Order Pattern Matching for ADHD analysis using EEG waves - User Guide

25-1-R-1

Lecturers Guide:
Dr. Anat Dahan
Dr. Samah Idrees Ghazawi

Students Submit:

Tamir Nahari
Yaakov Shitrit

User Guide - Approach 1

1. Introduction

1. Introduction

This user guide provides detailed instructions for using Approach 1 in the EEG-ADHD analysis system. Approach 1 focuses on detecting recurring patterns directly from the waveform structure of the EEG signal, without relying solely on frequency-domain transformations or statistical summaries.

The primary goal of this approach is to discover shape-based motifs—distinct waveform cycles that appear repeatedly and may differ between ADHD and control groups. By identifying these patterns, we aim to reveal structural differences in EEG dynamics that could serve as potential biomarkers for neurological conditions such as ADHD.

Approach 1 operates on the raw EEG signal (after denoising and ICA decomposition) and applies a cycle detection algorithm to segment the signal into meaningful units (bursts). Each segment is then analyzed for its structural characteristics, and symbolic representations are assigned to capture waveform motifs. These motifs are cataloged and compared across groups.

This guide will walk you through the full process, including:

- Required input files and folder structure
- Running the preprocessing and motif discovery pipeline
- Interpreting the results (motif tables, visualizations, statistics)
- Using the GUI tools to browse and explore patterns

Note: Approach 1 is especially suited for research questions that require precise analysis of waveform morphology, as opposed to statistical features.

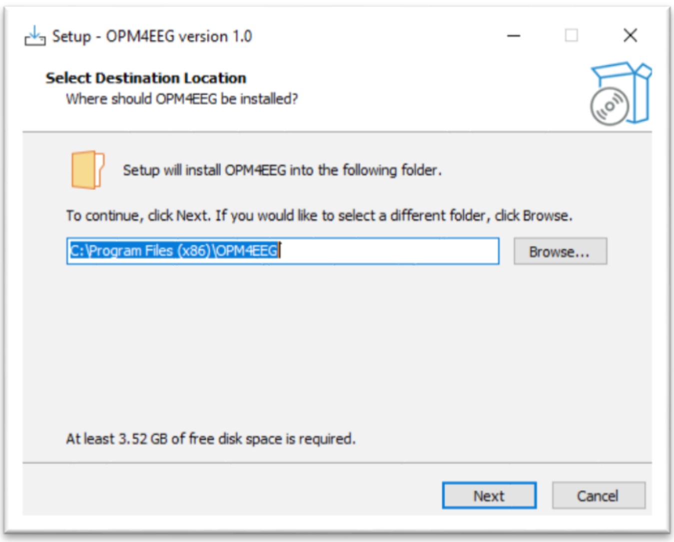
2. Installation

2.1. Go to the following link:
https://mega.nz/file/3FhXHZoA#aKZXVPIHNPrUEk1PrsJuH4dvcqWF7kes-m2AO4BB0LQ

https://mega.nz/file/3FhXHZoA#aKZXVPIHNPrUEk1PrsJuH4dvcqWF7kes-m2AO4BB0LQ
And download from there the .exe file of the user interface program of Research Approach 1 called "OPM4EEG_installer.exe".

2.2. Double-click and install the file according to the following installation steps:

🖼 Screen 1 – Select Destination Location



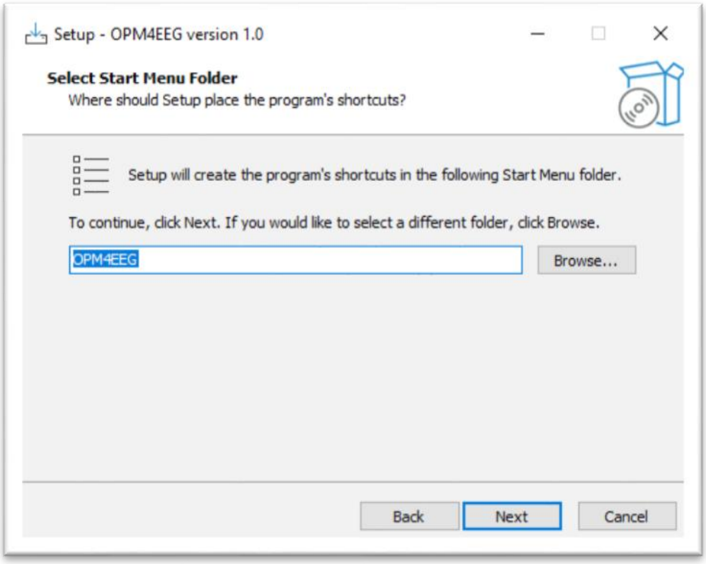
What you see:
This screen asks you to choose the folder where the program will be installed. The default location is:
`C:\Program Files (x86)\OPM4EEG`

What to do:

- If you're happy with the default location, simply click **Next**.

- If you'd like to install the program in a different folder (e.g., Desktop or another drive), click **Browse**, choose a folder, then click **Next**.

🖼 Screen 2 – Select Start Menu Folder

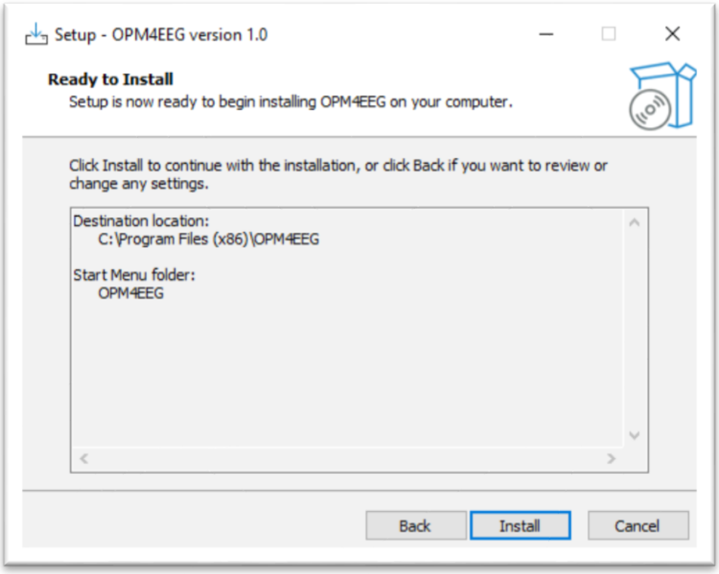


What you see:
This screen asks you to choose the name of the Start Menu folder where the program shortcuts will be created.

What to do:

- The default name is `OPM4EEG`. This is the folder that will appear in your Windows Start Menu.
- You can change the name or choose a different location by clicking **Browse**, but it's usually fine to leave as is.
- Click **Next** to continue.

🖼 Screen 3 – Ready to Install

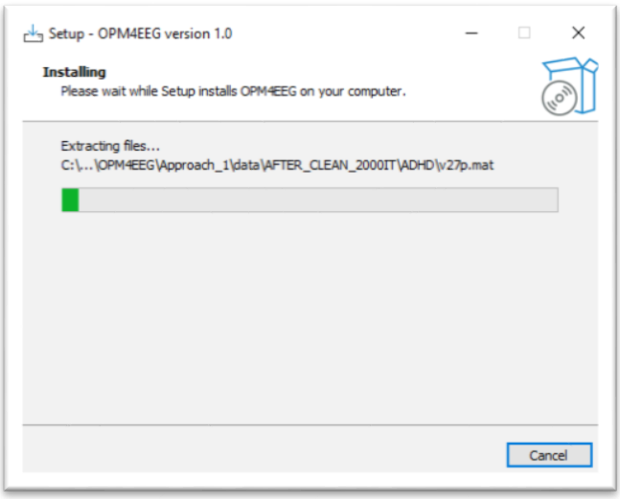


What you see:
This is a summary screen that shows where the program will be installed and what the Start Menu folder will be.

What to do:

- Review the installation settings.
- If everything looks correct, click **Install** to begin installation.
- If you want to go back and change something, click **Back**.

🖼 Screen 4 – Installing



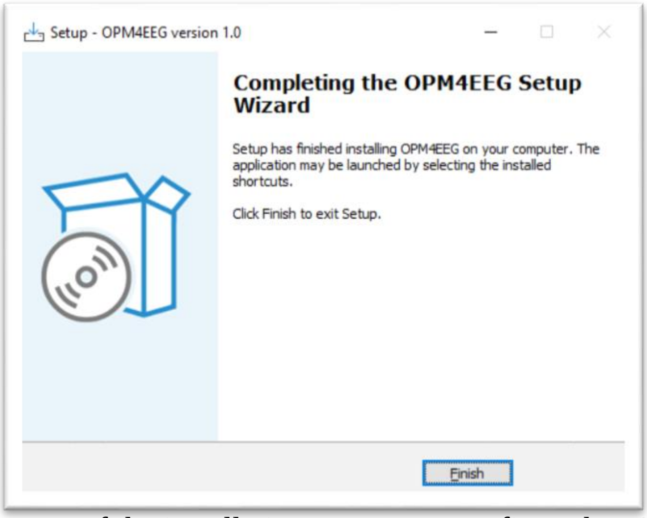
What you see:
The installation is now in progress. You'll see a progress bar and a list of files being extracted (for example: `.mat` files from the `Approach_1/data` directory).

What to do:

- Wait for the installation to finish. Do not close the window or turn off your computer.
- If necessary, you can click **Cancel** to stop the installation, but this is not recommended unless something is wrong.

🖼 Screen 5 – Completing the Setup Wizard

What you see:



This is the final screen of the installation process. It confirms that **OPM4EEG** has been successfully installed on your computer.

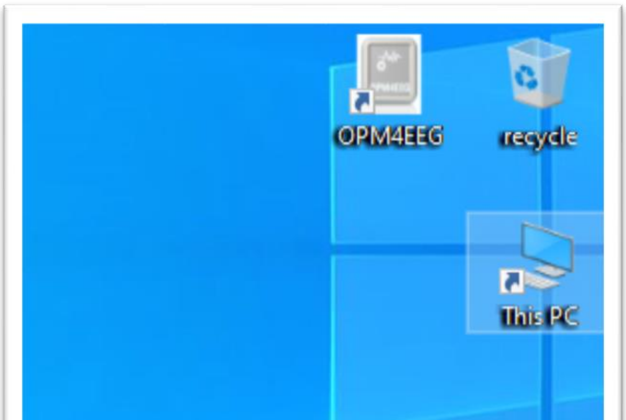
What to do:

- Click **Finish** to exit the setup wizard.
- You can now launch the program by using the shortcut in your **Start Menu** under the folder you chose earlier (e.g., `OPM4EEG`), or by navigating to the installation folder.

✅ **Installation Complete!**
You're now ready to begin using the OPM4EEG application.

3. Running the Application

🖥 Desktop Shortcut – OPM4EEG



After installation, a shortcut icon named **OPM4EEG** is created on your desktop.

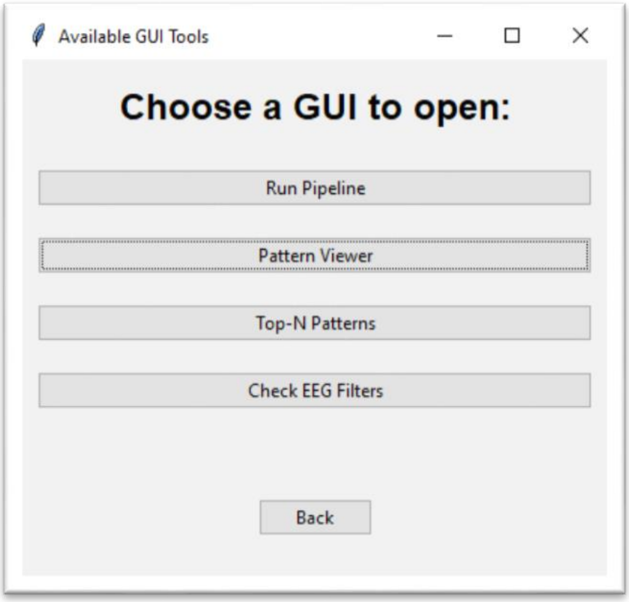
▶ To run the program:

1. **Locate the OPM4EEG icon** on your desktop (as shown above).
2. **Double-click** the icon using the left mouse button.
3. The program will launch and display the main interface or menu window.
4. Follow the on-screen instructions to begin using the OPM4EEG system.

💡 **Tip**: If nothing happens, try right-clicking the icon and selecting **"Run as administrator"** (especially on systems with strict permissions).

☐ Available GUI Tools – Main Menu

This window appears after launching the **OPM4EEG** application.
It serves as the **main control panel**, allowing you to select which graphical tool (GUI) to open based on your task.



🛠 Menu Options Explained:

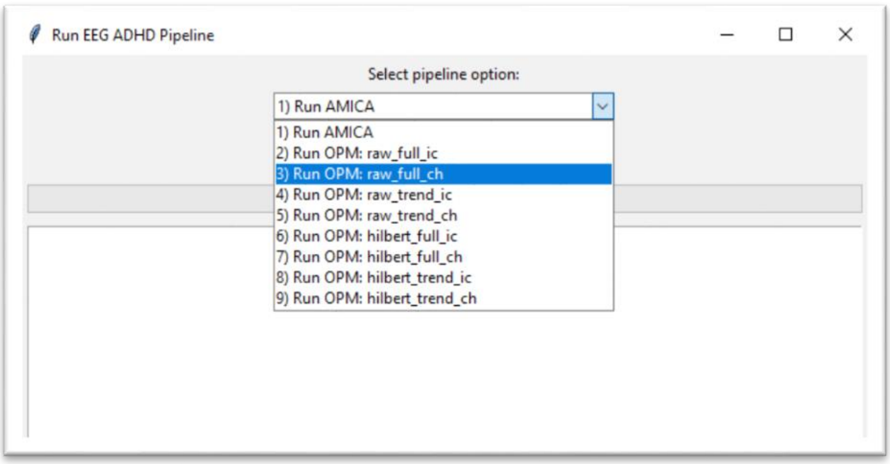
1. **Run Pipeline**
   Launches the full EEG processing pipeline.

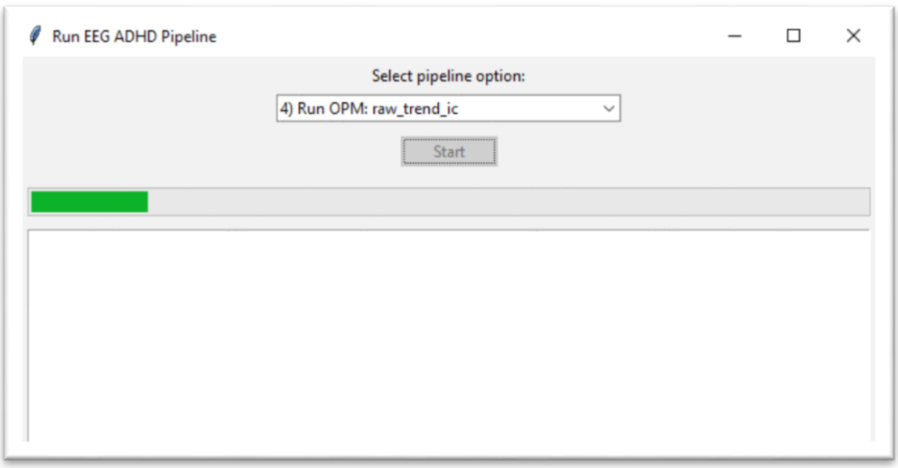
This includes preprocessing steps such as filtering, ICA (AMICA), motif discovery using OPM, and result saving.

2. **Pattern Viewer**
   Opens a visualization tool that lets you explore detected EEG motifs.
   You can select a subject, channel, or IC and view waveform segments and motif matches.
3. **Top-N Patterns**
   Displays the most frequently occurring EEG motifs across subjects.
   Useful for identifying common patterns and comparing between ADHD and control groups.
4. **Check EEG Filters**
   Opens a diagnostic tool to inspect each stage of signal filtering for a specific subject.
   Shows raw, notch, CAR, low-pass, high-pass, and theta-filtered waveforms.
5. **Back**
   Returns to the previous menu or closes the menu (depending on your app's structure).

□ Run EEG ADHD Pipeline – Main Processing Window

This dropdown menu allows the user to choose which step of the **Approach 1 pipeline** to run. Each option corresponds to a different method of analyzing EEG data using AMICA and OPM (Oscillatory Pattern Matching).





1) Run AMICA

- Performs Independent Component Analysis (ICA) using the **AMICA algorithm**.
- This is usually the **first step**, separating EEG signals into statistically independent sources (ICs).

---

2) Run OPM: raw_full_ic

- Runs OPM motif discovery on the **raw EEG signal**, using the **full signal (no trend removal)**, grouped **by IC** (independent component).

3) Run OPM: raw_full_ch

- Like the previous option, but groups motifs **by channel** instead of by IC.

---

4) Run OPM: raw_trend_ic

- Runs OPM on the **detrended raw signal**, grouped **by IC**.
- Focuses on oscillatory cycles **after removing slow trends** from the waveform.

5) Run OPM: raw_trend_ch

- Same as above, but groups motifs **by channel**.

---

6) Run OPM: hilbert_full_ic

- Applies the **Hilbert transform** to extract the envelope of the raw signal.
- Performs OPM on the **full Hilbert envelope**, grouped **by IC**.

7) Run OPM: hilbert_full_ch

- Same Hilbert-based analysis, but motifs are grouped **by channel**.

---

8) Run OPM: hilbert_trend_ic

- Runs OPM on the **Hilbert envelope after trend removal**, grouped **by IC**.
- Ideal for identifying bursts with clean cycle transitions.

9) Run OPM: hilbert_trend_ch

- Same as above, but groups results **by channel**.

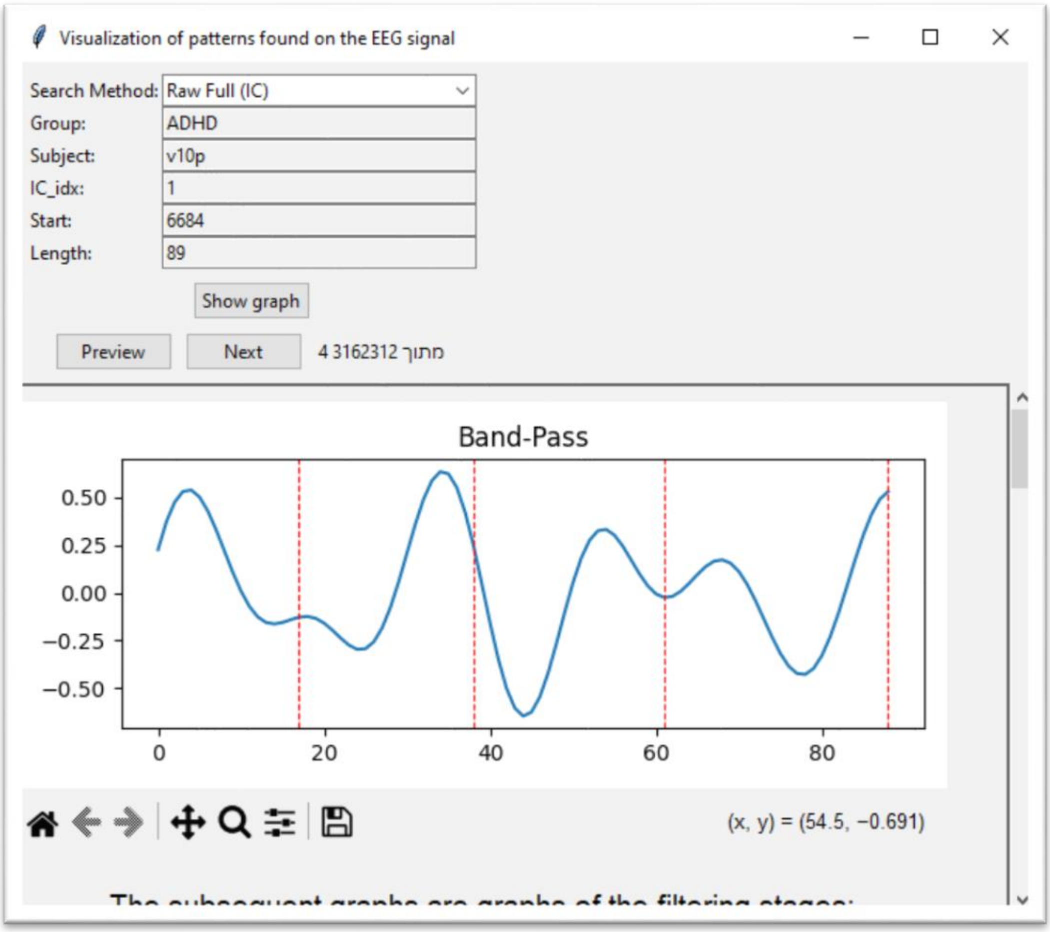
📝 Notes:

- All OPM steps search for **repeating EEG waveform patterns** (motifs).
- "Full" = no trend removal.
  "Trend" = trend-removed signal.
  "IC" = grouped by independent component.
  "Ch" = grouped by EEG channel.

💡 **Tip:** Always run **AMICA first (option 1)** before running any of the OPM-based steps.

🔍 Pattern Viewer – Visualization of Patterns Found on the EEG Signal

This screen allows the user to explore EEG waveform **motifs (patterns)** that were detected using the selected **OPM search method**.



Graph Sections Explained:

1. Displaying a specific wave after the last filtering (Band-pass) with the wave divided by its cycles (performed using Hilbert)

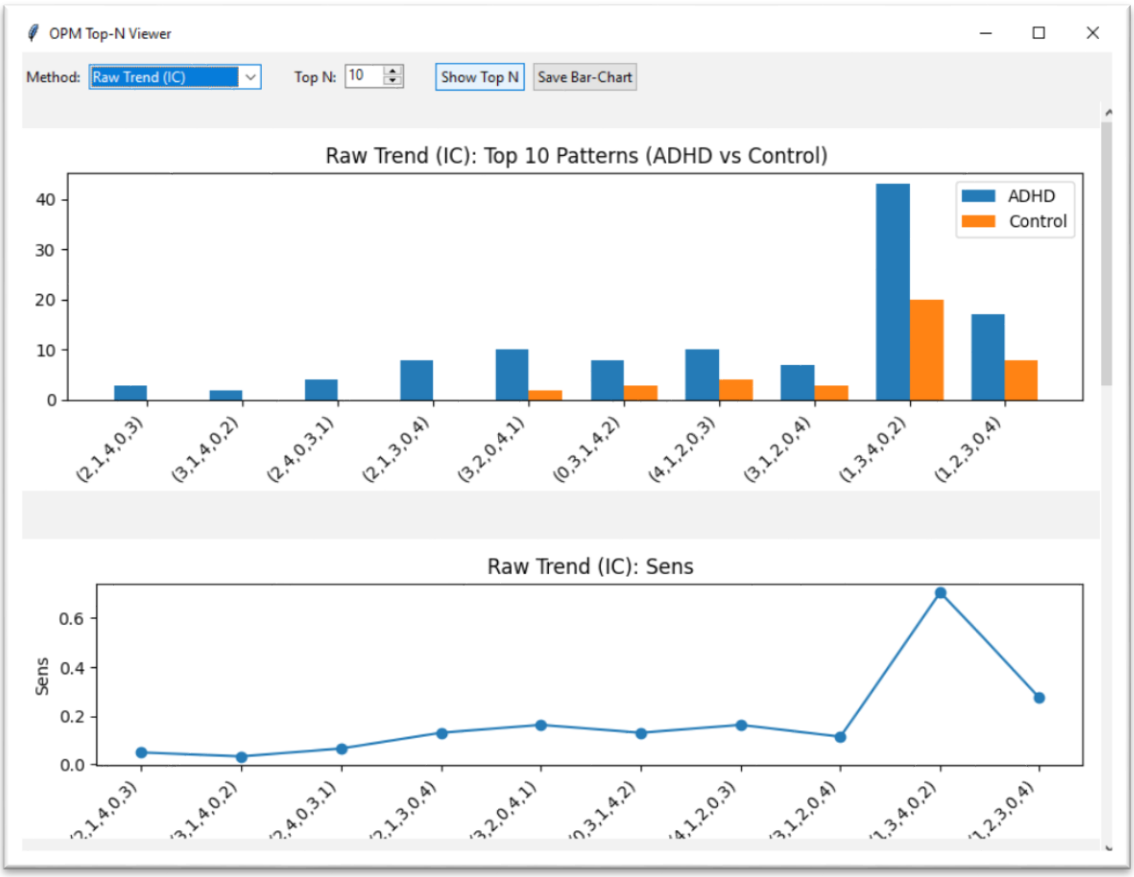
- Displays the motif segment after applying band-pass filtering
- Red vertical lines indicate **cycle boundaries** detected in this segment.
- Useful for understanding waveform structure.

2. Raw (Below Band-Pass)

- This graph shows the same segment **before any filtering.**
- The raw signal is useful for comparing how the band-pass filtering affects the waveform.

Middle Panel – Buttons:

- **Show graph:**
  Displays the waveform segment associated with the motif (shown in the Band-Pass plot below).
- **Preview / Next:**
  Navigate between different motif occurrences.
  The counter at the bottom (e.g., `4 3162312 מתוך`) shows the current motif index out of the total number of patterns.



📊 OPM Top-N Viewer – Pattern Frequency & Sensitivity Analysis

This tool allows the user to compare the **top N most frequent patterns** found in EEG signals between the **ADHD** and **Control** groups.

Top Section Controls:

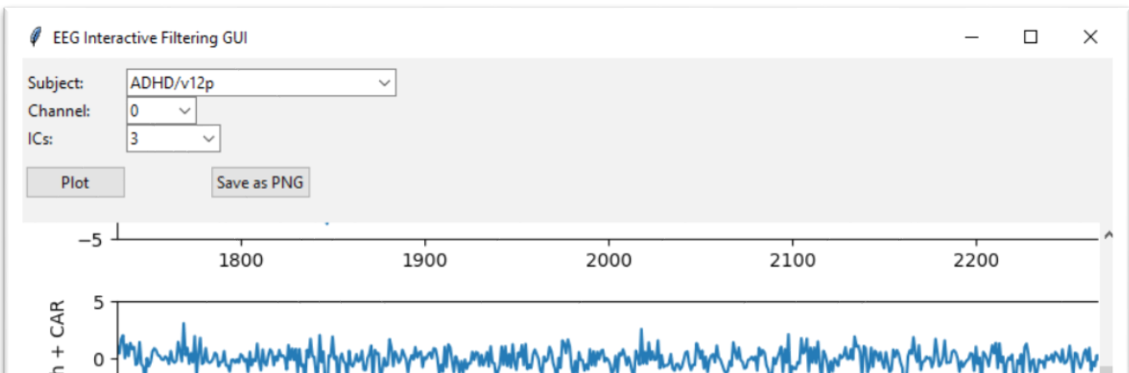
- **Method:**
  Selects the OPM search method used to generate the results (e.g., `Raw Trend (IC)`).
- **Top N:**
  Sets how many of the most frequent patterns to display (e.g., Top 10, Top 20, etc.).
- **Show Top N:**
  Refreshes the chart with the currently selected method and Top-N value.
- **Save Bar-Chart:**
  Saves the bar chart as an image file for documentation or publication.

📉 Upper Graph – Frequency of Top Patterns:

- **Title:**
  `**For example:** Raw Trend (IC): Top 10 Patterns (ADHD vs Control)`
- **Bars:**
  Each pattern (e.g., `(3,4,0,2)`) represents a symbolic frequency label sequence found in the EEG motif discovery.
  - Blue bars: number of times this pattern was found in **ADHD** subjects.
  - Orange bars: number of times in **Control** subjects.
- **Interpretation:**
  Patterns that appear **significantly more often in one group** may indicate **neurophysiological differences** and serve as potential biomarkers.

EEG Interactive Filtering GUI – Full Filter Chain Visualization

This view shows the full **filtering pipeline** applied to either a selected **channel** or **independent component (IC)** for a specific EEG subject.



📈 Graph Sections Explained (Top to Bottom):

1. Notch + CAR

- This signal has been:
  - **Notch filtered**: to remove 50/60 Hz electrical noise.
  - **CAR (Common Average Referencing)**: to reduce background noise by subtracting the average of all channels.

2. LP (Low-Pass Filter)

- Shows the signal after removing **high-frequency noise**, typically above ~30 Hz.
- Smooths the waveform and enhances visible slow rhythms.

3. HP (High-Pass Filter)

- Removes **slow drift or baseline wandering**, keeping only frequencies above ~1 Hz.
- Highlights transient oscillatory activity.

4. Theta & Envelope

- Blue line: the **bandpass-filtered IC** in the theta range (4–8 Hz).
- Orange dashed line: the **Hilbert envelope** – represents the **instantaneous amplitude** of the signal.
- Red dashed vertical lines: **detected cycle boundaries**, representing full theta wave cycles.

☐ Top Controls Recap:

- **Subject / Channel / ICs** – lets you select the signal to inspect.
- **Plot** – renders all the graphs above for the chosen signal.
- **Save as PNG** – exports the full view to an image file.

📌 Interpretation:

This tool is critical for:

- **Verifying** the preprocessing steps on a single IC or channel
- **Diagnosing** signal issues such as residual noise or distortion
- **Manually inspecting** theta cycles and envelope accuracy

User Guide - Approach 2

1. Introduction

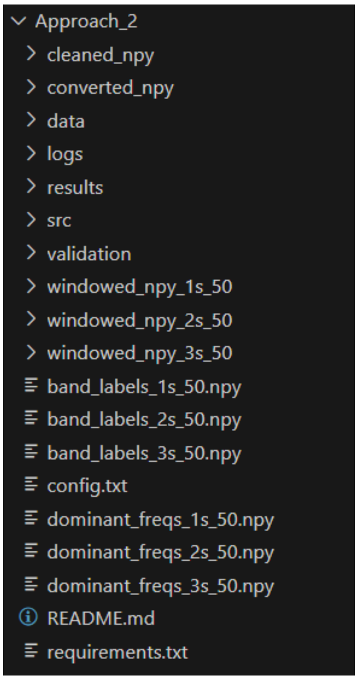
This project analyzes EEG signals from individuals with ADHD and a control group to identify group-specific brain activity patterns using the Order-Preserving Matching (OPM) algorithm — the core of the system.

To make the research process more accessible, we transformed the entire EEG analysis pipeline into a user-friendly system with a simple graphical interface.
This allows even non-technical users to run complex signal analysis steps with just a few clicks.

The system provides a full pipeline that allows users to:

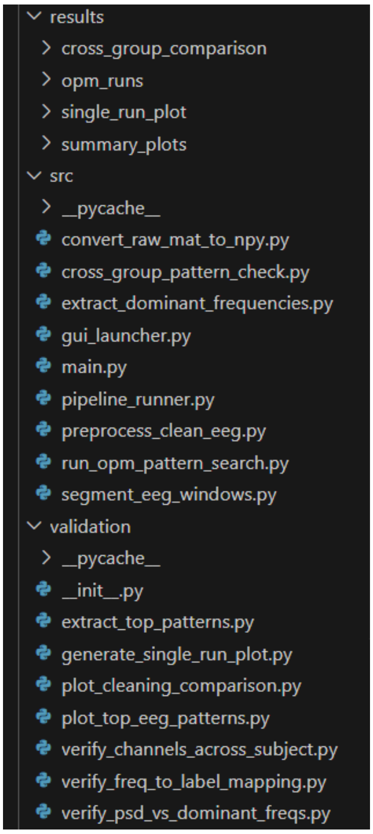
- Convert and **clean** raw EEG .mat files (filters, ICA, normalization),
- Segment signals into overlapping windows,
- Extract dominant frequencies using **Welch's PSD** and map them to EEG bands (δ, θ, α, β, γ),
- Detect repeated **order-based patterns** using OPM,
- Compare pattern frequency between ADHD and Control groups,
- **Visualize results** using a GUI or CLI.

2. Project Directory Structure Overview



The figure shows the main folders and files included in the project repository.

- src/ – contains the main source code of the project, including data processing and GUI files.
- validation/ – includes scripts used for evaluating and visualizing results.
- results/ – holds output plots and pattern comparisons after each run.
- converted_npy/, cleaned_npy/, etc. – contain intermediate EEG data from different preprocessing stages.
- windowed_npy_* folders – Contain the segmented EEG signals (overlapping windows) for 1s, 2s, and 3s durations, respectively, used in further pattern analysis.
- band_labels_* files – Store the corresponding EEG band label (δ, θ, α, β, γ) for each window and each channel based on the dominant frequency extracted.
- dominant_freqs_* files – Contain the dominant frequency per channel for each EEG window (calculated using Welch's method) for the 1s, 2s, and 3s window sizes.
- logs/ – contains execution logs of the pipeline.
- README.md – general project description and usage instructions.
- requirements.txt – list of required Python packages.



The following figure shows the contents of the src/ and validation/ folders. These are the main Python scripts that implement the project's logic:

src/ – Core logic

- main.py – Full pipeline runner via command line.
- gui_launcher.py – Launches the graphical interface.
- convert_raw_mat_to_npy.py – Converts EEG .mat files to .npy.
- preprocess_clean_eeg.py – Filters and cleans EEG signals using ICA and normalization.
- segment_eeg_windows.py – Divides EEG signals into overlapping time windows.
- extract_dominant_frequencies.py – Computes dominant frequencies per channel using Welch's method.
- run_opm_pattern_search.py – Detects repeated patterns using the OPM algorithm.
- cross_group_pattern_check.py – Compares patterns between ADHD and Control groups.
- pipeline_runner.py – Internal class used for running the pipeline with parameter control (used by the GUI).

validation/ – Visual checks and pattern quality control

- plot_cleaning_comparison.py – Visualizes EEG before and after cleaning.

- verify_psd_vs_dominant_freqs.py – Checks if dominant frequencies align with highest PSD values.
- extract_top_patterns.py – Extracts the top 10 most frequent patterns across all runs for a specific window size.
- plot_top_eeg_patterns.py – Plots frequently detected EEG patterns.
- verify_freq_to_label_mapping.py – Confirms mapping from frequency to EEG bands (δ, θ, α, β, γ).
- verify_channels_across_subject.py – Verifies that all 19 EEG channels are correctly present across subject.
- generate_single_run_plot.py – Generates a comprehensive summary plot per subject.

3. Installation

Python 3.8+, packages via:

pip install -r requirements.txt

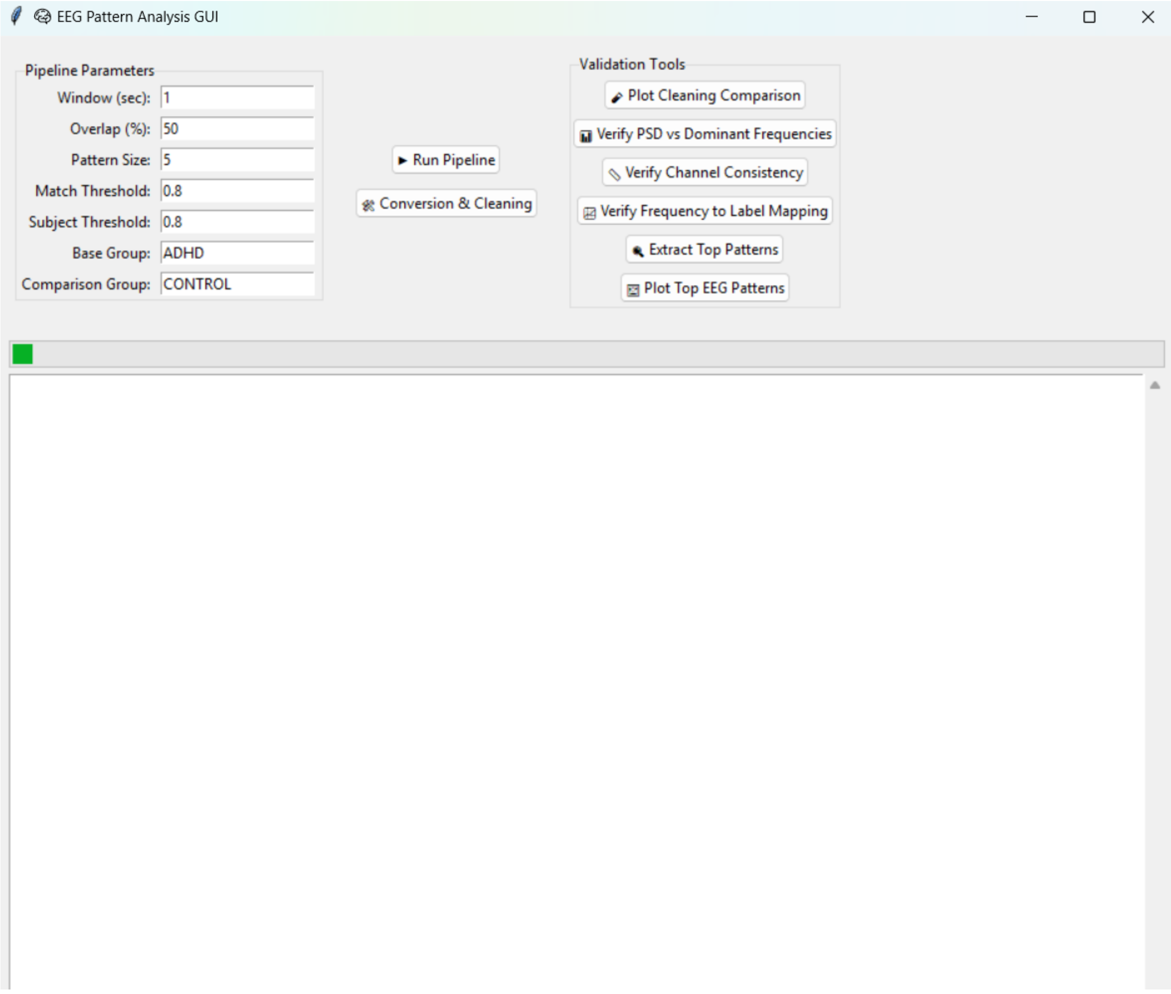
4. Running the Application

To launch the graphical interface, open a terminal in the Approach_2 directory and run:

python -m src.gui_launcher

This will open the **EEG Pattern Detection GUI**, allowing you to:

- Configure and run the full EEG analysis pipeline (conversion → cleaning → segmentation → frequency analysis → OPM).
- Choose parameters like sampling rate, window size, and thresholds.
- Track progress in real time and review output summaries and logs directly in the interface.



The following interface provides full control and validation of the EEG analysis pipeline.
Each section serves a specific purpose:

🧠 Pipeline Parameters (left)

- **Window (sec):** Duration (in seconds) of each EEG segment. Default: 1s.
- **Overlap (%):** Percentage of overlap between consecutive windows. Default: 50%.
- **Pattern Size:** Number of frequency elements in each pattern used by the OPM algorithm. Default: 5.
- **Match Threshold:** Minimum similarity (0–1) required for two patterns to be considered a match. Default: 0.8.
- **Subject Threshold:** Minimum proportion of subjects in the base group that must exhibit a pattern for it to be considered representative. Default: 0.8.
- **Base Group:** Group used to detect patterns (default: “ADHD”).
- **Comparison Group:** Group used for comparison (default: “CONTROL”).

⚙️ Main Controls (center)

- **Run Pipeline:** Executes the full pipeline: segmentation → frequency extraction → OPM → cross-group comparison.
- **Conversion & Cleaning:** *Optional step.*
  Converts raw .mat EEG files to .npy and applies signal cleaning (filters, ICA, normalization).
  ✅ *Note:* This step is usually not needed since data is already preprocessed, but can be used for review or demonstration.

🧪 **Validation Tools (right)**

Tools to visually inspect and verify pipeline outputs:

- **Plot Cleaning Comparison:** Displays EEG signals before and after filtering and ICA.
- **Verify PSD vs Dominant Frequencies:** Confirms that extracted dominant frequencies align with PSD peaks from Welch's method.
- **Verify Channel Consistency:** Ensures all 19 EEG channels are present across subjects.
- **Verify Frequency to Label Mapping:** Validates that frequencies are correctly categorized into EEG bands (δ, θ, α, β, γ).
- **Extract Top Patterns:** Identifies and saves the 10 most frequent patterns for the selected window size.
- **Plot Top EEG Patterns:** Graphically displays the most common detected EEG patterns (based on selected window length).

📊 **Output Console (bottom)**

- **Progress Bar (green):** Indicates the progress of current tasks.
- **Text Box:** Shows real-time logs, including execution status, warnings, and errors.

Running the Pipeline

Click **Run Pipeline** to execute the full process:

1. Segment EEG signals into overlapping windows.
2. Extract dominant frequencies using Welch's method.
3. Detect patterns with the OPM algorithm.
4. Compare pattern frequency across ADHD and CONTROL groups.
5. At the end, a summary plot is generated showing the top 10 most frequent patterns from the current OPM run across the base group.

🔁 5. CLI Option (Optional for Advanced Users)

For users who prefer the command-line interface, the pipeline can also be executed using:

python -m src.main

You will be prompted to choose between using default parameters or entering customized values.
🖨 Throughout the execution, printed messages will guide you step-by-step through the process.

📤 6. Outputs

All output files are automatically saved in the results/ directory. These include:

- **opm_runs/** – pattern detection results per subject
  Generated by: run_opm_pattern_search.py
- **cross_group_comparison/** – ADHD vs. Control pattern comparison
  Generated by: cross_group_pattern_check.py
- **single_run_plot/** – top 10 patterns summary plot (graphical view)
  Generated by: generate_single_run_plot.py
- **summary_plots/** – CSV summary of all OPM runs per window size
  Generated by: extract_top_patterns.py

🗂 Additional Directories Generated During Runtime

Several scripts automatically generate additional output directories, including:

- **converted_npy/** – stores EEG recordings converted from .mat to .npy
  Generated by: convert_raw_mat_to_npy.py
- **cleaned_npy/** – contains EEG data after preprocessing (filtering and ICA)
  Generated by: preprocess_clean_eeg.py
- **windowed_data_Xs/** – holds segmented EEG windows of X-second duration
  Generated by: segment_eeg_windows.py
- ***_freqs.npy** – dominant frequency of each EEG window
- ***_labels.npy** – corresponding symbolic label for each window (e.g., δ, θ, α)
  Generated by: extract_dominant_frequencies.py

All directories are created automatically by their respective scripts — no manual folder setup is required.

📃 Additional Output

A textual summary file of all detected patterns is also included for documentation and research purposes. These output files serve as the basis for validation, visualization, and comparison steps in the analysis pipeline.

📌7. Technical Notes

1. **EEG Channel Indexing**
   In this project, EEG channels are indexed from **0 to 18**, corresponding to the standard 1–19 channel configuration. This shift reflects Python's zero-based indexing convention.
2. **ICA Noise Removal Warnings**
   In rare cases, the ICA (Independent Component Analysis) process may not fully remove all noise artifacts.
   Occasionally, a warning may appear for a specific subject file.
   This has **no significant impact** on the overall OPM (Order Preserving Matching) analysis because the algorithm focuses on **patterns that appear consistently across the majority of subjects**. Isolated noise does not meet the inclusion criteria and is automatically disregarded during pattern selection.

🧑‍💻 8. Contact / Team

Authors: Tamir Nahari, Yaakov Shitrit