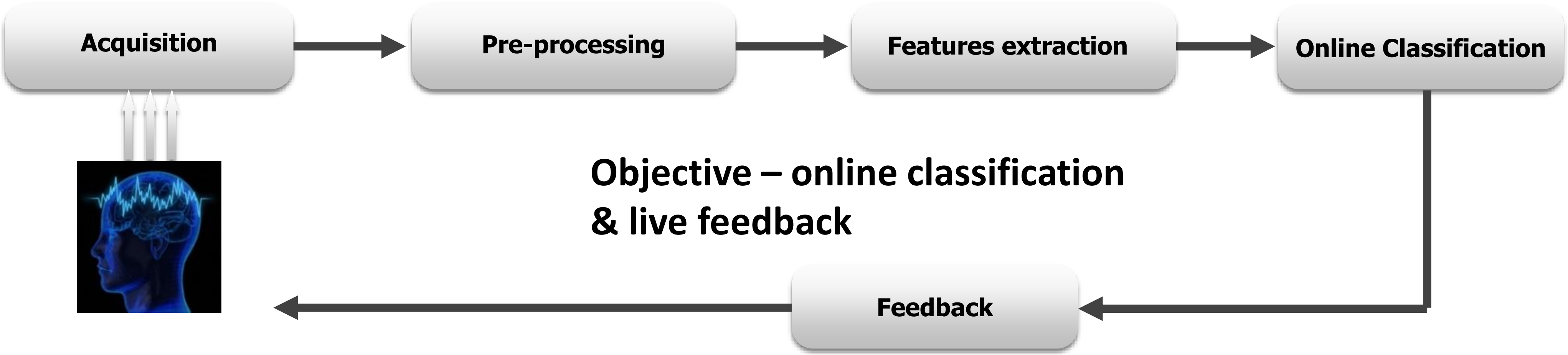
BCI Meeting - June 7th 2023

HappyFeat: an interactive and efficient BCI framework for clinical applications

Arthur Desbois,
Inria Paris, ARAMIS team, Paris Brain Institute (ICM)

What is “BCI”?



Acquisition

Pre-processing

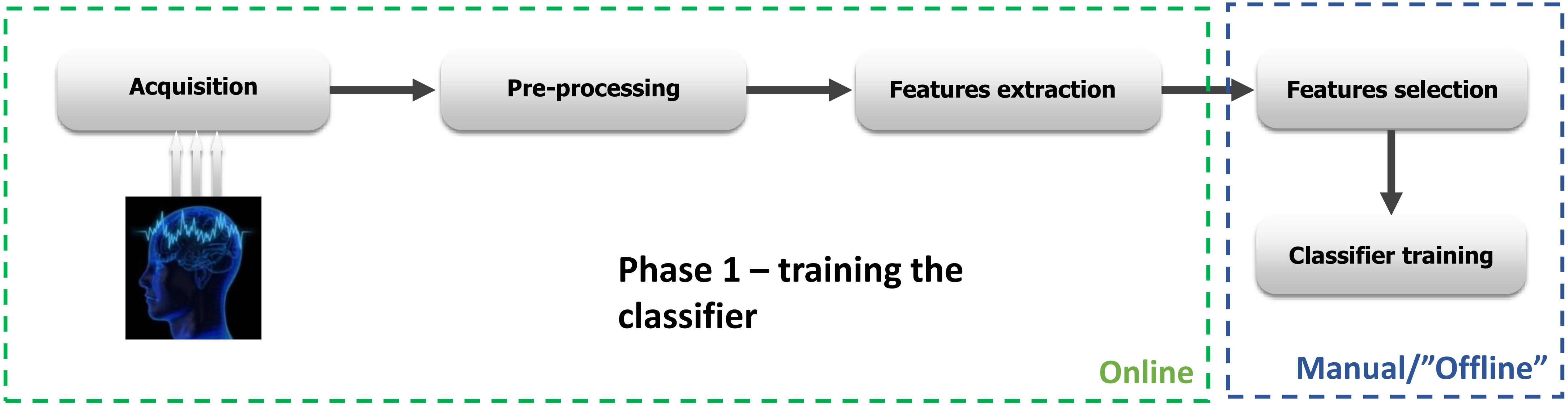
Features extraction

Online Classification

Feedback

Objective – online classification & live feedback

What is "BCI"?



Acquisition

Pre-processing

Features extraction

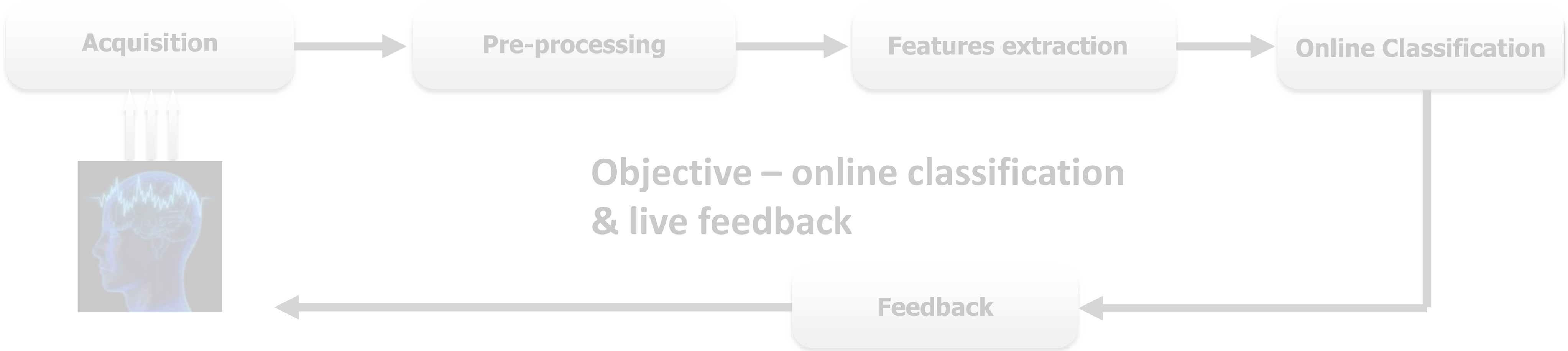
Features selection

Classifier training

Phase 1 – training the classifier

Online

Manual/"Offline"



Acquisition

Pre-processing

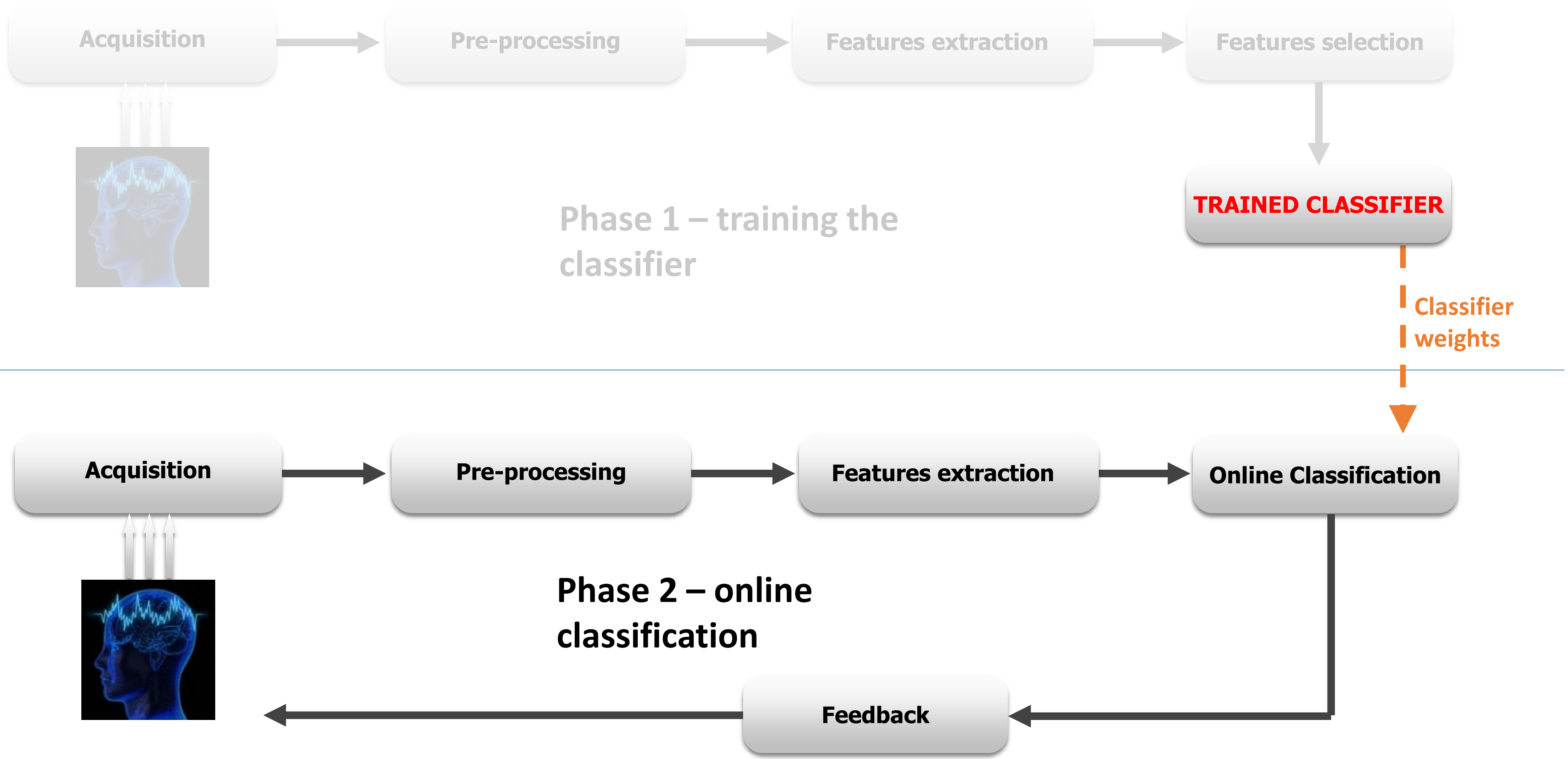
Features extraction

Online Classification

Objective – online classification & live feedback

Feedback

What is “BCI”?



Acquisition

Pre-processing

Features extraction

Features selection

TRAINED CLASSIFIER

Phase 1 – training the classifier

Classifier weights

Acquisition

Pre-processing

Features extraction

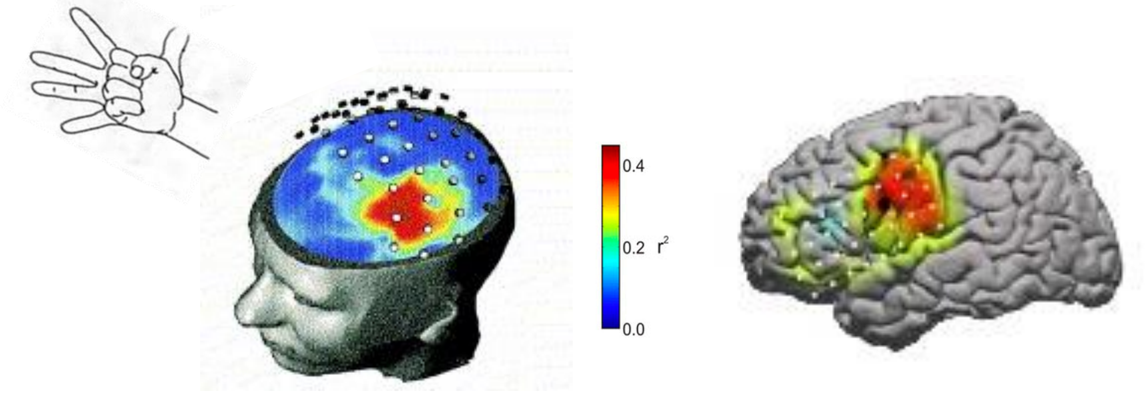
Online Classification

Phase 2 – online classification

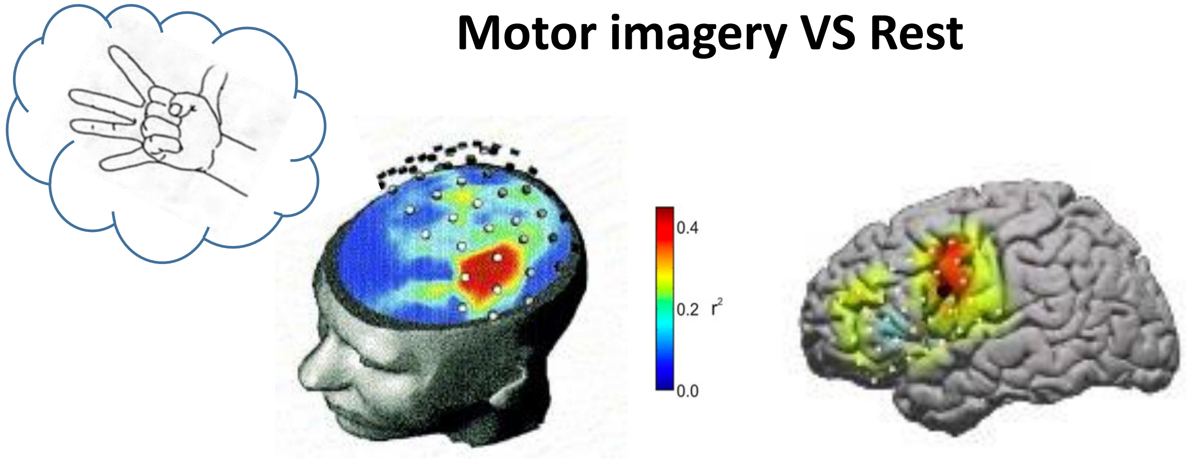
Feedback

Motor Imagery - Observations

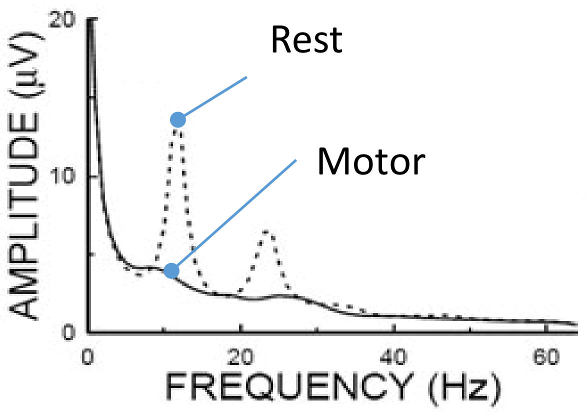
Motor execution VS Rest



Motor imagery VS Rest



Spectral Power Decrease

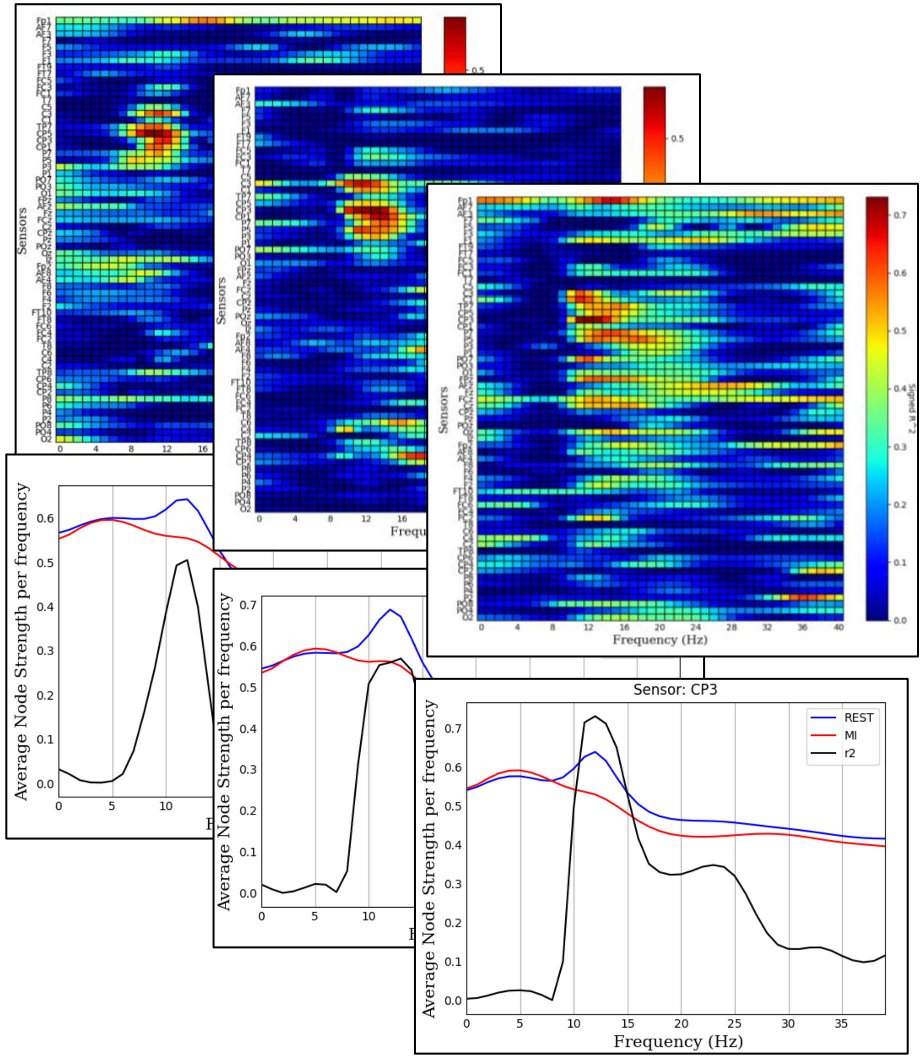


Desynchronization effect
(Pfurtscheller et al, 1999)

BCI in clinical settings - Feature selection

- ## Features of interest (FOI)
  - Selecting adequate FOIs is a **crucial step for BCI performance**.
  - After EEG signals acquisition, an **analysis phase** is needed to select best FOIs.
    - ➔ Scientific softwares (i.e., MATLAB)
    - ➔ Manual step, expertise needed
  - If this analysis phase is too long, a lot can change in the meantime:
    - EEG sensors impedances
    - Subject’s brain behaviors
    - Subject’s attention & motivation

➔ Signal characteristics might be very different between Acq/Training phase and Online phase…

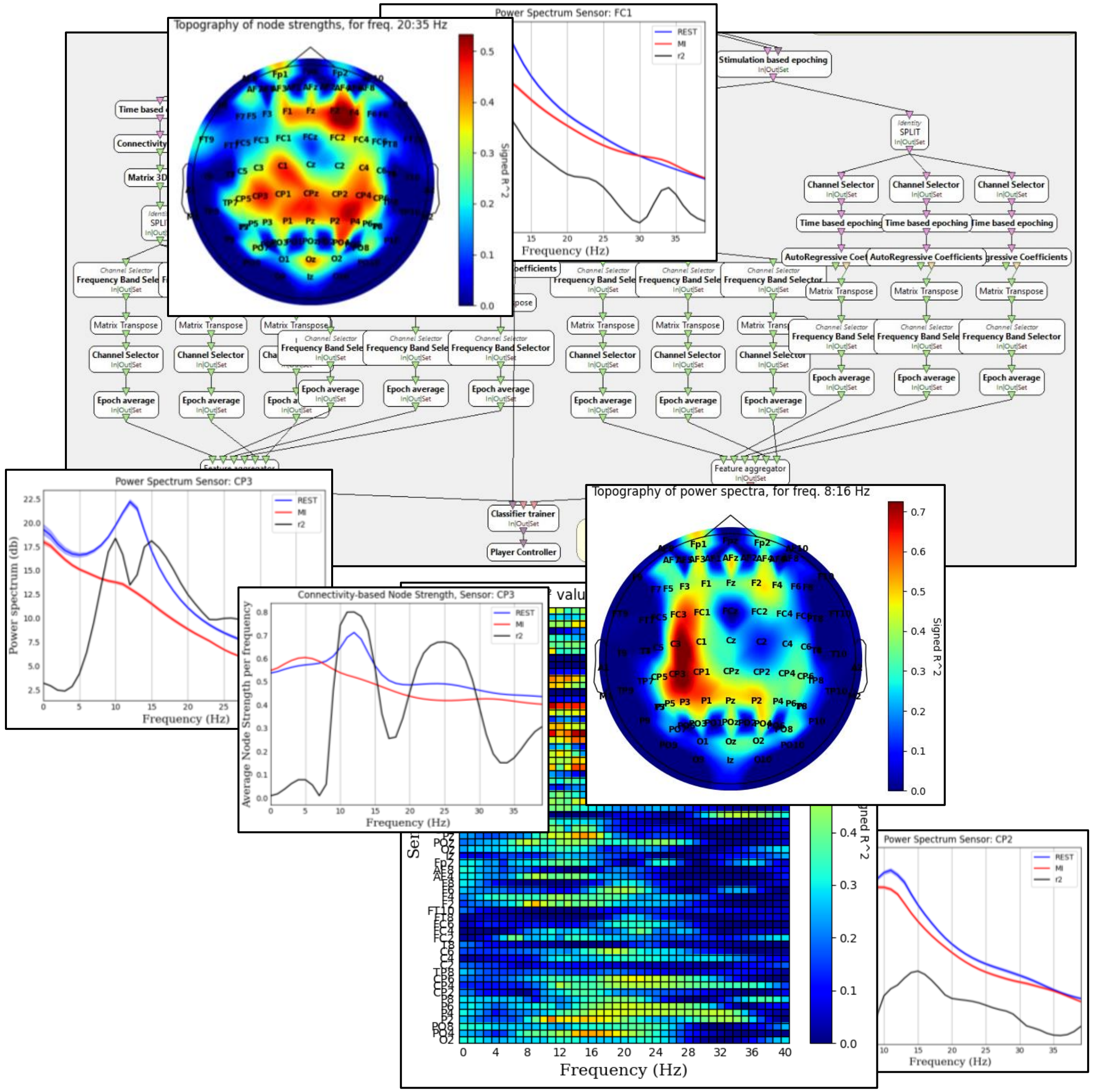


BCI in clinical settings - Feature selection

The “analysis phase” involves many manipulations. An example:

- Setting up “**feature extraction**” scenarios in OpenViBE…
- Finding FOIs through **visualization**…
- Setting up & running **training** scenarios in OpenViBE…
- … and maybe **going again** through those steps multiple times until “correct” features have been found, or to account for inter-run variability

➔ **Tedious, error-prone,**
hard to achieve in a limited time


Topography of node strengths, for freq. 20:35 Hz
Power Spectrum Sensor: FC1
REST
MI
r2
Signed R^2
Frequency (Hz)
Stimulation based epoching
SPLIT
Channel Selector
Time based epoching
AutoRegressive Coefficients
Frequency Band Selector
Matrix Transpose
Epoch average
Feature aggregator
Classifier trainer
Player Controller
Power Spectrum Sensor: CP3
Power spectrum (db)
Connectivity-based Node Strength, Sensor: CP3
Average Node Strength per frequency
Topography of power spectra, for freq. 8:16 Hz
Power Spectrum Sensor: CP2

HappyFeat - Main Concepts

Python-based framework for facilitating MI pipelines

Main focus:

making Feature Selection

+ Classif. Training phases

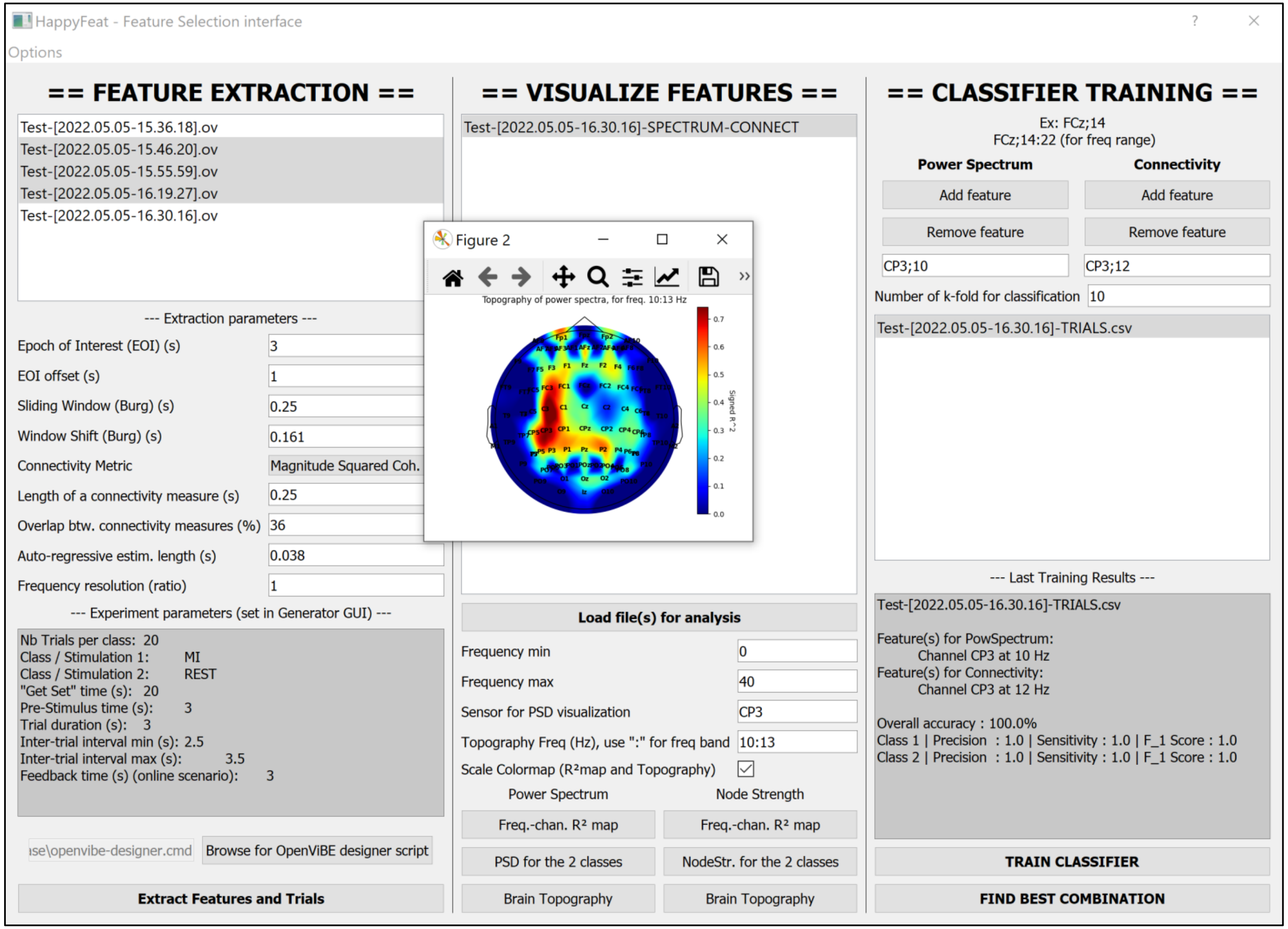
easy & fast

Analyze your data,

select your Features

& train your classifier

in less than 5 minutes!



== FEATURE EXTRACTION ==

Test-[2022.05.05-15.36.18].ov
Test-[2022.05.05-15.46.20].ov
Test-[2022.05.05-15.55.59].ov
Test-[2022.05.05-16.19.27].ov
Test-[2022.05.05-16.30.16].ov

--- Extraction parameters ---

Epoch of Interest (EOI) (s)	3
EOI offset (s)	1
Sliding Window (Burg) (s)	0.25
Window Shift (Burg) (s)	0.161
Connectivity Metric	Magnitude Squared Coh.
Length of a connectivity measure (s)	0.25
Overlap btw. connectivity measures (%)	36
Auto-regressive estim. length (s)	0.038
Frequency resolution (ratio)	1

--- Experiment parameters (set in Generator GUI) ---

```
Nb Trials per class:  20
Class / Stimulation 1:        MI
Class / Stimulation 2:        REST
"Get Set" time (s):  20
Pre-Stimulus time (s):       3
Trial duration (s):    3
Inter-trial interval min (s): 2.5
Inter-trial interval max (s):          3.5
Feedback time (s) (online scenario):       3
```

ıse\openvibe-designer.cmd

== VISUALIZE FEATURES ==

Test-[2022.05.05-16.30.16]-SPECTRUM-CONNECT

Frequency min	0
Frequency max	40
Sensor for PSD visualization	CP3
Topography Freq (Hz), use ":" for freq band	10:13
Scale Colormap (R²map and Topography)	☑

== CLASSIFIER TRAINING ==

Ex: FCz;14
FCz;14:22 (for freq range)

Power Spectrum	Connectivity
CP3;10	CP3;12

Number of k-fold for classification 10

Test-[2022.05.05-16.30.16]-TRIALS.csv

--- Last Training Results ---

```
Test-[2022.05.05-16.30.16]-TRIALS.csv

Feature(s) for PowSpectrum:
        Channel CP3 at 10 Hz
Feature(s) for Connectivity:
        Channel CP3 at 12 Hz

Overall accuracy : 100.0%
Class 1 | Precision  : 1.0 | Sensitivity : 1.0 | F_1 Score : 1.0
Class 2 | Precision  : 1.0 | Sensitivity : 1.0 | F_1 Score : 1.0
```

HappyFeat - Main Concepts

- ## Key feats & mechanisms:
  - **Clean, risk-free environment**

    ➔ avoid unnecessary & error-prone manipulations.
  - **Trial-and-error oriented workflow**

    ➔ all steps can be repeated quickly & as many times as needed
  - **Unified "dashboard" GUI**
  - **OpenViBE** used in the background, as a fast & efficient processing engine.

    ➔ no scenario edition/manipulation necessary: everything is **automated**!
- ## Two main use cases:
  - **Make BCI pipelines smoother/easier to use** and allow **reproducibility** of exps.
  - **Prospective works & comparison of alternative features of interest** (connectivity, networks…)

HappyFeat - Main Concepts

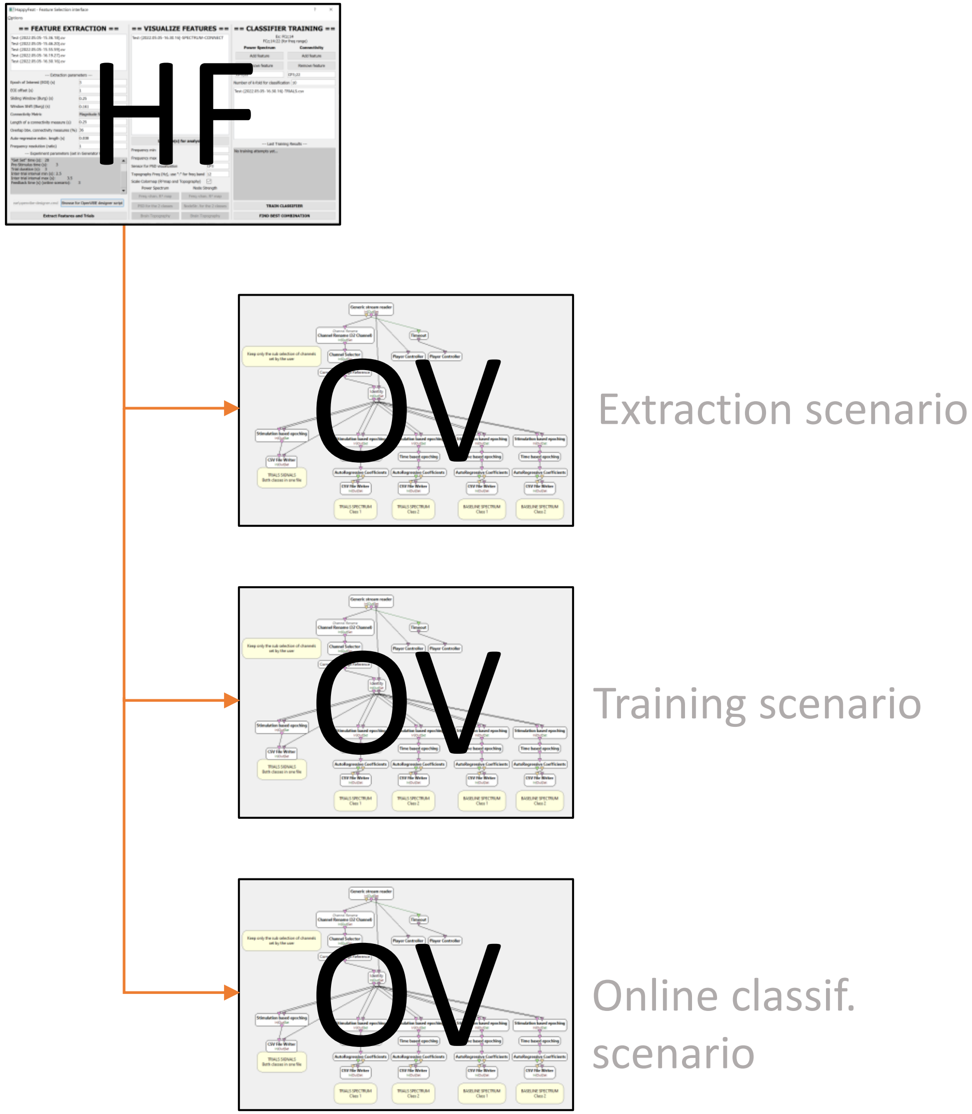
- **Efficient processing pipelines**
  - Available features for classification: *Spectral Power*, *Connectivity-based network metrics*
  - ... It’s also possible to train the classifier using **a fusion of both features**.
- **Feature extraction**
  - Easy access to all experiment & signal processing parameters.
  - Use **pre-recorded signals** or **on-the-fly during acquisition phase.**
- **Visual Analysis for feature selection**
  - R^2 maps, PSD comparison across trials, time/freq. ERD/ERS analysis, brain topography…
- **Classifier training**
  - **Run as many training attempts as needed**, using different features, in only a few clicks.
  - Concatenate trials from multiple recorded sessions
  - OpenViBE scenarios are **updated and launched on-the-fly**.
  - Automatically generates/updates the “**online classification**” **scenario**.

HappyFeat - How?

1. **MI Pipeline / “Feature type” selection**

- Selecting btw. multiple “template” scenarios depending on the use case (power spectrum, connectivity type…)

- Edit basic/common parameters (acquisition, extraction, training…)



Extraction scenario

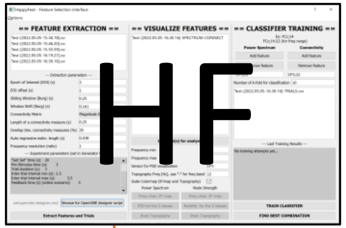
Training scenario

Online classif. scenario

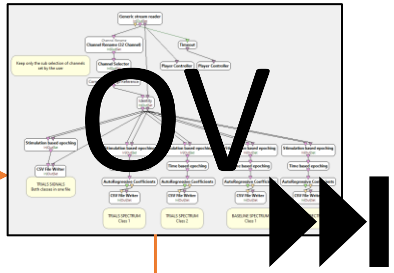
HappyFeat - How?

1. MI Pipeline / “Feature type” selection
2. **Feature Extraction**

- **Select signal files**, and **extraction parameters** (lengths and overlap of windows, FFT size…)
- **Run** the generated extraction OpenViBE scenario (**in the background**) for all selected signal files:
  - ➔ **Extract metadata** (sampling freq, electrodes…)
  - ➔ Cut the signal to **regions of interest** (MI trials & baseline portions), **generate CSV file** with only these chunks (for the training step)
  - ➔ Apply a **signal processing** pipeline (PSD computation, connectivity measure…) to the signal chunks of interest, **generate CSV files for future analysis**
- **Runs in an autonomous thread:** You can do visualizations and training attempts for signals already processed in the meantime.



Extraction scenario



Signal 1 (.ov)

- Spectral Power (class 1) CSV
- Spectral Power (class 2) CSV
- Trials CSV
- Metadata CSV

Signal 2 (.ov)

- Spectral Power (class 1) CSV
- Spectral Power (class 2) CSV
- Trials CSV
- Metadata CSV

…

HappyFeat - How?

1. MI Pipeline / “Feature type” selection
2. Feature Extraction
3. **Analysis, Feature Selection**

- Select one or multiple signals & load their spectral/connectivity data (CSV work files generated during “Feature Extraction”)

- Use different **Visualization Tools** to help find & select **features of interest** (FOIs) for training

  ➔ Frequency/channel R² map

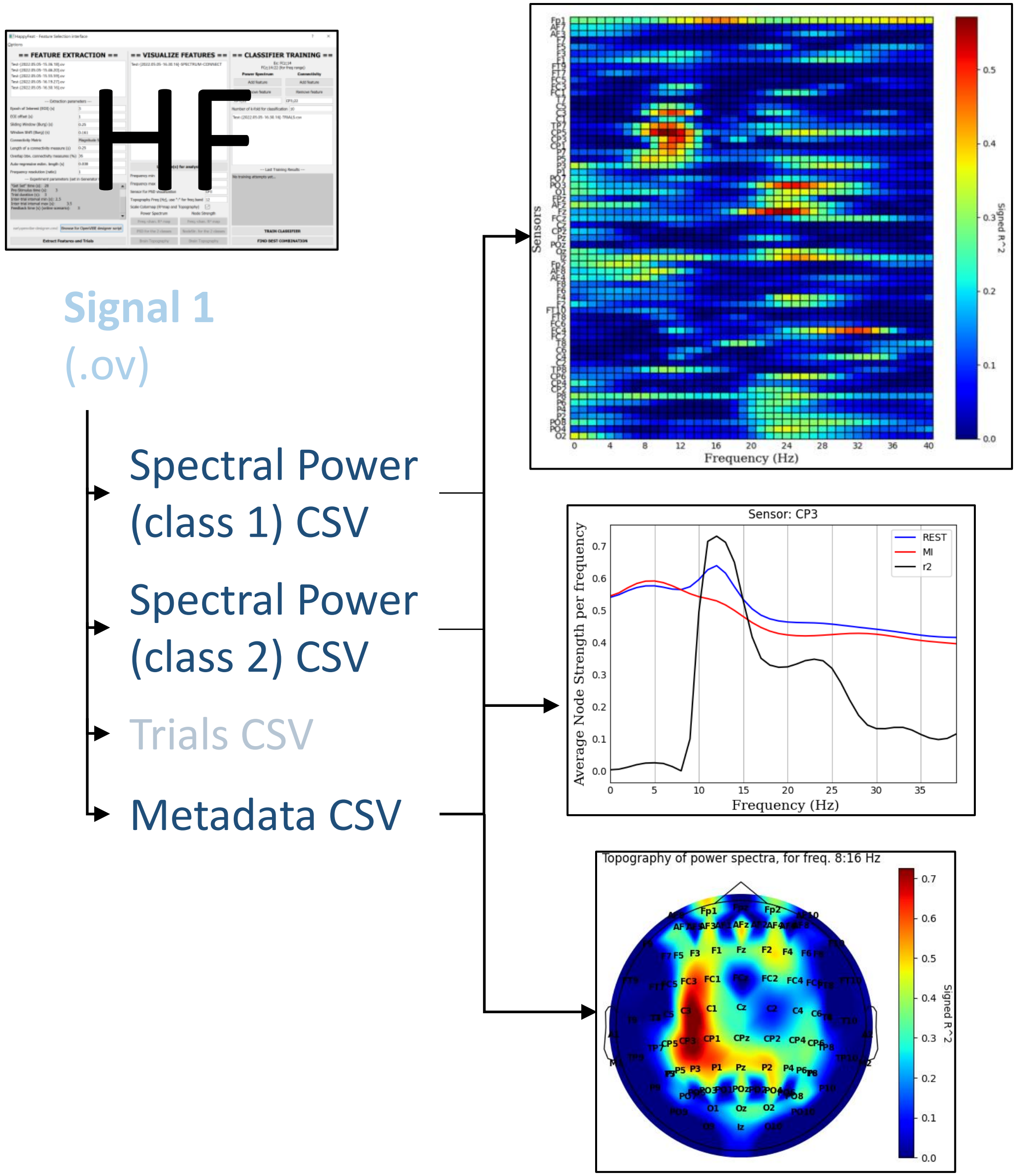
  ➔ PSD (or connect. metric) comparison btw. the 2 conditions (MI/REST) for a given electrode

  ➔ Time/frequency ERD/ERS analysis for each condition

  ➔ R² mapped as a brain topography for a given frequency (or range)

Combine as many visualization windows as you need

A “Dual metric” pipeline allows to show (for ex.) R² maps for both Power Spectrum and Connectivity in parallel


Signal 1 (.ov)

Spectral Power (class 1) CSV

Spectral Power (class 2) CSV

Trials CSV

Metadata CSV

HappyFeat - How?

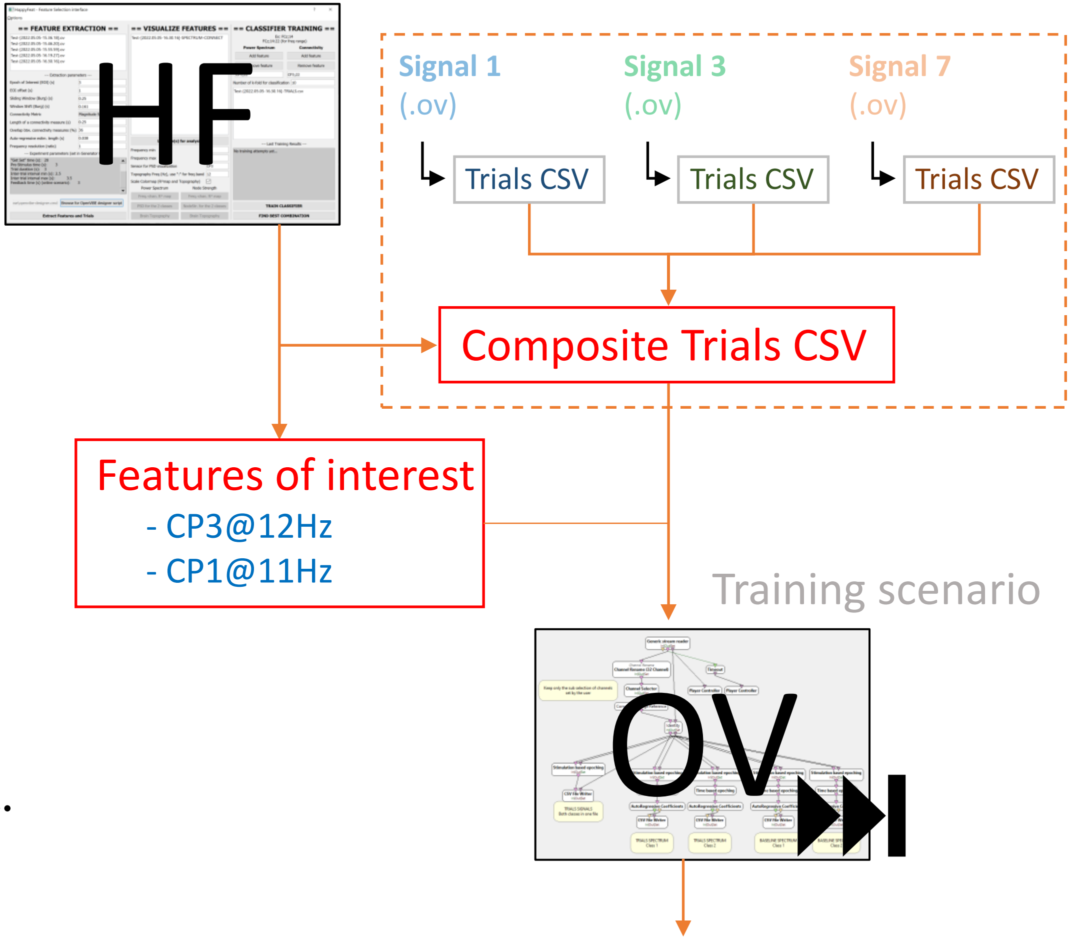
1. MI Pipeline / “Feature type” selection
2. Feature Extraction
3. Analysis, Feature Selection
4. **Classifier Training**

- Set one or more **Features of Interest** (Electrode/Freq.)
- **Select file(s)** with which you want to train your classifier
  If > 1 file : their trials **are automatically concatenated**
- **Run the Classifier Training scenario**
  (Auto. generated in step 1 + auto. edited with FOIs)
  ➔ Classification **ACCURACY** + **WEIGHTS**

Disappointing results? (“My accuracy is 50% ☹”)
Maybe try again with other features. **It only takes a few seconds…**

Satisfying results?! (“OMG 95%”)
Good news! The “Online Classification” scenario has already been automatically been updated with:

- Classifier training weights
- Features of interest used for training



Training accuracy “**score**”
+ Classifier **Weights**
+ **Online scenario updated**

A few technical details…

Full list of dependencies:

- **Python 3.9**
  - shutils>=0.1.0
  - mne>=0.23.0
  - numpy>=1.21.1
  - pandas>=1.3.1
  - PyQt5>=5.15.6
  - statsmodels>=0.13.1
  - scipy>=1.7.1
  - spectrum>=0.8.0
  - matplotlib>=3.4.2
- **OpenViBE v3.5.0**

Current limitations & future works

- **Current limitations**
  - Low flexibility regarding electrode schemes. *(work in progress…)*
  - Only one type of classification algo. proposed (LDA) *(work in progress…)*

  - Pipelines are “fixed”:

    − trading OpenViBE’s high level of flexibility…
    + …for a high comfort of pipeline settings customization
    + & “trial-and-error” workflow
  - Only three types of pipelines/feature types:
    - Power Spectrum Density
    - Connectivity (coherence & its variants)
    - Dual (mixing PSD & Connectivity)

➔ In project (*long term!*) for more “**prospective power**”:

allow the user to choose btw. 1 and 3 feature-types & network metrics to mix as they see fit (MSC/node strength + Imag(Coh)/Laterality + …)

Current limitations & future works

- ## Current limitations
  - Every time a new feature type is selected, or extraction parameters are modified… all work files need recomputating from scratch.

    ➔ Necessity for a robust “work session” save/load mechanism

  - In project: fully autonomous 100% Python version, without OpenViBE
    - — No acquisition/online possibilities
    - — Obviously slower… (no C++ optimizations!)
    - + More portable, all types of platforms supported (MacOS!)
    - + Other (more flexible) formats for I/O and work files (CSV, EDF…)

HappyFeat - Conclusion

- Already available online, work-in-progress version:

  https://github.com/Inria-NERV/happyFeat

- To be continued…
  - **More flexibility** (pipeline options, mixing metrics…)
  - **More network metrics** (based on connectivity)
    - **… and associated visualization tools**
  - **More options for classification algorithms**
  - **Workspace/session manager** to save/load session settings
  - Fully autonomous Python version, for cross-platform usage

Stay tuned!

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

Thanks for your attention! Any questions?