WS: BRAIN-COMPUTER INTERFACE USING OPENVIBE, AN OPEN-SOURCE SOFTWARE PLATFORM – PART 2

Arthur Desbois & Marie-Constance Corsi

ARAMIS team, Paris Brain Institute

CHAPTER 1

PREREQUISITES BEFORE PERFORMING A MI-BCI EXPERIMENT

DIFFERENT TYPES OF BCI

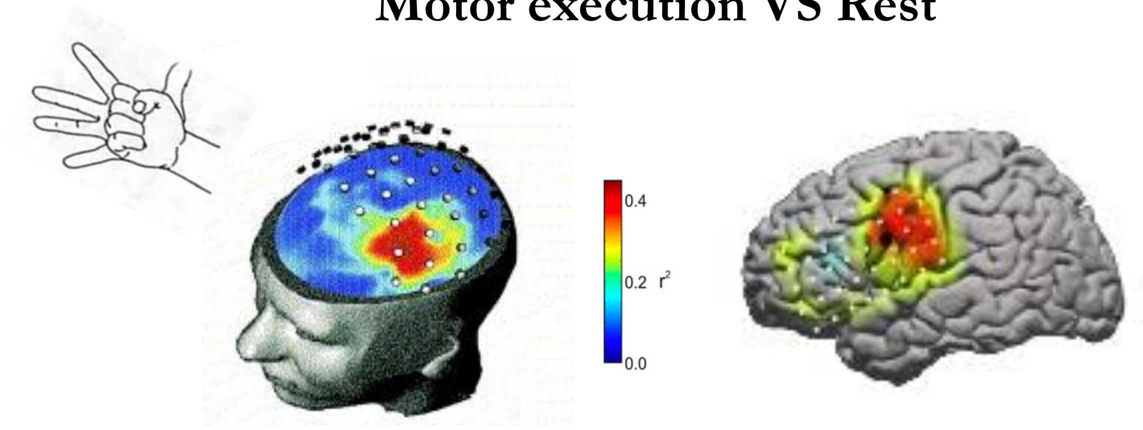
Underlying idea

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

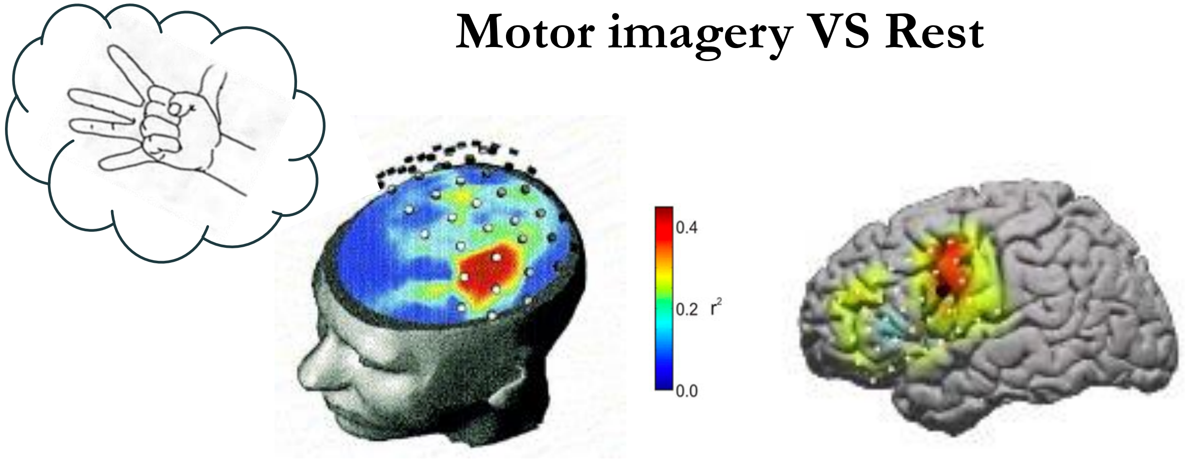
Illustration with Motor imagery-based BCI

MOTOR IMAGERY – OBSERVATIONS

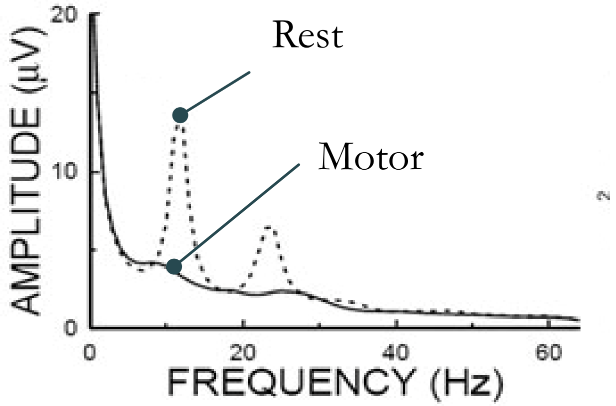
Motor execution VS Rest



Motor imagery VS Rest



Power decrease

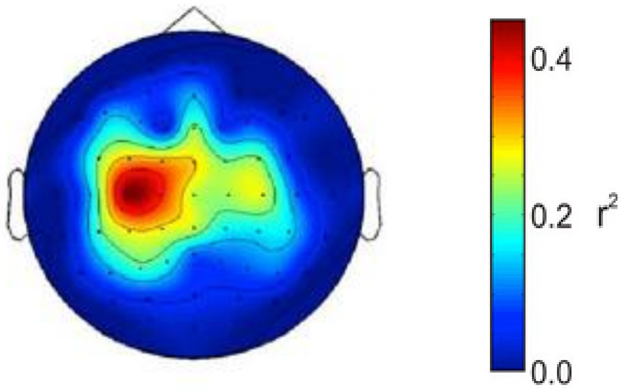


Desynchronization effect
(Pfurtscheller et al, 1999)

MOTOR IMAGERY – MU-BETA RHYTHM

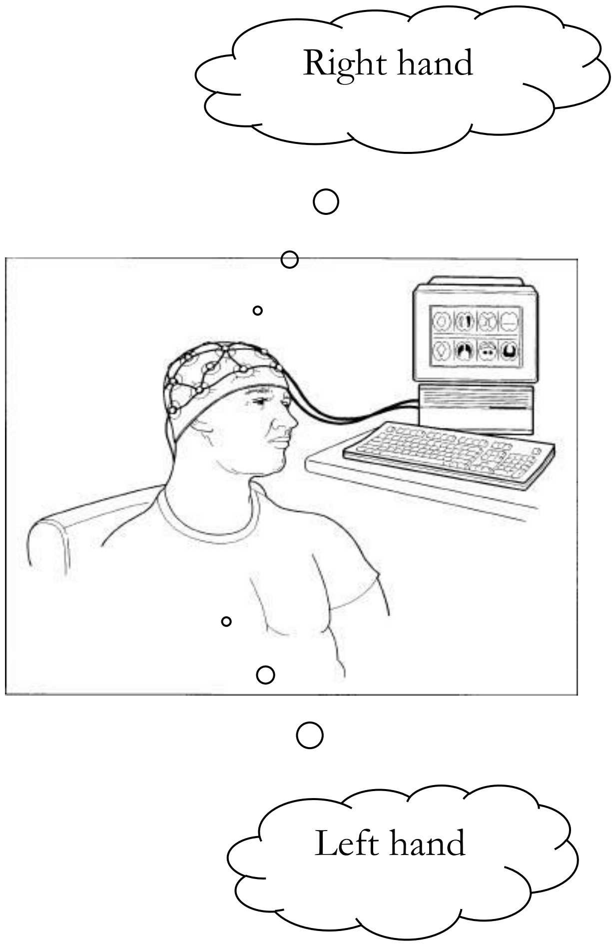
- Behavioral properties
  - Movement / preparation for movement : Event-related desynchronization (ERD) (Pfurtscheller, G, Lopes da Silva, FH, 1999)
  - With relaxation/post-movement period : ERS

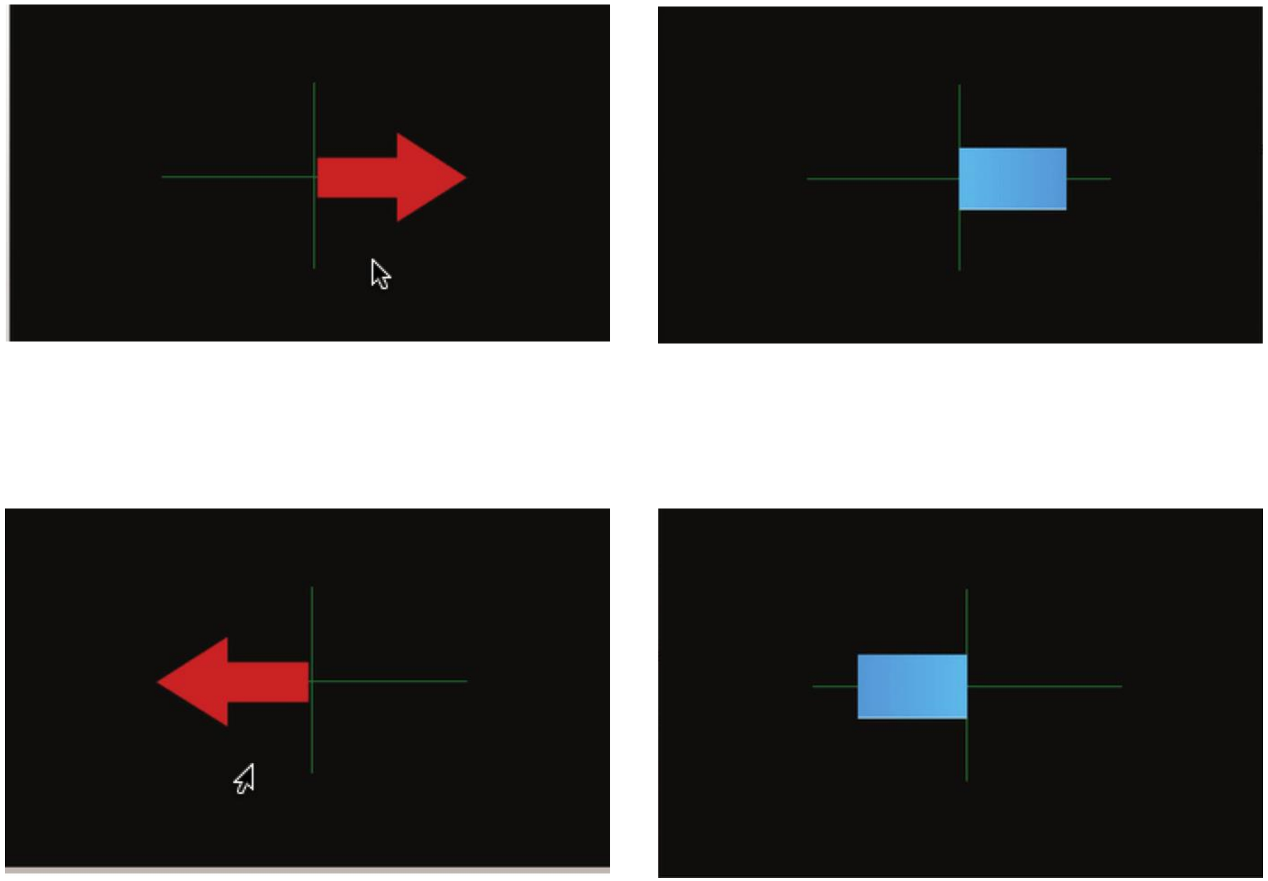
- Why using it in BCI ?
  - Mu/Beta activity modulation by motor-imagery, a way to communicate
  - Use of power spectra
  - To establish this communication :
    - Spatial selection
    - Frequency selection



Illustrations from BCI2000 website

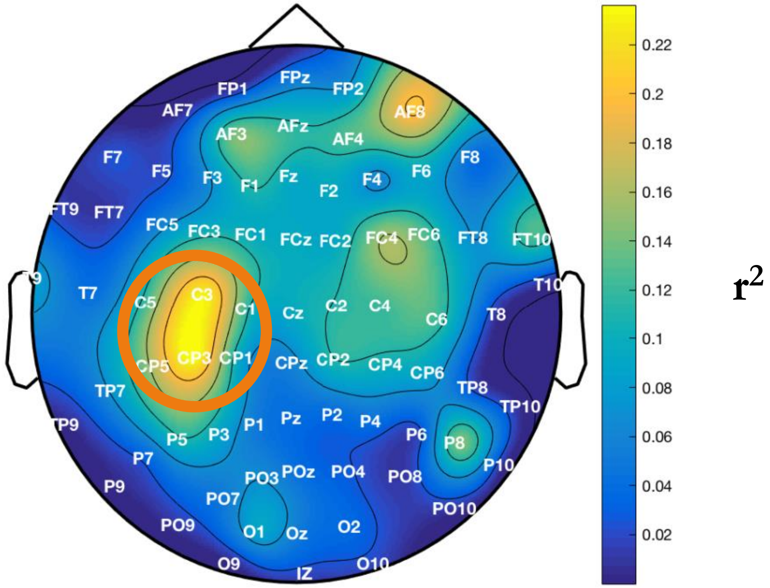
MOTOR IMAGERY – IN PRACTICE





MOTOR IMAGERY – IN PRACTICE

@ 10Hz



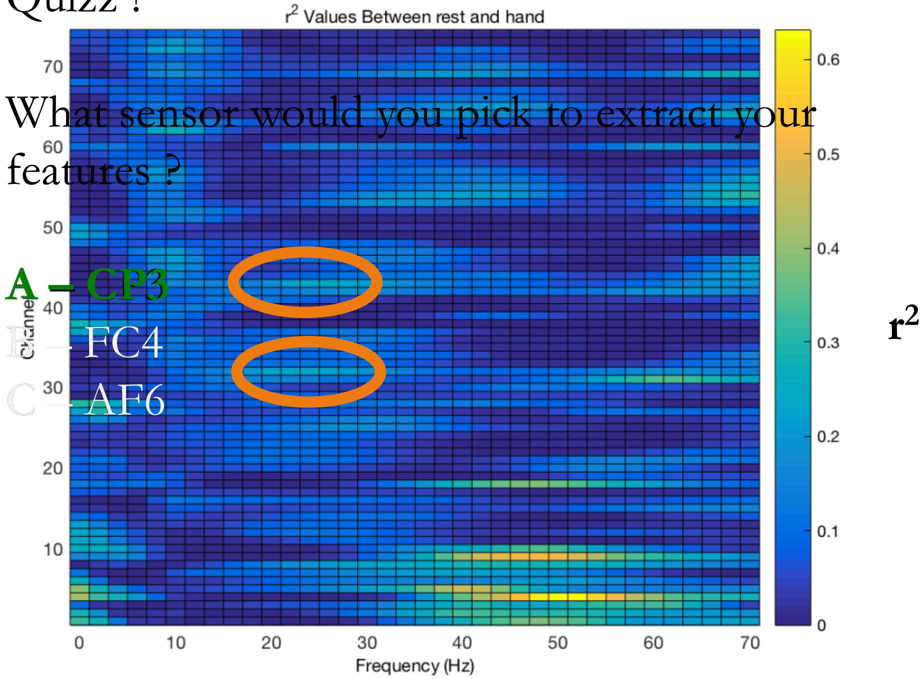
Quizz !

What sensor would you pick to extract your features ?

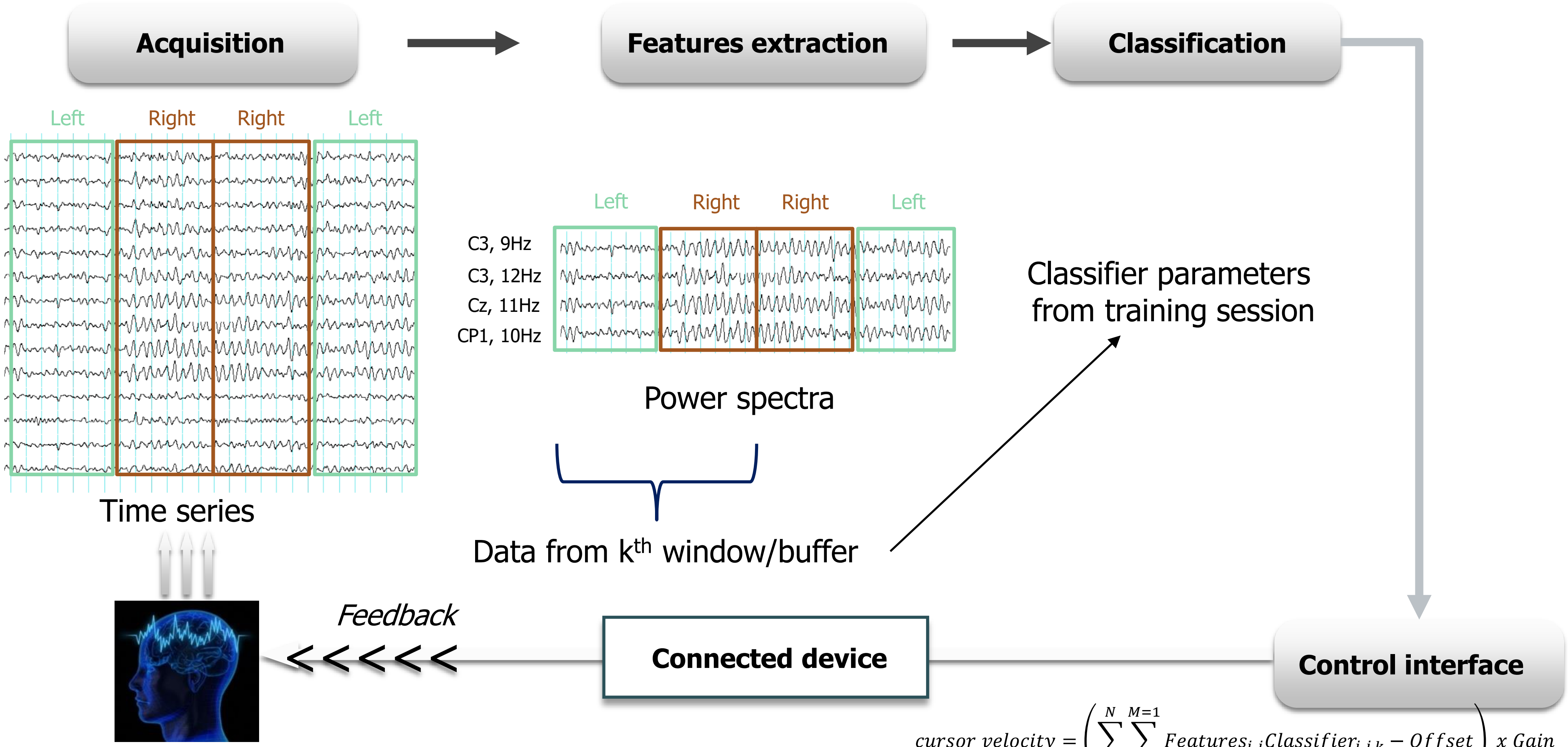
A – CP3

B – FC4

C – AF6

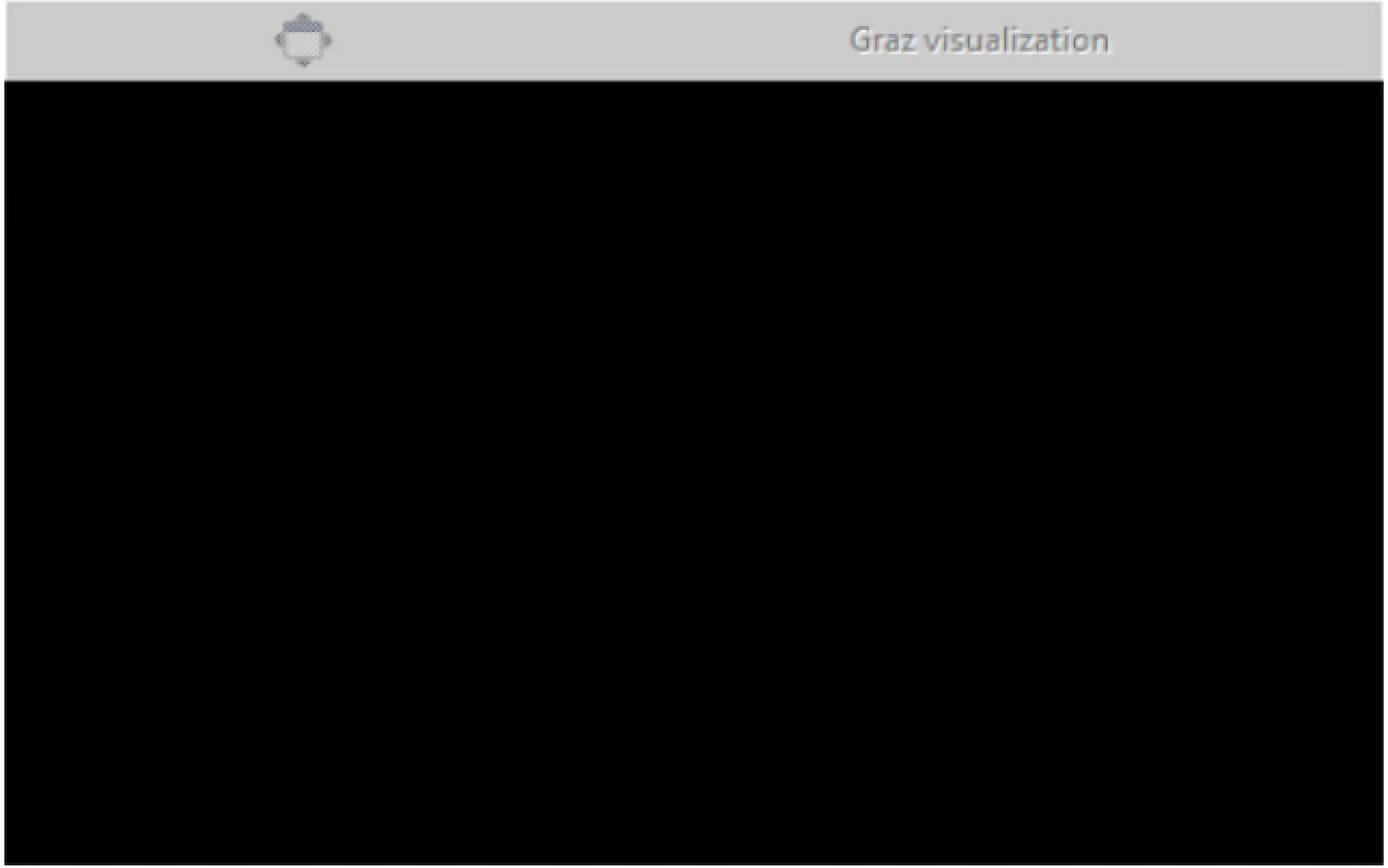


MOTOR IMAGERY – IN PRACTICE



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

MOTOR IMAGERY – IN PRACTICE



CHAPTER 2

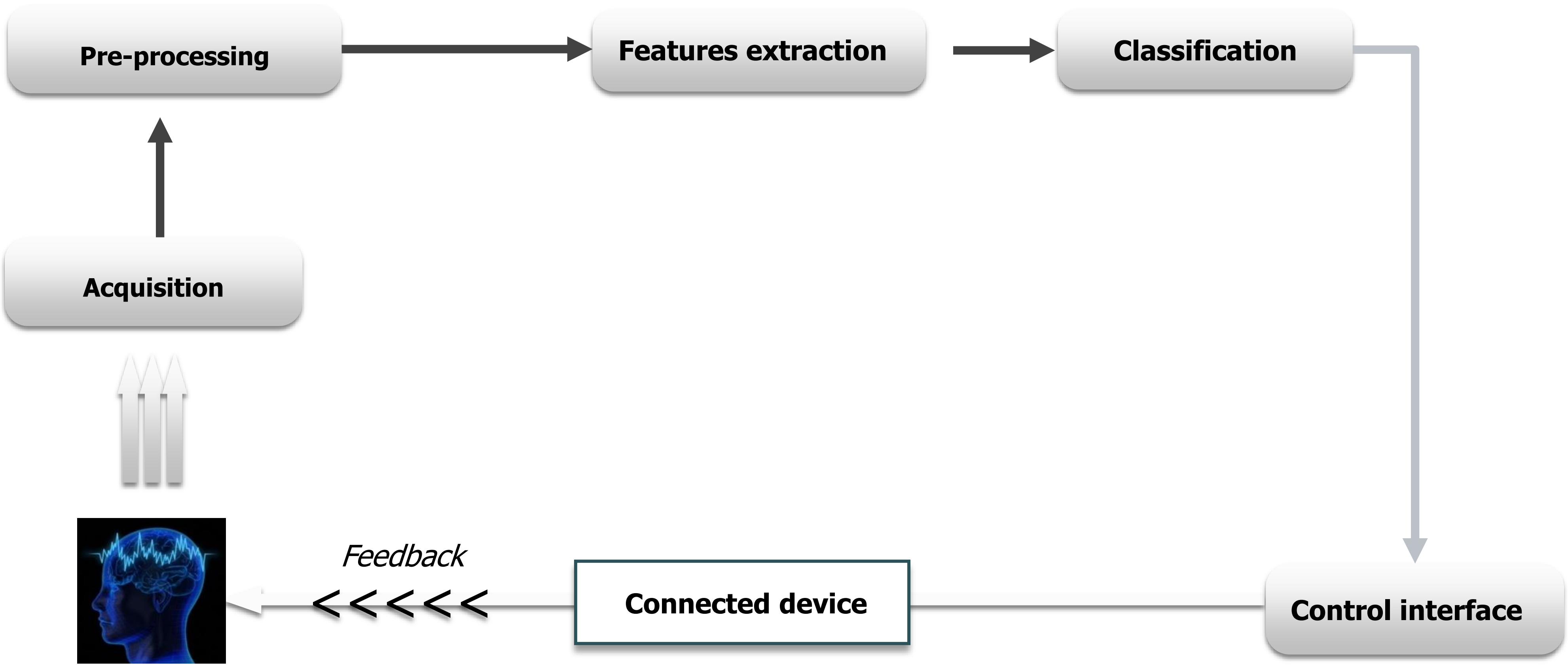
BCI-CUTTINGEEG PROTOCOL SET-UP

RESOURCES

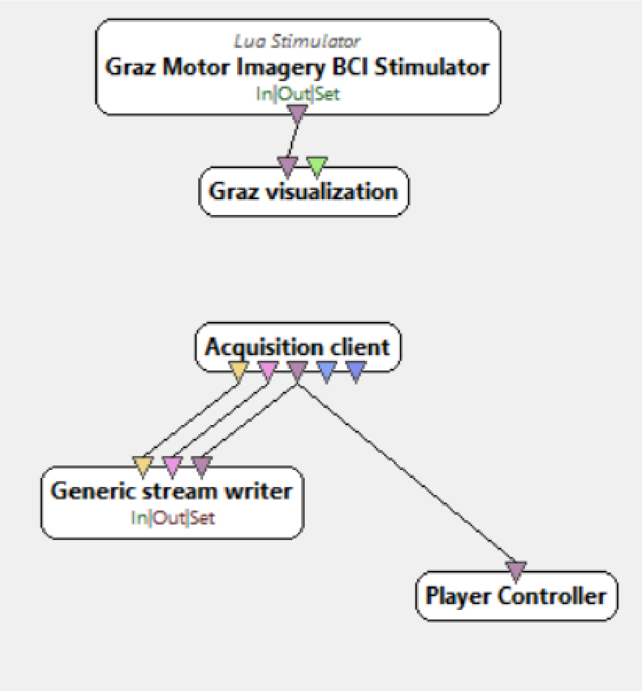
- GitHub repo:

  https://github.com/BCI-NET/BCI-OpenViBE-CuttingEEG2021/

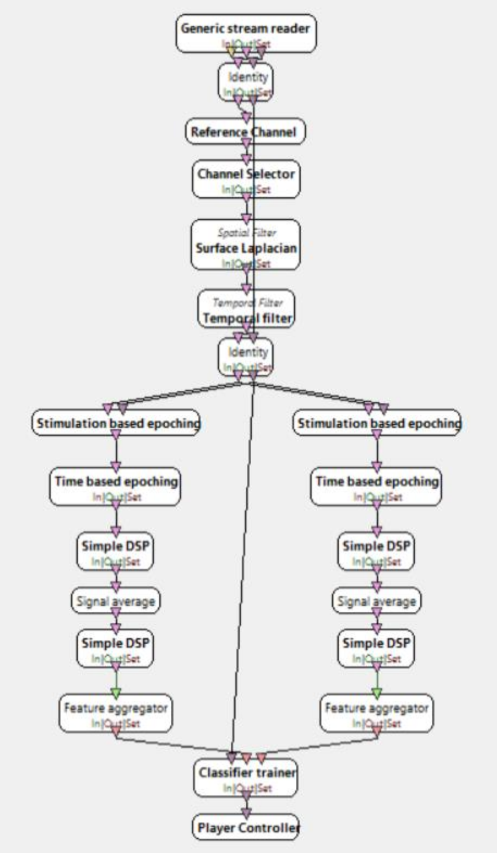
AIM OF THE PART



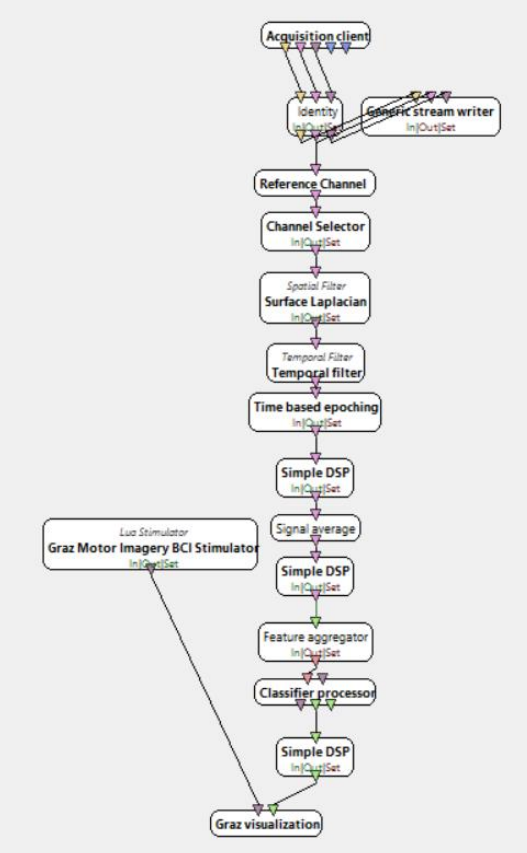
AIM OF THE PART



1- Data acquisition

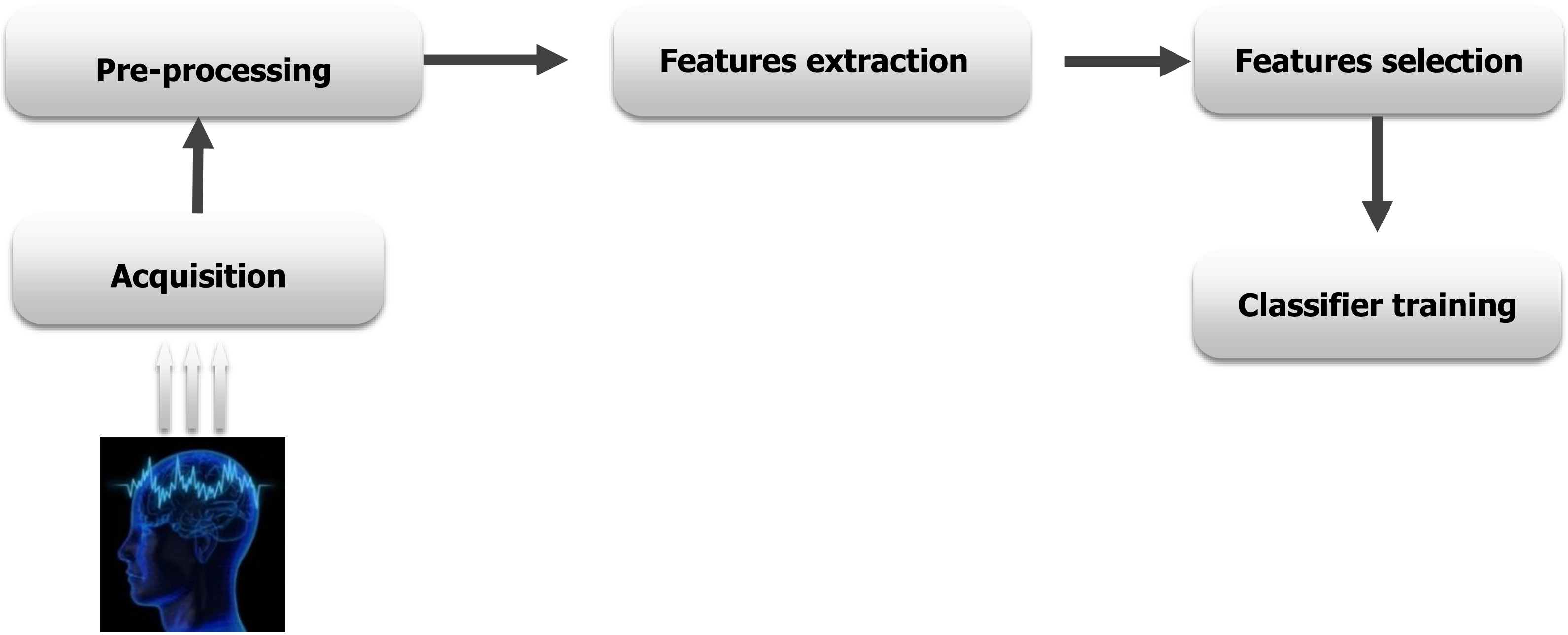


2- Features extraction
& Classification



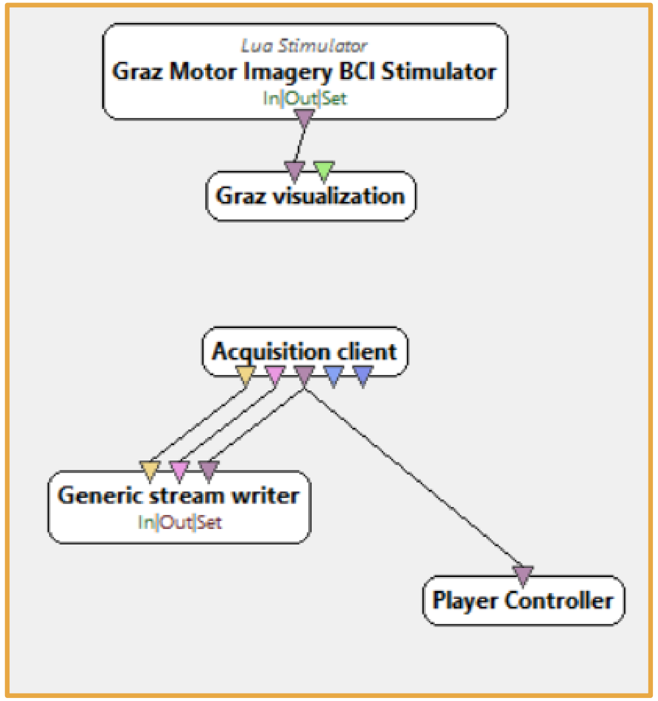
3- Online feedback

AIM 1 – BUILDING THE CLASSIFIER…



Adapted from (X. Navarro & F. Grosselin)

SC 1 – DATA ACQUISITION

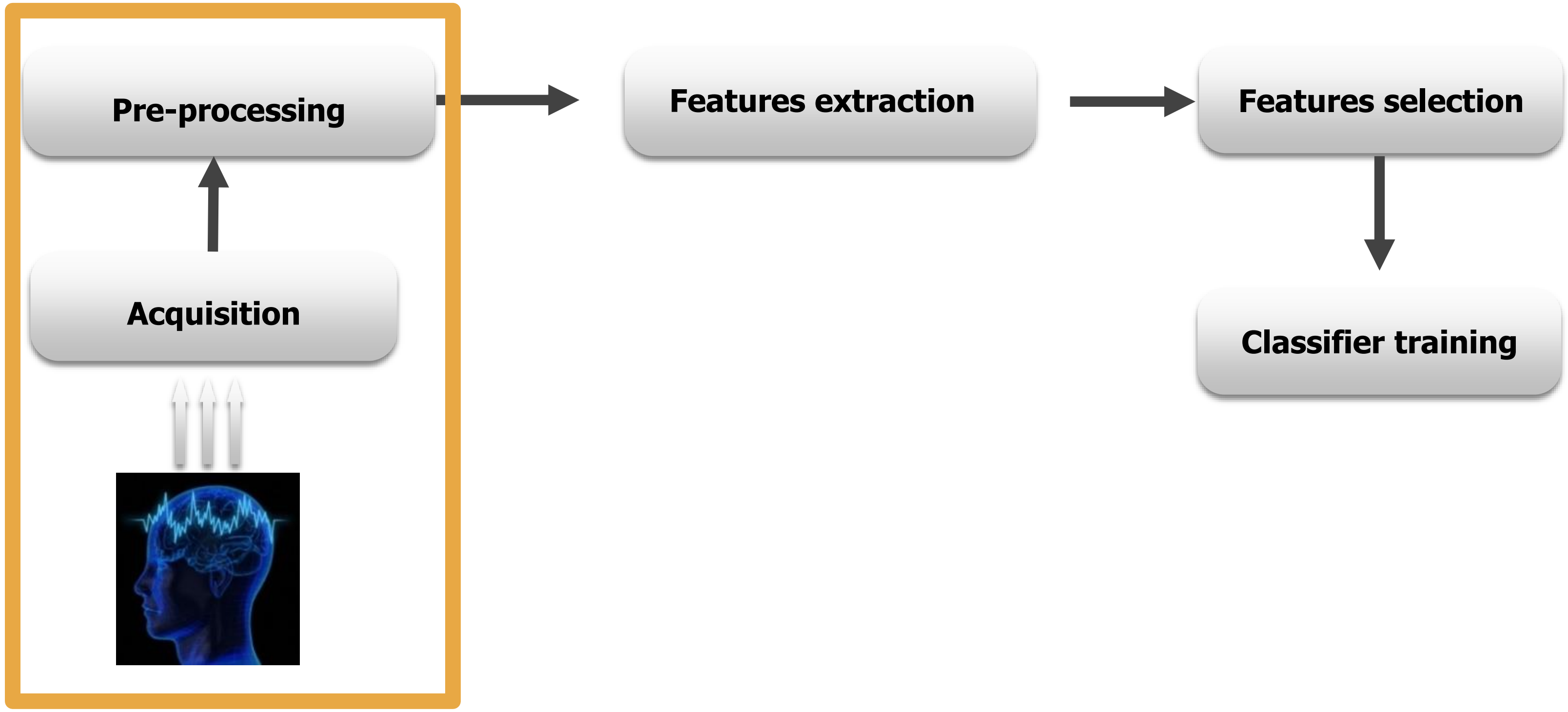


1- Data acquisition

2- Features extraction
& Classification

3- Online feedback

SC 1 – DATA ACQUISITION



Adapted from (X. Navarro & F. Grosselin)

SC1 – DATA ACQUISITION

- **Sub-chapters:**
  - Warm-up, first steps: simple I/O
  - Warm-up 2: acquisition server
  - BCI acquisition protocol, Stimulations

SC1 - STEP 1 – WARMUP ! SIMPLE I/O AND DISPLAY

- **Sub-chapters:**
  - Warm-up, first steps: simple I/O
  - Warm-up 2: acquisition server
  - BCI acquisition protocol, Stimulations

SC1 - STEP 1 – WARMUP ! SIMPLE I/O AND DISPLAY

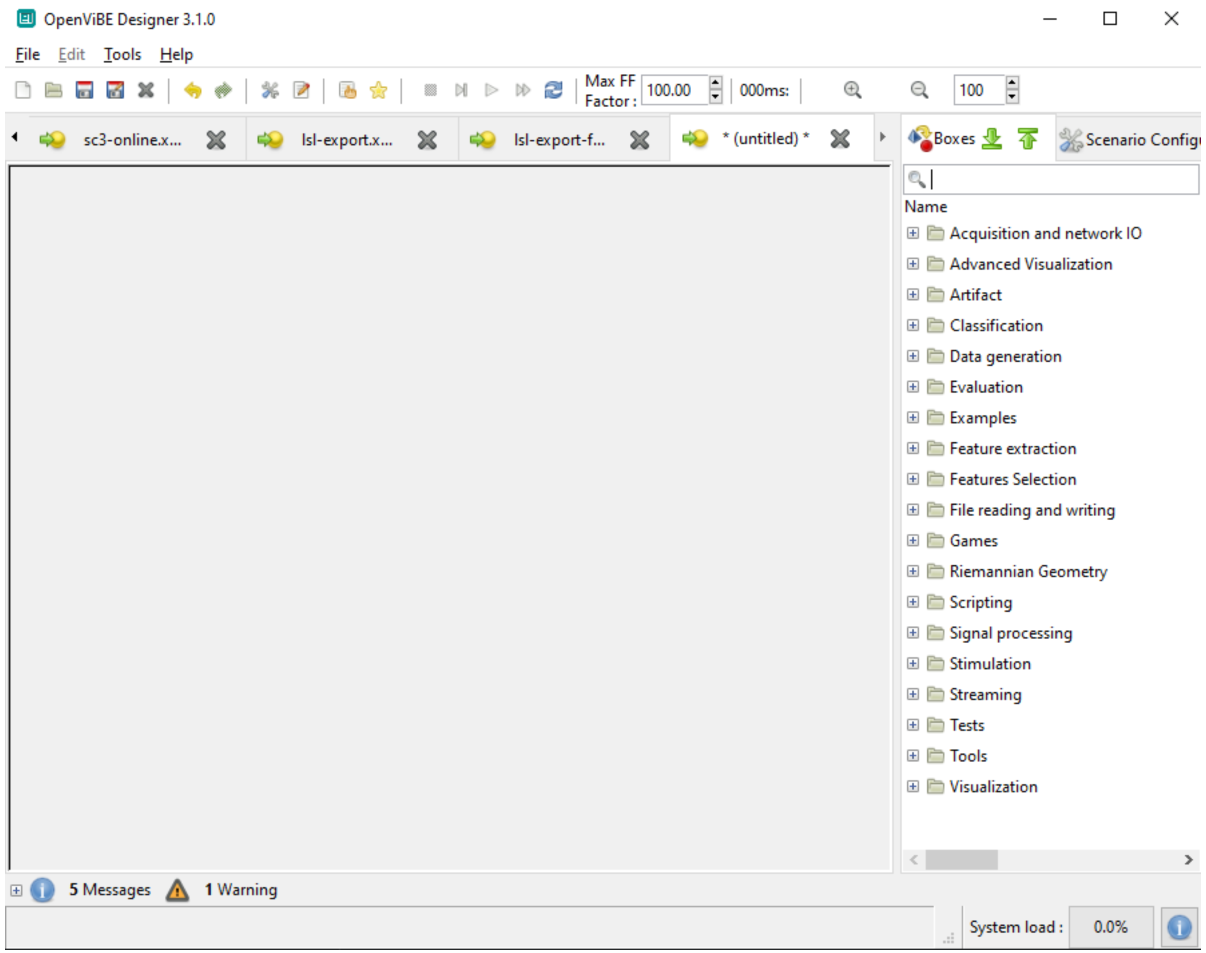
- **Goal : load & display file :**

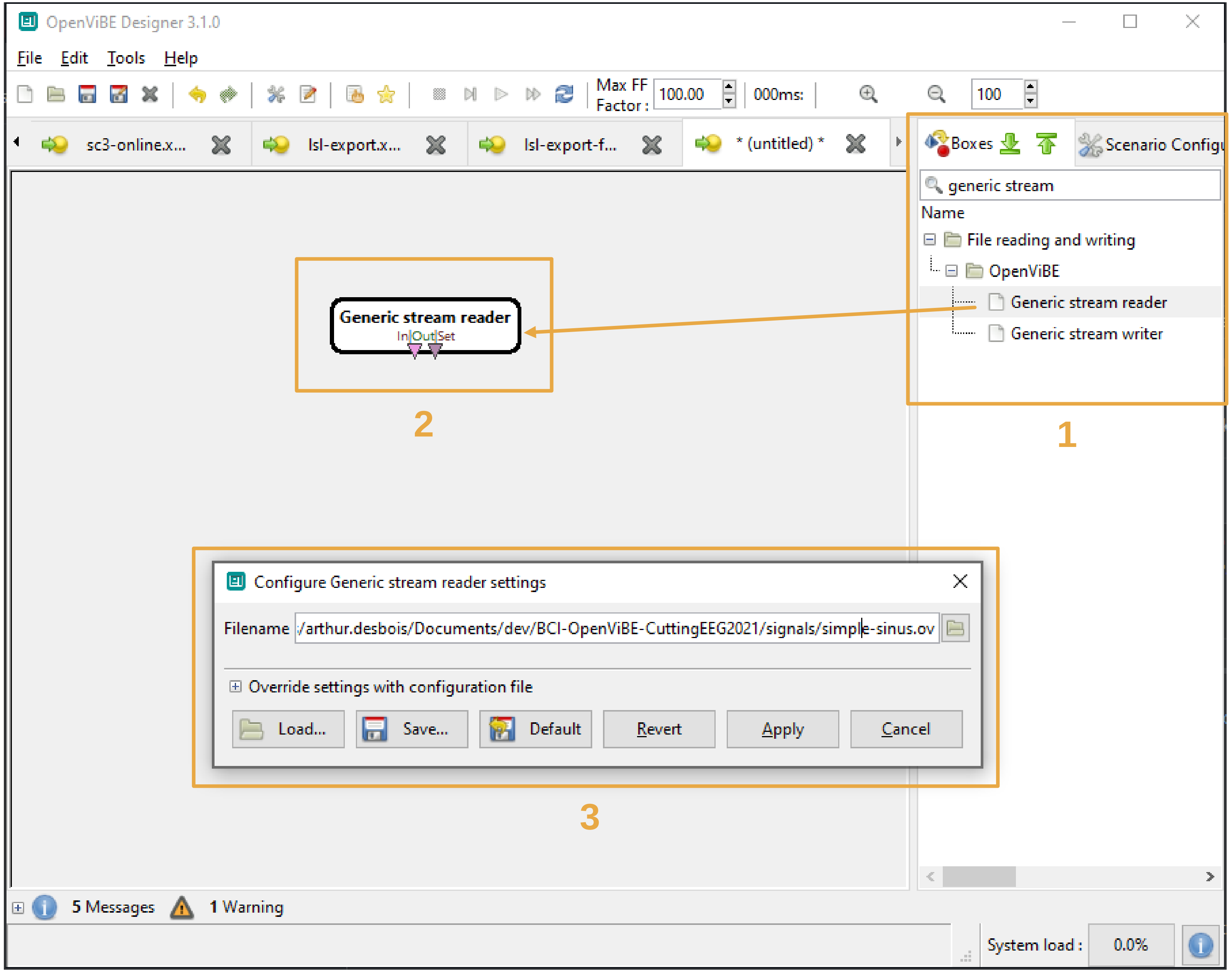
  signals/simple-sinus.ov

  (in Github repo)

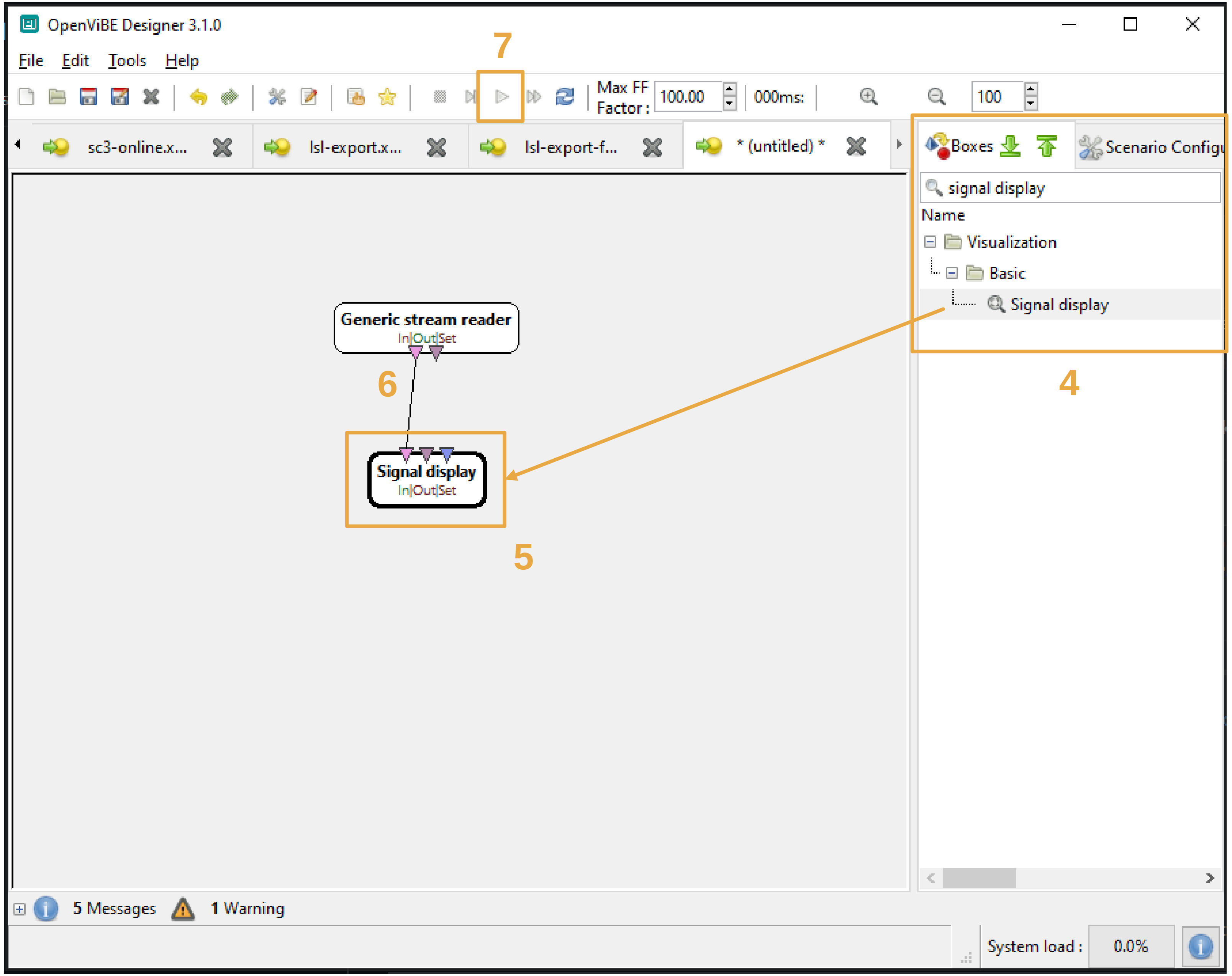
- **0- Launch OpenViBE Designer :**

  openvibe-designer.cmd





- 1- Search for “generic stream reader” in boxes list
- 2- Drag & drop to designer window
- 3- Set filename with browser



- 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!

SC1 - STEP 2 – WARMUP ! ACQ SERVER

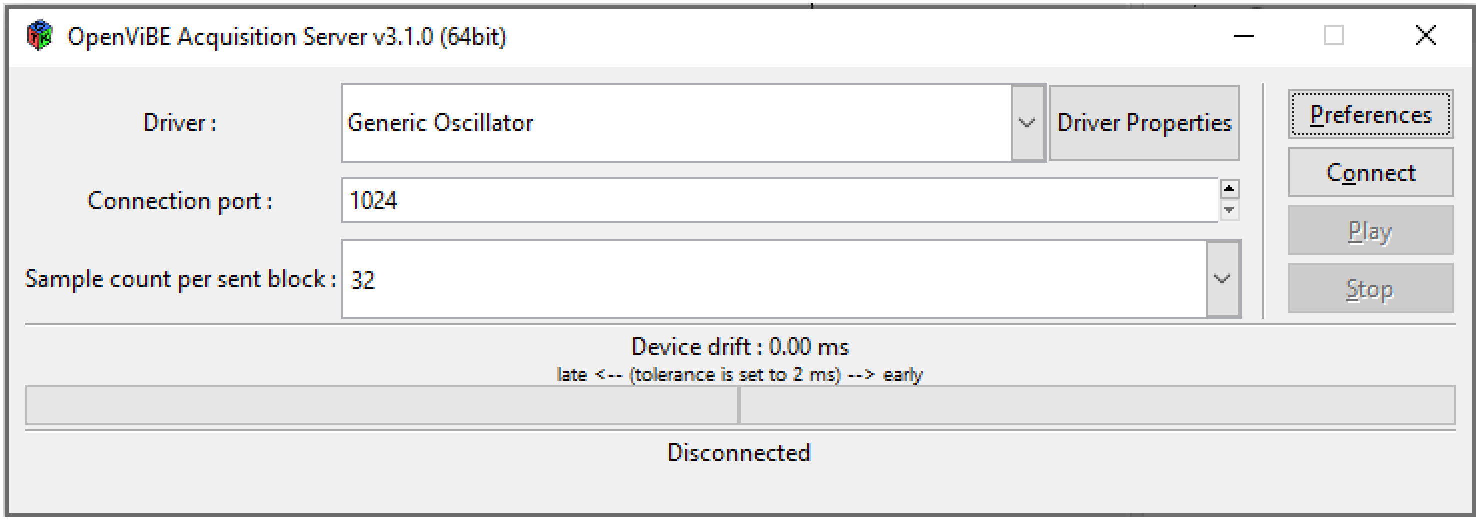
- **Sub-chapters:**
  - Warm-up, first steps: simple I/O
  - Warm-up 2: acquisition server
  - BCI acquisition protocol, Stimulations

SC1 - STEP 2 – WARMUP ! ACQ SERVER

- **Goal : use acquisition server/client** with a simple sine generator

- **0- Launch OpenViBE Acquisition server :**

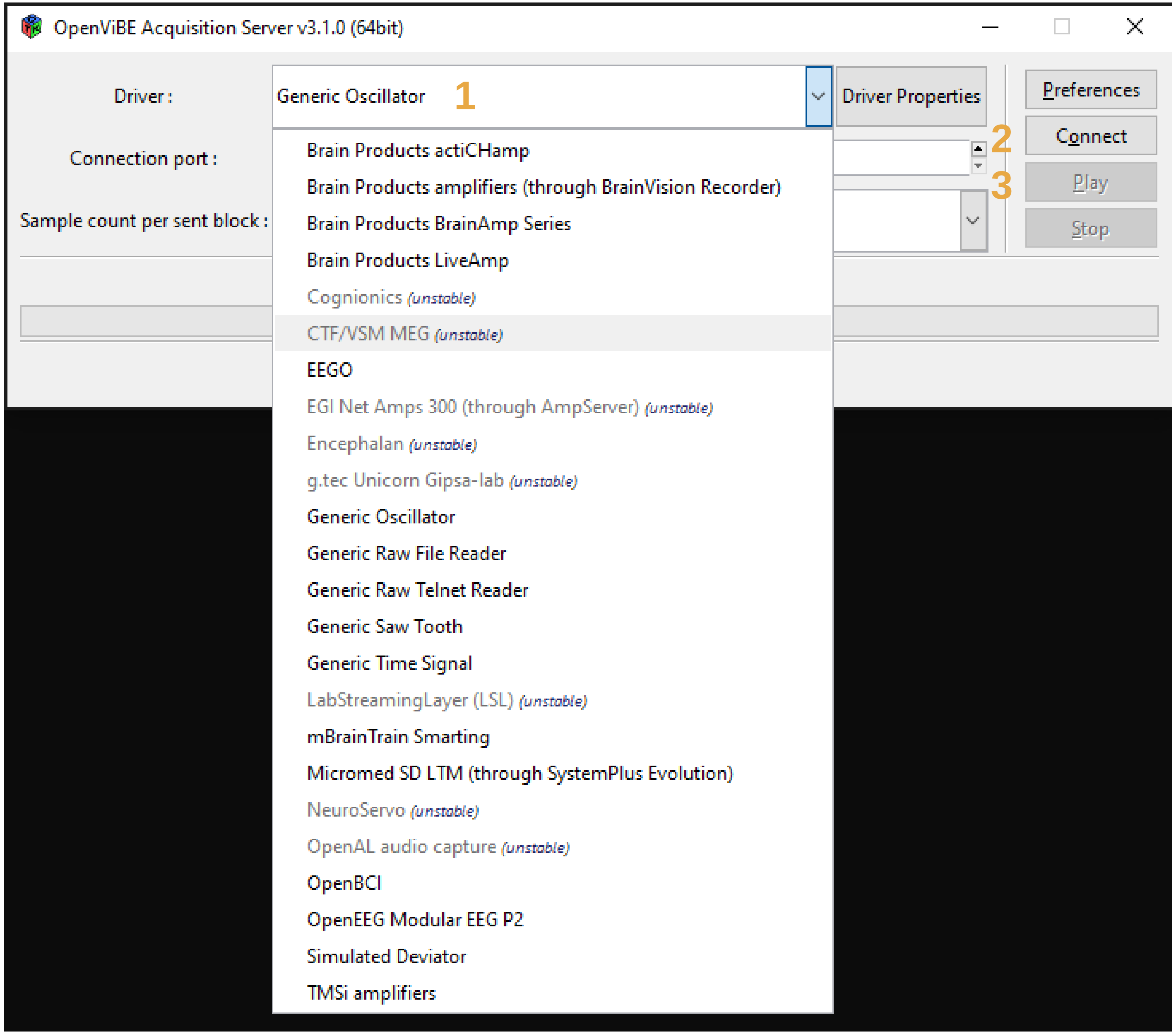
  openvibe-acquisition-server.cmd

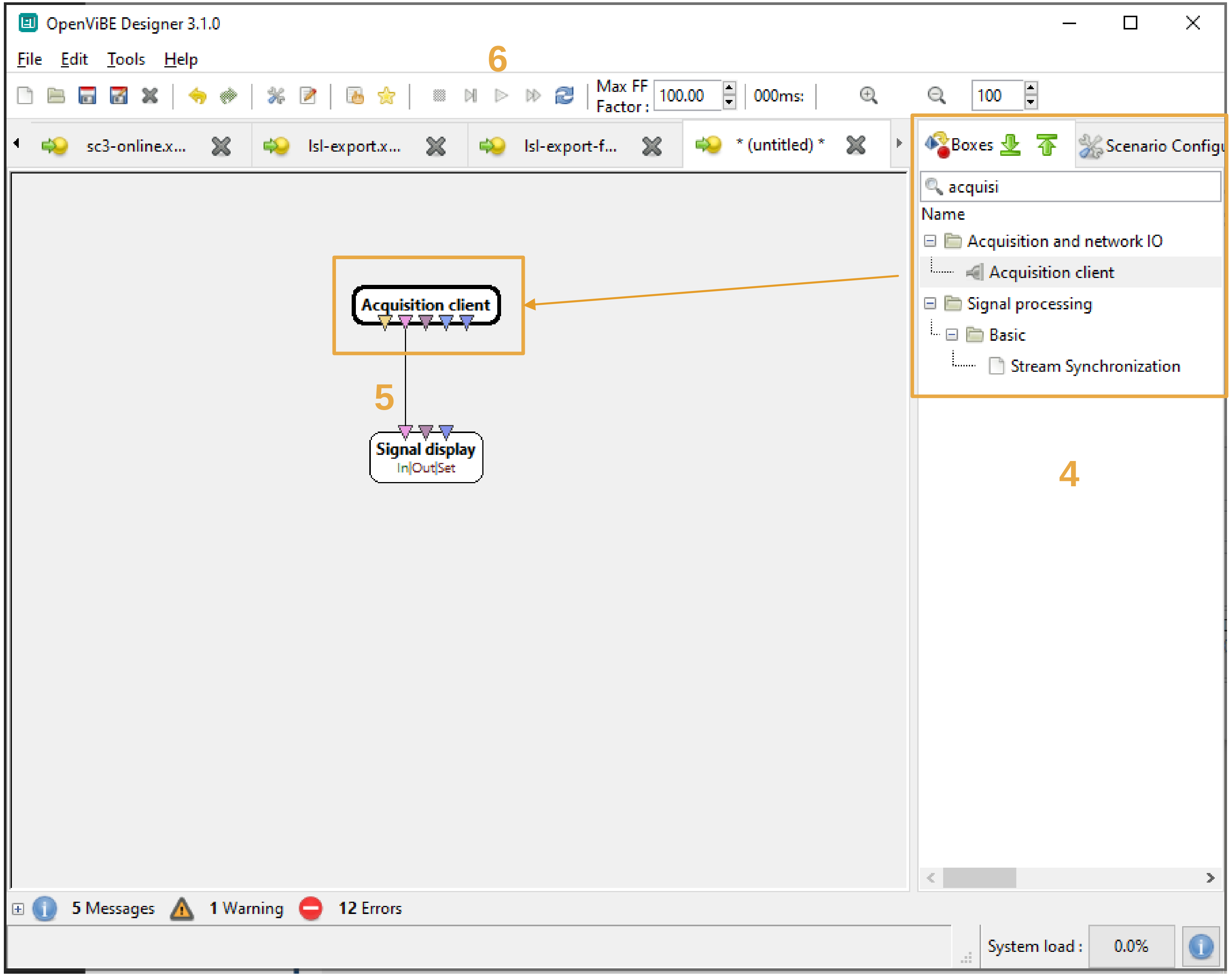


- **1- In the “Driver” list, select “Generic Oscillator”**
  In this list, you’ll find all the drivers for OpenViBE’s supported EEG hardware

- **2- Click “Connect”**

- **3- Click “Play”**





- 4- In the Designer, look for the “Acquisition client” Box and add it to your scenario
- 5- Connect it to the Signal Display Box
- 6- Play the scenario

SC1 - STEP 3 – PROTOCOL MANAGEMENT & STIMULATIONS…

- **Sub-chapters:**
  - Warm-up, first steps: simple I/O
  - Warm-up 2: acquisition server
  - BCI acquisition protocol, Stimulations

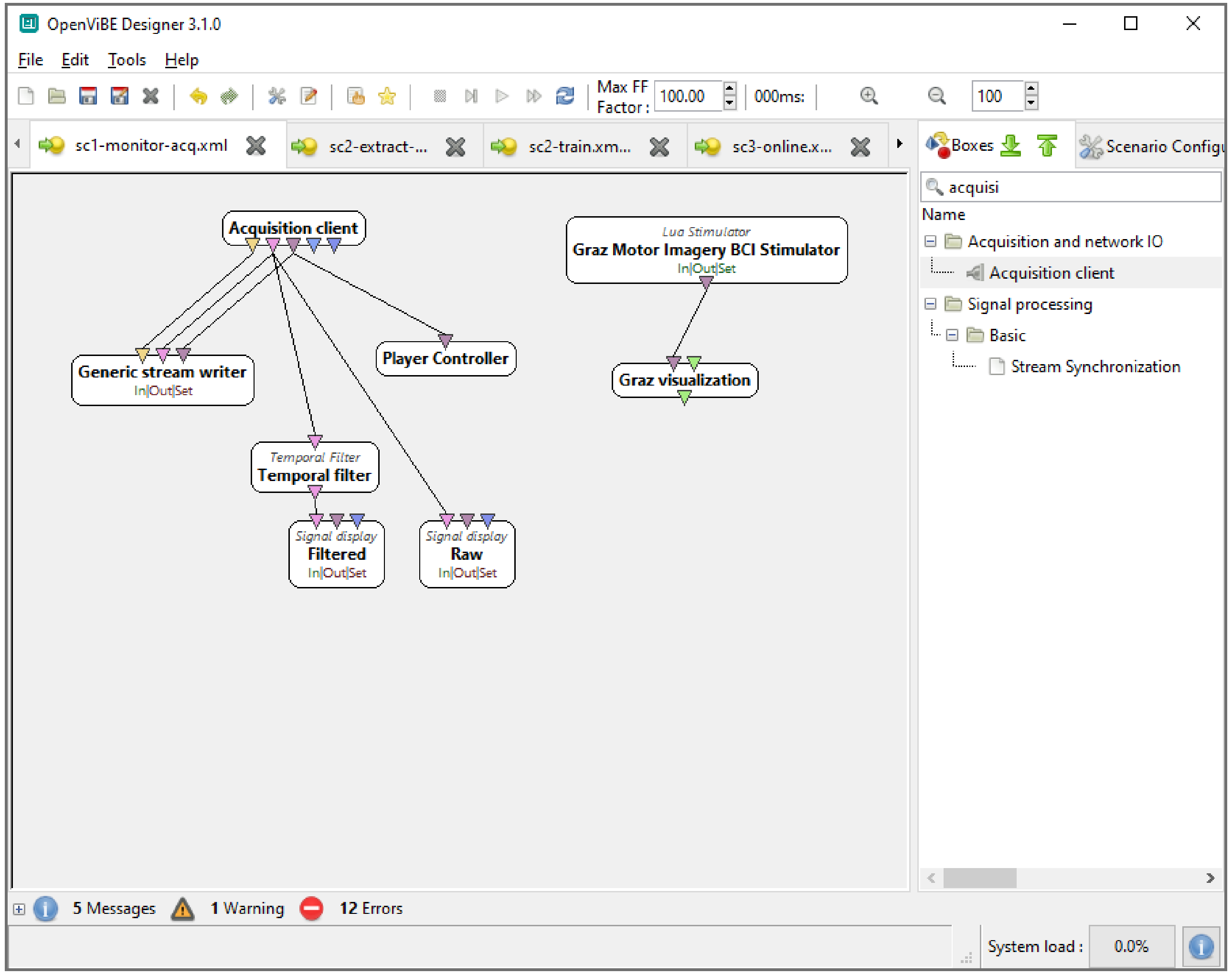
SC1 - STEP 3 – PROTOCOL MANAGEMENT & STIMULATIONS…

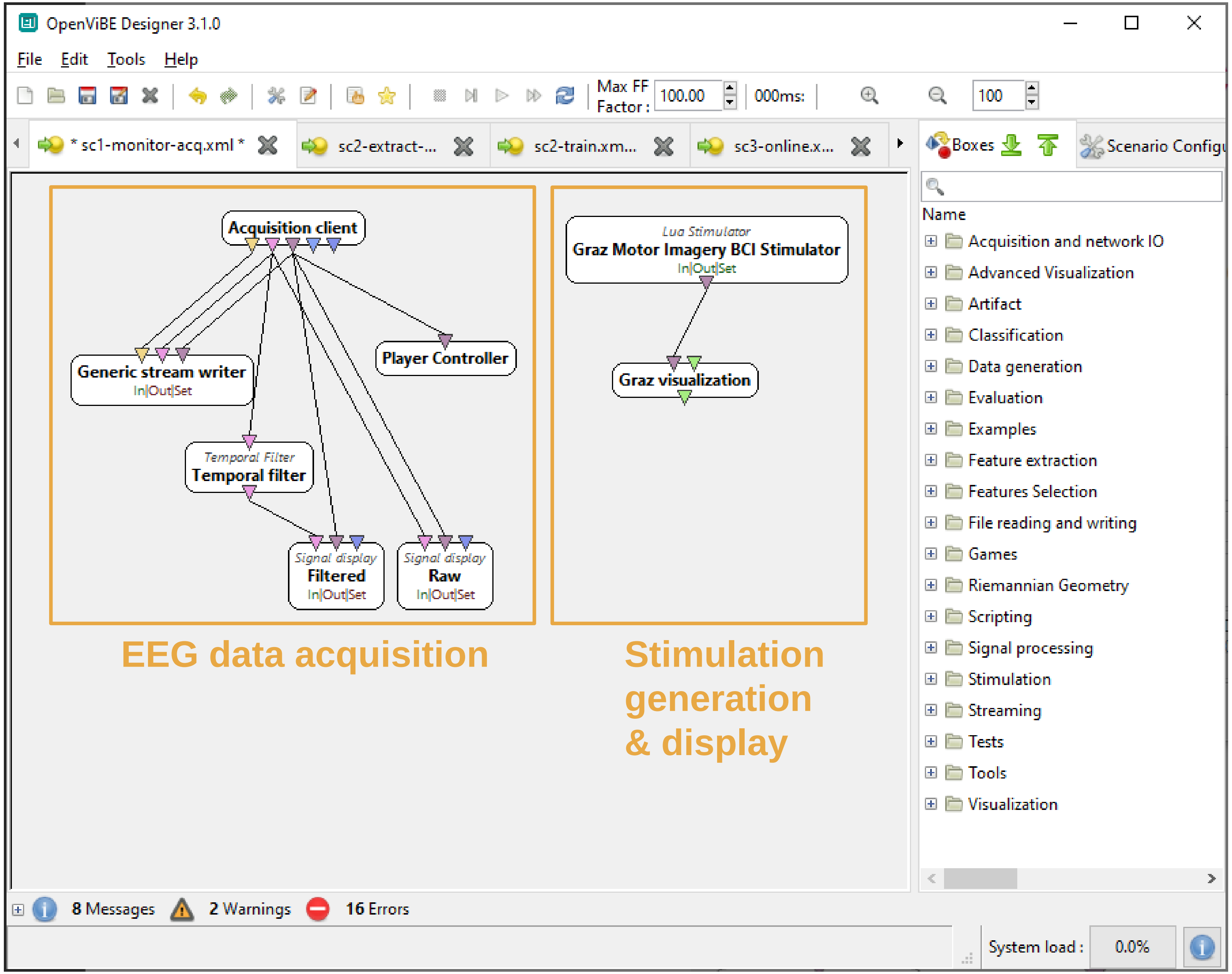
- Goal : understand data acquisition & synchro. with stimulations

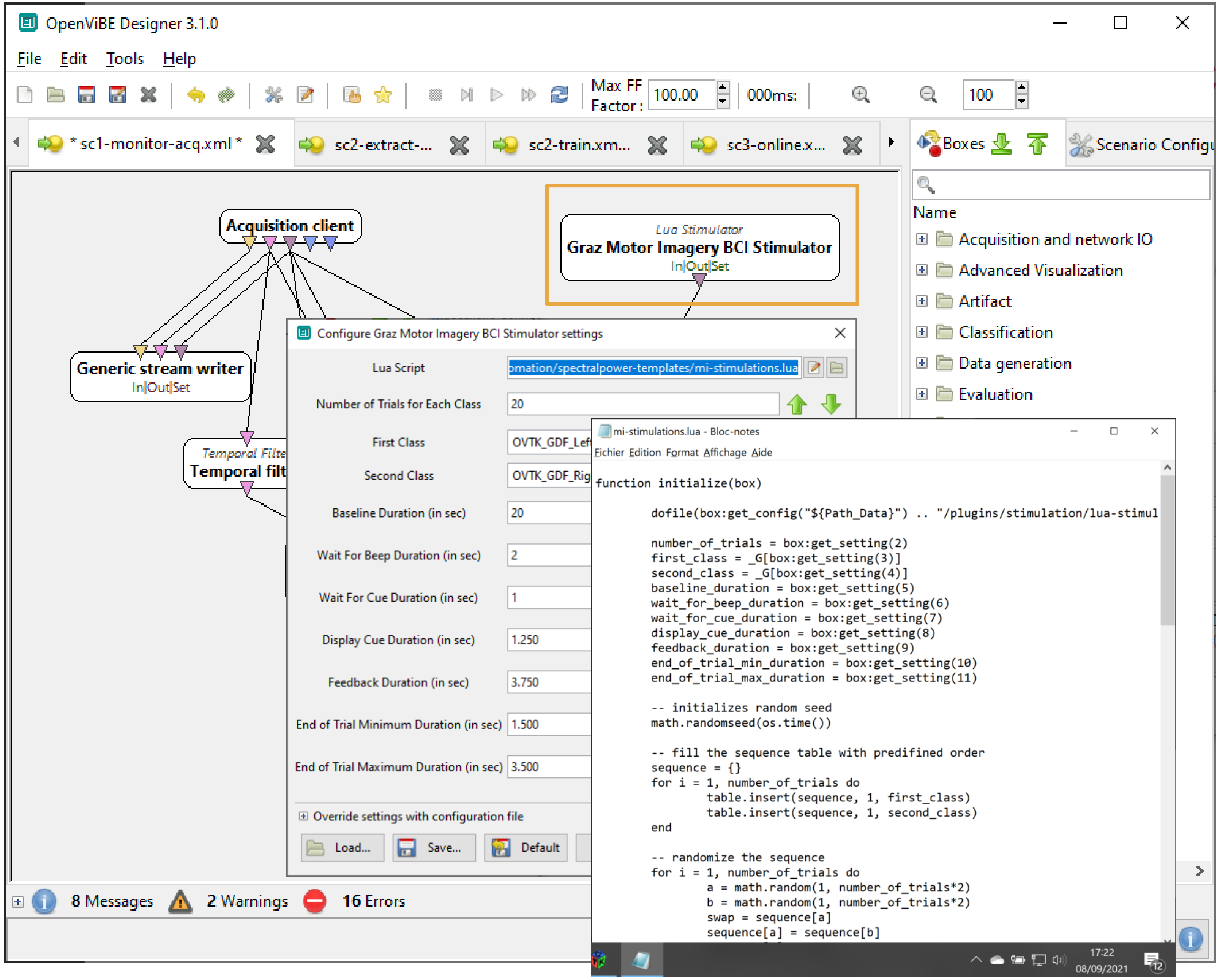
- Load scenario:

  BCI-OpenViBE-CuttingEEG2021/scenarios/sc1-monitor-acq.xml

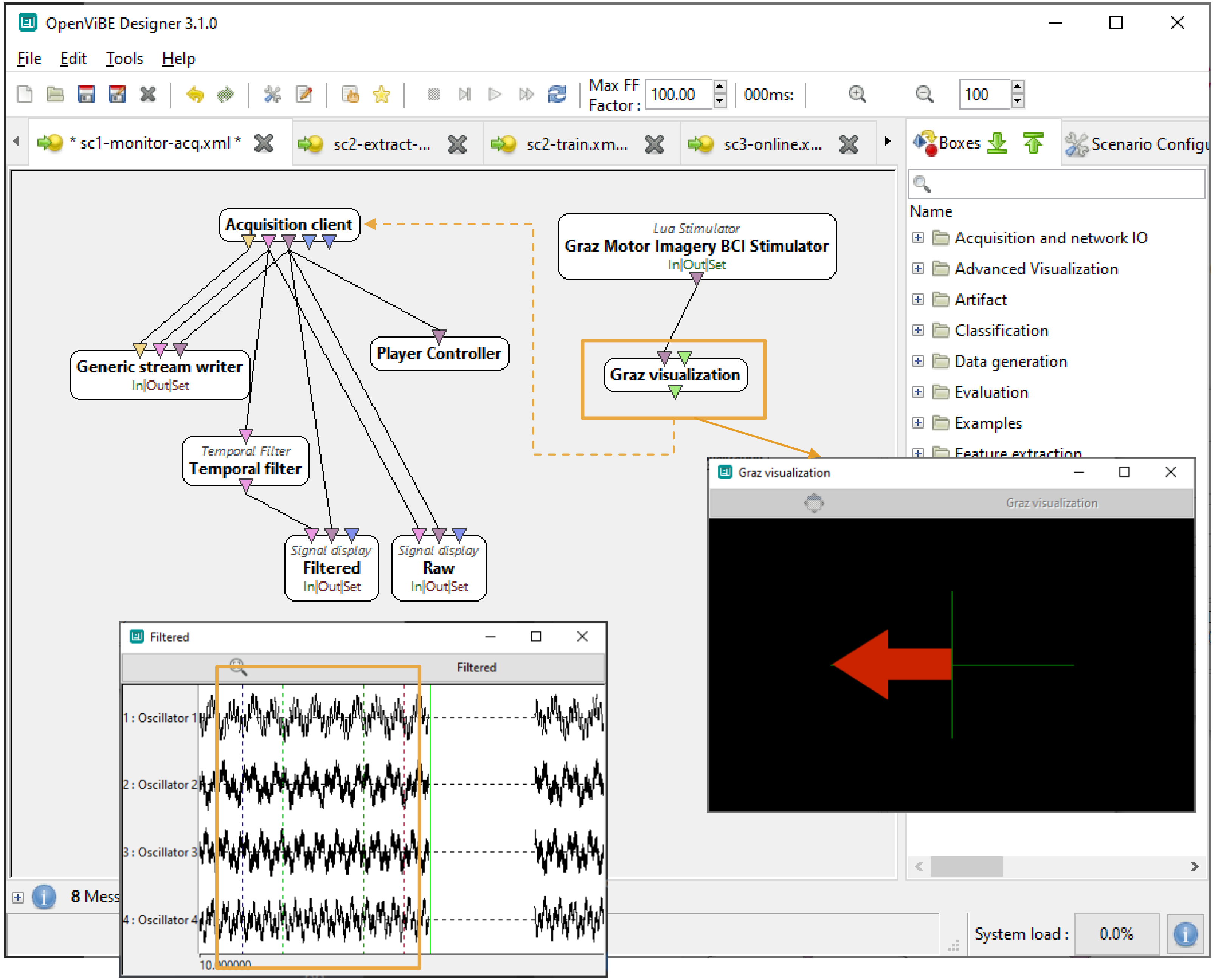
  (github)



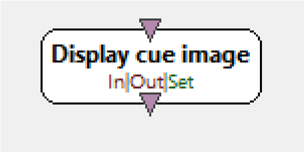

OpenViBE Designer 3.1.0
File Edit Tools Help
* sc1-monitor-acq.xml *
sc2-extract-...
sc2-train.xm...
sc3-online.x...
Boxes
Scenario Configu
Acquisition client
Generic stream writer
Player Controller
Temporal Filter
Temporal filter
Signal display
Filtered
Signal display
Raw
Lua Stimulator
Graz Motor Imagery BCI Stimulator
Graz visualization
EEG data acquisition
Stimulation generation & display
Name
Acquisition and network IO
Advanced Visualization
Artifact
Classification
Data generation
Evaluation
Examples
Feature extraction
Features Selection
File reading and writing
Games
Riemannian Geometry
Scripting
Signal processing
Stimulation
Streaming
Tests
Tools
Visualization
8 Messages
2 Warnings
16 Errors
System load :
0.0%



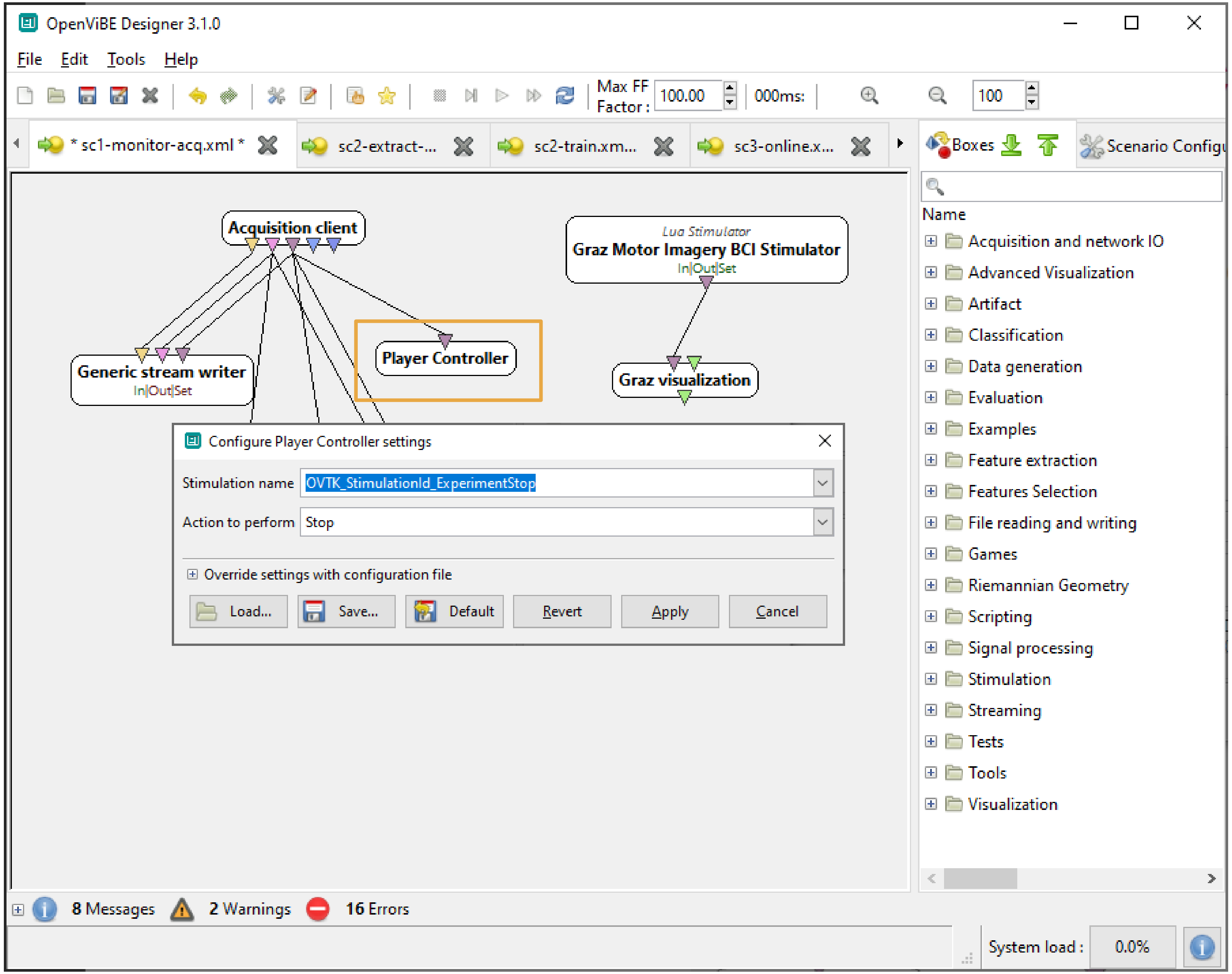
- **LUA Stimulator Box**
  (LUA = scripting language)
- **Experiment example:**
  Different stimulation/event codes at different times.
  Useful for signal segmentation ("epoching")
- **Stimulation label examples**:
  - Experiment Start/Stop
  - Trial Start/Stop
  - LEFT / RIGHT
  - Button pressed
 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



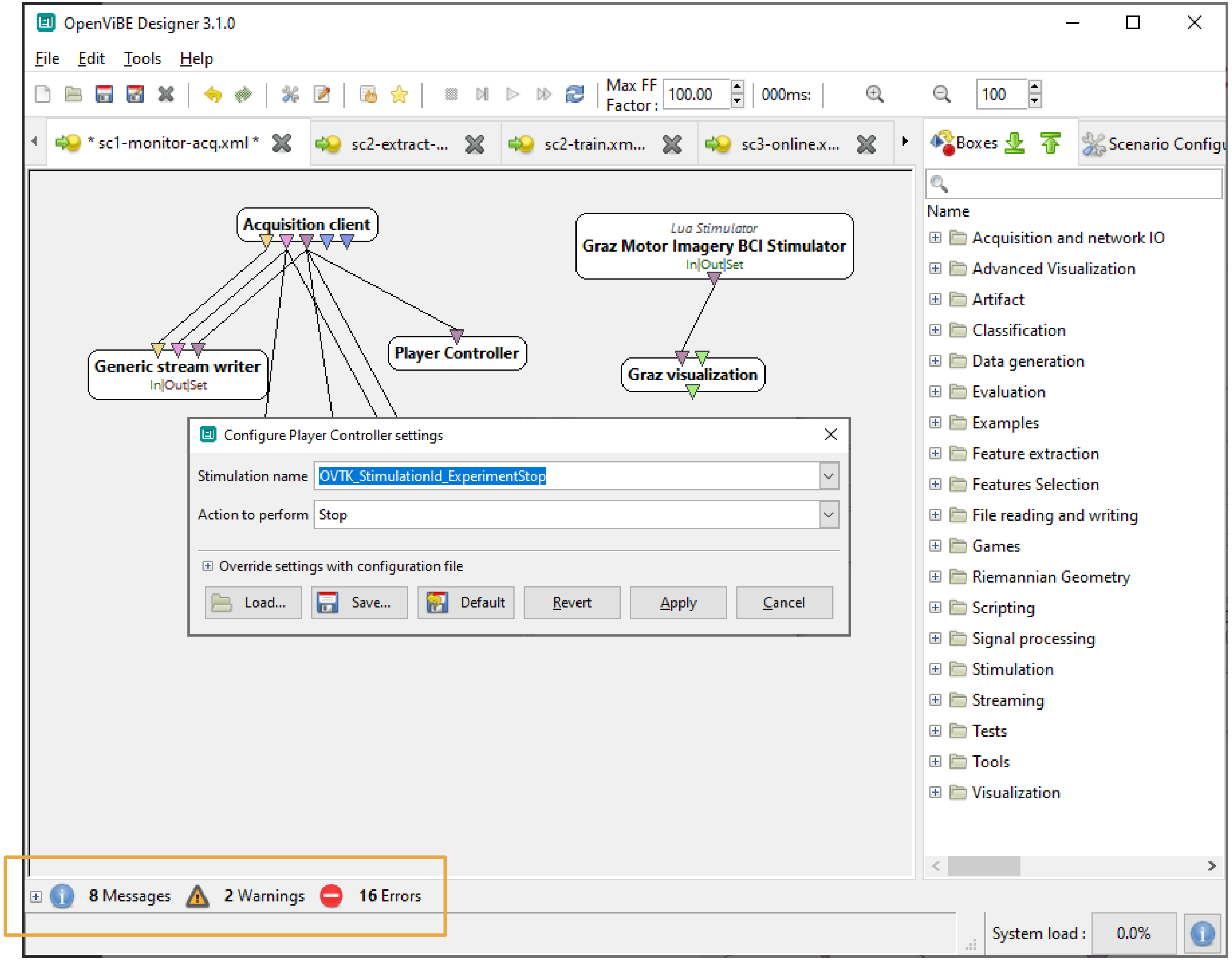
- **"Graz Visualization"** (specific for "Graz Protocol", based on Box "Display Cue Image")



- 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**



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



- Check the embedded log console if anything goes wrong!

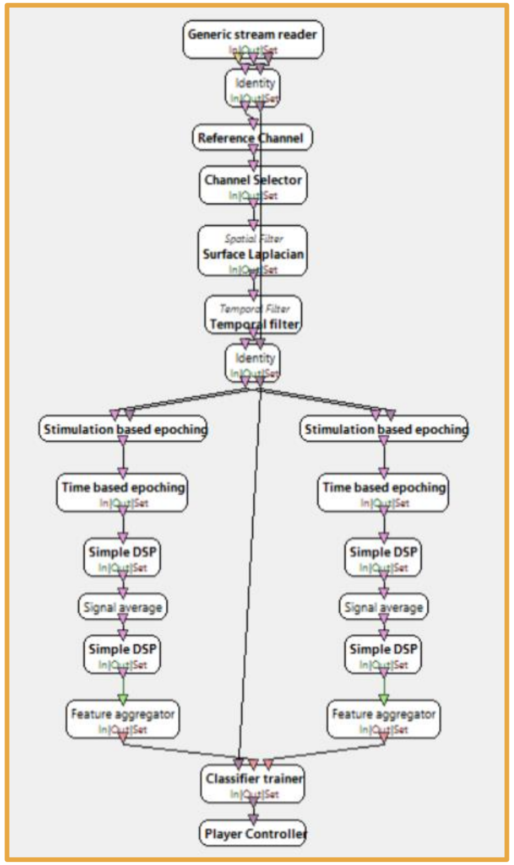
SC1 – FINAL WORDS / Q&A

- **Recap:**
  - Simple I/O operations (signal loading, writing)
  - Signal Display
  - Acquisition Server / Client concept
  - Stimulations
  - Typical BCI protocol acquisition scenario

- **Note: from now on, we will only be using pre-generated/recorded signals**

SC2 – CLASSIFIER TRAINING

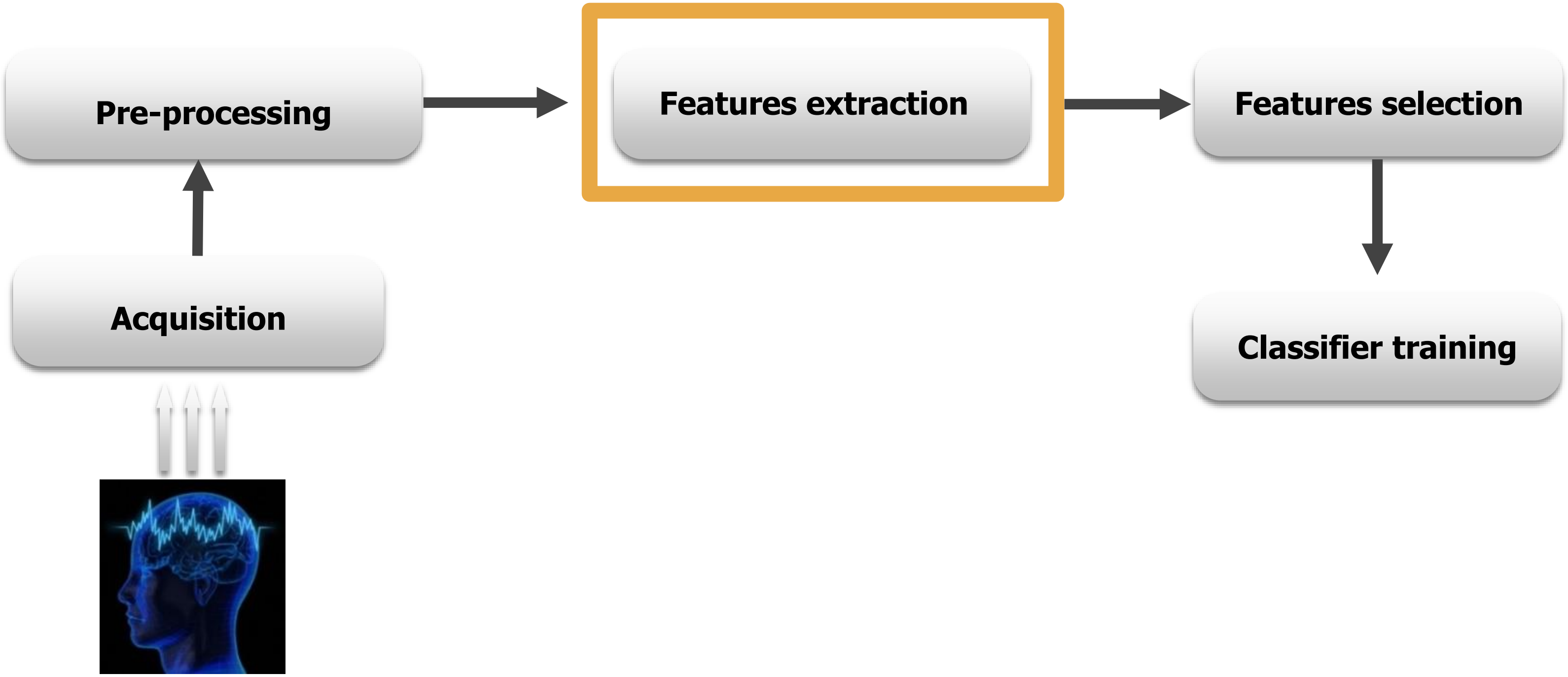
1- Data acquisition



2- Features extraction & Classification

3- Online feedback

SC2 – CLASSIFIER TRAINING

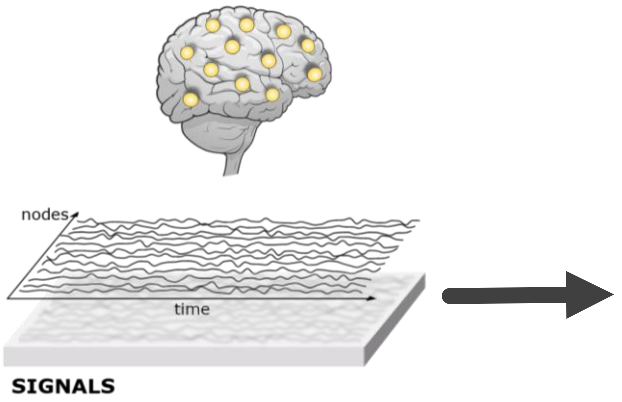


Adapted from (X. Navarro & F. Grosselin)

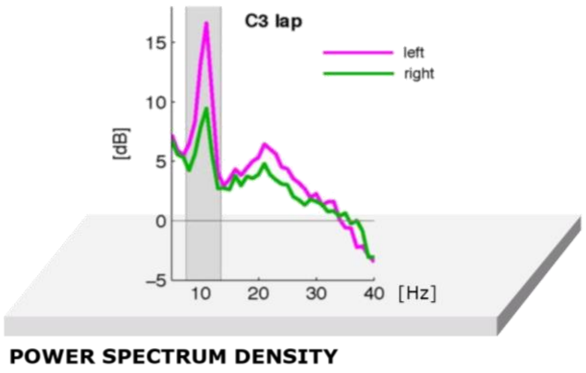
STEP 2 – SIGNAL PROCESSING & VISUALIZATION

Features to extract (recap)

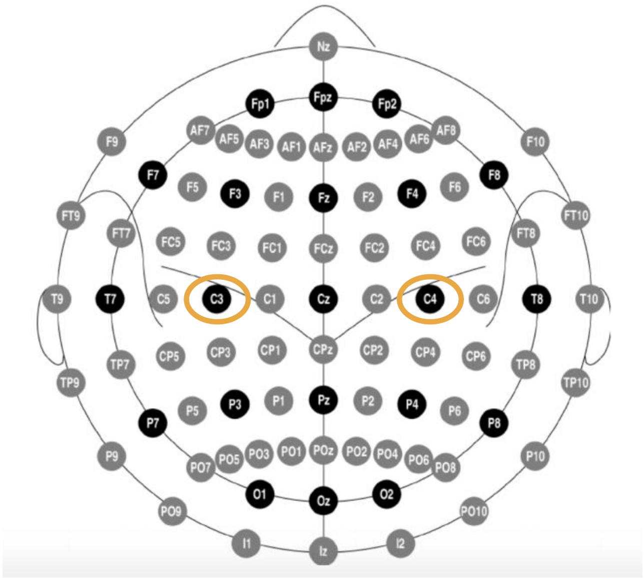
- Power spectra
- Sensorimotor area
- Mu: 8-12Hz &/OR Beta: 14-29Hz



(Gonzalez-Astudillo et al, 2020)



(Maeder et al., 2012)



SC2 – CLASSIFIER TRAINING

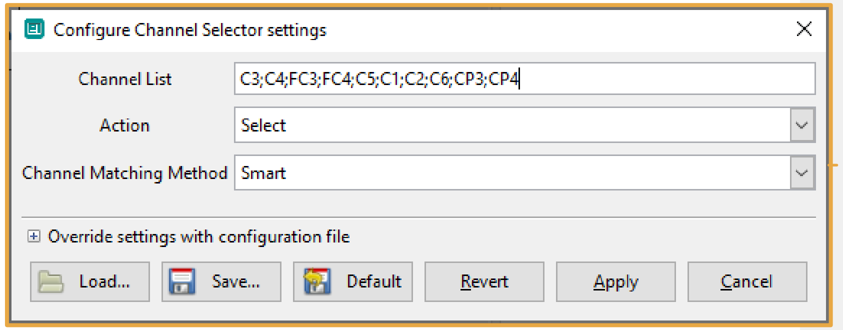
- **Sub-chapters:**
  - Channel Selection
  - Basic Signal Processing / Spectral analysis
  - Topography visualization
  - Feature Selection + Spatial Filtering
  - Classifier training

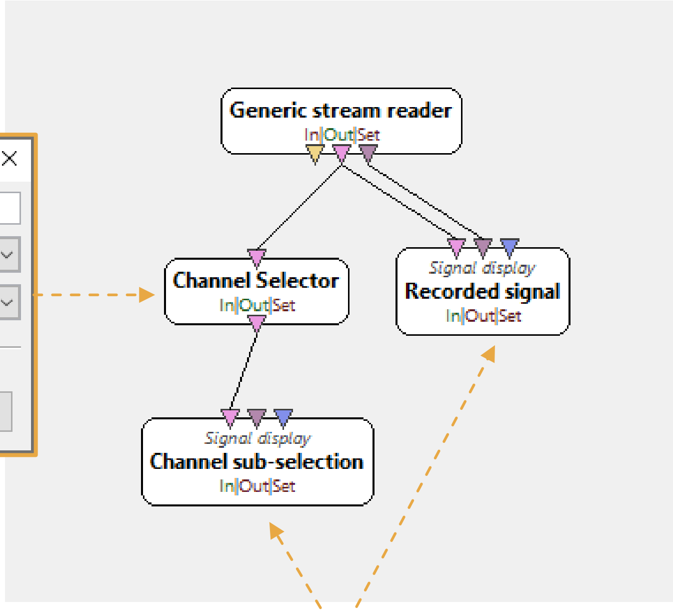
SC2 – STEP 1 - CHANNEL SELECTION

- **Sub-chapters:**
  - Channel Selection
  - Basic Signal Processing / Spectral analysis
  - Topography visualization
  - Feature Selection + Spatial Filtering
  - Classifier training

SC2 – STEP 1 - CHANNEL SELECTION

- **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.





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

Display windows will use « renamed » label

- 1- Import a "Generic stream reader" box, and set the filename to:

  *<openvibe-3.1.0-64bit>\share\openvibe\scenarios\signals***bci-motor-imagery.ov**

- (optional – have a look at the recorded signals with "**Signal Display**")

- 2- Import a "**Channel Selector**" & link the boxes
- 3- Set the selection to :

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

- 4- Display the output

SC2 – STEP 2 – SIGNAL PROCESSING & SPECTRAL ANALYSIS

- **Sub-chapters:**
  - Channel Selection
  - Basic Signal Processing / Spectral analysis
  - Topography visualization
  - Feature Selection + Spatial Filtering
  - Classifier training

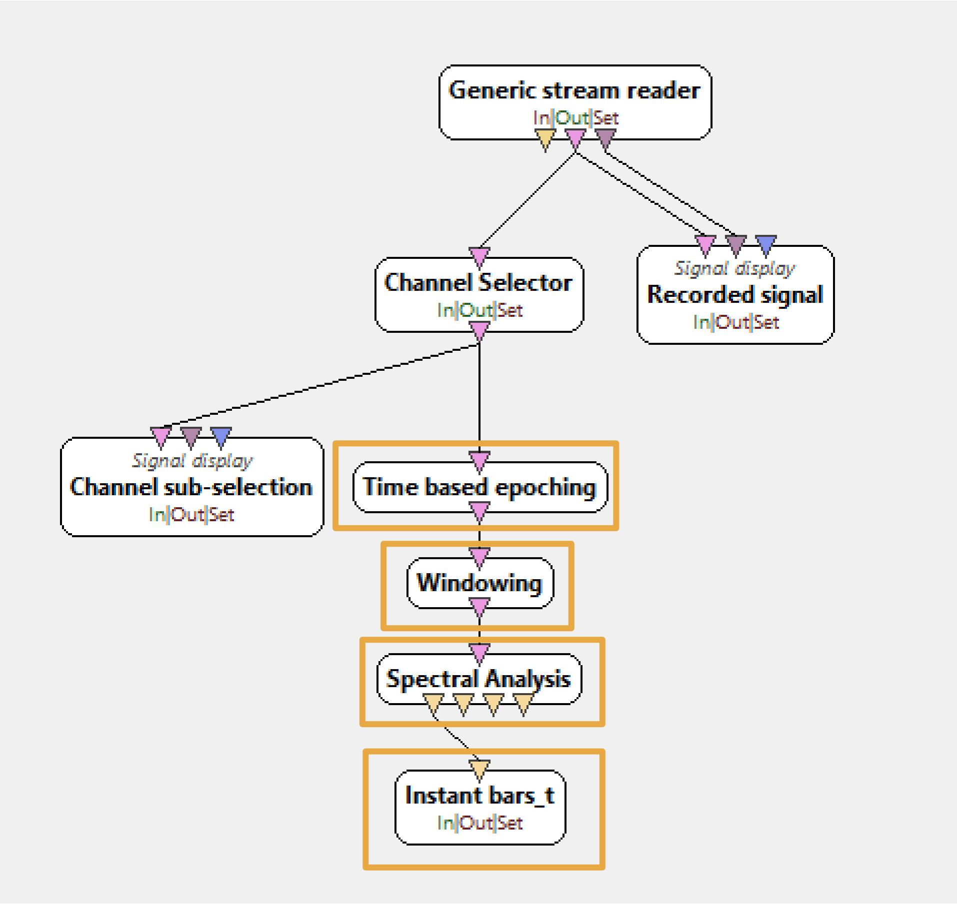
SC2 – STEP 2 – SIGNAL PROCESSING & SPECTRAL ANALYSIS

- **Goals:**

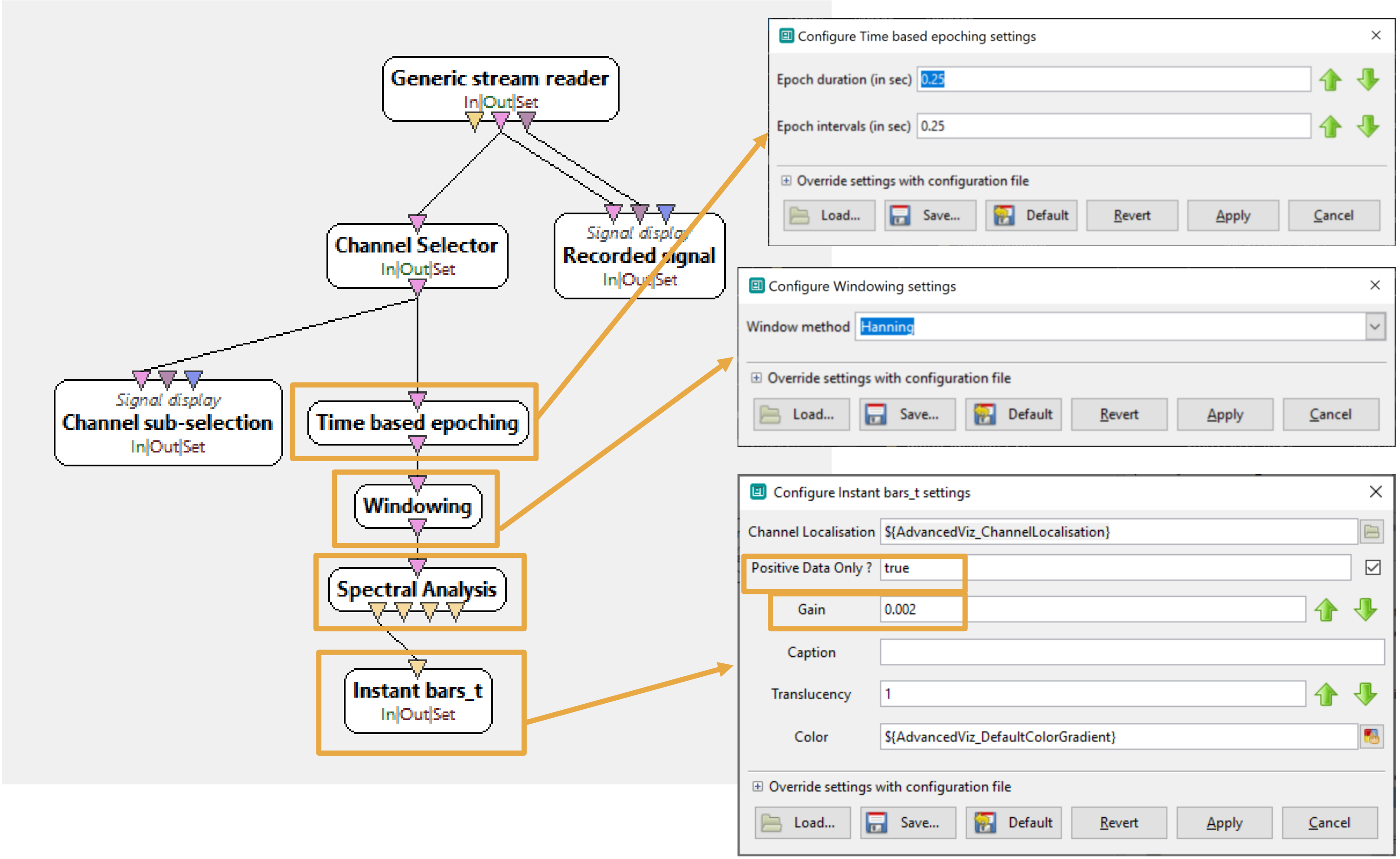
  Discover & use filtering, epoching, windowing tools

  Compute & display spectra for each channel/electrodes:

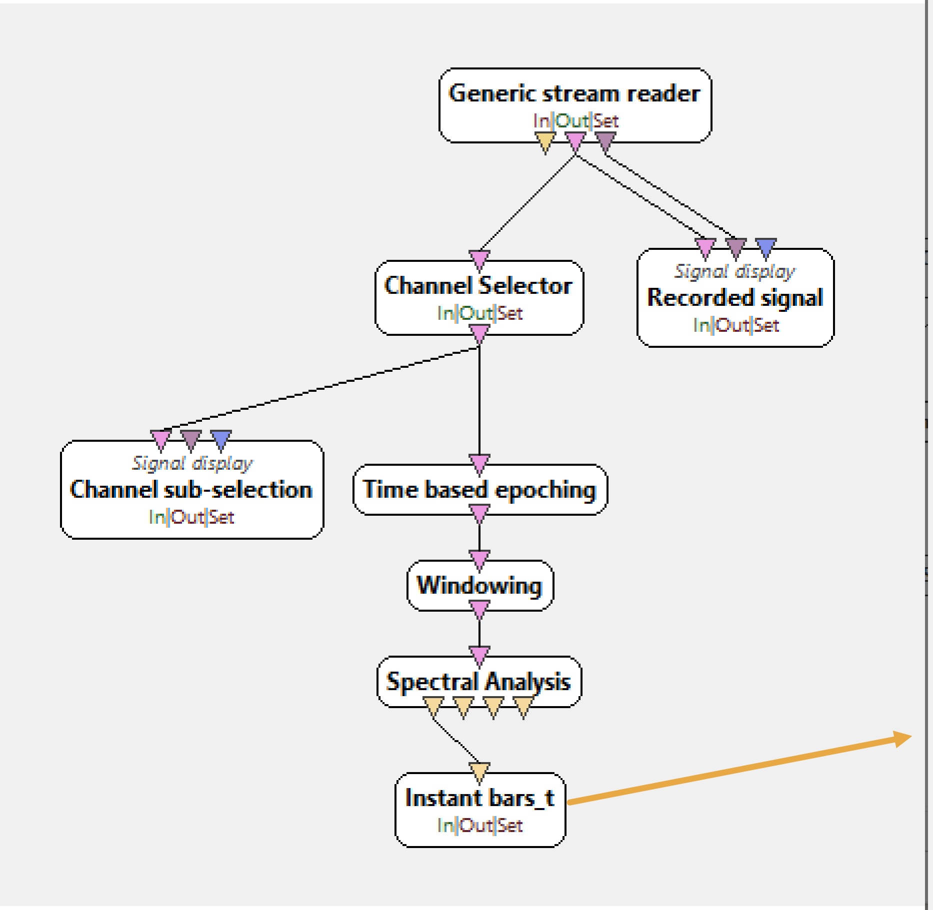
  - for the whole band
  - for different frequency bands (alpha, beta, etc)

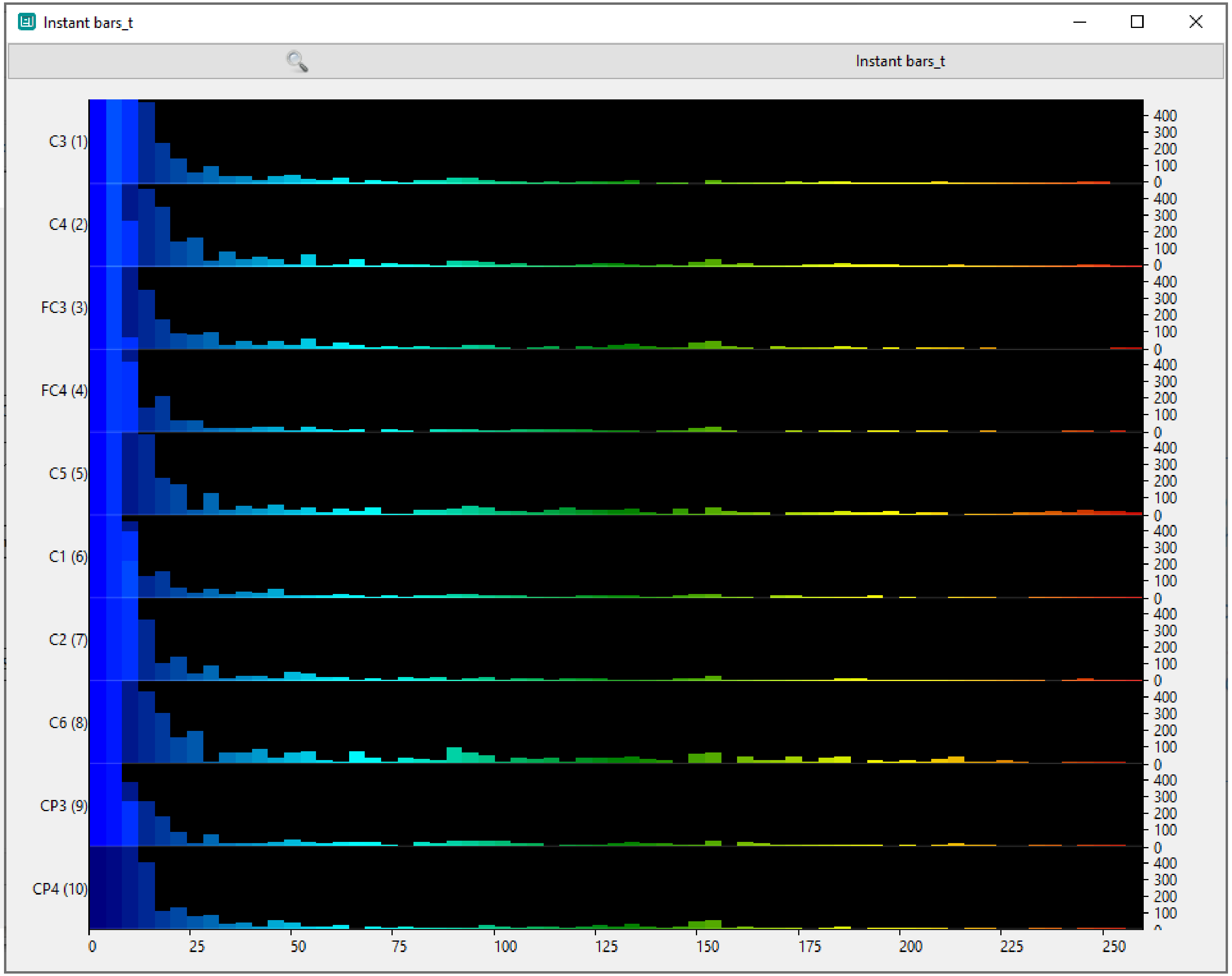


- 1- Import:
  "Time Based Epoching",
  "Windowing",
  "Spectral Analysis",
  "Instant bars"

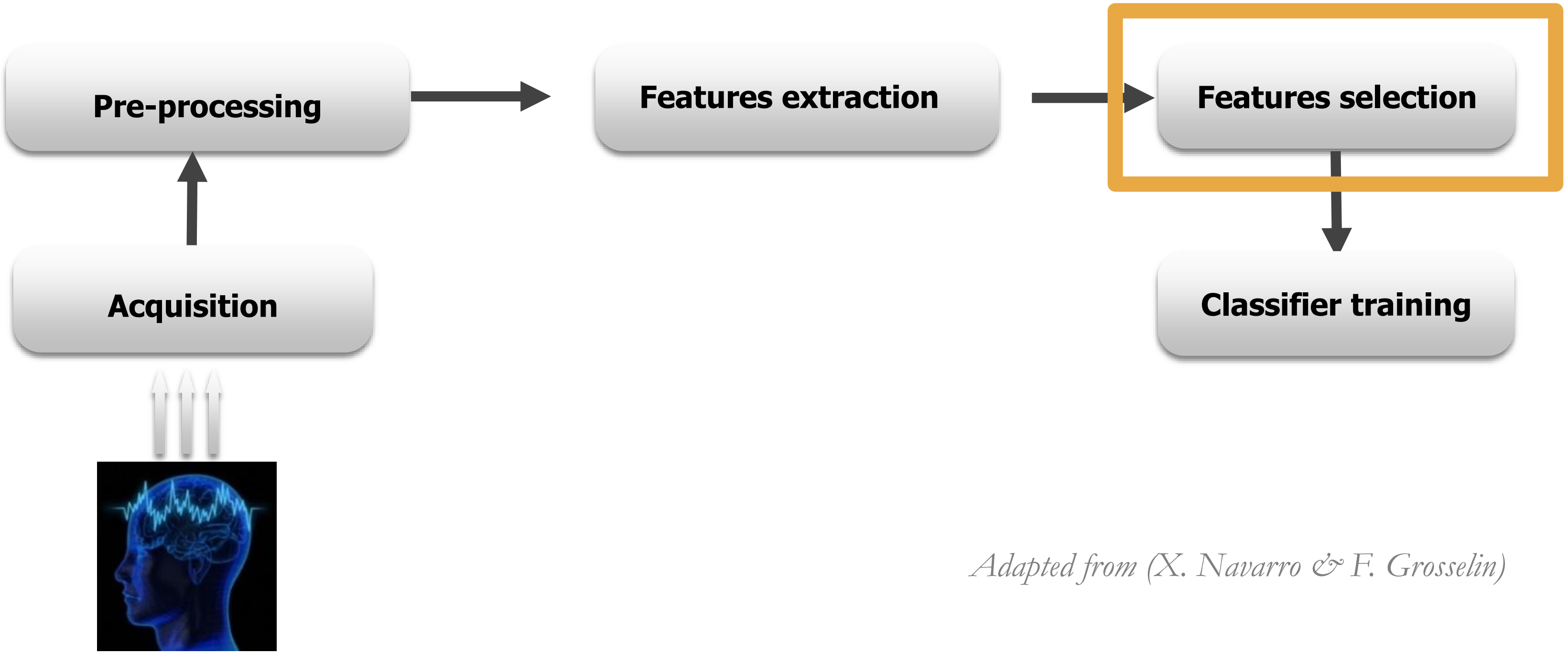


- 1- Import: **"Time Based Epoching", "Windowing", "Spectral Analysis", "Instant bars"**
- 2- Set preferred settings for Epoching and windowing (ex: 0,25s, no overlap, Hann Window)
- Note : Spectral Analysis has 4 outputs (see parameters for details).
- 3- Set gain for "instant bars" ! No automatic scaling…


Generic stream reader
In|Out|Set
Channel Selector
In|Out|Set
Signal display
Recorded signal
In|Out|Set
Signal display
Channel sub-selection
In|Out|Set
Time based epoching
Windowing
Spectral Analysis
Instant bars_t
In|Out|Set

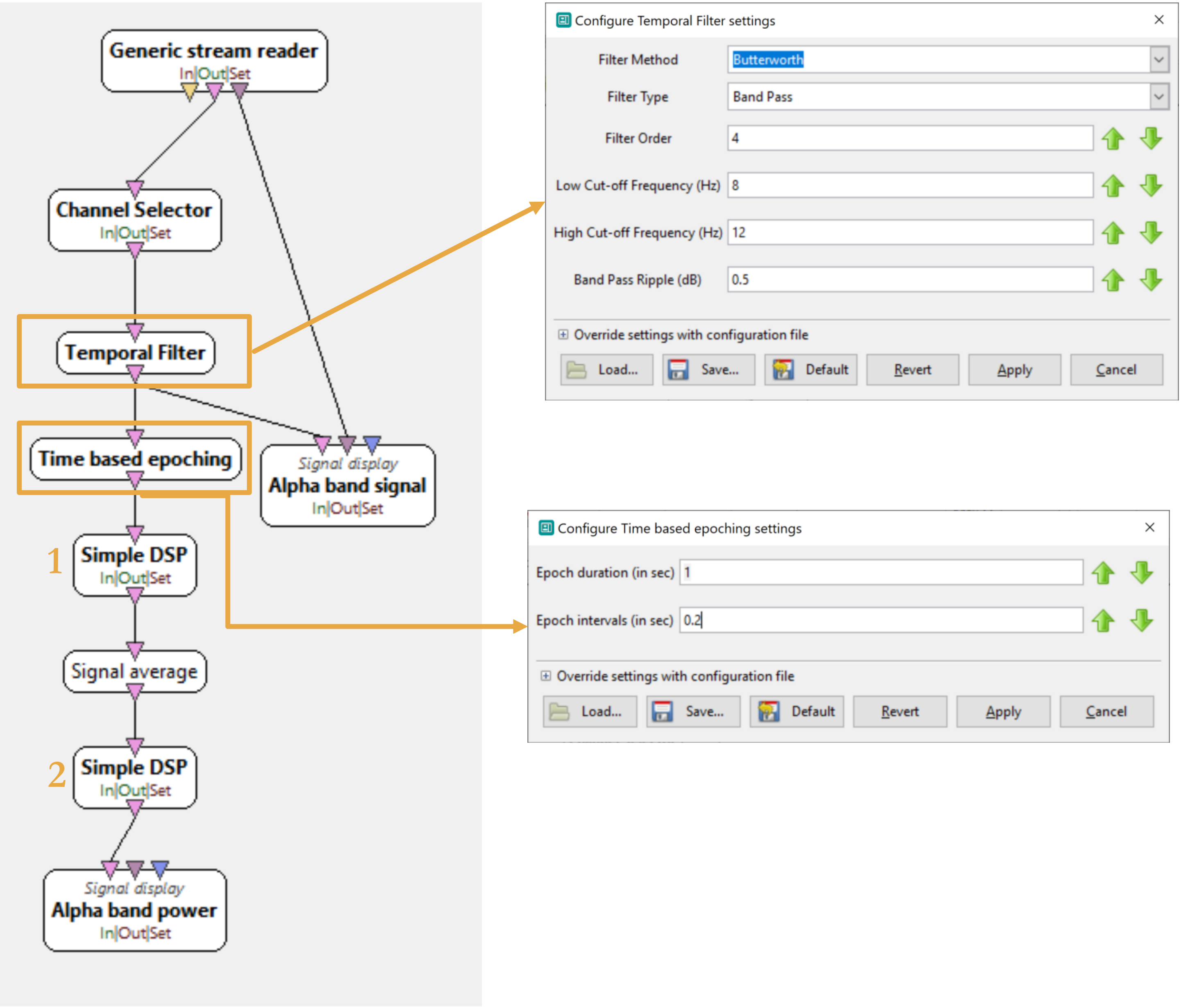

Instant bars_t
Instant bars_t
C3 (1)
C4 (2)
FC3 (3)
FC4 (4)
C5 (5)
C1 (6)
C2 (7)
C6 (8)
CP3 (9)
CP4 (10)
400
300
200
100
0
0
25
50
75
100
125
150
175
200
225
250

- We now have access to a view of the full band spectrum
- However, the feature we need for training the classifier is the **spectral power in the alpha/beta band**

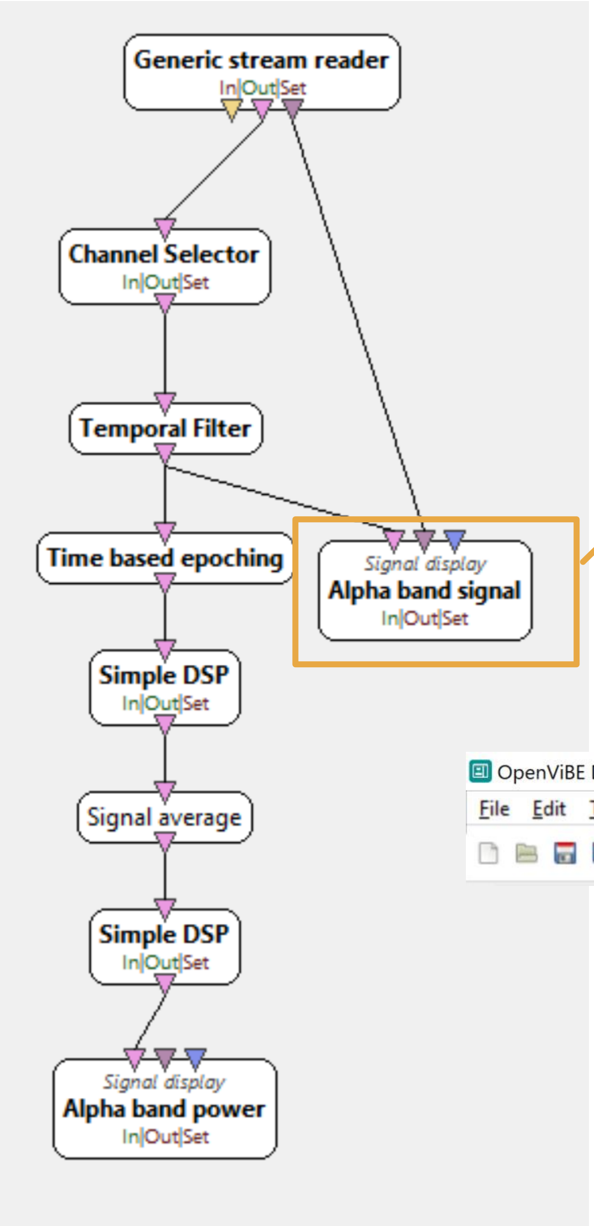


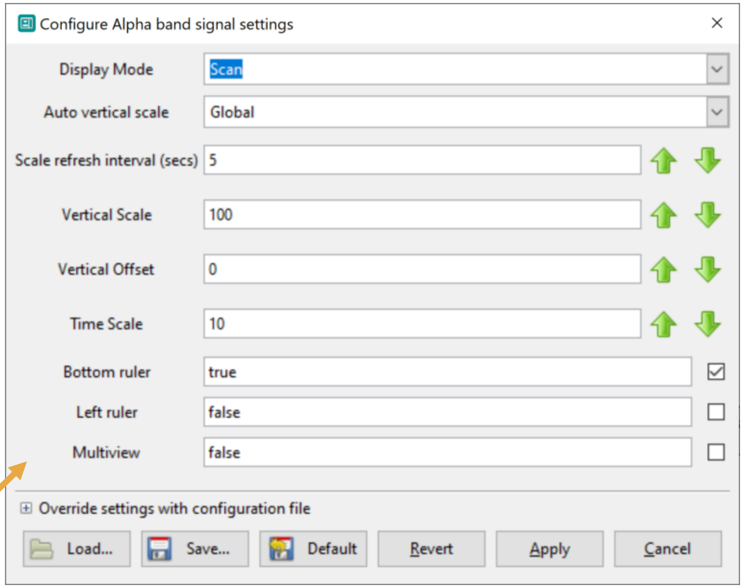
Manually or automatically

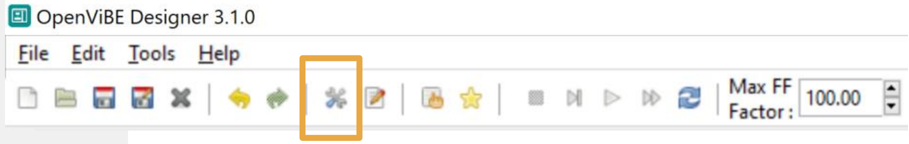
Adapted from (X. Navarro & F. Grosselin)



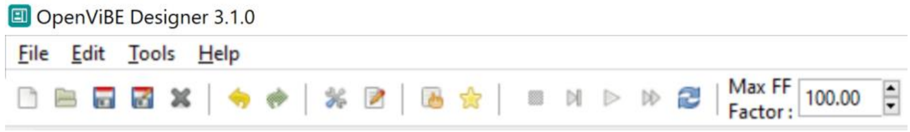
- “**Temporal Filter**”, set to [8;12] Hz band
- Signal processing chain:
  - “**Time based epoching**” of 1s, every 0.2s (overlapping windows of signal for power computation)
  - **DSP 1** formula: x*x
  - **Average** (= of x^2 across a window of 1s)
  - **DSP 2** formula: Log10(1+x)
  - => $\log10(1+\mathrm{avg}(x^2))$
- Add **Displays** and rename them
  - Set time scales & vertical scaling

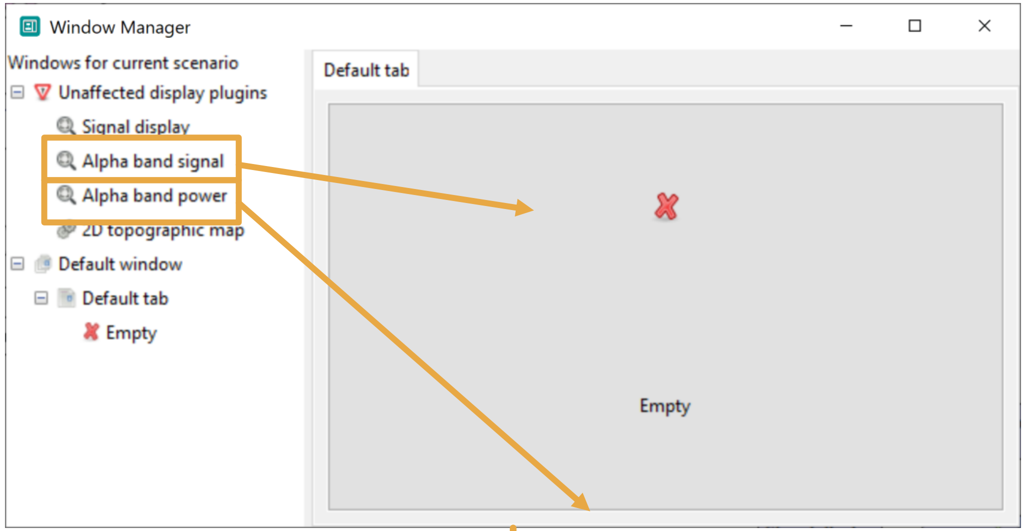


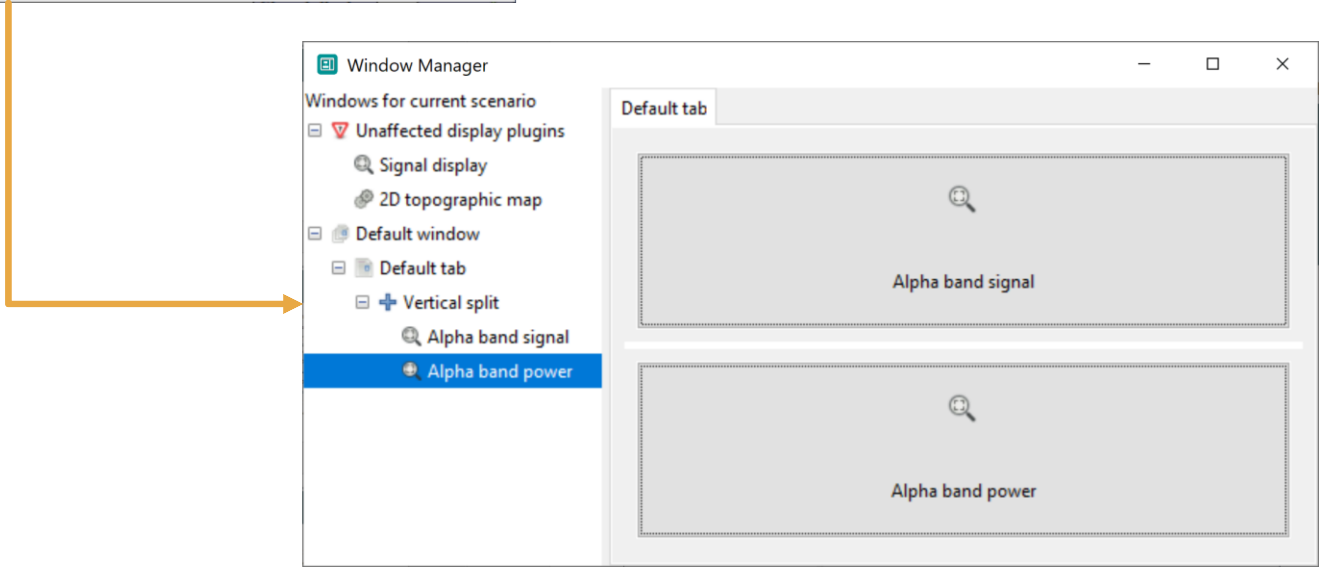




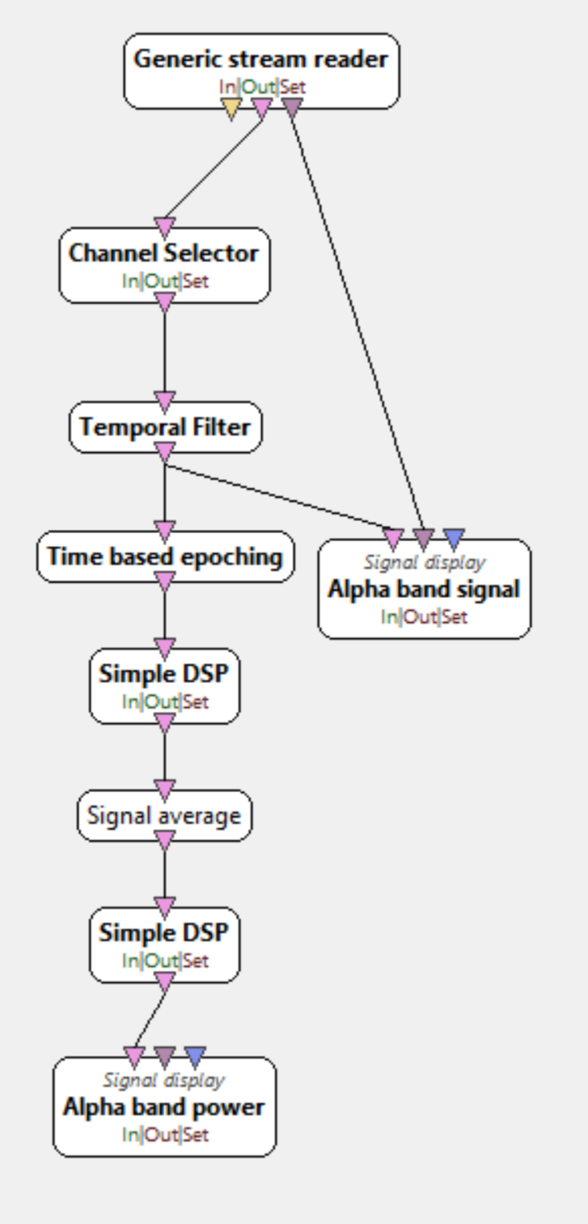
- Add **Displays** and rename them
  - Set time scales : 10 (seconds) for alpha band signal display, 50 for alpha band power (since we have 5 times more data to display)
  - Set vertical scaling to "global"
- Use widget reordering tool for ease of use!

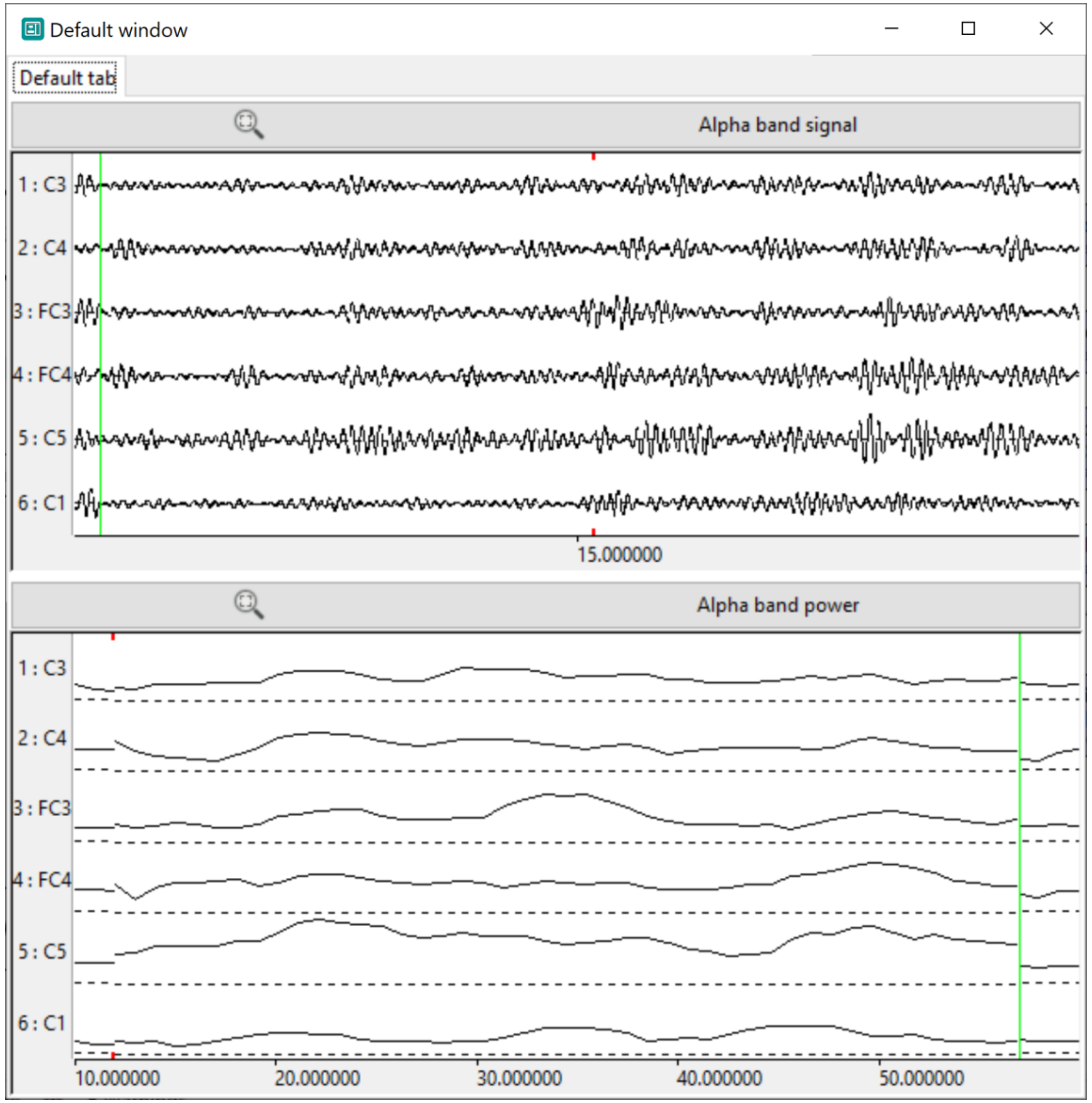




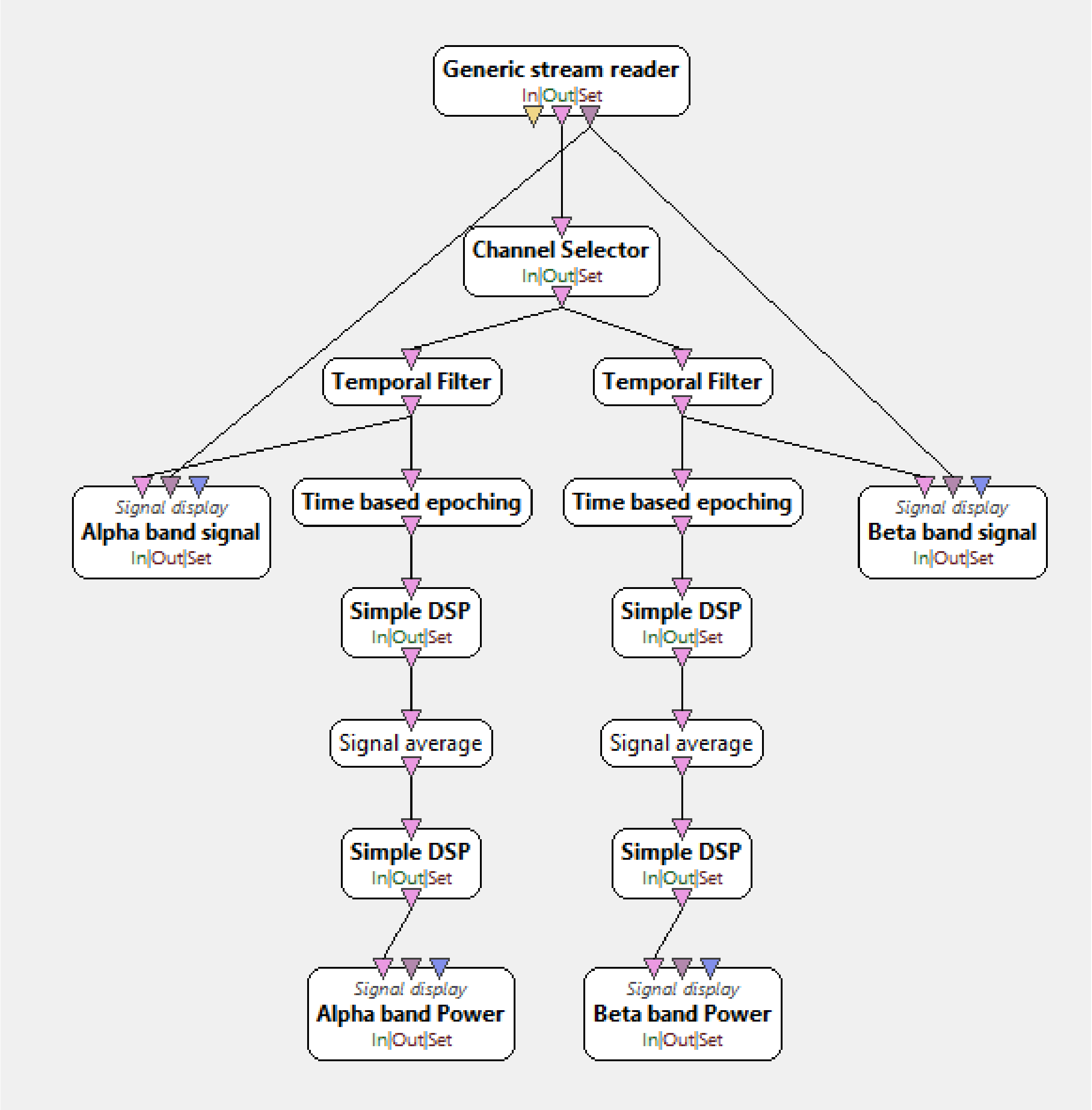


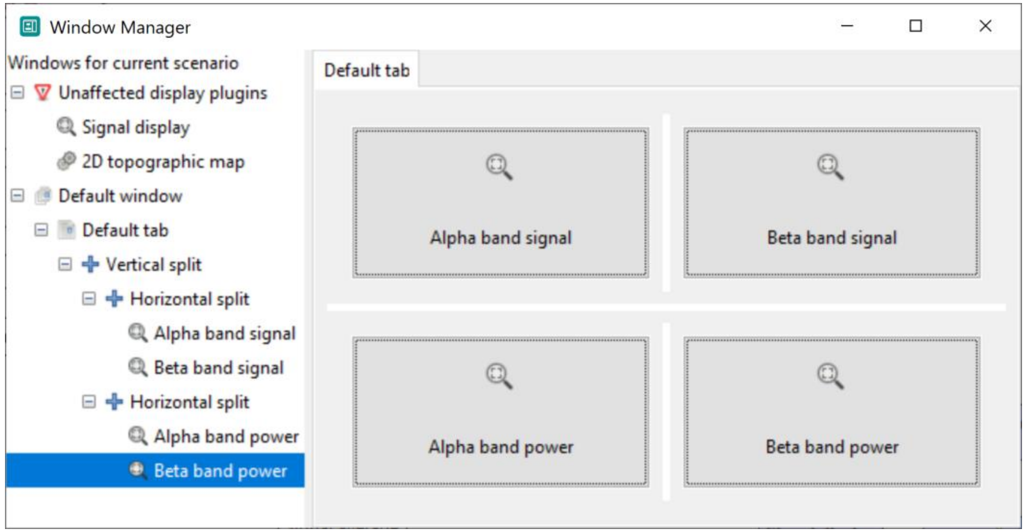
- Use widget reordering tool for ease of use!

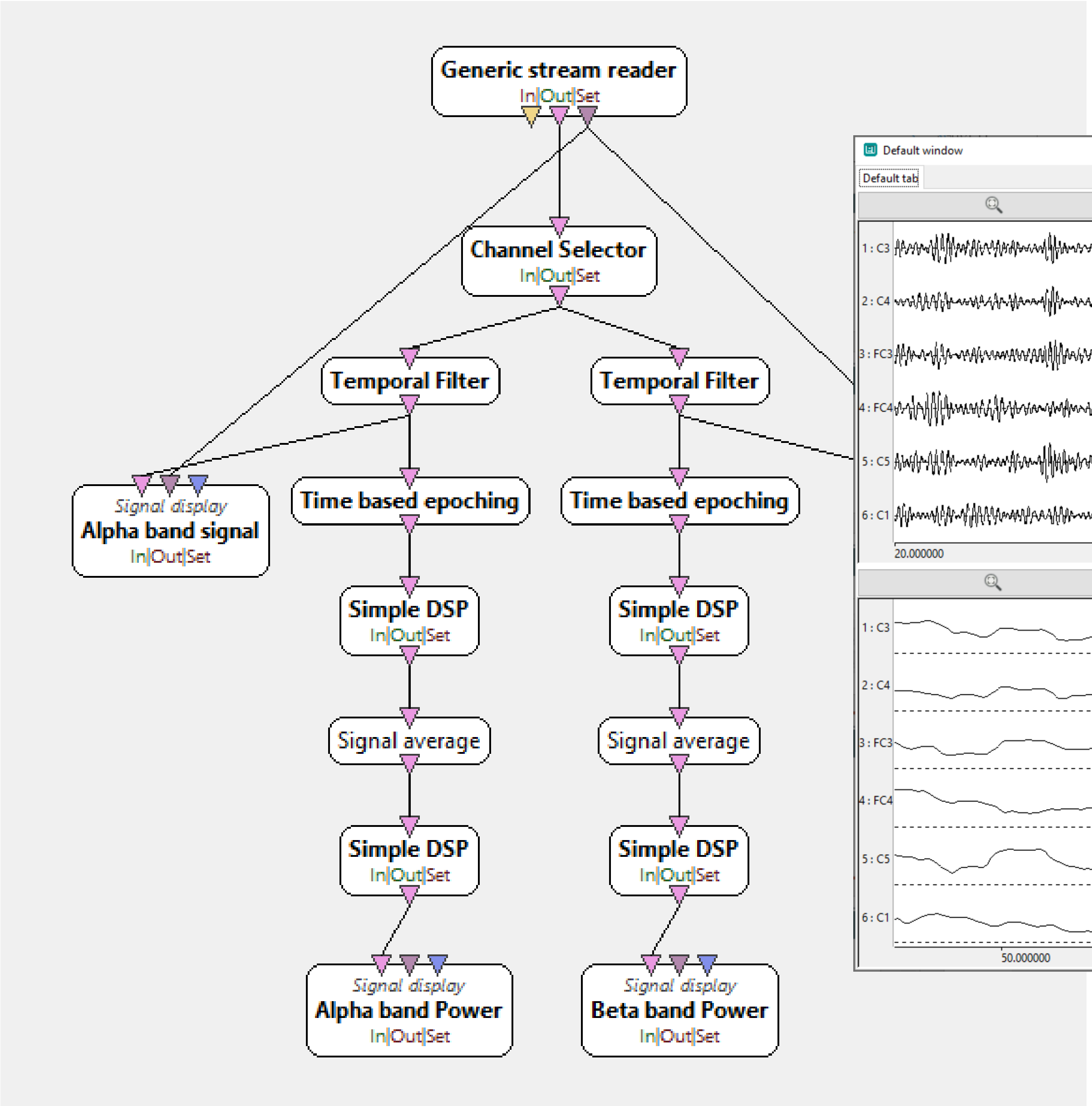

Generic stream reader
In|Out|Set
Channel Selector
In|Out|Set
Temporal Filter
Time based epoching
Signal display
Alpha band signal
In|Out|Set
Simple DSP
In|Out|Set
Signal average
Simple DSP
In|Out|Set
Signal display
Alpha band power
In|Out|Set

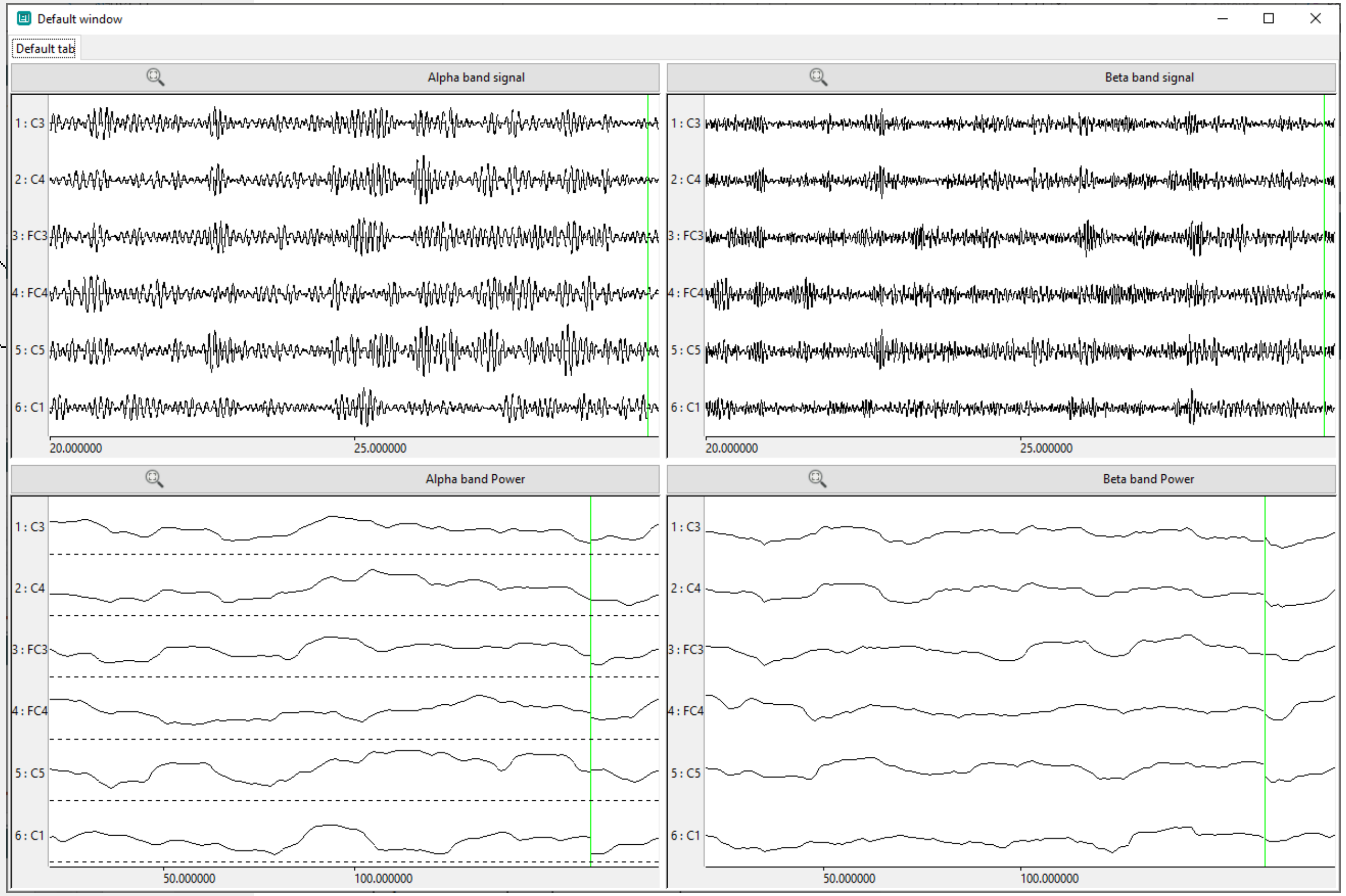

Default window
Default tab
Alpha band signal
1 : C3
2 : C4
3 : FC3
4 : FC4
5 : C5
6 : C1
15.000000
Alpha band power
1 : C3
2 : C4
3 : FC3
4 : FC4
5 : C5
6 : C1
10.000000
20.000000
30.000000
40.000000
50.000000

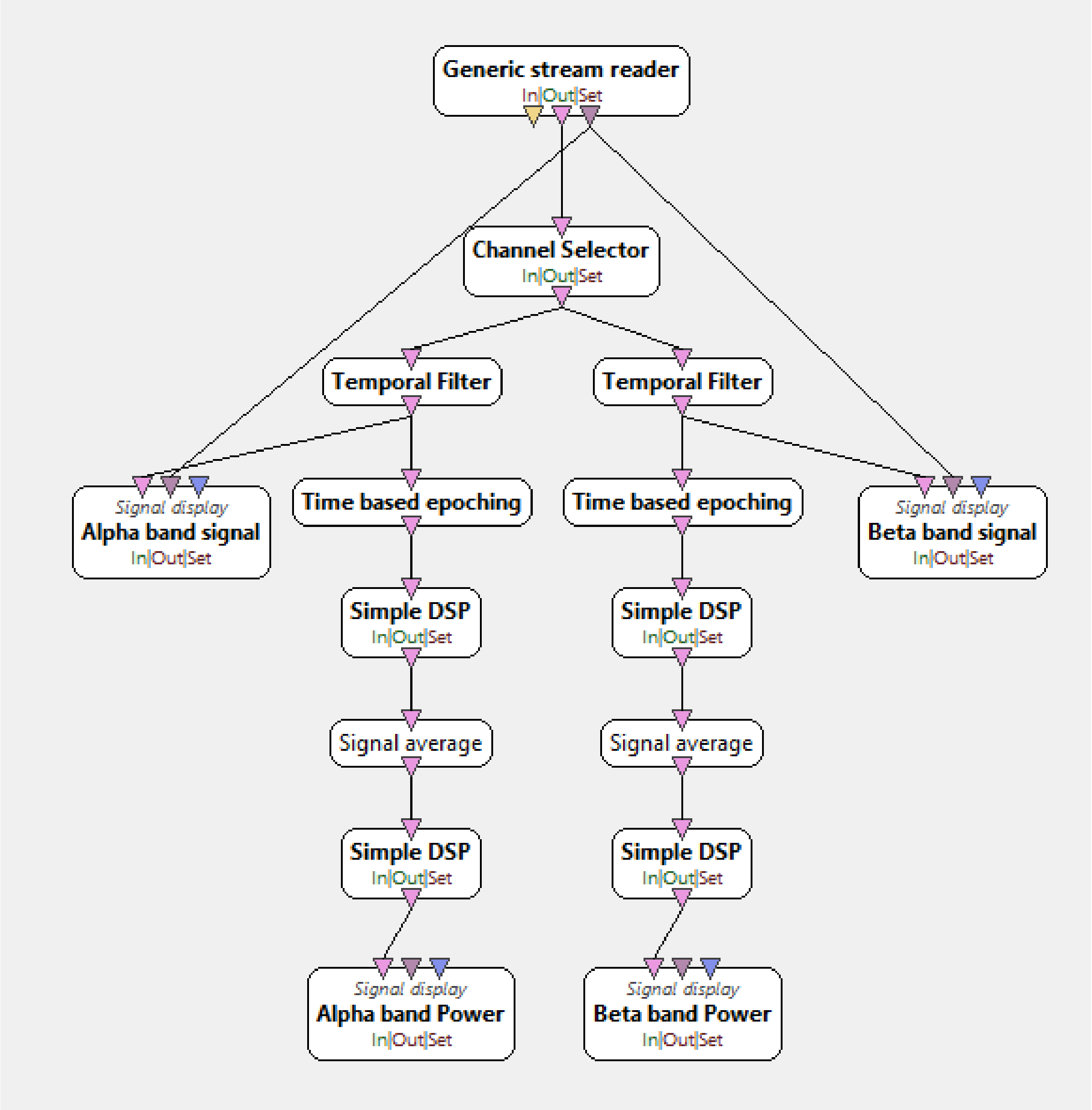
- Repeat for **Beta Band** [12;24]Hz
- Don't hesitate to copy/paste boxes…!






Generic stream reader
In|Out|Set
Channel Selector
In|Out|Set
Temporal Filter
Temporal Filter
Signal display
Alpha band signal
In|Out|Set
Time based epoching
Time based epoching
Simple DSP
In|Out|Set
Simple DSP
In|Out|Set
Signal average
Signal average
Simple DSP
In|Out|Set
Simple DSP
In|Out|Set
Signal display
Alpha band Power
In|Out|Set
Signal display
Beta band Power
In|Out|Set


Default window
Default tab
Alpha band signal
Beta band signal
Alpha band Power
Beta band Power
1 : C3
2 : C4
3 : FC3
4 : FC4
5 : C5
6 : C1
20.000000
25.000000
50.000000
100.000000



- Scenario is available on the github repo:

BCI-OpenViBE-CuttingEEG2021/
scenarios/sc2-spectralAnalysis.xml

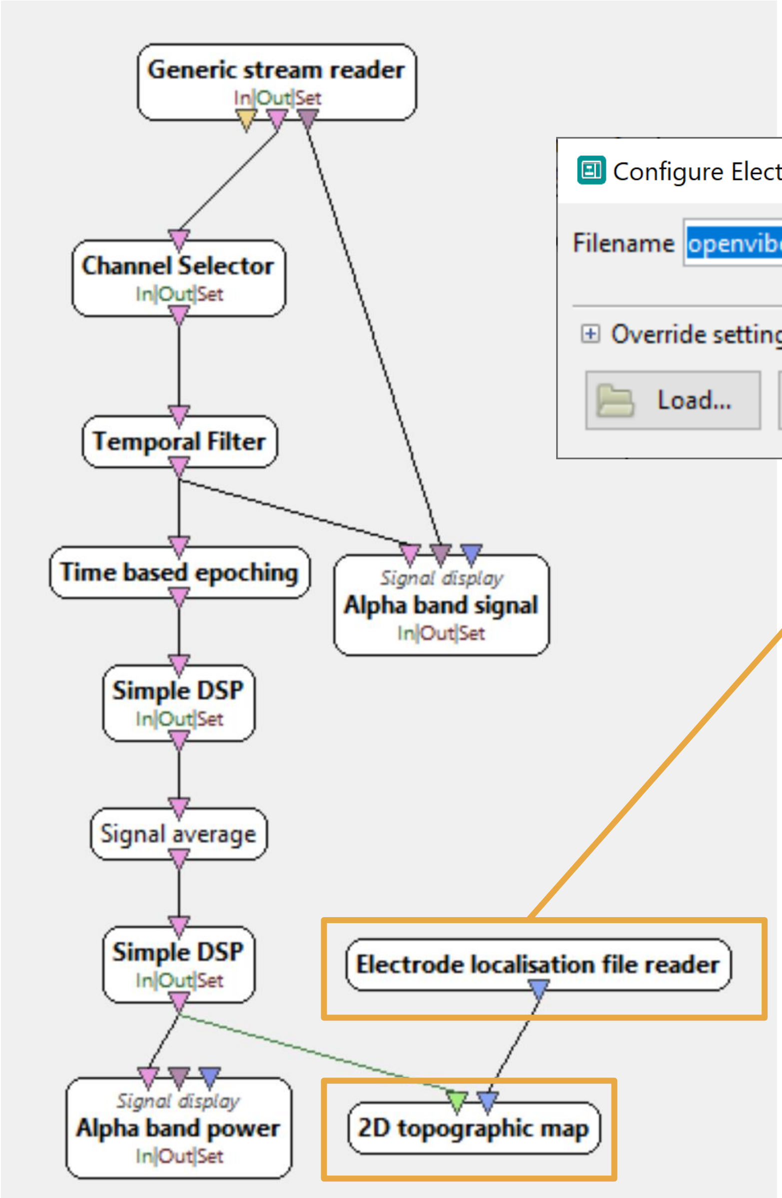
SC2 – STEP 3 – VISUALIZING BRAIN TOPOGRAPHY

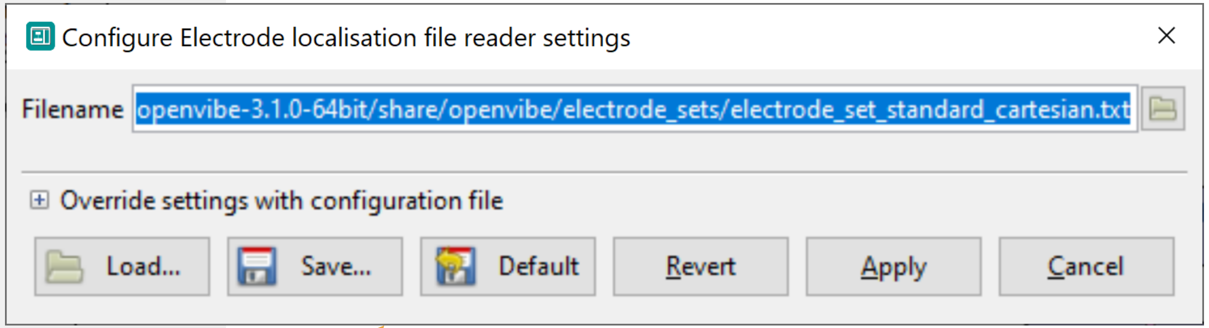
- **Sub-chapters:**
  - Channel Selection
  - Basic Signal Processing / Spectral analysis
  - Topography visualization
  - Feature Selection + Spatial Filtering
  - Classifier training

SC2 – STEP 3 – VISUALIZING BRAIN TOPOGRAPHY

- **Goals:**

- Load electrode localization data
- Topography viewer for a specific band

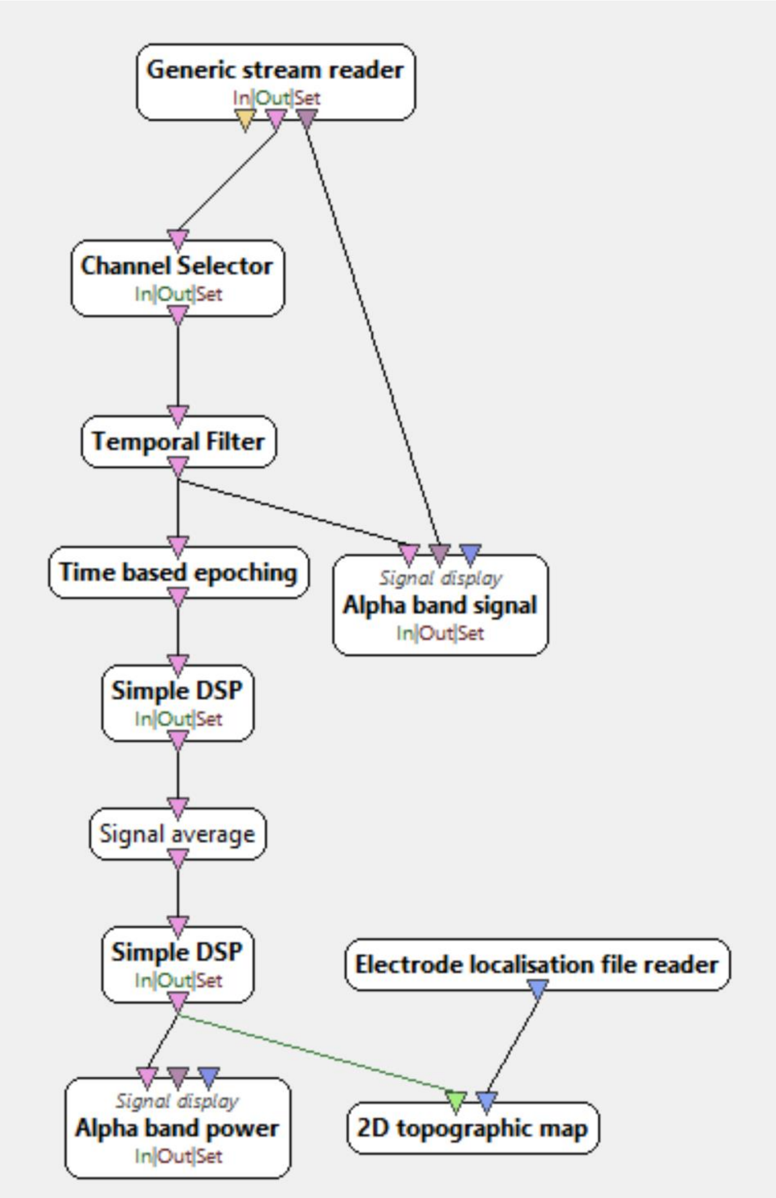




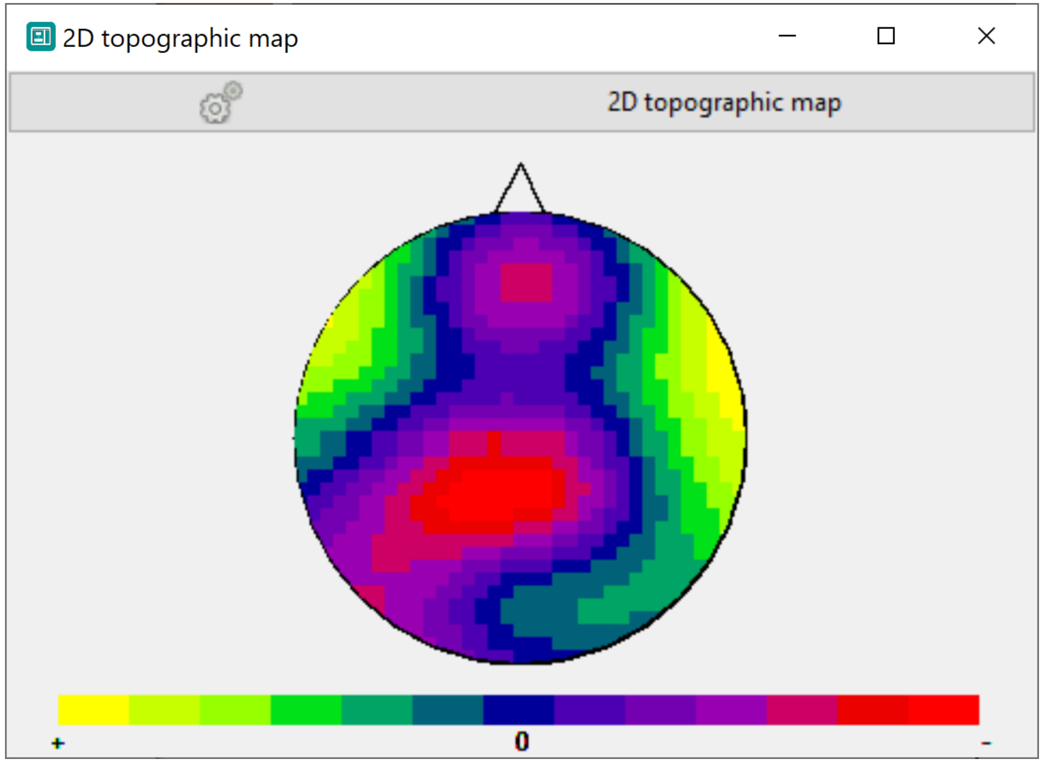
- Using the previous scenario computing the spectral power (in band alpha only):
- Add “**Electrode localization file reader**” and check the file location

<openvibe-3,1,0-64bits>/share/openvibe/electrode_sets/electrode_set_standard_cartesian.txt

- Add “**2D topographic map**”
- Connect boxes and play the scenario



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
-

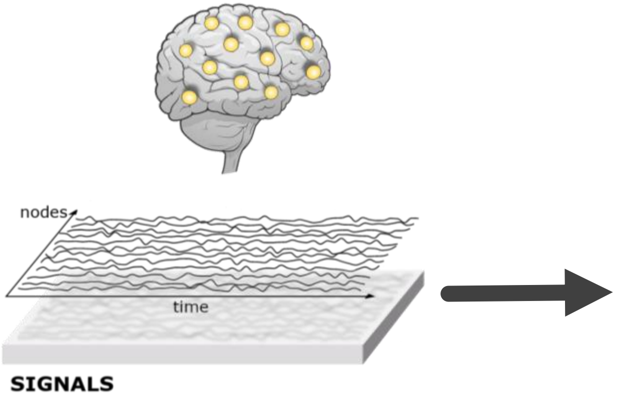
SC2 – STEP 4 – FEATURE SELECTION

- **Sub-chapters:**
  - Channel Selection
  - Basic Signal Processing / Spectral analysis
  - Topography visualization
  - Feature Selection + Spatial Filtering
  - Classifier training

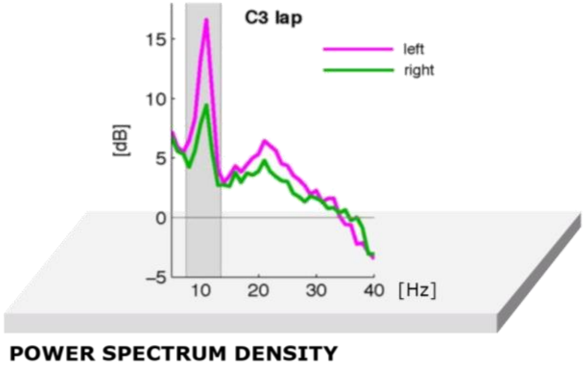
SC2 – STEP 4 – FEATURE SELECTION

Features to extract (recap)

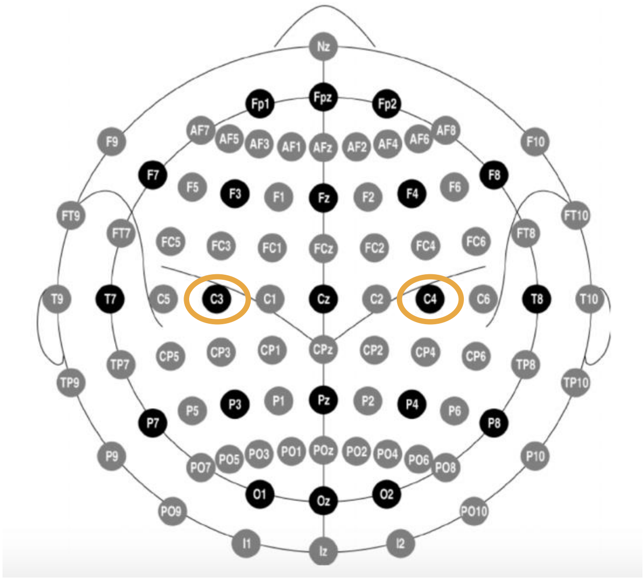
- Power spectra
- Sensorimotor area
- Mu: 8-12Hz &/OR Beta: 14-29Hz



(Gonzalez-Astudillo et al, 2020)



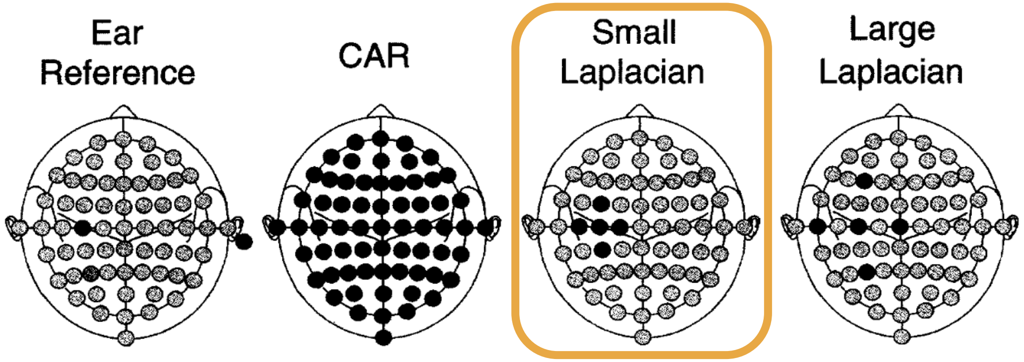
(Maeder et al., 2012)



SC2 – STEP 4 – FEATURE SELECTION – SPATIAL FILTERING

- **Spatial Filtering**
  - Common Average Reference (CAR)
  - Surface Laplacian – used here

C3' = 4*C3 - FC3 - C5 - C1 - CP3
C4' = 4*C4 - FC4 - C2 - C6 - CP4

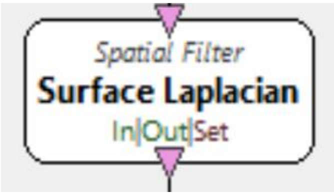


Spatial filtering
(McFarland et al, 1997)

- OV box: **Spatial Filter**
- **10 inputs => 2 outputs**

  C3' = 4*C3 - FC3 - C5 - C1 - CP3

  C4' = 4*C4 - FC4 - C2 - C6 - CP4



- **Our list of channels:**

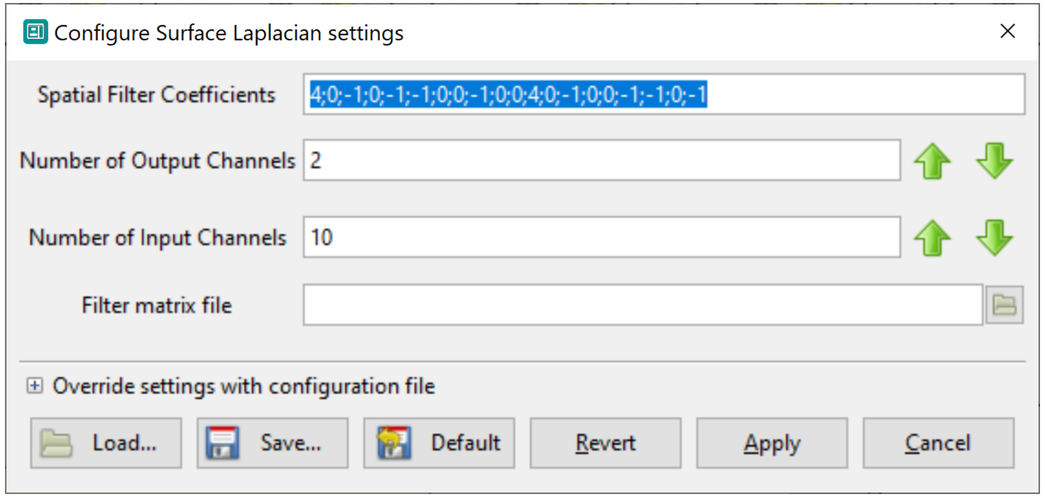
  C3; C4; FC3; FC4; C5; C1; C2; C6; CP3; CP4

C3' = 4 ; 0 ; -1 ; 0 ; -1; -1; 0; 0; -1 ; 0

C4' = 0 ; 4 ; 0 ; -1 ; 0 ; 0 ; -1; -1; 0 ; -1

- **Serialize formulae in box parameters**

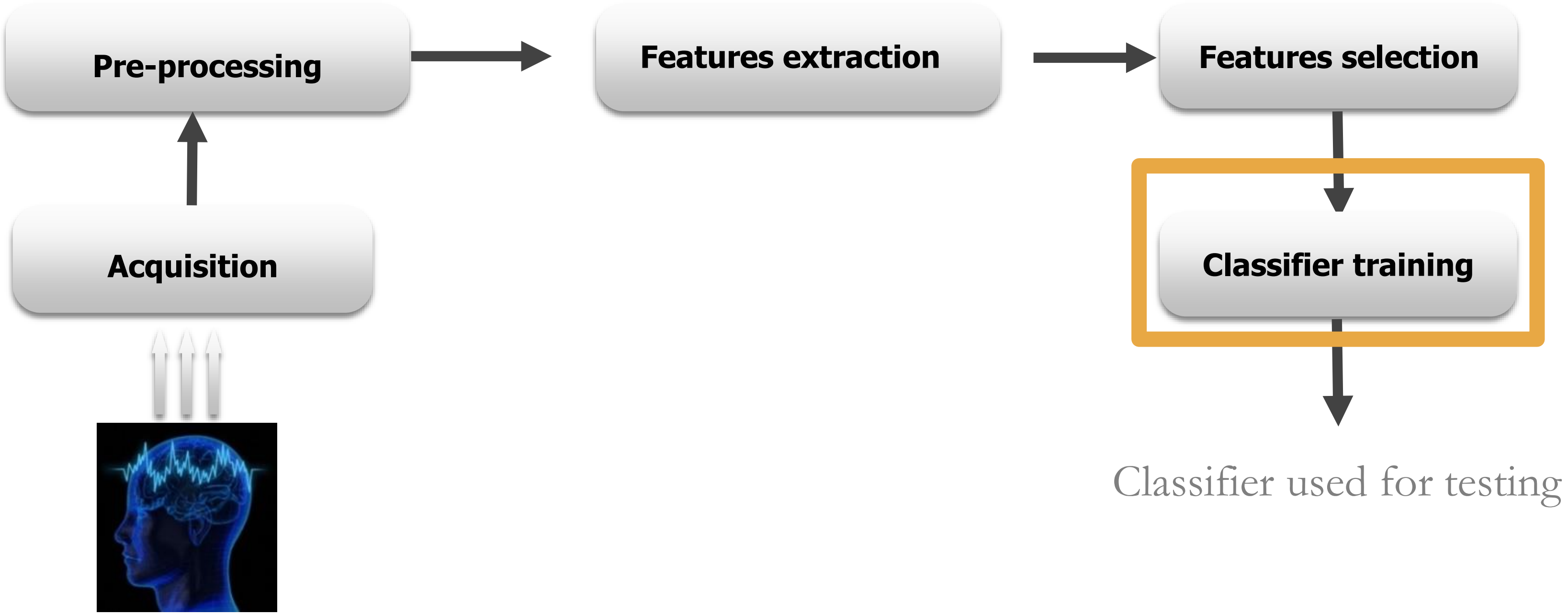
4;0;-1;0;-1;-1;0;0;-1;0;0;4;0;-1;0;0;-1;-1;0;-1



SC2 – STEP 5 – CLASSIFIER TRAINING

- **Sub-chapters:**
  - Channel Selection
  - Basic Signal Processing / Spectral analysis
  - Topography visualization
  - Feature Selection + Spatial Filtering
  - Classifier training

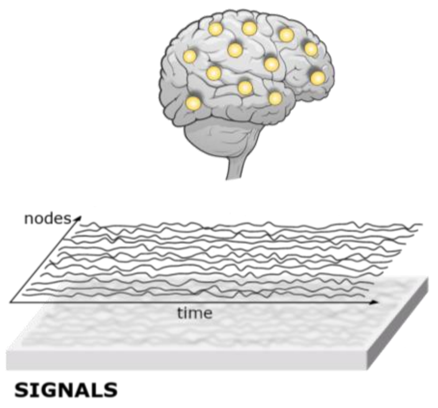
SC2 – STEP 5 – CLASSIFIER TRAINING



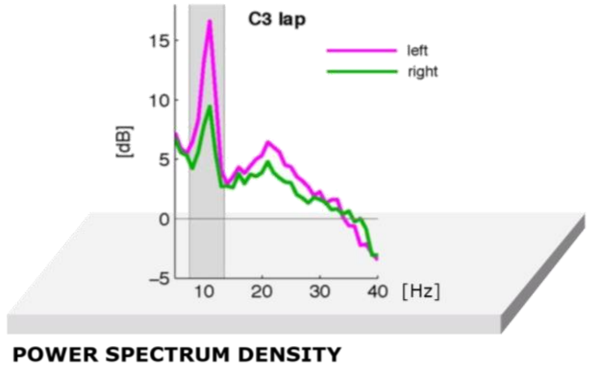
Adapted from (X. Navarro & F. Grosselin)

SC2 – STEP 5 – CLASSIFIER TRAINING

- **Goal reminder:**
- We want to train a classifier to distinguish between mental states, based on instantaneous spectral power in bands alpha and/or beta



(Gonzalez-Astudillo et al, 2020)



(Maeder et al., 2012)

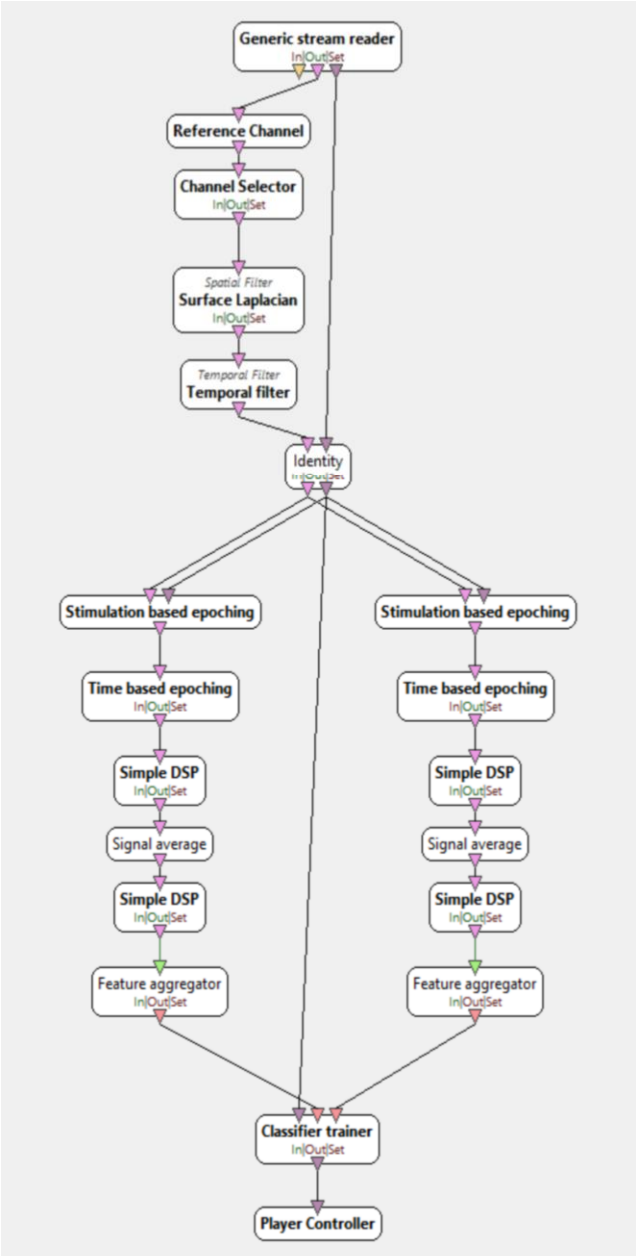
SC2 – STEP 5 – CLASSIFIER TRAINING

- **Load the full scenario so we can analyze it step by step…**

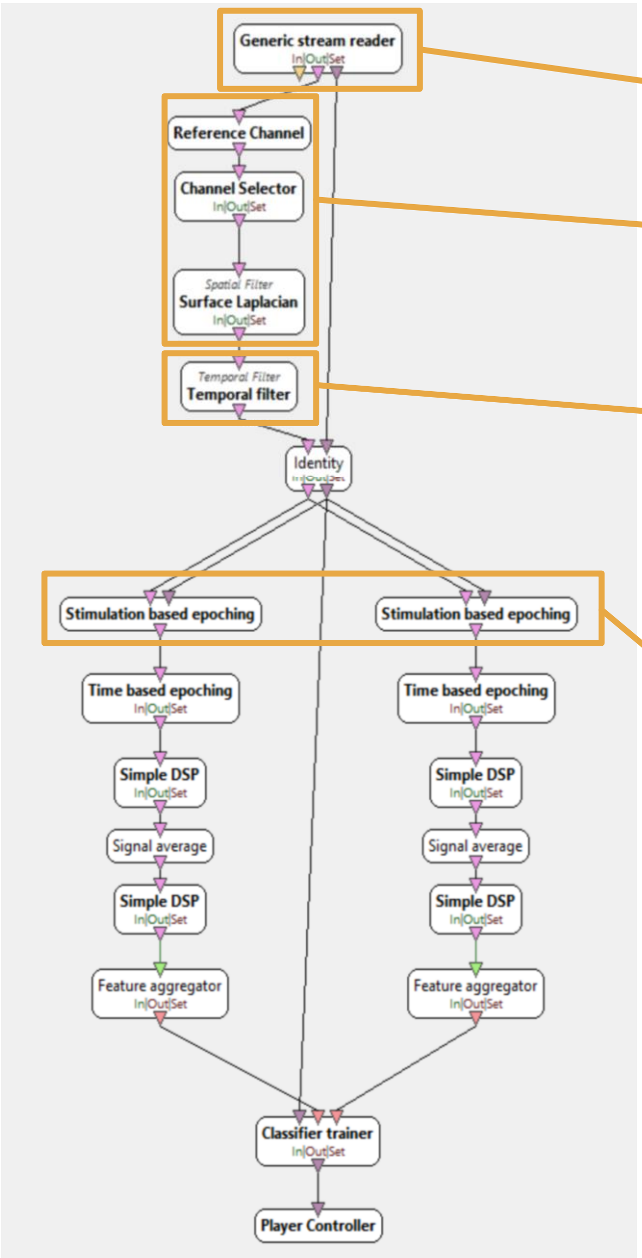
<OpenViBE_install_path>\
share\openvibe\scenarios\bci-examples\motor-imagery\
motor-imagery-bci-2-classifier-trainer.xml

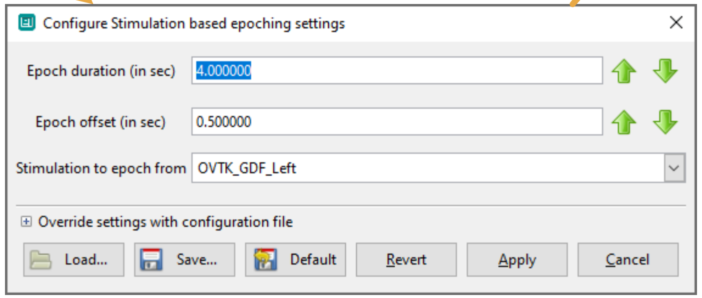
SC2 – STEP 5 – CLASSIFIER TRAINING

- **Goal: putting it all together in a BCI scenario…**
  - **Load** pre-recorded signal file
  - Select **sub-set of electrodes**
  - Surface Laplacian
  - Filter to the **frequency band** of interest
  - Stimulation-based Epoching: **split between conditions** (LEFT vs RIGHT…)
  - Epoching & Signal processing: compute a "**classifier feature**" based on **spectral power**
  - **Train the classifier!**

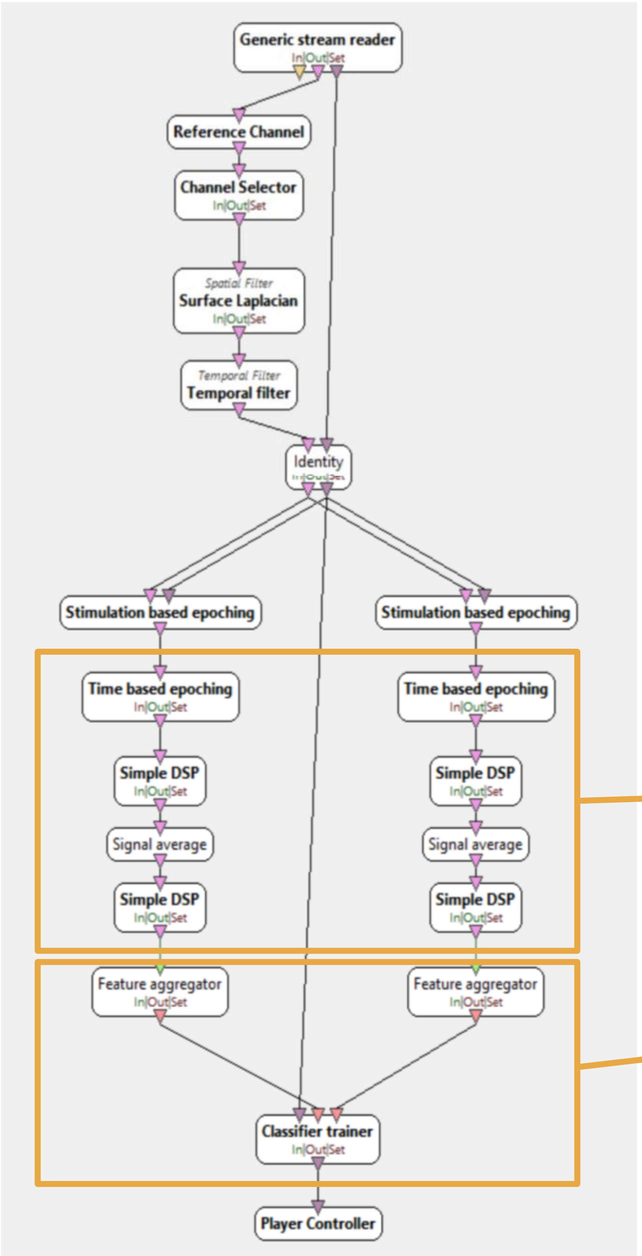


- **Load** pre-recorded signal file
- Select **sub-set of electrodes,** set **Reference Channel + apply spatial filtering**
- Filter to the **frequency band** of interest
- Stimulation-based Epoching: **split between conditions** (LEFT vs RIGHT…)
- Epoching & Signal processing: compute a "**classifier feature**" based on **spectral power**
- **Train the classifier!**

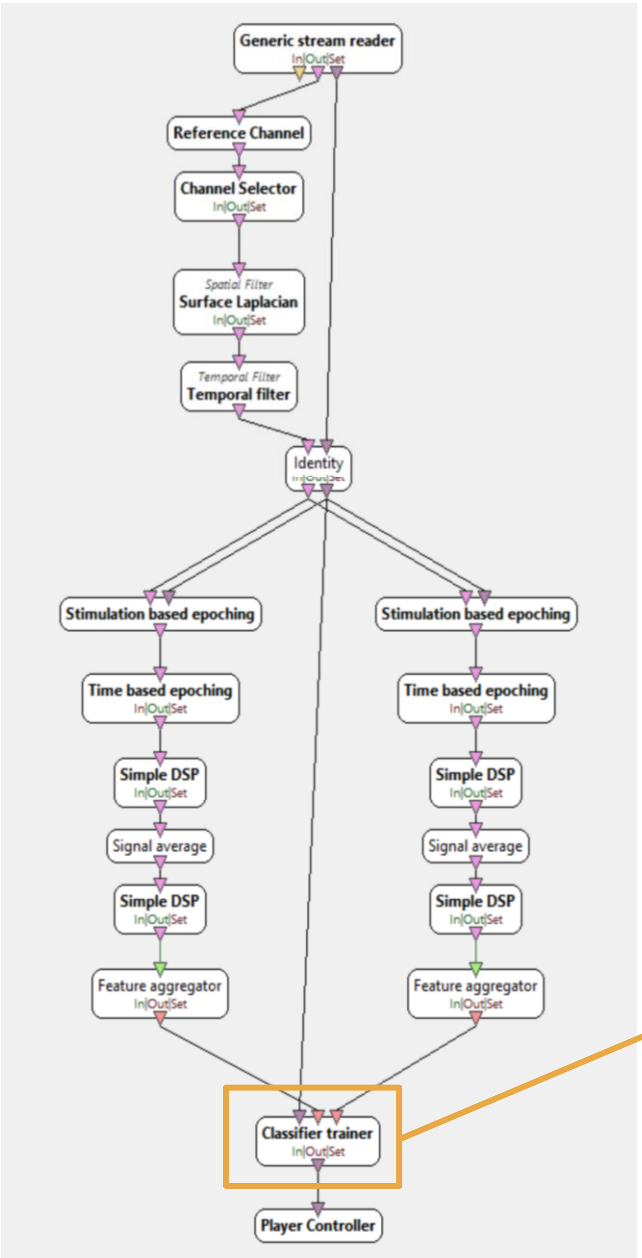


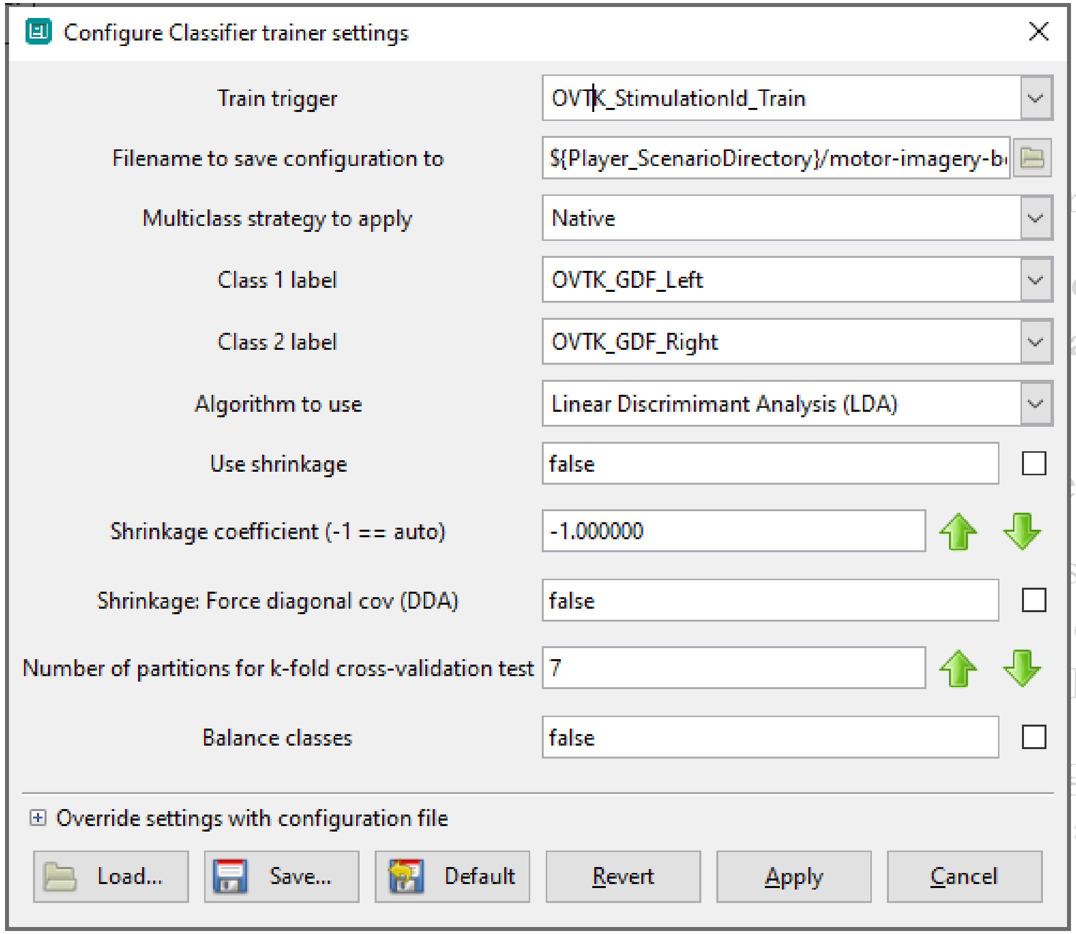


- **Load** pre-recorded signal file
- Select **sub-set of electrodes,** set **Reference Channel + apply spatial filtering**
- Filter to the **frequency band** of interest
- Stimulation-based Epoching: **split between conditions** (LEFT vs RIGHT…)
- Epoching & Signal processing: compute a "**classifier feature**" based on **spectral power**
- **Train the classifier!**



- **Load** pre-recorded signal file
- Select **sub-set of electrodes,** set **Reference Channel + apply spatial filtering**
- Filter to the **frequency band** of interest
- Stimulation-based Epoching: **split between conditions** (LEFT vs RIGHT…)
- Epoching & Signal processing: compute a "**classifier feature**" based on **spectral power**
- **Train the classifier!**





rded signal file

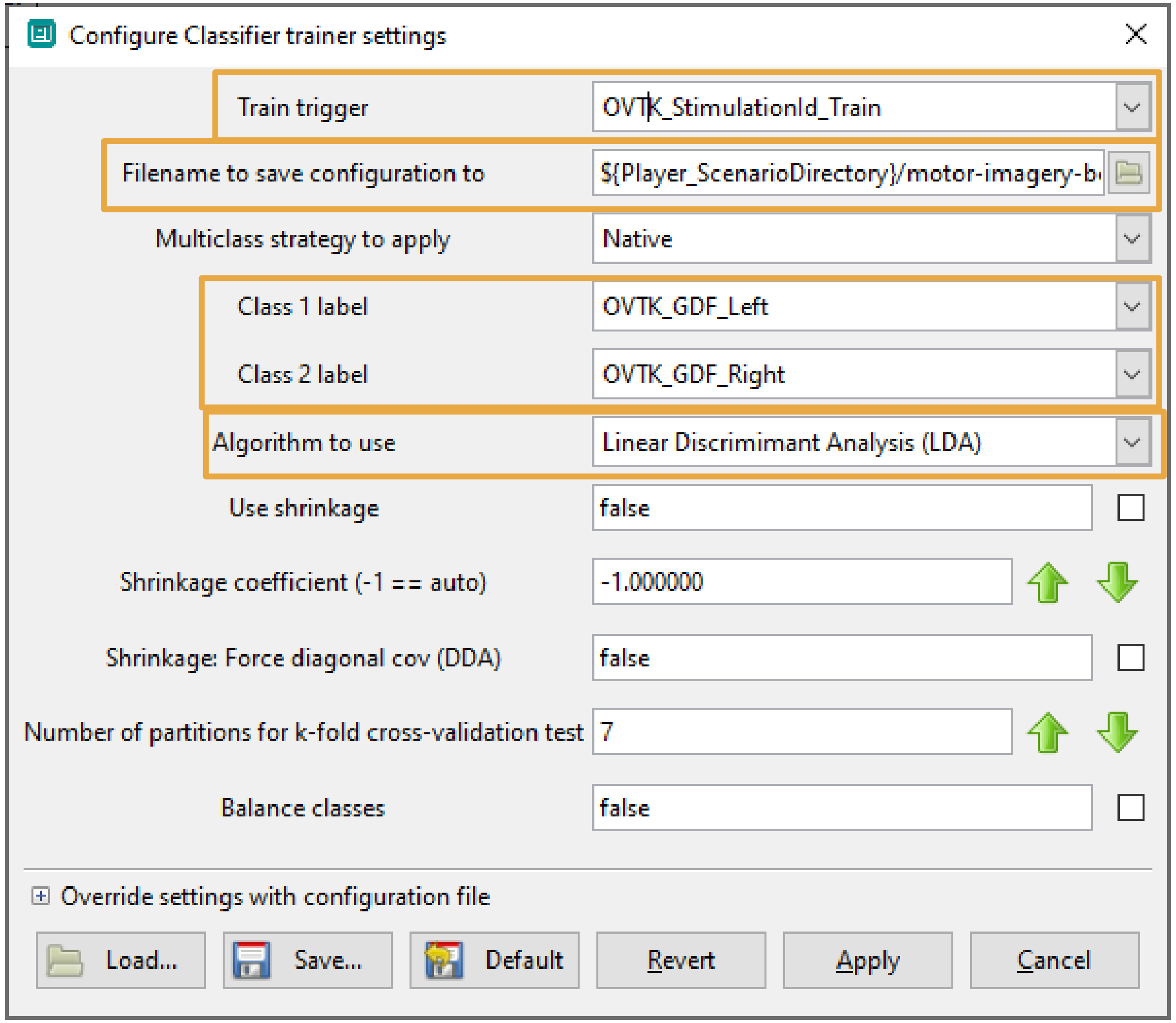
of electrodes, set
annel + apply spatial

equency band of interest

sed Epoching:
conditions
HT…)

gnal processing: compute
ature" based on spectral

- **Train the classifier!**



- **"Train trigger"** stimulation should be set in the acquisition LUA script, at the end of the experiment
- Set the **path/filename** for the training "weights" (results)
- Check that the classifier classes are correctly labeled
- Select the classifying algorithm (LDA, SVM…) and set the parameters

```
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Received train stimulation. Data dim is [1764x2]
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> For information, we have 931 feature vector(s) for input 1
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> For information, we have 833 feature vector(s) for input 2
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> k-fold test could take quite a long time, be patient
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 1 / 7 (performance : 75.000000%)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 2 / 7 (performance : 74.206349%)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 3 / 7 (performance : 59.126984%)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 4 / 7 (performance : 88.492063%)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 5 / 7 (performance : 75.000000%)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 6 / 7 (performance : 88.095238%)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 7 / 7 (performance : 82.539683%)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Cross-validation test accuracy is 77.494331% (sigma = 9.406674%)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Cls vs cls       1      2
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  1:    82.7  17.3 %, 931 examples
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  2:    28.3  71.7 %, 833 examples
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Training set accuracy is 80.045351% (optimistic)
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Cls vs cls       1      2
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  1:    85.2  14.8 %, 931 examples
[ INF ] At time 462.008 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  2:    25.7  74.3 %, 833 examples
```

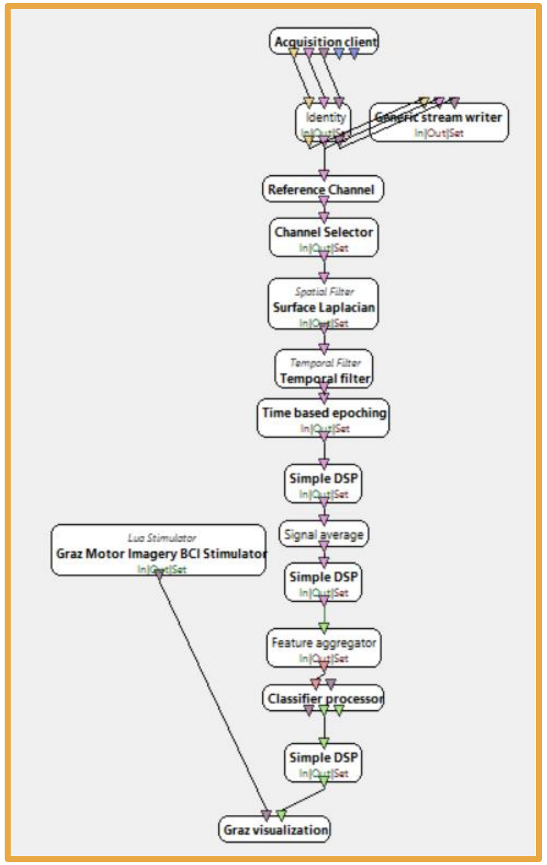
SC2 – FINAL WORDS / Q&A

- **Recap:**
  - Signal processing / Channel selection / Filtering / Epoching
  - Spectral Analysis, spectrum display
  - Brain topography
  - Feature selection
  - Classifier training

SC3 – ONLINE CLASSIFICATION & FEEDBACK

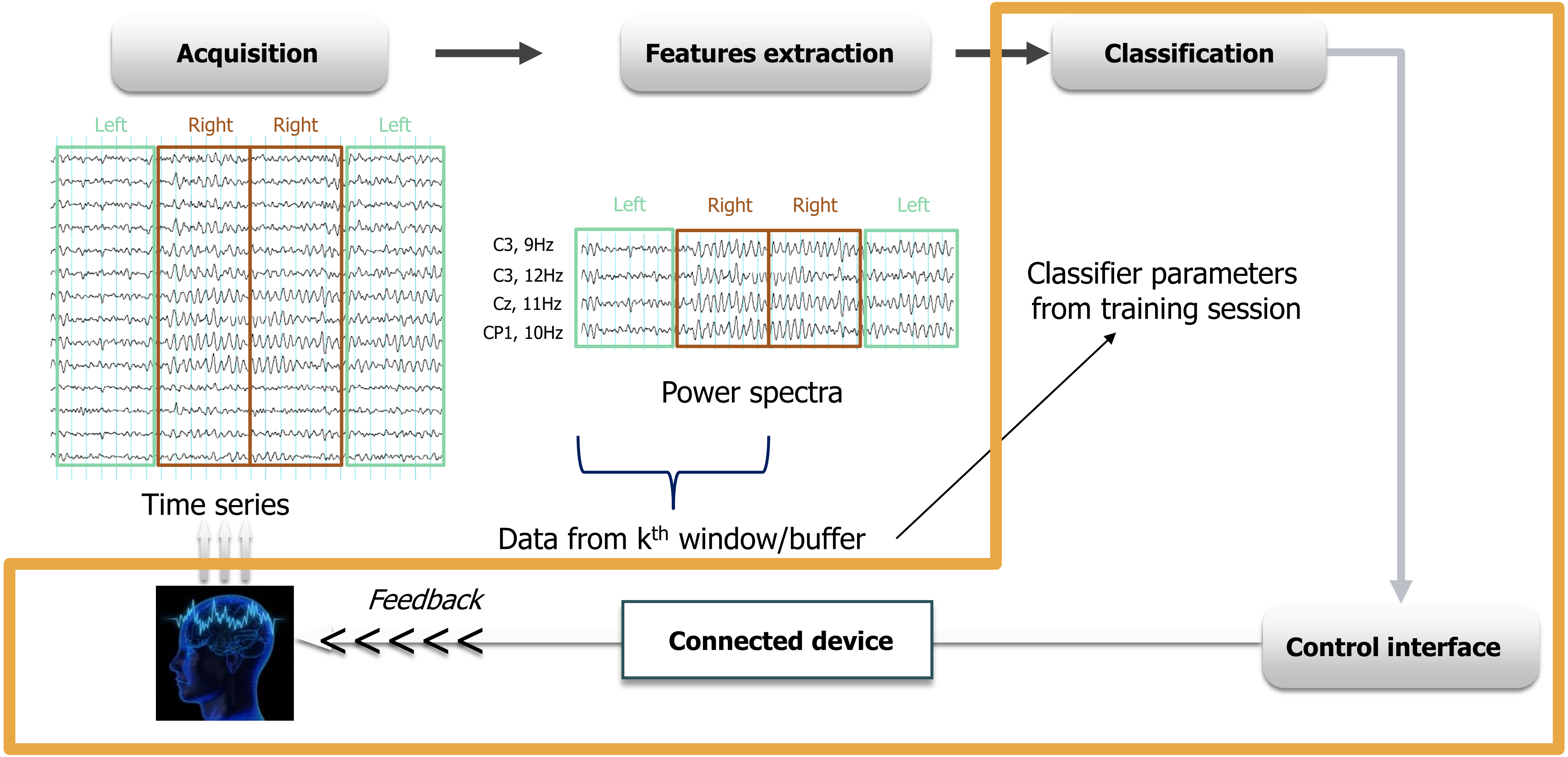
1- Training data acquisition

2- Features extraction
& Classification



3- Online feedback

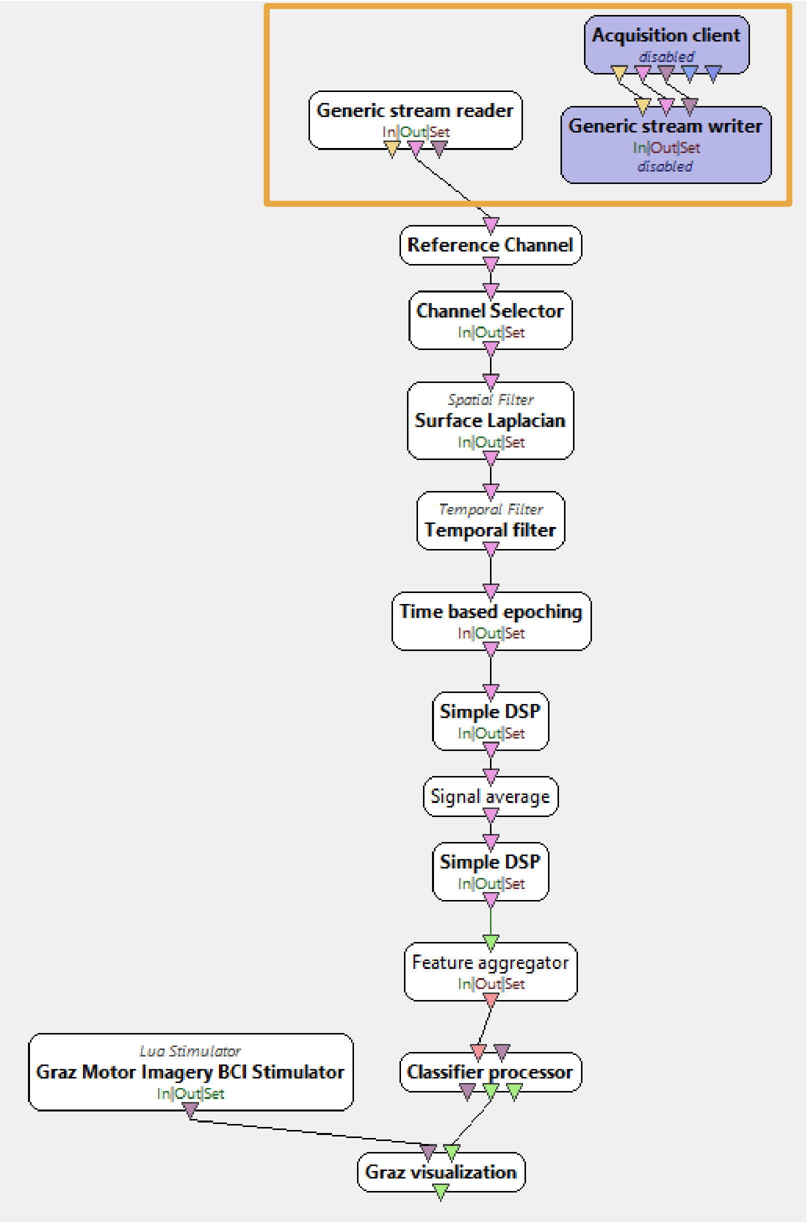
SC3 – ONLINE CLASSIFICATION



SC3 – ONLINE CLASSIFICATION

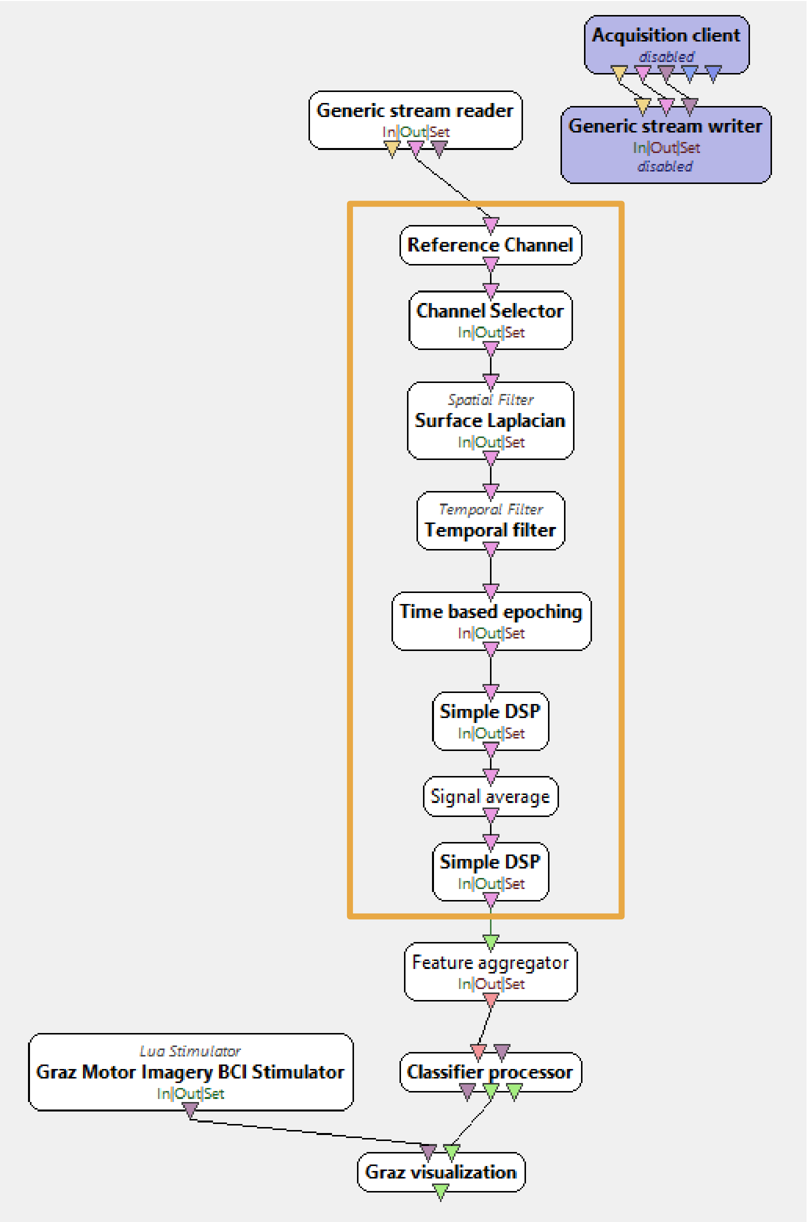
- **Goals:**
  - Acquire EEG data in real-time, while providing “tasks” to the subject
  - Classify his/her mental states (also in real-time) using the classifier trained in SC2
  - (almost) everything we need is already there!
- **Let’s load the scenario and analyze it…**

  <OpenViBE_install_path>\
  share\openvibe\scenarios\bci-examples\motor-imagery\
  motor-imagery-bci-3-online.xml



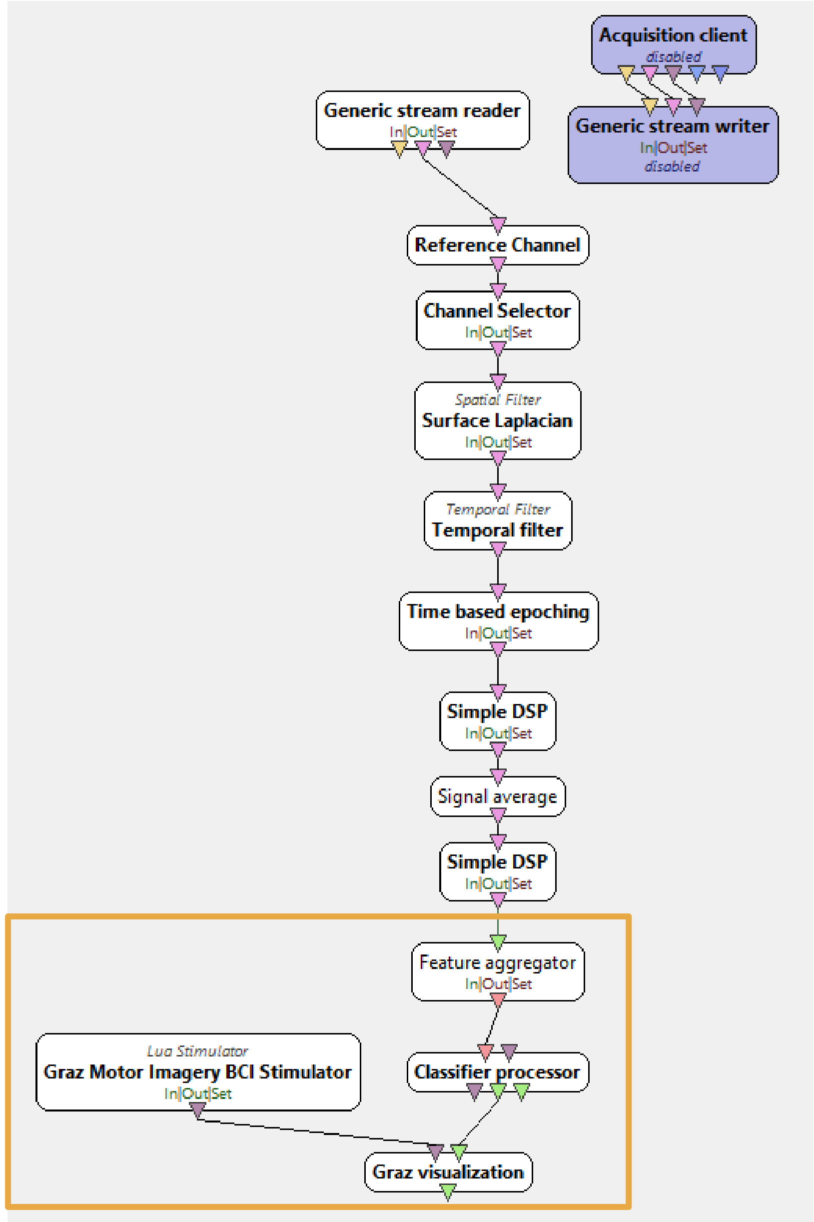
- In a real BCI experiment, we should of course use the Acquisition Client (and record the incoming signals for future replays and studies)

- Here, for the sake of an already long workshop, we'll use the same signal file as before…



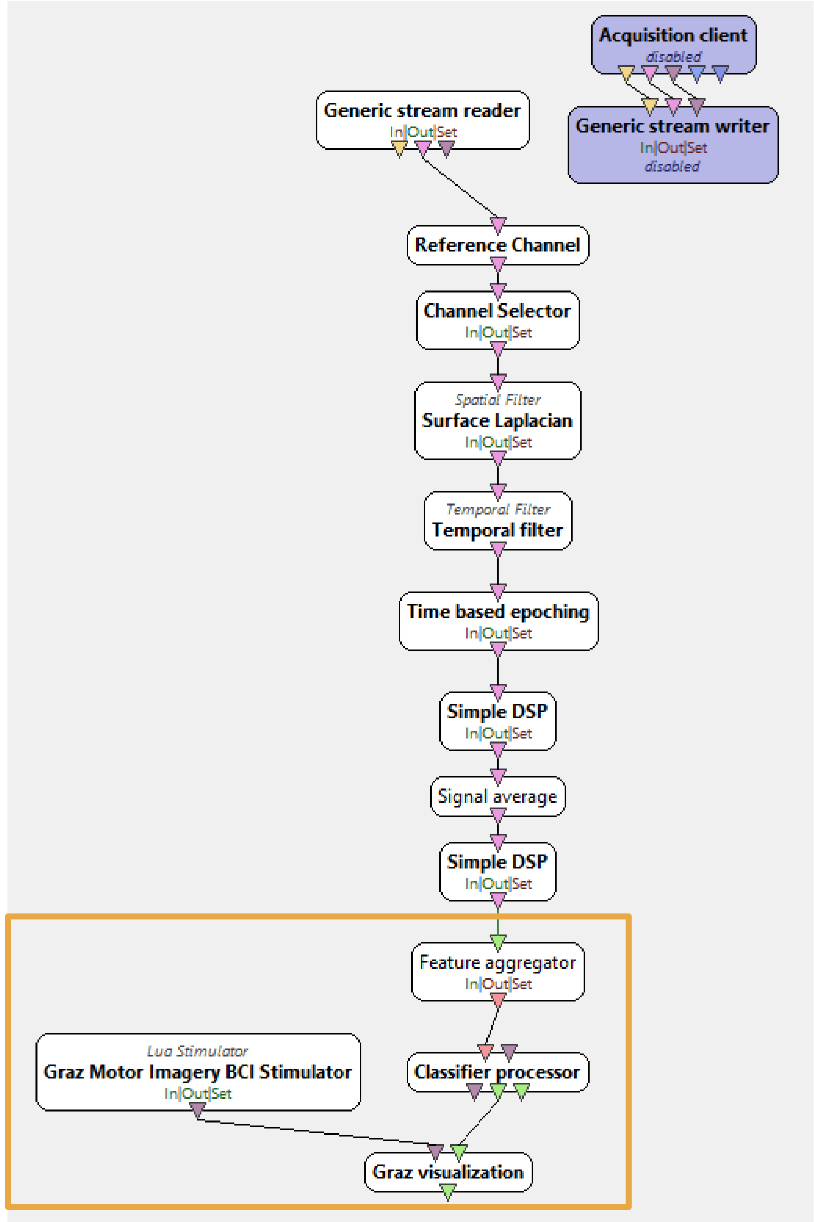
- We want the classifier to have the same type of data used for training
- (It's actually the other way around: train your classifier with the same data type you use in the online step)

- The whole selection, filtering, epoching & Signal Processing chain is the same as in SC2 !

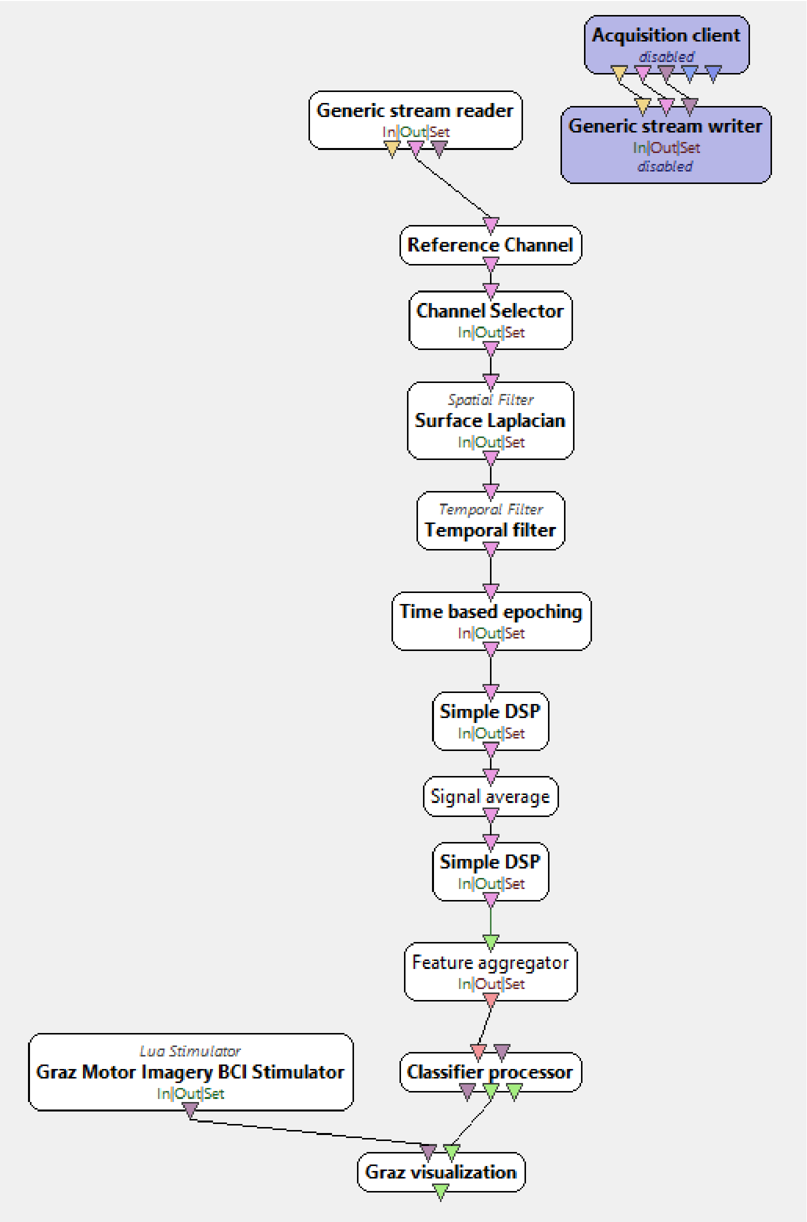


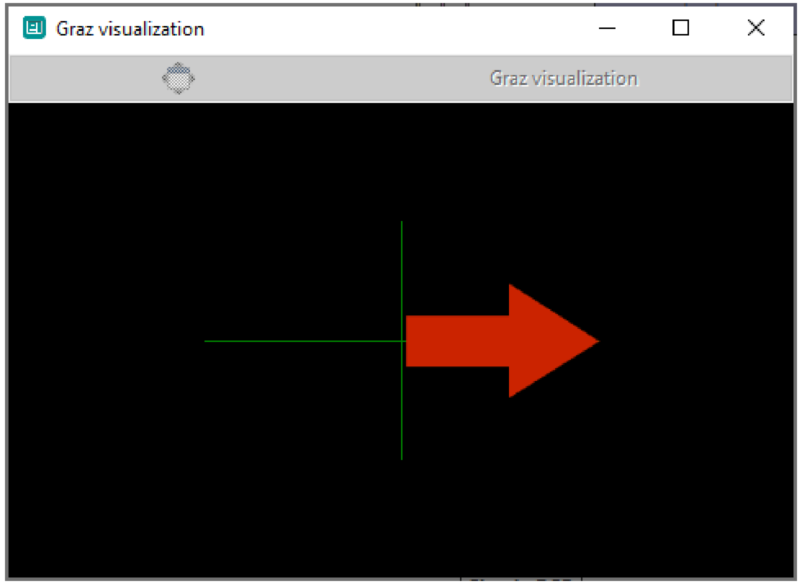
- Use the "**Classifier Processor**" box, loading the correct configuration file (= the training step output)

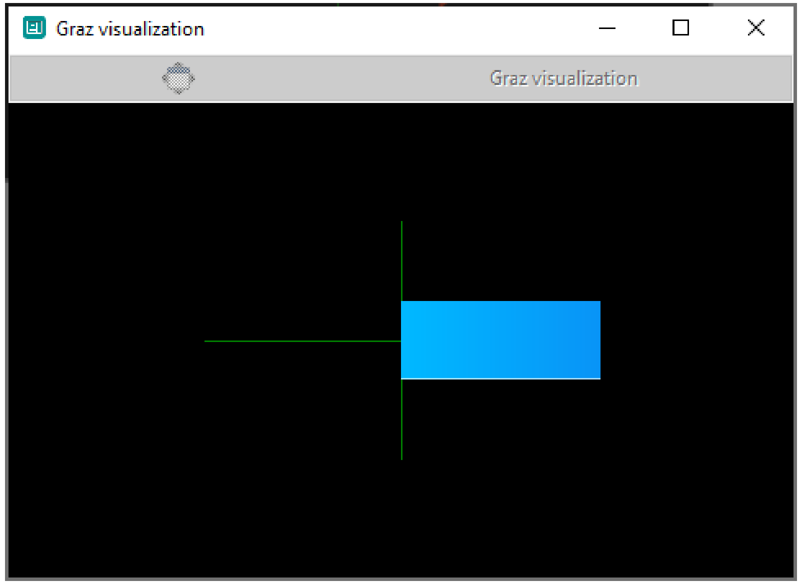
- Use the same LUA script & Graz Visualization boxes as for the SC1 (acquisition) to generate stimulations & "tasks"
- The "**streamed matrix**" stream from **Classifier processor** to **Graz Visualization** corresponds to the "classification **accuracy**"



- **Notes** regarding the difference between this simulation and an actual BCI/MI protocol
- In a real BCI experiment, the user would receive as visual inputs:
  - first, an arrow towards the side on which to perform Motor Imagery task,
  - then a continuous feedback of the classifier's performance (in separating MI from rest)
- As in SC1, the Graz visualization box is used both to update the display, and to propagate the stimulations to the ACQ server for synchronization with the signal stream.
- Here, we use pre-recorded signals, so this stimulation sync. is not done…


Acquisition client
disabled
Generic stream reader
In|Out|Set
Generic stream writer
In|Out|Set
disabled
Reference Channel
Channel Selector
In|Out|Set
Spatial Filter
Surface Laplacian
In|Out|Set
Temporal Filter
Temporal filter
Time based epoching
In|Out|Set
Simple DSP
In|Out|Set
Signal average
Simple DSP
In|Out|Set
Feature aggregator
In|Out|Set
Lua Stimulator
Graz Motor Imagery BCI Stimulator
In|Out|Set
Classifier processor
Graz visualization


Graz visualization
Graz visualization


Graz visualization
Graz visualization

SC3 – FINAL WORDS / Q&A

- **Recap:**
  - Online classification pipeline
  - Example of real-time visual feedback

CHAPTER 2 – Q&A

BCI Motor Imagery with OpenViBE in X-Men: First Class