

SignalFlow EEG

E. Pedapati

2023-06-30

Table of contents

Welcome	3
Core Features	3
Let's Get Started	4
SignalFlow in a Nutshell	4
An Brief Example Workflow	4
1 Installing SignalFlow	6
1.1 Overview	6
1.2 Method 1: Using Git	6
1.3 Method 2: Downloading Toolkits as Zip Files	7
2 Preparing Data Files	9
2.1 Overview	9
2.2 Using the BIDS format	9
3 Starting SignalFlow	11
4 Mouse EEG Tutorial	14
4.1 Overview	14
4.2 Setting up your environment	14
4.2.1 Installing SignalFlowEEG	14

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.

Let's Get Started

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 and provides a unified wrapper around opensource EEG toolkits (e.g., EEGLAB, FieldTrip, MNE, etc.) to process data and generate uniform outputs (tables and figures). Signalflow can handle EEG data from human recordings but also from other signal sources such as `@sec-mouse-eeg`.

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 (CSV) 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 Installing SignalFlow

1.1 Overview

The source code for SignalFlow is hosted on GitHub. There are two primary methods to install SignalFlow:

1. Using Git software to directly download the code
2. Downloading the code as a zip file from the GitHub 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.

1.2 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/scn/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
```

1.3 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.

2 Preparing Data Files

2.1 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.

2.2 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.

3 Starting SignalFlow

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.

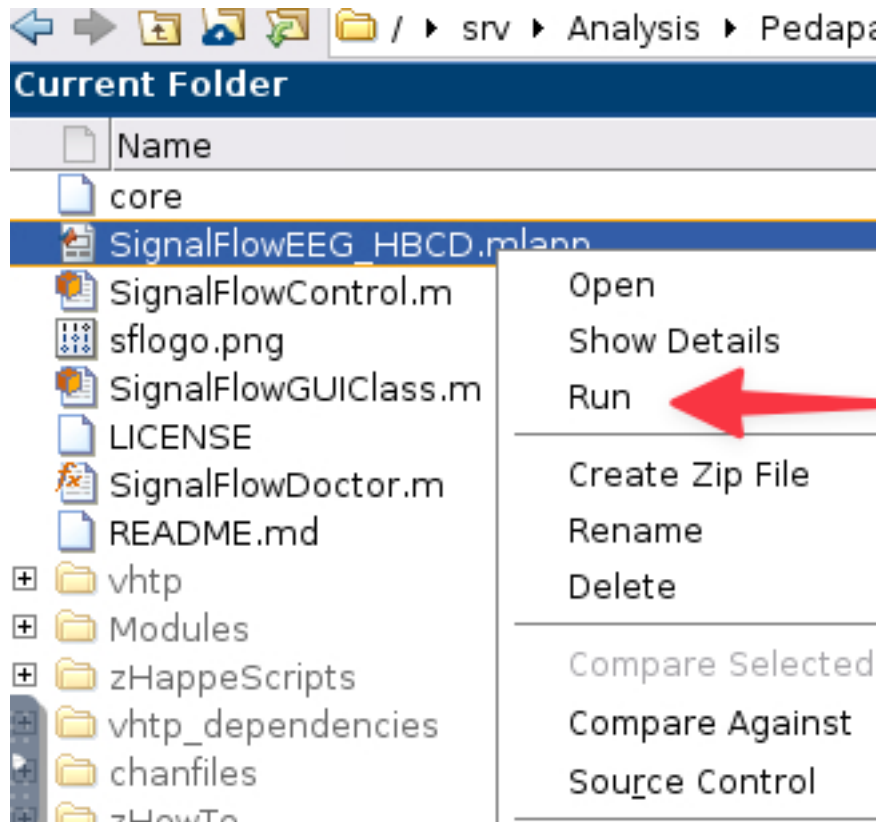
To navigate to this directory in MATLAB, you can use the `cd` (change directory) command in MATLAB's Command Window:

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

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

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

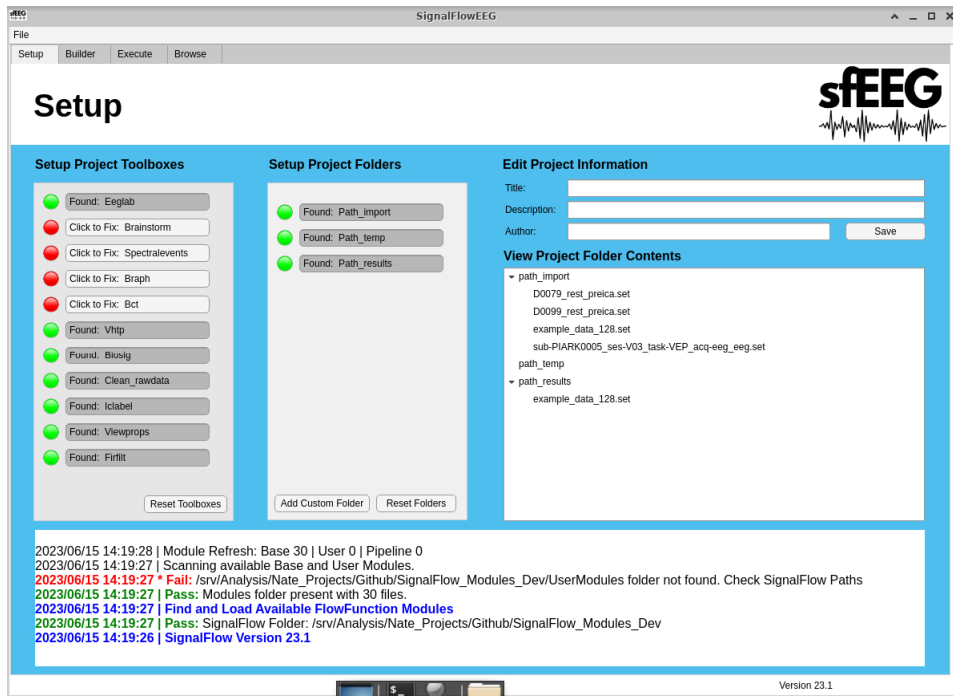
In MATLAB's Current Folder pane (typically located on the left side of the interface), scroll to locate the SignalFlow application file named `"SignalFlowEEG.mlapp"`.



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.

6. Check dependencies

If you check the command window, you will see that SignalFlow will do a careful check for any missing dependencies. If you encounter any errors, revisit the Chapter 1 section to ensure that you have installed all the necessary dependencies.

The command `SignalFlowDoctor` will do a careful check for missing toolkits and also have options for repair. Restart SignalFlow after installing any missing dependencies.

```
![The SignalFlow "Doctor"](..\assets\images\SignalFlowDoctor.png)
```

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

4 Mouse EEG Tutorial

In this tutorial, we will guide you on how to use **SignalFlowEEG** analyze EEG tracings from mouse multielectrode array (MEA).

Specifically, we will analyzing data from the Neuronexus Mouse EEG v2 H32 probe is a 32-channel EEG MEA specifically designed for mice. Combined with the Omnetics Amplifier Adapter, it provides a complete system for acquiring high-fidelity EEG data from the mouse brain.

4.1 Overview

In this walkthrough, we will cover the following topics:

1. Setting up your environment
2. Loading data
3. Outputting data
4. Visualize data
5. Statistical Analysis
6. Reporting

4.2 Setting up your environment

4.2.1 Installing SignalFlowEEG

See [Chapter 1](#).