
NFB Lab Documentation

Release 0.1

Nikolai Smetanin

May 17, 2021

Contents


1	Experimental designer	3
1.1	Derived signal settings	5
1.2	Composite signal settings	6
1.3	Protocol settings	7
2	Experiment	11
2.1	Composite signal settings	12
2.2	Subject window	13
3	Data-driven filters (signals manager)	15
3.1	ICA	16
3.2	CSP	18
3.3	SSD	20
4	Occipital alpha rhythm neurofeedback	23
5	Results file structure	25
6	Requirements	27

NFBLab is a software that allows you to configure the design and conduct an experiment to study neurofeedback. The program code is laid out in a github repository <https://github.com/nikolaims/nfb>.

Contents:

CHAPTER 1

Experimental designer

 Experiment settings

—

□

×

File

Name:

experiment

Inlet:

LSL stream

NVX136_Data

Events inlet:

☐ Use events LSL Stream

Reference:

Exclude channels:

Print channels to exclude (labels)

Subtract channel from other:

Print subtractive channel (labels)

Plot raw:

☒

Plot signals:

☒

Plot source space:

☐

Show subject window:

☒

Reward period [s]:

0,25

Test beep sound:

Beep

Enable DC Blocker:

☐

Show photo-sensor rect.:

☐

Pre-filtering band:

low:

0,00

high:

0,00

Signals:

Signal

Add

Remove

Composite signals:

Composite

Add

Remove

Protocols:

Protocol

Add

Remove

Protocol groups:

Group

Add

Remove

Protocols sequence:

Protocol

Remove

Start

This module allows you to configure the experiment design. Customized design is saved in an .xml file and can be loaded with further use of the program. ‘Start’ button starts the module of carrying out experiments in which an experiment is conducted with your settings.

Below is a simplified version that describes the basic settings of the experiment in order from top to bottom and left to

right.

Name: the name of the experiment (in a folder 'results' there is a folder with the same name and added timestamp).

Inlet: selection of the data stream to which you want to connect. There is a choice of four options: Normal LSL stream (for connection of devices with LSL support), LSL generator (created LSL flow with a model signal for the test program), LSL from file (created LSL signal playback stream recorded in the file during the previous experiments), FieldTripBuffer (connection for FieldTripBuffer Protocol).

Reference: Exclude channels: a list of channels that should not be taken into account when conducting the experiment (in the construction of spatial filters).

Subtract channel from other: a list of channels that will be subtracted from others.

Plot raw: disables / enables online drawing of raw signals.

Plot signals: disables / enables online drawing of processed signals.

Plot source space: disables / enables online drawing of spatially distributed signals.

Show subject window: disables / enables the window with messages or feedback presentation for a subject.

Reward period: the period of accrual encouragement (if the processed signal from the test exceeds a predetermined threshold, then the subject is beginning to accrued encouragement points with the given period).

Test beep sound: check of the sound stimulus.

Enable DC-blocker: disables / enables DC-blocker. Notice, DC-blocker applies to all raw data including EEG channels, photosensors, etc.

Show photo-sensor rect.: enables / disables the rectangle at the bottom right of the screen during the experiment. Its brightness depends on the derived signal amplitude and imitate the feedback stimulus. It is used for computation of the latency (photo-sensor is needed).

Signals: configuration of the processed signals (such signals obtained from raw signals through the application of spatial filters, frequency filters, amplitudes calculation, subtracting the mean and dividing by the standard deviation, thus the smoothing is performed). When you double-click on one of the signals you open the composite signal settings:

1.1 Derived signal settings

Properties: Signal

Name:

Spatial filter: CUSTOM

Temporal settings:

Type: envdetector

Band: low high

Filter type: fft

Window size [samp.]:

Filter order:

Smoother type: exp

Smoother factor:

Artif. delay [ms]:

BCI mode: ☐ BCI mode

Signal Settings include:

Name: name of the signal.

Spatial filter: spatial filter can be chosen through the list of available filters or entered manually (“CUSTOM” type). A current signal sample is computed as a weighted sum of the raw samples, and that weights are construct the spatial filter. In the case of the custom filter, enter it as following: Cz=1, F4=1 (others will not be processed).

Temporal settings: settings of the temporal filter.

Type: type of the filter. envdetector - signal will be filtered and the envelope will be computed. filter - only temporal filtering without the envelope. identity - raw signal. Accordingly to the type, the following settings will be available or not.

Band: frequency filter border.

Filter type: FFT, butter (for Butterworth), complexdem, cfir.

Window size [samp.]: the size of the window in samples used for FFT and CFIR.

Filter order: order of the filter used for Butterworth and complexdem.

Smoother type: exponential (“exp”) or Savitzky–Golay (savgol) smoother.

Smoother factor: factor used in the exponential smoothing. Higher values correspond to a higher smoothness of the signal.

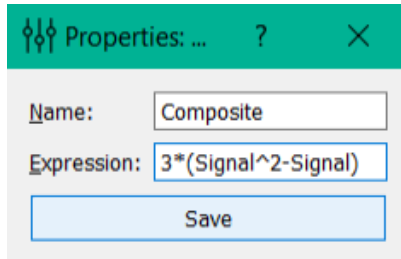
Artif. delay [ms]: artificial delay, used to add a latency between the signal and the feedback presentation.

BCI mode: the mode for using imaginary movements (in testing).

Save: saving of the signal.

Composite signals: configurations of composite signals (samples of these signals are obtained by means of an algebraic expression of the signals samples from the field of “Signals”). When you double-click on one of the signals you open the settings of the composite signal:

1.2 Composite signal settings



Properties: ... ? X

Name: Composite

Expression: $3 * (\text{Signal}^2 - \text{Signal})$

Save

Composite signal settings include:

Name: name of the signal.

Expression: expression used to compute samples of the composite signal from samples of processed signal (as the name of the variables used the names of the processed signals; supported operations: +, -, *, /, ^ and brackets).

Protocols: includes settings of protocols (experiment consists of several protocols, each of which has a duration, settings of signal processing and rendering properties window for subject). When you double-click on one of the protocols you open protocol settings.

1.3 Protocol settings

Properties: Protocol ? X

Name:	Protocol	
Duration [s]:	10,00	
Random over time [s]:	0,00	
Open signal manager in the end (SSD, CSP, ICA):	<input type="checkbox"/>	
Update statistics:	<input type="checkbox"/>	meanstd
Beep after protocol:	<input type="checkbox"/>	
Auto BCI fitting:	<input type="checkbox"/>	
Mock source:	<input type="checkbox"/>	
Make a pause after protocol:	<input type="checkbox"/>	
Enable detection task:	<input type="checkbox"/>	
Drop outliers [std]:	0	
Signal:	All	
Type:	Baseline	
Feedback type:	SinCircle	
Blink duration [ms]:	50	
Blink threshold:	0,00	
Mock signals file:		Select file...
Mock signals file dataset:	protocol1	
Mock from previous protocol raw data	<input type="checkbox"/> Protocol # 1	<input type="checkbox"/> Shuffle <input type="checkbox"/> Reverse
Muscular signal:	None	Threshold: 1,00
Message:		
Voiceover:	<input type="checkbox"/>	
Add half time extra message (for CSP):	<input type="checkbox"/>	
Half time extra message:		
Reward signal:	Signal	
Reward threshold:	0,00	
Show reward:	<input type="checkbox"/>	
Video file:		Select file...
Save		

Protocol settings include:

Name: name of the protocol.

Duration [s]: the duration of the protocol in seconds.

Random over time [s]: a random number between 0 and the entered value will be added or subtracted from the main duration each time the protocol is used.

Open signal manager in the end (SSD, CSP, ICA): opens at the end of protocol signaling manager where, by using different methods (SSD, CSP, ICA), computed spatial filters signals.

Update statistics: if this checkmark is exhibited and “meanstd” is selected, then, at the end of the protocol the average (mean) and standard deviation (std) of the signal will be recalculated, and subsequently mean will be subtracted from the signal and it will be divided by std. Min/max calculates values above zero (minimum) and usually below one

(maximum), where minimum and maximum are taken from the finished protocol.

Beep after protocol: after the protocol there will be sound indicating the end of the protocol.

Auto BCI fitting: ...

Mock source: ...

Make a pause after protocol: enables / disables a pause after the protocol.

Enable detection task: an obsolete function. It will be removed in the next version.

Drop outliers [std]: if “Update statistics” is checked, you can enter a number indicating a border of standard deviation to detect outliers and delete them.

Signal: name of the signal that will be processing. Corresponds to the name from the “Signals” or “Composite signals” properties. It can be set to “All” in the case when no specified signal is applied, during baseline, for instance.

Type: type of the protocol. Available 4 types of protocols: Baseline (on the screen of subject will be displayed a message, is used to collect data, to update the signal statistics, calculation of spatial filters, etc.), Feedback (if “Feedback type” is SinCircle, on the subject’s screen will be displayed the circle, with jagged edges, whom degree of unevenness depends on the particular signal in the “Signal” (see previous paragraph.), the subjects task is to achieve complete smoothness of the edge of the circle), ThresholdBlink (subjects screen at the time specified in paragraph “Blink duration” will turn white if the signal exceeds threshold defined in “Blink threshold”; this protocol is needed for measuring delays and can be considered obsolete), Video (in testing).

Feedback type: type of the feedback, if it is chosen in the “Type” field. SinCircle - a shrinking circle; RandomCircle - a shrinking circle with a random bound distortion; Bar - a bar that gets higher when the signal amplitude increases.

Blink duration: only available for ThresholdBlink protocol type (see above). Deprecated function.

Blink threshold: only available for ThresholdBlink protocol type (see above). Deprecated function.

Mock signals file: only available for FeedbackProtocol. Here you can specify the file from which to read the signal for FeedbackProtocol recorded previously. This signal is false and will be changed at the time of the signal with the actual signal.

Mock signals file dataset: only available for FeedbackProtocol. Here specifies the protocol of the last experiment whose data is stored in a file specified by paragraph above. The signal recorded in this protocol will substitute the actual signal.

Mock from previous protocol raw data: only available for FeedbackProtocol. This sets an alternative method of signal substitution. Data for substitution are taken from the experiment itself (from the previous protocol, its number is indicated).

Muscular signal: only available for FeedbackProtocol. ...

Message: Here you specify the message that will be displayed and presented to the subject during the block.

Voiceover: enables / disables audio message for the subject.

Add half time extra message (for CSP): if a tick is exhibited, then after half the time pass, a beep sounds will occur and message will change (additional message given by paragraph below). Deprecated function.

Half time extra message: Additional message. Deprecated function.

Reward signal: the signal that will be the basic for determining the awarding for subject.

Reward threshold: the threshold, above which will launch a counter points for subject award.

Show reward: if the user is exhibited the tick count points, awards will be presented.

Video file: add a path to the video if block type is video.

Protocol groups: inside a group it is possible to unite several blocks (protocols) in order to use them together in the general sequence. When you double-click on one of the groups you open the group settings where you can enter the blocks to add and their sequence within the group.

Protocols sequence: here you set the protocol sequence that will be implemented during the experiment (minutes can be dragged from the fields “Protocols” and “Protocol groups”).

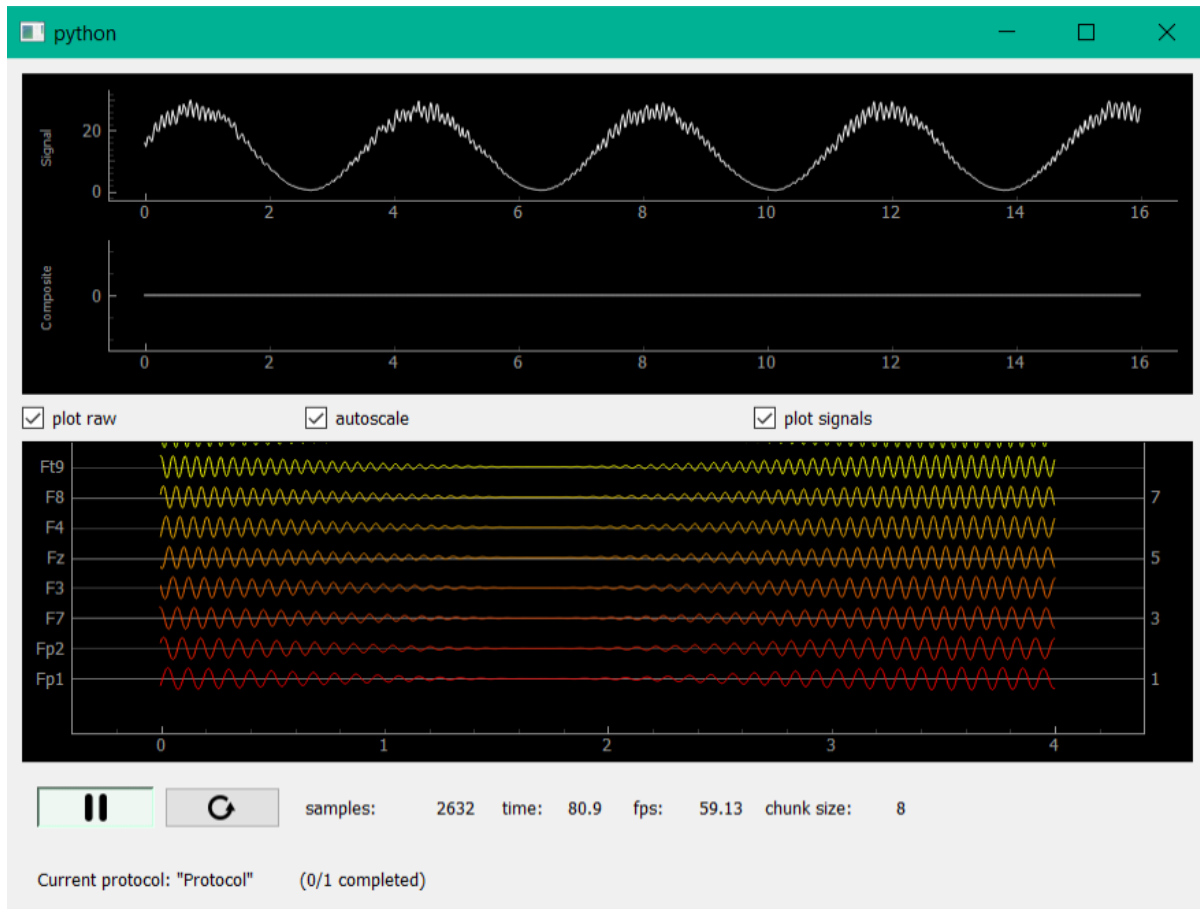
The **Start** button starts the experiment.

CHAPTER 2

Experiment

This module consist of two main windows (the window of the experimenter, the subject window) and one optional (signal manager), which appears only at the end of the protocols with exhibited tick “Open signal manager in the end (SSD, CSP, ICA)”.

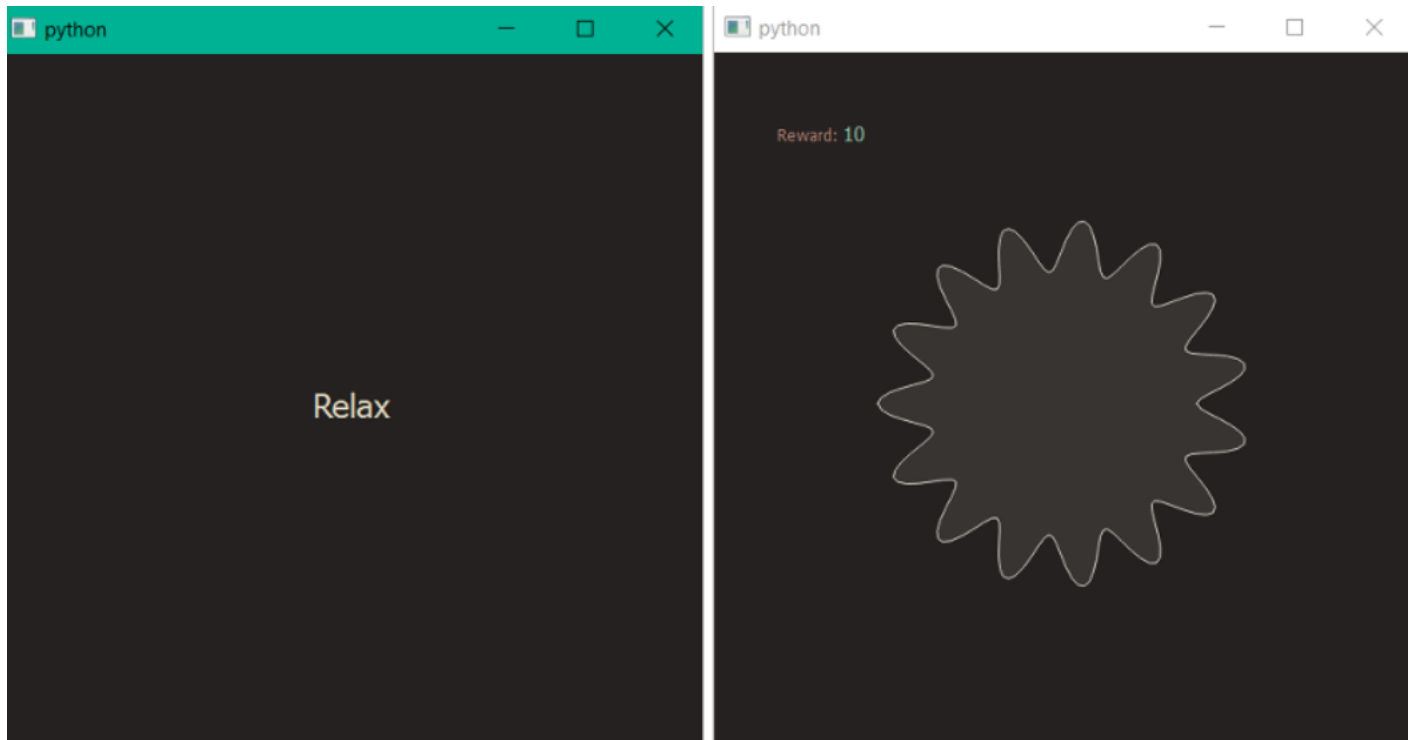
2.1 Composite signal settings



This window displays the processed, composite and raw signals, as well as some adjustments of its displaying (plot raw - enables / disables rendering of the raw signals; autoscale - enables / disables the automatic scaling of raw signal; plot signals - enables / disables rendering of processed and composite signals).

Below there are two buttons, like a “player” buttons for the experiment (start / pause button and the restart button) as well as information about the status of experiment: the number of samples received, the elapsed time, fps (frames per seconds) rendering, chunk size coming from the raw data stream. The bottom line displays information about the current record.

2.2 Subject window

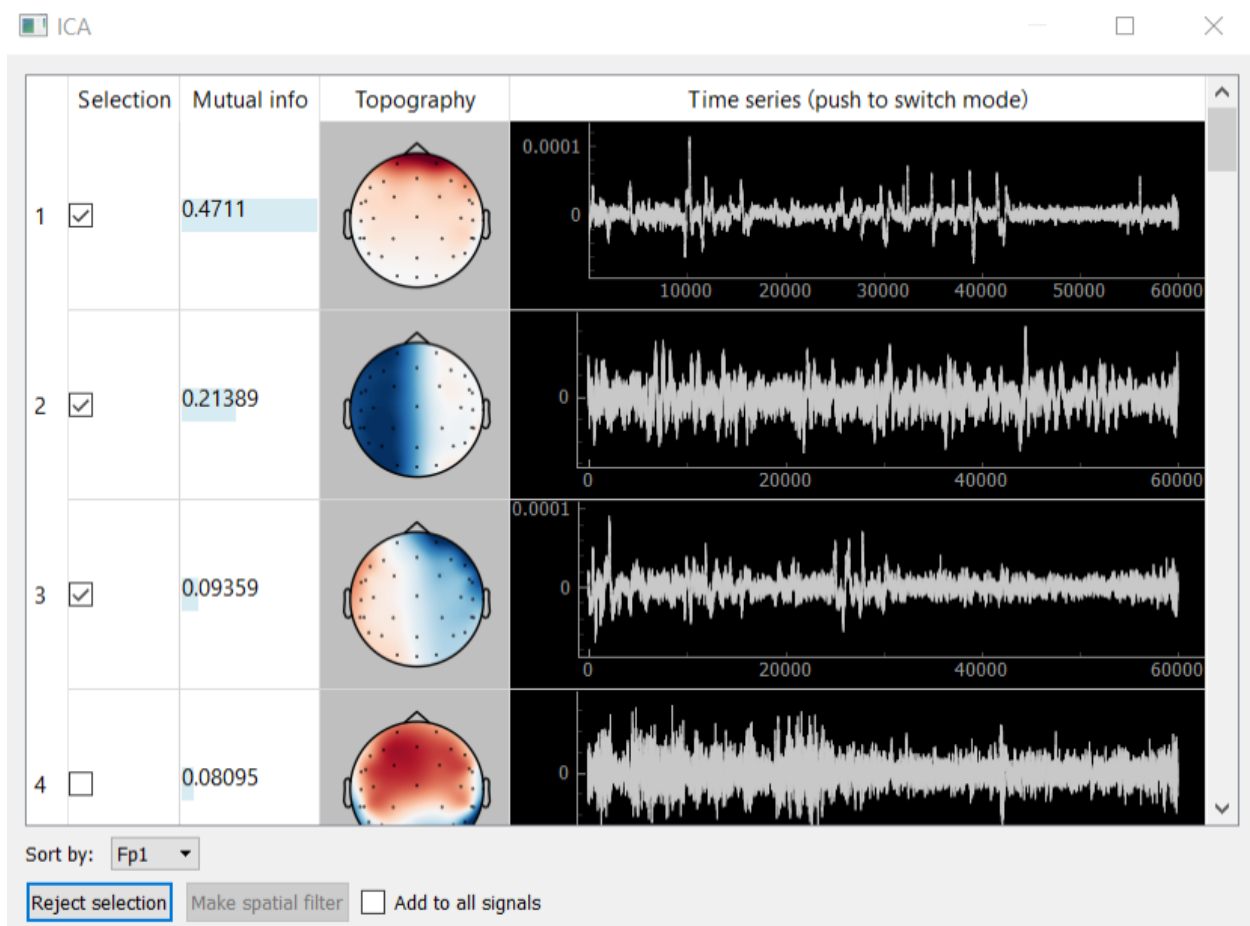


This window is displayed on the monitor for the subject. Depending on it can be drawn different objects of protocols settings.

CHAPTER 3

Data-driven filters (signals manager)

3.1 ICA



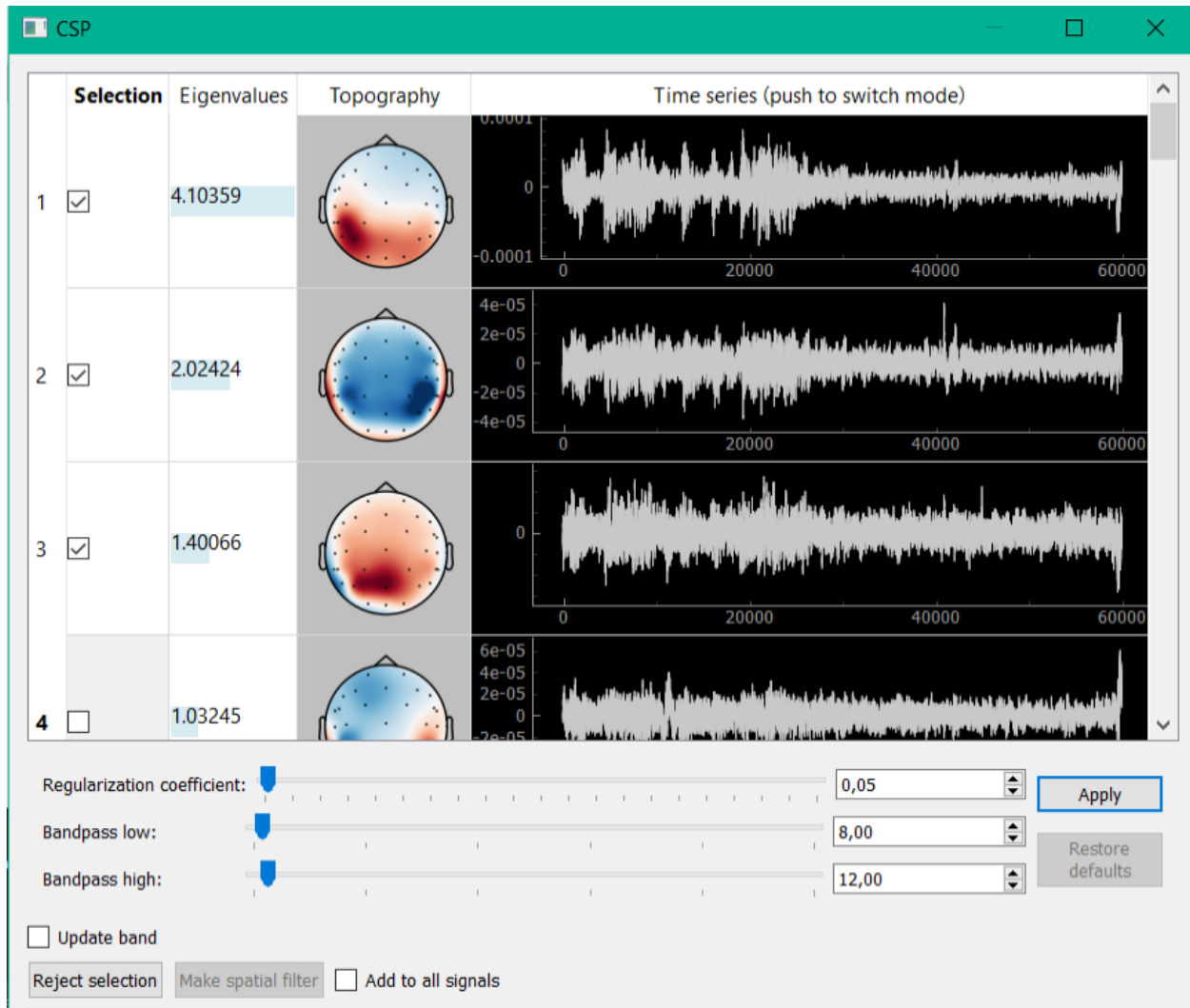
Method ICA (independent component analysis) is used to decompose the signal into independent components and is used for the isolation and removal of various kinds of artifacts. Designed graphical interface to visualize and select the component allows you to display the components themselves (column Time series), their spectra (column Time series after clicking on the column header) and topography (Topography column), sort them by mutual information with a certain channel. Calculations of the expansion are carried out using the ICA class of mne package. Buttons allow the bottom panel to add to the signal bandstop filter for the artifact removal filter or to isolate any of the components.

3.1.1 Example: removing eye artifacts

One of the examples of the ICA application is to create a spatial filter, which allows you to remove eye artifacts. The following describes a typical algorithm of this procedure:

1. Data collection. To do this in one of the protocols (e.g. call it “Filters”) give to the subject the task “look at the screen” during the protocol and protocol settings, set the checkbox “Open signal manager in the end (SSD, CSP, ICA)”.
2. ICA decomposition of the collected data. During the experiment, at the end of the protocol, at the time of opening the signals manager, necessary to select the signal to which you want to add eye filter, then click on the appropriate line of the open button from ICA column. Collected in the “Protocol Filters” multichannel data will be decomposed into n components by ICA (n - number of channels). Next window appears to visualize data components; in Figure 8 you could see an example.
3. Selection of the artifact components. Further, necessary to note rows, those correspond to the components, which are observed eye artifacts. In the example of Figure 8 only one such component (first component).
4. Creating a spatial bandstop filter. After selecting the components, that you want to exclude from the data, you must click “Reject selection”. For the selected signals will be created a spatial filter, which removes the selected ICA components. If you check the checkbox “add to all signals”, then the filter will be added to all derived signals.
5. Continuation of the experiment. After closing signals manager in the real-time for selected derived signals will be applied a spatial filter, which removes the selected ICA components. Thus, in this example will be deleted subjects’ eye artifacts, during “Filters” protocol.

3.2 CSP



Interface for CSP (common spatial pattern) decomposition looks similarly. This method allows you to select components with the highest ratio of signal power to the two windows (usually the first window corresponds to the first half of the recording, in which subject stays with closed eyes, in the second - open). A key part of the algorithm is the solution of the generalized eigenvalue problem. Algorithm parameters handed down and can be adjusted by sliders at the bottom of the window.

3.2.1 Example: allotment and removal of the alpha activity.

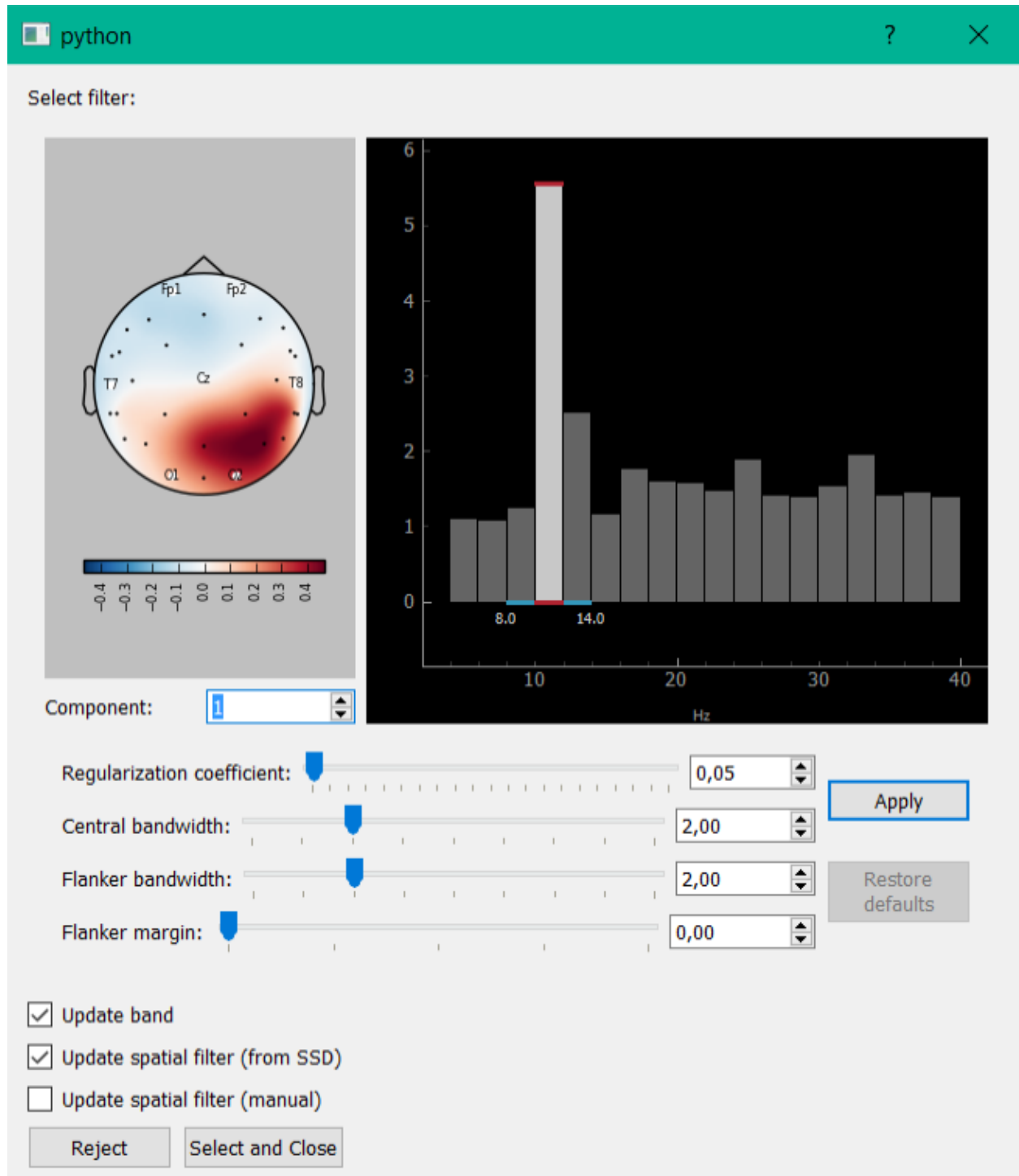
This example is relevant for experiments in training the mu rhythm. As mu and alpha rhythms are in the same frequency range, only frequency filtering (specified in the properties of the derived signal) is not enough to highlight the mu rhythm. With CSP analysis it is possible to create a spatial filter, which reduces the contribution of alpha activity in the derived signal. The following describes a typical algorithm of this procedure:

1. Data collection. Allotment of the alpha activity is possible with the help of the following protocol: in the first half of a protocol give the task to the subject to seat with his/her eyes closed, and in the second - to open (in the settings of the protocol can be set checkbox “Add half time extra message” to change the message displayed on

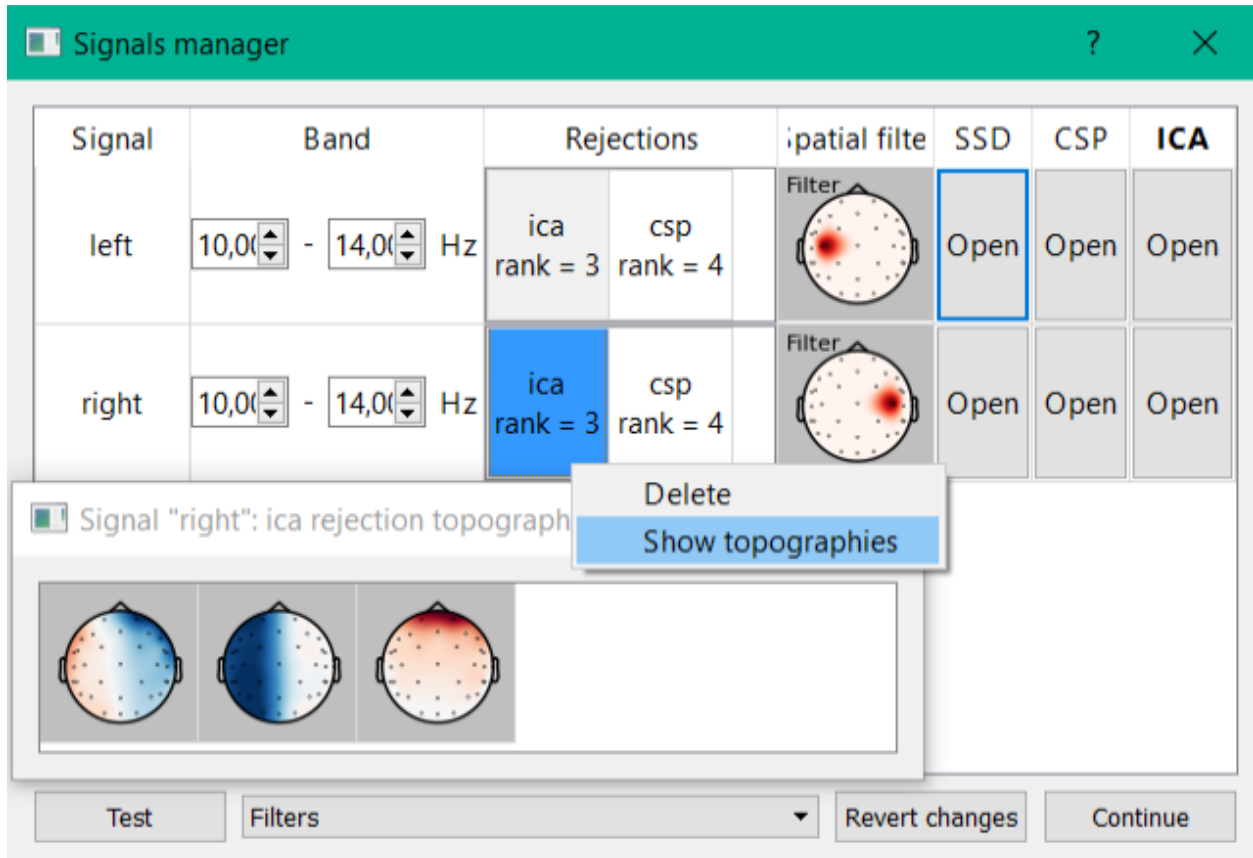
the screen at the moment when first half was completed). These two contrasting states are able to allocate space alpha activity in CSP algorithm.

2. CSP decomposition of the collected data. During the experiment, at the end of the protocol, at the time of opening the signals manager, necessary to select the signal to which you want to add to filter alpha activity, then click on the appropriate line of the open button from ICA column. Collected in the “Protocol Filters” multichannel data will be decomposed into n components by CSP (n - number of channels). Next window appears to visualize data components; in Figure 9 you could see an example.
3. Selection a component corresponding to the alpha activity. Further, it is necessary to note row corresponded to components, in which observes alpha activity. In the example of Figure 9 such component 3 (1.3 parts).
4. Creation spatial bandstop filter. After selecting the components that you want to exclude from the data, you must click “Reject selection”. For the selected signals will be created a spatial filter, which removes the selected CSP components. If you check the checkbox “add to all signals”, then the filter will be added to all derived signals.
5. Continuation of the experiment. After closing signals manager in the real-time for selected derived signals will be applied a spatial filter, which removes the selected CSP components with alpha activity.

3.3 SSD



In the SSD method (spatio-spectral decomposition) decomposition also occurs by solving the generalized eigenvalue problem. This method allows you to select components with maximum signal power ratio for two different frequency bands (the central strip and the adjacent frequency). Implemented the possibility to partitioning the signal into strips, control their width. At the bottom, you allow to add to the bandstop filter signal from the selected filter components, or to isolate any of the components.



This window is used to adjust the spatial filters using a variety of methods. It is also possible to edit the temporary filter options. When you click the “Open” button for the signal of the matching line, you will open a window for the corresponding column of the analysis (SSD, CSP, ICA). Each type of analysis produces the decomposition of raw signal into several components from which you can build spatial filter for selection or rejection corresponding components. The following describes the window for different types of analysis.

Occipital alpha rhythm neurofeedback

The simplest example of an experiment using NFB Lab is alpha rhythm training in the neurofeedback paradigm. As a feedback signal, the derived signal was used with the following settings: the spatial filter is determined during the experiment, the frequency filtering band is 9-11 Hz, the envelope detector is a sequence of complex demodulation, the 4th-order Butterworth filter is used in the specified band, and smoothing by the Savitzky–Golay filter of the 2nd order with a window width of 151 samples. At a sampling rate of 250 Hz, these settings, due to the causality of the used filters, introduce a delay of 131 ms. When evaluating the envelope versus non-causal filtering, the correlation of the reconstructed envelope versus the envelope obtained by the Hilbert transform is 0.7. The experiment is divided into 19 blocks

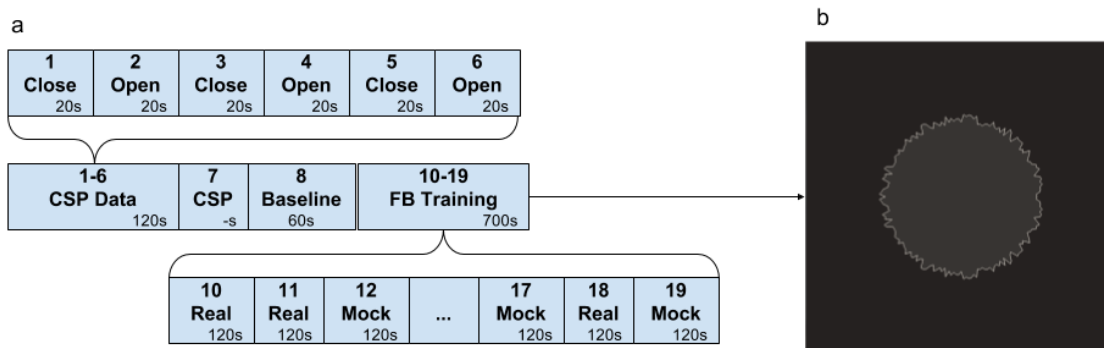


Fig. 1: Fig. 1. Experiment sequence (a) and feedback stimulus (b).

In the first part of the experiment (blocks 1-6 “Open” and “Close”) EEG data are recorded in states with eyes open and closed, respectively. Further, in block 7 (“CSP”), using the module “Filter settings by collected data”, a spatial filter corresponding to the alpha rhythm is selected from the functional samples obtained in blocks 1-6. For this, CSP analysis of “Open” vs. “Close” blocks is used. The use of CSP is motivated by the assumption that the source generating the alpha rhythm works in a synchronous mode with the eyes closed and de-synchronizes when the eyes are open.

Figure 2 shows the properties of the selected CSP component, namely, a portion of the time series with a pronounced

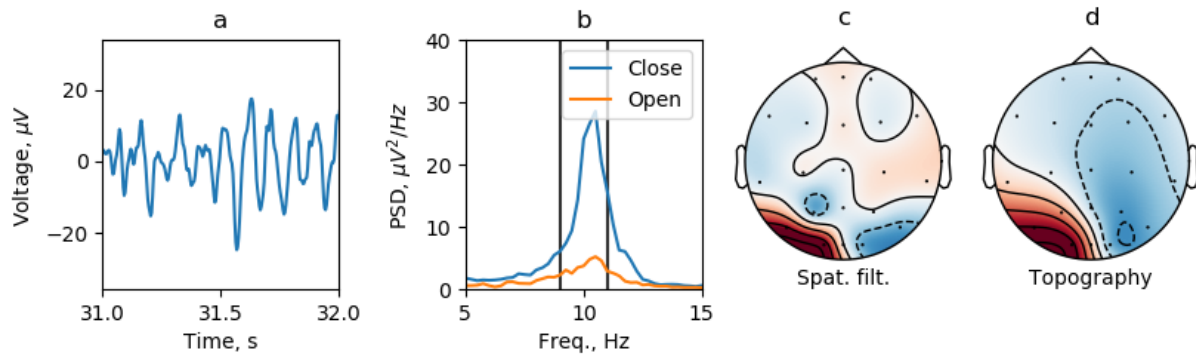


Fig. 2: Fig. 2. Properties of the feedback signal.

alpha spindle (a), the spectrum of the component in two states Open and Close (b), a spatial filter (c), and topography (d). Next, the resting state (Baseline) is recorded to calculate the z-score of signal statistics (mean and standard deviation). These statistics are needed to correctly display the reinforcing stimulus (b) in the future. This stimulus is an inflating circle and assumes that the mean of the selected signal is 0 and the standard deviation is 1. The roughness of the border of the circle is inversely proportional to the input signal. After that, training is carried out in the NFB paradigm: 10 sessions of 2 minutes without a break, among which half of the sessions are based on the presentation of a false (Mock) feedback (not corresponding to the current activity of the brain), and the second half is real feedback (Real).

The experiment design is available at the link (https://github.com/nikolaims/nfb/blob/master/tests/designs/alpha_nfb_settings.xml).

Results file structure

- **protocol0** (group; initial stats of signals)
- **protocol1** (group; recorded data and signals stats after the first protocol, see *protocol<k>*)
- ...
- **protocol<k>** (group; recorded data and signals stats after the <k>-th protocol)
 - **raw_data** (dataset; raw data recordings except *ignored* channels)
 - **raw_other_data** (dataset; *ignored* channels, for example , for example reference channel)
 - **reward_data** (dataset; reward dynamics time series)
 - **signals_data** (dataset; signals data recordings)
 - **signals_stats** (group; signals stats for every signal)
 - * **signal_name1** (group; stats for signal “signal_name1”, see **signal_name<m>**)
 - * **signal_name2** (group; stats for signal “signal_name2”, see **signal_name<m>**)
 - * ...
 - * **signal_name<m>** (group; stats for signal “signal_name<m>”)
 - **mean** (dataset; mean value of signal “signal_name<m>”)
 - **std** (dataset; std value of signal “signal_name<m>”)
 - **bandpass** (dataset; bandpass for signal “signal_name<m>”, for Derived signal only)
 - **spatial_filter** (dataset; spatial filter dataset for signal “signal_name<m>”, for Derived signal only)
 - **rejections** (group; rejections for signal “signal_name<m>”, for Derived signal only)
 - rejection1** (dataset; the first rejection dataset for signal “signal_name<m>”)
 - rejection2** (dataset; the second rejection dataset for signal “signal_name<m>”)

...

rejection<j> (dataset; the <j>-th rejection dataset for signal “signal_name<m>”)

CHAPTER 6

Requirements

You can find all python dependencies here:

[Conda requirements](#)

[PIP requirements](#)