

SignalFlow EEG

E. Pedapati

2023-06-30

Table of contents

| | |
|---|---------------|
| Welcome | 4 |
| Core Features | 4 |
| I Let's Get Started | 5 |
| SignalFlow in a Nutshell | 6 |
| An Brief Example Workflow | 6 |
| 1 Introduction | 7 |
| 1.1 A Reproducible EEG Processing Framework | 7 |
| 1.2 Core Features | 7 |
| References | 9 |
| 2 Visual Evoked Potentials (VEP) Tutorial | 11 |
| 2.1 Overview | 11 |
| 3 Walkthrough | 12 |
| 3.1 Downloading SignalFlow from GitHub | 13 |
| 3.2 Overview | 13 |
| 3.3 Method 1: Using Git | 13 |
| 3.4 Method 2: Downloading Toolkits as Zip Files | 14 |
| 4 Preparing and Organizing Files | 16 |
| 4.1 Overview | 16 |
| 4.2 Using the BIDS format | 16 |
| 5 Starting the SignalFlow Application | 18 |
| 6 Setup Tab | 21 |
| 6.1 Adding Dependancies | 21 |
| 6.2 Adding Project Folder locations | 22 |
| 6.3 Adding Custom Project Folder locations | 22 |
| 6.4 Edit Project Information | 22 |
| 7 Builder Tab | 23 |
| 7.1 Filter Modules | 23 |

| | | |
|----------|---|-----------|
| 7.2 | Adding Modules | 23 |
| 7.3 | Copy and Edit Module TODO this might have a bug | 24 |
| 7.4 | Delete Module | 24 |
| 7.5 | Reorder Modules | 24 |
| 7.6 | Reset Pipeline | 25 |
| 7.7 | Change Module Information | 25 |
| 8 | Execute Tab | 26 |
| 8.1 | Refresh | 26 |
| 8.2 | Execute (Loop) | 27 |
| 8.3 | Execute (Parallel) | 27 |
| 8.4 | Execute (File) | 27 |
| 8.5 | Open Import Folder | 27 |
| 8.6 | Open Export Folder | 27 |
| 8.7 | Open in EEGLAB | 28 |
| 8.8 | Plot Data | 28 |
| 8.9 | View Electrode Map | 28 |
| 8.10 | View Channel Spectra | 28 |
| 9 | Browse Tab | 29 |
| 9.1 | Select Project Folder | 29 |
| 9.2 | Refresh | 30 |
| 9.3 | Open Selected Folder | 30 |
| 9.4 | Open in EEGLAB | 30 |
| 9.5 | Plot Data | 30 |
| 9.6 | View Electrode Map | 30 |
| 9.7 | View Channel Spectra | 31 |
| 9.8 | Close Windows | 31 |

Welcome

SignalFlowEEG (SF) is a comprehensive and versatile framework designed to streamline the processing of Electroencephalograms (EEG). Utilizing an array of different pipelines, SF aims to ensure reproducibility and consistency across EEG studies.

The overarching objective of SF is to generate robust data tables that facilitate summary generation and statistical analysis. These tables can encompass a wide range of EEG features, from intrinsic characteristics to preprocessing steps and sophisticated analysis.

Core Features

1. **Focus on Data Tables:** SignalFlowEEG generates tables encapsulating diverse aspects of EEG research, including EEG characteristics, preprocessing procedures, and comprehensive analysis.
2. **EEGLAB SET Format Compatibility:** SF employs the universally accepted EEGLAB SET format, providing an efficient data structure for EEG processing and encouraging interoperability with other tools.
3. **Audit Capabilities:** To maintain transparency and accountability, SF provides the ability to perform audits on logged actions. This feature allows researchers to trace back and review the sequence of operations applied on the EEG data.
4. **High-Performance File Formats:** For handling high-dimensional data such as connectivity matrices, SignalFlowEEG adopts high-performance file formats like Apache Parquet. This approach not only ensures data integrity but also facilitates speedy retrieval and efficient storage.

Part I

Let's Get Started

Before getting invested in a new workflow, I think it's important to understand if the results are a good fit for your laboratory.

In these introductory chapter we will give you a rapid overview of examples of some of the capabilities of SignalFlow and how they can be used to process your EEG data.

SignalFlow in a Nutshell

At it's essence, SignalFlow is a MATLAB toolbox provides a unified wrapper around open-source EEG toolkits (e.g., EEGLAB, FieldTrip, MNE, etc.) to process data and generate uniform outputs (tables and figures).

This flexibility should allow you to chose the best toolkit for a specific task, while still providing uniform outputs that can be analyzed similarly.

An Brief Example Workflow

Let's give a prototype example of a SignalFlow workflow that might be difficult in any single EEG toolkit.

1. MATLAB: Use the EEGLAB import functions to properly import an Magstim/EGI .mff file.
2. MATLAB: Use a visual pipeline to preprocess several subjects using EEGLAB's pre-processing functions including using Artifact Subspace Reconstruction (ASR) to remove artifacts.
3. MATLAB: Use an optimized parallel ICA function to run ICA on all subjects in parallel using a custom ICA binary that is optimized for your operating system.
4. MATLAB: Following preprocessing, use EEGLAB to save the files as SET files in a BIDS compatible format and save preprocessing details as a CSV file.
5. Use the artifact-free SET to perform analyzsers across toolkits but create R-statistics compatible tables.
 - a. Use MATLAB to generate band power tables
 - b. Use Python to generate connectivity tables
 - c. Use R to import the tables and run statistics and perform visualization.

We will go through variations of this workflow in the following sections to give a broad overview of the capabilities of SignalFlow.

1 Introduction



1.1 A Reproducible EEG Processing Framework

SignalFlowEEG (SF) is a comprehensive and versatile framework designed to streamline the processing of Electroencephalograms (EEG). Utilizing an array of different pipelines, SF aims to ensure reproducibility and consistency across EEG studies.

The overarching objective of SF is to generate robust data tables that facilitate summary generation and statistical analysis. These tables can encompass a wide range of EEG features, from intrinsic characteristics to preprocessing steps and sophisticated analysis.

1.2 Core Features

1. **Focus on Data Tables:** SignalFlowEEG generates tables encapsulating diverse aspects of EEG research, including EEG characteristics, preprocessing procedures, and comprehensive analysis.
2. **EEGLAB SET Format Compatibility:** SF employs the universally accepted EEGLAB SET format, providing an efficient data structure for EEG processing and encouraging interoperability with other tools.
3. **Audit Capabilities:** To maintain transparency and accountability, SF provides the ability to perform audits on logged actions. This feature allows researchers to trace back and review the sequence of operations applied on the EEG data.

4. **High-Performance File Formats:** For handling high-dimensional data such as connectivity matrices, SignalFlowEEG adopts high-performance file formats like Apache Parquet. This approach not only ensures data integrity but also facilitates speedy retrieval and efficient storage.

References

Part II

2 Visual Evoked Potentials (VEP) Tutorial

In this tutorial, we will guide you on how to use **SignalFlowEEG** to preprocess and generate ERP (Event-Related Potential) tracings from a VEP (Visual Evoked Potential) dataset using the HAPPE pipeline.

2.1 Overview

In this walkthrough, we will guide you on how to use **SignalFlowEEG** to preprocess and generate ERP (Event-Related Potential) tracings from a VEP (Visual Evoked Potential) dataset using the HAPPE pipeline.

In the following sections, we will dive deeper into each of these steps:

1. **Setting up SignalFlowEEG and its Dependencies:** Learn how to install and configure SignalFlowEEG and its dependencies.
2. **Arrange Files for Optimized Processing:** We'll begin by organizing our files in a way that optimizes SignalFlowEEG's processing capabilities.
3. **Preprocess Files Using HAPPE Pipeline:** Next, we'll utilize the popular HAPPE pipeline to preprocess our VEP dataset.
4. **Output Data Tables:** After preprocessing, we will output three data tables:
 - **EEG Characteristics Table:** This table will present the EEG characteristics of our VEP data, such as original length.
 - **Preprocessing Results Table:** This table will showcase the results of our preprocessing steps, including the number of epochs.
 - **Amplitude Table for ERPs:** This table will contain the amplitudes necessary to generate ERPs from our VEP dataset.
5. **Modify Preprocessing Step to Observe Effect on ERPs:** Lastly, we will modify one of our preprocessing steps and examine its impact on the resulting ERPs.

Let's get started!

3 Walkthrough

1. Install SignalFlow and other toolkits. See Section [3.1](#).
2. Organize EEG files. See Chapter [4](#).
3. Start the GUI. See Chapter [5](#) for more details.
4. Setup toolboxes, dependancies, folders, and project. See Chapter [6](#) for more details.
5. Importing data and setting up project folders
6. Navigate to the Builder tab. See Chapter [7](#) for more details.
7. Add the following modules in order, for information on adding modules to the pipeline see Section [7.2](#).
 1. Import **Insert EEG file type** File
 2. Resample EEG data (500 Hz)
 3. Highpass Filter (0.3 Hz)
 4. Lowpass Filter (35 HZ)
 5. Wavelet Denoise
 6. Artifact Subspace Reconstruction
 7. Channel Interpolation
 8. TODO Add ERP to SignalFlow
 9. Export Set
8. Navigate to the Execute tab. See Chapter [8](#) for more details.
9. Left-click the “Execute (Loop)” button.
10. Wait for pipeline to complete. You should be able to watch the Matlab Command Window.
11. Navigate to the Browse tab. See Chapter [9](#) for more details.
12. View your data using the tools provided.

3.1 Downloading SignalFlow from GitHub

3.2 Overview

The source code for SignalFlow is hosted on GitHub. There are two primary methods to acquire the code - either through utilizing Git software or directly downloading from the website. However, due to the inclusion of submodules pertaining to other necessary programs, we highly recommend the use of Git for downloading SignalFlow. The direct web download approach requires manual downloading and arrangement of each piece of software within the source code directory, which could prove to be a more time-consuming process.

3.3 Method 1: Using Git

1. Check if Git is installed:

Open a terminal or command prompt and run the following command:

```
git --version
```

If Git is installed, you will see the version number. If not, proceed to the next step.

2. Install Git:

Download the install from the official Git website for your operating system at [official Git website](#).

3. Create a ‘github’ directory:

Choose a location on your drive that will be used to store your toolboxes.

Run use your file explorer or run following command to create a new directory called ‘github’ and navigate to it:

```
mkdir github  
cd github
```

4. Clone the SignalFlow repository with submodules:

Run the following command to clone the repository and its submodules:

```
git clone --recurse-submodules https://github.com/cincibrainlab/SignalFlowEEG_HBCD.git
```

This will clone the repository and its submodules into the ‘github’ directory. The output will be displayed in the terminal or command prompt.

5. **Clone other necessary toolkits:** In addition to SignalFlowEEG, you may need to clone several other toolkits. Here are the necessary commands:

- **EEGLAB:**

```
git clone --recurse-submodules https://github.com/sccn/eeglab.git
```

- **Brainstorm:**

```
git clone --recurse-submodules https://github.com/brainstorm-tools/brainstorm3.git
```

- **HAPPE Pipeline:**

```
git clone --recurse-submodules https://github.com/PINE-Lab/HAPPE.git
```

- **MADE Pipeline:**

```
git clone --recurse-submodules https://github.com/ChildDevLab/MADE-EEG-preprocessing-
```

After cloning these repositories, you'll have all the necessary toolkits for your EEG data analysis. Make sure to keep these repositories up-to-date by pulling the latest changes regularly by using the command:

```
git pull --recurse-submodules
```

3.4 Method 2: Downloading Toolkits as Zip Files

If you prefer, you can download each repository individually as a zip file. This can be done from the main page of each repository on GitHub. Note that this method will require additional steps to organize the downloaded content.

Here are the steps to download and organize the repositories:

1. Navigate to each repository's GitHub page in your web browser.
2. Click on the **Code** button and then click **Download ZIP**.
3. Once the download is complete, extract the ZIP file.
4. Create a new folder in your **github** directory for each extracted repository. The name of the folder should *match* the repository name.

Here are the links to each repository: - **SignalFlow**: https://github.com/cincibrainlab/SignalFlowEEG_HBCD.
- **VHTP**: <https://github.com/cincibrainlab/vhtp> - **VHTP Dependencies**: <https://github.com/cincibrainlab/v>
- **HAPPE Pipeline**: <https://github.com/PINE-Lab/HAPPE> - **MADE Pipeline**:
<https://github.com/ChildDevLab/MADE-EEG-preprocessing-pipeline>

Note: For the VHTP and VHTP Dependencies, make sure to place these folders in the **SignalFlowEEG_HBCD** folder, as they are necessary for certain SignalFlow operations.

EEGLAB and Brainstorm must be downloaded through their respective non-github websites. This is because any submodules cannot be downloaded through the github website without git.

- **EEGLAB**: <https://scn.ucsd.edu/eeglab/index.php>
- **Brainstorm**: <https://neuroimage.usc.edu/brainstorm/Introduction>

After extraction, you will have the complete repository with all its files and folders available on your computer.

4 Preparing and Organizing Files

4.1 Overview

When working with complex data like EEG datasets, a clear and concise directory structure is essential for efficient data management and processing. A well-organized file structure improves reproducibility and ease of understanding for others (and your future self).

Considering the diverse storage requirements of different processes, we suggest splitting up your directories to increase versatility. Our recommendation includes creating three main directories: **data**, **temp**, and **results**.

These directories should be nested within a project-specific folder to separate and manage multiple projects effortlessly. We also advise creating a **readme.txt** file in the main folder and describing the contents and purpose of the project and folders.

Here's a simple, recommended structure:

- ProjectName/
 - readme.txt
 - data/
 - temp/
 - results/

The **data** directory should contain all the raw data that will be processed. The **temp** directory will store any temporary files created during data processing, and the **results** directory will hold the output from your analyses.

4.2 Using the BIDS format

The *Brain Imaging Data Structure (BIDS)* is a formal way to name and organize neuroimaging data and metadata in a file system. The goal is to simplify communication and collaboration among users. The format enables easier data validation and software development through consistent paths and naming for data files.

SignalFlow does not require the use of BIDS, but it maybe useful for certain projects. You can incorporate the BIDS structure into the project organization strategy described above by placing a BIDS-compliant data folder inside the data directory.

```
- ProjectName/
- readme.txt
- data/
  - BIDS/
    - sub-01/
      - ses-01/
        - eeg/
          - sub-01_ses-01_task-mytask_eeg.edf
          - sub-01_ses-01_task-mytask_channels.tsv
          - sub-01_ses-01_task-mytask_electrodes.tsv
          - sub-01_ses-01_task-mytask_events.tsv
          - ...
        - ...
      - temp/
      - results/
```

SignalFlow relies on helper functions from the Cincinnati Visual High Throughput Pipeline (vHTP). The standard file list utility ([util_httpDirListing.m](#)) can create a custom file list from a BIDS directory structure.

Remember, a well-planned and maintained directory structure can save time and prevent confusion as your project grows.

5 Starting the SignalFlow Application

Follow the steps below to launch the SignalFlow graphical user interface (GUI) or Application via MATLAB:

1. **Open MATLAB.**

Launch the MATLAB software from your applications menu or desktop shortcut.

2. **Navigate to the SignalFlow Directory.**

The SignalFlow directory is typically located within the `GitHub` folder in your `Documents`. This might vary depending on where you cloned the repository. The paths are generally as follows:

- For Windows: `C:\Users**YourUsername**\Documents\github\SignalFlow_HBCD`
- For Linux or macOS: `/home/**YourUsername**/Documents/github/SignalFlow_HBCD`

Replace `**YourUsername**` with your actual username on your computer.

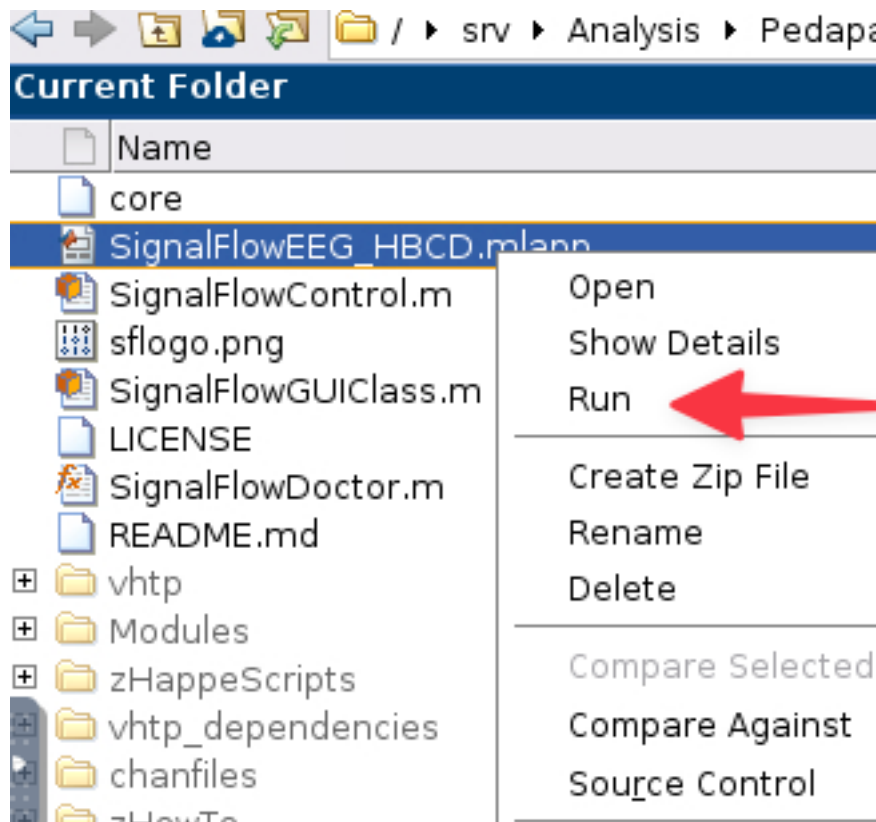
To navigate to this directory in MATLAB, you can use the `cd` (change directory) command in MATLAB's Command Window:

```
cd('path-to-directory')
```

Replace `'path-to-directory'` with the appropriate path above.

3. **Find the SignalFlow application file.**

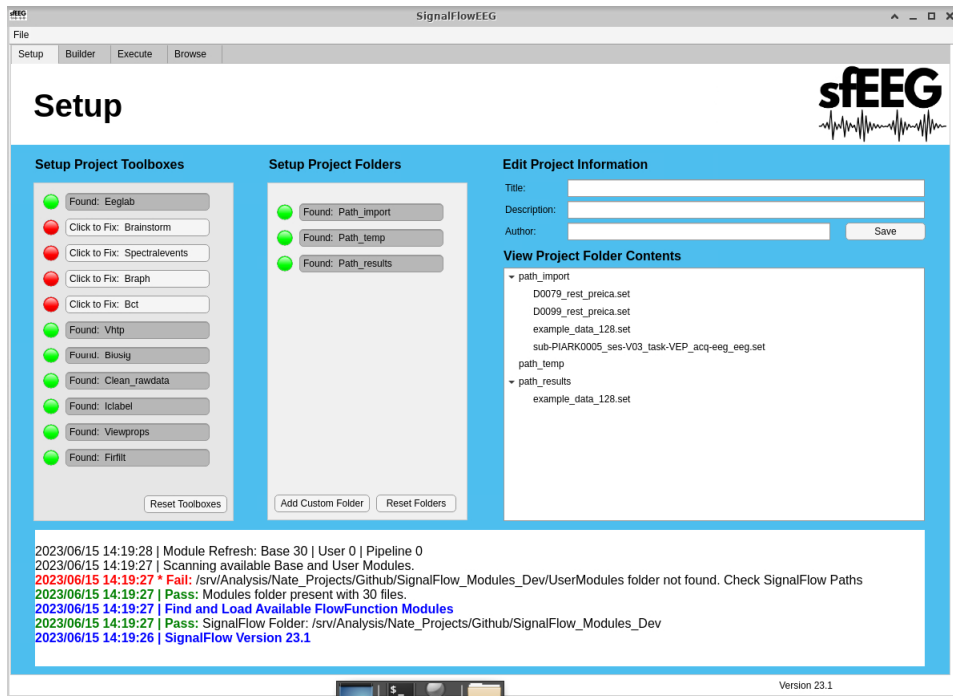
In MATLAB's Current Folder pane (typically located on the left side of the interface), scroll to locate the SignalFlow application file named `"SignalFlowEEG.mlapp"`.



4. **Run the SignalFlow application.**

Right-click on the “SignalFlowEEG.mlapp” file and select “Run” from the context menu.

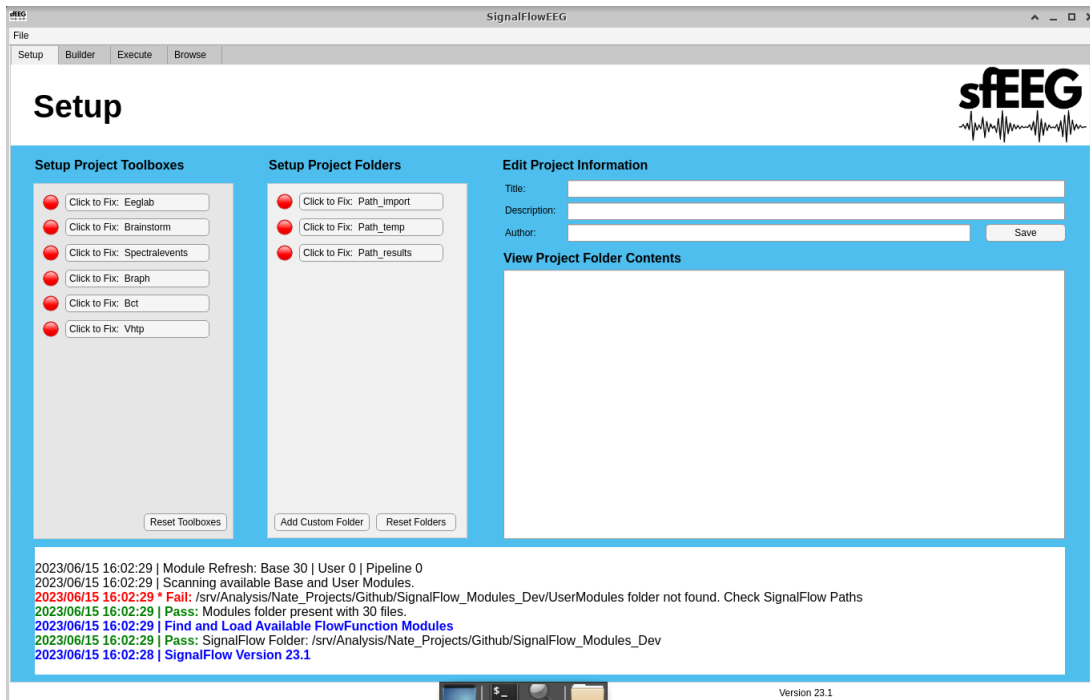
5. **Check SignalFlow Application.**



The SignalFlow Application should now open. If the GUI does not open or you encounter an error, ensure that you have the correct directory and the “SignalFlowEEG.mlapp” file is present.

With these steps, you should be able to successfully launch the SignalFlow EEG Application in MATLAB.

6 Setup Tab



* The setup tab is the starting page for every user. It allows the user to configure their workstation and pipeline to their individual needs.

6.1 Adding Dependancies

1. Navigate to the “Setup” Tab.
2. Decide which toolbox/dependency you would like to add . A. A Red circle indicated the toolbox/dependency has not been added. B. A Green circle indicates that the toolbox/dependency has been added.
3. Left-click on the “Click to Fix: *Insert Toolbox*”.
4. File Explorer will open.
5. Select the directory that contains your toolbox. Click “Open” in the file explorer. A. This can vary based on the toolbox, contact us if you have issues .

6. The circle should turn from red to green, may take a second due to SignalFlow checking if it's the right toolbox.

6.2 Adding Project Folder locations

1. Navigate to the “Setup” Tab.
2. Decide which folder you would like to add. A. A Red circle indicated the folder has not been added. B. A Green circle indicates that the folder has been added.
3. Left-click on the “Click to Fix: *Insert folder*”.
4. File Explorer will open.
5. Select the directory you would like to add. Left-click “Open” in the file explorer.
6. The circle will turn from red to green.

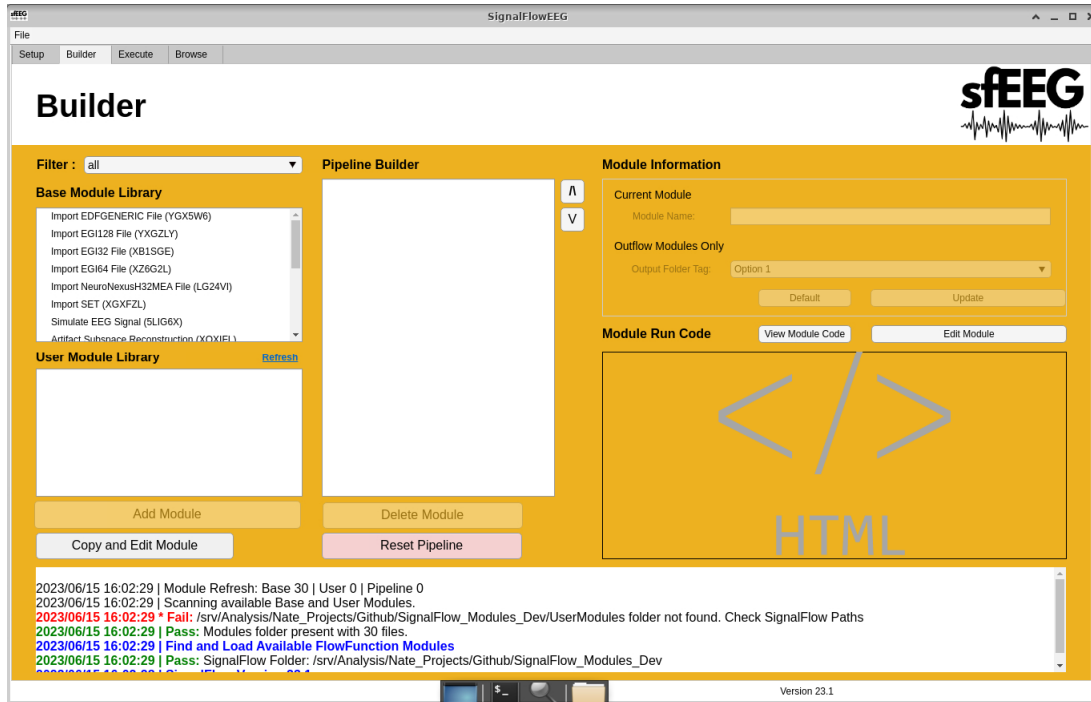
6.3 Adding Custom Project Folder locations

1. Navigate to the “Setup” Tab.
2. Left-click the “Add Custom Folder” button.
3. A Matlab figure will open.
4. Left-click the “Select Directory” button.
5. Select the directory you would like to add. Left-click “Open” in the file explorer.
6. Enter a tag. A tag is an easily recognizable name used for selecting this Project Folder later.
7. Left-click the “Confirm” button.
8. A new project folder will be added. The circle will turn from red to green.

6.4 Edit Project Information

1. Navigate to the “Setup” Tab.
2. Enter a Title for your project.
3. Enter a Description for you project.
4. Enter an Author for your project.
5. Left-click the “Save” button.

7 Builder Tab



* The

Builder tab is used to create the pipeline and edit modules. Most work is done in this tab.

7.1 Filter Modules

1. Navigate to the “Builder” Tab.
2. Left-click the filter dropdown.
3. Select the filter you would like to use.
4. Module Libraries will only show modules that agree with your filter.

7.2 Adding Modules

1. Navigate to the “Builder” Tab.
2. Left-click the module you would like to use.

- A. This can be from the both the Base and User module libraries.
3. Below the Base and User module libraries, there is an “Add Module” button, left-click it.
4. The module will the be added to the pipeline. It will show up in the Pipeline Builder.

7.3 Copy and Edit Module TODO this might have a bug

1. Navigate to the “Builder” Tab.
2. Left-click the module you would like to use.
 - A. This can be from the both the Base and User module libraries.
3. Below the Base and User module libraries, there is an “Add Module” button, left-click it.
4. The module will then be opened in Matlab.
5. Edit any parameters or code.
6. Save the Module file.
7. The Module file should show up in the User module library.

7.4 Delete Module

1. Navigate to the “Builder” Tab.
2. Left-click the module you would like to delete.
 - A. This can only be from the Pipeline Builder
3. Below the Pipeline Builder, there is an “Delete Module” button, left-click it.
4. The Module will be removed from the pipeline.

7.5 Reorder Modules

1. Navigate to the “Builder” Tab.
2. Left-click the module you would like to reorder.
 - A. This can only be from the Pipeline Builder.
3. To the right of the Pipeline Builder, there are 2 arrows.
4. Left-click the arrow that corresponds with the direction you would like to move the module.
5. The Module will be reordered in the pipeline.

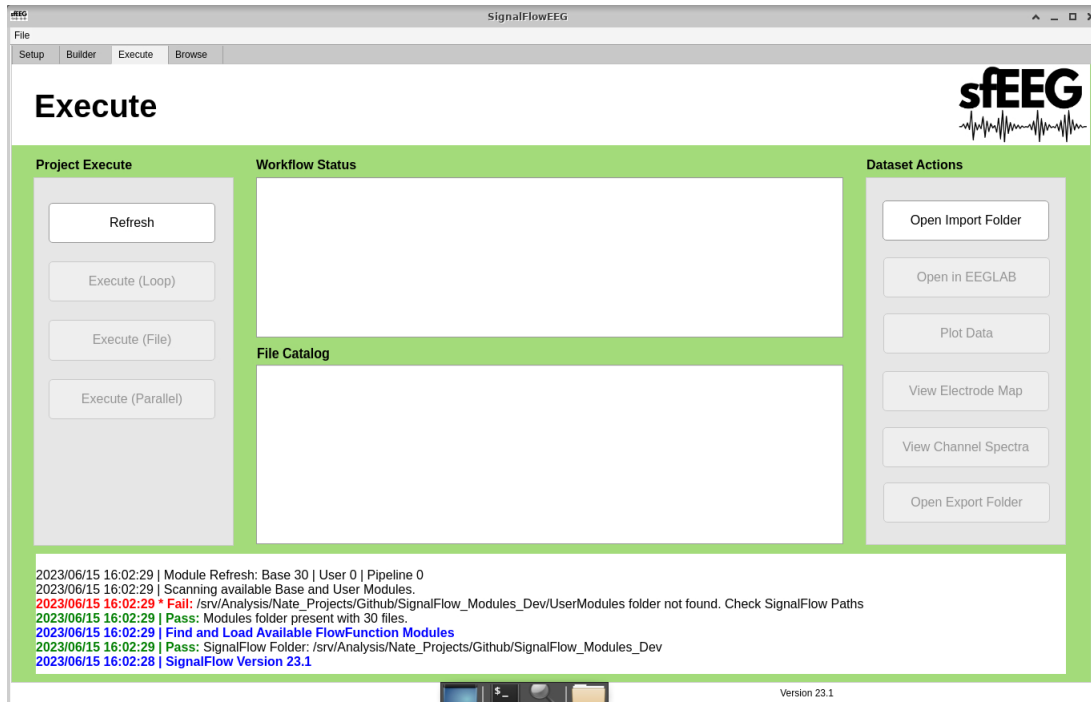
7.6 Reset Pipeline

1. Navigate to the “Builder” Tab.
2. Left-click the “Reset Pipeline” button.
3. All Modules within the Pipeline Builder will be deleted.

7.7 Change Module Information

1. Navigate to the “Builder” Tab.
2. Enter the new Module Name.
3. If the module is an outflow module. A. Select the needed output folder tag in the dropdown.
4. Left-click the “Update” button.
5. Your changes will be saved.
6. If you would like to return to default name and folder tag A. Left-click the “Default” button.

8 Execute Tab



* The execute tab is used to execute the pipeline on EEG files. Users also have buttons that serve as helpful tools.

8.1 Refresh

- This button is used to refresh the workflow status and file catalog when a change to the file system or pipeline has occurred.
 - A. An example of this would be adding a file to the import directory. The user must click the “Refresh” button in order to see it in the File Catalog.

8.2 Execute (Loop)

- This button is used to execute the pipeline on all eeg files in the import directory. These will be done one at a time.
- Use this if there are steps that require user input. A. Examples of this include: Remove channels, remove epochs, etc.

8.3 Execute (Parallel)

- This button is used to execute the pipeline on all eeg files in the import directory. These will be done all at the same time.
- This should only be used when the following are true. A. Use this if there are no steps that require user input. Examples of this include: Remove channels, remove epochs, etc. B. You have a high performance computer. C. The pipeline takes longer than 1 minutes to run on a single eeg file. The setup and tear down of workers may take some time. inefficient use could make the processing time longer. D. You have the Matlab Parallel Processing Toolbox

8.4 Execute (File)

- This button is used to execute the pipeline on a selected eeg file in the import directory.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “Execute (File)” button.

8.5 Open Import Folder

- This button is used to open the import directory in the computers file explorer.

8.6 Open Export Folder

- This button is used to open the selected directory in the computers file explorer.
- To use this button:
 1. Left-click your desired directory in the File Catalog.
 2. Left-click the “Open Export Folder” button.

8.7 Open in EEGLAB

- This button is used to open a selected eeg file in the EEGLAB gui.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “Open in EEGLAB” button.

8.8 Plot Data

- This button is used to plot data for a selected eeg file.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “Plot Data” button.

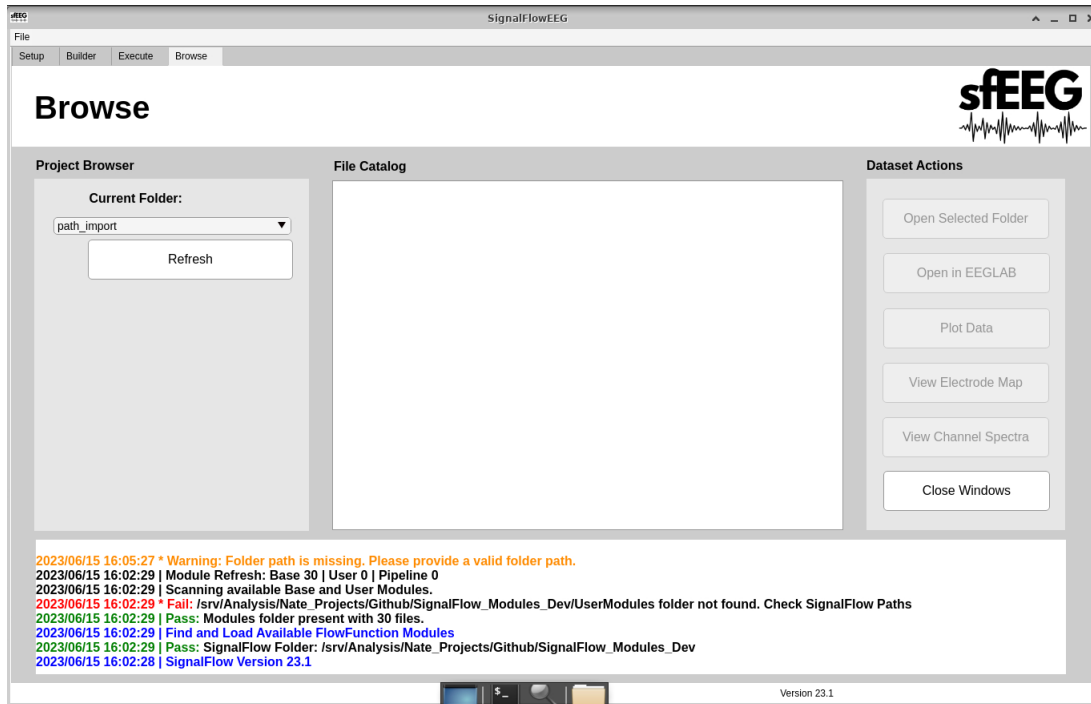
8.9 View Electrode Map

- This button is used to view the electrode map for a selected eeg file.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “View Electrode Map” button.

8.10 View Channel Spectra

- This button is used to view the channel spectra for a selected eeg file.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “View Channel Spectra” button.

9 Browse Tab



* The browse tab is used to analyze and look at data. The user has access to many tools to aid with this.

9.1 Select Project Folder

1. Navigate to the “Browse” Tab.
2. Left-click the current folder dropdown.
3. Select the folder you would like to see.
4. The file catalog will be updated based on your selection.

9.2 Refresh

- This button is used to refresh the file catalog when a change to the file system or pipeline has occurred.
 - A. An example of this would be adding a file to the import directory. The user must click the “Refresh” button in order to see it in the File Catalog.

9.3 Open Selected Folder

- This button is used to open the selected directory in the computer's file explorer.
- To use this button:
 1. Left-click your desired directory in the File Catalog.
 2. Left-click the “Open Export Folder” button.

9.4 Open in EEGLAB

- This button is used to open a selected eeg file in the EEGLAB gui.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “Open in EEGLAB” button.

9.5 Plot Data

- This button is used to plot data for a selected eeg file.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “Plot Data” button.

9.6 View Electrode Map

- This button is used to view the electrode map for a selected eeg file.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “View Electrode Map” button.

9.7 View Channel Spectra

- This button is used to view the channel spectra for a selected eeg file.
- To use this button:
 1. Left-click your desired file in the File Catalog.
 2. Left-click the “View Channel Spectra” button.

9.8 Close Windows

- This button is used to close all matlab figures/windows. Matlab and the SignalFlowEEG GUI will not be closed.