NFB Lab Documentation

Release 0.1

Nikolai Smetanin

Mar 07, 2018
# Contents

1 Experimental designer
   1.1 Derived signal settings .................................................. 4
   1.2 Composite signal settings ............................................... 5
   1.3 Protocol settings ....................................................... 6

2 Experiment
   2.1 Composite signal settings ............................................... 10
   2.2 Subject window .......................................................... 11

3 Data-driven filters (signals manager)
   3.1 ICA ................................................................. 14
   3.2 CSP ................................................................. 16
   3.3 SSD ................................................................. 18

4 Results file structure ..................................................... 21

5 Requirements .............................................................. 23
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:
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 of file (created LSL signal playback stream recorded in the file during the previous experiments), FieldTripBuffer (connection for FieldTripBuffer Protocol).

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

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

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

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.

Signals list: 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

![Derived signal settings](image)

Signal Settings include:

Name: name of the signal.

Spatial filter: a spatial filter (a current signal sample is computed as a weighted sum of the raw samples, and that weights are construct the spatial filter)

Disable spectrum evaluation: disables the calculation of FFT and signal amplitude.

Bandpass: frequency filter border. FFT window size: the size of the window in samples used for FFT.

Smoothing factor: factor used in the exponential smoothing.

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

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 list**: 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

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>name of the protocol.</td>
</tr>
<tr>
<td>Duration [s]</td>
<td>the duration of the protocol in seconds.</td>
</tr>
<tr>
<td>Open signal manager in the end (SSD, CSP, ICA)</td>
<td>opens at the end of protocol signaling manager where, by using different methods (SSD, CSP, ICA), computed spatial filters signals.</td>
</tr>
<tr>
<td>Update statistics</td>
<td>if this checkmark is exhibited, then, at the end of the protocol will be recalculated the average (mean) and standard deviation (std) of the signal, and subsequently from the signal will be subtracted mean and then the signal be divided by std.</td>
</tr>
<tr>
<td>Type</td>
<td>type of the protocol. Available 3 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.), CircleFeedback (on the subjects screen will be displayed, with jagged edges, the 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).</td>
</tr>
<tr>
<td>Blink duration [ms]</td>
<td>only available for ThresholdBlink protocol type (see above).</td>
</tr>
<tr>
<td>Blink threshold</td>
<td>only available for ThresholdBlink protocol type (see above).</td>
</tr>
</tbody>
</table>
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 protocol which was previously, its number is indicated).

Message: Here you specify the message that will be displayed and presented to 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).

Half time extra message: Additional message.

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.

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

The Start button starts the experiment.
This module consists 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.
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,
t 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.
CHAPTER 4

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, 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>”)

21
... 

\textit{rejection}^{<j>}$ (dataset; the $<j>$-th rejection dataset for signal “signal\_name$^m$”)


CHAPTER 5

Requirements

Software is written in python (version 3.4) and used the following packages:

- PyQt4 - for creating graphical user interfaces;
- pyqtgraph and matplotlib - to draw graphs;
- pylsl - to connect various devices, which support work with Lab Stream Layer protocol;
- numpy - to store matrices and perform basic operations on them;
- scipy - to use various kinds of filters and performing a fast Fourier transform;
- sympy - for symbolic computations (for composite signals);
- mne - for processing the neurophysiological teme series;
- h5py - to save the results of experiments and recorded signals;