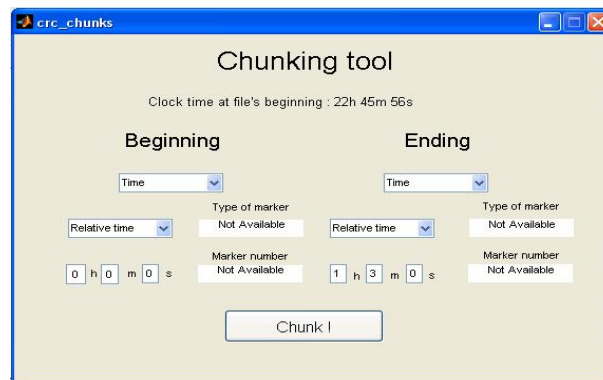


Figure 6: Chunking tool GUI.

If the user chooses the time option, he will have to choose between “Relative time” (time from the beginning of the file) and “Clock time” (the real world recording time). If the user chooses marker, he will have first to choose which type of marker to use (each type of marker has a different ‘label’) and next, the index of the marker. For example, one can use the type marker 0 and choose the 1st marker, i.e. the index will be 1.

The end of the file is fixed in the same way. When the desired parameters are entered for the beginning and end of the chunk, the user may press the ‘Chunk!’ button to perform the chunking

#### 4.1.5 Computing the spectrogram of one M/EEG data file

This routine computes the spectrogram, using the Welch periodogram method, of the selected file according to parameters specified by the user. The result is saved into a time-frequency data file. Note that this file is **not** in a true SPM8 `meeg` object format: it is saved as a separate `file.array` data format, stored in the `meeg` object structure. Before the computation itself, the data are filtered (bandpass filter between 0.5 and 25 Hz by default, Fig. 7-A). The spectrogram is computed over time windows (of 4s by default, Fig. 7-B) every time steps specified by the user (2 by default, Fig. 7-C).

If the file was scored using the scoring tool (see section 4.3.1), the sections scored as movement time or marked as artefacts are left out of the spectrogram calculation (power is set to zero). If the file was scored by multiple users, the score to be used is selected via the pull-down menu

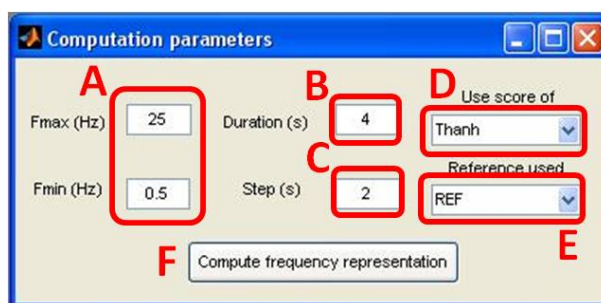


Figure 7: Spectrogram calculation GUI: A. band-pass filtering, B. time window, C. time step, D. scorer selection (if available), E. reference channel, F. launch calculation.

(Fig. 7-D). The artefacted, ‘movement’ or ‘unscorable’ periods are displayed on the spectrogram by blue lines, the width depending on the duration of the arefact, movement/unscorable period. On the frequency band display, the display is just blank at those periods.

The last parameter is the reference taken for the computations (Fig. 7-E). Once all the parameters have been fixed the user may click on the ‘Compute spectrogram’ button (Fig. 7-F). When the computation is complete, a “.frq” file containing the spectrogram is available, and the `file_array` information (as well as the calculation parameters) are stored in the `D.CRC.pwrspect` substructure.

#### 4.1.6 Displaying the spectrogram or power spectrum of an M/EEG data file

This tool is designed to display the spectrogram computed by the ‘Compute spectrogram of one EEG file’ (cfr. section 4.1.5). Two display modes are available (Fig. 8-A): “Spectrogram” and “Frequency band”. “Spectrogram” is a time-frequency representation of one channels, dark blue representing low power and bright-red, higher power. A pull-down menu, Fig. 8-B, is used to select the channel to display.

In the “Frequency band” display mode, the evolution of the power in a specified frequency band (Fig. 9-D) is displayed for one or two channels (channels are picked using 2 pull-down menus, Fig. 9-A). In this display mode, three scaling modes are available (Fig. 9-B): “Absolute value”; “Relative value”, i.e. how much power is dissipated at time  $t$  in the considered band divided by the whole power at time  $t$ ; and the “Mongrain view” (usually used by sleep specialist), it shows the power dissipated at time  $t$  in the selected frequency band divided by the mean power in deep sleep stage during the night. Once the scaling mode chosen, the scale itself can be fixed (Fig. 9-C).

Right clicking on the main plot (either the spectrogram or the frequency band display) lets you define ‘Periods of Interests’ (POI), which will be saved in a new `freq_POI` field in the data structure, as well as in a text file, along the data set. Periods of Interest are defined in the same way as artefacts in the scoring option: the user defines a start and an end points, which relative times are saved in the data set in seconds. The absolute (if available) or relative times in the hh-mm-ss format of all POIs can be displayed in a new figure using the ‘POI info’ option of the right-click menu. POIs can be deleted by right-clicking on the vertical lines defining either their start or end points.

If the file was previously scored (see section 4.3.1), the hypnogram is also displayed (Fig. 9-E). The user may choose which score to use, and compute the mean power spectrum during a specific sleep stage (Fig. 9-F, results are shown in Fig. 10).



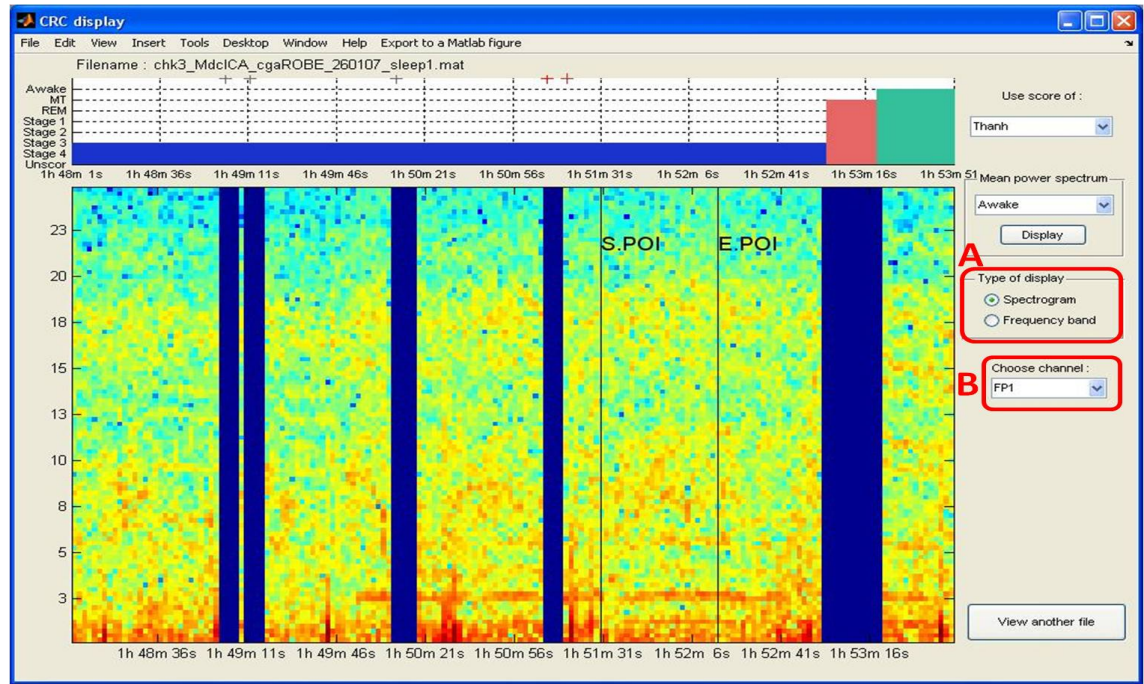


Figure 8: Main display of spectrogram GUI, “spectrogram display”: A. display mode toggle, B. channel selection.

## 4.2 EEG-MRI artefact rejection tools

In this section we describe the fMRI artefact rejection tools. When EEG is recorded in a MR scanner during image acquisition (typically fMRI), two types of artefacts are induced on top of the neural EEG signal:

- The Gradient Artefact (GA) is induced by the gradient switching of the imaging sequence of the MR scanner[1]. Note that this several order of amplitude larger than the genuine EEG signal.
- The Pulse Artefact (PA) is due to the interaction between the *static* field of the MR scanner and the heartbeats[2]. Note that this artefact is thus present even if no fMRI data are acquired.

One should always suppress the gradient artefact before the pulse artefact.

### 4.2.1 Gradient artefact rejection

This tool allows the user to remove the Gradient Artefact (GA) resulting from the gradient switching of the MR acquisition sequence. The removal method is based on the “Average Artefact Subtraction” (AAS) method developed by Allen et al.[1]. AAS estimates the shape of the GA over a TR (the time to acquire a fMRI volume) by averaging the signal over several (typically at least 10) contiguous MR volume acquisitions. This “averaged artefact” is estimated for each TR and subtracted from the recorded EEG signal. The efficiency of the AAS approach relies on the stationarity of the GA picked in the EEG signal. This stationarity can be enforced by synchronizing the clocks of the EEG amplifier(s) with that of the MR scanner. This is a crucial

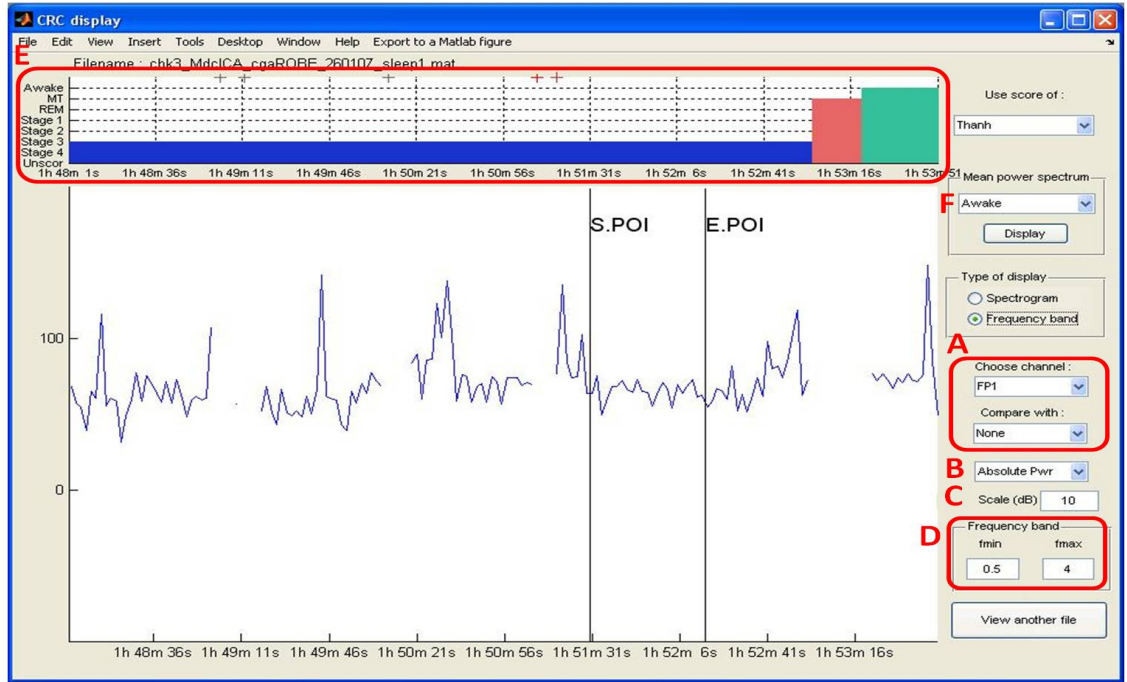


Figure 9: Main display of spectrogram GUI, “frequency band display”: A. channel(s) selection, B. scaling type, C. scale for display.

point and any user applying this algorithm on data acquired without clock synchronization may (and most certainly will) have improper GA rejection.

This tool is quite simple to use: just select the file(s) you want to correct for gradient artefact. For each file selected, a “wait bar” shows you the evolution of the correction and a new data file is (slowly) generated. After corrections your corrected files have the same name as the original ones prefixed with `cga_` and are available for further processing. The parameters of the gradient rejection are located in the `crc_defaults.m` file which you should edit in order to make the program compatible with your specific MR sequence specification and amplifier setup.

There are 2 options to detect the beginning of each scan:

- if you record markers coming directly from the scanners to indicate the beginning of each volume (or slice) acquisition, then set the variable `crc_defaults.gar.UseScanMrk` to 1. You then have to specify the type of marker you are using in `crc_defaults.gar.ScanMrk1` (for example `crc_defaults.gar.ScanMrk1 = 128`<sup>13</sup>). If you use slice markers, i.e. one trigger from the scanner every slice, rather than volume marker, you have to set the `crc_defaults.gar.MrkStep` as the number of slices composing a volume.
- If you do not record the triggers coming from the scanner (slice or volume) to define the beginning of each volume, set the variable `crc_defaults.gar.UseScanMrk` to 0. You then have to specify the exact TR of your sequence in seconds (`crc_defaults.gar.TR` field). You may automatically detect the beginning and end of the scanning episode in the EEG file using an automatic check (field `crc_defaults.gar.Autochk = 1`). You can also manually specify the beginning and end of the scanning sequence (`crc_defaults.gar.beg` and `crc_defaults.gar.nd`).

<sup>13</sup>A secondary marker type can be specified in the field `crc_defaults.gar.ScanMrk2`.

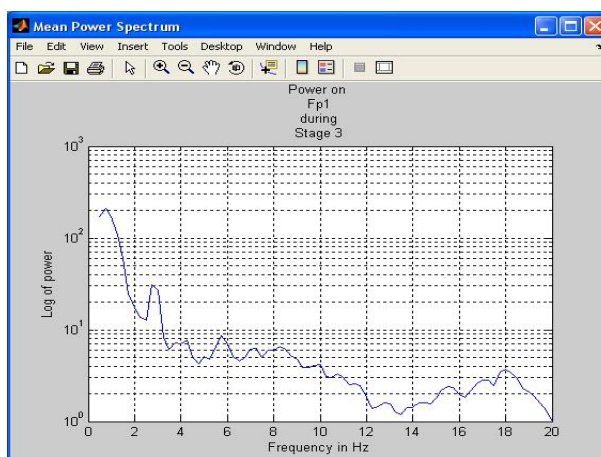


Figure 10: Mean power spectrum during a specific sleep stage (stage 2) for one channel (Oz here).

The sampling frequency of the original file (typically 5kHz) is usually higher than necessary for further processing, you can thus specify the sampling frequency of the output file in `crc_defaults.gar.output_fs` (500Hz by default). The other parameters specify the prefix added to the filename for the corrected file (`crc_defaults.gar.prefix`), the number of scan on which the average is performed for artefact rejection (`crc_defaults.gar.Nsc_aver`) and the number of scans skipped at the beginning of the scanning sequence (`crc_defaults.gar.Nsc_skipped_be`). Note that the corrected data saved will start at the beginning of the 1st corrected TR, i.e. any pre-scan data are omitted. If you're interested in the signal prior that point, then you should extract it (see section 4.1.4) from the uncorrected file and possibly append it afterwards (see section 4.1.3).

#### 4.2.2 Pulse artefact rejection

This tool allows the user to reject the “Pulse Artefact” (PA) in the EEG recordings. This artefact is induced by the heartbeats of the subject lying in the static field of the MR scanner and is thus present even if no MR data are acquired.

When the tool is launched, the user has to choose which channels will be corrected. Usually EMG or ECG channels are not selected for correction. This step is **crucial for ICA-based methods**. ICA-based methods are multi-channel methods, i.e. the signal from all the channels are considered together, therefore, if inappropriate (ECG/EMG-like) channels are selected, some EMG information might contaminate EEG data after an ICA-based correction. So you should also ensure that any bad channel(s) are left out from the pulse artefact correction process as they could confuse the ICA-based methods.

Once you have selected the channels to correct for each file, you choose the ECG peak detection method. At the moment the only available method is the one developed by R. K. Niazy and referred to as “the FMRIB plug-in for EEGLAB”, provided by the University of Oxford Centre for Functional MRI of the Brain (FMRIB) [18]. We found it very robust even on relatively noisy ECG channels.

After this step, you can choose the type of pulse artefact rejection method you will use. There are five of them: “PCA” (FMRIB plug-in), “Gaussian mean” (AAS from FMRIB plug-in) [17, 6], “constrained ICA (automatic)”, “constrained ICA (manual)” [10] and “AAS & PCA combined”

(based on the FMRIB plug-in)<sup>14</sup>. The last step lets you choose which channel number is the ECG. If you leave 0, the algorithm will automatically detect it.

**Constrained-ICA method.** Be aware that long episodes with movement artefacts can compromise cICA correction. Indeed cICA works on one time-window (90 seconds by default) at a time and episodes showing “movement activity” are not used for the correction estimate (see parameter `crc_defaults.par.bcgreml.scSNR` here under). So if there is proportionally too much movement during a time-window, not enough “good” signal will be available and cICA will not be able to properly decompose the recorded signal and reject the pulse artefact. If you see messages in MATLAB command window such as

```
Rejecting more than 50% of time bins from current time window!
```

```
Warning: Algorithm fails to correct the EEG recordings, probably due to'
too much artefacts.
```

You should worry about your data: either the gradient artefact rejection did not work so well and your data are still too noisy, or there are some ‘bad’ channels and you should exclude them from the cICA corrections.

Since a separate cICA correction matrix is estimated and saved for every 90 second time window, it is possible to replace ‘bad’ ones by a better one. The selection of the optimal cICA correction matrix can be automatic (option “constrained ICA (automatic)”) or manual (option “constrained ICA (manual)”):

- With the automatic approach a ‘mutual information’<sup>[21]</sup> criteria is used to decide which cICA correction matrix to use: Let’s define 2 mutual information measures  $MI_1$  and  $MI_2$  like this:  $MI_1 = MI(\text{original data, AAS artefact template})$  and  $MI_2 = MI(\text{corrected data, AAS artefact template})$ . Then the cICA correction matrix leading to the smallest ratio  $MI_2/MI_1$  is deemed optimal. The MI measures are estimated from random samples of data in the data set, as specified in `crc_defaults.par.useinitseg`, `.additiosegl` and `.length`. Then the corrected data are saved on disk as a new data set.
- With the manual approach, no corrected data are saved on disk and the user must decide which correction matrix(-ces) to employ. This selection is performed via the usual display function (see section 4.1.1) and the user can directly visualize the corrected signal obtained by any saved cICA correction matrix. Once the cICA correction matrix is selected, the corrected data are generated and saved on disk in a new data file.

**AAS, PCA and other methods.** We would advise users with 30 channels or more to choose a “constrained ICA” (cICA) method. cICA was shown to be more efficient than AAS and PCA at rejecting the PA and to better preserve the spectrum of the “true” EEG signal <sup>[10]</sup>. This is particularly important when analyzing the time course of spontaneous activity (such as in sleep studies).

With fewer channels (<30) or lots of movement activity, single-channel methods are better suited. “Gaussian mean method” (AAS) is often preferable but PCA can also do a good job. The “AAS & PCA combined” method is experimental and has not been rigorously tested. During

<sup>14</sup>a “multi ICA” approach <sup>[22]</sup> is also available but not through the GUI as it takes far too much time to calculate and has not been extensively validated

the preparation of [10], we noticed that AAS was more efficient for the lower part of the data spectrum and PCA for the higher part<sup>15</sup>. “AAS & PCA combined” thus uses a combination of AAS and PCA: AAS is applied on the low-pass filtered (<4Hz) signal and PCA on the high-pass filtered (>4Hz) signal, then the 2 corrected parts are recombined afterwards.

**Specific options.** Users are in effect advised to test test correction methods and find out the “best” one for their own data. In `crc_defaults.par.bcgrem`, you can change some parameters linked to the PA rejection, but most of these concern the constrained ICA method such as

`.size` the window size (in seconds) on which cICA is performed;

`.step` the step between the beginning of two windows;

`.scSNR` the number of standard deviation signal must deviate from to be considered as “noise” or “movement”, i.e. to be excluded from the cICA estimation;

`.Refnames` the channels representative of the shape of the pulse artefact;

`.NitKmeans` the number of iteration of the K-means step.

### 4.3 Sleep specific tools

These tools and GUI's were specifically developed for sleep data analysis, like the manual scoring of sleep M/EEG (section 4.3.1). The power spectrum of scored data can be estimated and displayed. Finally some summary statistics, about sleeping time, sleep stage duration, etc., can also be calculated afterwards. The spectral power through some sleep stages can also be estimated (section 4.3.2).

You can also automatically detect some sleep patterns in your file. This version of FAST deals with ‘Slow Waves’ (section 4.4.1) and ‘Sleep Spindles’ (section 4.4.2)<sup>16</sup>.

#### 4.3.1 Manual sleep scoring

Once the user has selected the file to score, the selection channel interface appears (same interface as in section 4.1.1). Once you have selected your favourite channels, hit the ‘Score’ button.

The main display then appears, with all scoring related tools enabled. In particular, a ‘Scoring’ pulldown menu is now visible on the top left of the window. With this menu, you are able to choose the scorer or define another scorer, compare scores, import scores or display grids (vertical every second or horizontal at half of the main scale). Since this is a scoring tool the size of the displayed time-window is fixed and cannot be changed once it is chosen.

If the file is scored for the first time, the program asks a username and a window size. The same file may be scored by different users. To introduce a new scorer, the user has to select “New scorer” at the bottom of the scorers option of the ‘Scoring’ menu.

<sup>15</sup>In particular, slow waves were sometimes removed by the PCA methods

<sup>16</sup>this latter part of the code is in beta version.



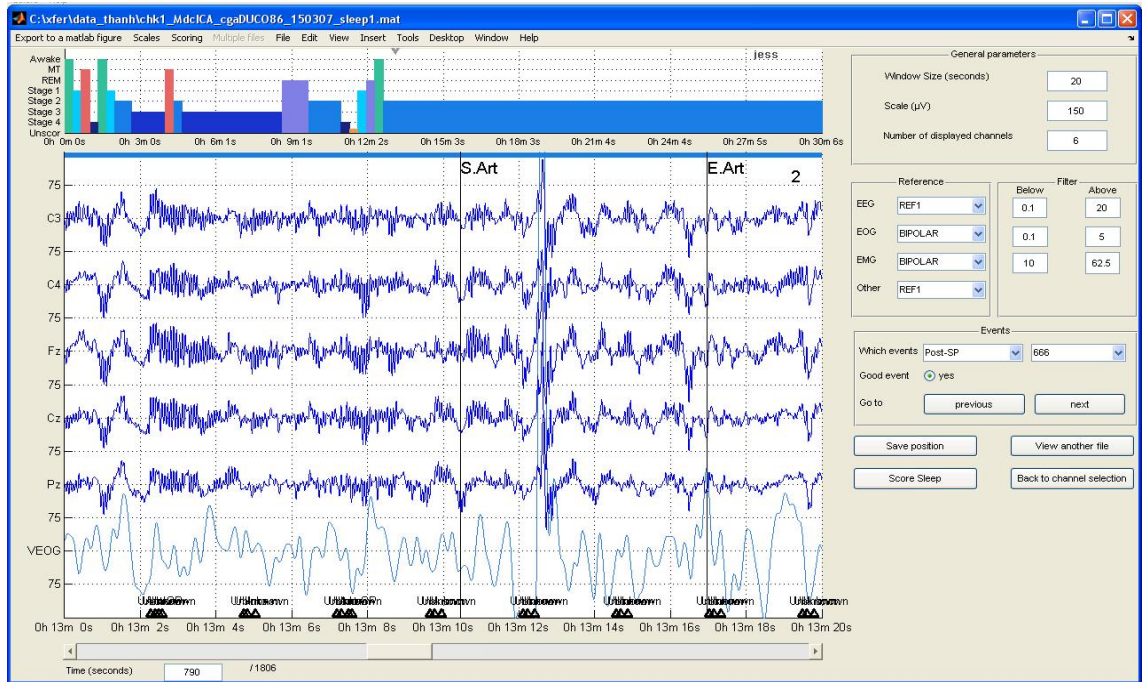


Figure 11: Sleep scoring GUI, with the hypnogram display on top of the the EEG signal.

The keypad is used to assign a score to the current window. Each number corresponds to a specific stage<sup>17</sup>. By default these are set as: “Wakefulness” (0), “Sleep stage 1-4” (1-4 respectively), “REM sleep” (5), “Movement Time” (6), and “Unscorable” (7). Each time a score is assigned to the current window, the display moves on to the next time-window. The letters “b” and “f” can be used to move backward and forward by one time window without assigning any score to the current window. During scoring a hypnogram is automatically constructed and a tick mark shows the current position in the signal. Note that to score a window the main display window has to be selected (i.e. clicked in).

The right-click menu allows the user to add 4 types of events:

- “artefact & arousal”, which will be considered as artefact for power spectrum computation, cfr. section 4.1.5. Artefacts can be defined in two ways: either as ‘unspecified’ artefacts or as ‘specified’ artefacts. This last option allows to choose a label for the manually defined artefacts, amongst common categories of artefacts (i.e. blink, eye movement, movement). When selecting an unspecified artefact, the label associated to that artefact will be left empty. It is therefore possible to define specified and unspecified artefacts in the same score<sup>18</sup>. When defining an artefact (or arousal resp.), the background color of the plot will be changed from white to light blue (light red resp.) to avoid opening an artefact (arousal resp.) without closing it.
- “event of interest”. This generic event can be used in different ways depending the interest of the user: for example to manually mark spindles, epileptic discharge,...
- “FPL marker”, i.e. “closing door & light” marker (in French “Fermer Porte & Lumière”), to indicate the beginning of the night of recording.
- “OPL marker”, i.e. “opening door & light” marker (in French “Ouvrir Porte & Lumière”), to indicate the end of the night of recording.

<sup>17</sup>The labels can be modified in the default file but NOT their number, which is hard coded in some routines.

<sup>18</sup>Labels can then be found in the 8th row of the score field created in the data set.

The latter two, FPL and OPL markers are important to compute sleep statistics. Artefacts, arousals and events of interest are marked as crosses (respectively grey, red and dark red crosses) on the hypnogram to allow a global visualisation of them along the night.

### 4.3.2 Spectral power calculation and statistics

The last item of the right-click pulldown menu is the compute power spectrum tool, same as in section 4.1.1. These tool are exactly the same as those described in sections 4.1.5 and 4.1.6. If the file was scored using the sleep scoring function, the sections scored as movement time or marked as artefacts are left out of the spectrogram calculation (power is set to zero). If the file was scored by multiple scorers, the user has to decide which score to use (Fig. 7-D).

The hypnogram is also displayed (Fig. 9-E) alongside the spectrogram display. The user may choose which score to use, and compute the mean power spectrum during a specific sleep stage (Fig. 9, results are shown in Fig. 10).

Finally the ‘Compute Statistics’ button is accessible via the ‘Scoring’ pull-down menu: the proportion of the different sleep stages, of the movement time and of the wakefulness stage, the latency of REM sleep, etc. are then displayed (Fig. 12).

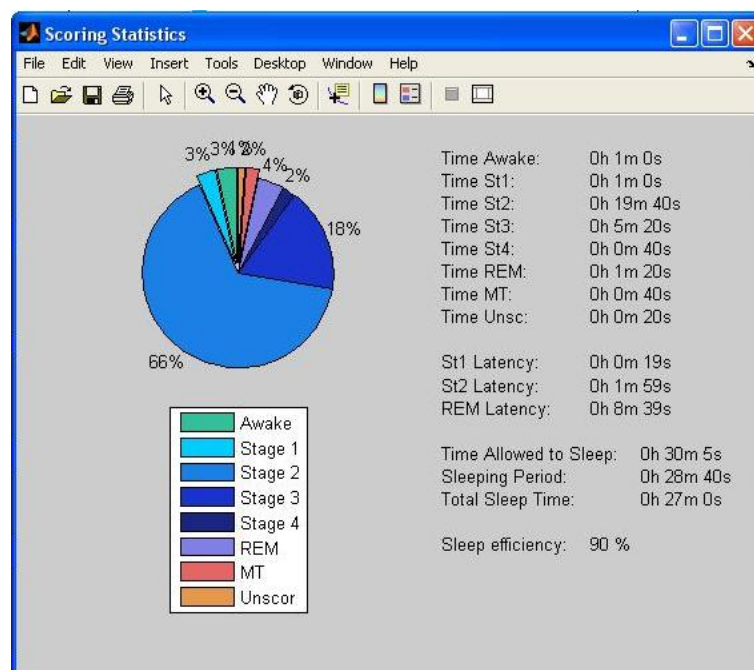


Figure 12: Sleep statistics results.

## 4.4 Automatic wave detection tools

Automatically extrating features from continuous EEG recording is a complicated task mainly because of the poor signal-to-noise ratio and the variability (within and between subjects) of the “waves” to detect. We focused on two types of spontaneous sleep waves, namely “slow waves” and “spindles”. The methods are described in the next two sections. Still, when using these tools, keep in mind that wave detection is not “bullet proof” and paramter may need to be tweaked to achieve the best results on your data.

#### 4.4.1 Slow Wave detection

This tool aims at detecting Slow Waves (SWs) in a EEG recording. SWs detection is performed on the averaged ROI signals (i.e. over 4 averaged signals in the case of automatic selection of ROI, see here under). The detection was designed in a spatio-temporal way following Massimini's criteria [14] but were adapted according to our observations on different data sets (Fig. 13):

1. A negative zero crossing (down-zero crossing) and a subsequent positive zero crossing (up-zero crossing) separated by 0.25-1.25sec,
2. A negative peak between the two zero crossings with voltage  $< -80\mu\text{V}$ ,
3. A negative-to-positive peak-to-peak amplitude  $> 140\mu\text{V}$ ,
4. A positive slope  $> 90\%$  of the maximum slope,
5. A positive zero crossing and a subsequent positive peak separated by maximum 2 sec.

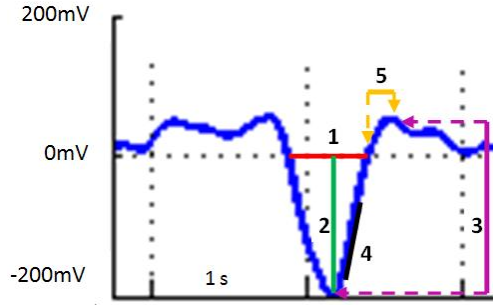


Figure 13: SW detection criteria: (1) A negative zero crossing (down-zero crossing) and a subsequent positive zero crossing (up-zero crossing) separated by 0.25-1.25sec, (2) A negative peak between the two zero crossings with voltage  $< -80\mu\text{V}$ , (3) A negative-to-positive peak-to-peak amplitude  $> 140\mu\text{V}$ , (4) A positive slope  $> 90\%$  of the maximum slope, (5) A positive zero crossing and a subsequent positive peak separated by maximum 2 sec.

Each detected SW is used to build a “SW trajectory” over the scalp, using all the channels. Channels presenting a SW are detected and a “delay value” is estimated for each of them, based on the difference between the negative peak time at this electrode and the negative peak time at the first electrode detecting the specific SW. A code is also attributed to each wave in function of its type (delta or SW). No user participation is required.

The detection procedure relies on the following steps:

1. Extraction of the data within a time window of interest in the provided file,
2. Filtering between 0.2 and 4 Hz of the extracted data,
3. Creation of Regions of Interest (ROI),
4. preliminary SWs detection on the signals averaged over ROI,
5. Detection of each SW on all the channels.



Once the SWs are detected, the trajectory of the peak negativity over the scalp surface is also extracted using the information from all channels (SW detection and channel location). Finally the SWs are displayed.

The characteristics of the detected waves are saved in a new field `D.CRC.SW` of the original data structures<sup>19</sup>. Their most important characteristics are also saved in the field `D.events` to allow an easy epoching of the waves. This new field `D.CRC.SW` is also saved in the chunked and SW files if the option in ‘Analyse’ was not ‘all file’. In that case, the time characteristics of the waves will be relative to the start of the input file in its SW field while they will be relative to the start of the chunked or SW file in theirs. This allows using either the input file or the chunked/SW file for SW analysis, according to what suits your needs best.

After clicking on ‘SWS detection’ in the main **FAST** window, a GUI appears and lets you specify the parameters of the detection procedure (Fig. 14). This interface is divided into four main parts: the EEG file selection (on top), analysis parameters (left), review parameters (middle right) and parameters specific to joint EEG-fMRI data (bottom right).

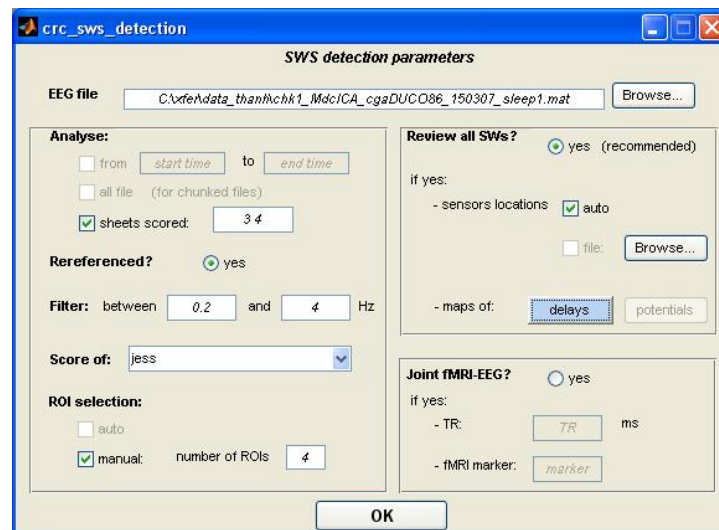


Figure 14: User interface allowing for the specification of the different steps parameters

**Analysis parameters.** These are the parameters of the detection procedure (time window, filtering, etc.), the thresholds used for the detection itself are defined in the `cwc_defaults.m` file.

- **Time window selection.** This step allows extracting data of interest within the provided file. Three options are available:

`from ... to ....` chunks the file according to the user specified times. If the file contains time information (typically if it was imported using **FAST**), the ‘start time’ and ‘end time’ fields should be specified in real-world time (for example, [23 45 02] for ‘23h 45m 2s’). Otherwise, they should be considered as relative times, 0h 0m 0s being the start of the input file. In all cases, the correct format for those entries is: hh-mm-ss.

`all file` for already chunked files or small files containing only Slow Wave Sleep (SWS) stages. The SWs detection is performed directly on this file, without the creation of any other file.

<sup>19</sup>For more information on this field content, type `help cwc_SWS_detect` at the Matlab prompt.

**sheets scored** builds a new file from the concatenation of all sheets scored ‘scores’ (specified by spaced numbers, e.g. 3 4). This new file is stored on the drive under the name **SW\_filename**. This option is particularly useful if there is no specific moment of interest in the file: if the input file is a whole night recording and by specifying the scores as 3 and 4, the SWs detection will be performed on all SWS stages.

Note that, in all cases, periods marked as artefacted (artefacts or arousals) or as movement time or unscorable will be skipped from the analysis.

- **Re-reference parameters.** The algorithm checks for a re-referencing operation in the field **D.history** (modified after each SPM8 processing). If it does not find such an operation, the user is asked if he wants to continue or not. Please, apply your montage before any processing of the data.
- **Filter parameters.** If the file wasn’t filtered in the frequency band of interest, the procedure will band-pass filter the data (by default, 0.2-4Hz) by applying first a 0.2Hz highpass filter and then a lowpass filter with cutoff frequency at 4Hz to ensure stability. No files are created from this operation. If such filtered files are useful for you, you should preferably first filter the data in SPM8 and specify the newly created file as input. The filters used are the same as those of SPM8 (Butterworth type) <sup>20</sup>.
- **Scorer Pull down menu.** To choose the scorer’s score to use.
- **ROI selection parameters.** To decrease the computational load, SWs detection is first performed on averaged signals over (typically) 4 ROIs: frontal, central left, central right and parietal. Two options are available for the channel selection:

**automatic** Channels in a radius (below about 15% of scalp radius) around the 2D normalized flattened positions of respectively Fz, C3, C4 and Pz channels in the extended 10-20 system<sup>21</sup>;

**manual** You need to specify the number of ROI desired and their names. The ROI specification is then performed for each ROI in order. This manual selection is based on the channel selection GUI (see section 4.1.1 and Fig. 15). Already hand-built ROI selections can be loaded and saved via this GUI.

In case your file has not been previously prepared in SPM8 and for an optimal electrode selection (manual or automatic), please edit the **crc\_electrodes.mat** via the matlab script **crc\_electrodes.m** if you have other electrodes systems than BrainProducts or EGI systems.

**Display parameters.** When answering ‘yes’ to the ‘Review all SWs?’ question, the main display appears after the end of the detection. The user has to choose the channels to display and select either ‘SW’ or ‘delta’ types in the ‘Events’ panel to review the detected waves. In case no other events were present in the data set, ‘All’ can also be selected and will show both types of slow waves. If the ‘delay map’ was selected previous to the wave detection, a delay map will appear in the bottom right corner of the display in an inverted hot colormap (Fig. 16).

To compute the interpolated values, the algorithm needs the 3D position of the electrodes. This is specified by ticking the ‘file’ option under the ‘sensors locations’ parameters. However an ‘auto’ option is available, which will compute a 2D interpolation of the signal based on the 2D sensors positions. Once again, the algorithm will check if the coordinates were assigned to the

<sup>20</sup>For data sets with a relatively high sampling rate (up to 500 Hz), the user should enlarge the frequency bands to ensure the stability of the filters. Another parameter which can be modified is the order of the filter. For the Butterworth filters used in this toolbox, the default value is 4. However, decreasing this order leads to an increased stability. If the user wants to keep the defaults settings, he should downsample his file.

<sup>21</sup>If the file was previously prepared within SPM8, the algorithm will select the coordinates saved in the data file. Otherwise, theoretical positions will be used (**crc\_electrodes.mat**).

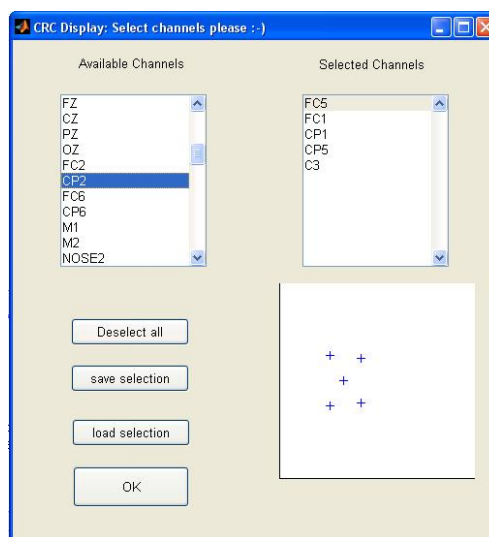


Figure 15: User interface allowing for a manual selection of ROI channels

channels (`spm_eeg_prepare`) or not. In the latter case, to obtain a good interpolation, please edit the `crc_electrodes.mat` via the matlab script `crc_electrodes.m` or SPM8 GUI, if you have other electrodes systems than BrainProducts or EGI systems. Please note that, for the moment, the ‘auto’ option is only available if you display maps of delays.

**Joint EEG-fMRI acquisitions.** Due to the growing use of combined EEG-fMRI techniques, the toolbox can compute the main SW characteristic in units of TR, if the EEG recordings has an identifiable event (marked with ‘marker’) indicating the start of the MRI recording. Therefore, by specifying the TR and this marker, the value of the negative peak maximum power can be computed in terms of TR. This information is stored in the SW structure contained in the `D.CRC.SW` field under the name `negmax_TR`. Please note that this computation only takes into account a shift between the start of the EEG recording and the beginning of the scan. If the relationship between the timing of your EEG and fMRI recordings is more complex, use this tool carefully.

Please note that when repeating a SW detection on the same file, any event with a ‘SW’ or ‘delta’ type will be erased to avoid redundancy. The field `goodevents` will also be reset since the number of events was changed.

#### 4.4.2 Sleep Spindles detection (*BETA version!*)

The spindle detection algorithm was derived from the method described by M. Mölle et al. [16]. To match empirical results obtained with different data sets, some of the detection parameters were however adjusted.

To reduce computational effort, the detection is performed on 3 ‘reference’ channels (i.e. Fz, Cz and Pz) only; these channels are selected according to their scalp position (the channels being closest to the theoretical positions of the 3 channels of reference will be selected). The 2D positions considered for channels are the coordinates saved by the SPM8 preparation step. If this step was not performed, the positions will be retrieved from the `crc_electrodes.mat`. In this last case, you must ensure that the system used to acquire the data is supported by this file (otherwise, you need to edit then run the file `crc_electrodes.m`). The algorithm will then check

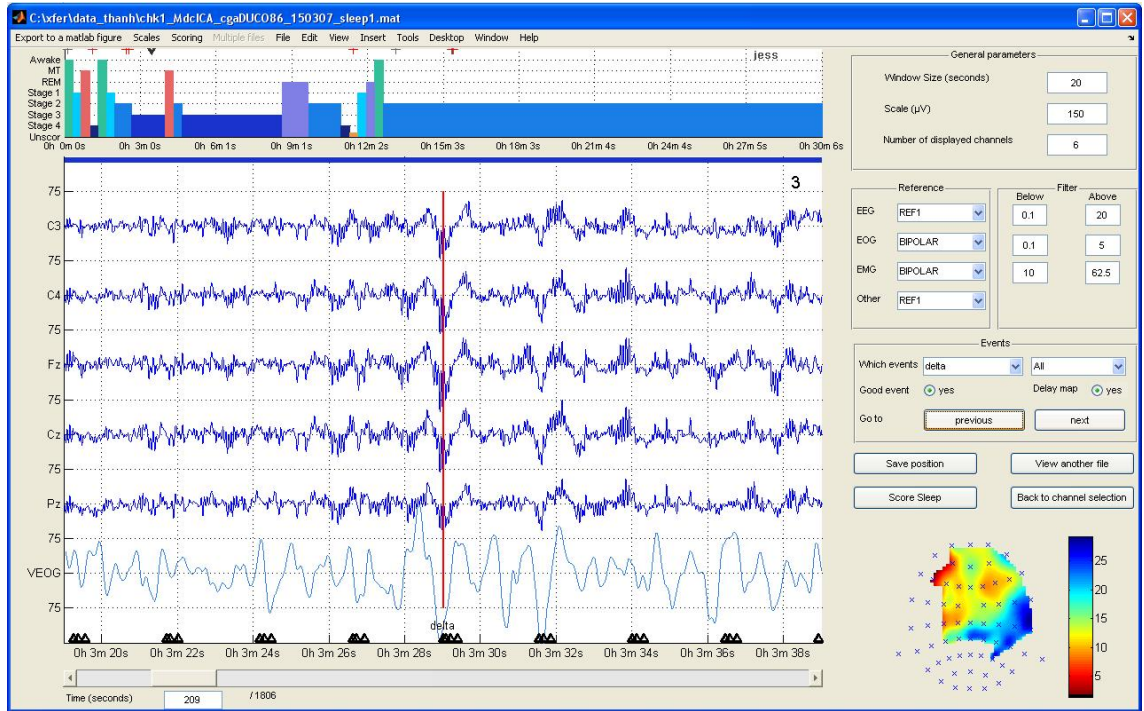


Figure 16: Sleep scoring display with a colour-coded delay map (scale is in ms) in lower right corner.

that none of these channels are bad. If this condition is not fulfilled, the detection is interrupted. For the moment, the only way to perform spindle detection with one of the reference channels being ‘bad’ is to change the channels of interest for that particular file in the file `crc.defaults.m`.

The signal from the reference channels is extracted, discarding all artefacted/arousals, movement or ‘unscorable’ periods, and filtered (user-defined frequency band, Butterworth bandpass filter). According to [16], only the largest spindles are to be considered and, therefore, thresholds have to be computed. In our implementation, the thresholds are computed as the percentile 95 of the extracted signal of the reference channels scored as sleep stage 2 (and only this stage). Three thresholds are obtained, one for each reference channel. These thresholds are displayed in the workspace, and, as in [16], are usually within  $5\mu\text{V}$  of each other.

Example of workspace display:

```
.....
..... Computing RMS and detecting spindles .....
.....
Threshold of detection for channel Fz : 13.9959
Threshold of detection for channel Cz : 12.7297
Threshold of detection for channel Pz : 11.859
```

A root mean square (RMS) signal is then computed from the previously extracted signals (all user defined stages or time period), using a rectangular window of 0.1 second and a time resolution of 0.05 second. A binary signal is then computed: whenever the RMS signal is larger than the previously computed corresponding threshold, the value is 1, otherwise it is 0 (Fig. 17). The average of the 3 binary signals is then computed and ceiled (i.e. whenever there was a 1 on one binary signal, there is a one on the ceiled signal). Using this ceiled average, one can detect bursts

of activity in the frequency band of interest lasting at least 400ms and separated by at least 1000ms, which are then considered as spindles. This last criteria was added to those of M. Mölle et al. [16], according to our experimental results, to reduce of false positive detection.

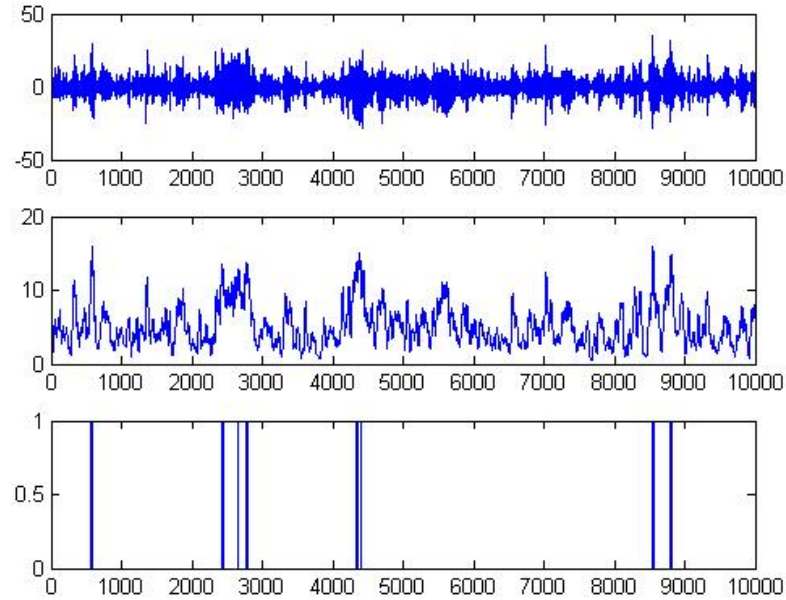


Figure 17: Illustration of the spindle detection using the extracted (top plot), RMS (middle plot) and binary (bottom plot) signals on Cz, the threshold being  $12.7297\mu\text{V}$  as computed previously.

After selecting of the ‘spindles’ detection in the wave detection tool, the user has to define the different parameters of the spindles detection (Fig. 18).

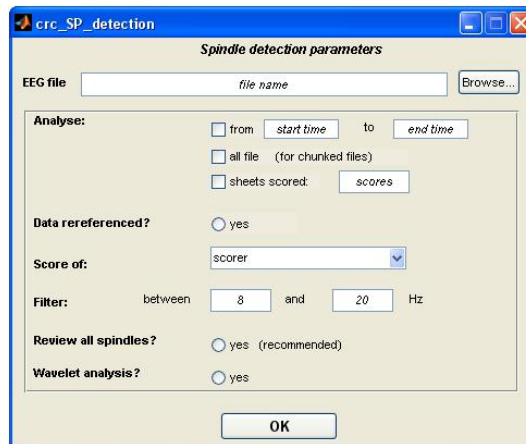


Figure 18: Graphical user interface allowing to define the parameters of the spindle detection.

- **EEG file.** To select the input `.mat` file corresponding to the data set to analyze. The name of the file can either be typed in manually or browsed.
- **Analyze.** Options similar to those of the SW detection (see previous section).



- **Data rereferenced?** To check if the file was already rereferenced.
- **Score of.** To choose the scorer. Please note that this step determines the time periods used since artefacts, arousals, movement or unscorable periods will be withdrawn from further analysis.
- **Filter.** Choose the filter to apply. Different tests on data sets showed that the 8-20 band was optimal for spindles detection. Therefore, if you want to reduce that band (for example to 12-15 Hz), please note that filter ripples can affect the signal.
- **Review all spindles?** Check if you want to review the spindles after detection. This option launches the main display after the detection. After choosing the channels to display (reference channels are preselected), go to the ‘Events’ panel and choose to display ‘SP’ types (‘Ant\_SP’ or ‘Post\_SP’ if the next option is selected) and ‘All’ values to review the detected spindles (Fig. 19). Please note that some spindles are not visible when Slow Wave activity is displayed simultaneously. Therefore, changing the filters might be useful (one possibility might be to choose the frequency band used to detect the spindles).

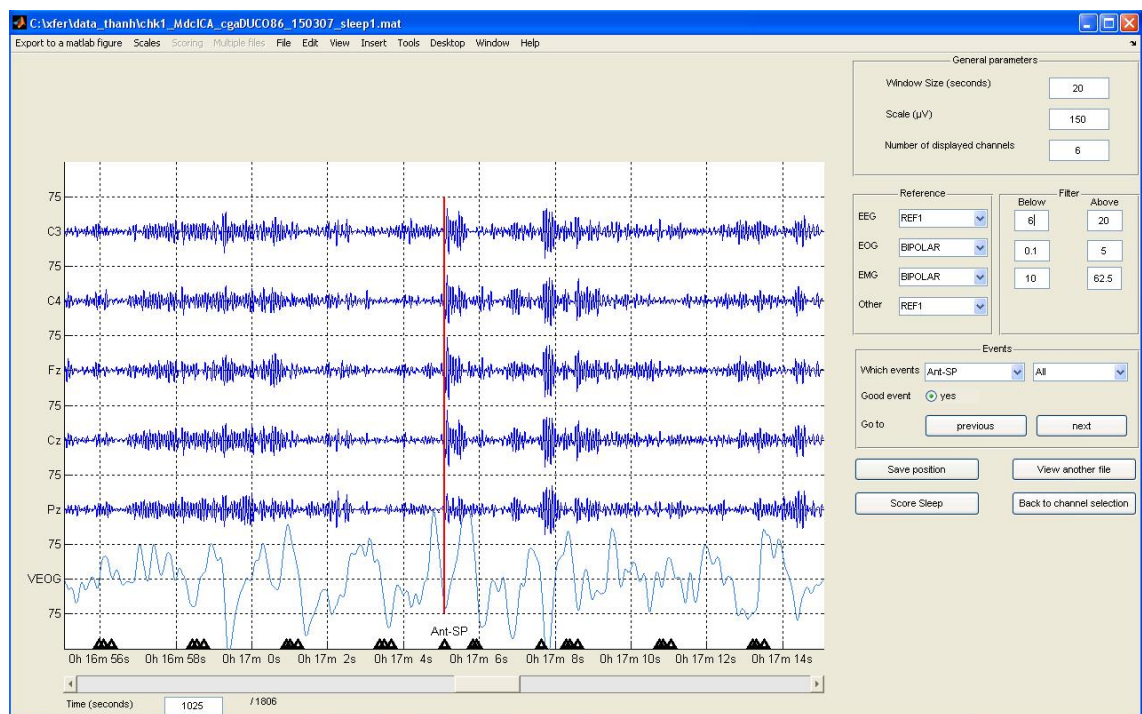


Figure 19: Review of detected spindles (here anterior spindles) on the reference channels.

- **Wavelet analysis?** This option performs a wavelet analysis of the detected spindles (previously epoched around their starting time). The power spectra on the anterior and posterior channels in the band 11-16Hz are then compared and the spindle is then labeled as ‘anterior’ (Ant-SP) or ‘posterior’ (Post-SP) according to the largest power spectrum. The anterior or posterior character of the channels is computed according to their y-coordinate, the reference being Cz (theoretical 2D coordinates: [0.5 0.5]). Therefore, using theoretical 2D coordinates (`crc.electrodes`) or preparing the file using SPM8 can lead to slightly different results as different electrodes could be picked as references.

During detection, different characteristics of the spindles are computed, such as their starting and ending points, their duration, the maximum amplitude and the channel on which it occurred. When the wavelet analysis is selected, the ‘anterior’ or ‘posterior’ character of the spindles is also computed, as well as their frequency. All these parameters are saved in the data set

(`D.CRC.spindles`) and spindles are considered as events (to epoch the spindles if needed, codes 777 for ‘SP’, 555 for ‘Ant-SP’ and 666 for ‘Post-SP’), which can therefore be marked as ‘good’ or ‘bad’ during reviewing.

## 5 Advanced Users

The first part of this section describes the fields added by **FAST** to the standard SPM8 structure/object (.mat file). These additional fields do NOT interfere with SPM8 machinery, they merely contain extra-information used by **FAST** for the time-frequency analysis, sleep scoring and wave detection. The second part is dedicated to the functions not accessible from the main GUI.

### 5.1 Addition to the SPM8 structure

As stated previously **FAST** does not modify the main fields of the SPM8 structure, except when events are added, and all additional structures are placed in separate subfields (`info` and `CRC` at the moment).

#### 5.1.1 The `D.info` field

When **FAST** creates a .mat header from a .vhdr one (BrainProducts Recording System), the time and the date at the beginning of the recording are stored respectively in the `D.info.hour` and `D.info.date` fields. Note that when these informations are available, the time axis of the different display (scoring, display, frequency representation) is expressed in “clock time”, otherwise it is expressed in “relative time” from the beginning of the recording.

#### 5.1.2 The `D.CRC` field

**Spectrogram information** When the spectrogram of a file is computed, several bits of information are added to the structure `C.CRC.spectpwr`:

- `.frqname` contains the file name of the spectrogram data;
- `.frqNsamples` is the number of instantaneous power spectra computed for each channel;
- `.frqNbins` is the number of frequency bins for the spectrogram;
- `.frqdata` is the `file_array` pointing to the spectrogram data, use it as a matrix;
- `.frqscale` is the scale factor for each channel;
- `.frqbins` contains `frqNbins` elements which are the centre of the frequency bins in Hz;
- `.step` is the step (in seconds) between two window used to compute two following power spectra;
- `.duration` is the length (in seconds) of the windows used to compute the power spectra;
- `.frq_POI` contains the time (in seconds) of the user-defined ‘Periods of Interest’.

**Scoring information** When a file has been scored at least one time, a cell array `D.CRC.score` of dimension  $8 \times N$  is added to the structure, where  $N$  is the number of scorer(s). Each column contains the details of one score, from row one to eight:

- `score{1,:}`, a  $1 \times R$  vector containing numbers representing the score, where  $R$  is the number of scoring windows. “Stage” labels are defined in the `crc_defaults.m` file. By default they are set as: 0 correspond to “Wakefulness”, 1-4 to “Sleep stage 1-4” respectively, 5 to “REM sleep”, 6 to “Movement Time”, and 7 to “Unscorable”;
- `score{2,:}`, the name of the scorer;
- `score{3,:}`, the length of a scoring window (in seconds);
- `score{4,:}`, two numbers indicating the time when the light was shut off and door was closed (FPL, first number), and when the light was turned on and the door was opened (OPL, second number). These measures are given in seconds relative to the beginning of the file;
- `score{5,:}`, a  $T \times 3$  array, where  $T$  is the number of artefact episodes marked. The first and the second columns is the beginning and end of each artefactual episode respectively, with all the times expressed in seconds relative to the beginning of the file. The third column indicates the channel index if the artefact is channel specific or 0 if it applies to all channels;
- `score{6,:}`, same as artefact episodes but for arousals;
- `score{7,:}`, same as artefact episodes but for events/episodes of interest;
- `score{8,:}`, a  $T \times 1$  cell array, containing the label of each artefact. The labels for ‘unspecified’ artefacts are left empty.

**SW detection information** Other fields added to the `D.CRC.SW` object for the SW detection:

- `.SW`, a  $1 \times T$  structure array,  $T$  being the number of detected waves, containing all the properties of the waves (type `help crc_sws_detect` for the specification of these fields);
- `.origin_count`, a  $Nel \times 3$  cell array,  $Nel$  being the number of non bad EEG electrodes in the file. This array contains, for each electrode (column 1), the number of waves first detected on that electrode (column 2) and the total number of waves detected on this electrode (column 3).

**Spindle detection information** Other fields added to the `D.CRC.spindles` object for the SP detection:

- `D.spindles` contains different properties of the spindles, each in the  $Nsp \times k$  size,  $Nsp$  being the number of detected spindles and  $k$  depending of the subfield. All fields are explicit, except `maxelectrodes`, which contains the electrode on which the spindle has maximum amplitude.

**Other information** Other fields added to the `D.CRC` structure for display management:

- `.lastdisp`, time in seconds saving the position when clicking on the ‘save position’ button;
- `.goodevents`, a  $1 \times Nev$  vector of binary elements,  $Nev$  being the number of events in the file, as in the `D.events` field. 1 represents events marked as good (default) while 0 represents events marked as ‘bad’ via the display.



## 5.2 “Hidden” functions

`crc_statgen` can only be used on files which were previously sleep-scored and computes statistics such as the percentage of the different sleep stages, the latency of the sleep stages, the efficiency of sleep, the total sleep time, the number of arousal per hour, etc. The function creates a text file, named `Statgen.txt`, with these values and also return a matrix in the MATLAB workspace. The content of this file can be copy-pasted in a cell-formated program such as Excel. If more than one file is selected, then the text file and the matrix will contains as many lines as there are selected files.

## 6 Acknowledgments

This M/EEG toolbox is developed with the financial support of the Fonds de la Recherche Scientifique-FNRS, the Queen Elizabeth fund, and the University of Liège.

The authors would like to thank the CRC staff for their comments and suggestion during the development of FAST. A special thank goes to Virginie Sterpenich and Christina Schmidt whose suggestions were very helpful during the design and the very birth of FAST, and also to Pierre Maquet who imagined FAST before we made it real. We are also grateful to all the users at the CRC (especially Luca Matarazzo, Ariane Foret, Laura Mascetti and Thanh Dang Vu) who helped us debug the code by boldly using the alpha version of the toolbox. Thanks also go to the “foreign labs” who provided some data sets to test the reading functions and the MR-artefact rejection tools.

## References

- [1] Philip J. Allen, Oliver Josephs, and Robert Turner. A method for removing imaging artifact from continuous EEG recorded during functional MRI. *NeuroImage*, 12:230–239, 2000.
- [2] Stefan Debener, Karen J Mullinger, Rami K Niazy, and Richard W Bowtell. Properties of the ballistocardiogram artefact as revealed by EEG recordings at 1.5, 3 and 7 T static magnetic field strength. *Int J Psychophysiol*, 67(3):189–199, March 2008.
- [3] Arnaud Delorme and Scott Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. *Journal of Neuroscience Methods*, 134:9–21, 2004.
- [4] Arnaud Delorme, Tim Mullen, Christian Kothe, Zeynep Akalin Acar, Nima Bigdely-Shamlo, Andrey Vankov, and Scott Makeig. EEGLAB, SIFT, NFT, BCILAB, and ERICA: New Tools for Advanced EEG Processing. *Computational Intelligence and Neuroscience*, 2011:12, 2011.
- [5] Volkmar Glauche. MATLAB batch system. <http://sourceforge.net/projects/matlabbatch/>.
- [6] G. D. Iannetti, R. K. Niazy, R. G. Wise, P. Jezzard, J. C W Brooks, L. Zambrenu, W. Venart, P. M. Matthews, and I. Tracey. Simultaneous recording of laser-evoked brain potentials and continuous, high-field functional magnetic resonance imaging in humans. *NeuroImage*, 28(3):708–719, Nov 2005.
- [7] B. Kemp, A. Varri, A. C. Rosa, K. D. Nielsen, and J. Gade. A simple format for exchange of digitized polygraphic recordings. *Electroencephalogr Clin Neurophysiol*, 82(5):391–393, May 1992.

- [8] Bob Kemp. European Data Format. <http://www.edfplus.info/>.
- [9] Bob Kemp and Jesus Olivan. European data format ‘plus’ (EDF+), an EDF alike standard format for the exchange of physiological data. *Clin Neurophysiol*, 114(9):1755–1761, 2003.
- [10] Yves Leclercq, Evelyne Balteau, Thanh Dang-Vu, Manuel Schabus, André Luxen, Pierre Maquet, and Christophe Phillips. Rejection of pulse related artefact (PRA) from continuous electroencephalographic (EEG) time series recorded during functional magnetic resonance imaging (fMRI) using constraint independent component analysis (cICA). *NeuroImage*, 44(3):679–691, 2009.
- [11] Yves Leclercq, Pierre Maquet, Quentin Noirhomme, Christian Degueldre, Evelyne Balteau, Andrea Soddu, Jessica Schrouff, Mohamed Bahri, and Christophe Phillips. fMRI Artefact rejection and Sleep Scoring Toolbox (FASST). <http://www.montefiore.ulg.ac.be/~phillips/FASST.html>, 2009. Cyclotron Research Centre, University of Liège, Belgium.
- [12] Yves Leclercq, Jessica Schrouff, Quentin Noirhomme, Pierre Maquet, and Christophe Phillips. fMRI Artefact rejection and Sleep Scoring Toolbox. *Computational Intelligence and Neuroscience*, 2011:11, 2011.
- [13] Vladimir Litvak, Jeremie Mattout, Stefan Kiebel, Christophe Phillips, Richard Henson, James Kilner, Gareth Barnes, Robert Oostenveld, Jean Daunizeau, Guillaume Flandin, Will Penny, and Karl Friston. EEG and MEG data analysis in SPM8. *Computational Intelligence and Neuroscience*, 2011:32, 2011.
- [14] Marcello Massimini, Reto Huber, Fabio Ferrarelli, Sean Hill, and Giulio Tononi. The sleep slow oscillation as a traveling wave. *J Neurosci*, 24(31):6862–6870, Aug 2004.
- [15] Members and collaborators of the Wellcome Trust Centre for Neuroimaging. Statistical Parametric Mapping, SPM8. <http://www.fil.ion.ucl.ac.uk/spm>, 2008. Wellcome Trust Centre for Neuroimaging, University College London, UK.
- [16] Matthias Mölle, Lisa Marshall, Steffen Gais, and Jan Born. Grouping of spindle activity during slow oscillations in human non-rapid eye movement sleep. *J Neurosci*, 22(24):10941–10947, Dec 2002.
- [17] R. K. Niazy, C. F. Beckmann, G. D. Iannetti, J. M. Brady, and S. M. Smith. Removal of fMRI environment artifacts from EEG data using optimal basis sets. *NeuroImage*, 28(3):720–737, Nov 2005.
- [18] Rami K. Niazy. FMrib plug-in for EEGLAB. <http://users.fmrib.ox.ac.uk/~rami/fmribplugin/>, 2006. Centre for Functional MRI of the Brain (FMrib), University of Oxford, UK.
- [19] Robert Oostenveld, Pascal Fries, Eric Maris, and Jan-Mathijs Schoffelen. FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Computational Intelligence and Neuroscience*, 2011:9, 2011.
- [20] Robert Oostenveld, Jan-Mathijs Schoffelen, Eric Maris, Pascal Fries, and Ole Jensen. Field-trip toolbox for EEG/MEG-analysis. <http://www.ru.nl/neuroimaging/fieldtrip>, 2009. Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, the Netherlands.
- [21] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.
- [22] Christophe Phillips, Gérald Thielen, and Pierre Maquet. Iteratively applied ICA to reject ballistocardiogram artefact of EEG data acquired during sleep fMRI. Abstract of the Human Brain Mapping conference, June 2006.