December 14th 2022

OpenViBE: an open source BCI software suite

PART 2 – Designing BCIs with OpenViBE

Arthur Desbois, Marie-Constance Corsi
ARAMIS team, Paris Brain Institute

PART 2 – Designing BCI systems with OpenViBE

2.1 – The Motor Imagery Paradigm

Brain Computer Interface: how?

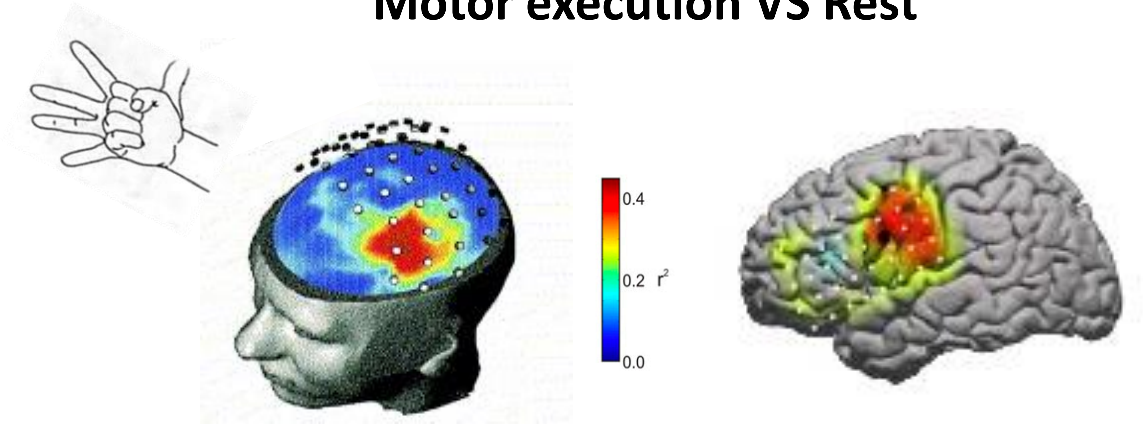
- **Underlying idea:**

Taking advantage of a **neurophysiological phenomenon** to establish a communication between the brain and the computer

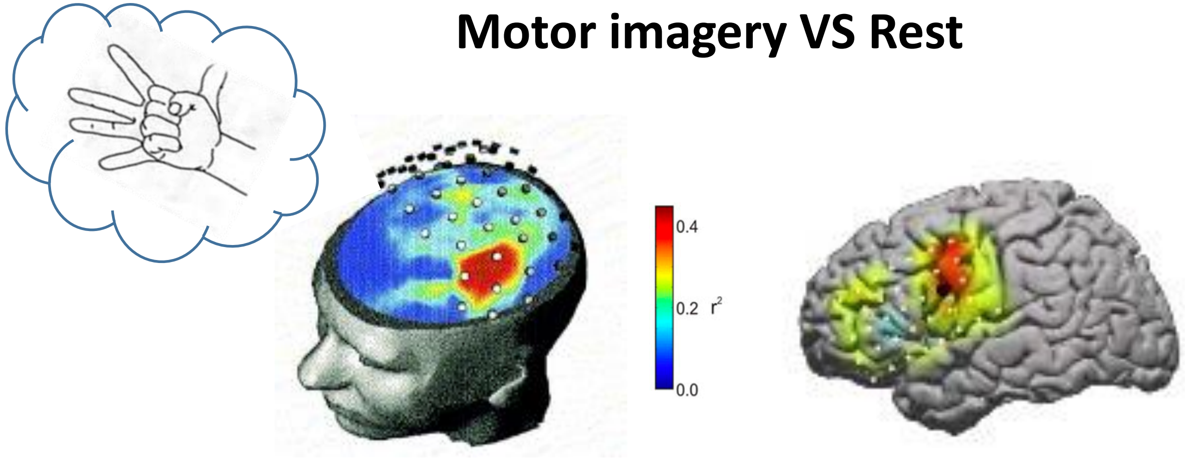
Illustration with **Motor imagery-based BCI**

Motor Imagery - Observations

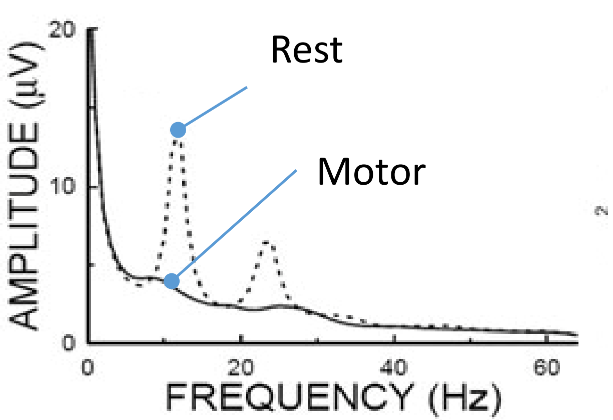
Motor execution VS Rest



Motor imagery VS Rest



Power decrease

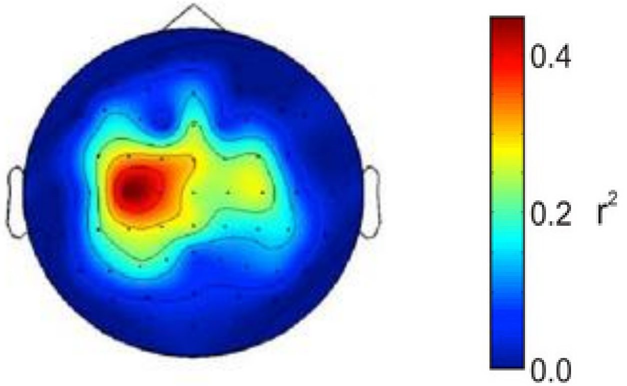


Desynchronization effect
(Pfurtscheller et al, 1999)

Motor Imagery – Mu-Beta Rhythm

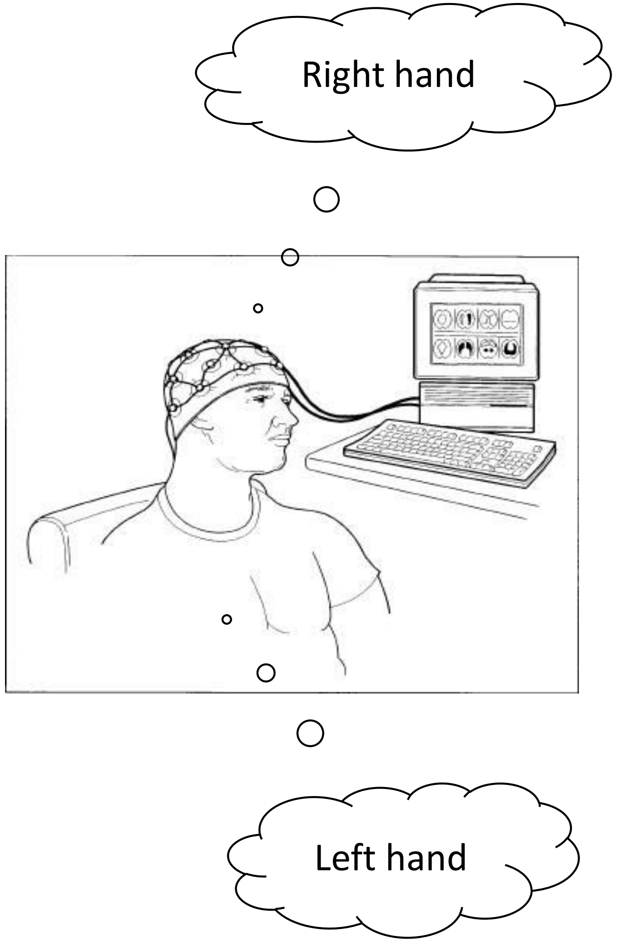
- ## Behavioral properties
  - Movement / preparation for movement : Event-related desynchronization (ERD) (Pfurtscheller, G, Lopes da Silva, FH, 1999)
  - With relaxation/post-movement period : ERS

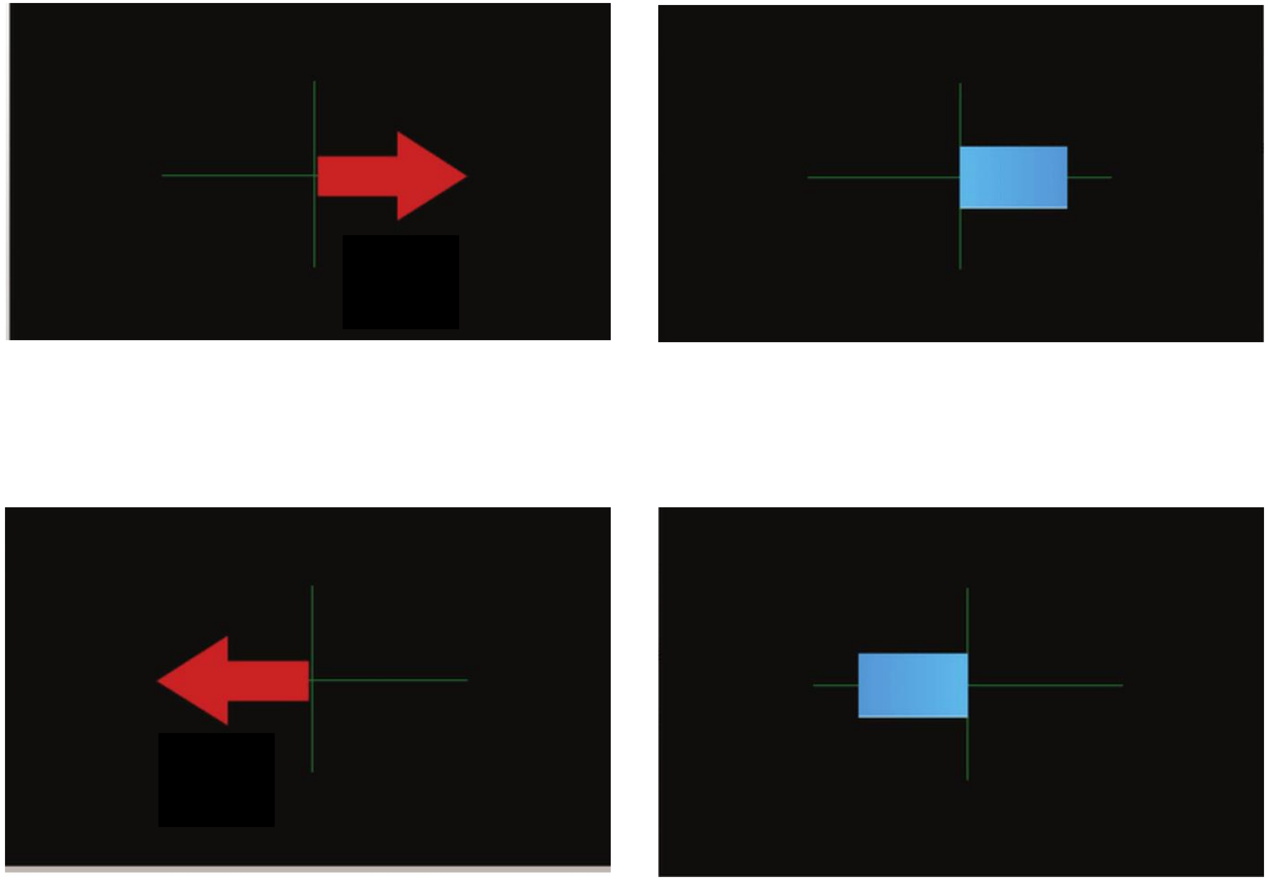
- ## Why using it in BCI ?
  - Mu/Beta activity modulation by motor-imagery, a way to communicate
  - Use of power spectra
  - To establish this communication :
    - Spatial selection
    - Frequency selection



Illustrations from BCI2000 website

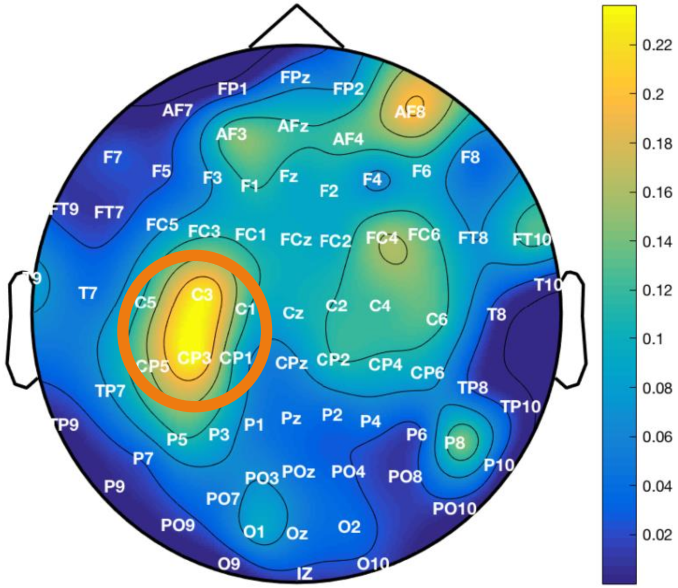
Motor Imagery - In Practice…



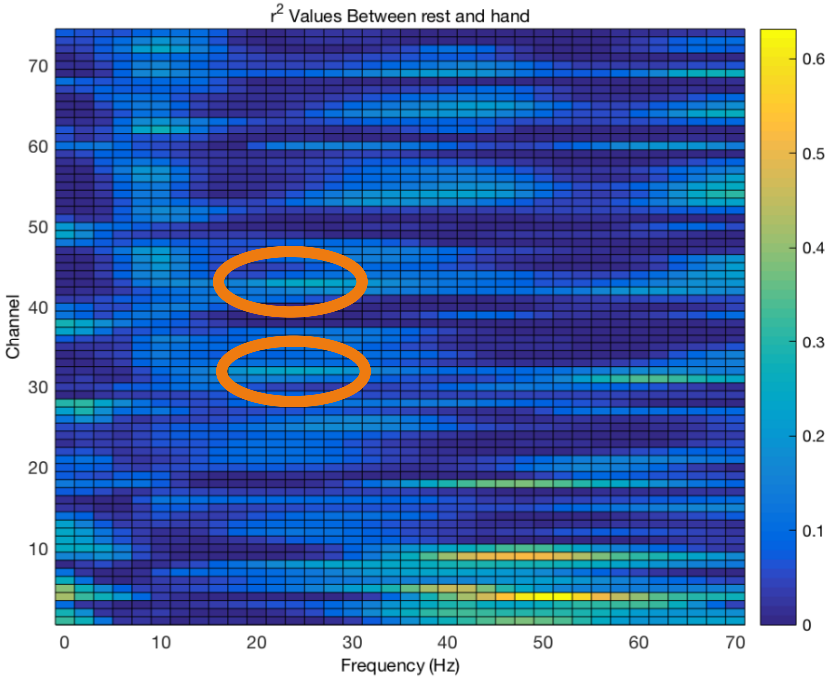


Motor Imagery – In Practice…

@ 10Hz

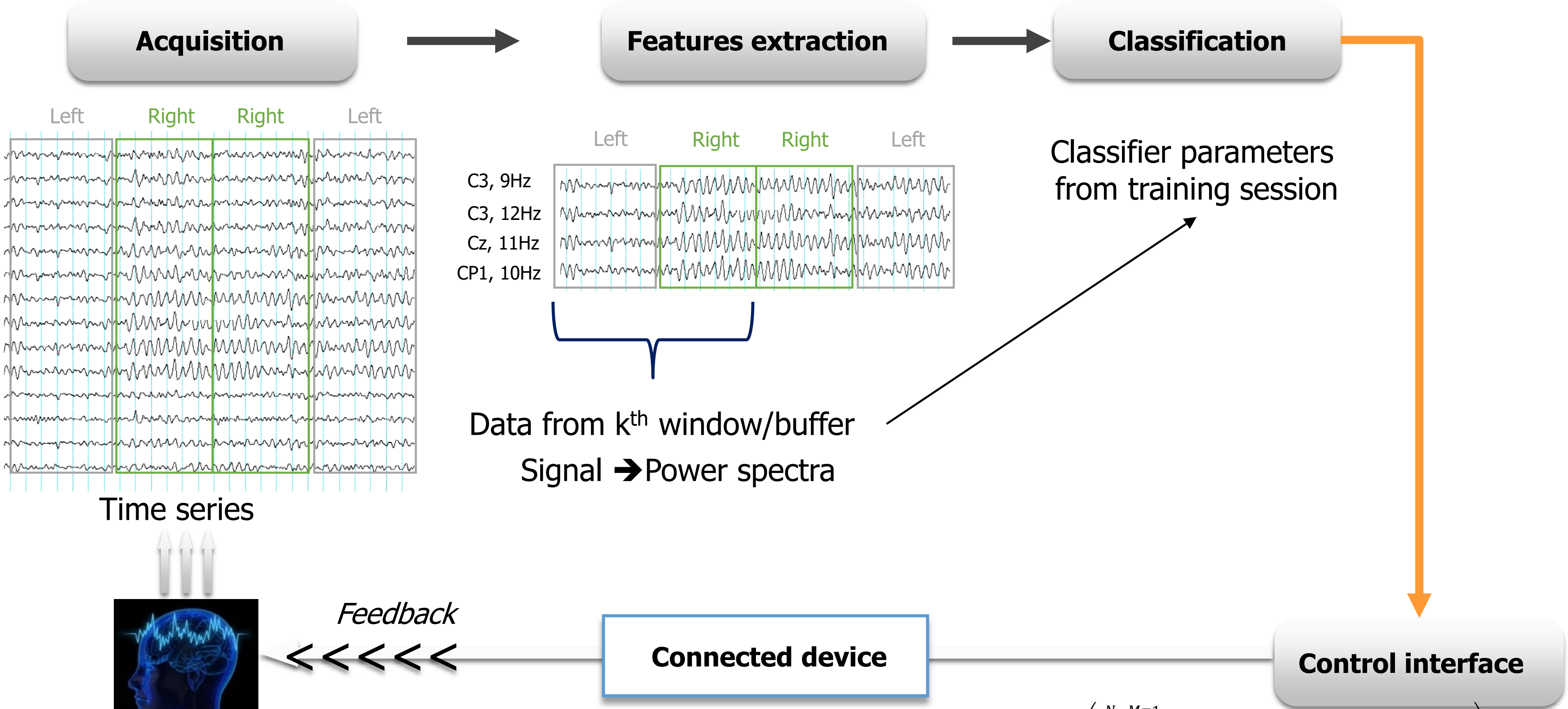


r²



r²

Motor Imagery - In Practice…



Acquisition → Features extraction → Classification

Time series

Data from kth window/buffer

Signal ➔Power spectra

Classifier parameters from training session

Control interface

Connected device

Feedback

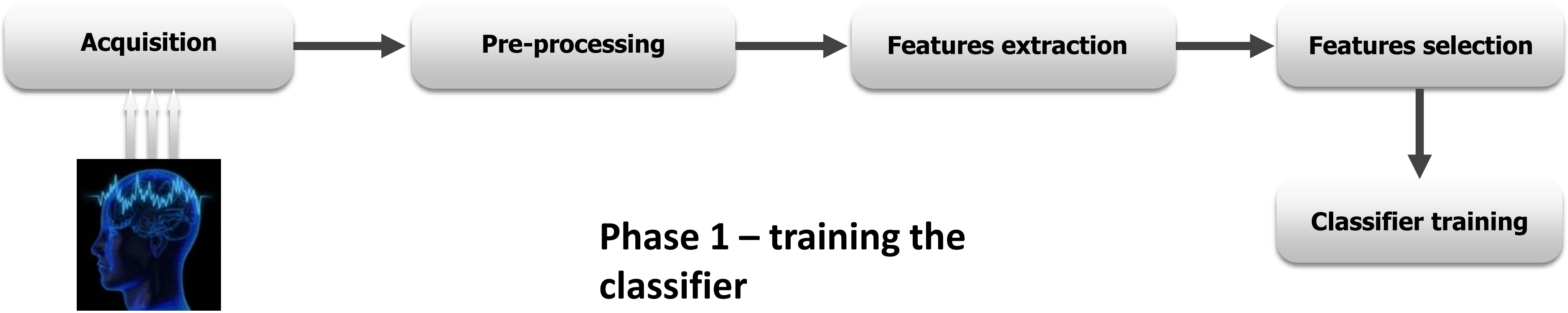
<<<<<

$$cursor\ velocity = \left(\sum_{i=1}^{N} \sum_{j=1}^{M=1} Features_{i,j} Classifier_{i,j,k} - Offset \right) x\ Gain$$

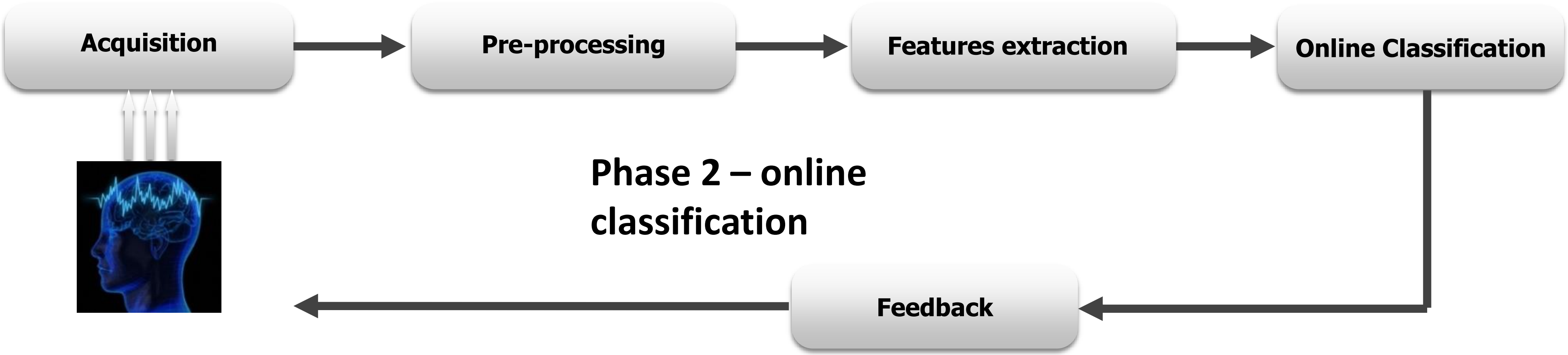
PART 2 – Designing BCI systems with OpenViBE

2.2 – Technical Demo

Motor Imagery Protocol

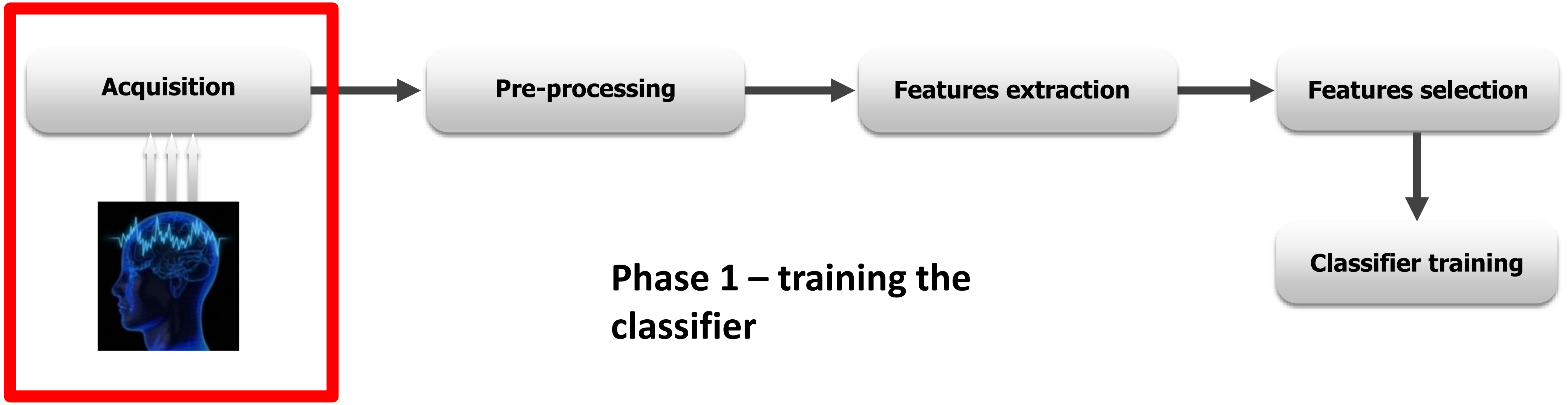


Phase 1 – training the classifier

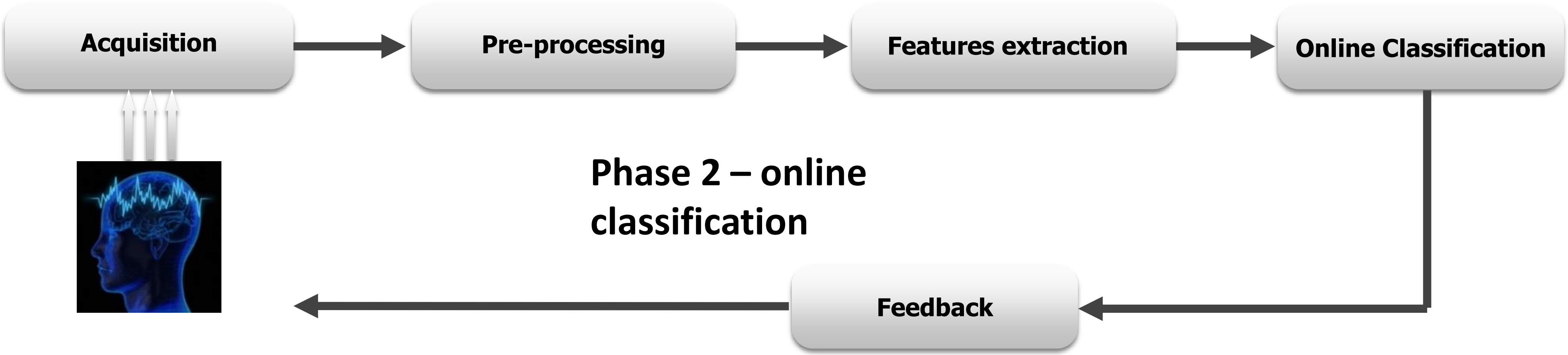


Phase 2 – online classification

Motor Imagery Protocol

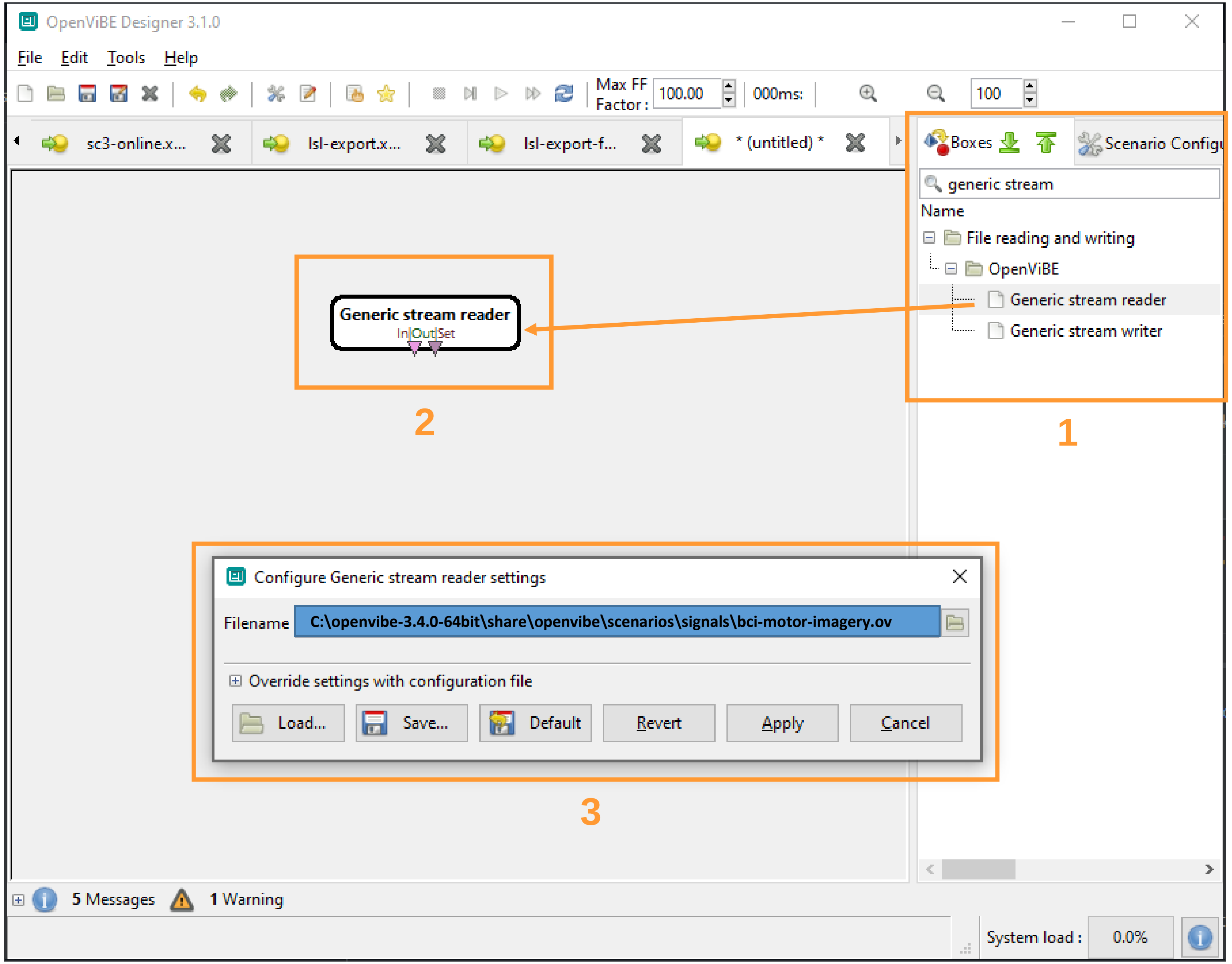


Phase 1 – training the classifier



Phase 2 – online classification

Step 1 - Warmup! Simple I/O & Display



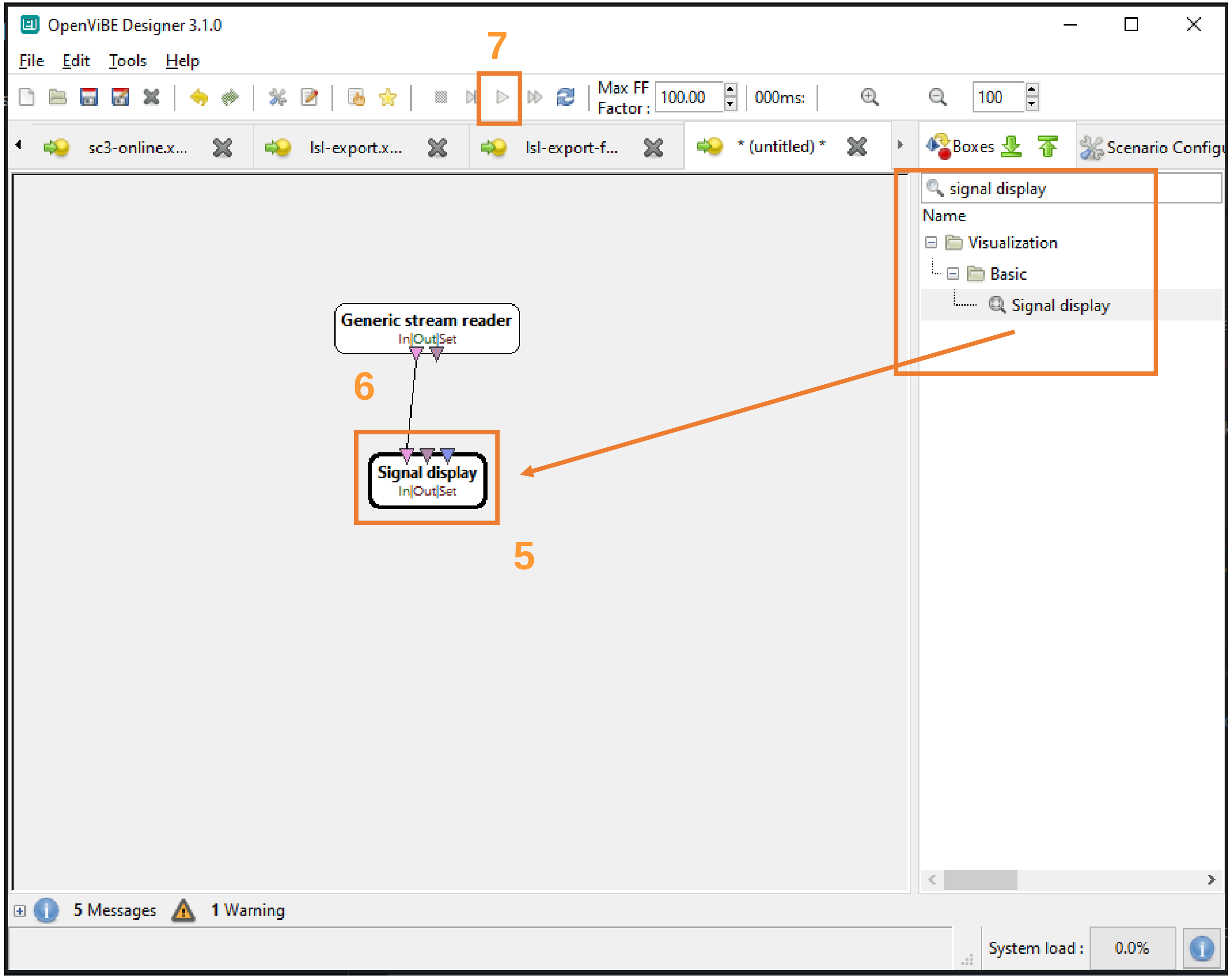
Filename: C:\openvibe-3.4.0-64bit\share\openvibe\scenarios\signals\bci-motor-imagery.ov

1. **Search** for "generic stream reader" in boxes list

2. **Drag & drop** to designer window

3. Set **filename** with browser `C:\openvibe-3.4.0-64bit\share\openvibe\scenarios\signals\bci-motor-imagery.ov`

Step 1 - Warmup! Simple I/O & Display



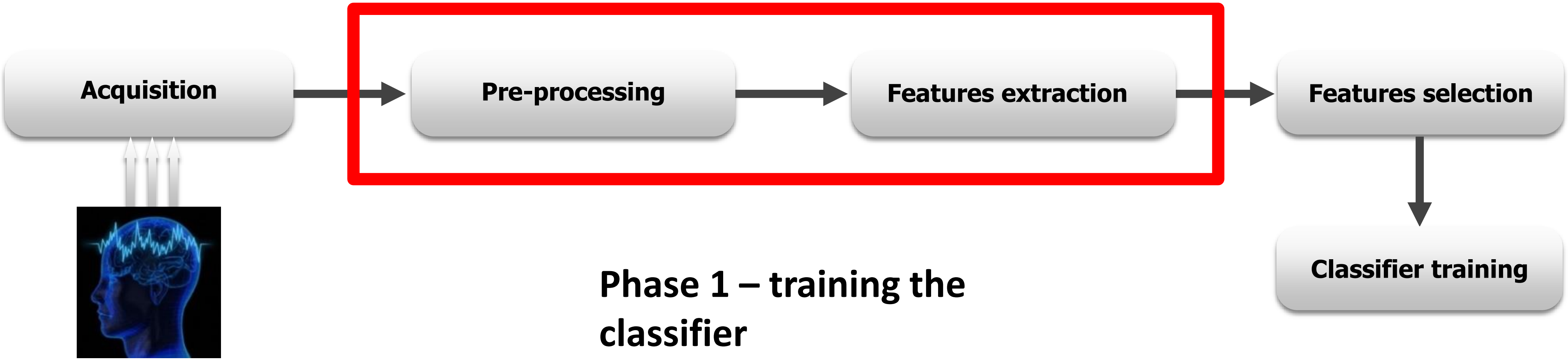
4. **Search** for "signal display" in boxes list

5. **Drag & drop** to designer window

6. **Drag & drop** connection between boxes, using the "signal" stream type

7. Press **Play**!

Motor Imagery Protocol



Acquisition

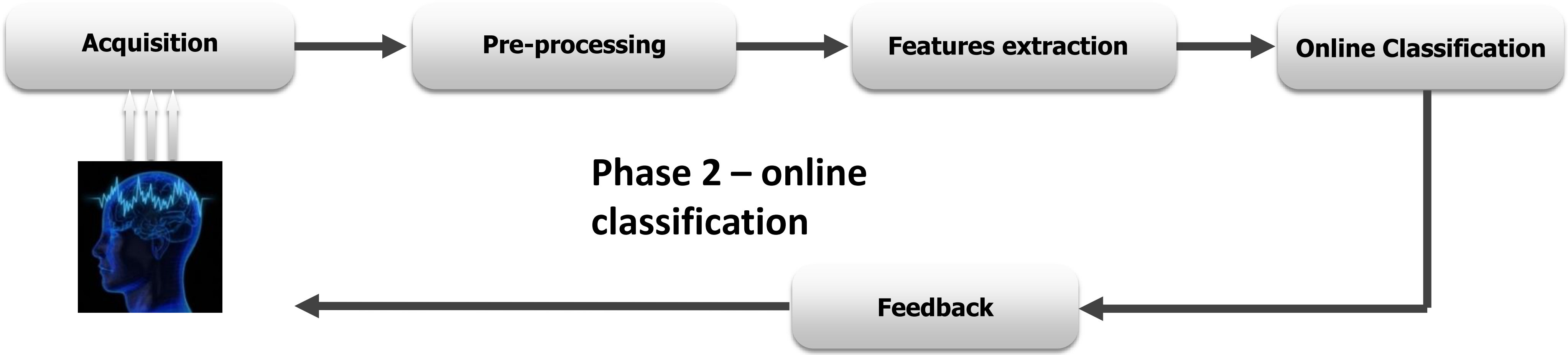
Pre-processing

Features extraction

Features selection

Classifier training

Phase 1 – training the classifier



Acquisition

Pre-processing

Features extraction

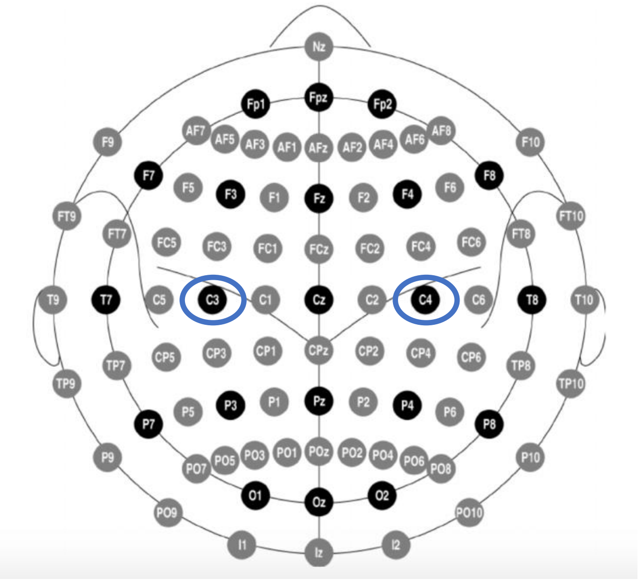
Online Classification

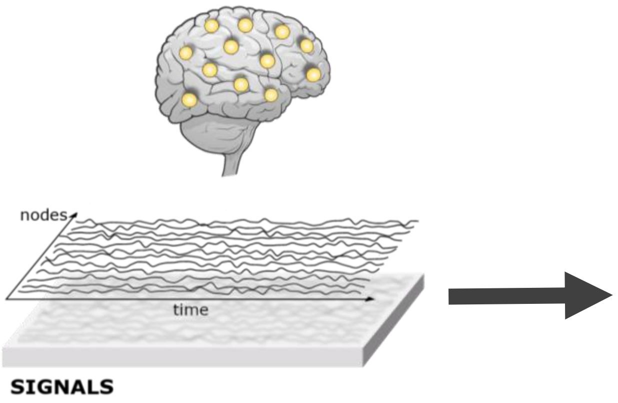
Feedback

Phase 2 – online classification

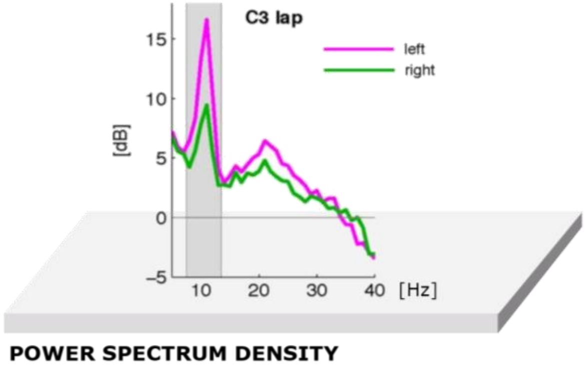
Step 2 - DSP, Visualization

- **Features to extract (recap)**
  - Power Spectra
  - Sensorimotor Area
  - Mu/Alpha (8-12Hz) &/or Beta (14-29Hz)


Nz
Fp1 Fpz Fp2
F9 AF7 AF5 AF3 AF1 AFz AF2 AF4 AF6 AF8 F10
F7 F5 F3 F1 Fz F2 F4 F6 F8
FT9 FT7 FC5 FC3 FC1 FCz FC2 FC4 FC6 FT8 FT10
T9 T7 C5 C3 C1 Cz C2 C4 C6 T8 T10
TP9 TP7 CP5 CP3 CP1 CPz CP2 CP4 CP6 TP8 TP10
P7 P5 P3 P1 Pz P2 P4 P6 P8
P9 PO7 PO5 PO3 PO1 POz PO2 PO4 PO6 PO8 P10
PO9 O1 Oz O2 PO10
I1 Iz I2



(Gonzalez-Astudillo et al, 2020)

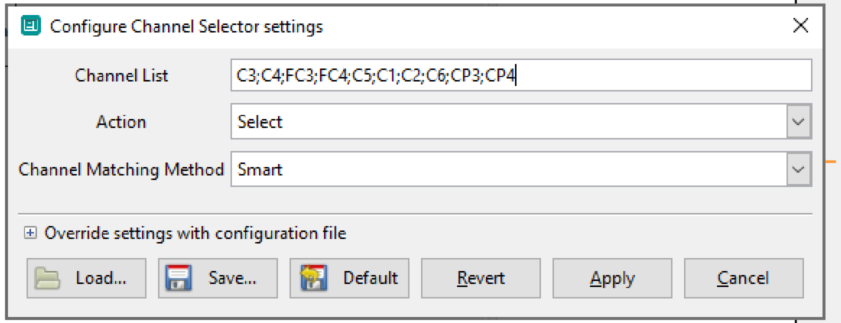


(Maeder et al., 2012)

Step 2 – DSP, Visualization

- OpenViBE can manage multiple signal streams in parallel
  - In our example, **1 channel = 1 EEG electrode**
  - We will load a pre-recorded signal file, that used 11 labeled electrodes and consider only a sub-set of those electrodes.

Step 2 - DSP, Visualization



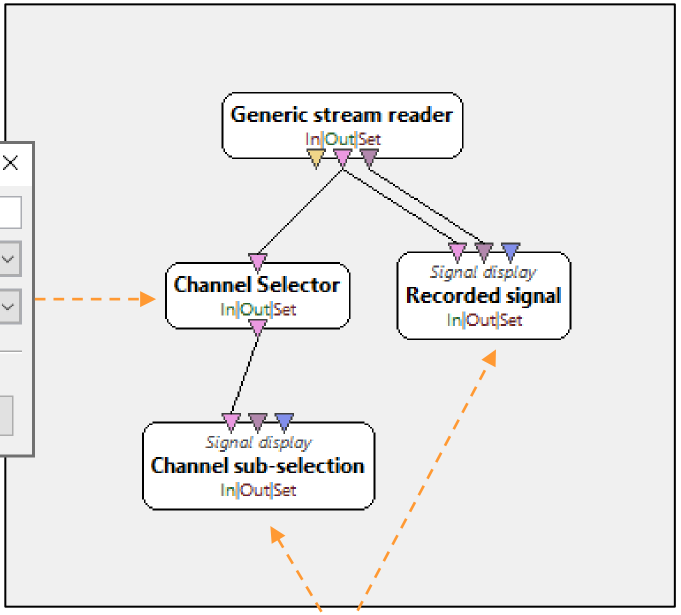
Configure Channel Selector settings

Channel List: C3;C4;FC3;FC4;C5;C1;C2;C6;CP3;CP4

Action: Select

Channel Matching Method: Smart

⊞ Override settings with configuration file



Note:
Right click + Rename can be useful when designing a scenario…

Display windows will use « renamed » label

1. **Import a Generic Stream Reader Box,** set the filename to: `<install folder>\share\openvibe\scenarios\signals\bci-motor-imagery.ov`

2. **Import a Channel Selector Box,** link the boxes together

3. **Set the selection to:**

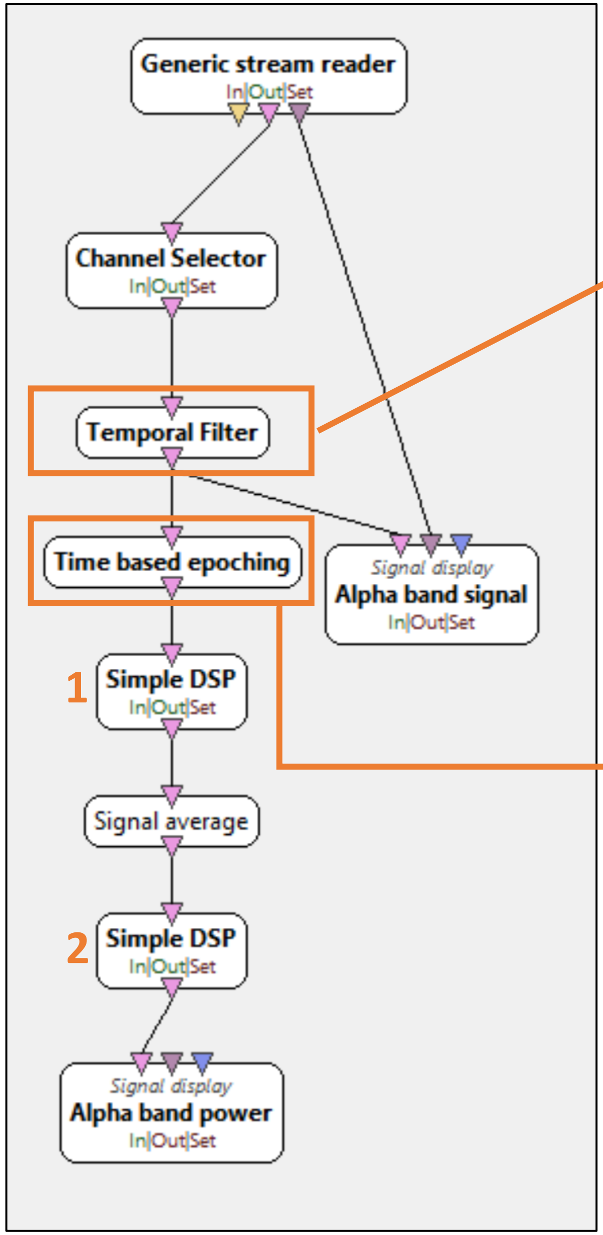
   C3;C4;FC3;FC4;C5;C1;C2;C6;CP3;CP4

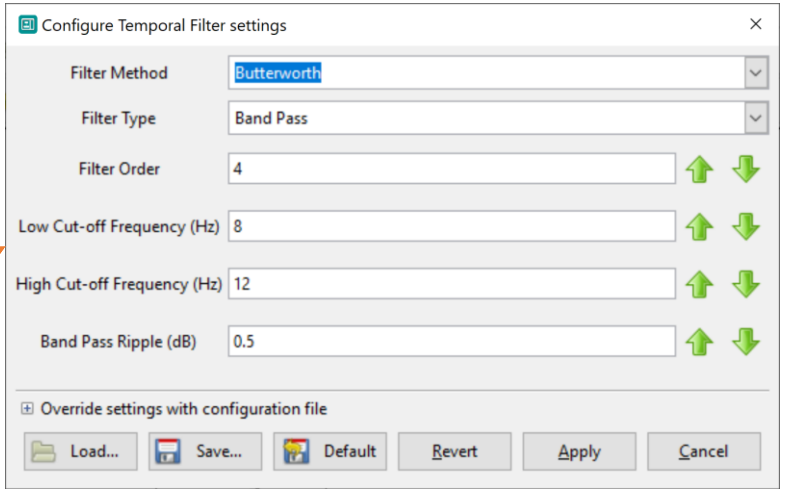
4. Display the output

Step 2 – DSP, Visualization

- We now need to compute power spectra for the alpha & beta bands, for the selected electrodes

Step 2 - DSP, Visualization




Configure Temporal Filter settings

Filter Method: Butterworth

Filter Type: Band Pass

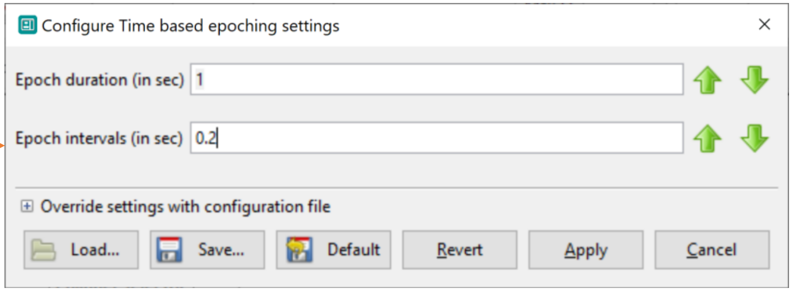
Filter Order: 4

Low Cut-off Frequency (Hz): 8

High Cut-off Frequency (Hz): 12

Band Pass Ripple (dB): 0.5

⊞ Override settings with configuration file


Configure Time based epoching settings

Epoch duration (in sec): 1

Epoch intervals (in sec): 0.2

⊞ Override settings with configuration file

1. **“Temporal Filter”**, set to [8;12]Hz
2. **“Time based epoching”**, of 1s, every 0.2s (overlapping windows of signal for power computation)
3. **DSP #1**

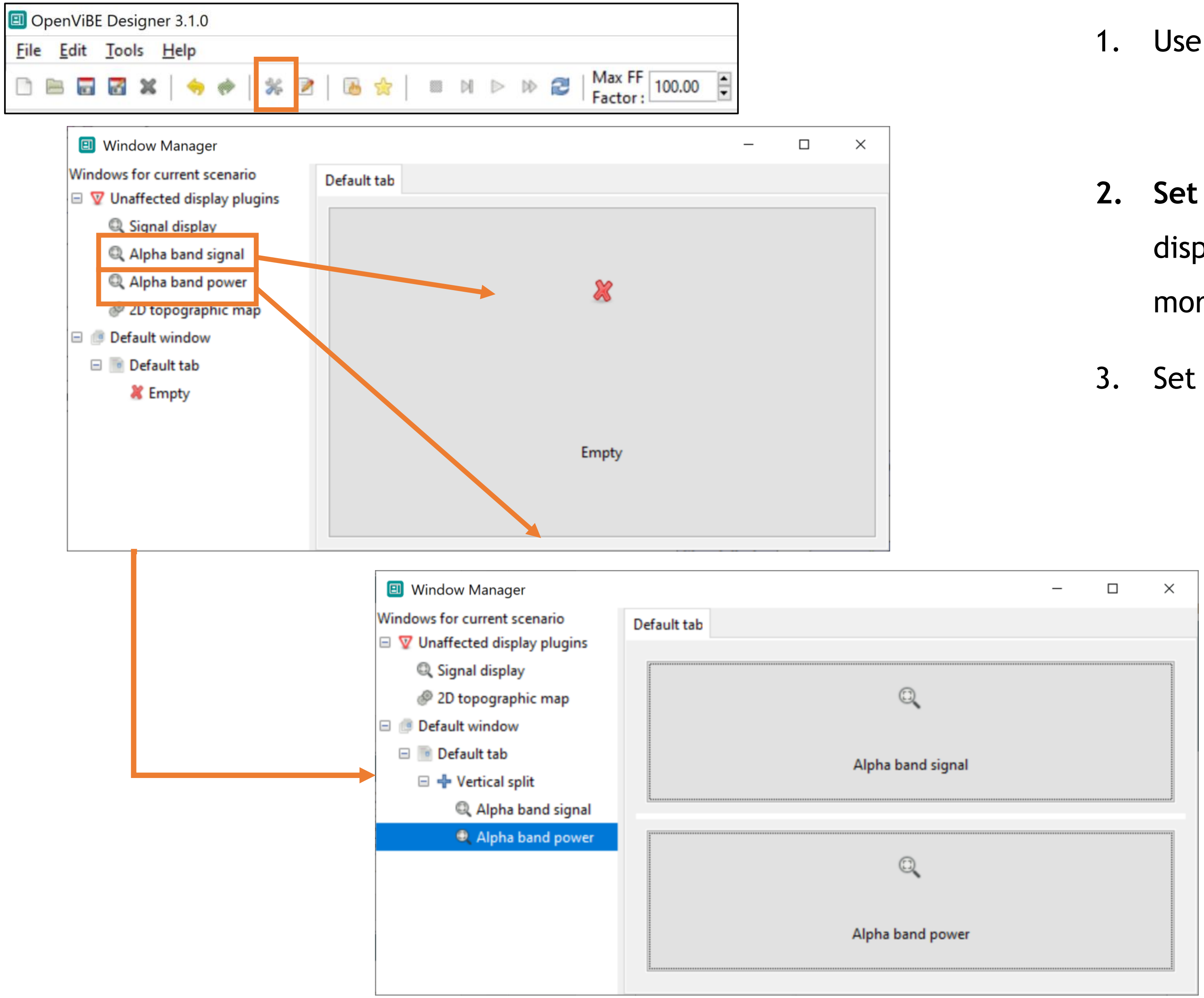
   formula `x*x`
4. **Average**

   (averaging x^2 across a window of 1s)
5. **DSP #2**

   formula `log10(1+x)`

   => log10(1+avg(x^2))
6. Add Displays and rename them to “alpha band power” and “alpha band signal”

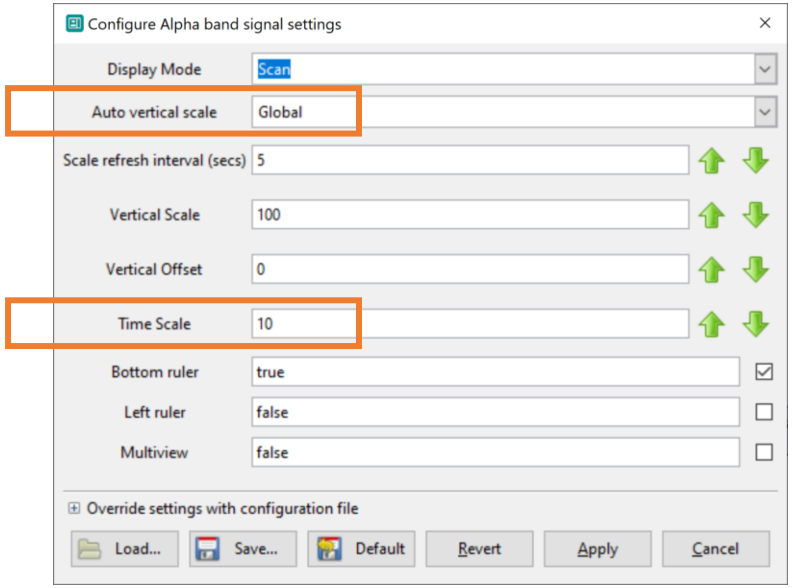
Step 2 - DSP, Visualization



1. Use the “widget reordering tool” for ease of use!

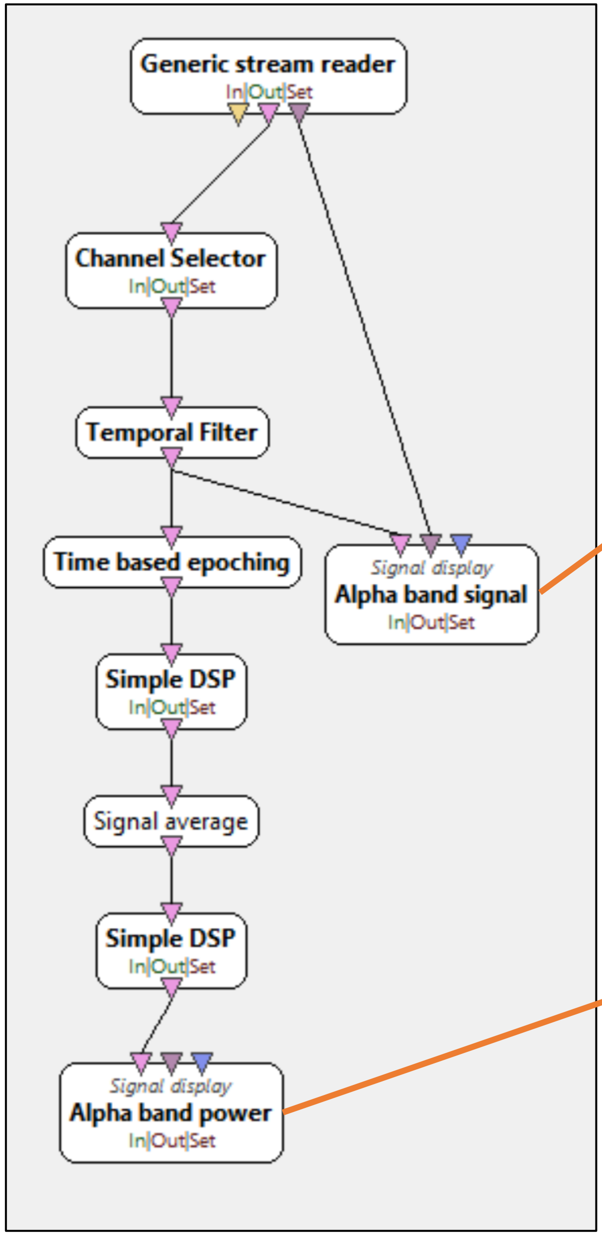
2. **Set time scales :** 10 (seconds) for alpha band signal display, 50 for alpha band power (since we have 5 times more data to display)

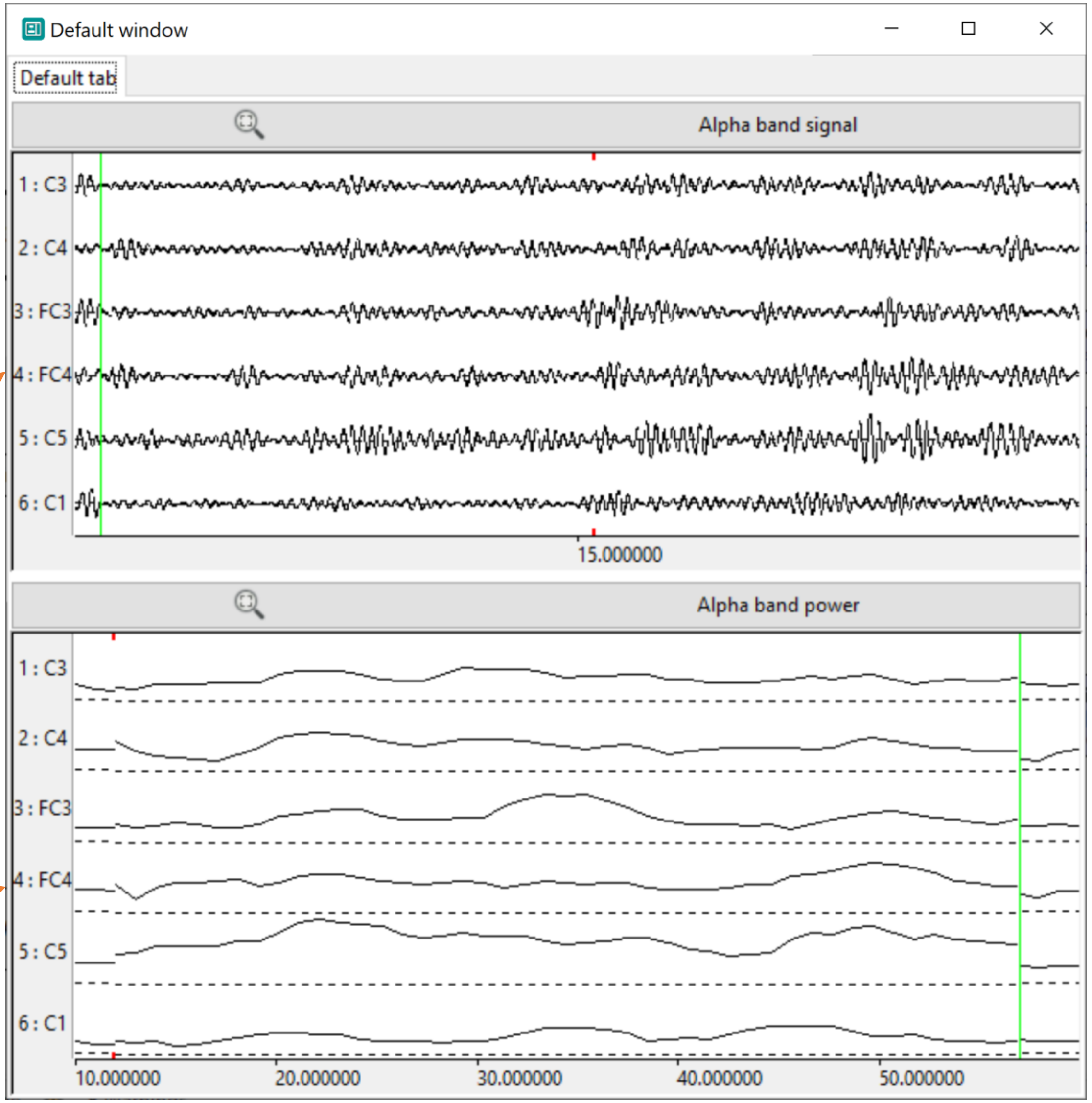
3. Set vertical scaling to “global”


Configure Alpha band signal settings

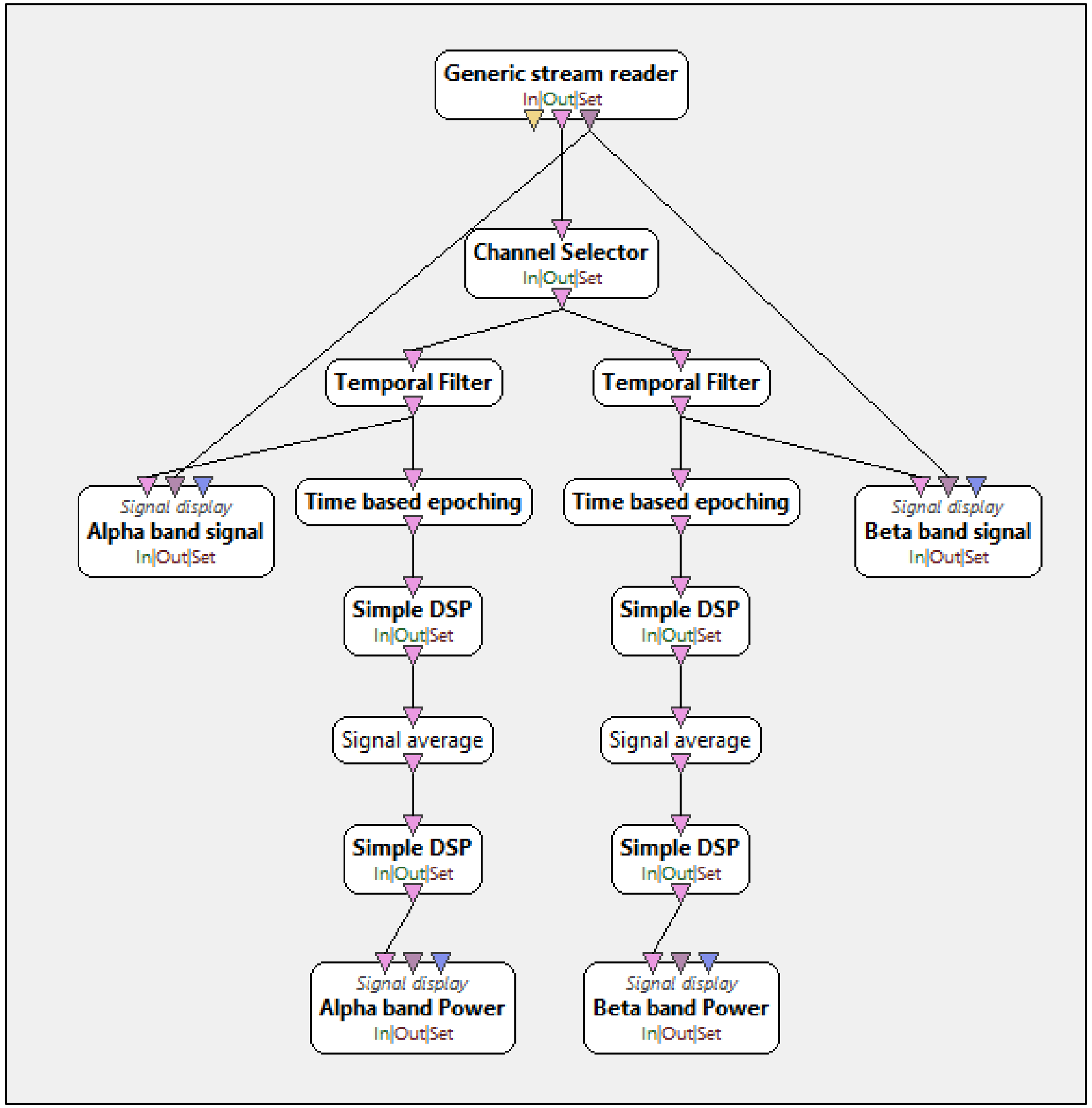
Display Mode	Scan
Auto vertical scale	Global
Scale refresh interval (secs)	5
Vertical Scale	100
Vertical Offset	0
Time Scale	10
Bottom ruler	true ☑
Left ruler	false □
Multiview	false □

Step 2 – DSP, Visualization



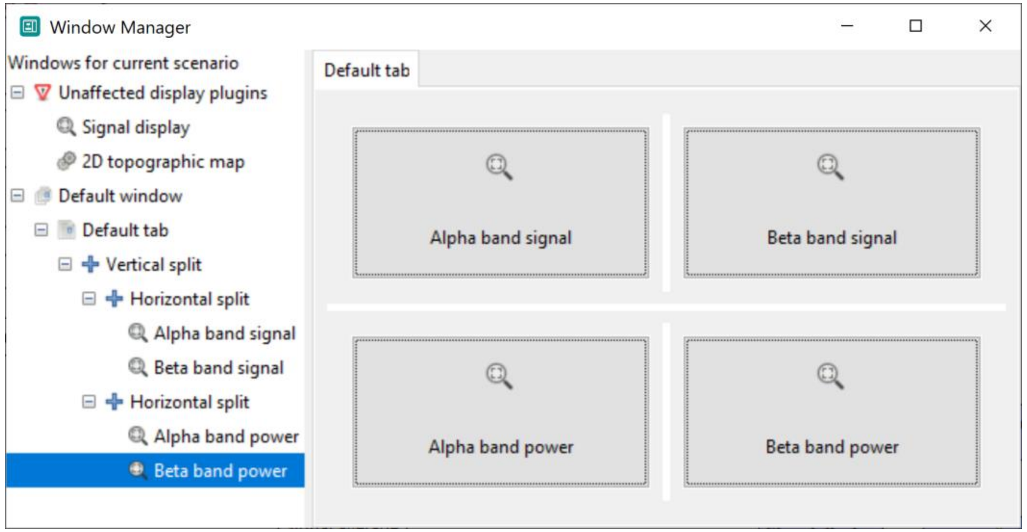


Step 2 - DSP, Visualization

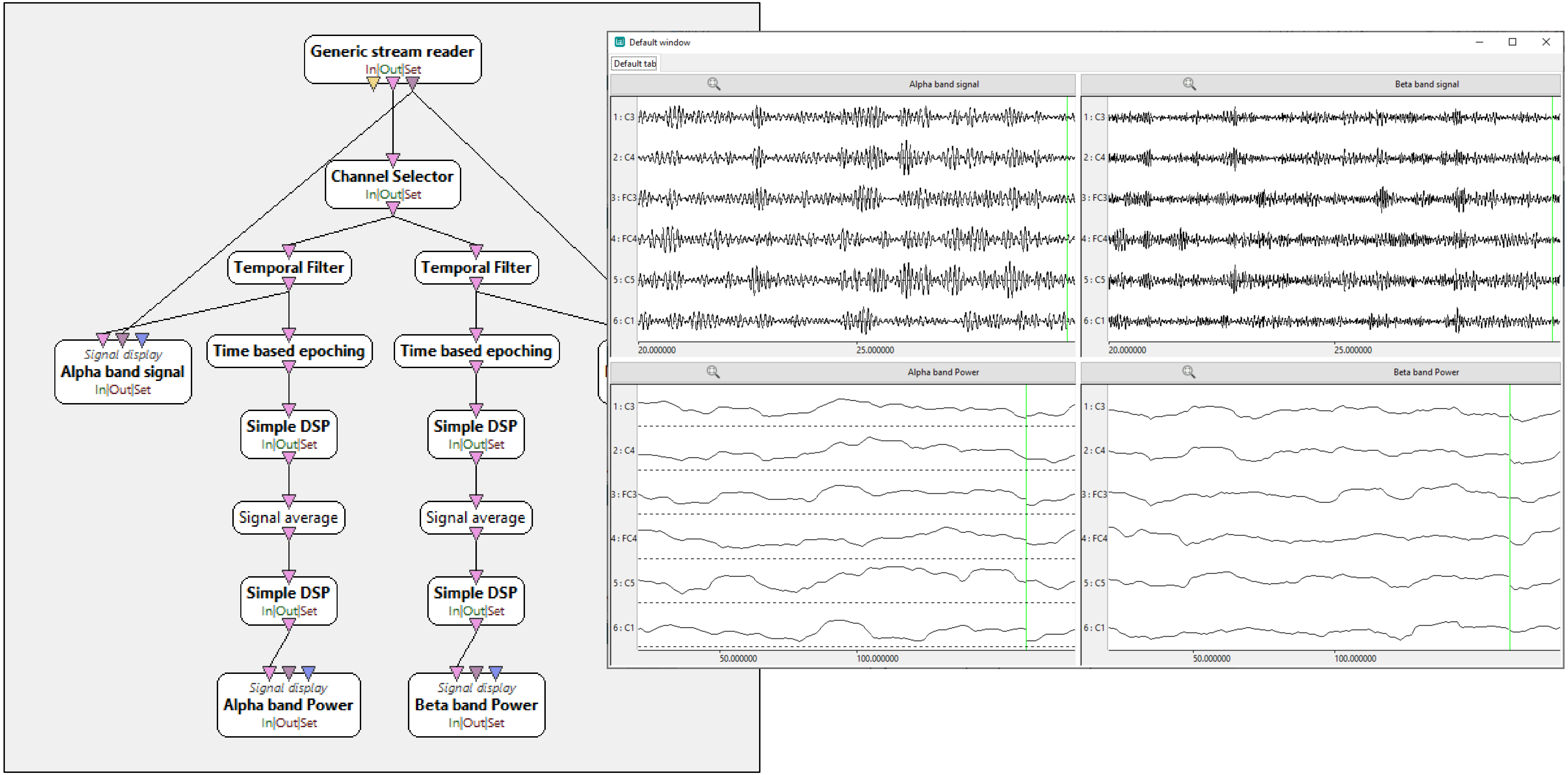


- Repeat for Beta Band [12;29hz]
- Don't hesitate to copy/paste boxes…

  … but make sure you check/update their parameters

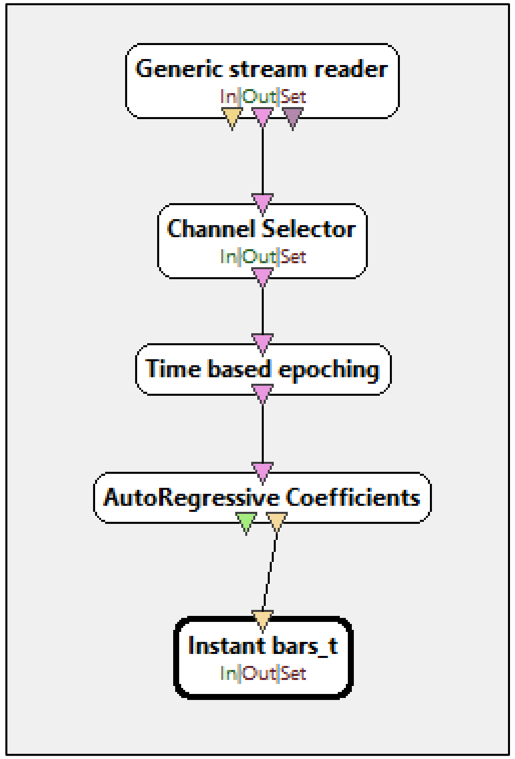


Step 2 - DSP, Visualization



Step 2 - DSP, Visualization

Alternative boxes for spectral estimation & visualization…

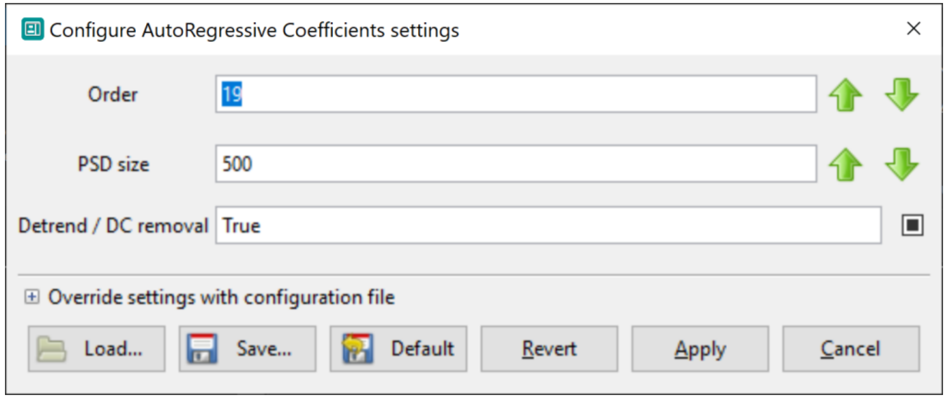


- **“Autoregressive Coefficients”**

  Estimates an AR model for the signal

  Output #1 = the AR coefficients

  Output #2 = the PSD of the signal, estimated as the FFT of the AR model.


Configure AutoRegressive Coefficients settings

Order: 19

PSD size: 500

Detrend / DC removal: True

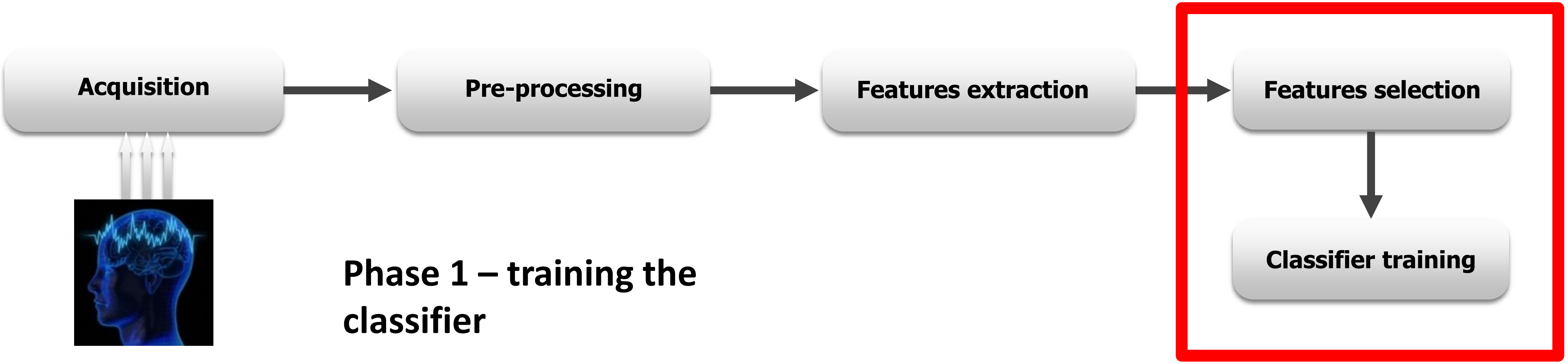
⊞ Override settings with configuration file
Load...
Save...
Default
Revert
Apply
Cancel

- **“Instant Bars”**

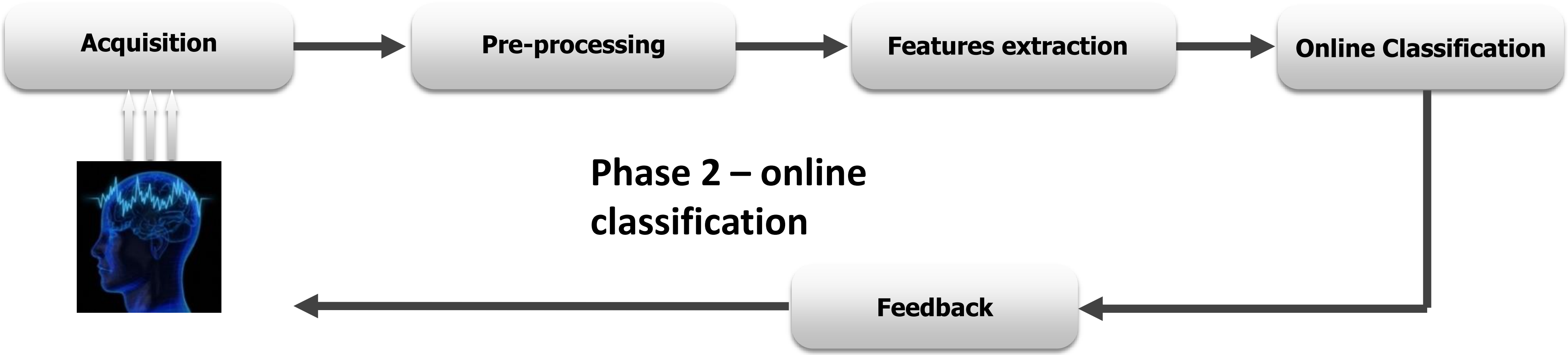
  Allows for quickly visualizing spectra / vectors / etc…

  ⚠ a bit bulky to setup

Motor Imagery Protocol



Phase 1 – training the classifier



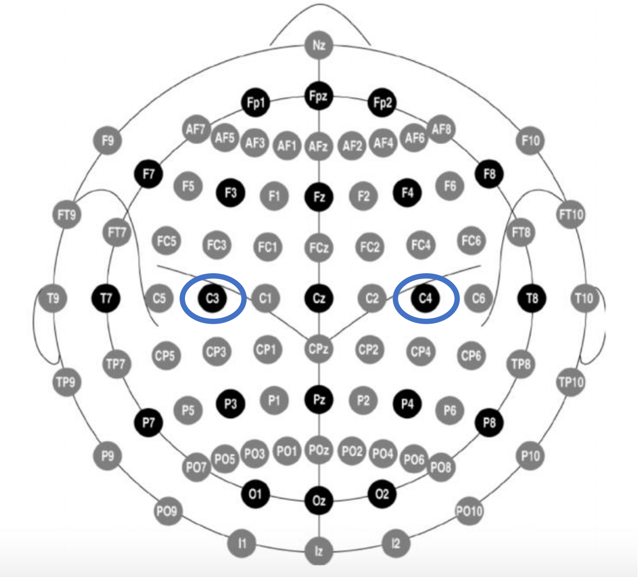
Phase 2 – online classification

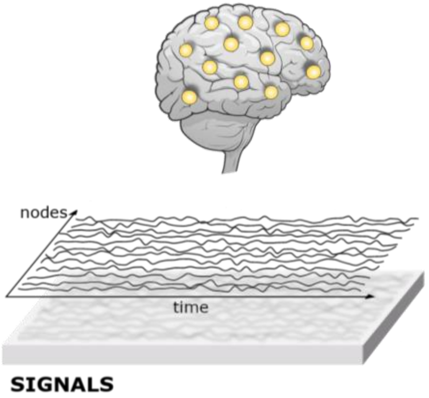
Step 3 - Training the classifier

- **Goal reminder:**

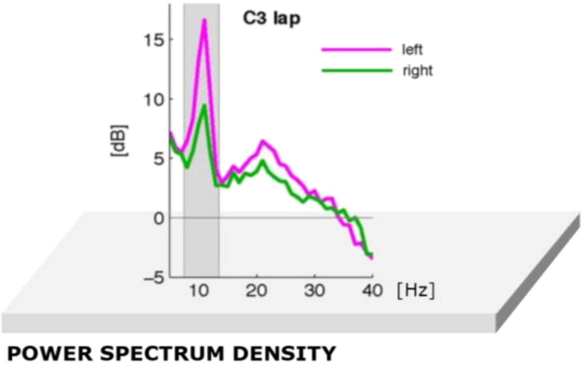
We want to train a classification algorithm…

- …to distinguish between mental states (MI /REST) …
- … using EEG signals from the sensorimotor areas …
- … more precisely instantaneous spectral power…
- … in bands alpha and/or beta





(Gonzalez-Astudillo et al, 2020)



(Maeder et al., 2012)

Step 3 - Training the classifier

- **Goal reminder:**

  We want to train a classification algorithm…

  - …to distinguish between mental states (MI /REST) …
  - … using EEG signals from the sensorimotor areas …
  - … more precisely instantaneous spectral power…
  - … in bands alpha and/or beta

- **In OpenViBE:**

Step 3 - Training the classifier

- ## Goal reminder:

We want to train a classification algorithm…

- …to distinguish between mental states (MI /REST) …
- … using EEG signals from the sensorimotor areas …
- … more precisely instantaneous spectral power…
- … in bands alpha and/or beta

- ## In OpenViBE:

- Feature Aggregator
- Classifier Trainer

Step 3 - Training the classifier

- **Goal reminder:**

  We want to train a classification algorithm…

  - …to distinguish between mental states (MI /REST) …
  - … using EEG signals from the sensorimotor areas …
  - … more precisely instantaneous spectral power…
  - … in bands alpha and/or beta

- **In OpenViBE:**
  - Stimulation Based Epoching
  - Feature Aggregator
  - Classifier Trainer

Step 3 - Training the classifier

- **Goal reminder:**

  We want to train a classification algorithm…

  - …to distinguish between mental states (MI /REST) …
  - … using EEG signals from the sensorimotor areas …
  - … more precisely instantaneous spectral power…
  - … in bands alpha and/or beta

- **In OpenViBE:**
  - Stimulation Based Epoching
  - Channel Selector
  - Feature Aggregator
  - Classifier Trainer

Step 3 - Training the classifier

- **Goal reminder:**

  We want to train a classification algorithm…

  - …to distinguish between mental states (MI /REST) …
  - … using EEG signals from the sensorimotor areas …
  - … more precisely instantaneous spectral power…
  - … in bands alpha and/or beta

- **In OpenViBE:**
  - Stimulation Based Epoching
  - Channel Selector
  - Time Based Epoching
  - AutoRegressive Coefficients
  - Feature Aggregator
  - Classifier Trainer

Step 3 - Training the classifier

- **Goal reminder:**

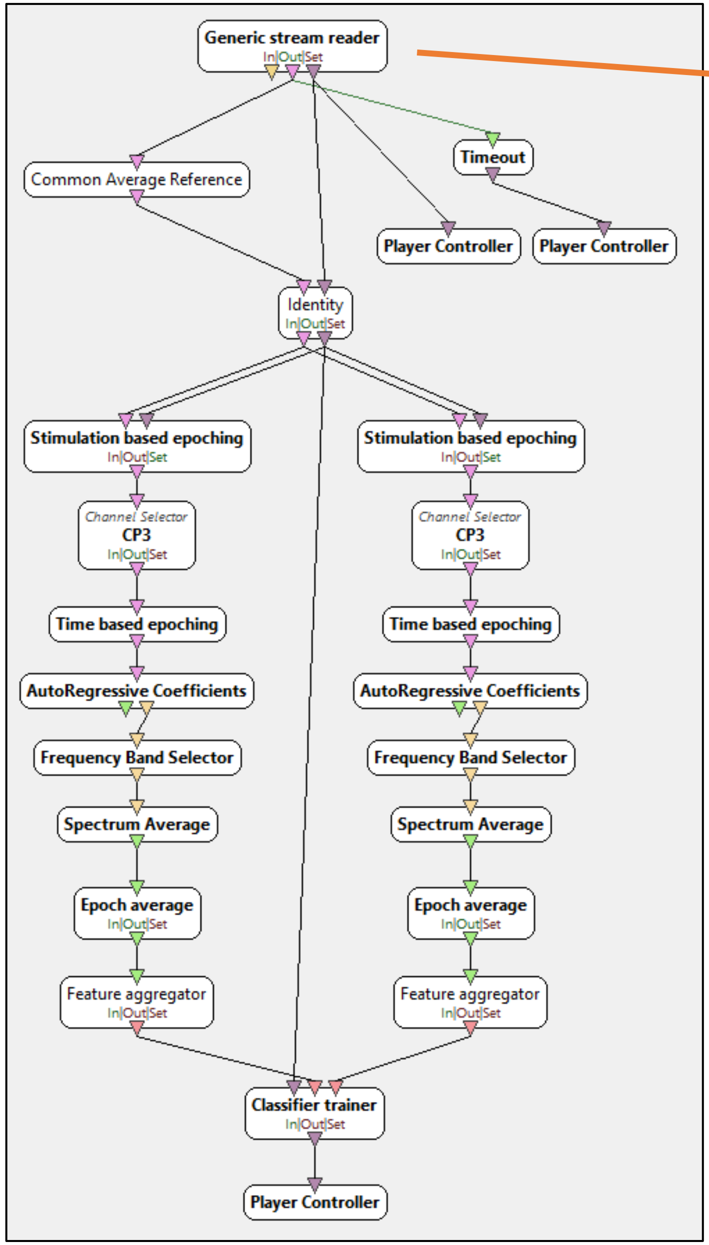
  We want to train a classification algorithm…

  - …to distinguish between mental states (MI /REST) …
  - … using EEG signals from the sensorimotor areas …
  - … more precisely instantaneous spectral power…
  - … in bands alpha and/or beta →

- **In OpenViBE:**
  - Stimulation Based Epoching
  - Channel Selector
  - Time Based Epoching
  - AutoRegressive Coefficients
  - Frequency Band Selector
  - Feature Aggregator
  - Classifier Trainer

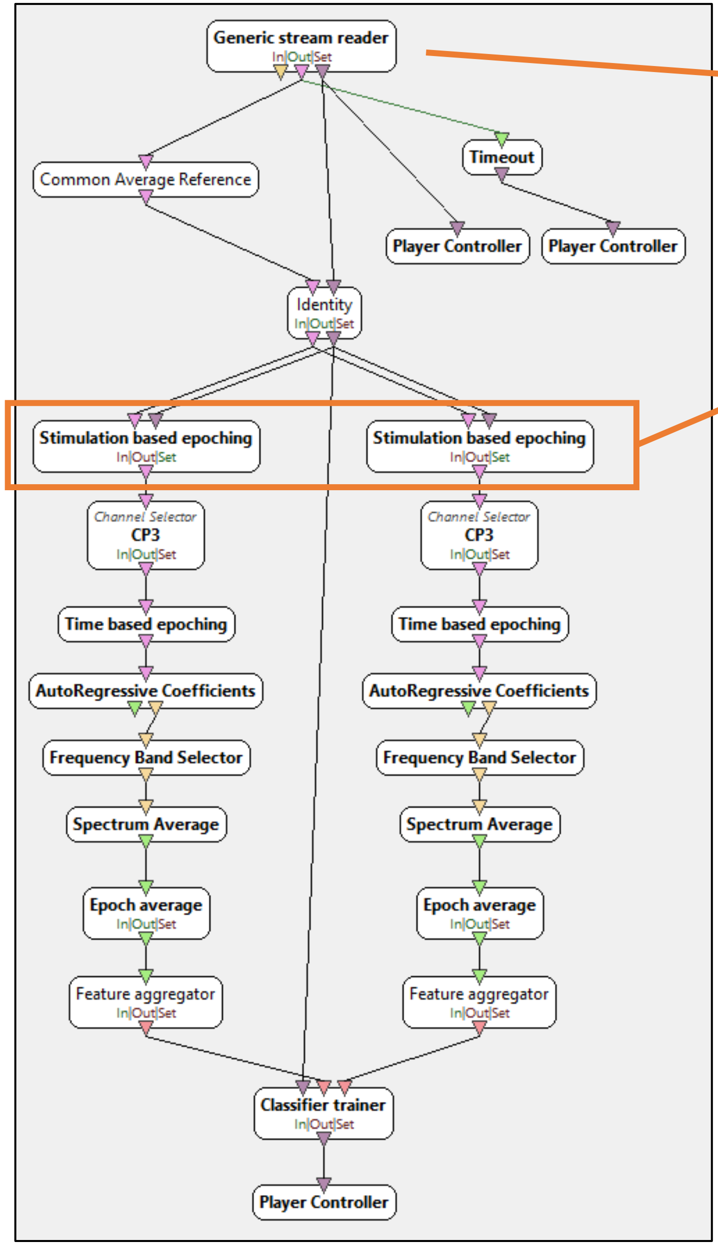
  … + a few other boxes

Step 3 - Training the classifier



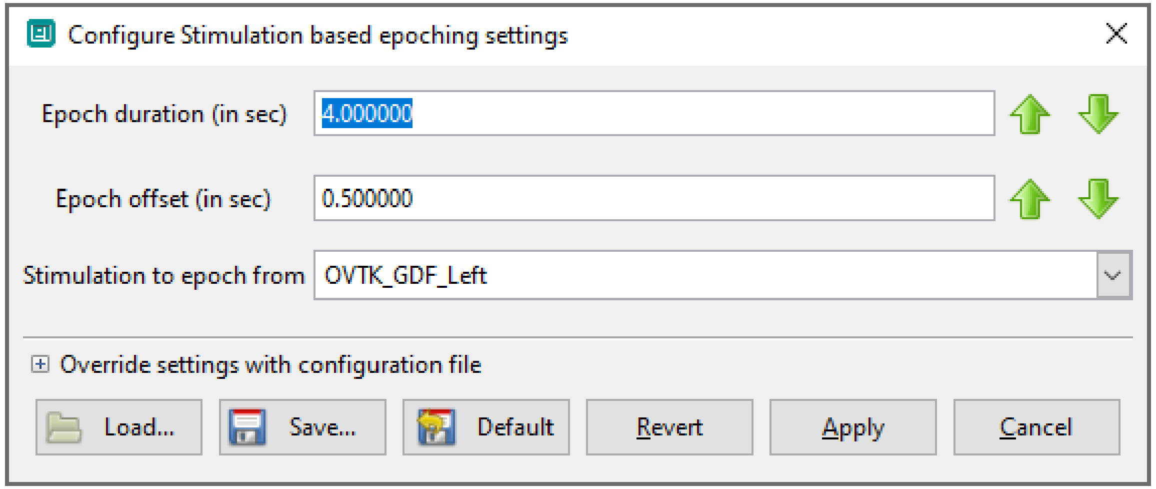
Load a signal file…

Step 3 - Training the classifier



Load a signal file…

- … **distinguish** between mental states (MI /REST)…


Configure Stimulation based epoching settings

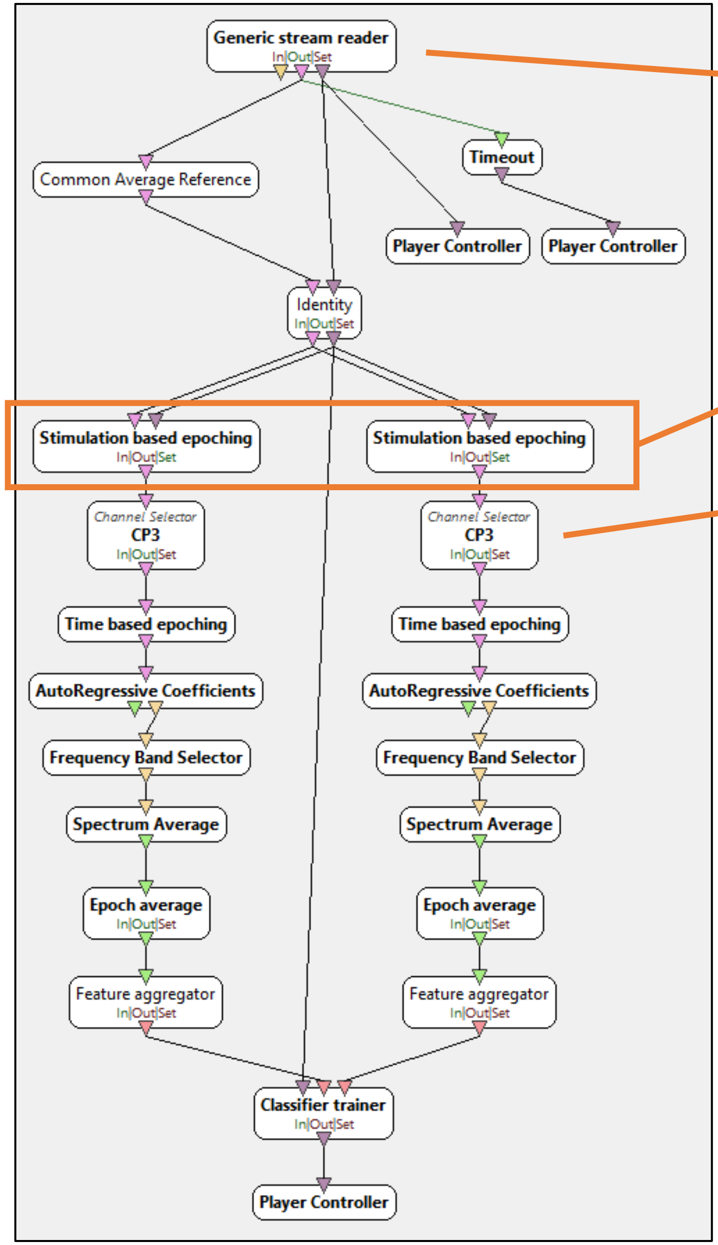
Epoch duration (in sec) 4.000000

Epoch offset (in sec) 0.500000

Stimulation to epoch from OVTK_GDF_Left

⊞ Override settings with configuration file

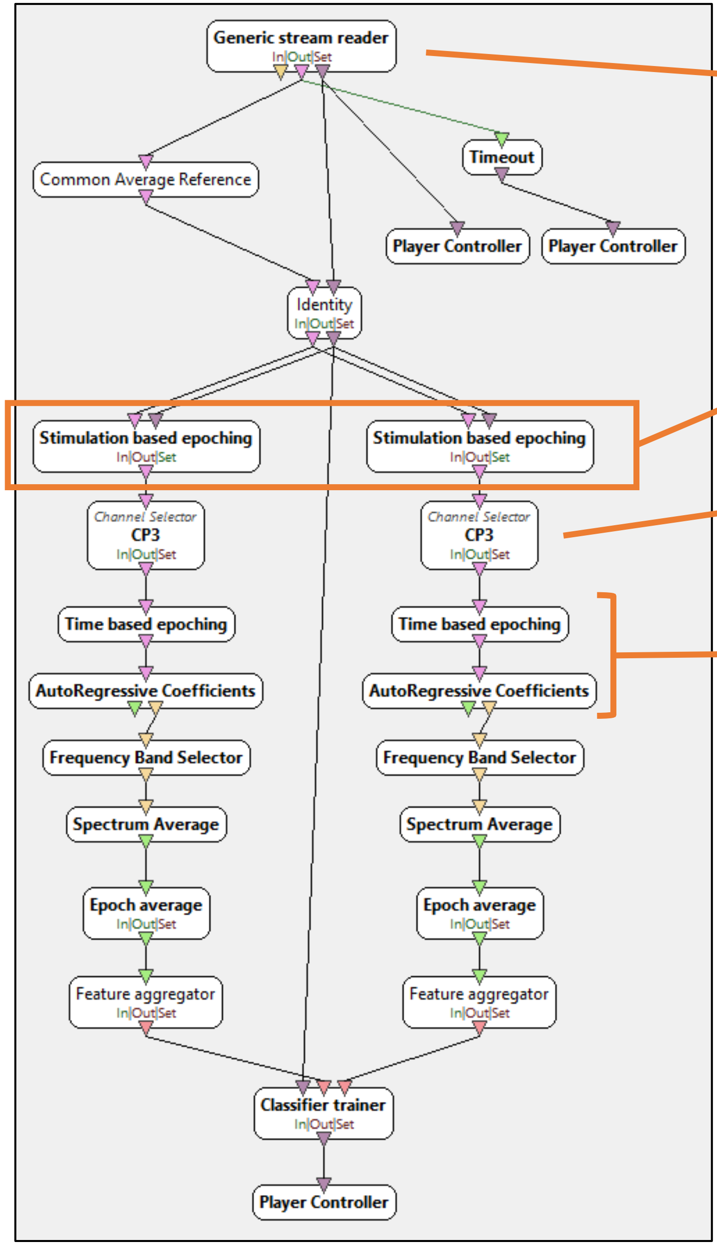
Step 3 - Training the classifier



Load a signal file…

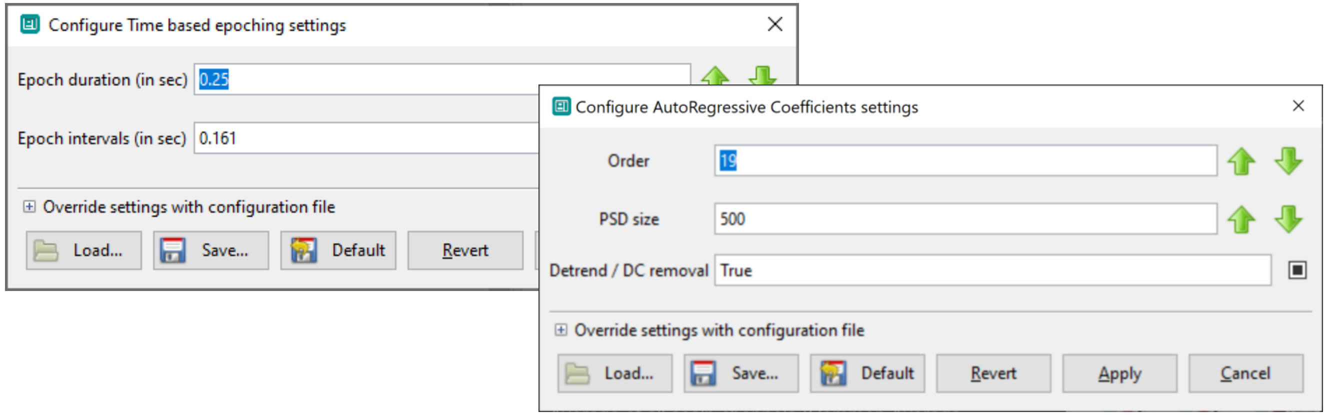
- … **distinguish** between mental states (MI /REST)…
- … using EEG signals **from the sensorimotor areas…**

Step 3 - Training the classifier



Load a signal file…

- … **distinguish** between mental states (MI /REST)…
- … using EEG signals **from the sensorimotor areas…**
- … more precisely **instantaneous spectral power…**



Configure Time based epoching settings

Epoch duration (in sec) 0.25

Epoch intervals (in sec) 0.161

⊞ Override settings with configuration file

Configure AutoRegressive Coefficients settings

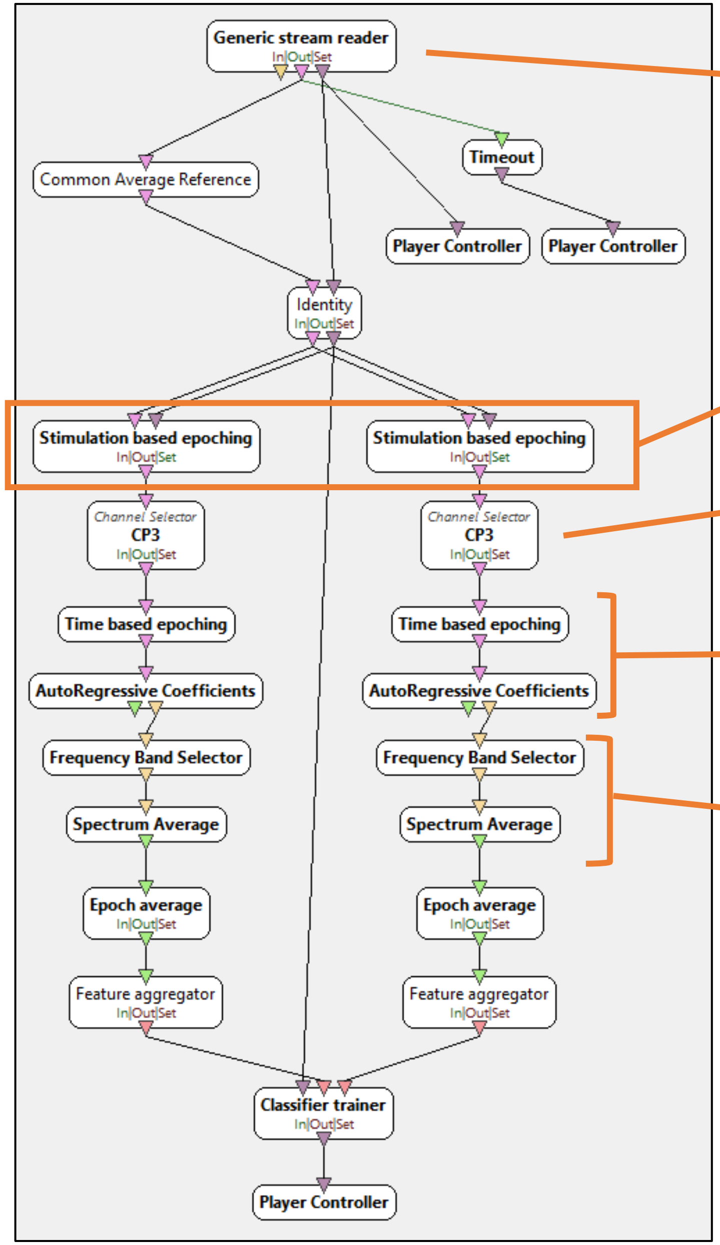
Order 19

PSD size 500

Detrend / DC removal True

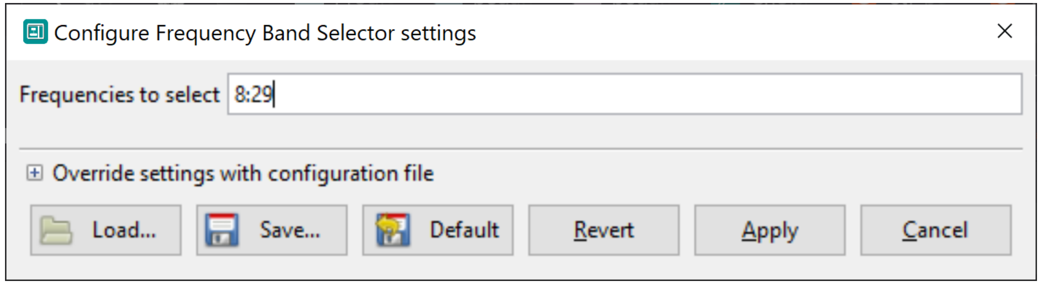
⊞ Override settings with configuration file

Step 3 - Training the classifier



Load a signal file…

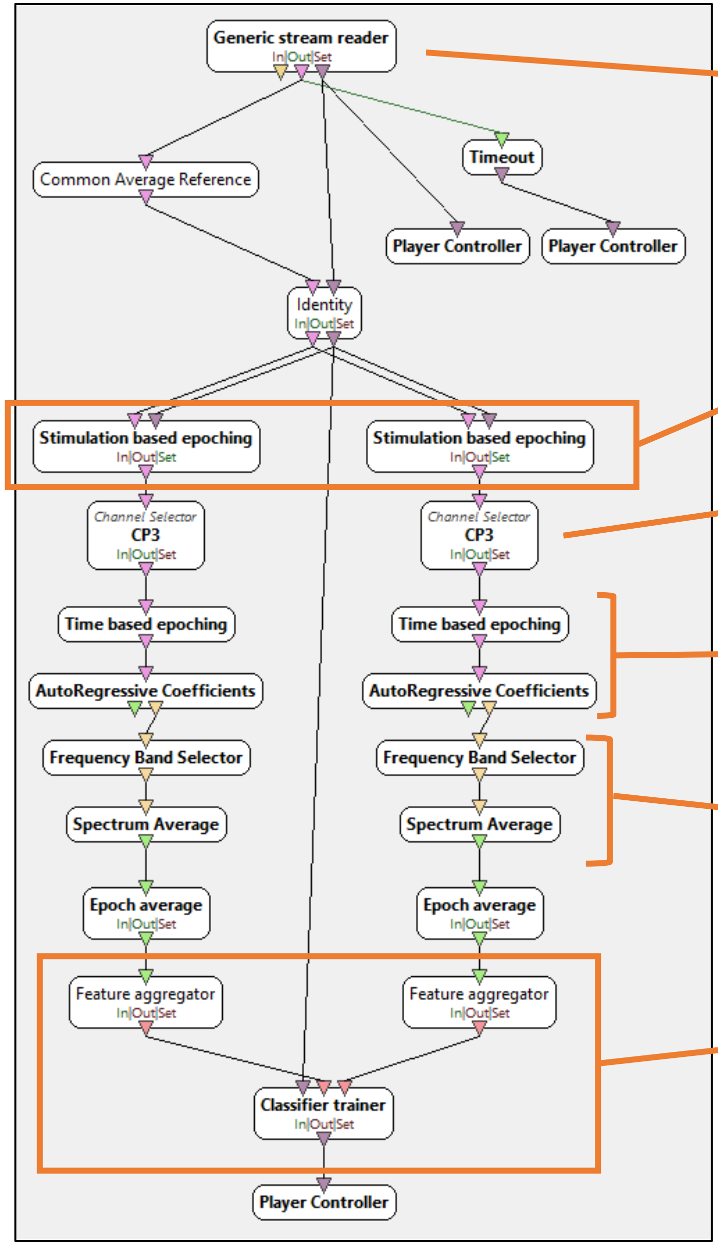
- … **distinguish** between mental states (MI /REST)…
- … using EEG signals **from the sensorimotor areas…**
- … more precisely **instantaneous spectral power…**
- … in **bands alpha and/or beta**


Configure Frequency Band Selector settings
×
⊞ Override settings with configuration file
Load...
Save...
Default
Revert
Apply
Cancel

Frequencies to select: 8:29

(+ average across the freq. band)

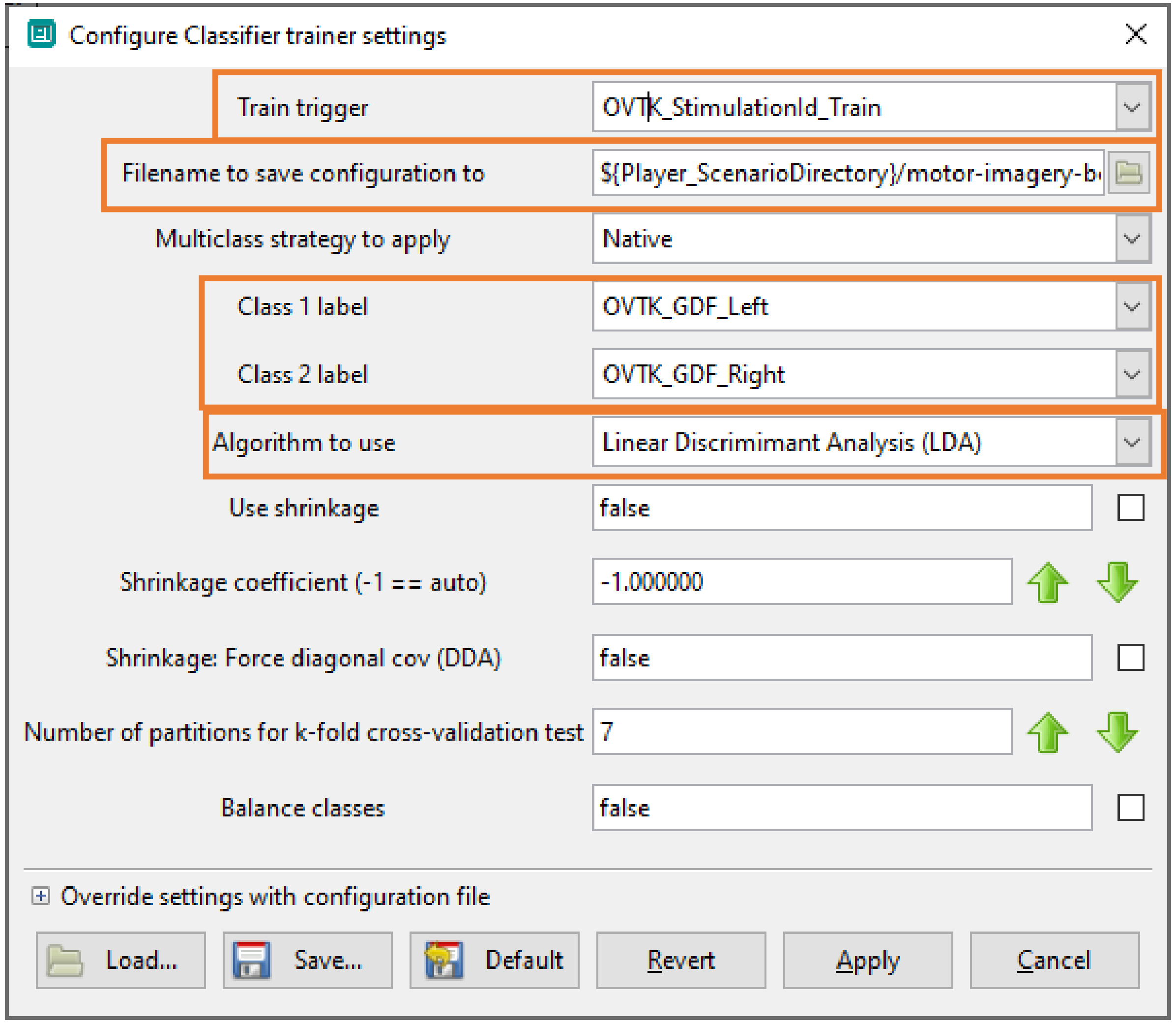
Step 3 - Training the classifier



Load a signal file…

- … **distinguish** between mental states (MI /REST)…
- … using EEG signals **from the sensorimotor areas…**
- … more precisely **instantaneous spectral power…**
- … in **bands alpha and/or beta**
- … to **train the classification algo**

Step 3 - Training the classifier



Configure Classifier trainer settings

Train trigger: OVTK_StimulationId_Train

Filename to save configuration to: ${Player_ScenarioDirectory}/motor-imagery-b

Multiclass strategy to apply: Native

Class 1 label: OVTK_GDF_Left

Class 2 label: OVTK_GDF_Right

Algorithm to use: Linear Discrimimant Analysis (LDA)

Use shrinkage: false □

Shrinkage coefficient (-1 == auto): -1.000000

Shrinkage: Force diagonal cov (DDA): false □

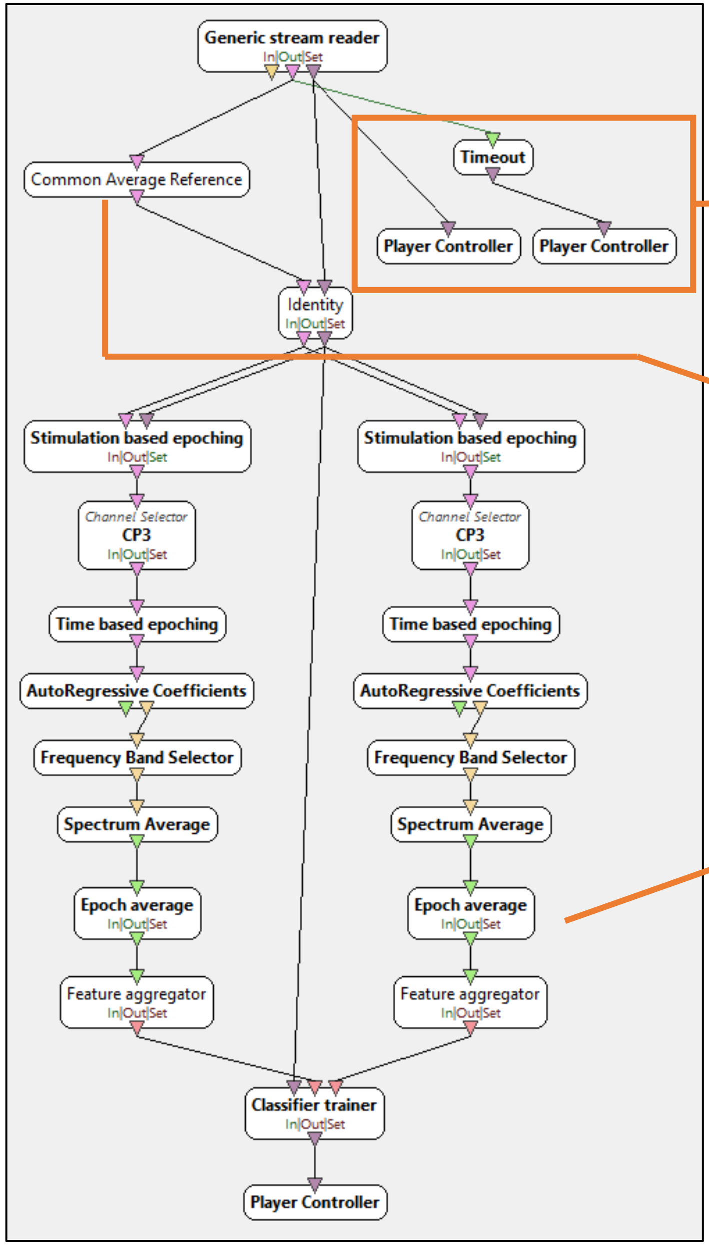
Number of partitions for k-fold cross-validation test: 7

Balance classes: false □

⊞ Override settings with configuration file

- **"Train Trigger"**: Set in the acquisition LUA script (see Annex) to a specific stimulation code, triggered at the end of the experiment
- Set the **path/filename** for the training "weights" (results)
- Check that the **classifier classes** are correctly labeled
- Select a **classification algo.** (LDA, SVM…) and set its parameters

Step 3 - Training the classifier



Other (useful!) boxes

- **Player Controller** / Timeout:

  Control the flow of the experiment

- **Common Avg Reference**

  Make sure the avg. of all channels' signals is zero

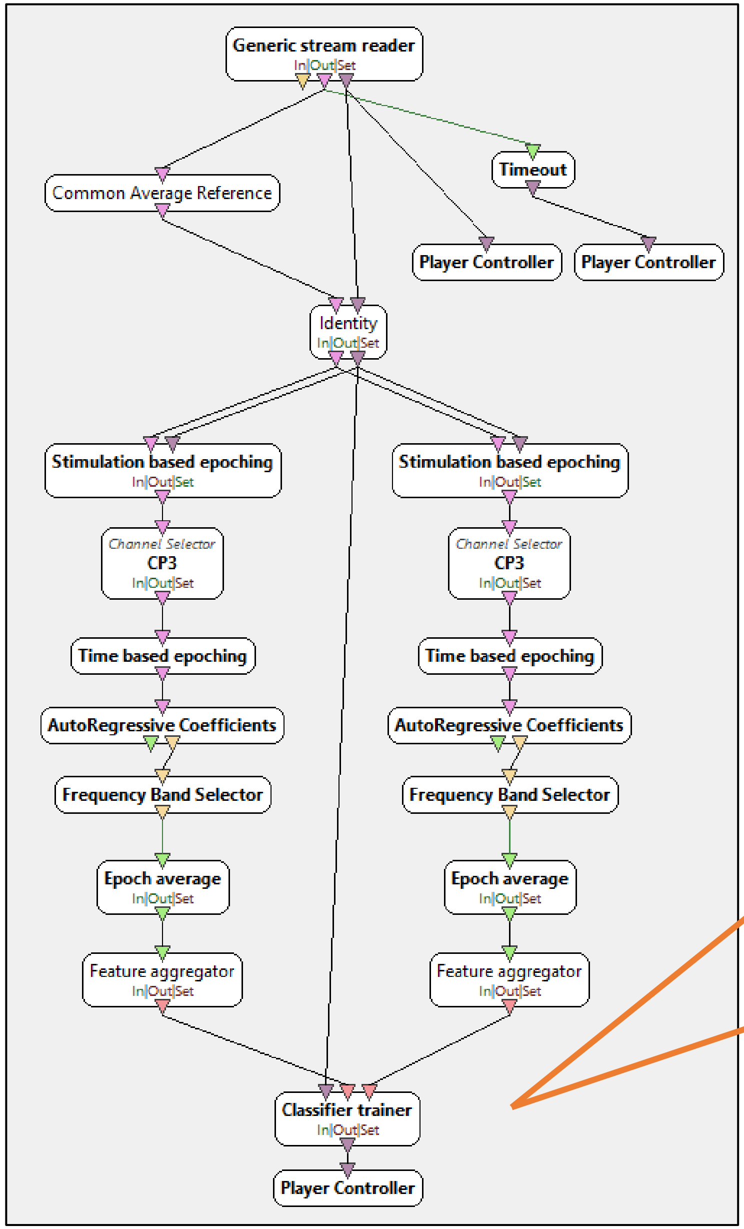
- **Epoch Average**

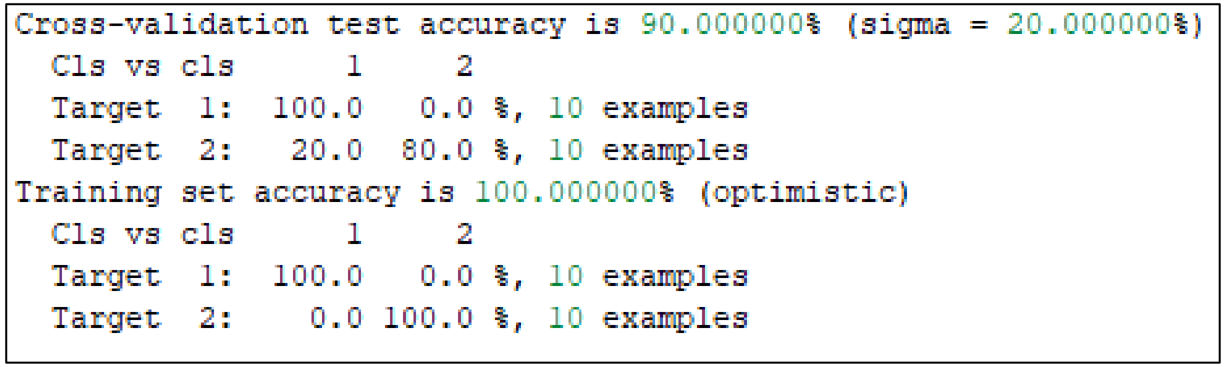
  Average "Feature Vectors" used for classification across a full "Epoch" (here, the duration of a MI trial)

- **Identity**

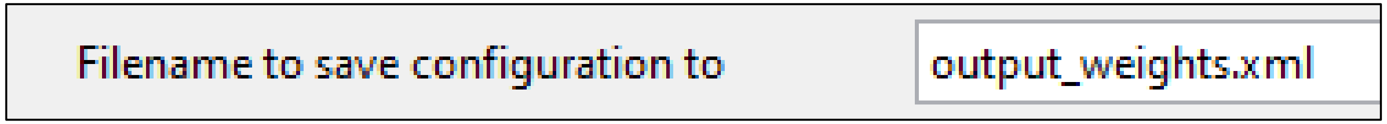
  Simply propagates signals & stims. Useful for readability

Step 3 - Training the classifier



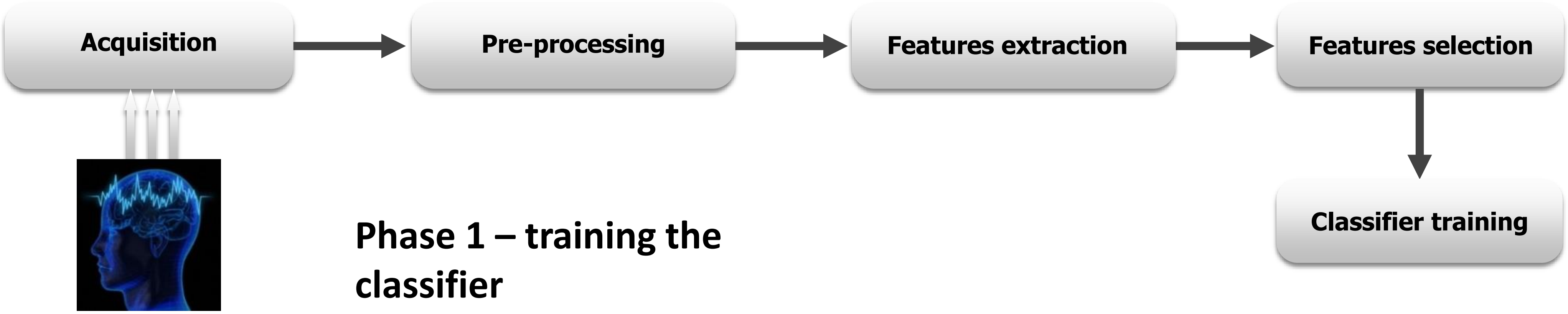


```
Cross-validation test accuracy is 90.000000% (sigma = 20.000000%)
  Cls vs cls       1      2
  Target  1:  100.0    0.0 %, 10 examples
  Target  2:   20.0   80.0 %, 10 examples
Training set accuracy is 100.000000% (optimistic)
  Cls vs cls       1      2
  Target  1:  100.0    0.0 %, 10 examples
  Target  2:    0.0 100.0 %, 10 examples
```



Filename to save configuration to: output_weights.xml

Motor Imagery Protocol - Phase 2



Acquisition

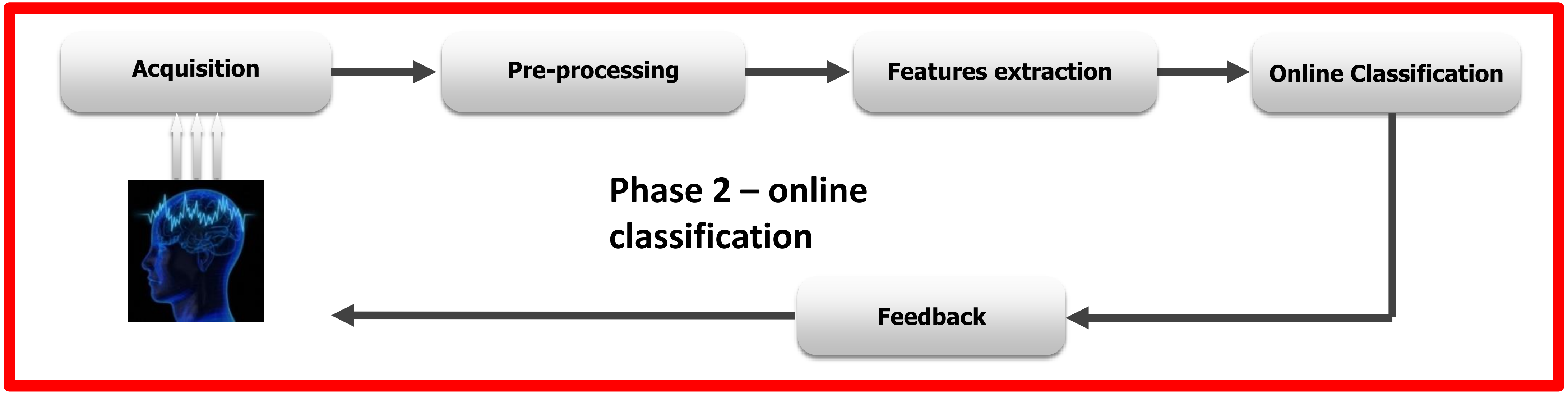
Pre-processing

Features extraction

Features selection

Classifier training

Phase 1 – training the classifier



Acquisition

Pre-processing

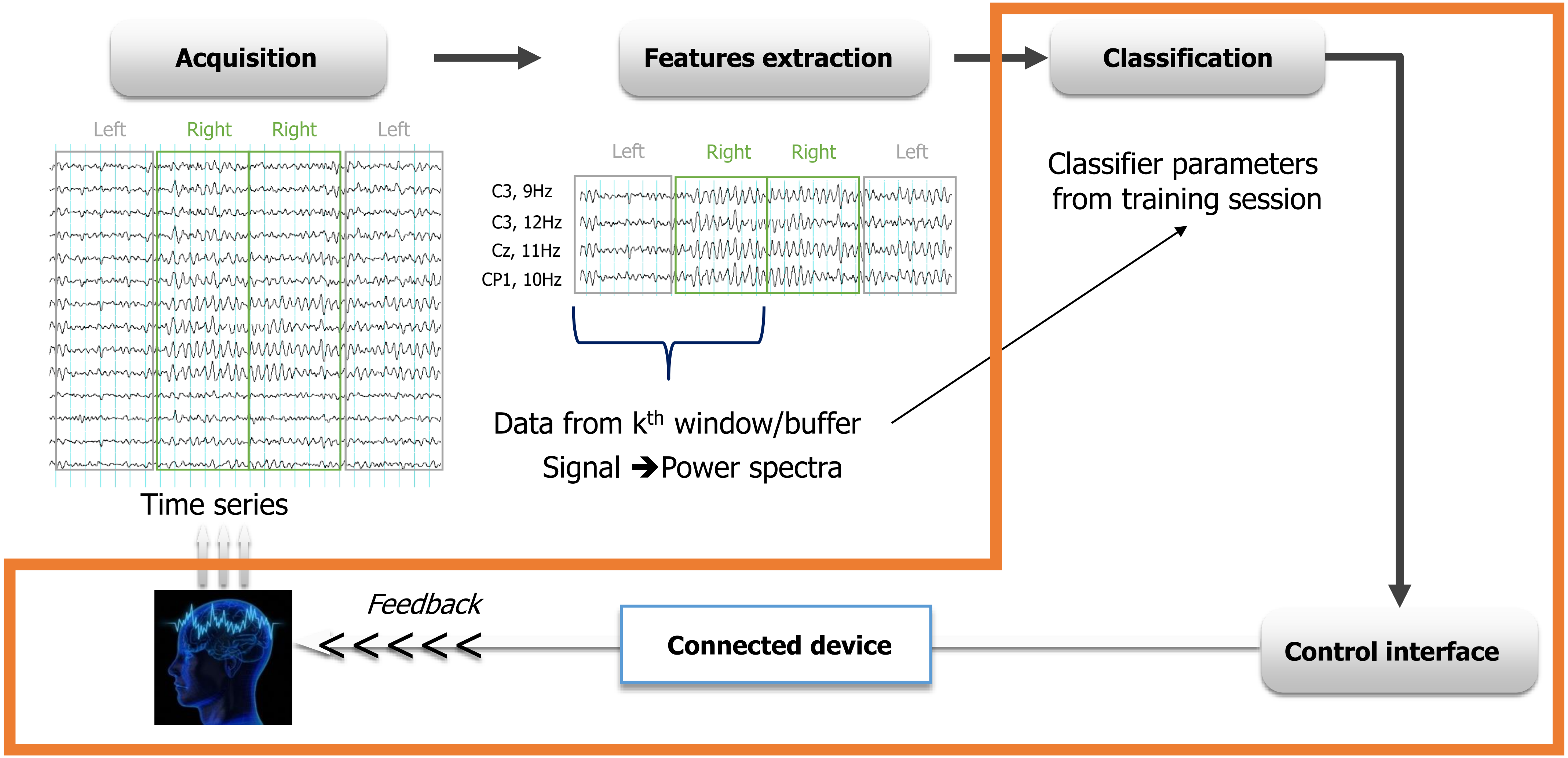
Features extraction

Online Classification

Phase 2 – online classification

Feedback

Motor Imagery Protocol - Phase 2



Step 3 - Online classification

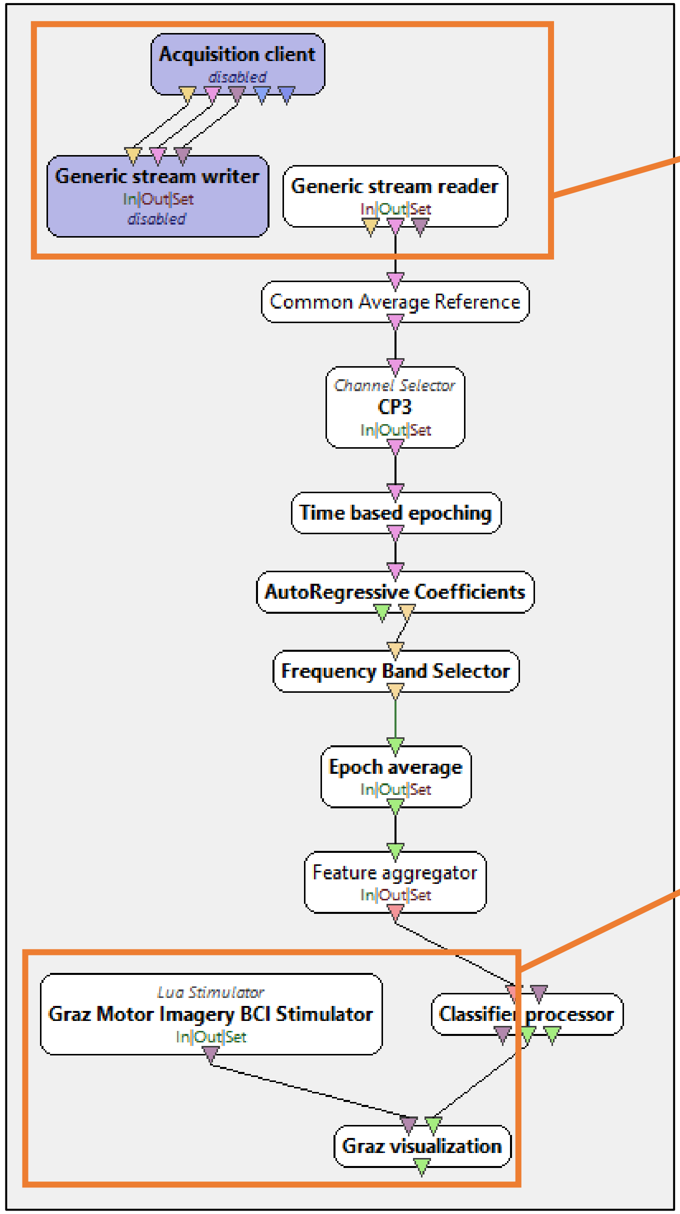
- **Goals:**

From EEG data acquired in **real-time**, while providing **MI tasks** to the subject…

- We want to **classify their mental states** (also in real-time!)
- **… using the classifier previously trained.**

➔ We will reuse most of the building blocks set up so far

Step 3 - Online classification



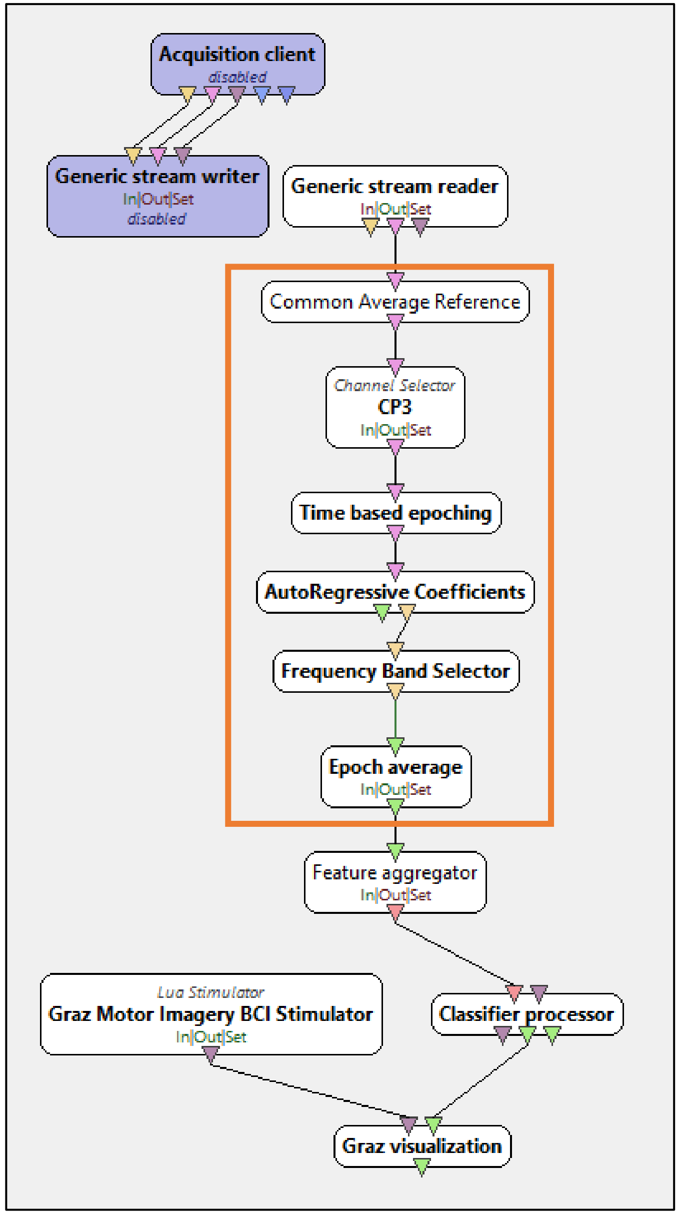
Using a pre-recorded signal file

In a real BCI experiment, we should of course use the **Acquisition Client** (and record the incoming signals for future replays and studies).

Today, we’re running an online demonstration without EEG hardware, so we shall use a pre-recorded signal file…

The subject would receive instructions synchronized with stimulations set up in the **LUA stimulator Box.** Let’s set this up exactly as for a real-life experiment.

Step 3 - Online classification

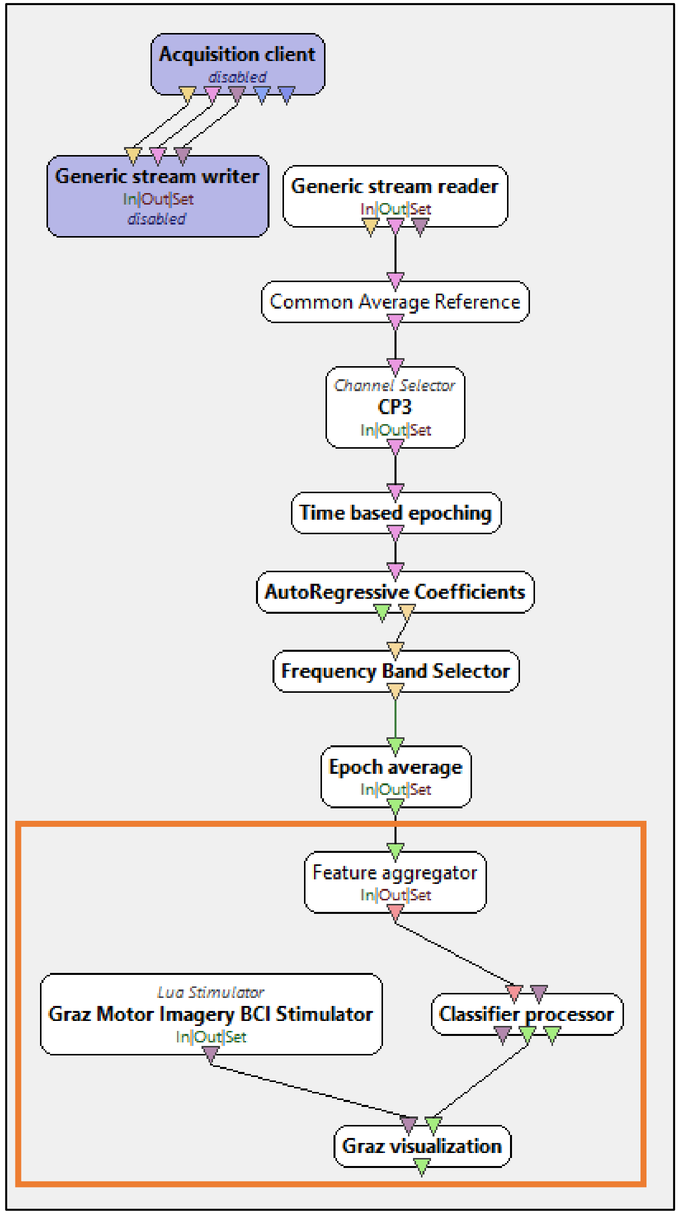


Signal Processing / Feature extraction

We want the classifier to receive the same type of data used for training.

Here, we shall use the same features previously used for training the classifier.

Step 3 - Online classification



Use the “**Classifier Processor**” box, loading the “**weights**” file generated in the training phase

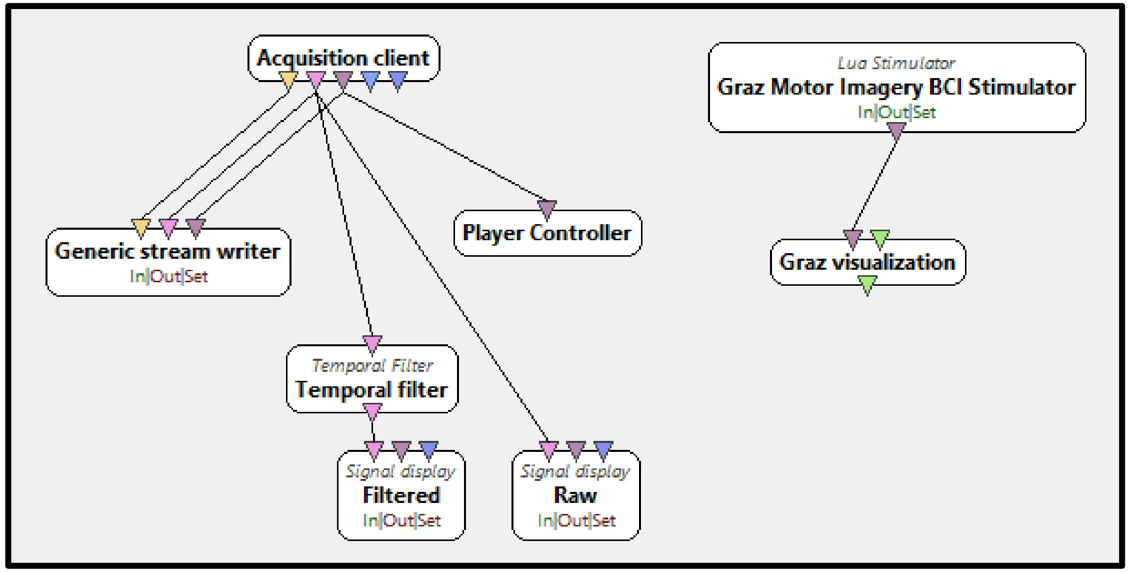
Use the same **LUA script** & **Graz Visualization** boxes as for the **Acquisition phase** to generate stimulations & “tasks”, and display instructions for the subject.

The “**streamed matrix**” stream from **Classifier processor** to **Graz Visualization** corresponds to the “classification **accuracy**”

ACQUISITION SERVER/CLIENT

PROTOCOL MANAGEMENT / STIMULATIONS

Acquisition - Protocol mgmt. & Stimulations



- **Scenario:**

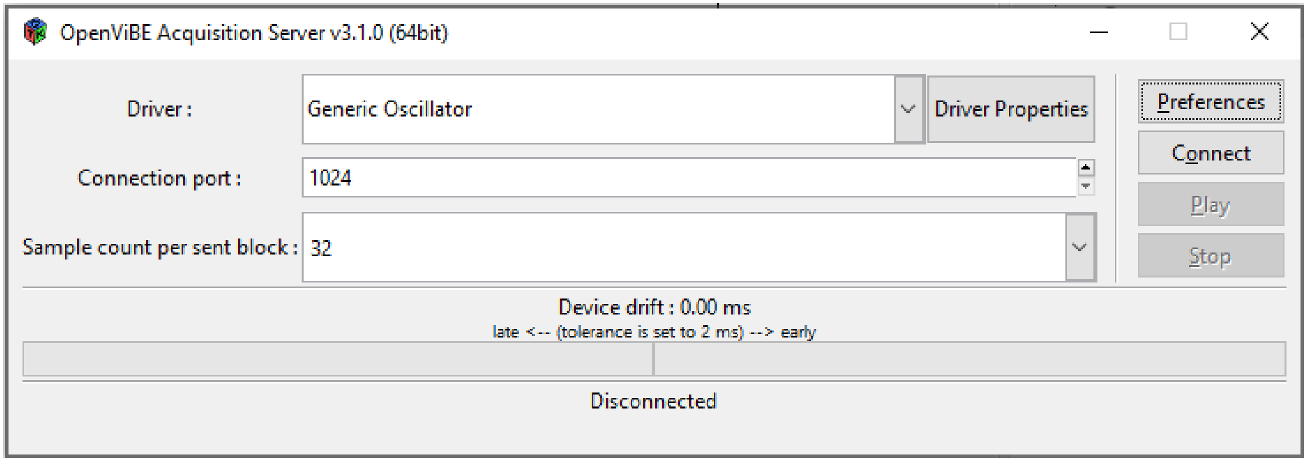
  in the workshop github:

  `sc1-monitor-acq.xml`

- **OpenViBE Acquisition server:**

  in the install folder:

  `openvibe-acquisition-server.cmd`



Driver : Generic Oscillator

Connection port : 1024

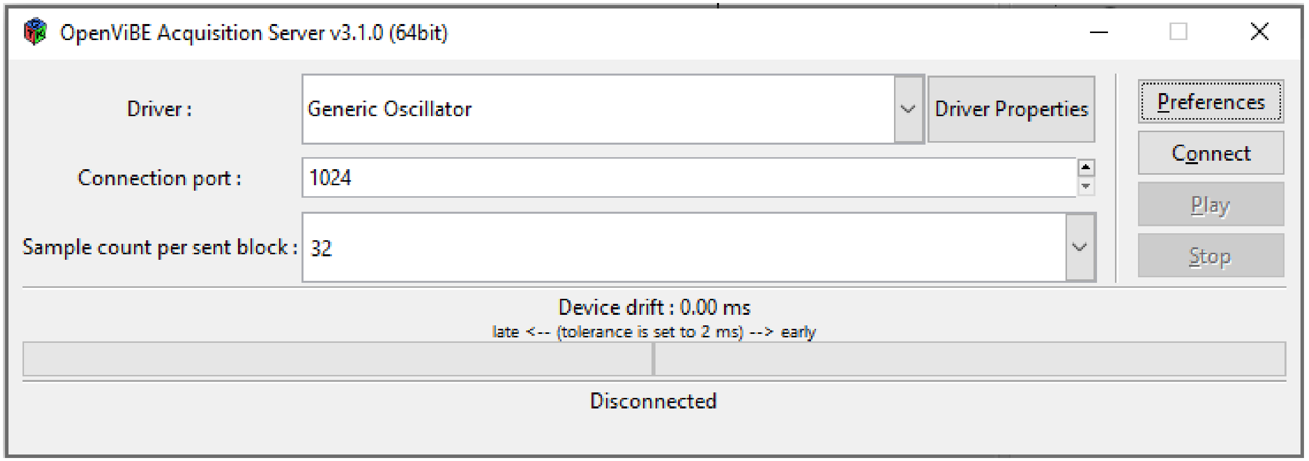
Sample count per sent block : 32

Device drift : 0.00 ms

late <-- (tolerance is set to 2 ms) --> early

Disconnected

Acquisition - Protocol mgmt. & Stimulations



OpenViBE Acquisition Server v3.1.0 (64bit)

Driver : Generic Oscillator

Connection port : 1024

Sample count per sent block : 32

Device drift : 0.00 ms

late <-- (tolerance is set to 2 ms) --> early

Disconnected

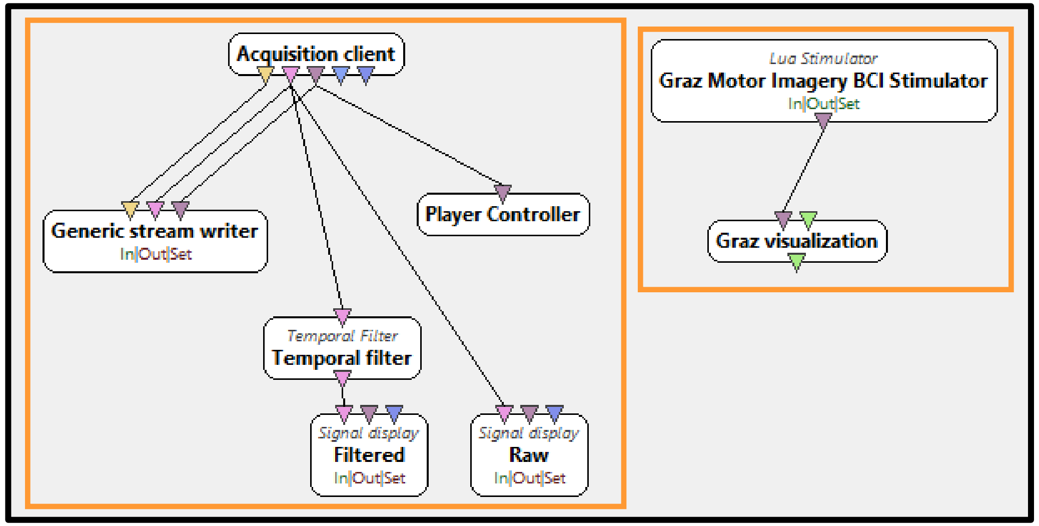
1. **In the “Driver” list, select “Generic Oscillator”**

   In this list, you’ll find all the drivers for OpenViBE’s supported EEG hardware

2. **Click “Connect”**

3. **Click “Play”**

Acquisition - Protocol mgmt. & Stimulations

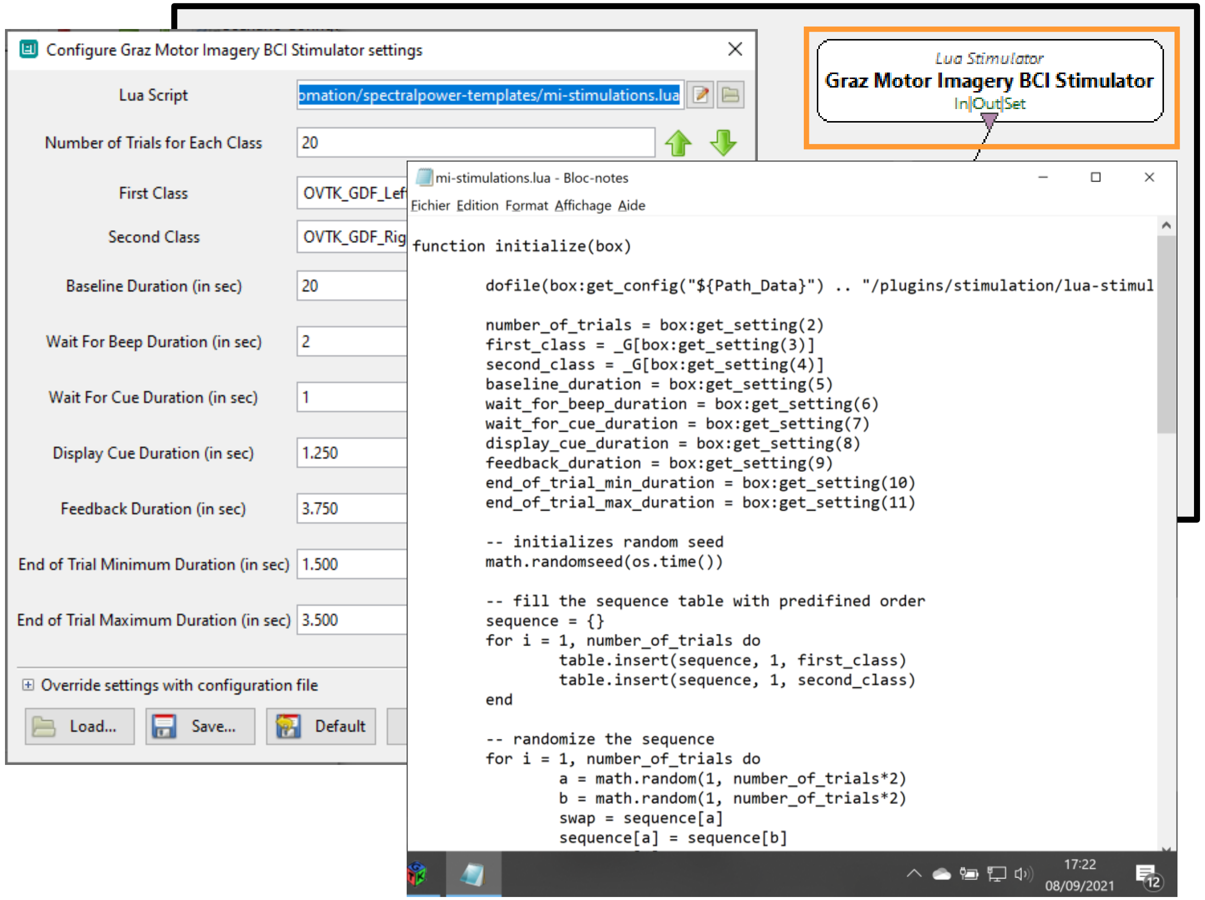


Stimulation generation

& display

EEG data acquisition

Acquisition - Protocol mgmt. & Stimulations


Lua Stimulator
Graz Motor Imagery BCI Stimulator
In|Out|Set
Load...
Save...
Default
Fichier Edition Format Affichage Aide
17:22
08/09/2021

Configure Graz Motor Imagery BCI Stimulator settings

Lua Script	pmation/spectralpower-templates/mi-stimulations.lua
Number of Trials for Each Class	20
First Class	OVTK_GDF_Lef
Second Class	OVTK_GDF_Rig
Baseline Duration (in sec)	20
Wait For Beep Duration (in sec)	2
Wait For Cue Duration (in sec)	1
Display Cue Duration (in sec)	1.250
Feedback Duration (in sec)	3.750
End of Trial Minimum Duration (in sec)	1.500
End of Trial Maximum Duration (in sec)	3.500

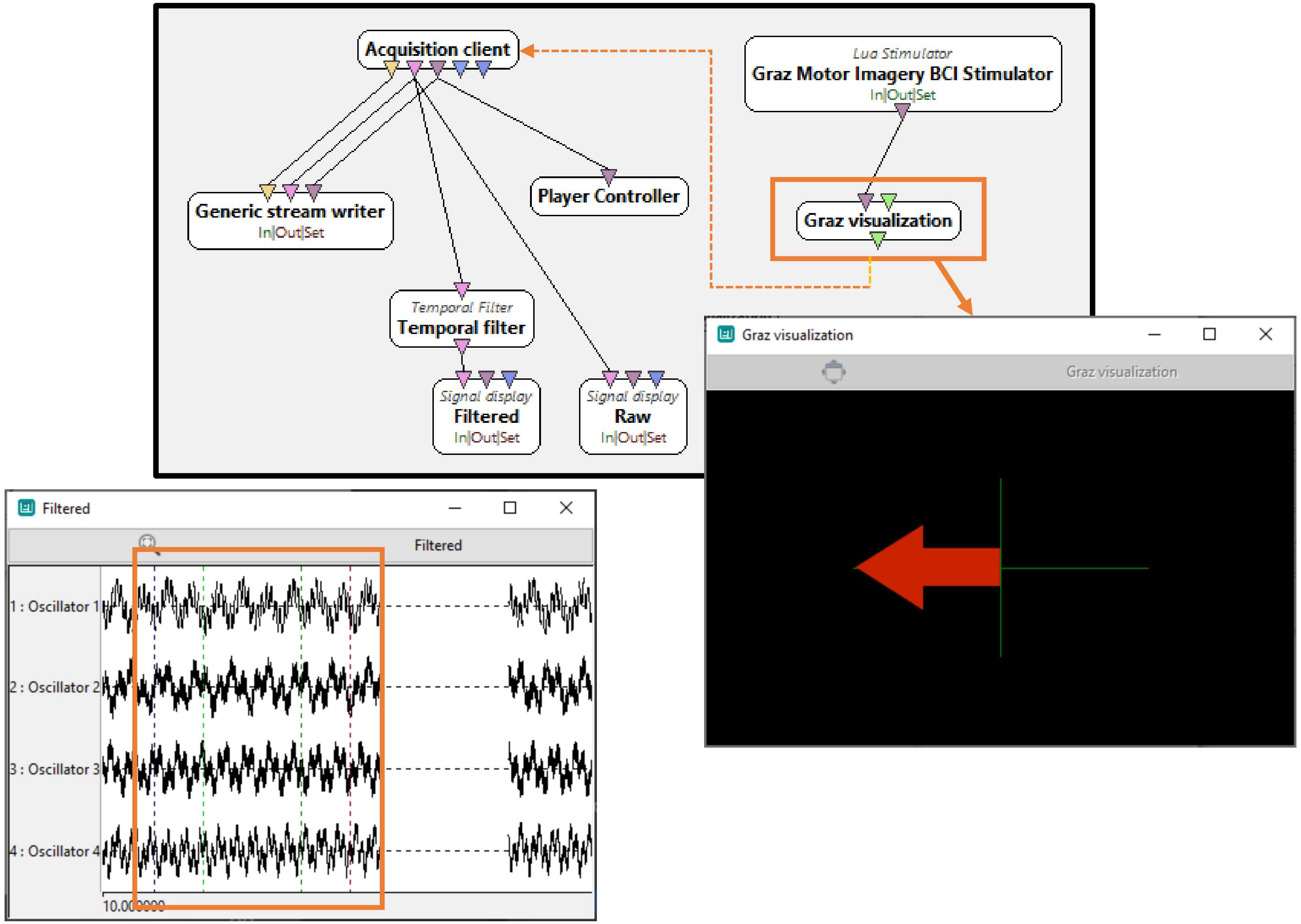
⊞ Override settings with configuration file

mi-stimulations.lua - Bloc-notes

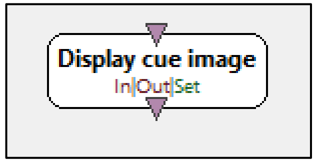
```
function initialize(box)

        dofile(box:get_config("${Path_Data}") .. "/plugins/stimulation/lua-stimul

        number_of_trials = box:get_setting(2)
        first_class = _G[box:get_setting(3)]
        second_class = _G[box:get_setting(4)]
        baseline_duration = box:get_setting(5)
        wait_for_beep_duration = box:get_setting(6)
        wait_for_cue_duration = box:get_setting(7)
        display_cue_duration = box:get_setting(8)
        feedback_duration = box:get_setting(9)
        end_of_trial_min_duration = box:get_setting(10)
        end_of_trial_max_duration = box:get_setting(11)

        -- initializes random seed
        math.randomseed(os.time())

        -- fill the sequence table with predifined order
        sequence = {}
        for i = 1, number_of_trials do
                table.insert(sequence, 1, first_class)
                table.insert(sequence, 1, second_class)
        end

        -- randomize the sequence
        for i = 1, number_of_trials do
                a = math.random(1, number_of_trials*2)
                b = math.random(1, number_of_trials*2)
                swap = sequence[a]
                sequence[a] = sequence[b]
```

- **LUA Stimulator Box**
  (LUA = scripting language)
- **Experiment example:**
  Different stimulation/event codes at different times.
  Useful for signal segmentation ("epoching")
- **Stimulation label examples:**
  - Experiment Start/Stop
  - Trial Start/Stop
  - LEFT / RIGHT
  - Button pressed
  - etc

Acquisition - Protocol mgmt. & Stimulations

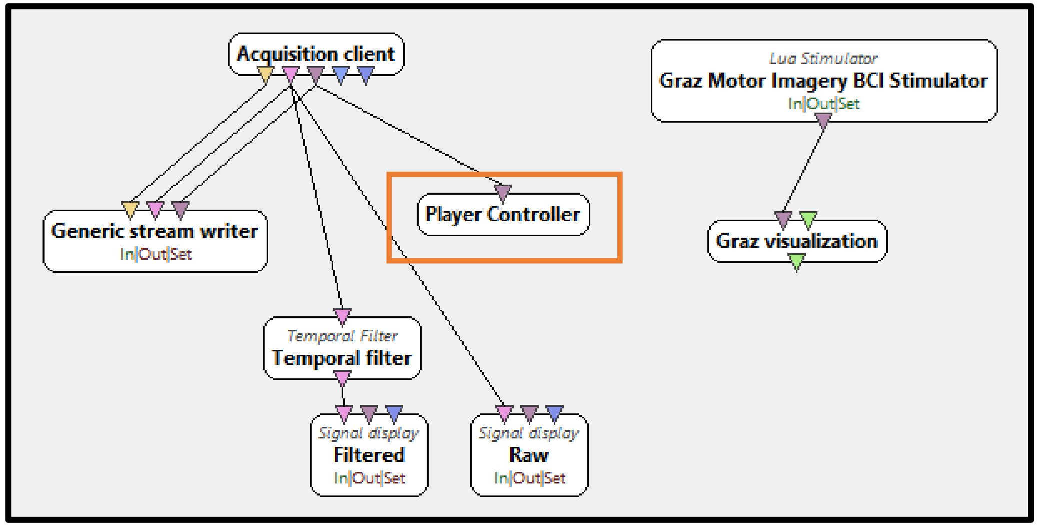


- **“Graz Visualization”** (specific for “Graz Protocol”, based on Box “Display Cue Image”)



- Displays specified image upon receiving specified stimulation code
- Transmits the **stimulation code & time** to the Acquisition Server
- **The stimulations are received by the ACQ Client, synchronized with the signal**

Acquisition - Protocol mgmt. & Stimulations

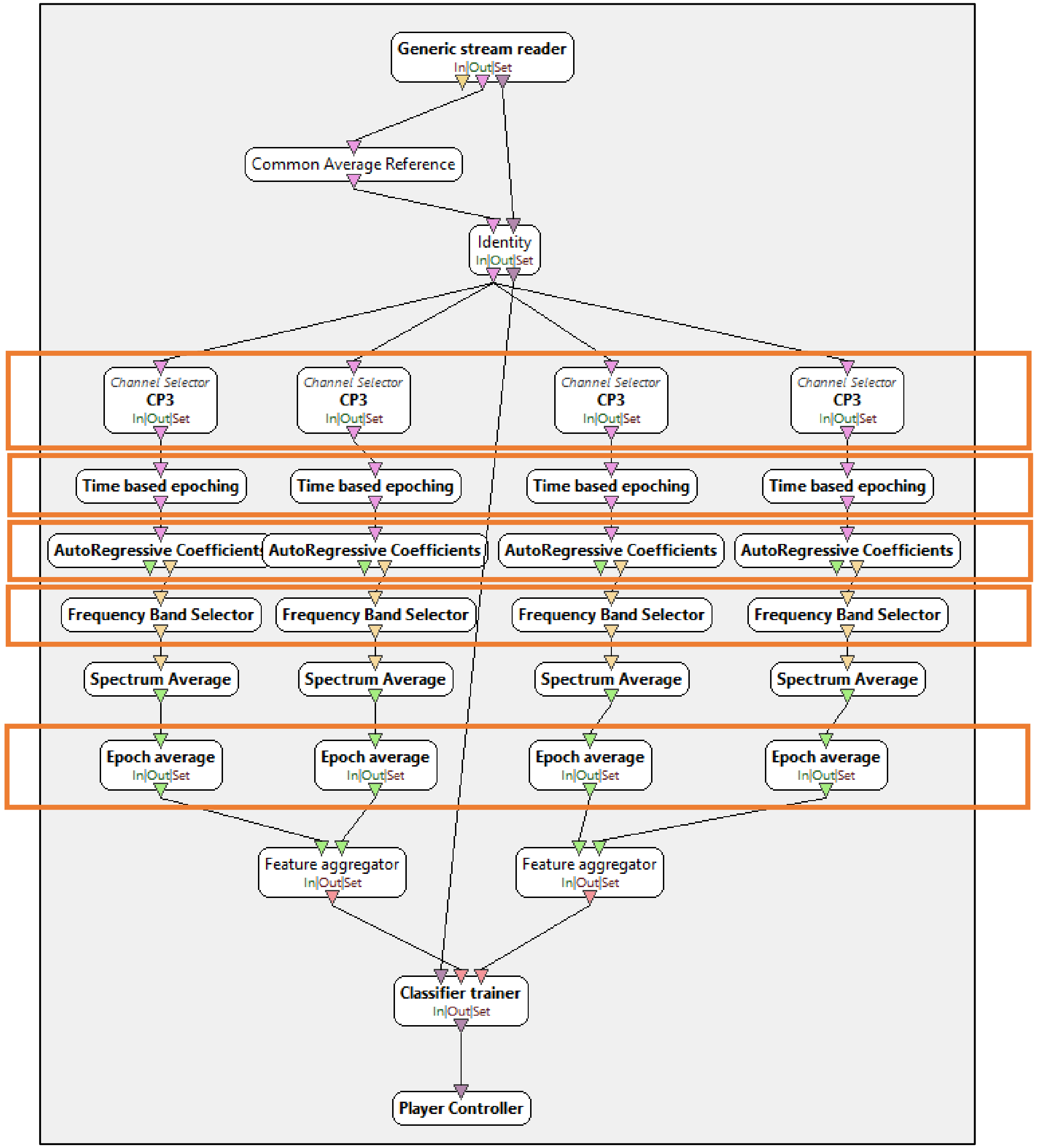


- **“Player Controller”:** Orchestrates the experiment course, by applying an action upon receiving a stimulation

GOING FURTHER…

SHARED PARAMETERS, METABOXES

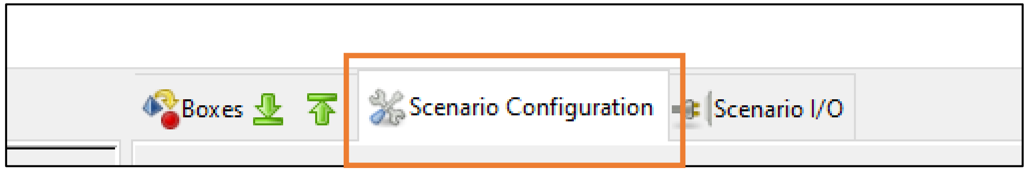
Shared parameters / “Scenario configuration”

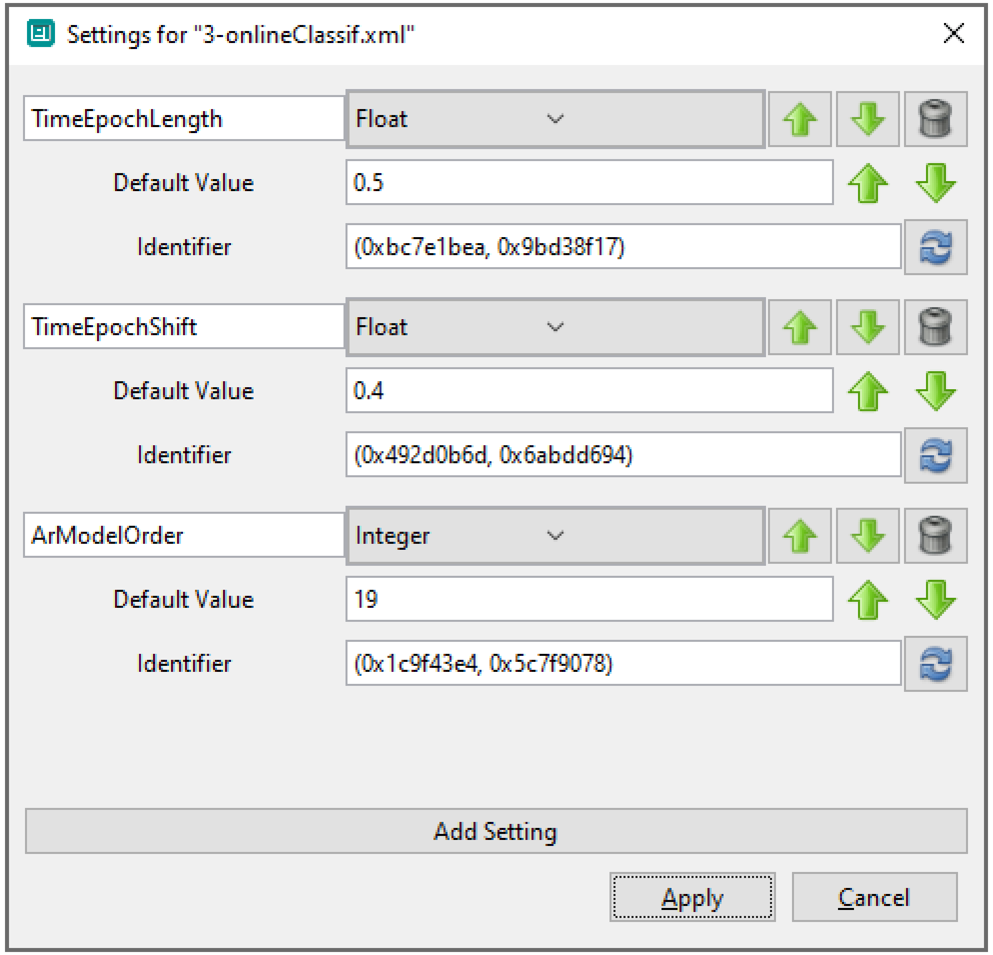


When using more complex scenarios, with parameters shared by multiple boxes…

- … using **shared parameters** can ease manipulations and avoid making mistakes!

Shared parameters / “Scenario configuration”




Settings for "3-onlineClassif.xml"

TimeEpochLength: Float
Default Value: 0.5
Identifier: (0xbc7e1bea, 0x9bd38f17)

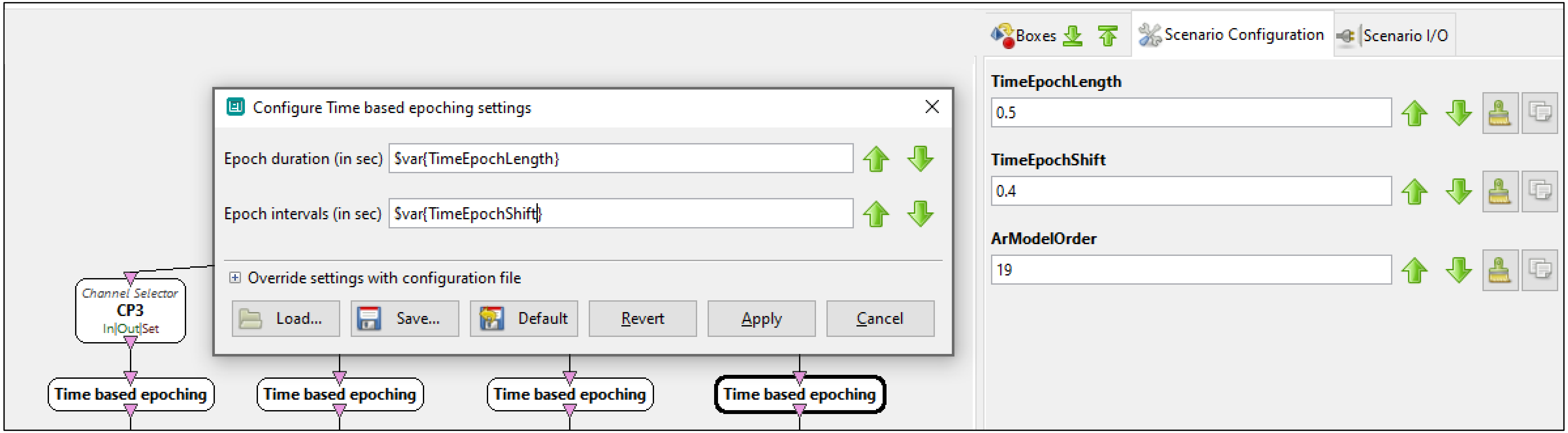
TimeEpochShift: Float
Default Value: 0.4
Identifier: (0x492d0b6d, 0x6abdd694)

ArModelOrder: Integer
Default Value: 19
Identifier: (0x1c9f43e4, 0x5c7f9078)

When using more complex scenarios, with parameters shared by multiple boxes…

- … using **shared parameters** can ease manipulations and avoid making mistakes!
- **Set up shared parameters** in the “Scenario configuration” tab, next to “Boxes” in the right-part of the Designer GUI.
- Add/edit your parameters by clicking on **“Configure settings”**

Shared parameters / “Scenario configuration”



Configure Time based epoching settings

Epoch duration (in sec): $var{TimeEpochLength}

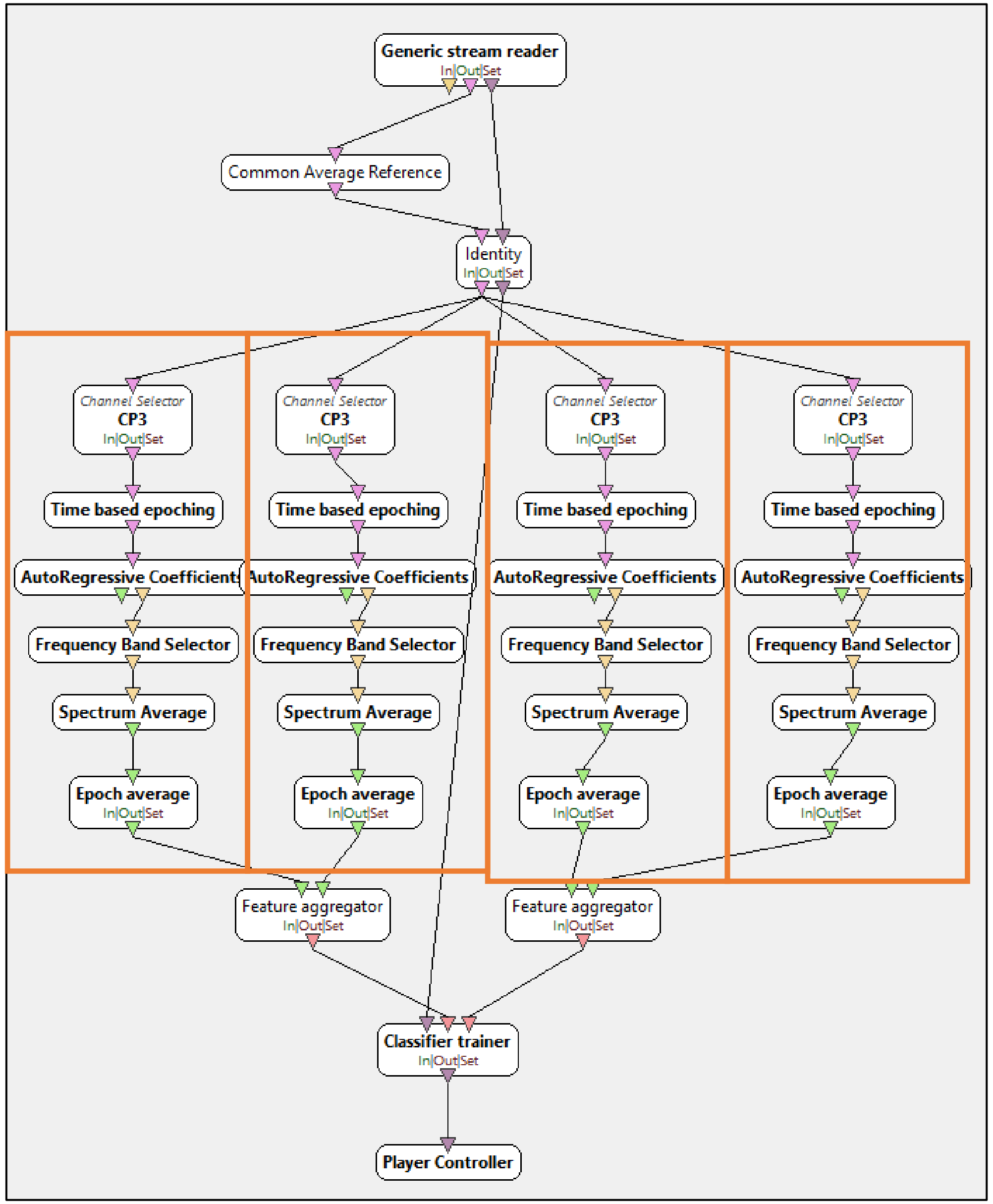
Epoch intervals (in sec): $var{TimeEpochShift}

TimeEpochLength: 0.5

TimeEpochShift: 0.4

ArModelOrder: 19

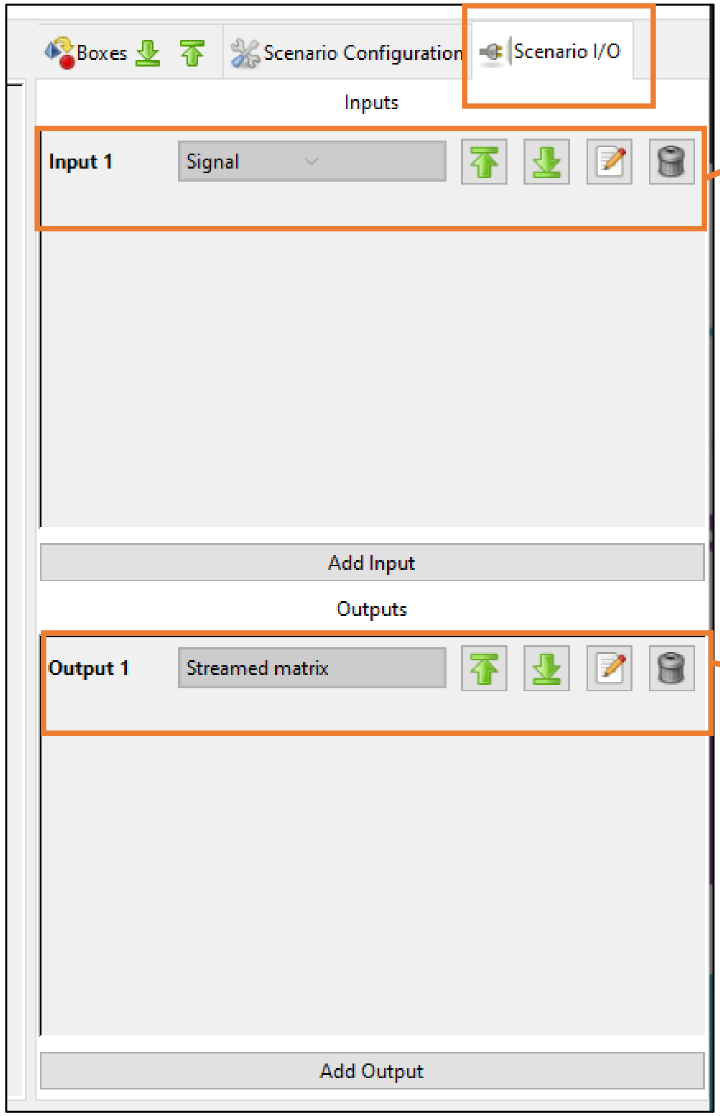
Metaboxes



- “**Metaboxes**” can be useful to replace recurring/repeating processing chains.

- Use them in conjunction with scenario configurations to simplify your scenarios

Metaboxes



Boxes | Scenario Configuration | Scenario I/O

Inputs

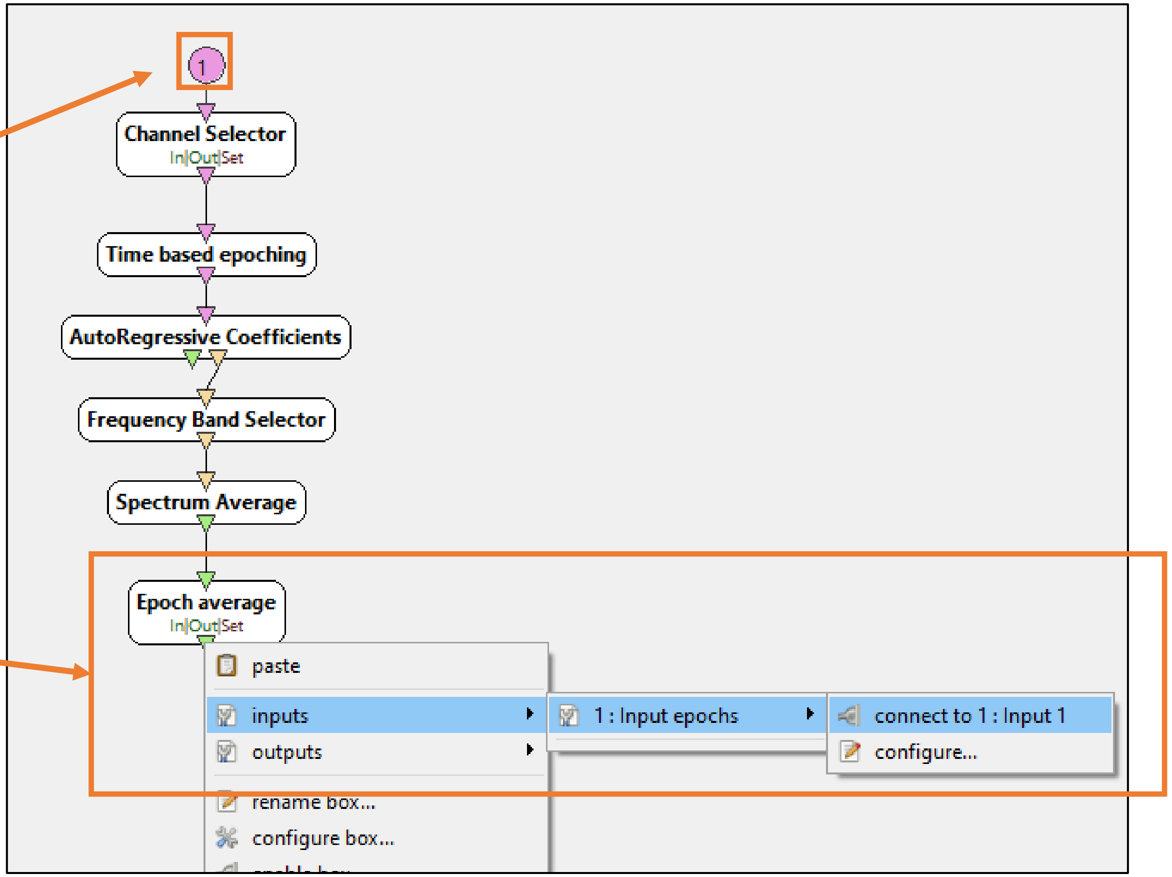
Input 1 — Signal

Add Input

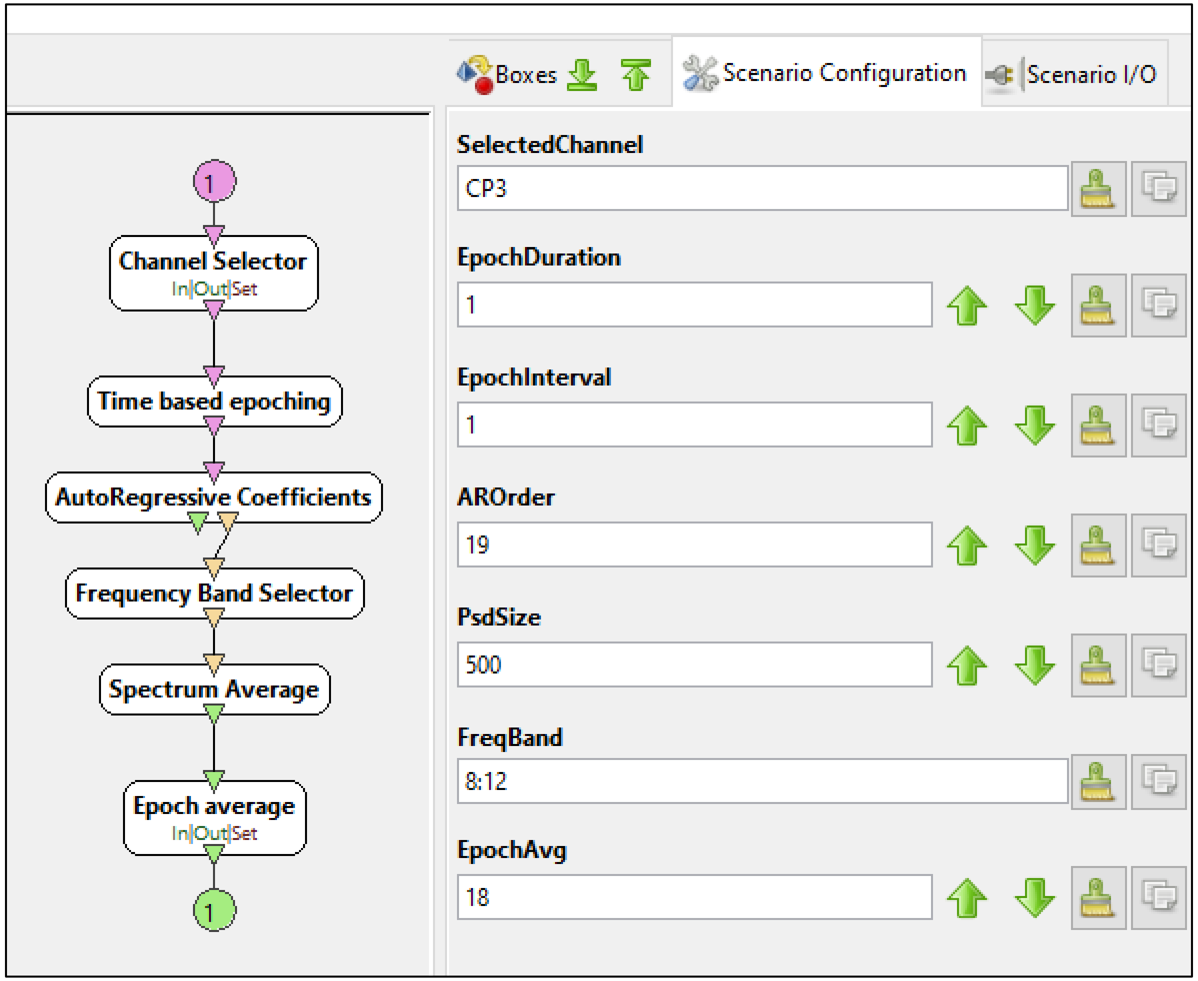
Outputs

Output 1 — Streamed matrix

Add Output



Metaboxes



SelectedChannel: CP3

EpochDuration: 1

EpochInterval: 1

AROrder: 19

PsdSize: 500

FreqBand: 8:12

EpochAvg: 18

Metaboxes

- **Saving your metaboxes:**

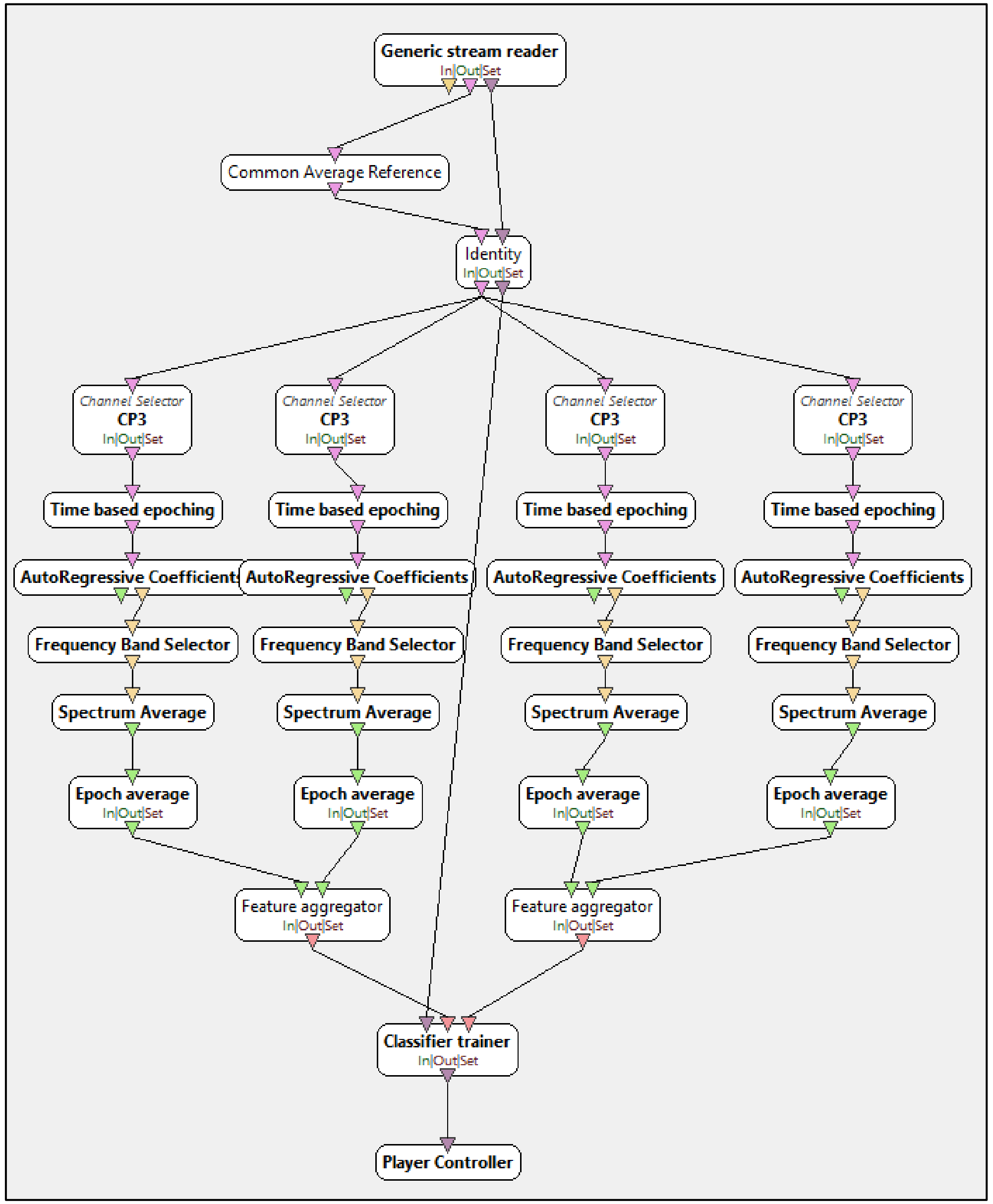
```
<install folder>\share\openvibe\metaboxes\
```

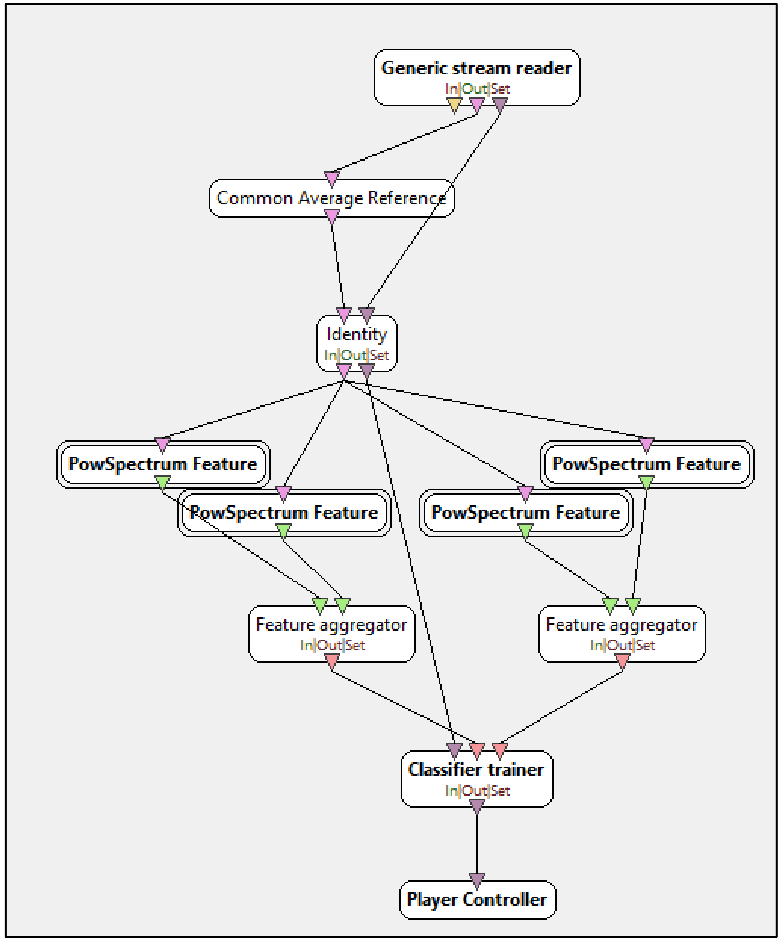
… they will appear in the “Metabox” folder the Box Browser next time you start the Designer.

- **Full tutorial:**

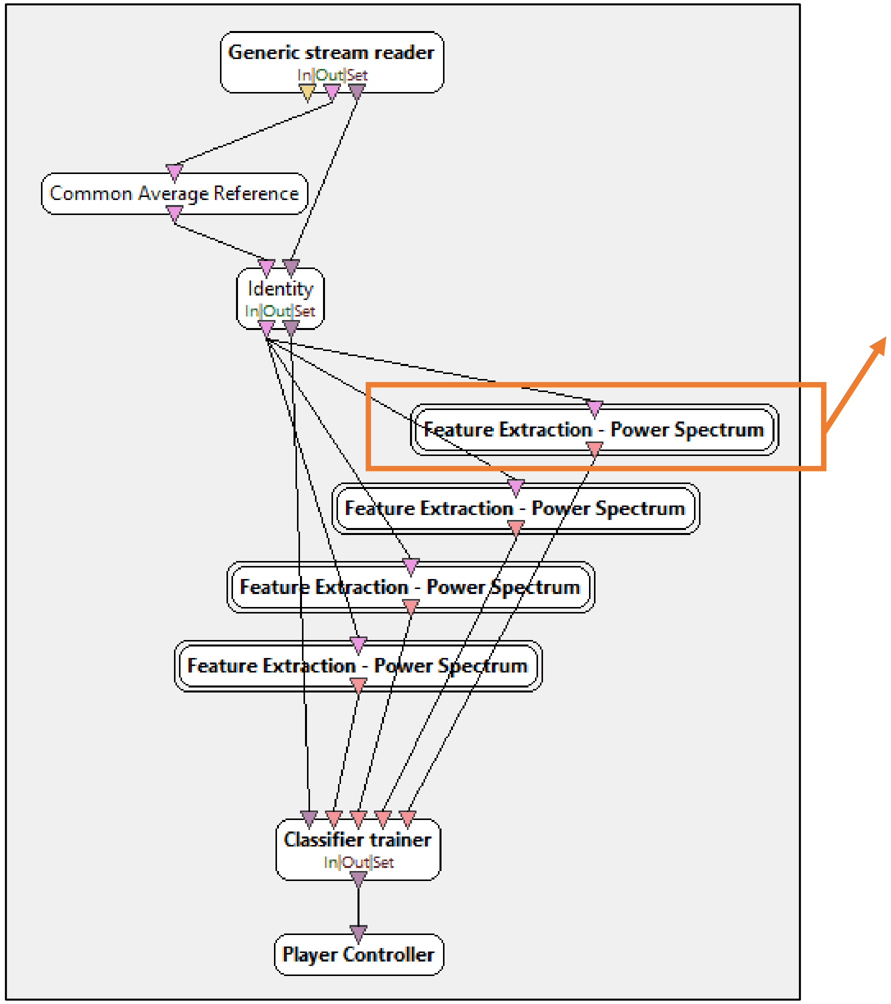
  http://openvibe.inria.fr/designer-tutorial-5-metaboxes/

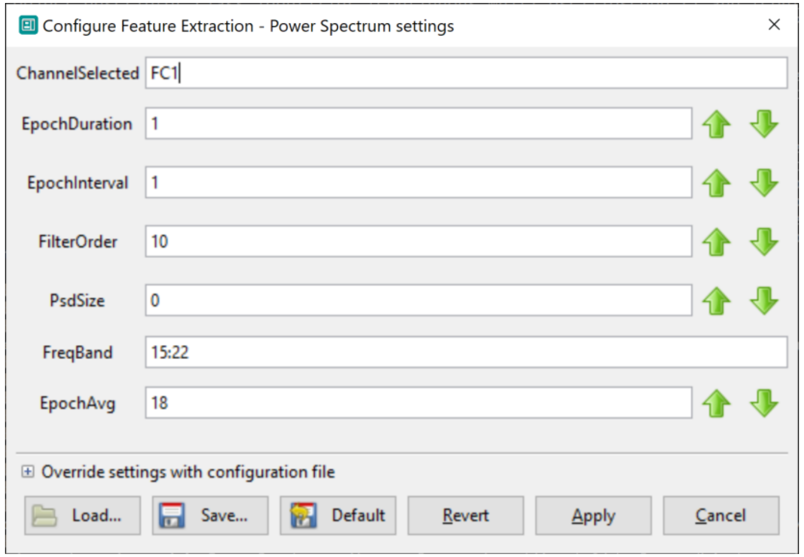
Metaboxes


Generic stream reader
In|Out|Set
Common Average Reference
Identity
In|Out|Set
Channel Selector
CP3
In|Out|Set
Channel Selector
CP3
In|Out|Set
Channel Selector
CP3
In|Out|Set
Channel Selector
CP3
In|Out|Set
Time based epoching
Time based epoching
Time based epoching
Time based epoching
AutoRegressive Coefficient
AutoRegressive Coefficients
AutoRegressive Coefficients
AutoRegressive Coefficients
Frequency Band Selector
Frequency Band Selector
Frequency Band Selector
Frequency Band Selector
Spectrum Average
Spectrum Average
Spectrum Average
Spectrum Average
Epoch average
In|Out|Set
Epoch average
In|Out|Set
Epoch average
In|Out|Set
Epoch average
In|Out|Set
Feature aggregator
In|Out|Set
Feature aggregator
In|Out|Set
Classifier trainer
In|Out|Set
Player Controller


Generic stream reader
In|Out|Set
Common Average Reference
Identity
In|Out|Set
PowSpectrum Feature
PowSpectrum Feature
PowSpectrum Feature
PowSpectrum Feature
Feature aggregator
In|Out|Set
Feature aggregator
In|Out|Set
Classifier trainer
In|Out|Set
Player Controller

Metaboxes


Generic stream reader
In|Out|Set
Common Average Reference
Identity
In|Out|Set
Feature Extraction - Power Spectrum
Feature Extraction - Power Spectrum
Feature Extraction - Power Spectrum
Feature Extraction - Power Spectrum
Classifier trainer
In|Out|Set
Player Controller


Configure Feature Extraction - Power Spectrum settings

ChannelSelected: FC1

EpochDuration: 1

EpochInterval: 1

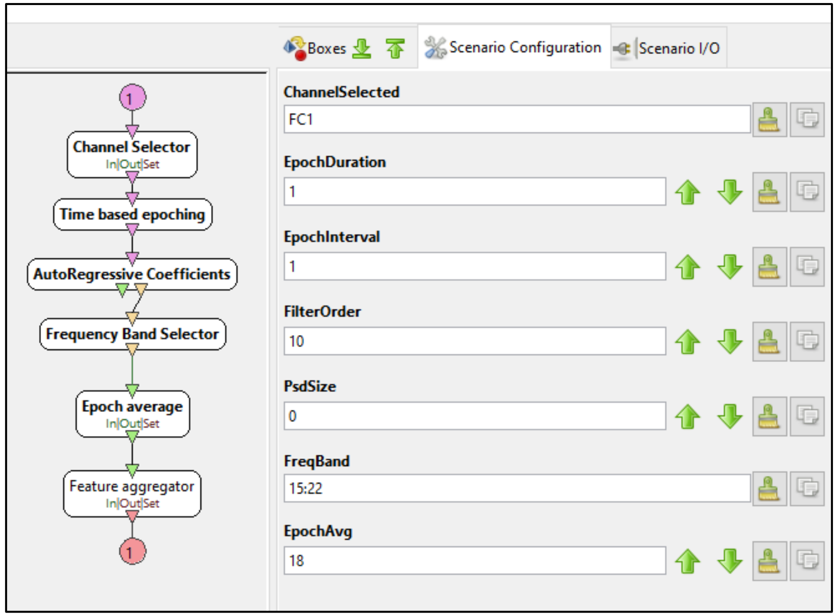
FilterOrder: 10

PsdSize: 0

FreqBand: 15:22

EpochAvg: 18

The **metabox's parameters** are the ones set in the "scenario configuration" of the metabox scenario



ChannelSelected: FC1

EpochDuration: 1

EpochInterval: 1

FilterOrder: 10

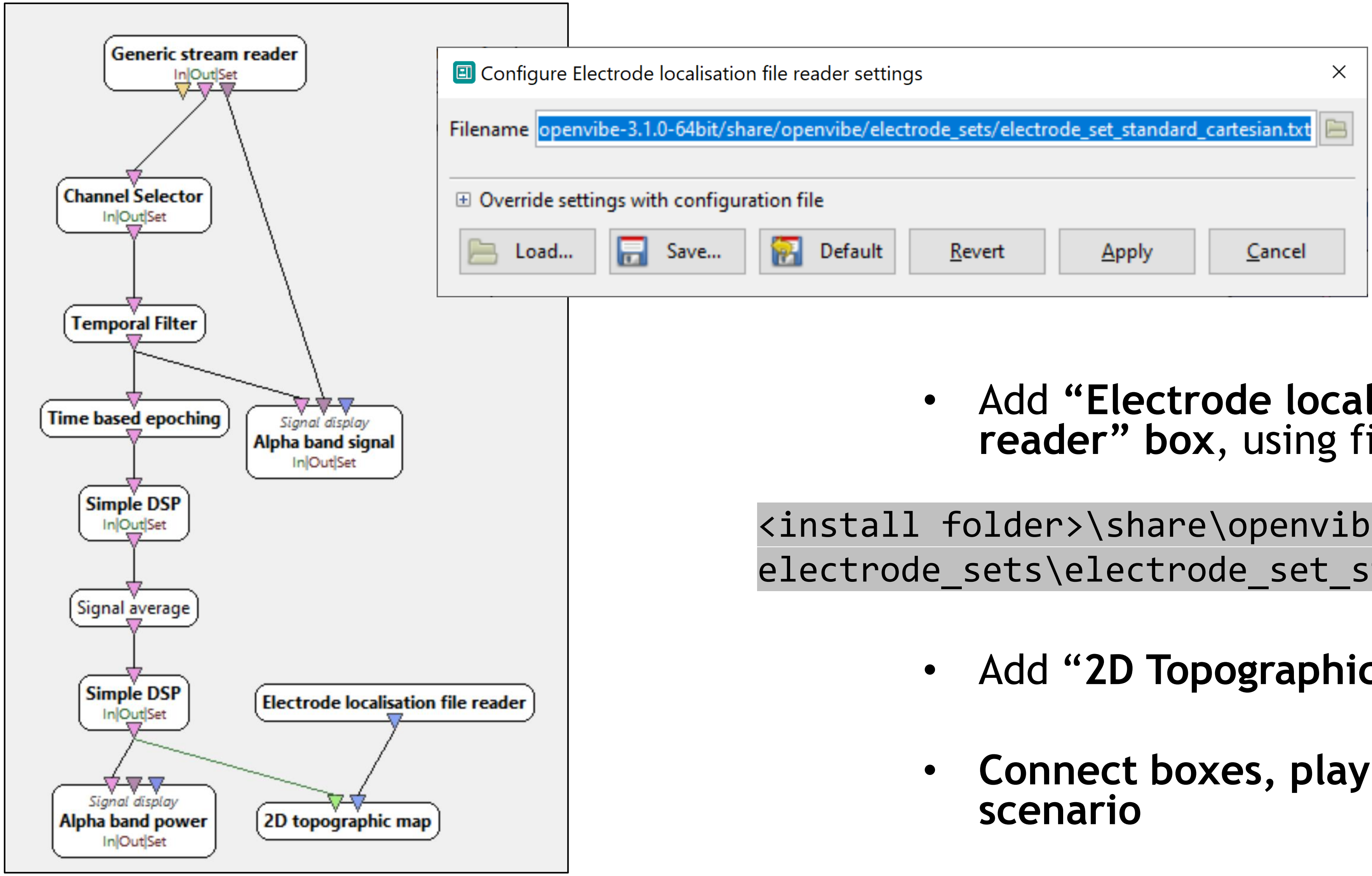
PsdSize: 0

FreqBand: 15:22

EpochAvg: 18

TOPOGRAPHY VISUALIZATION

Acquisition - Protocol mgmt. & Stimulations



Configure Electrode localisation file reader settings

Filename: openvibe-3.1.0-64bit/share/openvibe/electrode_sets/electrode_set_standard_cartesian.txt

⊞ Override settings with configuration file

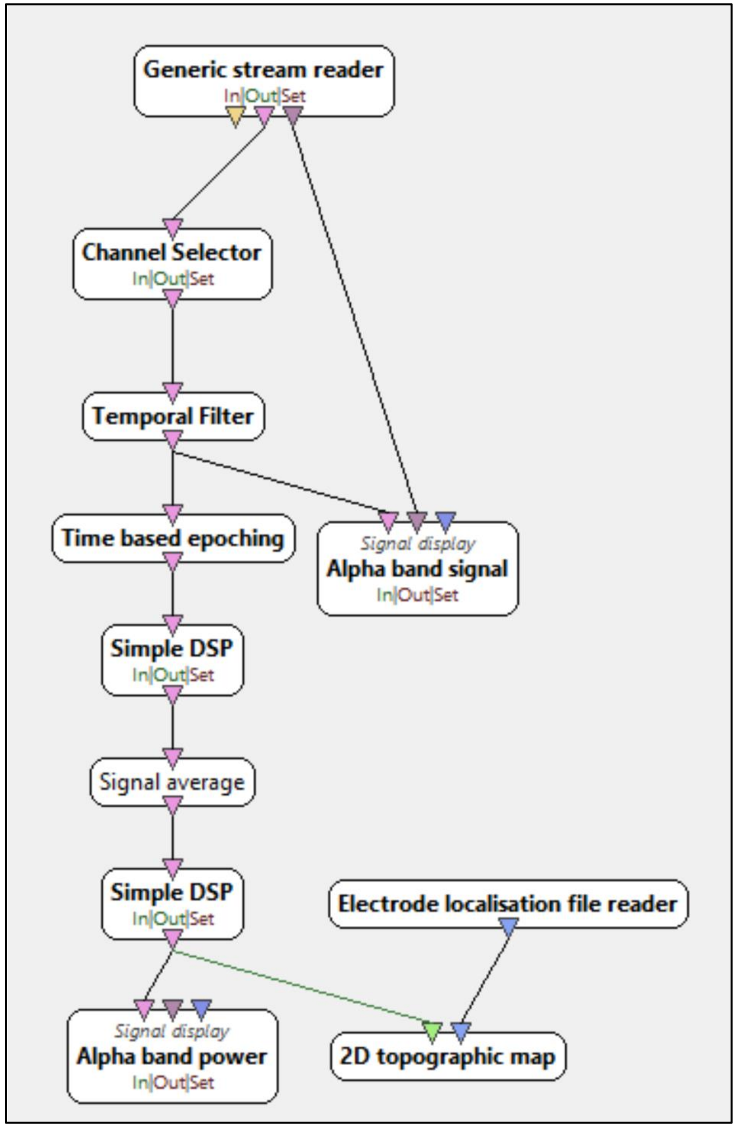
- Add “**Electrode localization file reader**” **box**, using file:

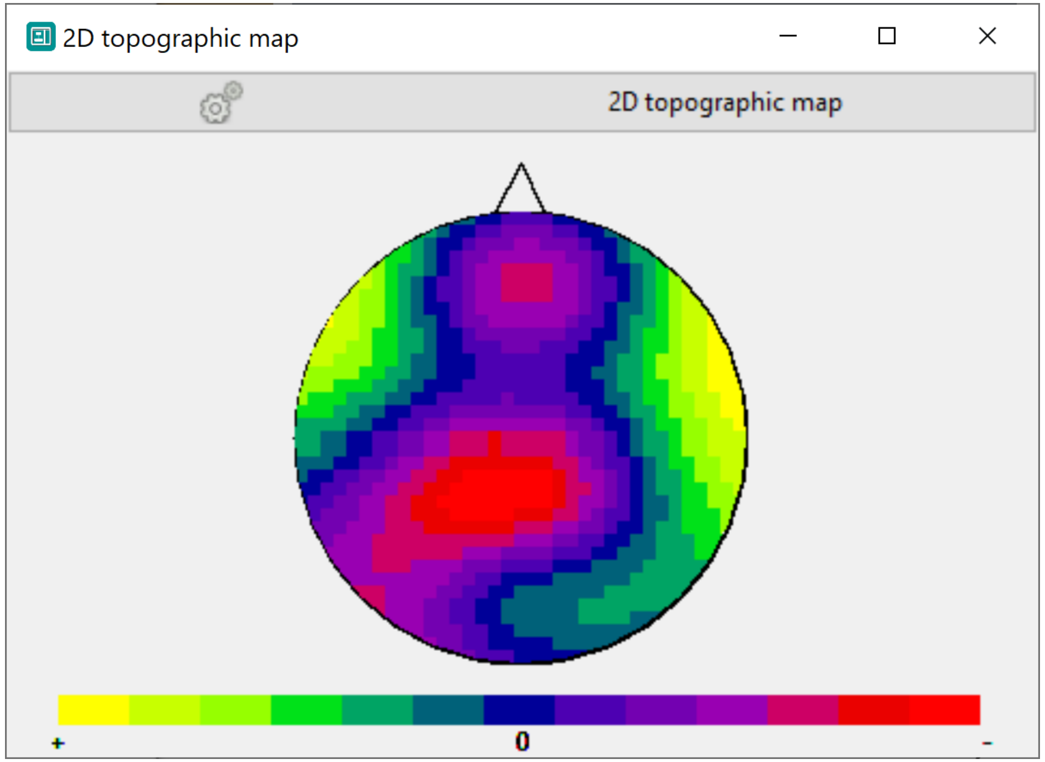
```
<install folder>\share\openvibe\
electrode_sets\electrode_set_standard_cartesian.txt
```

- Add “**2D Topographic map**”

- **Connect boxes, play the scenario**

Acquisition - Protocol mgmt. & Stimulations


Generic stream reader
In|Out|Set
Channel Selector
In|Out|Set
Temporal Filter
Time based epoching
Signal display
Alpha band signal
In|Out|Set
Simple DSP
In|Out|Set
Signal average
Simple DSP
In|Out|Set
Electrode localisation file reader
Signal display
Alpha band power
In|Out|Set
2D topographic map


2D topographic map
2D topographic map
+
0
-