

Technical Document

FASTER v1.2.3b

Fully Automated Statistical Thresholding
for EEG artifact Reduction

Hugh Nolan

Robert Whelan

Released: 13th February 2012

Introduction

FASTER is a software suite that works in tandem with EEGLAB. *FASTER* can be used to automatically process batches of files, taking as input raw unprocessed data in .bdf or .set format, and creating as output epoched data with artifacts removed, as well as grand averages of different conditions. Artifact removal is automated, using statistical methods to remove bad channels, epoch and ICA-separated non-EEG signals.

For full details of the rejection algorithms / properties used, see Nolan, Whelan, Reilly: “FASTER: Fully Automated Statistical Thresholding for EEG artifact Reduction”, *Journal of Neuroscience Methods*, 192(1), pp. 152-62, 2010.

This manual details the practical working of the user interface designed for this program.

1: Installation

Requirements for FASTER:

- MATLAB v7.4/R2007a or above (an older version *may* work, but version 7.0 has been tested and does not)
- Signal processing toolbox
- Statistics toolbox
- EEGLAB (has been tested with several versions from releases 7, 8 and 9).

- 1) Download the FASTER package from <http://www.mee.tcd.ie/~neuraleng/Research/Faster?action=downloadman&upname=FASTER.zip>
- 2) Unzip the FASTER.zip package to a new directory in the EEGLAB plugins directory, and add the directory to your MATLAB path.
- 3) Unzip the firfilt1.5.1.zip package to a new directory in the EEGLAB plugin directory, and add this directory to your MATLAB path.
- 4) Run “FASTER_GUI” from the MATLAB command line, or open EEGLAB, load files for processing, and click “Process using FASTER” in the “Tools” menu.

2: Quick Start

Note: It is highly recommended to read the options descriptions and the appendix as well as using the Quick Start.

2.1: Checklist

- 1) Select input and output directory, channel locations, and file type.
- 2) Set up file filter and prefix, if necessary.
- 3) Enter number of EEG and external channels, and select reference channel.
- 4) Select markers for epoching, epoch length and baseline subtraction length.
- 5) Enter channels for ICA, and then the EOG channels in ICA rejection options.
- 6) Turn on grand average and enter markers for making separate files.

2.2.1: Step-by-step (Command line)

- 1) Run “FASTER_GUI” from the MATLAB command line.
- 2) Click the “Job Directory” button, and select the top folder in which the .bdf or .set files are located. The files can be located in subdirectories of the job directory.
- 3) If a separate output directory is required, click the “Output Directory” button and select a folder in which the processed files will be stored.
- 4) If using .bdf files or .set files without channel locations already loaded, click the “Channel locations” button, and select the location file for the channels you wish to use. This should specify locations for the EEG channels and any external (e.g. EOG, EMG, mastoids, etc.) channels you wish to keep in the analysis. The number of channels in this file should be equal to the **number of EEG channels plus the number of External channels** specified in the channel options section. Note: if a .set file already has channel locations, they will not be overwritten.
- 5) Pick the appropriate file type from the drop down menu.
- 6) If you wish to process only files containing a certain string in the filename/filepath, enter this string in the “file filter” box.
- 7) If you wish to put a prefix in front of the output files (for example, to try multiple processing options on the same set of files without overwriting the previous output), enter it in the “Output file prefix” box.
- 8) Click the “Channels” option button. Double click the “EEG channels” option and enter the channels that are EEG channels (e.g. 1:128, or 1:64). If you are missing one or two channels, due to e.g. bad electrodes, mark them as EEG channels here and list them in the “Known bad channels” option box).
- 9) Double click the “External channels” options and enter the channels that are EOG, EMG, mastoids, etc (e.g. 129:132, 129:136).
- 10) Ensure that the reference channel number (for processing) is a valid EEG channel. Fz is recommended, as this was used for evaluation of the FASTER method, but it is only an **intermediate step** for artifact removal. The final output reference will by default be in **average reference**, although the data

can be re-referenced to any electrode(s) using the “Output reference” option.

- 11) If you wish the data to be epoched, click on the “Epoch” options button, and input the markers for epoching (e.g. [1 2 100]) in the “Markered epoching” section.
- 12) Enter the epoch length in **seconds** (e.g. [-0.5 1]).
- 13) Alternatively use the “Unmarked epoching” option to create periodic epochs.
- 14) Enter the baseline subtraction length (e.g. [-0.1 0], or 0 to turn off baseline subtraction).
- 15) Click the “ICA” options button, and ensure that the ICA channels are valid channels (unless you have a good reason, the best option is to include all EEG and External channels).
- 16) Open the “Rejection options” in the “ICA” options. Enter the EOG channels, if present. If not, using two frontal channels near the eyes, along with Fpz (i.e. selecting Fp1, Fp2 and Fpz), may suffice.
- 17) If a grand average is desired, click the “Grand average” option button.
- 18) Turn on the “Make grand average” checkbox.
- 19) If separate averages for different epoch markers are desired, enter them in the “Grand average markers” option box.
- 20) If the data is epoched during processing, ensure that the grand average length is less than or equal to the epoch length.
- 21) If desired, press “Save job” (recommended, allows resuming of cancelled job).
- 22) Press “Run job”.

2.2.1: Step-by-step (EEGLAB)

- 1) Run EEGLAB and open the dataset(s) for processing. Ensure that the option to keep at most one dataset in memory at once is **unchecked**. Support for this may be added in later versions of FASTER.
- 2) Click “Process using FASTER” from the “Tools” menu.
- 3) If a separate output directory is required, click the “Output Directory” button and select a folder in which the processed files will be stored. If this is selected, the “Save datasets” option will be turned on.
- 4) If using datasets without channel locations already loaded, click the “Channel locations” button, and select the location file for the channels you wish to use. This should specify locations for the EEG channels and any external (e.g. EOG, EMG, mastoids, etc.) channels you wish to keep in the analysis. The number of channels in this file should be equal to the **number of EEG channels plus the number of External channels** specified in the channel options section. Note: if a dataset already has channel locations, they will not be overwritten.
- 5) If you wish to process only files containing a certain string in their set name, enter this string in the “file filter” box.
- 6) If you wish to put a prefix in front of the output files (for example, to try multiple processing options on the same set of files without overwriting the previous output), enter it in the “Output file prefix” box. This is recommended if
- 7) If you wish to save the output datasets, click the “Save datasets” option. This is enabled by default if an output folder is set. See section 3 for more details on this option.
- 8) If you do not wish for new datasets to be created in EEGLAB – so that the datasets loaded will be replaced by their processed version – turn on the “Overwrite” option. If this is disabled, processed datasets will be added to the list of open datasets in EEGLAB.
- 9) Click the “Channels” option button. Double click the “EEG channels” option and enter the channels that are EEG channels (e.g. 1:128, or 1:64). If you are missing one or two channels, due to e.g. bad electrodes, mark them

as EEG channels here and list them in the “Known bad channels” option box).

- 10) Double click the “External channels” options and enter the channels that are EOG, EMG, mastoids, etc (e.g. 129:132, 129:136).
- 11) Ensure that the reference channel number (for processing) is a valid EEG channel. Fz is recommended, as this was used for evaluation of the FASTER method, but it is only an **intermediate step** for artifact removal. The final output reference will by default be in **average reference**, although the data can be re-referenced to any electrode(s) using the “Output reference” option.
- 12) If you wish the data to be epoched, click on the “Epoch” options button, and input markers for epoching (e.g. [1 2 100]) in the “Markered epoching” section.
- 13) Enter the epoch length in **seconds** (e.g. [-0.5 1]).
- 14) Alternatively use the “Unmarked epoching” option to create periodic epochs.
- 15) Enter the baseline subtraction length (e.g. [-0.1 0], or 0 to turn off baseline subtraction).
- 16) Click the “ICA” options button, and ensure that the ICA channels are valid channels (unless you have a good reason, the best option is to include all EEG and External channels).
- 17) Open the “Rejection options” in the “ICA” options. Enter the EOG channels, if present. If not, using two frontal channels near the eyes, along with Fpz, may suffice.
- 18) If a grand average is desired, click the “Grand average” option button.
- 19) Turn on the “Make grand average” checkbox.
- 20) If separate averages for different epoch markers are desired, enter them in the “Grand average markers” option box.
- 21) If the data is epoched during processing, ensure that the grand average length is less than or equal to the epoch length.
- 22) If desired, press “Save job” (recommended, allows resuming of cancelled job).
- 23) Press “Run job”.

3: Option Descriptions

Main panel

Note: options marked as **(Command)** are only available after opening FASTER from the command line, and options marked as **(EEGLAB)** are only available after opening FASTER from EEGLAB.

- **Job Directory (Command):** This is the directory in which all the .bdf or .set files for processing are stored. Files can be in subdirectories of this. If there is more than one file for processing in a (sub)directory, the files will be moved to their own subdirectories within this, named for each file.
- **Output Directory:** This is the directory in which all the .set files output are stored. If not set, output files will be in the Job Directory **(Command)** or their original directory **(EEGLAB)** or the MATLAB current directory **(EEGLAB)**. It is recommended to set this option, especially when using .set files, to avoid confusion with original and processed files.
- **Channel locations:** This specifies the channel location file to allocate to the files. If the files being processed are .set, original channel locations will not be overwritten. FASTER assumes all files have the same number of channels and channel locations – if using .set files with different channel locations, proper operation isn't guaranteed.
- **File type (Command):** specifies whether the files to search for are .bdf or .set files. If using the .set option to reprocess a directory, be aware that .set files in the "Intermediate" directories will not be processed, so if you want to include them, they must be moved. Also note that while processing .set files with no, the original .set file is saved in the "Intermediate" directory, prefixed with "Original_". Therefore if you want to reprocess a directory of .set files, you will need to move the original files.

- File filter: used to filter files within the job directory. Only files with the text specified in this box in the filename (**Command**) or dataset name (**EEGLAB**) will be processed. Leave empty to process all files.
- Output file prefix: text in this box is prepended to the filenames of the .set files produced. It is also prepended to the setname (**EEGLAB**). This could be used to run separate analyses on the same files without overwriting the output files.
- Individual folders (**Command**): If set, any directory containing more than one .bdf / .set file will have a series of subdirectories created for each file, and the files moved into them. If the Output Directory option is set, these directories will be created in the Output Directory, if not they will be created in the Job Directory.
- Resume (**Command**): used only for a loaded job file. If set, FASTER processing will resume where the operation was cancelled last time. If the operation finished, no new files will be processed, but a grand average may be created if enabled. Any new files added to the job directory will be ignored. Note that if the file filter is changed and resume is enabled, some files may not be processed, therefore it is better to disable resume if changing the file filter. The number beside it indicates what file number processing will resume from.
- Save datasets (**EEGLAB**): If set, datasets will be saved – if not, they will only be updated in the EEGLAB memory (list of datasets).
If a dataset is already saved, the output filename will be “prefix_filename.set” – e.g. if you load a file name “P3_Subj1_Run1.set” and specify the output file prefix as “Test_Proc_”, the output filename will be “Test_Proc_P3_Subj1_Run1.set”. If the output folder is specified, this file will be created there; otherwise it will be in the same folder as the file was loaded from.
If a dataset has not yet been saved, it will be created in the output folder, if set, or else the MATLAB current directory. If the set has been

named (e.g. using the Edit -> Dataset Name option in EEGLAB), the output file will be named “prefix_ALLEEG(x)_setname.set”; if there is no setname it will be called “prefix_ALLEEG(x).set”, where x is the dataset number in the EEGLAB list.

- **Overwrite (EEGLAB):** If set, datasets will be overwritten in memory. If not, the processed datasets will be added to the end of the EEGLAB list.
- **Save job:** saves the current job setup to a file. Default file extension is .eegjob. Depending on the operating system, the file extension may not be appended to the title automatically. If so, the file can still be loaded, but may need to be renamed manually to have the .eegjob extension.
- **Load job:** loads a previously saved job. If this file is already in use by another computer, or if execution of this file was previously stopped unexpectedly, you will be asked whether you would like to **join** the processing job (see the **appendix** for more details on the distributed processing), **reset the queue** (this will clear all processing tracking files – use this option if the jobfile was stopped unexpectedly before, but be careful that no other computers are using the file. The **resume function** will still work as normal), or **cancel** loading the file. Resuming is disabled when running from EEGLAB.

Save options

The checkboxes beside each option button allow saving of intermediate files before each step (or more exactly, before any information/noise is removed at each stage). This allows the processing options to be reviewed, and therefore changed if necessary. These files are saved in a new directory “Intermediate” which is created in the directories corresponding to each file.

- Save before filtering: The file has been opened / converted and channel locations have been added if necessary.
- Save before channel interpolation: The data has been filtered according to the filter options.
- Save before epoching: Bad channels have been detected and interpolated, if enabled.

Filter options

For more details on the meaning of the options, see the appendix.

- High pass on: turns on and off the high pass filter.
- High pass frequency: adjusts the high pass filter frequency.
- High pass options:
 - HP ripple: specifies the maximum ripple allowed in the pass band of the high pass filter.
 - HP attenuation: specifies the desired attenuation of the stop band of the high pass filter.
 - HP transition band width: specifies the width of the transition band of the high pass filter.
- Low pass on: turns on and off the low pass filter.
- Low pass frequency: adjusts the low pass filter frequency.
- Low pass options:

- LP ripple: specifies the maximum ripple allowed in the pass band of the low pass filter.
 - LP attenuation: specifies the desired attenuation of the stop band of the low pass filter.
 - LP transition band width: specifies the width of the transition band of the low pass filter.
- Notch on: turns on and off the notch filter.
- Notch frequency: adjusts the notch filter frequency.
- Notch options:
 - Notch bandwidth: specifies the width of the stop band of the notch filter.
 - Notch ripple: specifies the maximum ripple allowed in the pass band of the notch filter.
 - Notch attenuation: specifies the desired attenuation of the stop band of the notch filter.
 - Notch transition band width: specifies the width of the transition band of the notch filter.
- Resample:
 - Resample on: turns on and off resampling. Resampling has been seen to produce filter artifacts at beginning and end of files, and where any boundary events occur, and is generally not recommended if it can be avoided. If resampling is desired, ensure that there is some non-experimental data at the start / end of the files, or that you can afford to lose some epochs. Note that due to the statistical nature of the epoch rejection algorithm, not all filter artifacts may be removed. For single-trial use, resampling is not recommended.
 - Resample frequency: New sampling frequency.

Channel options

- **Reference channel:** here you can set the reference channel during processing (data is temporarily re-referenced to a single channel) and for final output. This value should be a channel which is relatively clean (don't choose a channel which was noted to have poor quality signal during acquisition). For evaluation of the FASTER method, Fz was used. The EEG data in the final output is referenced to the average reference, but you can also set a reference channel for the output file here. If left blank, this will default to average reference.
- **EEG channels:** the number of the channels which contain EEG data. Usually in Biosemi recordings, these will be the first 64 or 128. Normal MATLAB notation is recognised, so 1:128 specifies the first 128 channels as EEG, 1:64 specifies the first 64, etc. Note that the channel locations file must contain the correct amount of EEG and external channels.
- **External channels:** the number of the channels which contain EOG, EMG, etc, data. Usually in Biosemi recordings, these will come after the EEG channels, e.g. 129:136 specifies that the 8 channels after 128 EEG channels are external channels. Note that the channel locations file must contain the correct amount of EEG and external channels. Be aware that if a channel is not selected as an EEG or external channel, it is removed from the dataset. This may have consequences for the reference channel. For example, if channels [1:63 66:128] are selected to be EEG channels and channels [129 131 134] are selected to be external channels, and channel 85 was chosen as the reference, channel 85 would correspond to the original channel 87, as two channels (64 and 65) before it have been removed. To avoid confusion in this way, it is best to keep all channels, and indicate missing channels as "known bad channels". These will be interpolated.
- **Known bad channels:** these channels are known to be bad in all data sets. This could be due to missing or faulty electrodes throughout data collection. They will be interpolated, but will remain present in the

output file, therefore their channel locations will not change. This is the recommended way of dealing with known missing channels – leave their locations in the channel location file, and enter the number in this box.

- Interpolate after ICA: if set, this option runs the interpolation step after ICA, instead of before, to avoid mixing any non-linearities introduced by interpolation into the ICA decomposition. Any channels marked for interpolation will not contribute to the epoch rejection section or the ICA decomposition.
- Channel rejection: enables or disables the automated bad channel detection feature. Bad channels are interpolated when detected.
- Rejection options: allows you to turn on and off separate parameters of the channel rejection algorithm, and modify their statistical thresholds.
 - Channel correlation: rejects based on the channel's mean correlation with all other channels.
 - Channel variance: rejects based on the channel's variance.
 - Hurst exponent: rejects based on the Hurst exponent.
 - Exclude EOG channels: if set, EEG channels specified as EOG channels for use in ICA EOG detection will not be interpolated if detected to be bad channels.

Epoching options

- Markers for epoching: allows you to turn on marked epoching. The marker numbers that your data will be epoched on are entered in the field here. Text-based markers require a special input format – {'Markers 1', 'Marker 2', 'Marker 3', 'etc'}. **The curly brackets and quotation marks are necessary for this to work properly.** If no markers are entered, epoching will not occur, and data are assumed to be single trial.
- Unmarked epoching: if no markers are present in your data, this allows epochs to be created at a specific interval. This can be enabled and disabled, and the interval chosen. **If enabled, this overrides marked epoching.** Epoch length must be entered for this to work. The function allows overlapping epochs.
- Epoch limits: the length of the epoch, in **seconds**. It takes two values, one the minimum (usually a number of seconds before each marker, therefore a negative number) and the other a maximum (usually a number of seconds before each marker, therefore a positive number).
- Baseline subtraction: the times over which to take a mean value to subtract from the ERP. This occurs on a per-channel, per-epoch basis. This is used to line up ERPs to have a similar baseline period. This is in **seconds**, and must fit within the epoch limits.
- Epoch rejection: turns on or off the bad epoch detection feature. Bad epochs are removed from the dataset. This can be performed on pre-epoched data, if epoching is turned off in the above sections.
- Rejection options: allows you to turn on and off separate parameters of the epoch rejection algorithm, or modify their statistical thresholds.
 - Deviation from mean: rejects based on the epoch deviation from the mean value of each channel.
 - Variance: rejects based on the epoch's variance.

- Amplitude range: rejects based on the epoch's amplitude range.

ICA options

- Run ICA: enables or disables running ICA on the dataset. The ICA algorithm used is the default Infomax ICA algorithm. If binica is setup correctly, it will run here, otherwise the standard runica function will be called.
- Channels for ICA: selects what channels to use for ICA. Unless you have a good reason, this should be all of the EEG channels and all of the external channels.
- ICA k value: this is a quality value, which determines how many ICA components can be output, based on the length of the dataset and the number of channels. Unless you have a good reason, the default value of 25 should work well.
- Keep existing ICA weights: if using .set files with ICA decompositions already computed, this option will disable re-referencing and re-computing the ICA to maintain the existing decomposition. This may adversely affect the bad channel detection algorithm.
- Save component topography images: if set, this will save the topographical maps of the components to the “Intermediate” folder. If component rejection is enabled, rejected components will be surrounded in a red border.
- ICA component rejection: enables or disables the automatic ICA component rejection feature. **This feature can be used on previous ICA decompositions if the “keep existing ICA weights” option is turned on.**
- Rejection options: allows you to turn on and off separate parameters of the ICA component rejection algorithm, or modify their statistical thresholds.

- Median gradient: rejects based on the median gradient value. This tends to detect spurious high frequency activity.
- Spectral slope: rejects based on the slope of the frequency spectrum in the low-pass transition band. This detects white noise components.
- Spatial kurtosis: rejects based on the peakedness of the scalp maps which map ICA components back to scalp space. This detects artifactual activity which occurs on only one or two channels.
- Hurst exponent: rejects based on the Hurst exponent of the ICA component time-series. This detects general non-biological signals.
- EOG correlation: rejects based on the ICA component time-series correlation with the EOG channels. This detects EOG activity.
- EOG channels: specify any EOG channels recorded here. These will be a subset of the external channels. Alternatively, if no EOG channels have been recorded, try using one or two of the most frontal electrodes recorded – the EOG signal should still be suitably high to allow the correlation measure to detect an EOG component.
- Lowpass frequency: if not using the inbuilt filters, entering the lowpass cutoff frequency used here enables the “spectral slope” rejection method to be used. Otherwise, it will be ignored.

Epoch interpolation options

- Epoch interpolation: enables or disables interpolation of single channels within single epochs.
- Rejection options: allows you to turn on and off separate parameters of the epoch interpolation algorithm, or modify their statistical thresholds.
 - Median gradient: rejects based on the median gradient (higher gradient implies quick change, so this detects unusual high frequency noise) of channels within epochs.
 - Variance: rejects based on the variance (unusually high or low activity) of channels within epochs.
 - Amplitude range: rejects based on the amplitude range of channels within epochs.
 - Deviation from mean: rejects based on the deviation of channels within epochs from the whole dataset channel mean. Detects abnormal slow wave activity or uncorrected drift.

Grand average options

- Make grand average: enables or disables the grand average feature. Grand averages are saved in the job / output directory. If opened from EEGLAB, they are also added to the end of the EEGLAB dataset list.
- Grand average markers: if markers are entered, separate grand averages will be made for each marker. These will be prefixed with the marker number. These can be single markers, e.g. entering 1, 2, 3, 4, 5 will output separate averages for markers 1 - 5, or groups of markers, e.g. entering {[1, 2], [3, 4, 5]} will output two averages of markers 1-2 and 3-5. **Note the curly brackets around the grouped markers;** these are necessary for proper operation. Further syntax examples, for markers that have text: {'Stim1', 'Stim2', 'Stim3'} will output 3 average files for Stim1, Stim2 and Stim3 separately. {'Stim1', 'Stim2'}, {'Stim3', 'Stim4'}} will output 2 average files of Stim1 and Stim2 together, and Stim3 and Stim4 together. Note the curly brackets and single quotes around the markers with text in them; these are necessary for proper operation.
- Grand average length: the limits of the epoch taken for grand averaging. These should be less than or equal to the limits of the epochs in the file. If you get errors that no epochs were selected from your datasets, shortening the epoch length here may solve this.
- Trimmed mean: if set, FASTER uses a trimmed mean rather than sample mean to calculate the grand average. This removes a percentage of the most extreme data from the averaging, and so can reduce any noise remaining.
 - Trimmed mean percentage: percentage of data to remove from the mean. (Specifically, half of this percentage is removed from each tail of the data distribution). The default is 10%, but different values may suit different data types better.
- Subject removal: enables or disables automation detection and removal of bad subjects based on their ERP.

- Rejection options: allows you to turn on and off separate parameters of the subject removal algorithm, or modify their statistical thresholds.
 - Deviation from mean: rejects based on the deviation of the ERP from the mean channel values of all ERPs.
 - Variance: rejects based on the variance (unusually high or low activity) of the ERP.
 - Amplitude range: rejects based on the amplitude range of each ERP.
 - Max EOG value: rejects based on the highest value in the EOG channels per ERP.

More options

- Save as default options: saves the current option set as that which is loaded on startup of the FASTER GUI.
- Load default options: loads the default options.
- Save as new jobfile: saves the current jobfile under a new name.

4. Appendix

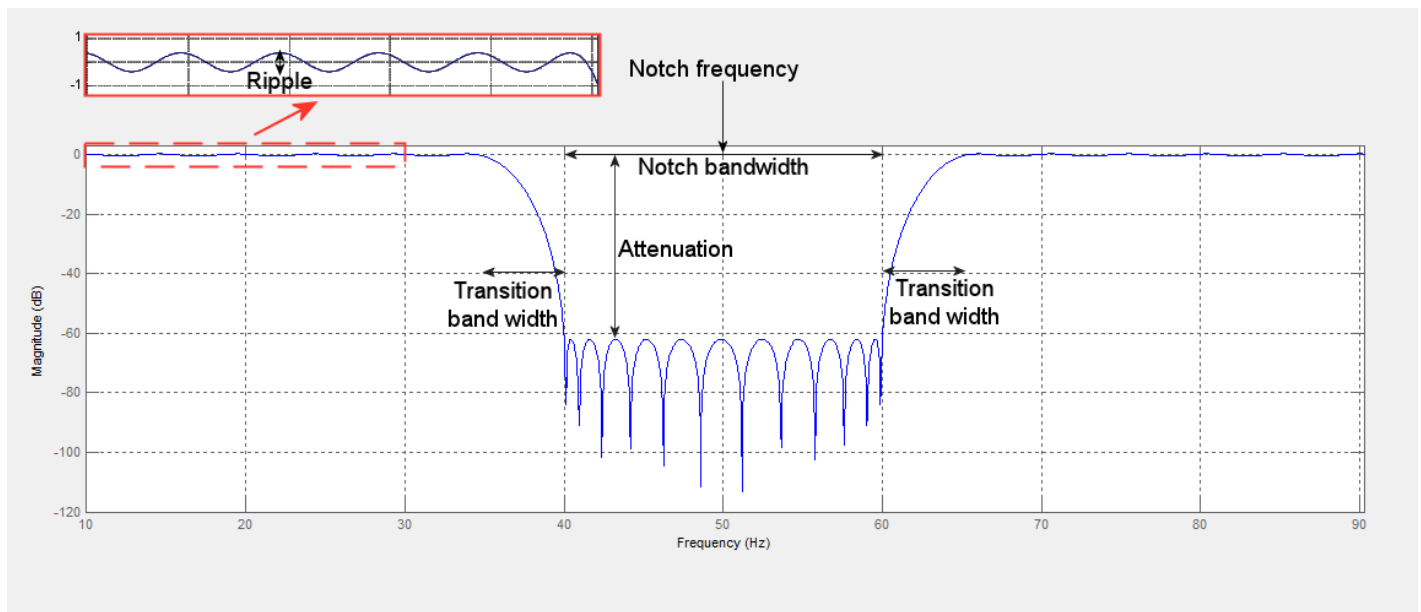


Figure 1. Notch filter design describing the different parameters available in the filter options section. High- and low-pass filters work similarly.

- Regarding **filtering**, Figure 1 depicts a notch filter design, exaggerated for clarity. The design features are noted on the figure. High- and low-pass filters share the same design options excepting the notch bandwidth, so their operation can be extrapolated from this diagram. If you are unsure about a parameter, the default options give sensible values. Adjust with care, as trying to use extreme values (very small transition band, very low frequencies, very high attenuation, etc.) may not be possible to design, and so the output from the filtering process may not be stable. **This will not be detected as an error.**
 - Notch frequency: the centre frequency of the notch filter.
 - Notch bandwidth: how wide the notch band is (corner frequencies are therefore centre frequency – bandwidth/2 and centre frequency + bandwidth/2).
 - Ripple: how much change there is in the passband. This should be small.
 - Attenuation: the minimum that the frequencies in the notch band are attenuated (then may be attenuated more, as can be seen in the ripple).

- Transition band width: how wide the transition between the notch band and the passband is.
- Regarding the **statistical thresholding**, the threshold which can be changed for each measure is a Z threshold, i.e. a measure of standard deviation from the mean. Increasing the threshold means that less data will be rejected and decreasing means that more data will be rejected. The standard value of 3Z was used for all testing, however other values may give better results for different recording paradigms / EEG setups.
- Regarding the possibility of **distributing processing** among multiple computers – the system is still in beta testing, and so may contain some bugs. The idea is to process multiple files stored in a networked drive (or a drive otherwise accessible by multiple computers) simultaneously.

The system works by creating an .eegQ file, a Queue folder and a Processing folder in the directory the .eegjob file is stored. The .eegQ file is used to determine what file is to be processed next. The Queue folder is used to track when a computer is accessing the .eegQ file, so that there is no conflict with multiple computers accessing the file and then processing the same file. The Processing folder is used to track when a computer is processing files, so that grand averaging is only performed when all datasets are processed.

This system allows an unlimited number of computers to open an eegjob file and process datasets. As an example, take 3 computers: the first computer processes dataset 1, the second computer processes dataset 2, the third computer processes dataset 3. The first computer finishes, then processes dataset 4, the third finishes and processes dataset 5, the second finishes dataset 6. The first finishes, and there are no more files, so quits. The second finishes and quits. The third finishes, detects that all computers are finished processing, and then creates the grand average (if set) and does cleanup.

Directory paths are stored relative to the location of the job file, so path issues between Linux and Windows etc should not cause problems. The main restriction of this system is that the jobfile **must** be stored on the same drive as the files to be processed, otherwise it can't work.

The disadvantage of using this approach is if an error occurs on one computer during processing, or if the user cancels the operation using Ctrl+C. A processing or queue file may remain, which is misleading to other computers. If a processing file remains, no grand averaging will be performed. If a queue file remains, computers may hang, as they wait for the file to be removed. If this happens, the queue file may be manually deleted (always delete the LOWEST queue file) or the operation can be cancelled (using Ctrl+C) on all computers, and the job reloaded using the “Reset Queue” option. Resuming processing from the first unprocessed file is still available after resetting the Queue. Errors should be caught and processing files removed, however the use of Ctrl+C to abort processing cannot be caught and so the files will remain.

This functionality can also be used to process multiple files simultaneously on one computer by opening multiple instances of MATLAB. This will only provide advantages on computers with multiple processors (multicore computers), as ICA and other computationally intensive operations are only processed using a single processor. If doing this, do ensure you have enough RAM (4GB+ recommended, especially for Windows systems)!

5. Version History

- v1.0 Original release
- v1.0.1 Fixed bug during reporting of baseline variance
- Allowed channel locations to be left unset if using .set files (all files need to have their locations already loaded)
- Included the FIRfilt plugin in the FASTER distribution
- Added option to specifically keep existing ICA decompositions (previously, re-referencing invalidated them)
- v1.0.2 Fixed crash if ICA activations were not stored in memory
- Fixed crash if no components were rejected
- Fixed bug where baseline subtraction was not working
- Added option to exclude EEG channels specified as EOG channels from interpolation
- V1.1 Fixed some bugs when using .fdt files
- Fixed bugs when using more than one reference electrode
- Fixed miscellaneous bugs where prefixes weren't added to output files
- Added grand averaging features:
- Grand average with more than one epoch marker
 - Use trimmed mean for grand average

Added the options to interpolate bad channels after performing ICA (bad channels will not be used in computing ICA decomposition)

Added option to epoch continuous data at set intervals

Added save-as option to save a jobfile under a new name

Added new error catching system (any errors can be saved so we can hunt them down quicker!) – run “FASTER_GUI(1)” from the command line to enable this

V1.1.1 Fixed some bugs between different MATLAB versions

Added ability to save ICA topographic images

Added explicit option to turn on or off markered epoching

Added option to change referencing of output file

Fixed some grand averaging issues

Expanded error-catching system to help catch any further bugs. It is now turned on by default so users experiencing bugs don't have to rerun data files again

Enabled binica, if it is setup as per the instructions in EEGLAB

Enabled the use of text-based markers as well as numerical markers (be sure to read to “markered epoching” section for the correct syntax)

Clarified an issue using ICA component rejection without using the built-in lowpass filter (see the new option in the ICA rejection options)

Fixed a small bug with external channels which have no locations

Miscellaneous bug fixes

V1.1.2 Minor bug fixes

Added logging of subject removal in grand average

V1.1.3 Bug fixes in interpolation functions (whole channel / epoch)

Bug fixes for interpolation after ICA options

Bug fixes in ICA component plots

Other minor bug fixes

Many thanks to Grega Repovs for all his help working out bugs in this version!

V1.2b Bugs

Bug fixed for channel properties when using external channels where were mixed in with EEG channels

Other minor bug fixes

Features

Integration with EEGLAB – process datasets that are open in EEGLAB, with options to save them

Output directory option added

Option to turn on or off individual subfolder creation

Distributed processing added (still beta!) – allows a number of computers to share an EEG job which is processing files on a network drive. This feature can also be used to simultaneously process files using multiple instances of MATLAB, taking advantage of multiple processors.

Note

This is a beta version as the distributed processing architecture may have introduced some bugs – it has been tested extensively, but there always seems to be something we forgot!

V1.2.1b Bug fix for always invalid channel locations (woops!)

Other minor bug fixes

New version of the firfilt plugin included (v1.5)

V1.2.2b Fixed fileparts bug in newer versions of MATLAB

New version of the firfilt plugin included (v1.5.1)

V1.2.3b Fix channel locations not being loaded with new version of EEGLAB

6. FAQs

- **Q. What is *FASTER*?**
- A. *FASTER* is a fully automated, unsupervised method for processing of high density EEG data. *FASTER* is free and the code is open source.
- **Q. What do I need to start?**
- A. You will need [Matlab](#) (with the statistics and signal processing toolboxes) and [EEGLAB](#) (version 7 or higher). You will also need a free filter plugin for EEGLAB called *firfilt*, which is written by [Andreas Widmann](#), who has kindly given us permission us to bundle his code with the *FASTER* code so you will not need to download *firfilt* separately.
- **Q. How much processing will *FASTER* do?**
- A. *FASTER* will take raw data and process it to the point at which statistics can be conducted on a between-subject level. You do not need to remove contaminated channels or bad epochs before using *FASTER*.
- **Q. How much signal processing do I need to know?**
- A. You can use *FASTER* without knowing any signal processing. The defaults in the current version are set to the values in the *FASTER* manuscript, which was based on real and simulated P3b data. All values are modifiable and all parameters can be turned on and off through the GUI. Adjusting some parameters may help improve the quality of your data, so users are encouraged to try *FASTER* in different configurations on a sample of their data. Our next goal is to determine the optimal configurations for particular types of data (e.g., P3a, MMN, N400) and for different populations (e.g., older adults, patient populations) and to make these configurations publicly available.
- **Q. How much Matlab do I need to know?**
- A. Almost none. If you can set the path and type in the command line then you are able to use *FASTER*. To set the path in Matlab (i.e., to tell Matlab where to find the files) click 'File' then 'Set Path', followed by 'Add with sub-folders' then Save and Close.
- **Q I don't have specific EOG channels - can I still use *FASTER*?**
- A. *FASTER* works best with recorded EOG channels: we tested specifically with EOG channels recorded from Biosemi devices with their flat-type electrodes. However, using some frontal EEG channels (e.g. Fpz, Fp1, Fp2, AF7, AF8) has also been seen to work, although it may not work every time.
- **Q. Is there a version of *FASTER* that doesn't need Matlab?**
- A. Not at the moment, but there will be in the future.
- **Q. Can I use *FASTER* in conjunction with other programs?**

- A. Yes, *FASTER* can be used to process data before using software such as [SPM](#), [BESA](#) and [CARTOOL](#). An automatic converter to these formats will be added to future versions of *FASTER*, including the construction of the first level design matrix for SPM.
- **Q. How many electrodes should I have?**
- A. *FASTER* is primarily designed for high-density EEG arrays (128 or greater channels) although it works well on 32-channel arrays. It has not been tested on fewer than 32 channels and we do not anticipate constructing a version for low-density arrays. If you do not have a high-density array, in particular you should check that the EOG is being removed. You may also want to turn off the "Channel rejection" and the "Epoch interpolation" options. *FASTER* should work well on arrays with greater than 128 channels.
- **Q. What operating systems can I run *FASTER* on?**
- A. *FASTER* has been tested on Windows and Linux (Ubuntu and Fedora). *FASTER* should work on any operating system that can run Matlab.
- **Q. What about channel locations?**
- A. *FASTER* uses several EEGLAB functions. Check out the [EEGLAB wiki](#) for information on channel locations etc.
- **Q. I'd like to script, rather than use the GUI. What are the important m-files?**
- A. The main functions used are the *_properties functions and the min_z function. The *_properties functions are somewhat idiosyncratic in that they each take different arguments, but their output is a standard format fed into min_z.

The general approach for artifact identification is (where X represents channel, epoch, component, etc). =

- `list_properties = X_properties(EEG, further, inputs);` % The further inputs can be seen in each file
- `exceeded_threshold = min_z(list_properties);` % min_z also takes further arguments to allow you to turn on or off specific testing properties or use a Z-score threshold other than 3
- `bad_X = find(exceeded_threshold);` % Exceeded threshold also details how many properties of each channel, epoch, component, etc exceeded the threshold, so you could use "`bad_X = find(exceeded_threshold >= 2)`" to tighten the rejection conditions to only reject Xs that are considered artifact-contaminated by 2 tests, if desired.

- **Q. I have an idea for improving *FASTER*. Can I contribute?**
- A. Yes, you can. The code is open source, and we are happy for anyone to improve *FASTER*. If you do modify the code, we would be grateful if you email us the changed file. We will include your modification as an option in future releases of *FASTER* and all contributions will be acknowledged. You can also send suggestions about improving *FASTER* to Hugh (nolanhu@tcd.ie) and/or Rob (robert.whelan@tcd.ie).
- **Q. What will be in future releases of *FASTER*?**
- A. The next steps are to add *FASTER* to EEGLAB as a plugin, and to allow processing of files that are already loaded in EEGLAB.
- **Q. Who should I send bug reports to?**
- A. Please email Hugh (nolanhu@tcd.ie) and/or Rob (robert.whelan@tcd.ie) with any bug reports. Please attach the “FASTER_errors.mat” file which was created in the job directory.