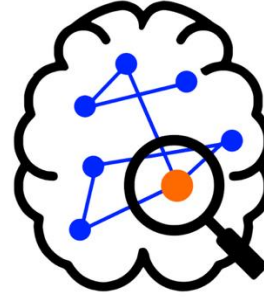


Inria



BCI Meeting 2025 - WS19 - June 5

**HappyFeat: an interactive and efficient
BCI framework for clinical applications**

Arthur Desbois

Inria Paris, NERV team, ICM (Paris Brain Institute)

1. Scientific & technical context

- BCI loop, experiments
- Why HappyFeat?

2. HappyFeat

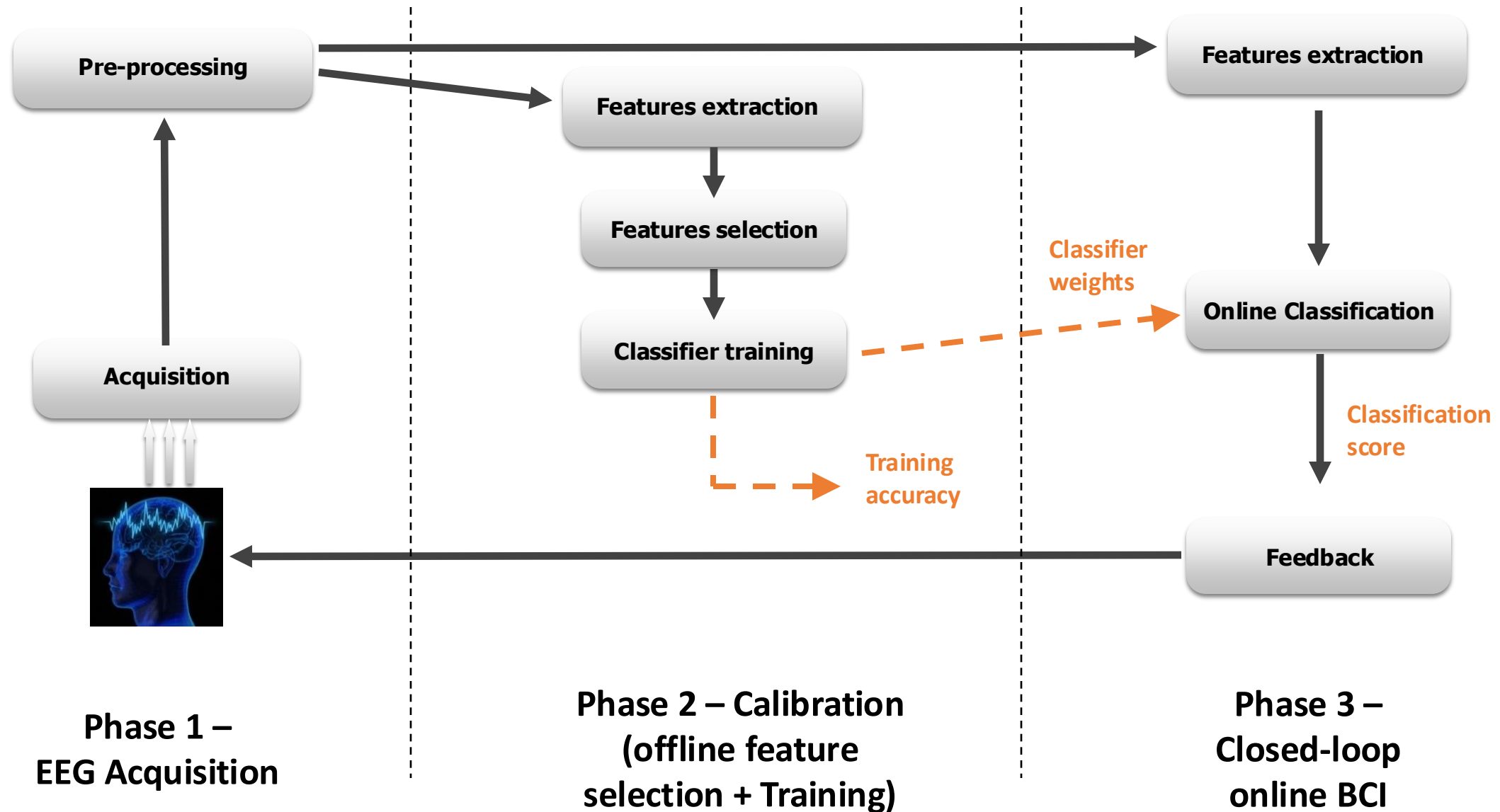
- Key mechanisms & features
- Usage in BCI experimental & clinical protocols

3. Wrapping up...

- Current technical status

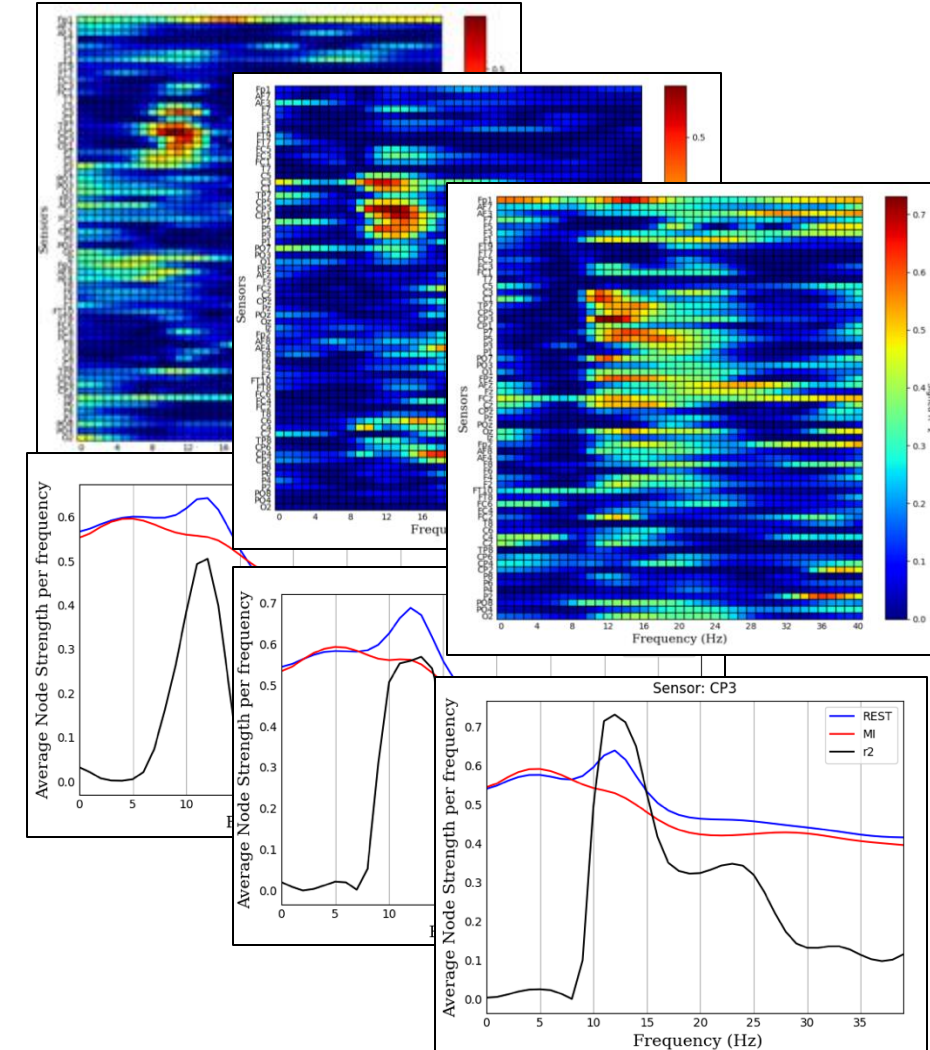
1. Scientific & technical context

- BCI loop, experiments
- Why HappyFeat?



- « **Features of Interest** » (FOI)

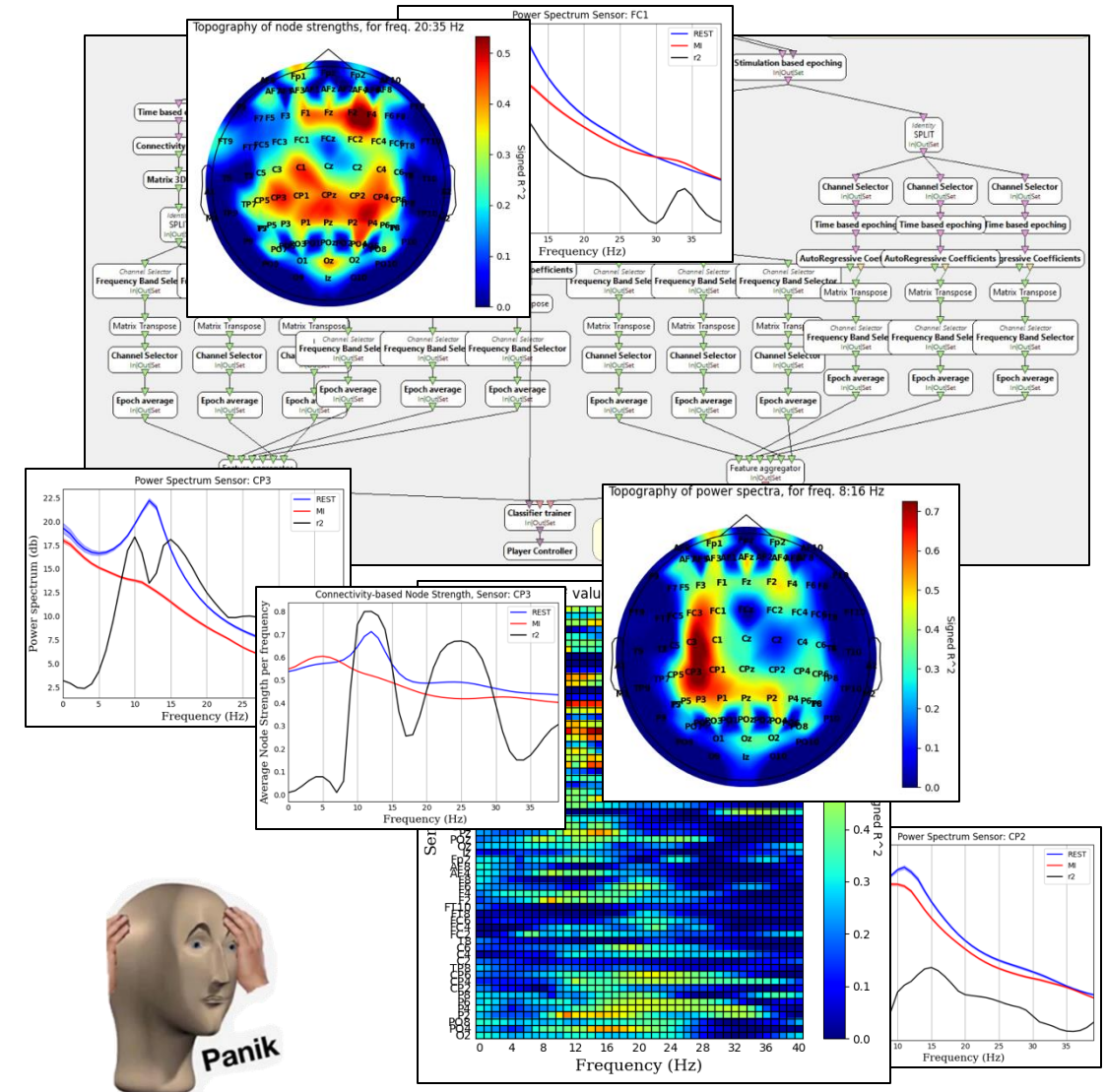
- Selecting adequate FOIs is a **crucial step** for BCI performance.
- After EEG signals acquisition, an **analysis phase** is needed to select best FOIs.
 - ➔ Scientific softwares (i.e., MATLAB)
 - ➔ Manual step, expertise needed
- If this analysis phase is too long, a lot can change in the meantime:
 - EEG sensors impedances
 - Subject's brain behaviors
 - Subject's attention & motivation
- ➔ Signal characteristics might be very different between Acq/Training phase and Online phase...



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

- Setting up “feature extraction” pipelines (e.g. OpenViBE)
- Finding FOIs through visualization (e.g. with Matlab, Python, etc.)
- Setting up & running training pipelines (e.g. 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



- **Experimental / clinical context**

- BCI protocols (and more specifically Motor Imagery (MI)) :
 - Long duration
 - Highly demanding in focus / concentration
 - Subjects may be affected by cognitive disorders

⊕ Need to select adequate features in order to obtain satisfactory performances

- With the software tools available in 2020:
 - ➔ **difficult in experimental setting, unrealistic in clinical setting.**
 - ➔ **Not an option to use « new » features (eg. graph-based) in real conditions**

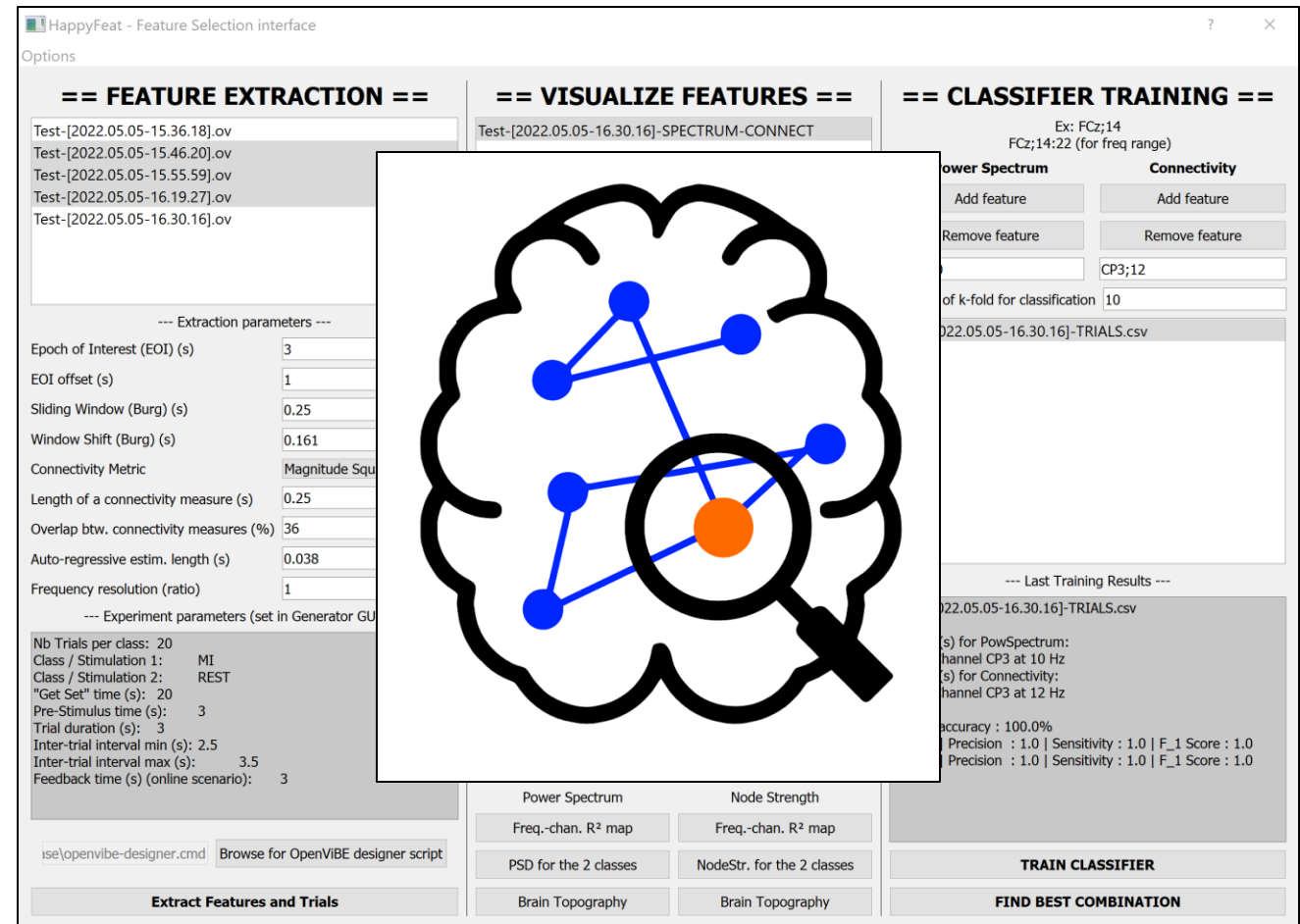
➔ **Decision :**
rationalize, simplify & automate the feature extraction/selection steps
as much as possible.

Python-based framework for facilitating ML pipelines

Main focus:

Simplify & accelerate

feature extraction,
feature selection,
classifier training



The screenshot displays the HappyFeat - Feature Selection interface, which is organized into three main sections: Feature Extraction, Visualize Features, and Classifier Training.

- Feature Extraction:** This section includes a list of test files (e.g., Test-[2022.05.05-15.36.18].ov) and a set of extraction parameters such as Epoch of Interest (EOI), EOI offset, Sliding Window, Window Shift, Connectivity Metric, Length of a connectivity measure, Overlap btw. connectivity measures, Auto-regressive estim. length, and Frequency resolution.
- Visualize Features:** This section features a central visualization of a brain network. The network is represented by a cloud-like shape containing several blue nodes connected by lines. A magnifying glass is positioned over one of the nodes, highlighting its connections. Below the visualization, there are buttons for Power Spectrum, Node Strength, Freq.-chan. R² map, PSD for the 2 classes, Brain Topography, and NodeStr. for the 2 classes.
- Classifier Training:** This section includes a list of features (e.g., FCz;14, FCz;14:22) and a table for training results. The table shows the accuracy, precision, sensitivity, and F₁ score for different feature sets. A "TRAIN CLASSIFIER" button is located at the bottom of this section.

At the bottom of the interface, there is a section for "Experiment parameters (set in Generator GUI)" which includes fields for Nb Trials per class, Class / Stimulation 1, Class / Stimulation 2, "Get Set" time, Pre-Stimulus time, Trial duration, Inter-trial interval min, Inter-trial interval max, and Feedback time. A "Browse for OpenVIBE designer script" button is also present.

1. Scientific & technical context

- BCI loop, experiments
- Why HappyFeat?

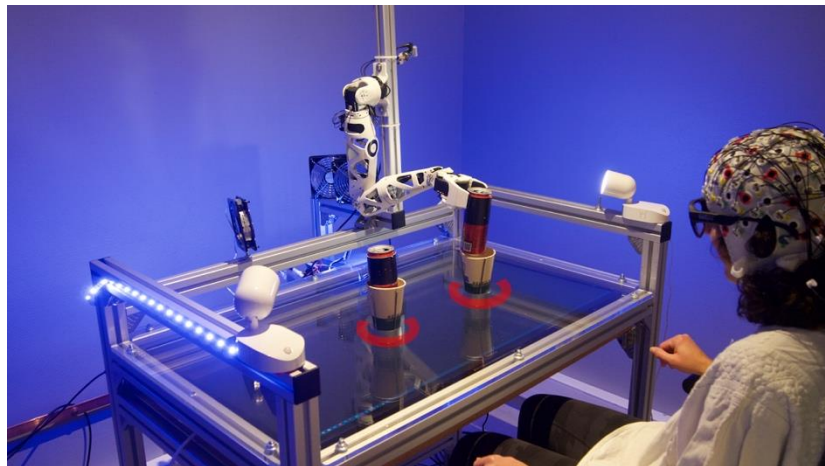
2. HappyFeat

- Key mechanisms & features
- Usage in BCI experimental & clinical protocols

- Main objectives

- « Online » :

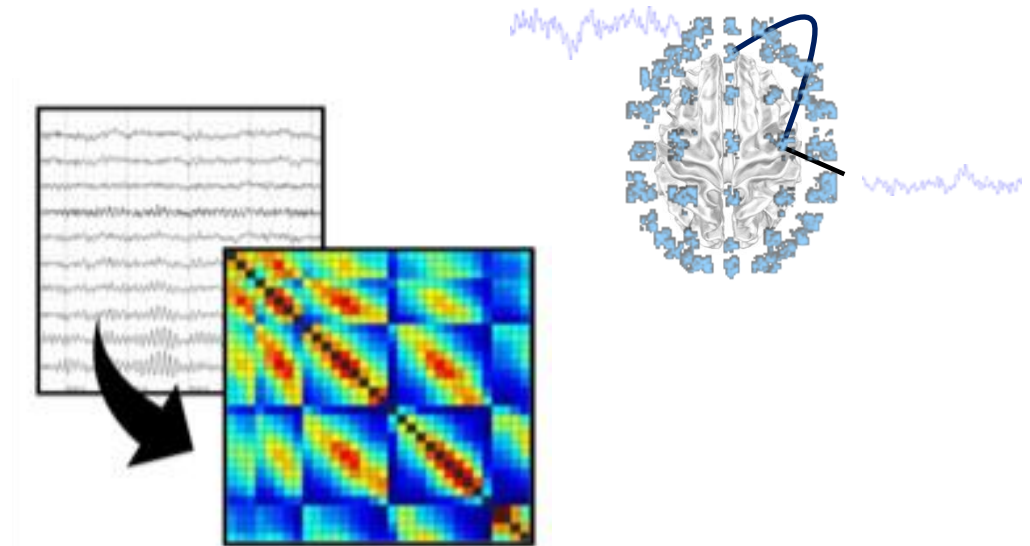
Simplify the setup and usage of BCI pipelines in experimental conditions



BRACCIO protocol – Venot et al., ICM/NERV

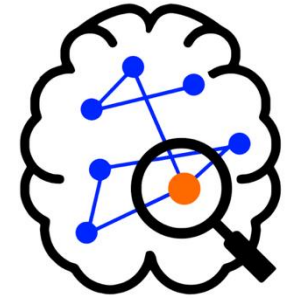
- « Offline » (... to online) :

Facilitate the integration & comparaison of new innovative metrics



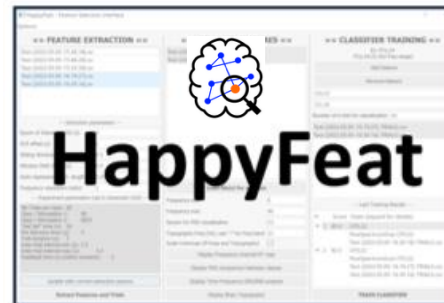
- **Key feats & mechanisms:**

- Clean, risk-free, reproducible 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 + Workspace management system
- Direct link with BCI software: OpenViBE (C++) (*and TimeFlux (Python) since v0.3.0*)
 - ➔ no scenario edition/manipulation necessary: everything is **automated!**



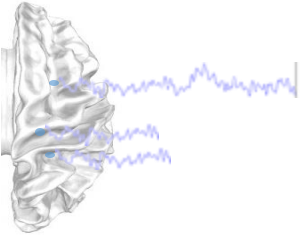
- **Two main use cases:**

- Make BCI pipelines smoother/easier to use and allow reproducibility of exps.
- Prospective works & comparison of alternative features of interest (connectivity, networks...)

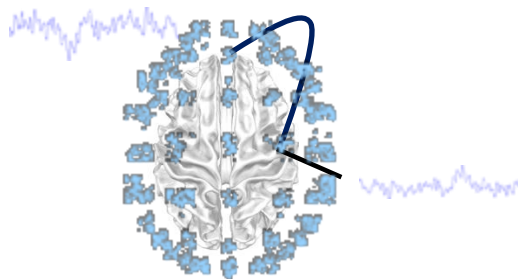


Unified, dashboard-like GUI

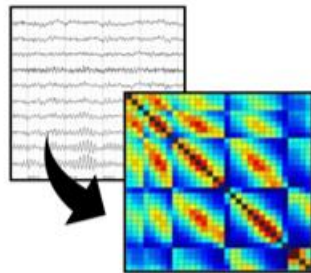
- Standard **local** measures

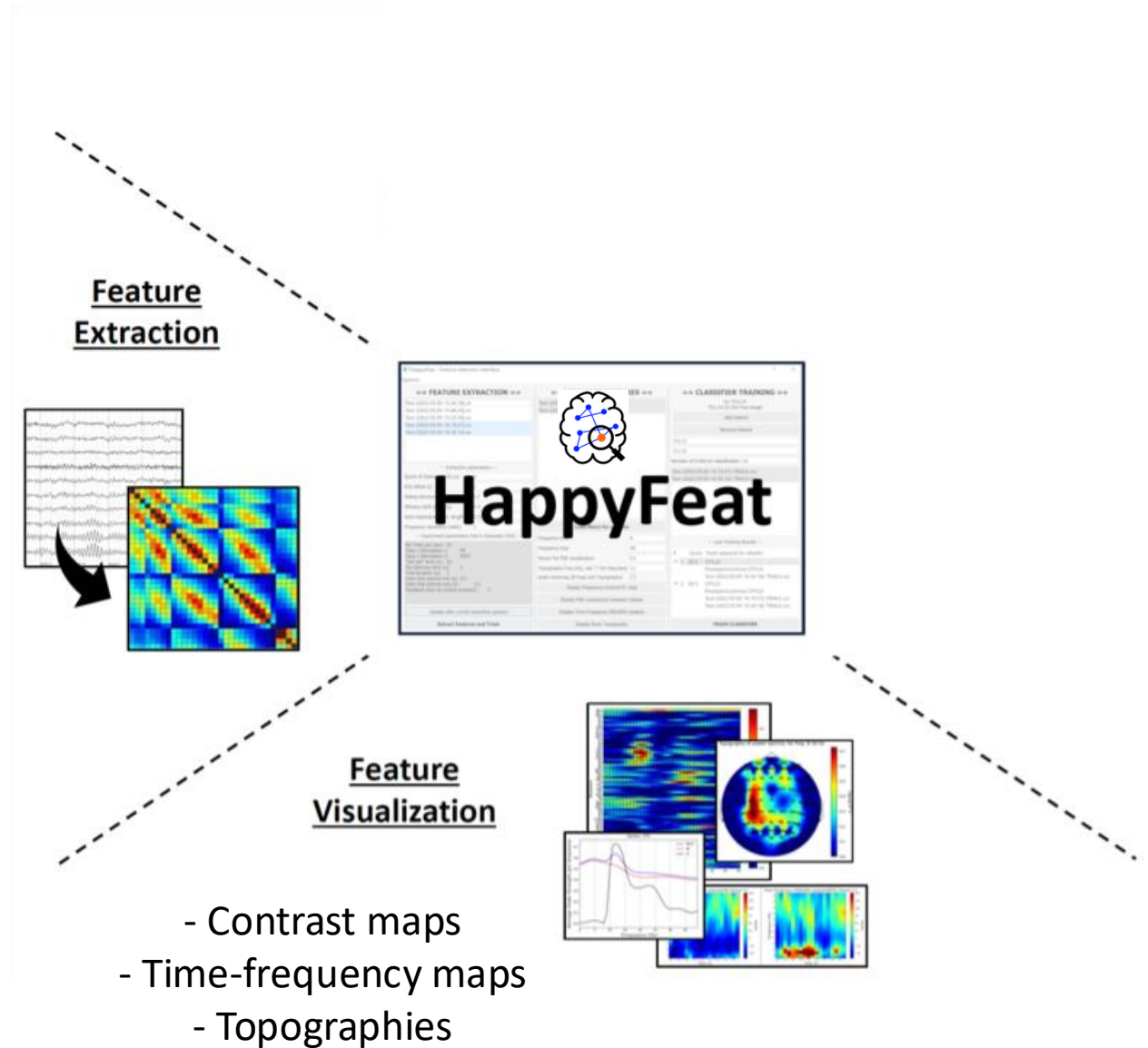


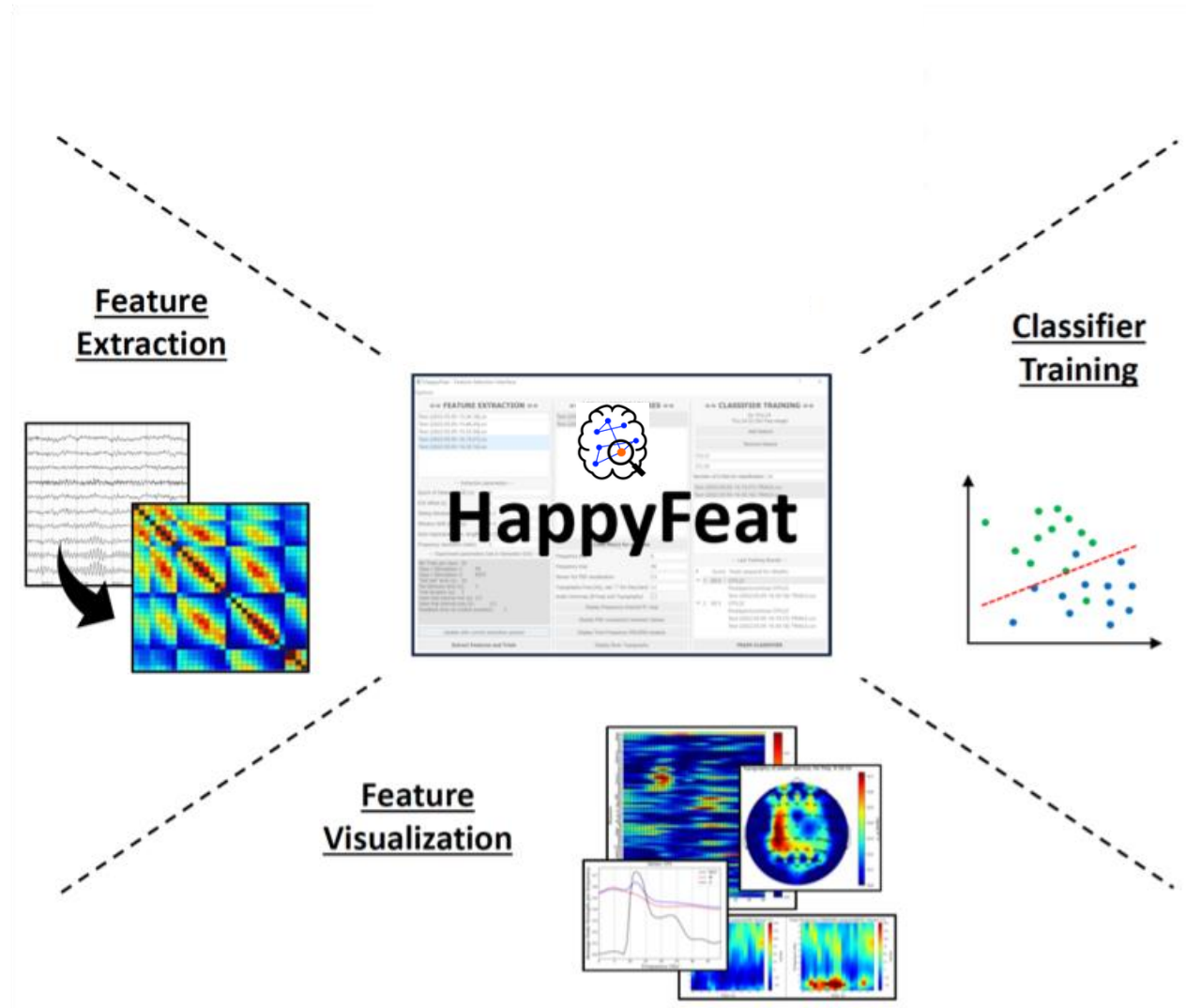
- **Network-based** approaches



Feature
Extraction

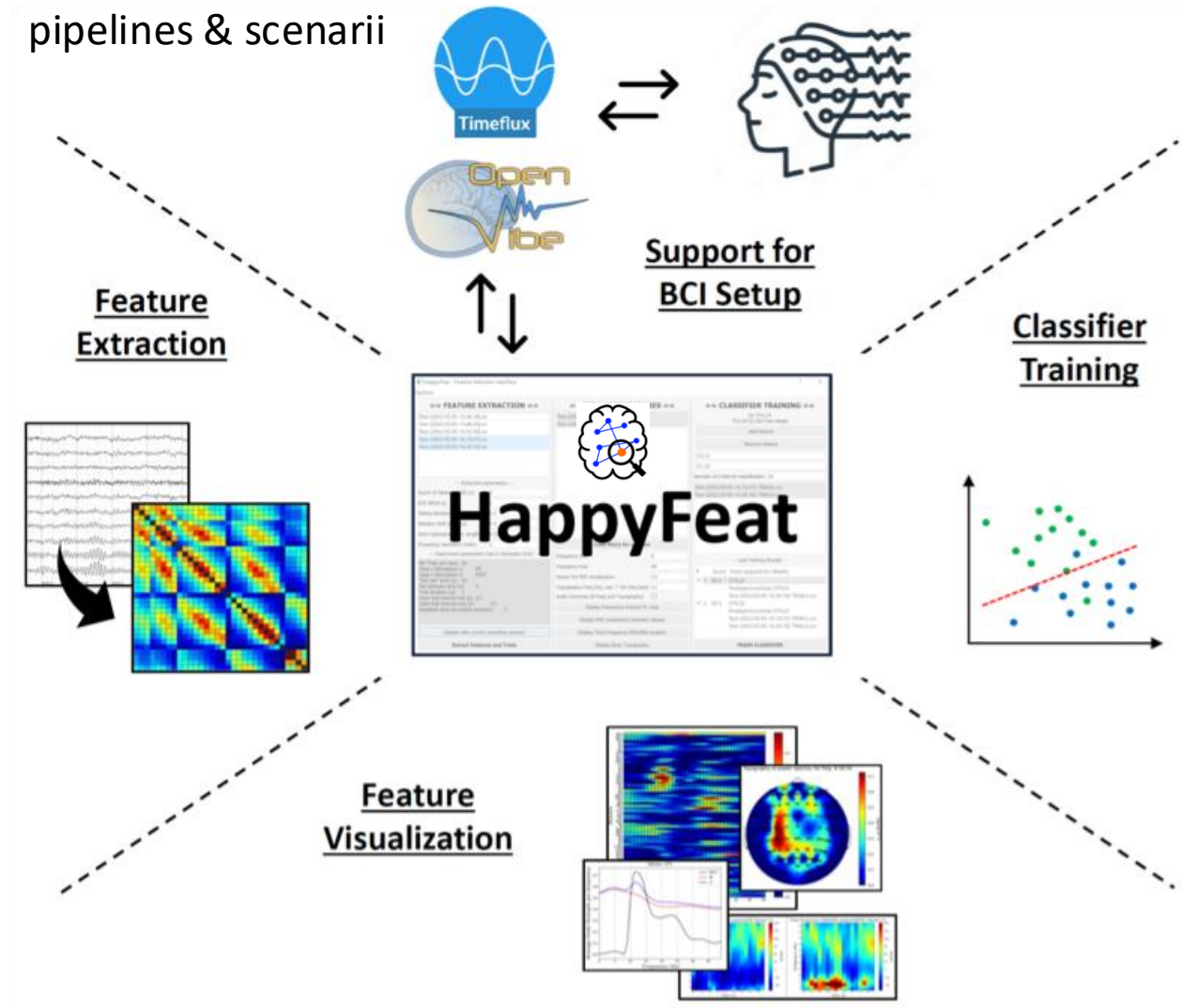


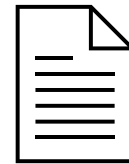
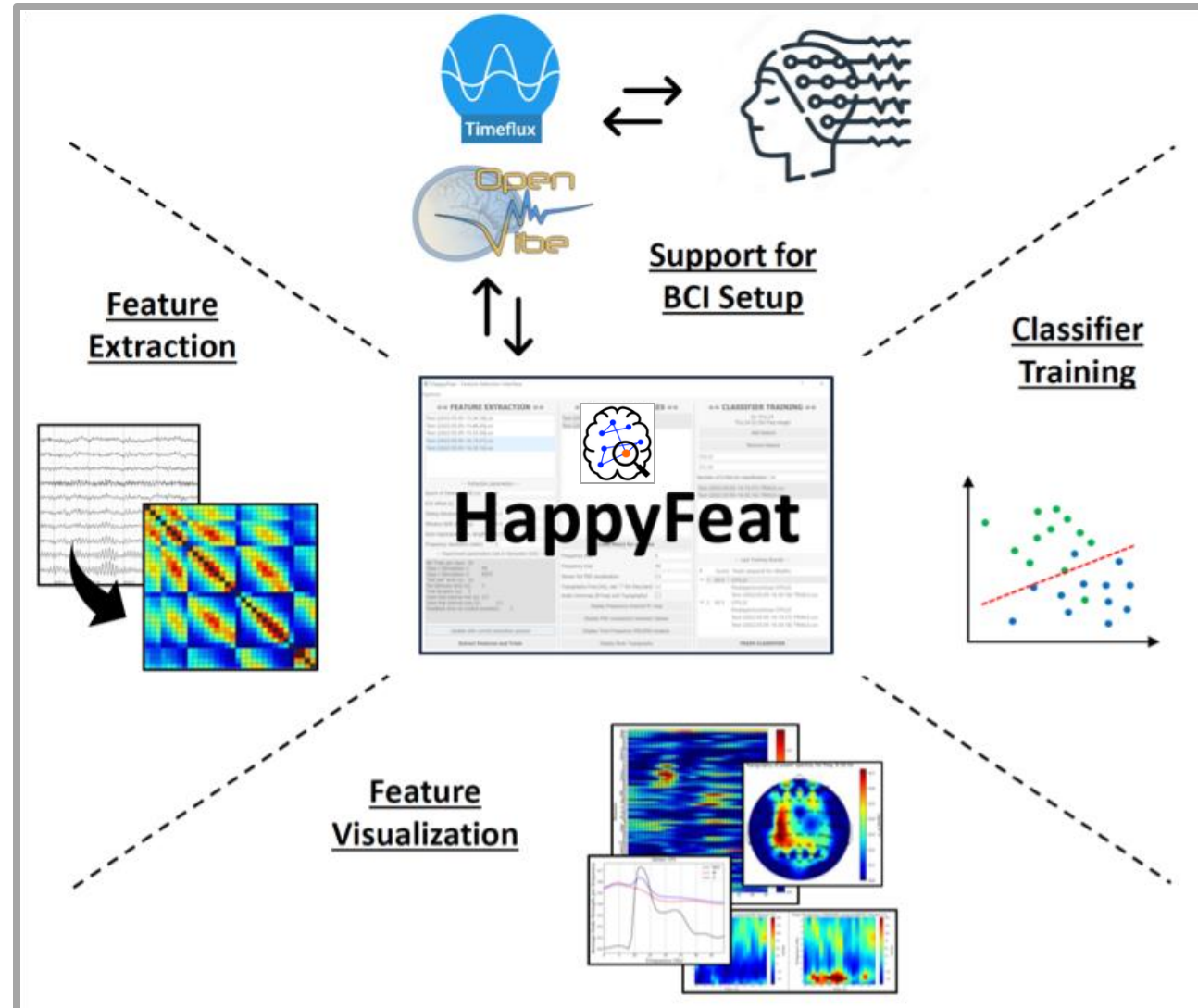




Tool for the decision

Automatic manipulation of
pipelines & scenarii

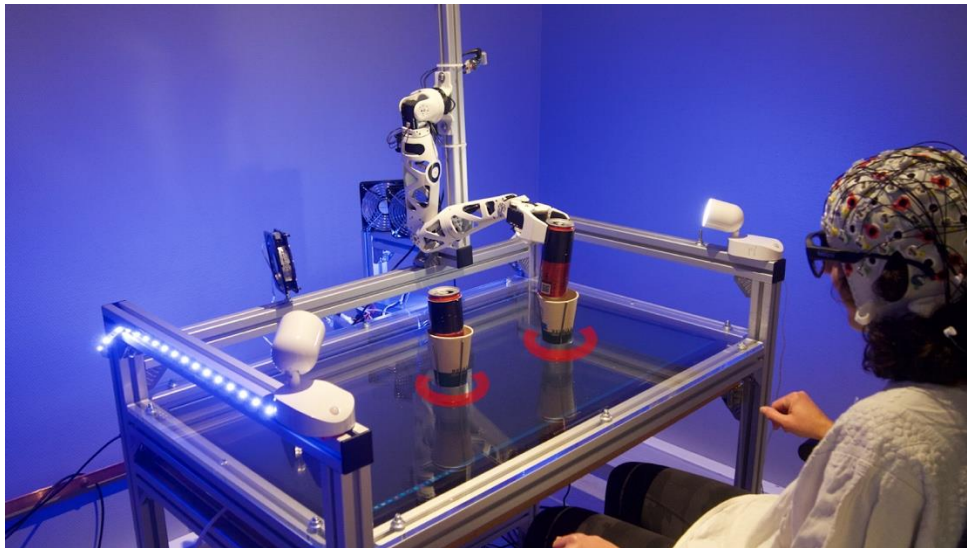




Workspace to ensure **reproducibility** and **replicability** of all the manipulations

- **2022-2024 : BRACCIO protocol (15 healthy subjects)**
 - Hybrid BCI system: **64 EEG channels + eye-tracking glasses + video table + robotic arm**
 - **Themes:** Binding, sense of agency, mixing standard and connectivity-based features

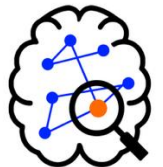
→ **Venot et al.** - *Intentional binding for noninvasive BCI control*
(Journal of Neural Engineering, July 2024 - [10.1088/1741-2552/ad628c](https://doi.org/10.1088/1741-2552/ad628c))



BRACCIO protocol – Venot et al., ICM/NERV

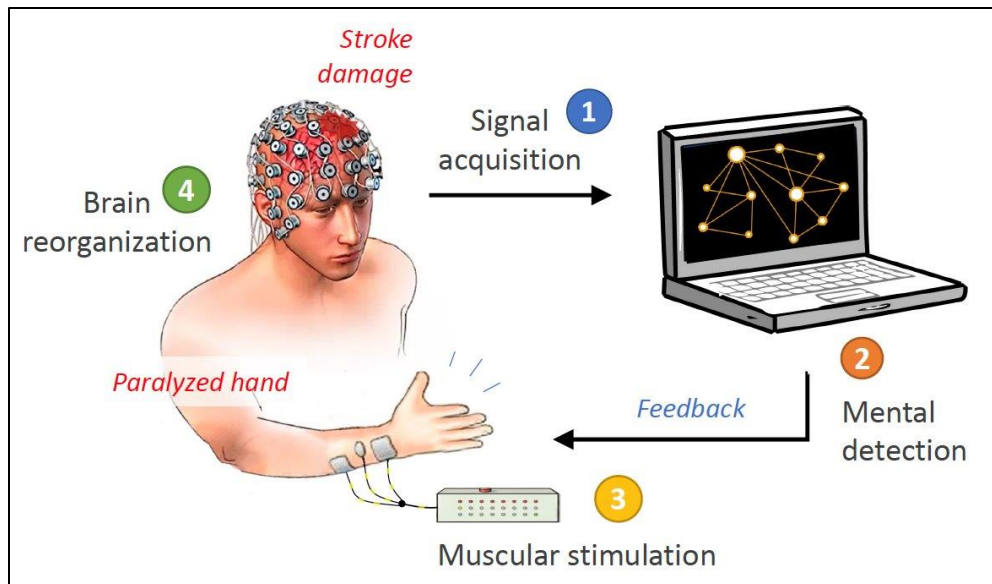
- **HappyFeat :**

- **Setup** of (OpenViBE) scenarios
(feat. extraction, visualization, training...)
- **Helping the (manual) selection** of optimal classification features during the exp. (PSD)
- **Acceleration & simplification** of the protocol



- **2025-202X : BCINET protocol**

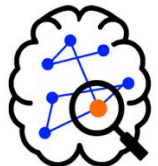
- **Post-stroke rehabilitation using BCI and rTMS.** 64 EEG channels + muscular stimulation + TMS
- **1st part:** 5 patients x 18 sessions
Feasibility, fatigue evaluation, evolution of PSD and Functional Connectivity (as $\text{imag}(\text{Coh})$)
- **2nd part:** 45 patients x 18 sessions: Evolution of BCI performance and of motor functions



Fabrizio de Vico Fallani, ERC Horizon 2020

- **HappyFeat :**

- **Setup** of (OpenViBE) scenarios (feat. extraction, visualization, training...)
- **Semi-automatic selection** of optimal classification features, using PSD and $\text{im}(\text{Coh})$
- **Actually making the protocol feasible!**



1. Scientific & technical context

- BCI loop, experiments
- Why HappyFeat?

2. HappyFeat

- Key mechanisms & features
- Usage in BCI experimental & clinical protocols

3. Wrapping up...

- **Current technical status**



GitHub repo

[https://github.com/
Inria-NERV/happyFeat](https://github.com/Inria-NERV/happyFeat)



Article:

[https://doi.org/10.1016/
j.simpa.2023.100610](https://doi.org/10.1016/j.simpa.2023.100610)



Try the tutorial!

[https://happyfeat.readthedocs.io/
en/latest/user_guide/tutorial/](https://happyfeat.readthedocs.io/en/latest/user_guide/tutorial/)

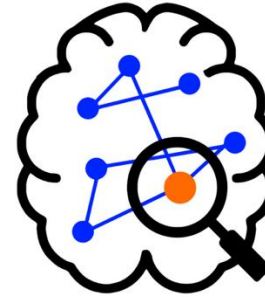


- **Version 0.3.0** (March 2025) - *patches soon, version 0.4.0 summer 2025* 🌟
- **Main dependencies :**
 - **Python 3.12.8**
 - **OpenViBE v3.6.0** or **Timeflux 0.17.2**
 - **MNE-python, PySide, Numpy, Plotly, Pandas** (full list on github...)
- **License:** Open Source BSD-3-Clause
- **PyPi package:** <https://pypi.org/project/happyfeat/>
- **Contributions** are possible (and welcome) via Github :
 - Standard, via fork + pull-requests
 - GitHub « issues » to track bugs, ideas, contributions



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

Thanks for your attention! Any questions?



BCI Meeting 2025 - WS19 - June 5

**HappyFeat: an interactive and efficient
BCI framework for clinical applications**



Additional material

Arthur Desbois

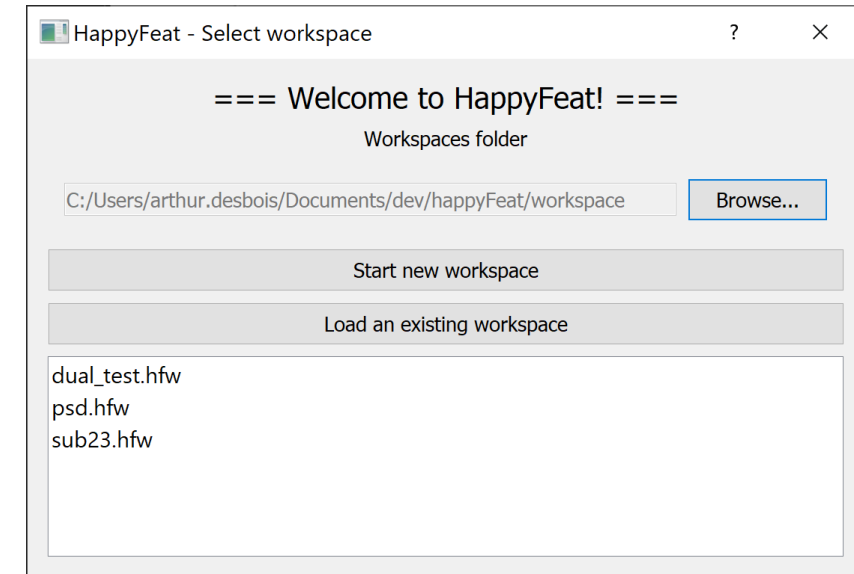
Inria Paris, NERV team, ICM (Paris Brain Institute)

1. Workspace creation/selection

- Create a new environment (file tree, configuration, data structures) for the experiment/analysis

Ex: **sub23_session01**

- ... or select an already existing one.



➔ All user settings & manipulations are saved and available in the workspace for future usage or sharing:

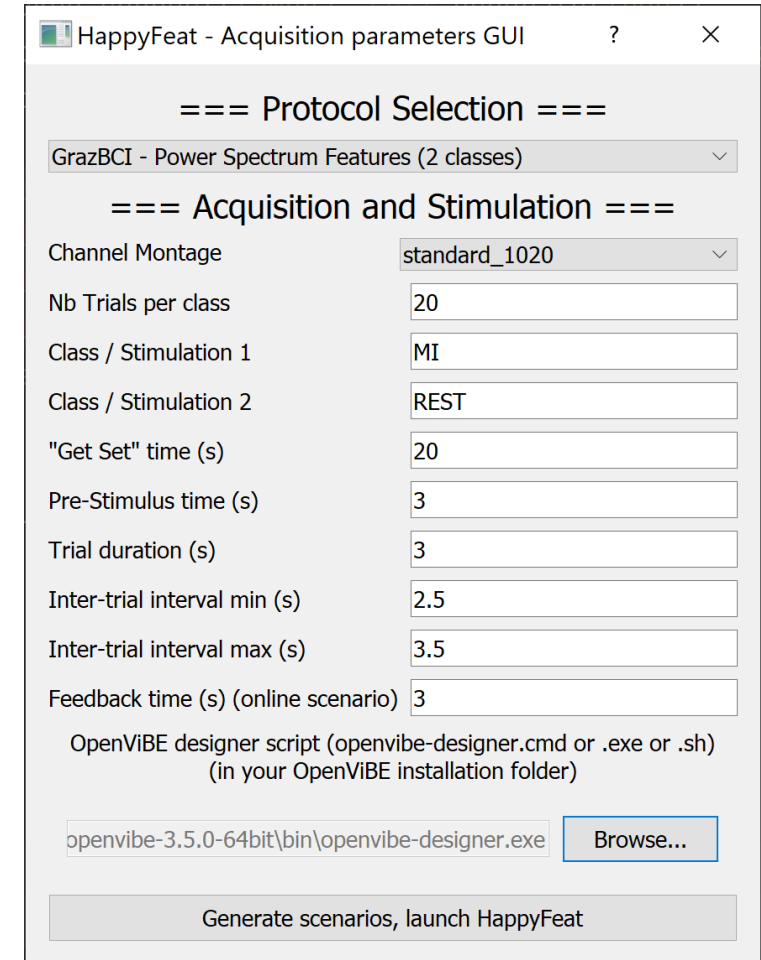
- Extraction parameters (spectral analysis, list of processed files...)
- Training attempts (runs & features used, training accuracy...)

1. Workspace creation/selection

- When creating a new workspace:
 - the feature type (PSD, Connectivity...) is chosen
 - experimental/acquisition parameters are entered (electrode montage, stimulation names, experimental layout...)

➔ This step allows to automatically set up the OpenViBE acquisition scenario.

These experimental parameters are also important for the different processing steps of HappyFeat.



The screenshot shows the 'HappyFeat - Acquisition parameters GUI' window. It features a title bar with a question mark and a close button. The main content area is divided into sections by separator lines. The first section, '=== Protocol Selection ===', contains a dropdown menu currently set to 'GrazBCI - Power Spectrum Features (2 classes)'. The second section, '=== Acquisition and Stimulation ===', contains several input fields: 'Channel Montage' (dropdown set to 'standard_1020'), 'Nb Trials per class' (text box with '20'), 'Class / Stimulation 1' (text box with 'MI'), 'Class / Stimulation 2' (text box with 'REST'), '"Get Set" time (s)' (text box with '20'), 'Pre-Stimulus time (s)' (text box with '3'), 'Trial duration (s)' (text box with '3'), 'Inter-trial interval min (s)' (text box with '2.5'), 'Inter-trial interval max (s)' (text box with '3.5'), and 'Feedback time (s) (online scenario)' (text box with '3'). Below these fields is a label for the 'OpenViBE designer script' with instructions to use 'openvibe-designer.cmd or .exe or .sh' in the installation folder. A text box contains the path 'openvibe-3.5.0-64bit\bin\openvibe-designer.exe', followed by a 'Browse...' button. At the bottom, a large button is labeled 'Generate scenarios, launch HappyFeat'.

HappyFeat - Typical use case



1. Workspace creation/selection

2. Feature Extraction

- Files in the “signals” folder of the workspace are listed in the left panel
- Select one or multiple files...
- Enter the parameters for feature extraction...
- Press the button!

💡 Along the **workspace** mechanism, HappyFeat also uses **work sessions** to keep track of extraction parameters.

Therefore, all steps realized with a set of extraction parameters (which files have undergone extraction + training attempts) are stored and can be recovered.

HappyFeat - Feature Selection interface

Options

== 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 (PSD) (s) 0.5
Window Shift (PSD) (s) 0.4
Connectivity Estimator abs(Coh.)
Sliding Window (Connect.) (s) 0.5
Window overlap (Connect.) (%) 20
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

Update with current extraction params

Extract Features and Trials

== VISUALIZE FEATURES ==

Test-[2022.05.05-16.19.27]-SPECTRUM-CONNECT
Test-[2022.05.05-16.30.16]-SPECTRUM-CONNECT

Load file(s) for analysis

Frequency min 0
Frequency max 40
Sensor for PSD visualization CP3
Topography Freq (Hz), use ":" for freq band 12
Scale Colormap (R²map and Topography) ☒

Power Spectrum Node Strength

Freq.-chan. R² map Freq.-chan. R² map

PSD for the 2 classes NodeStr. for the 2 classes

Brain Topography Brain Topography

== CLASSIFIER TRAINING ==

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

Power Spectrum Connectivity

Add feature Add feature

Remove feature Remove feature

CP3;22 CP3;22

Number of k-fold for classification 10

Test-[2022.05.05-16.19.27]-TRIALS.csv
Test-[2022.05.05-16.30.16]-TRIALS.csv

--- Last Training Results ---

#	Score	Feats (expand for details)
> 1	97.5	CP3;8
> 2	72.5	CP3;12
> 3	95.0	CP3;9
> 4	97.5	CP3;9 + others...

TRAIN CLASSIFIER

HappyFeat - Typical use case



1. Workspace creation/selection
2. Feature Extraction
3. Analysis & Feature Selection

- “Extracted” files (runs) appear in the central panel.
- Select which runs to analyze...

HappyFeat - Feature Selection interface

Options

== 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)
EOI offset (s)
Sliding Window (PSD) (s)
Window Shift (PSD) (s)
Connectivity Estimator
Sliding Window (Connect.) (s)
Window overlap (Connect.) (%)
Auto-regressive estim. length (s)
Frequency resolution (ratio)
--- 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

== VISUALIZE FEATURES ==

Test-[2022.05.05-16.19.27]-SPECTRUM-CONNECT
Test-[2022.05.05-16.30.16]-SPECTRUM-CONNECT

Frequency min
Frequency max
Sensor for PSD visualization
Topography Freq (Hz), use ":" for freq band
Scale Colormap (R²map and Topography) ☒

Power Spectrum Node Strength

== CLASSIFIER TRAINING ==

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

Power Spectrum Connectivity

CP3;22 CP3;22
Number of k-fold for classification
Test-[2022.05.05-16.19.27]-TRIALS.csv
Test-[2022.05.05-16.30.16]-TRIALS.csv

--- Last Training Results ---

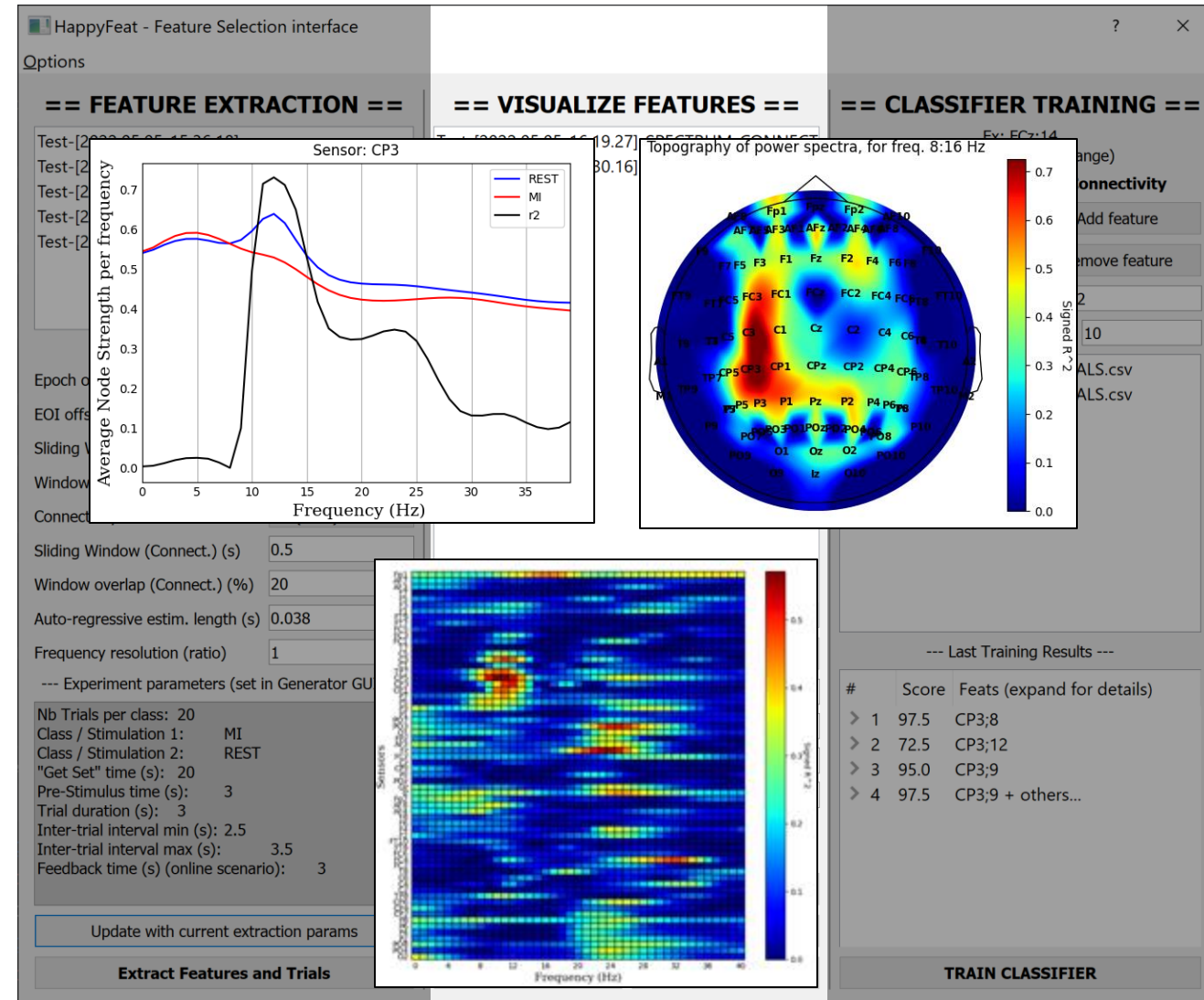
#	Score	Feats (expand for details)
> 1	97.5	CP3;8
> 2	72.5	CP3;12
> 3	95.0	CP3;9
> 4	97.5	CP3;9 + others...

HappyFeat - Typical use case



1. Workspace creation/selection
2. Feature Extraction
3. Analysis & Feature Selection

- “Extracted” files (runs) appear in the central panel.
- Select which runs to analyze...
- And use the different **visualization tools** to select adequate features:
 - Channel/frequency R^2 map
 - Per-channel metric comparison
 - Per-frequency R^2 topography
- More coming soon...

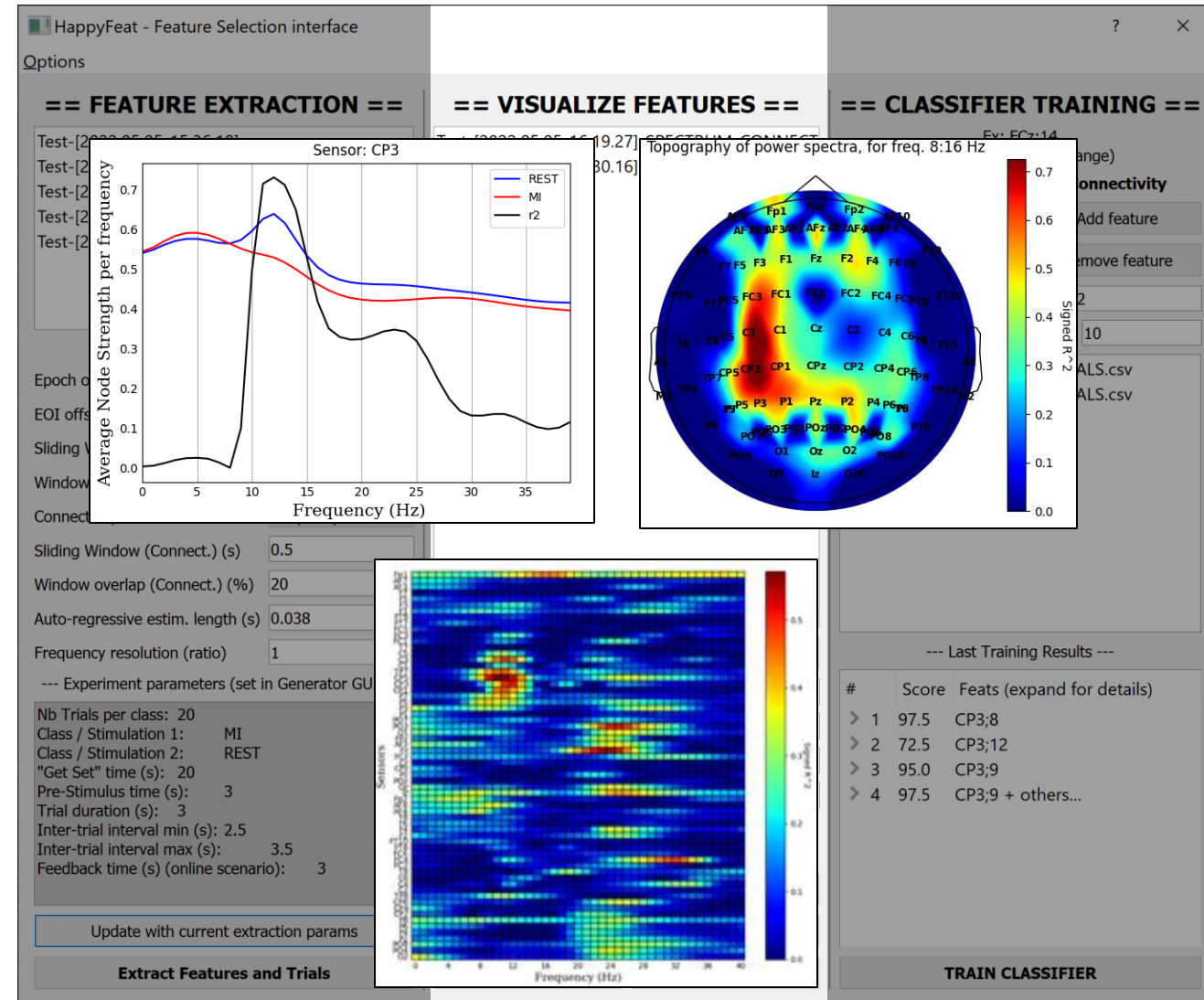


HappyFeat - Typical use case



1. Workspace creation/selection
2. Feature Extraction
3. Analysis & Feature Selection

- “Extracted” files (runs) appear in the central panel.
- Select which runs to analyze...
- And use the different **visualization tools** to select adequate features.
- ... or use the “**AutoFeat**” mechanism to automatically select a set of (channel/frequency) pairs with the highest R2



HappyFeat - Typical use case



1. Workspace creation/selection
2. Feature Extraction
3. Analysis & Feature Selection
4. Classifier training

- “Extracted” files (runs) also appear in the right panel.
- Select one or multiple runs...
... their trials will all be used for training.
- Select one or multiple feature(s) of interest (channel;frequency)
- Click the button!

HappyFeat - Feature Selection interface

Options

== 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 (PSD) (s) 0.5
Window Shift (PSD) (s) 0.4
Connectivity Estimator abs(Coh.)
Sliding Window (Connect.) (s) 0.5
Window overlap (Connect.) (%) 20
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

Update with current extraction params

== VISUALIZE FEATURES ==

Test-[2022.05.05-16.19.27]-SPECTRUM-CONNECT
Test-[2022.05.05-16.30.16]-SPECTRUM-CONNECT

Load file(s) for analysis

Frequency min 0
Frequency max 40
Sensor for PSD visualization CP3
Topography Freq (Hz), use ":" for freq band 12
Scale Colormap (R²map and Topography) ☒

Power Spectrum Node Strength
Freq.-chan. R² map Freq.-chan. R² map
PSD for the 2 classes NodeStr. for the 2 classes
Brain Topography Brain Topography

== CLASSIFIER TRAINING ==

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

Power Spectrum Connectivity
Add feature Add feature
Remove feature Remove feature
CP3;22 CP3;22
Number of k-fold for classification 10

Test-[2022.05.05-16.19.27]-TRIALS.csv
Test-[2022.05.05-16.30.16]-TRIALS.csv

--- Last Training Results ---

#	Score	Feats (expand for details)
> 1	97.5	CP3;8
> 2	72.5	CP3;12
> 3	95.0	CP3;9
> 4	97.5	CP3;9 + others...

TRAIN CLASSIFIER

HappyFeat - Typical use case



1. Workspace creation/selection
2. Feature Extraction
3. Analysis & Feature Selection
4. Classifier training

- “Extracted” files (runs) also appear in the right panel.
 - Select one or multiple runs...
... their trials will all be used for training.
 - Select one or multiple feature(s) of interest (channel;frequency)
 - Click the button!
- ➔ A training accuracy is provided
- ➔ All previous training results are available (displayed in the GUI, and classif. weights in the workspace)

The screenshot displays the HappyFeat - Feature Selection interface. A dialog box titled "Classifier Training Score" is open, showing the following information:

- Using spectral features: Channel C1 at 13 Hz
- Overall accuracy : 95.0%
- Class 1 | Precision : 0.91 | Sensitivity : 1.0 | F_1 Score : 0.95
- Class 2 | Precision : 1.0 | Sensitivity : 0.9 | F_1 Score : 0.95
- Results written in file: generated/classifier-weights.xml
- If those results are satisfying, you can now open generated/sc3-online.xml in the Designer

The background interface shows three main tabs: "FEATURE EXTRACTION", "VISUALIZE FEATURES", and "CLASSIFIER TRAINING". The "CLASSIFIER TRAINING" tab is active, displaying a table of "Last Training Results" and a "TRAIN CLASSIFIER" button.

#	Score	Feats (expand for details)
> 1	97.5	CP3;8
> 2	72.5	CP3;12
> 3	95.0	CP3;9
> 4	97.5	CP3;9 + others...

HappyFeat - Typical use case



1. Workspace creation/selection
2. Feature Extraction
3. Analysis & Feature Selection
4. Classifier training
5. Online classification

A satisfactory set of classification features has been found!

→ Generate a **ready-to-use**
OpenViBE (soon TimeFlux) scenario

- trained classifier
- selected classif.features

HappyFeat - Feature Selection interface

Options

== 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 (PSD) (s) 0.5
Window Shift (PSD) (s) 0.4
Connectivity Estimator abs(Coh.)
Sliding Window (Connect.) (s) 0.5
Window overlap (Connect.) (%) 20
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

Update with current extraction params

== VISUALIZE FEATURES ==

Test-[2022.05.05-16.19.27]-SPECTRUM-CONNECT
Test-[2022.05.05-16.30.16]-SPECTRUM-CONNECT

Load file(s) for analysis

Frequency min 0
Frequency max 40
Sensor for PSD visualization CP3
Topography Freq (Hz), use ":" for freq band 12
Scale Colormap (R²map and Topography) ☒

Power Spectrum Node Strength
Freq.-chan. R² map Freq.-chan. R² map
PSD for the 2 classes NodeStr. for the 2 classes
Brain Topography Brain Topography

== CLASSIFIER TRAINING ==

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

Power Spectrum Connectivity
Add feature Add feature
Remove feature Remove feature

CP3;22 CP3;22

Number of k-fold for classification 10

Test-[2022.05.05-16.19.27]-TRIALS.csv
Test-[2022.05.05-16.30.16]-TRIALS.csv

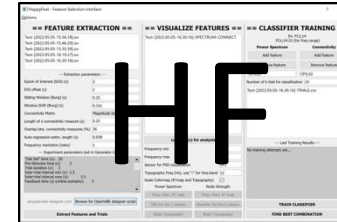
--- Last Training Results ---

#	Score	Feats (expand for details)
> 1	97.5	CP3;8
> 2	72.5	CP3;12
> 3	95.0	CP3;9
> 4	97.5	CP3;9 + others...

TRAIN CLASSIFIER

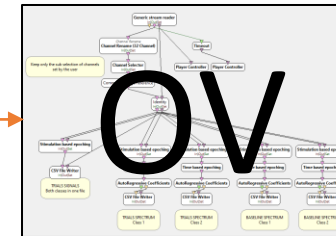
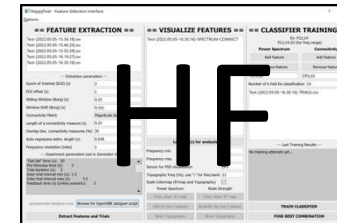
1. Workspace creation/selection

- Create a new environment (file tree, configuration, data structures) for the experiment/analysis
- ... or select an already existing one.
- All user settings & manipulations are saved and available in the workspace for future usage or sharing:
 - Extraction parameters (spectral analysis, list of processed files...)
 - Training attempts (runs & features used, training accuracy...)

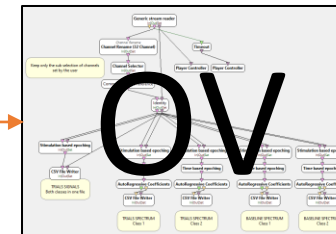


1. Workspace creation/selection
2. **ML Pipeline / “Feature type” selection**

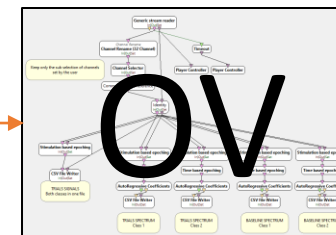
- Selecting btw. multiple “template” scenarios depending on the use case (power spectrum, connectivity type...)
- Edit basic/common parameters (acquisition, extraction, training...)



Extraction scenario



Training scenario

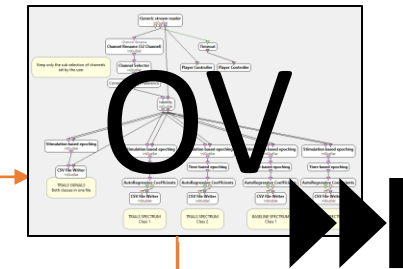


Online classif.
scenario

1. Workspace creation/selection
 2. ML Pipeline / “Feature type” selection
 3. 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** (ML 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? (with openvibe)

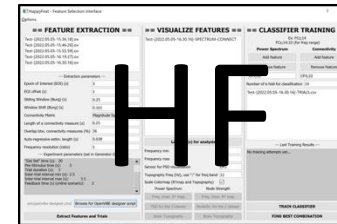


1. Workspace creation/selection
2. MI Pipeline / “Feature type” selection
3. Feature Extraction
4. Analysis, Feature Selection

- 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 R^2 map
 - ➔ PSD (or connect. metric) comparison btw. the 2 conditions (MI/REST) for a given electrode
 - ➔ Time/frequency ERD/ERS analysis for each condition
 - ➔ R^2 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.) R^2 maps for both Power Spectrum and Connectivity in parallel



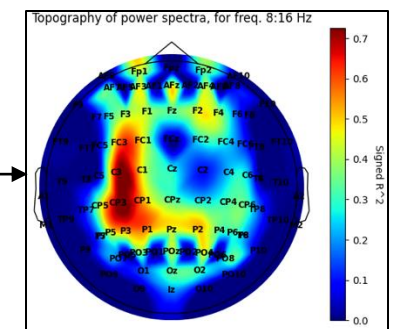
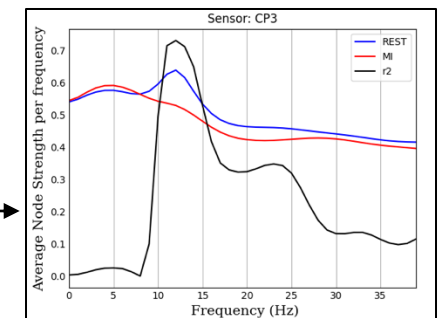
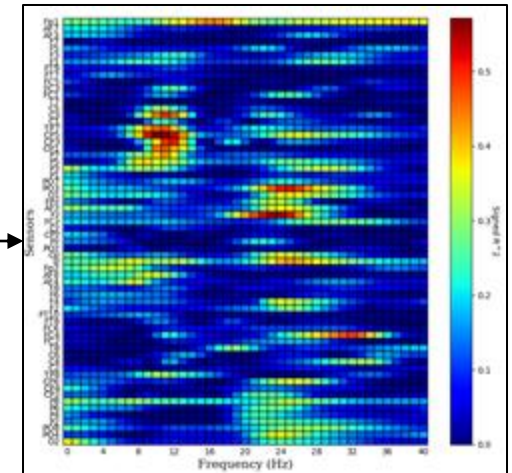
Signal 1
(.ov)

Spectral Power
(class 1) CSV

Spectral Power
(class 2) CSV

Trials CSV

Metadata CSV



HappyFeat - How? (with openvibe)



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

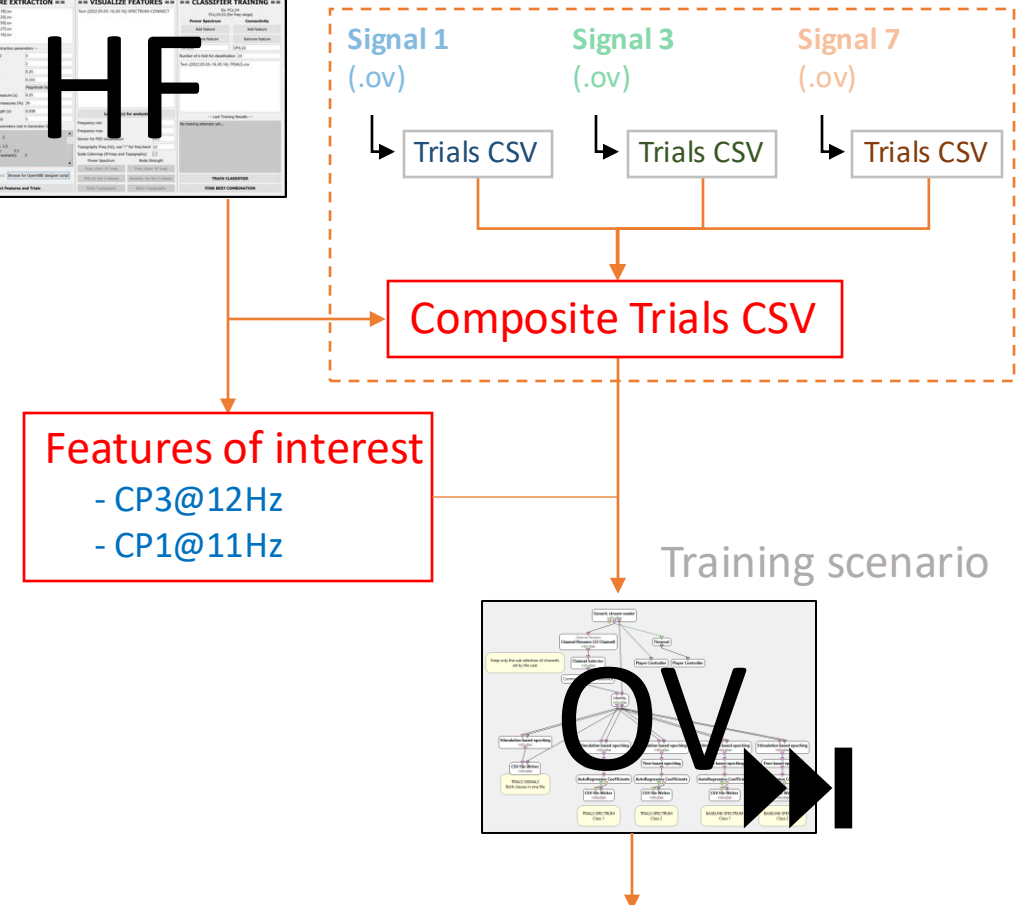
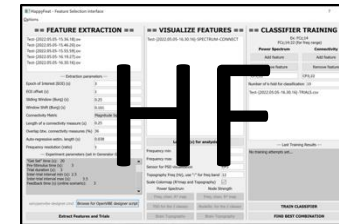
- Set one or more **Features of Interest** (Electrode/Freq.)
- **Select file(s)** with which you want to train your classifier
If > 1 file : their trials are **automatically concatenated**
- **Run the Classifier Training scenario**
(Auto. generated in step 1 + auto. edited with FOIs)
➔ Classification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

- **Arthur Desbois** (ingénieur Inria) - lead
 - Principal développeur et mainteneur de l'application (GUI et back end)
 - Supervision des travaux, gestion de projet, tests, packaging, documentation...
 - **Tristan Venot** (PhD, ingénieur Inria)
 - Protocoles expérimentaux BCI (Braccio)
 - Aide au développement algo. et à l'amélioration de l'application (GUI et back end)
 - **Wafa Skhiri** (stage M2 2024, étudiant PhD à partir de 2025)
 - Compatibilité HappyFeat + Timeflux
 - A partir de 2025 : aide au développement algo. (connectivité, graphes)
 - **Camile Bousfiha** (étudiant PhD depuis 2023)
 - Protocole BCINET (50 patients, 18 sessions) utilisant HappyFeat
- (+ Marie-Constance Corsi (PI) et Fabrizio De Vico Fallani (PI) - orientations stratégiques)