CONCLUDING REMARKS & PERSPECTIVES - OPENVIBE

ARAMIS LAB BRAIN DATA SCIENCE · Institut du Cerveau ICM · Inria · Inserm · CNRS · Sorbonne Université

PROTOTYPING, DESIGNING, TUTORIALS

- As seen in Chapter 2, with OpenViBE you can easily prototype and design BCI protocols and experiments

- Lots of **scenario examples and templates** are already available in the install!

  <openvibe-3.1.0-64bit>\share\openvibe\scenarios\bci-examples

- Wanting to use a particular box? **Tutorial scenarios** are there for you:

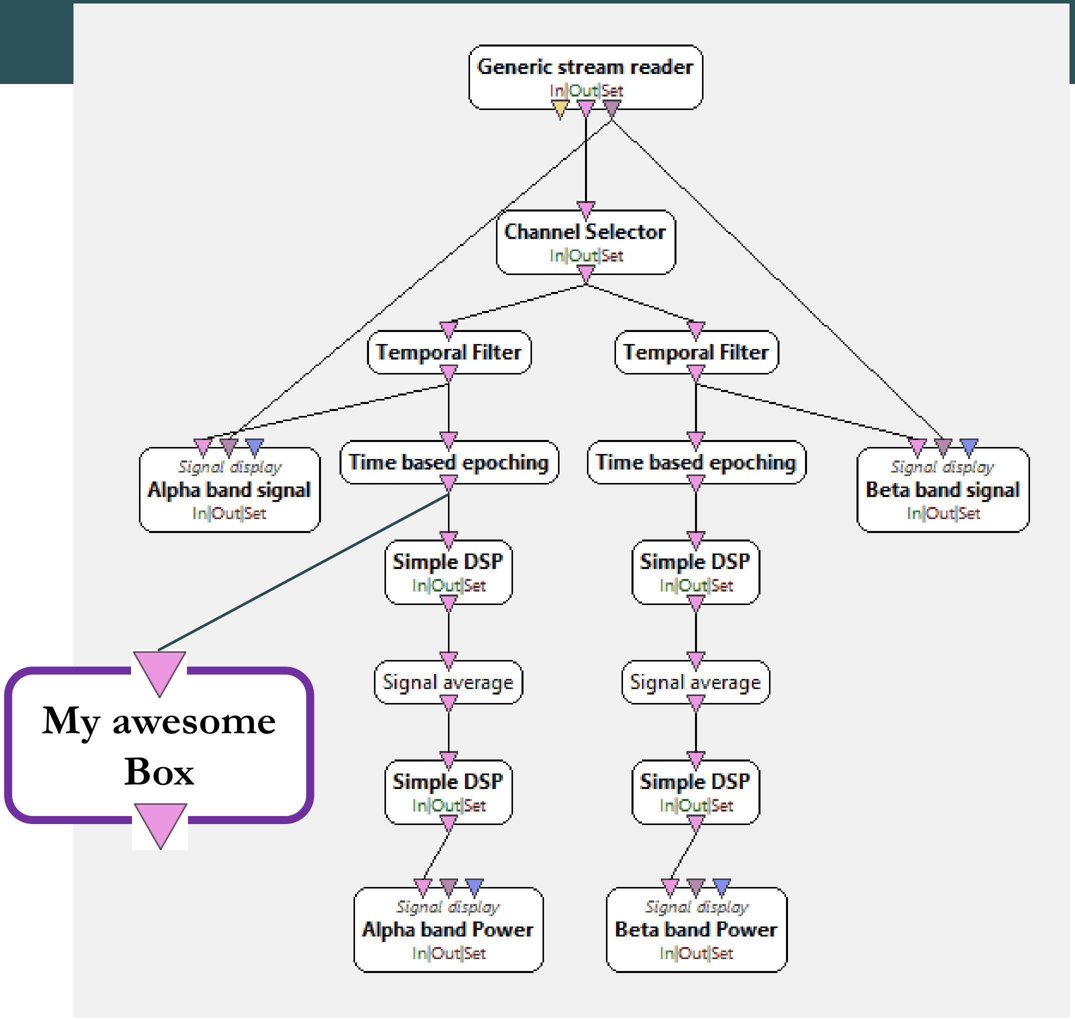
  <openvibe-3.1.0-64bit>\share\openvibe\scenarios\box-tutorials

- Check the general documentation for a great amount of info:

  http://openvibe.inria.fr/documentation-index/

BOX DEVELOPMENT

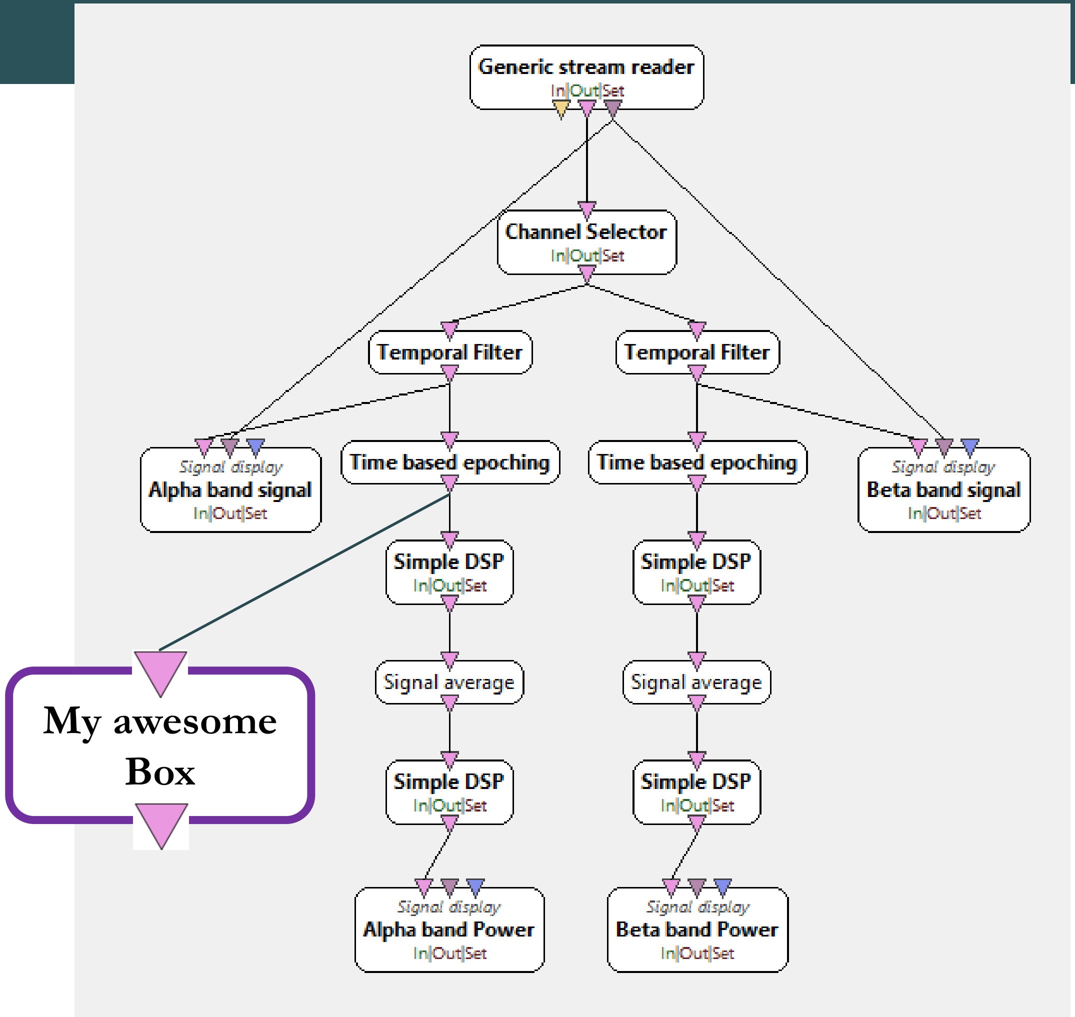
- So, you want to develop a **new processing box**?



BOX DEVELOPMENT

- So, you want to develop a **new processing box**?

- First step:
  check if an existing box has what you need…
  … or if you can do what you want using a combination of existing boxes.



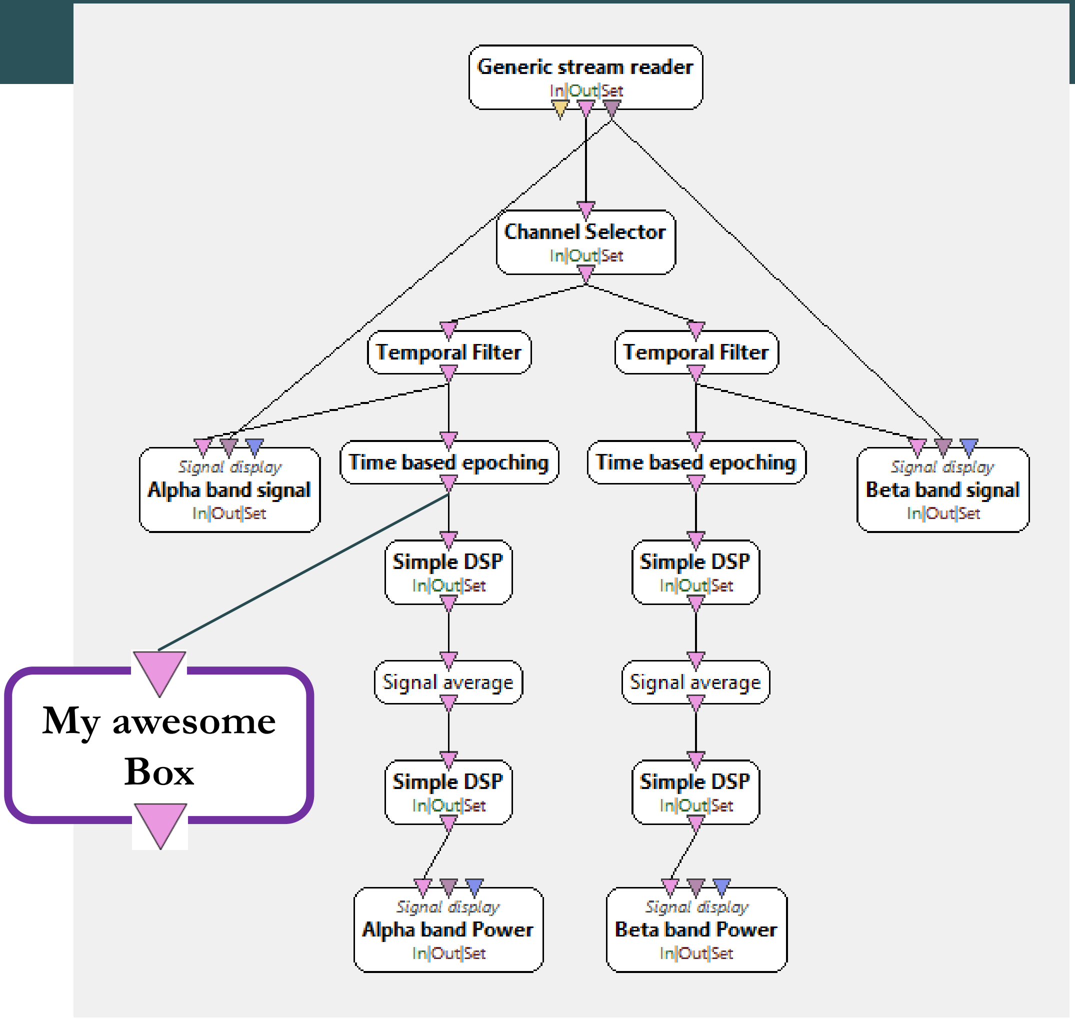
BOX DEVELOPMENT

- So, you want to develop a **new processing box**?

- **First step**:
  check if an existing box has what you need…
  … or if you can do what you want using a combination of existing boxes.

- If not – then:
  Do you want a quick&flexible prototype?
  → box calling **Python/Matlab scripts**

  … or a fine-tuned optimized algorithm?
  → **C++ Box & Algorithm** classes



BOX DEVELOPMENT – PYTHON/MATLAB

- **Using Python/Matlab scripts in OpenViBE scenarios**

Use cases:

  - Need for a quick proof-of-concept (e.g. signal processing)
  - Don’t want/need to code in C++
  - Python/Matlab implementation is already perfect
  - Need specific libraries (numpy, scikit-learn…)

http://openvibe.inria.fr/tutorial-using-matlab-with-openvibe/

http://openvibe.inria.fr/tutorial-using-python-with-openvibe/

- Great Python tutorial: (courtesy of MENSIA)
  - http://openvibe.inria.fr/openvibe/wp-content/uploads/2016/06/Quick-prototyping-in-OpenViBE-with-Python.pdf

BOX DEVELOPMENT – C++

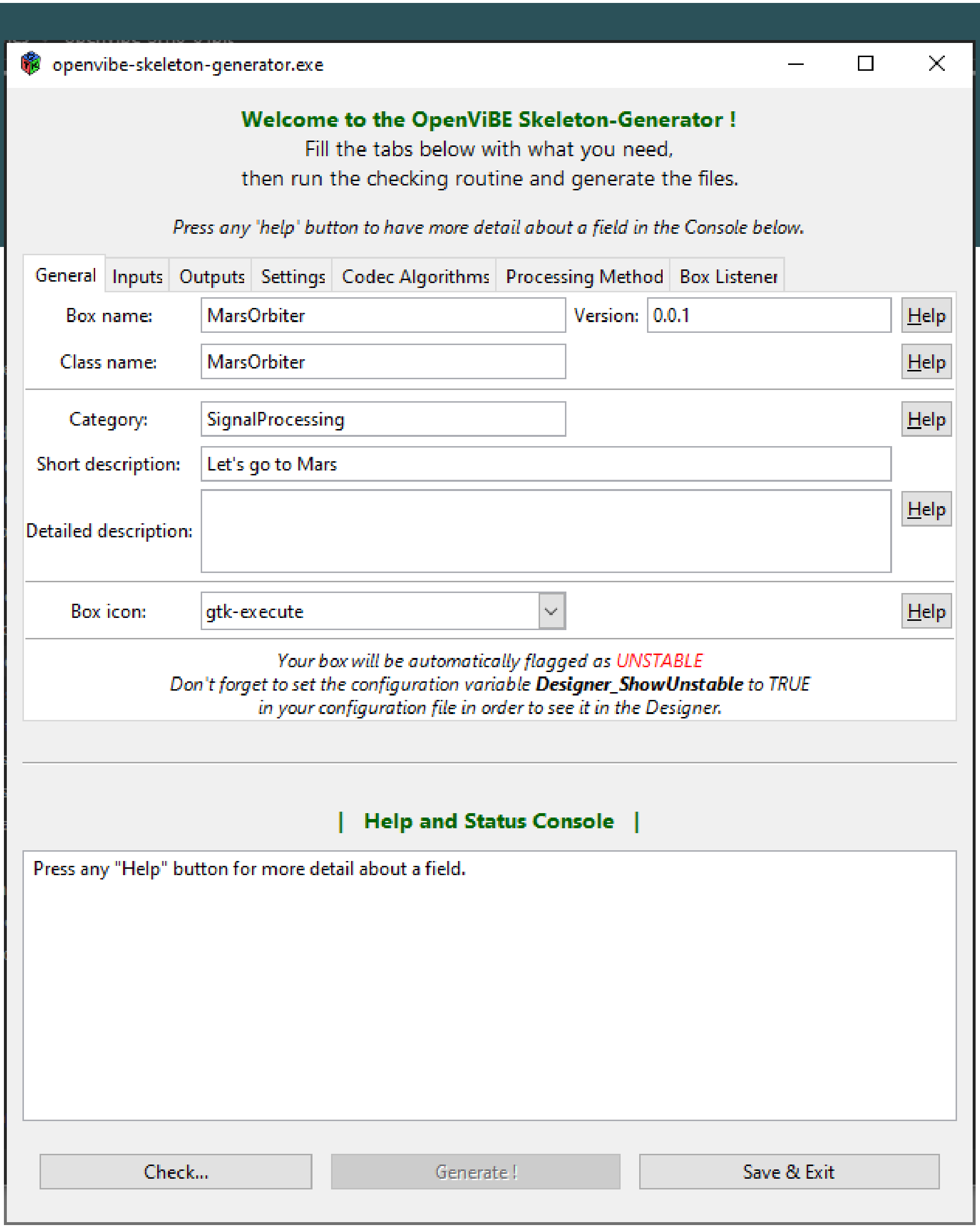
- **Developing C++ OpenViBE boxes**
- Use cases:
  - Need speed!
  - Complete integration with OpenViBE, contribution to the open-source project

- http://openvibe.inria.fr/build-instructions/
- **2016 Tutorial**: http://openvibe.inria.fr/openvibe/wp-content/uploads/2016/06/jl_hacking_boxes_2016.pdf

BOX DEVELOPMENT – C++

- **Skeleton generator**
  Simplest, fastest, go-to solution for beginners…
  openvibe-skeleton-generator.cmd
- GUI helping with creating the bare minimum a box needs, with given inputs/outputs, parameters, etc.

  All the "OpenViBE glue" is here!
  You "only" need to add your specific code.
- http://openvibe.inria.fr/tutorial-1-implementing-a-signal-processing-box/
- Tip: take inspiration from existing boxes…!



openvibe-skeleton-generator.exe

Welcome to the OpenViBE Skeleton-Generator !
Fill the tabs below with what you need,
then run the checking routine and generate the files.

Press any 'help' button to have more detail about a field in the Console below.

Box name: MarsOrbiter
Version: 0.0.1
Class name: MarsOrbiter
Category: SignalProcessing
Short description: Let's go to Mars
Detailed description:
Box icon: gtk-execute

Your box will be automatically flagged as UNSTABLE
*Don't forget to set the configuration variable **Designer_ShowUnstable** to TRUE*
in your configuration file in order to see it in the Designer.

| **Help and Status Console** |

Press any "Help" button for more detail about a field.

EXTERNAL INTERFACES

- Examples: interface w/ virtual reality products, video games, external data visualization toolboxes…
- Various ways exist to stream data/events between OpenViBE and external apps.
  - VRPN
  - TCP/IP
  - LSL (Lab Streaming Layer)
  - Python/Matlab boxes

- Demo using LSL to visualize Connectivity/Adjacency Matrices using an external Python script
- Code and example scenarios:

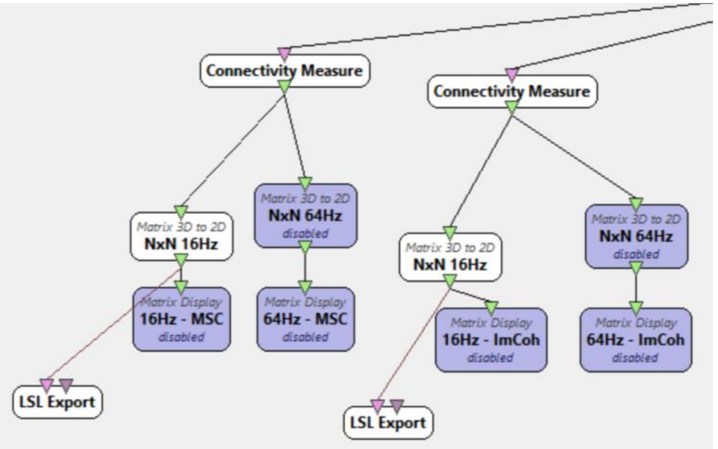
  https://github.com/AsteroidShrub/openVibe-Lsl-Demo

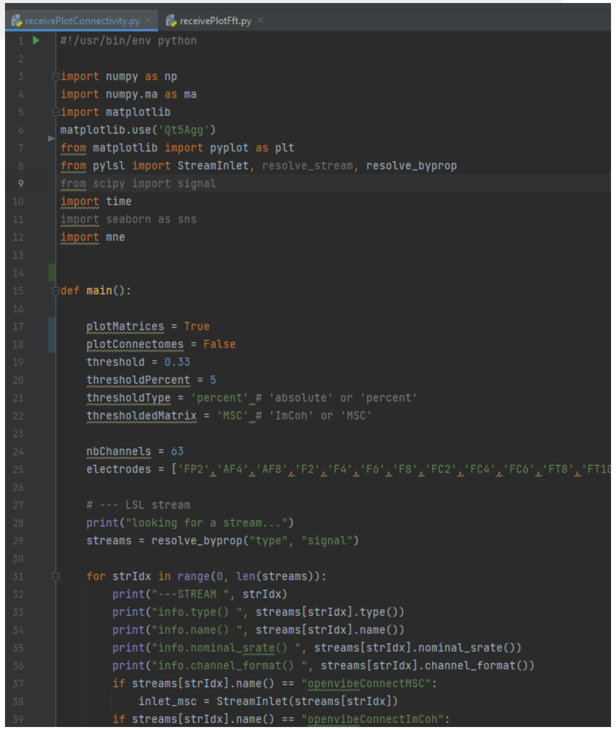
- OpenViBE “LSL Export” Box

Connectivity Measurement Box

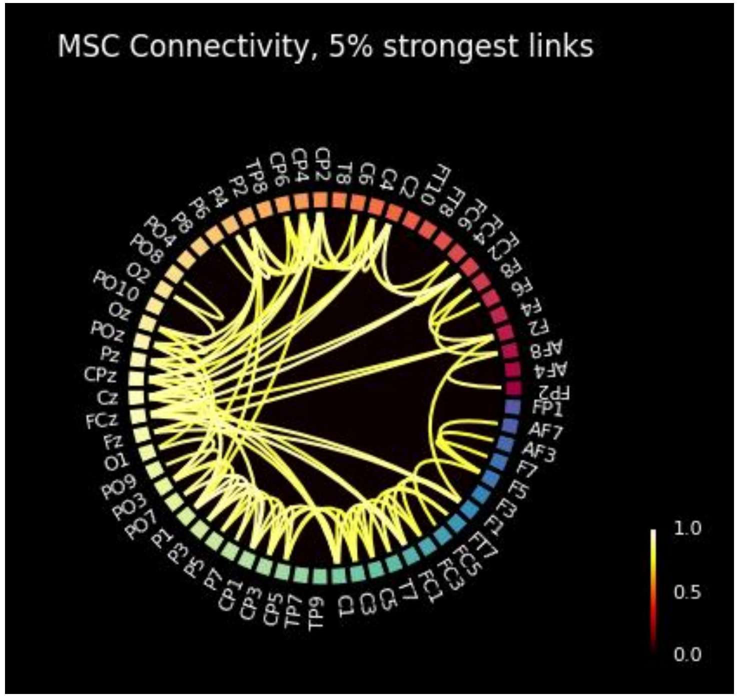
- + Python scripting (using pylsl)

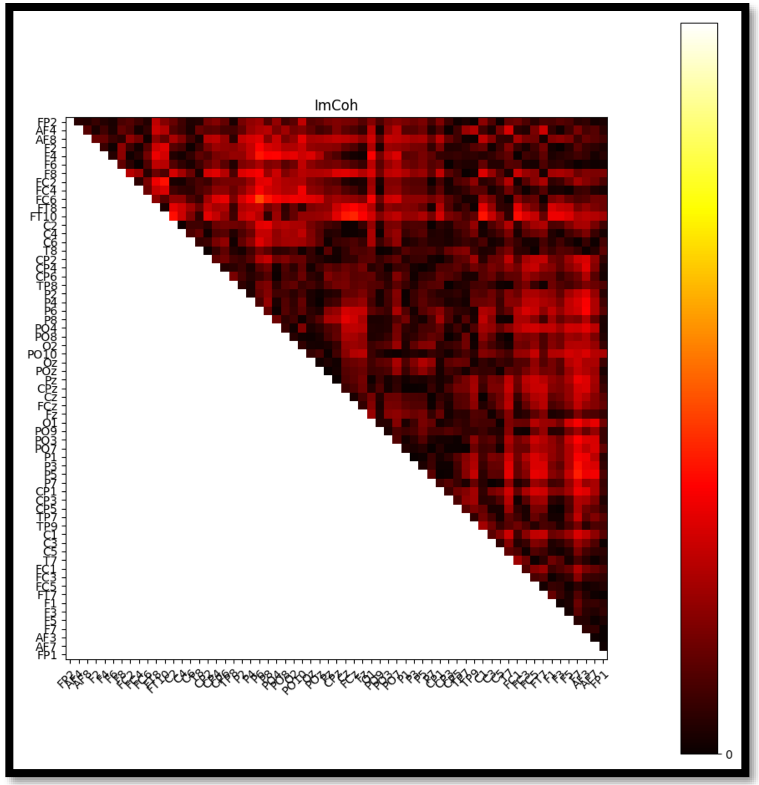
  ➔ external matrices analysis/plotting





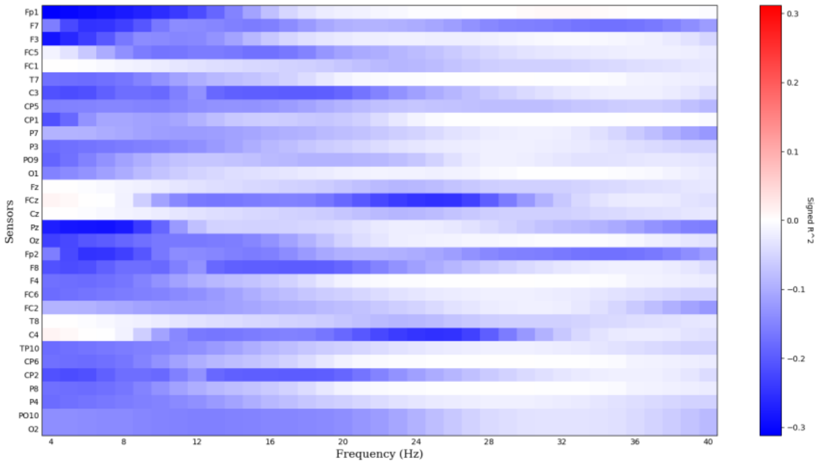
```
#!/usr/bin/env python

import numpy as np
import numpy.ma as ma
import matplotlib
matplotlib.use('Qt5Agg')
from matplotlib import pyplot as plt
from pylsl import StreamInlet, resolve_stream, resolve_byprop
from scipy import signal
import time
import seaborn as sns
import mne


def main():

    plotMatrices = True
    plotConnectomes = False
    threshold = 0.33
    thresholdPercent = 5
    thresholdType = 'percent' # 'absolute' or 'percent'
    thresholdedMatrix = 'MSC' # 'ImCoh' or 'MSC'

    nbChannels = 63
    electrodes = ['FP2','AF4','AF8','F2','F4','F6','F8','FC2','FC4','FC6','FT8','FT1

    # --- LSL stream
    print("looking for a stream...")
    streams = resolve_byprop("type", "signal")

    for strIdx in range(0, len(streams)):
        print("---STREAM ", strIdx)
        print("info.type() ", streams[strIdx].type())
        print("info.name() ", streams[strIdx].name())
        print("info.nominal_srate() ", streams[strIdx].nominal_srate())
        print("info.channel_format() ", streams[strIdx].channel_format())
        if streams[strIdx].name() == "openvibeConnectMSC":
            inlet_msc = StreamInlet(streams[strIdx])
        if streams[strIdx].name() == "openvibeConnectImCoh":
```

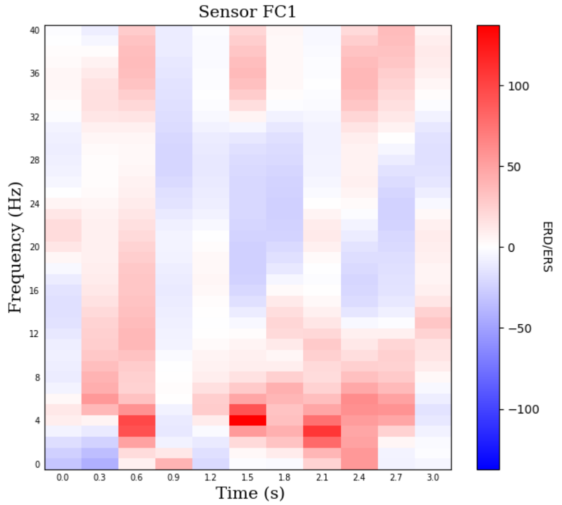




ONGOING PROJECT: BCI PIPELINE AUTOMATION

- **Goals:**
  - **GUI for automatic generation of scenarios**, in a unified & robust pipeline framework (acquisition / feature-extraction / training / online) Scenarios & parameters automatically generated depending on user's preferences, from template scenarios
  - **GUI with data viz for feature selection**, with automatic scenario update after selection ex: R² map from Spectral power in a set of frequency bands ex: Node Strength based on connectivity


Sensors
Fp1
F7
F3
FC5
FC1
T7
C3
CP5
CP1
P7
P3
PO9
O1
Fz
FCz
Cz
Pz
Oz
Fp2
F8
F4
FC6
FC2
T8
C4
TP10
CP6
CP2
P8
P4
PO10
O2
Frequency (Hz)
Signed R^2


Sensor FC1
Frequency (Hz)
Time (s)
ERD/ERS