SignalFlow EEG

E. Pedapati

2023-06-30

Table of contents

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 essense, 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 preprocessing 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 analyzses 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.gi
   ```

   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
  ```

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://sccn.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_htpDirListing.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.

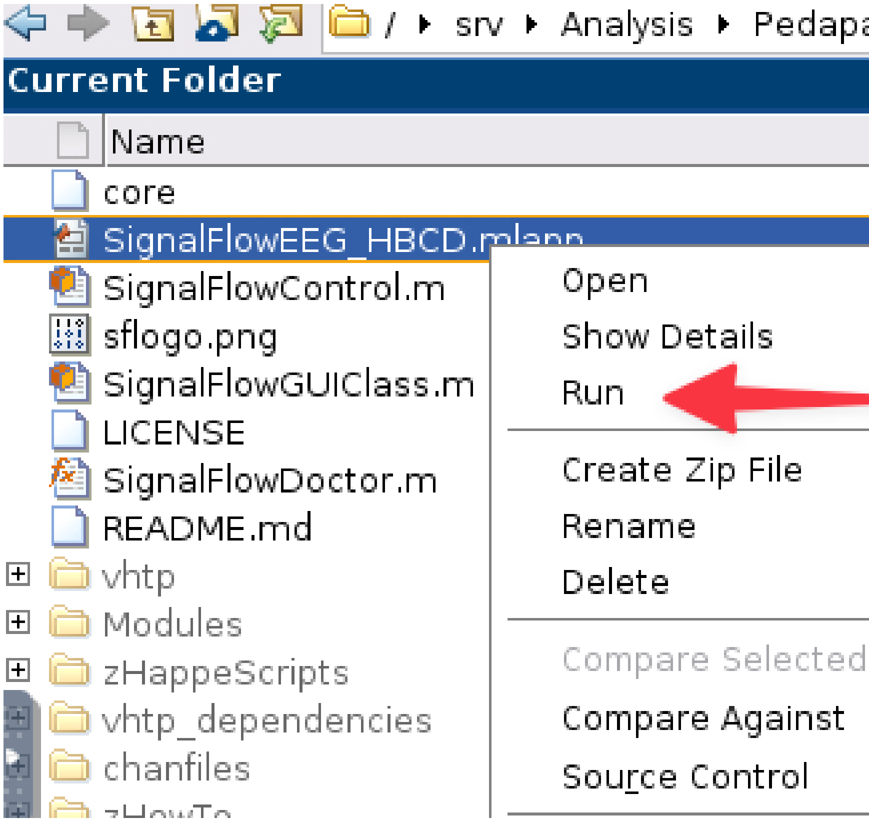
   To navigate to this directory in MATLAB, you can use the `cd` (change directory) command in MATLAB's Command Window:

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

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

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

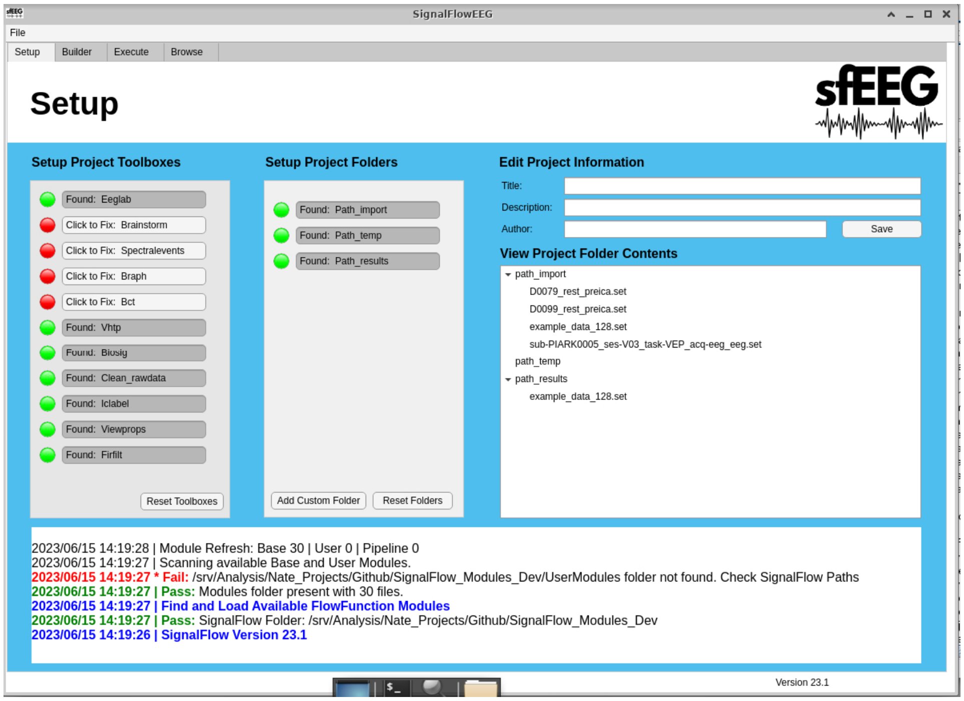
   In MATLAB's Current Folder pane (typically located on the left side of the interface), scroll to locate the SignalFlow application file named "SignalFlowEEG.mlapp".



4. **Run the SignalFlow application.**

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

5. **Check SignalFlow Application.**



```
▾ path_import
    D0079_rest_preica.set
    D0099_rest_preica.set
    example_data_128.set
    sub-PIARK0005_ses-V03_task-VEP_acq-eeg_eeg.set
  path_temp
▾ path_results
    example_data_128.set
```

```
2023/06/15 14:19:28 | Module Refresh: Base 30 | User 0 | Pipeline 0
2023/06/15 14:19:27 | Scanning available Base and User Modules.
2023/06/15 14:19:27 * Fail: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev/UserModules folder not found. Check SignalFlow Paths
2023/06/15 14:19:27 | Pass: Modules folder present with 30 files.
2023/06/15 14:19:27 | Find and Load Available FlowFunction Modules
2023/06/15 14:19:27 | Pass: SignalFlow Folder: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev
2023/06/15 14:19:26 | SignalFlow Version 23.1
```

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.

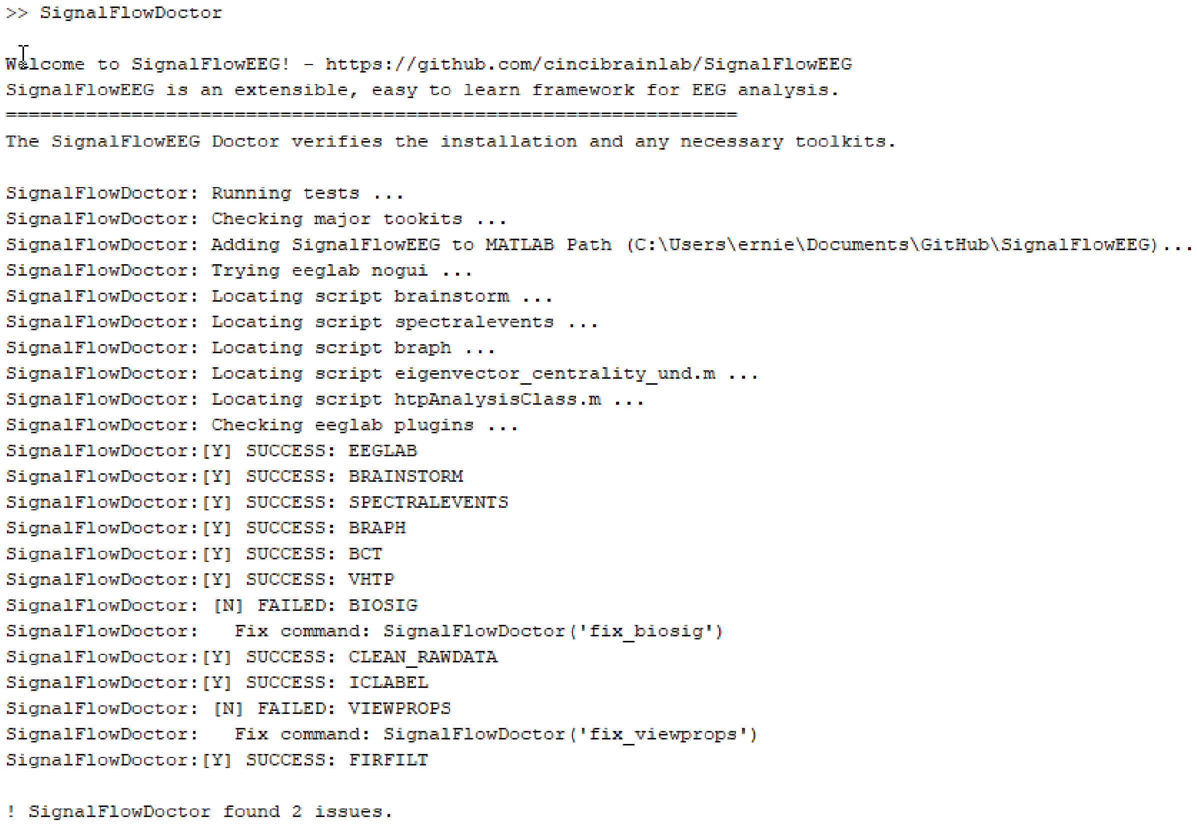
6. **Check dependencies**

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, revist 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.

In this case, `SignalFlowDoctor` identifies two issues as missing EEGLAB toolkits. Running the recommended commands in the command window will install the missing toolkits. If a plugin cannot be installed automatically, you may need to open EEGLAB and install it though the plugin manager.

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



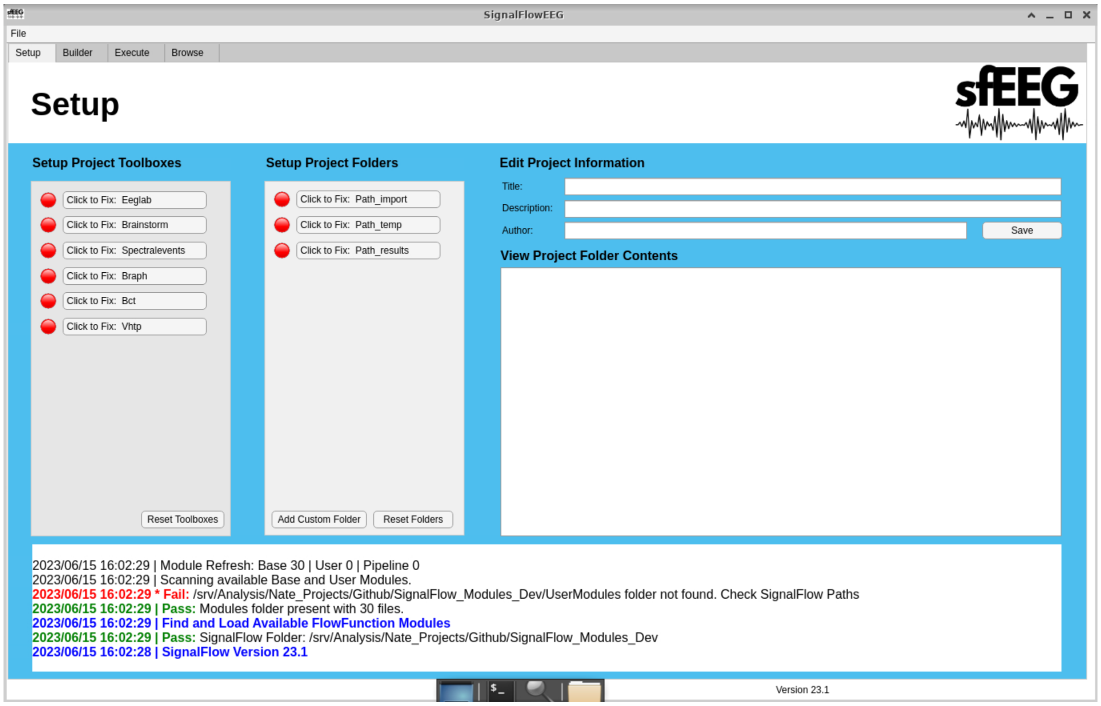
```
>> SignalFlowDoctor

Welcome to SignalFlowEEG! - https://github.com/cincibrainlab/SignalFlowEEG
SignalFlowEEG is an extensible, easy to learn framework for EEG analysis.
================================================================
The SignalFlowEEG Doctor verifies the installation and any necessary toolkits.

SignalFlowDoctor: Running tests ...
SignalFlowDoctor: Checking major tookits ...
SignalFlowDoctor: Adding SignalFlowEEG to MATLAB Path (C:\Users\ernie\Documents\GitHub\SignalFlowEEG)...
SignalFlowDoctor: Trying eeglab nogui ...
SignalFlowDoctor: Locating script brainstorm ...
SignalFlowDoctor: Locating script spectralevents ...
SignalFlowDoctor: Locating script braph ...
SignalFlowDoctor: Locating script eigenvector_centrality_und.m ...
SignalFlowDoctor: Locating script htpAnalysisClass.m ...
SignalFlowDoctor: Checking eeglab plugins ...
SignalFlowDoctor:[Y] SUCCESS: EEGLAB
SignalFlowDoctor:[Y] SUCCESS: BRAINSTORM
SignalFlowDoctor:[Y] SUCCESS: SPECTRALEVENTS
SignalFlowDoctor:[Y] SUCCESS: BRAPH
SignalFlowDoctor:[Y] SUCCESS: BCT
SignalFlowDoctor:[Y] SUCCESS: VHTP
SignalFlowDoctor: [N] FAILED: BIOSIG
SignalFlowDoctor:   Fix command: SignalFlowDoctor('fix_biosig')
SignalFlowDoctor:[Y] SUCCESS: CLEAN_RAWDATA
SignalFlowDoctor:[Y] SUCCESS: ICLABEL
SignalFlowDoctor: [N] FAILED: VIEWPROPS
SignalFlowDoctor:   Fix command: SignalFlowDoctor('fix_viewprops')
SignalFlowDoctor:[Y] SUCCESS: FIRFILT

! SignalFlowDoctor found 2 issues.
```

Figure 3.1: SignalFlowDoctor

4 SignalFlow: Setup Tab

The setup tab is the starting page for every user. It allows the user to configure their workstation and pipeline. This includes adding the correct paths to toolboxes and also adding folder locations for data.

In SignalFlow, assigned folders are given tags. These tags are used to reference the folder locations in the pipeline. This allows the user to change the folder location without having to hardcode a foldername into their code. This is useful when sharing code with other users.



```
2023/06/15 16:02:29 | Module Refresh: Base 30 | User 0 | Pipeline 0
2023/06/15 16:02:29 | Scanning available Base and User Modules.
2023/06/15 16:02:29 * Fail: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev/UserModules folder not found. Check SignalFlow Paths
2023/06/15 16:02:29 | Pass: Modules folder present with 30 files.
2023/06/15 16:02:29 | Find and Load Available FlowFunction Modules
2023/06/15 16:02:29 | Pass: SignalFlow Folder: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev
2023/06/15 16:02:28 | SignalFlow Version 23.1
```

Figure 4.1: SetupTab

4.1 Adding toolboxes to the MATLAB path

1. Navigate to the “Setup” Tab.

2. Decide which toolbox/dependency you would like to add to the MATLAB path. 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.

4.2 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.

4.3 Adding Custom Projct Folder locations

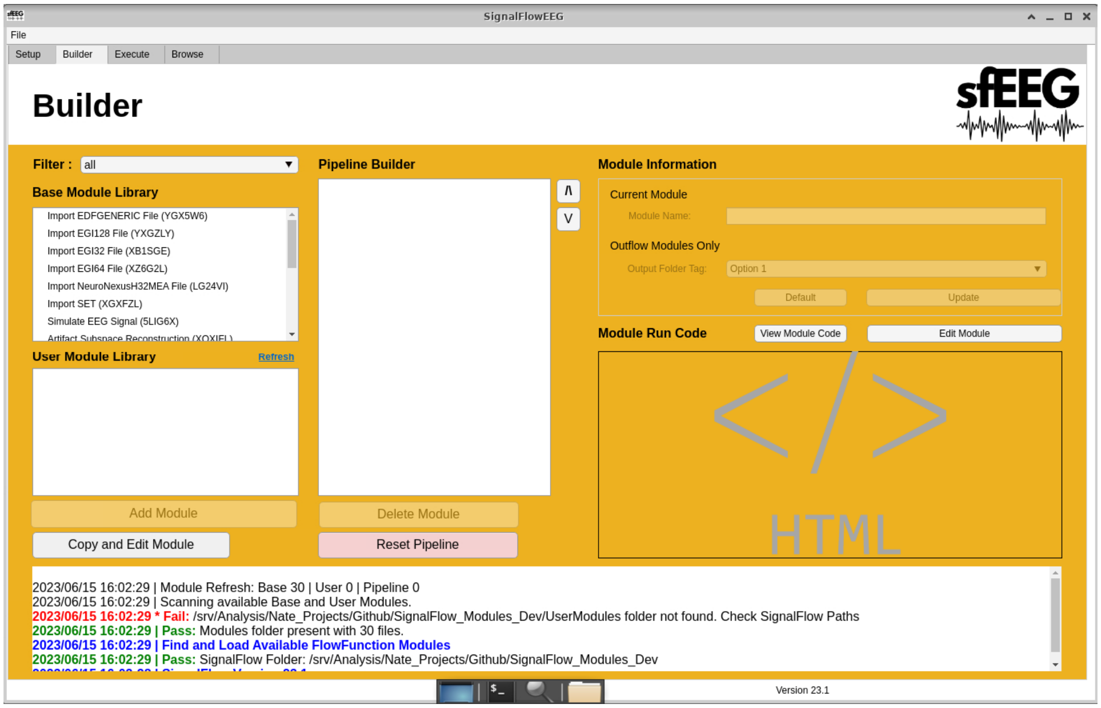
1. Navigate to the "Setup" Tab.
2. Left-click the "Add Custom Folder" button.
3. A Matlab figure will open.
4. Left-click the "Select Directory" button.
5. Select the directory you would like to add. Left-click "Open" in the file explorer.
6. Enter a tag. A tag is an easily recognizable name used for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

4.4 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.

5 SignalFlow: Builder Tab

The Builder Tab is where you can build your pipeline. You can add, delete, reorder, and edit modules. You can also filter modules to make it easier to find the module you are looking for.



Filter : all

Output Folder Tag: Option 1

```
2023/06/15 16:02:29 | Module Refresh: Base 30 | User 0 | Pipeline 0
2023/06/15 16:02:29 | Scanning available Base and User Modules.
2023/06/15 16:02:29 * Fail: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev/UserModules folder not found. Check SignalFlow Paths
2023/06/15 16:02:29 | Pass: Modules folder present with 30 files.
2023/06/15 16:02:29 | Find and Load Available FlowFunction Modules
2023/06/15 16:02:29 | Pass: SignalFlow Folder: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev
```

5.1 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.

5.2 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.

5.3 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.

5.4 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.

5.5 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.

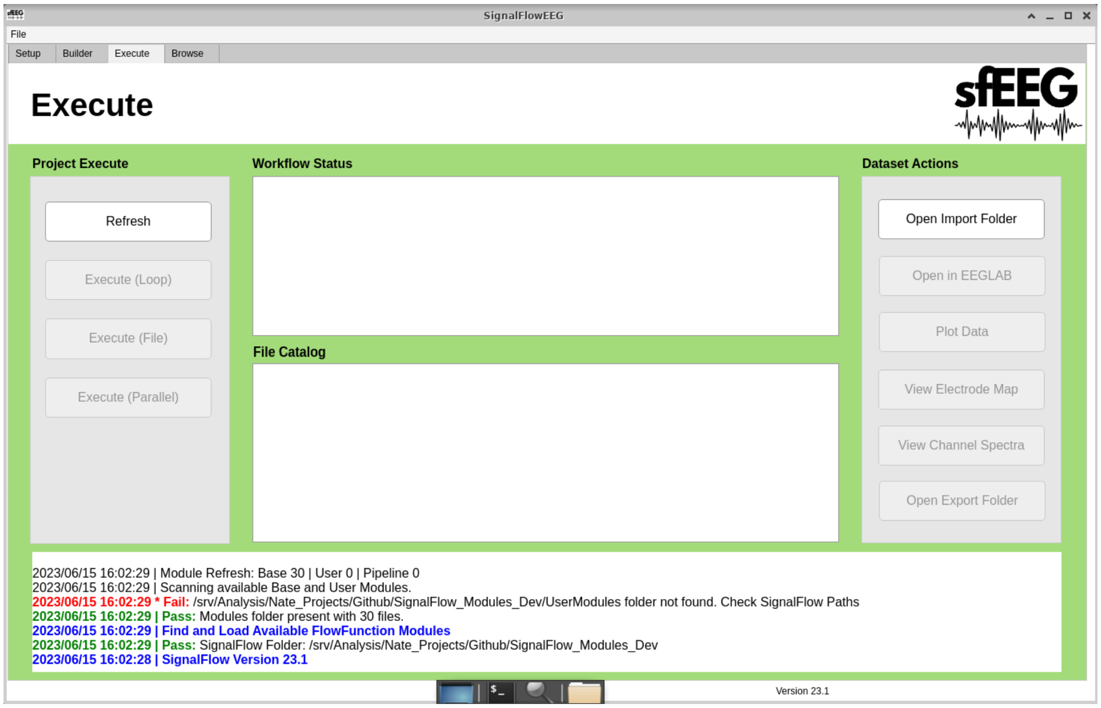
5.6 Reset Pipeline

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

5.7 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.

6 Execute Tab



```
2023/06/15 16:02:29 | Module Refresh: Base 30 | User 0 | Pipeline 0
2023/06/15 16:02:29 | Scanning available Base and User Modules.
2023/06/15 16:02:29 * Fail: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev/UserModules folder not found. Check SignalFlow Paths
2023/06/15 16:02:29 | Pass: Modules folder present with 30 files.
2023/06/15 16:02:29 | Find and Load Available FlowFunction Modules
2023/06/15 16:02:29 | Pass: SignalFlow Folder: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev
2023/06/15 16:02:28 | SignalFlow Version 23.1
```

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

6.1 Refresh

- This button is used to refresh the workflow status and file catalog when a change to the file sytem or pipeline has occured.
    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.

6.2 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.

6.3 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 follwoing 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 Parralel Processing Toolbox

6.4 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.

6.5 Open Import Folder

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

6.6 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.

6.7 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.

6.8 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.

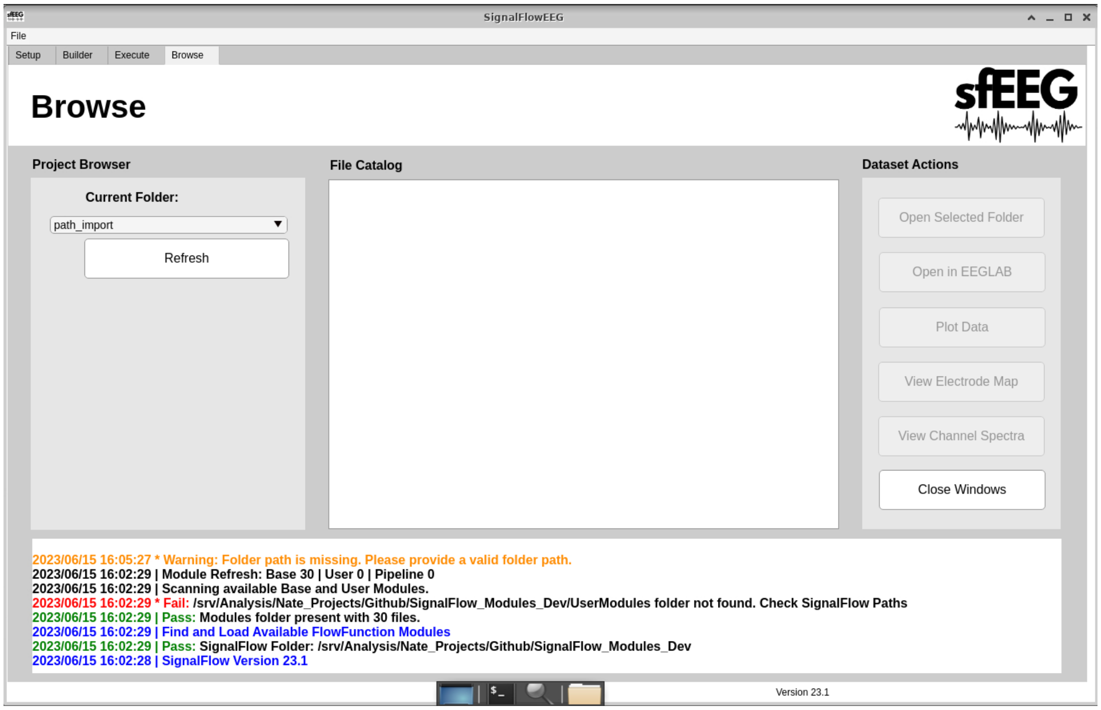
6.9 View Electrode Map

- 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.

6.10 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.

7 SignalFlow: Browse Tab



```
2023/06/15 16:05:27 * Warning: Folder path is missing. Please provide a valid folder path.
2023/06/15 16:02:29 | Module Refresh: Base 30 | User 0 | Pipeline 0
2023/06/15 16:02:29 | Scanning available Base and User Modules.
2023/06/15 16:02:29 * Fail: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev/UserModules folder not found. Check SignalFlow Paths
2023/06/15 16:02:29 | Pass: Modules folder present with 30 files.
2023/06/15 16:02:29 | Find and Load Available FlowFunction Modules
2023/06/15 16:02:29 | Pass: SignalFlow Folder: /srv/Analysis/Nate_Projects/Github/SignalFlow_Modules_Dev
2023/06/15 16:02:28 | SignalFlow Version 23.1
```

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

7.1 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.

7.2 Refresh

- This button is used to refresh the file catalog when a change to the file sytem or pipeline has occured.
    - 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.

7.3 Open Selected 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.

7.4 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.

7.5 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.

7.6 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.

7.7 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.

7.8 Close Windows

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

8 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 analzying 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.

8.1 Overview

This tutorial assumes that you have already installed **SignalFlowEEG** (Chapter 1) and are familiar with the SignalFlow interface (Chapter 4).

During this tutorial we will calculate spectral power and run a connectivity analysis.

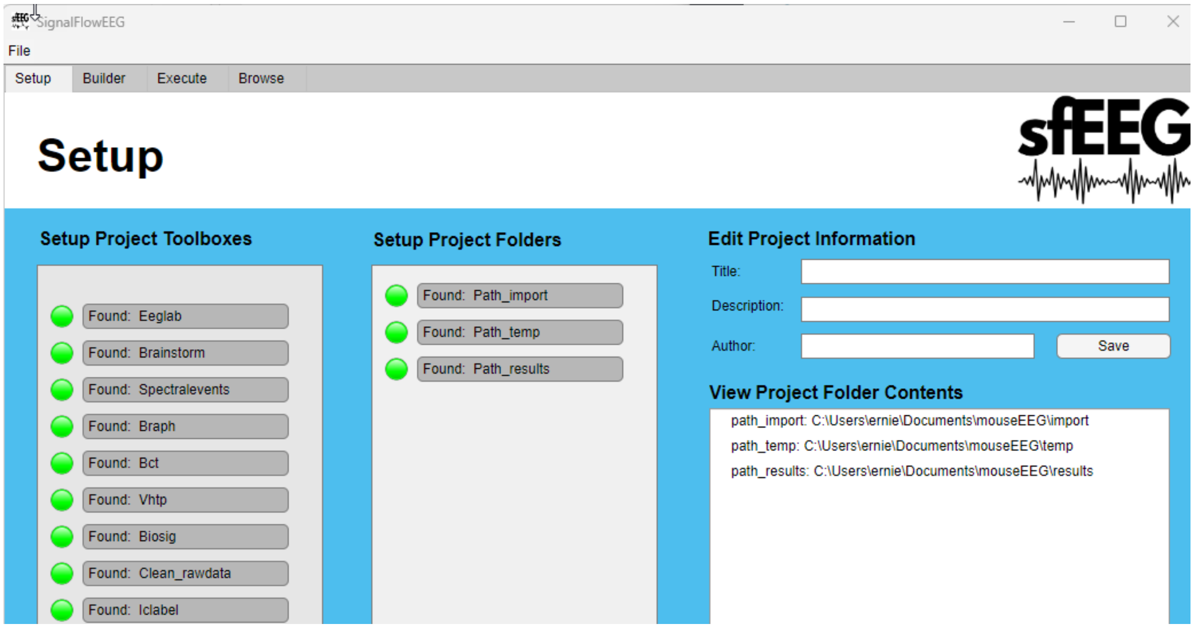
8.2 Loading Data

1. Create a new project folder in your filesystem to store the data and analysis results. For example, `~/Documents/mouseEEG`.
2. Create three subfolders in the project folder: 'import', 'temp', and 'results'.
3. On the SignalFlow GUI select the 'Setup' tab and setup your project folders by assigning them to each of the three folders you just created.

4. Let's import our mouse EEG files. In this example, we will be assuming you have already preprocessed the data into SET format. We will use the import tool to copy over the SET files to our project folder.

Select "Open the VHTP Import Tool" button underneath the Setup Project Folders pane:

5. The vHTP import tool is a multipurpose utility to import data from a variety of formats into EEGLAB SET format. It also can copy SET files directly. The Import tool has the ability to select SET files across subfolders and perform keyword filtering which has advantages compared to a manual file copy.



```
path_import: C:\Users\ernie\Documents\mouseEEG\import
path_temp: C:\Users\ernie\Documents\mouseEEG\temp
path_results: C:\Users\ernie\Documents\mouseEEG\results
```

Figure 8.1: Setup Folders

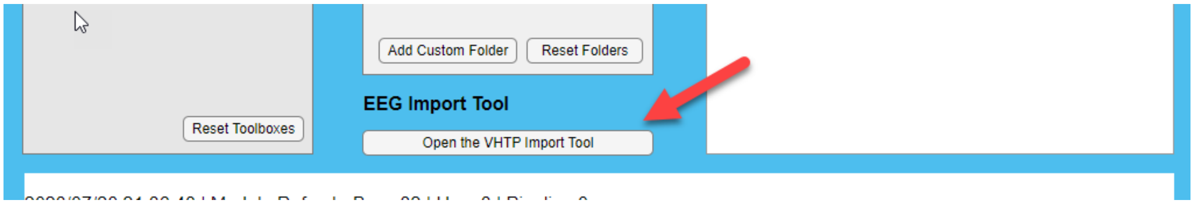
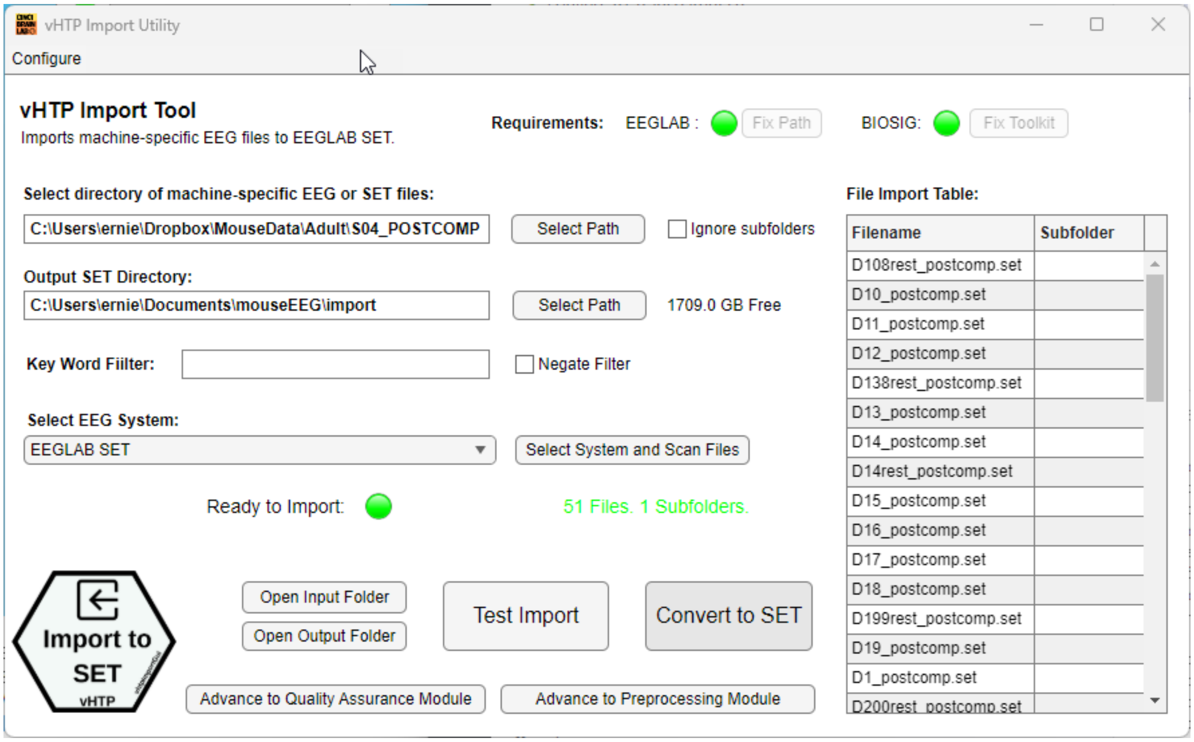


Figure 8.2: Setup Folders

Please refer to the vHTP import tool documentation and follow the instructions for copy SET files to your project folder (*Example Workflow 2*).

Link to vHTP import tool documentation

Here is example of the import tool with the SET files selected and the destination folder set to the 'import' folder in your project folder:



Select directory of machine-specific EEG or SET files: C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP

Output SET Directory: C:\Users\ernie\Documents\mouseEEG\import

1709.0 GB Free

Key Word Fiilter:

Select EEG System: EEGLAB SET

Ready to Import: ●

51 Files. 1 Subfolders.

File Import Table:

Filename	Subfolder
D108rest_postcomp.set	
D10_postcomp.set	
D11_postcomp.set	
D12_postcomp.set	
D138rest_postcomp.set	
D13_postcomp.set	
D14_postcomp.set	
D14rest_postcomp.set	
D15_postcomp.set	
D16_postcomp.set	
D17_postcomp.set	
D18_postcomp.set	
D199rest_postcomp.set	
D19_postcomp.set	
D1_postcomp.set	
D200rest_postcomp.set	

Figure 8.3: Ready to Import!

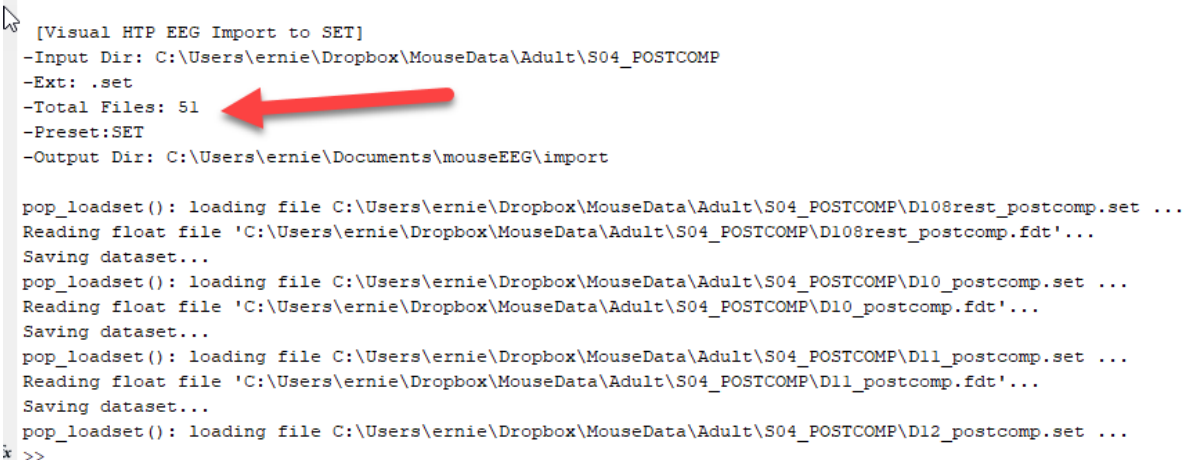
After you have selected the SET files and the destination folder, click the 'Import to SET' button to copy the files over. You will notice in the terminal window that each file will be loaded into EEGLAB and a channel montage graphic will be exported to confirm channel locations.

Details regarding the import will be in the command window:

Following the import, you should see the SET files in the 'import' folder:

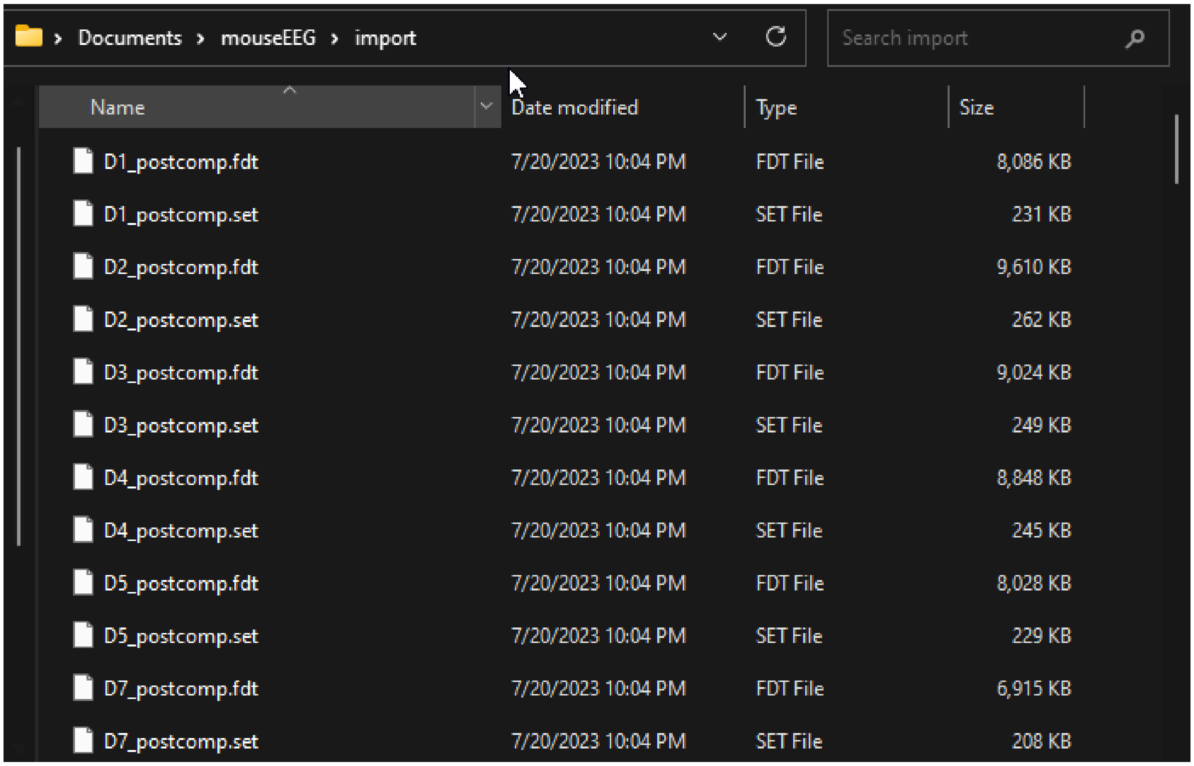
6. Return to the SignalFlow Application and fill in any Project Details you would like to include. These details will be included in the analysis report.

7. Next move to the Builder tab to start our analysis workflow.



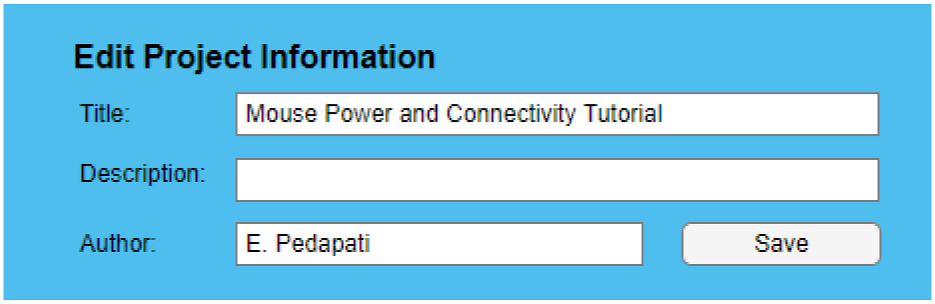
```
 [Visual HTP EEG Import to SET]
-Input Dir: C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP
-Ext: .set
-Total Files: 51
-Preset:SET
-Output Dir: C:\Users\ernie\Documents\mouseEEG\import

pop_loadset(): loading file C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP\D108rest_postcomp.set ...
Reading float file 'C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP\D108rest_postcomp.fdt'...
Saving dataset...
pop_loadset(): loading file C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP\D10_postcomp.set ...
Reading float file 'C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP\D10_postcomp.fdt'...
Saving dataset...
pop_loadset(): loading file C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP\D11_postcomp.set ...
Reading float file 'C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP\D11_postcomp.fdt'...
Saving dataset...
pop_loadset(): loading file C:\Users\ernie\Dropbox\MouseData\Adult\S04_POSTCOMP\D12_postcomp.set ...
>>
```

Figure 8.4: Import Details



Name	Date modified	Type	Size
D1_postcomp.fdt	7/20/2023 10:04 PM	FDT File	8,086 KB
D1_postcomp.set	7/20/2023 10:04 PM	SET File	231 KB
D2_postcomp.fdt	7/20/2023 10:04 PM	FDT File	9,610 KB
D2_postcomp.set	7/20/2023 10:04 PM	SET File	262 KB
D3_postcomp.fdt	7/20/2023 10:04 PM	FDT File	9,024 KB
D3_postcomp.set	7/20/2023 10:04 PM	SET File	249 KB
D4_postcomp.fdt	7/20/2023 10:04 PM	FDT File	8,848 KB
D4_postcomp.set	7/20/2023 10:04 PM	SET File	245 KB
D5_postcomp.fdt	7/20/2023 10:04 PM	FDT File	8,028 KB
D5_postcomp.set	7/20/2023 10:04 PM	SET File	229 KB
D7_postcomp.fdt	7/20/2023 10:04 PM	FDT File	6,915 KB
D7_postcomp.set	7/20/2023 10:04 PM	SET File	208 KB

Figure 8.5: Contents of Import Folder



Edit Project Information

Title: Mouse Power and Connectivity Tutorial

Description:

Author: E. Pedapati

Figure 8.6: Project Details

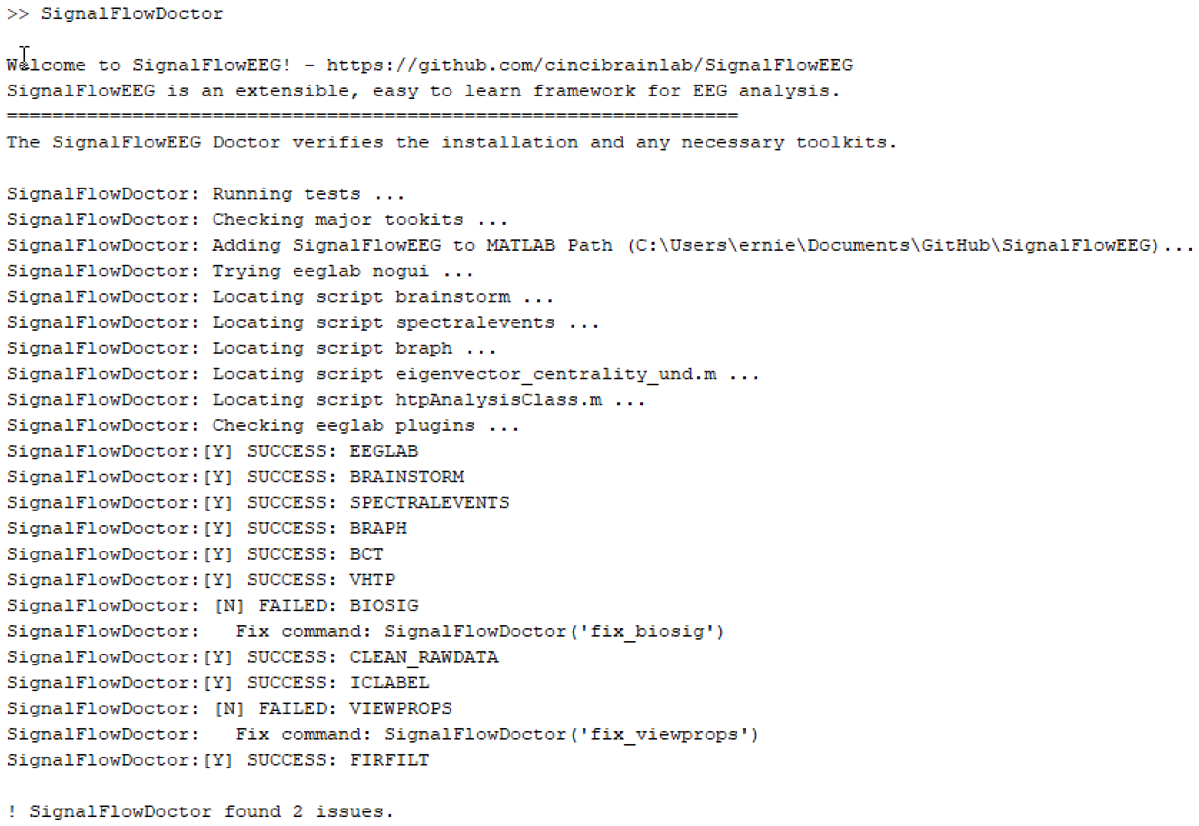
From the inflow modules select “Import SET” From the outflow modules select “Calculate Rest Power”

8. Confirm the module outflow path where the power results will be saved. In this example, we will save the results to the ‘results’ folder in our project folder.
9. Switch to the Execute Tab and click the “Execute (Loop)” Button to import each SET file sequentially and calculate the power. The results will be stored in table format in the ‘results’ folder.

8.3 Optional: Customize Power Bands

Published mouse EEG bands differ in exact frequencies from human EEG bands. You can customize the power bands by making a copy of the ‘Calculate Rest Power’ module and editing the ‘Power Bands’ parameter.

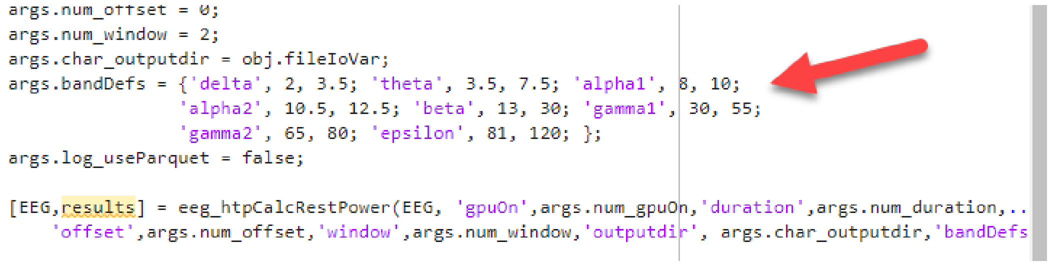
1. To start, go back to the setup tab. In your file system, create a new results folder called “results_custombands” to distinguish it from your other results folder.
2. Next, in the Setup tab select “Add Custom Folder” and select the “results_custombands” folder you just created. Add a easy to remember tag so you can identify it later.
3. Switch to the Builder tab. Click on “Calculate Rest Power” and press the “Delete Module” Button. This will remove the default power module.
4. Find “Calculate Rest Power” in the outflow modules and click on “Copy and Edit Module”. This will create a copy of the power module that you can edit.



```
>> SignalFlowDoctor

Welcome to SignalFlowEEG! - https://github.com/cincibrainlab/SignalFlowEEG
SignalFlowEEG is an extensible, easy to learn framework for EEG analysis.
================================================================
The SignalFlowEEG Doctor verifies the installation and any necessary toolkits.

SignalFlowDoctor: Running tests ...
SignalFlowDoctor: Checking major tookits ...
SignalFlowDoctor: Adding SignalFlowEEG to MATLAB Path (C:\Users\ernie\Documents\GitHub\SignalFlowEEG)...
SignalFlowDoctor: Trying eeglab nogui ...
SignalFlowDoctor: Locating script brainstorm ...
SignalFlowDoctor: Locating script spectralevents ...
SignalFlowDoctor: Locating script braph ...
SignalFlowDoctor: Locating script eigenvector_centrality_und.m ...
SignalFlowDoctor: Locating script htpAnalysisClass.m ...
SignalFlowDoctor: Checking eeglab plugins ...
SignalFlowDoctor:[Y] SUCCESS: EEGLAB
SignalFlowDoctor:[Y] SUCCESS: BRAINSTORM
SignalFlowDoctor:[Y] SUCCESS: SPECTRALEVENTS
SignalFlowDoctor:[Y] SUCCESS: BRAPH
SignalFlowDoctor:[Y] SUCCESS: BCT
SignalFlowDoctor:[Y] SUCCESS: VHTP
SignalFlowDoctor: [N] FAILED: BIOSIG
SignalFlowDoctor:   Fix command: SignalFlowDoctor('fix_biosig')
SignalFlowDoctor:[Y] SUCCESS: CLEAN_RAWDATA
SignalFlowDoctor:[Y] SUCCESS: ICLABEL
SignalFlowDoctor: [N] FAILED: VIEWPROPS
SignalFlowDoctor:   Fix command: SignalFlowDoctor('fix_viewprops')
SignalFlowDoctor:[Y] SUCCESS: FIRFILT

! SignalFlowDoctor found 2 issues.
```

Figure 8.7: Adding a custom folder

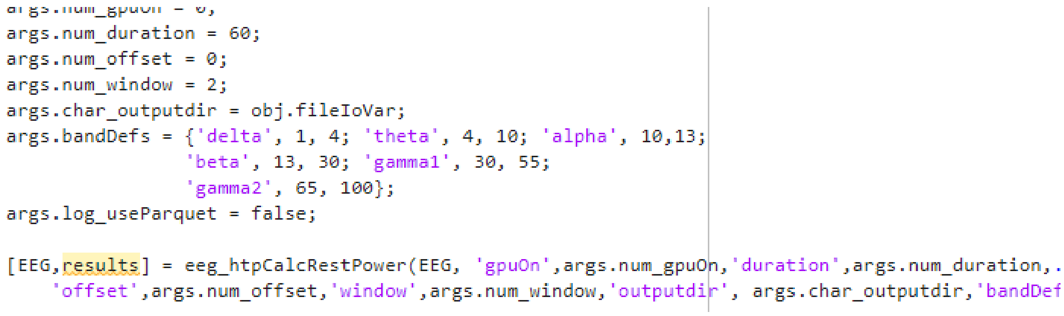
5. The MATLAB editor will open up with a copy of the power module. Here the parameters can be clearly seen, including the power band definitions.



```
args.num_offset = 0;
args.num_window = 2;
args.char_outputdir = obj.fileIoVar;
args.bandDefs = {'delta', 2, 3.5; 'theta', 3.5, 7.5; 'alpha1', 8, 10;
                 'alpha2', 10.5, 12.5; 'beta', 13, 30; 'gamma1', 30, 55;
                 'gamma2', 65, 80; 'epsilon', 81, 120; };
args.log_useParquet = false;

[EEG,results] = eeg_htpCalcRestPower(EEG, 'gpuOn',args.num_gpuOn,'duration',args.num_duration,..
    'offset',args.num_offset,'window',args.num_window,'outputdir', args.char_outputdir,'bandDefs
```

Figure 8.8: Power Band Definitions

6. Let's edit the power band definitions based on Jonak et al. 2020: "Power was then further binned into standard frequency bands: Delta (1–4 Hz), Theta (4–10 Hz), Alpha (10–13 Hz), Beta (13–30 Hz), and Gamma was divided into "Low Gamma" (30–55 Hz), and "High Gamma" (65–100 Hz)."

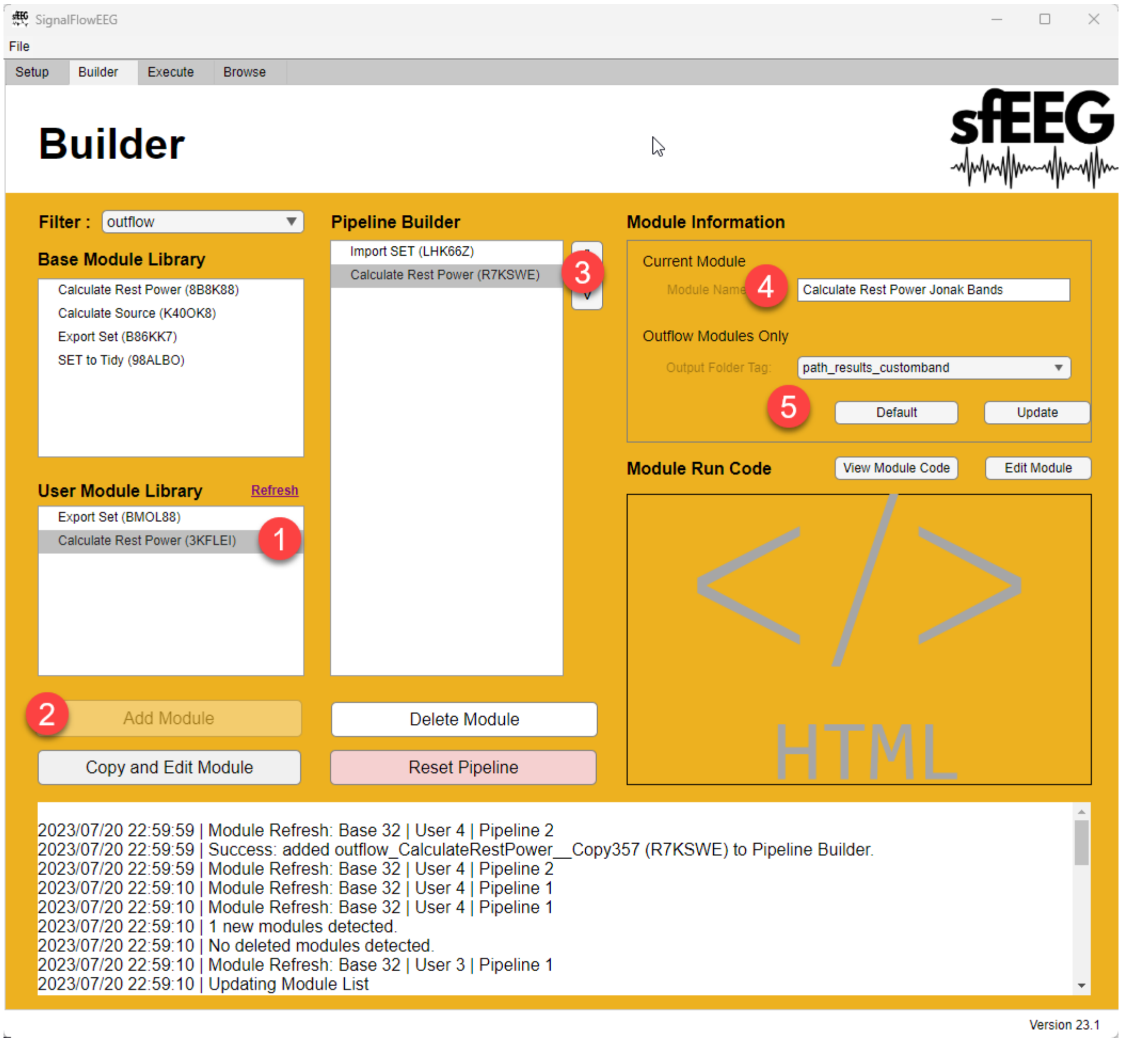


```
args.num_gpuOn = 0;
args.num_duration = 60;
args.num_offset = 0;
args.num_window = 2;
args.char_outputdir = obj.fileIoVar;
args.bandDefs = {'delta', 1, 4; 'theta', 4, 10; 'alpha', 10,13;
                 'beta', 13, 30; 'gamma1', 30, 55;
                 'gamma2', 65, 100};
args.log_useParquet = false;

[EEG,results] = eeg_htpCalcRestPower(EEG, 'gpuOn',args.num_gpuOn,'duration',args.num_duration,.
    'offset',args.num_offset,'window',args.num_window,'outputdir', args.char_outputdir,'bandDef
```

Figure 8.9: Modified bands based on Jonak et al.

Save and close the editor and return to the SignalFlow application.

7. Let's activate our new power module with custom bands. There are several steps to do



Filter : outflow

Module Name: Calculate Rest Power Jonak Bands

Output Folder Tag: path_results_customband

```
2023/07/20 22:59:59 | Module Refresh: Base 32 | User 4 | Pipeline 2
2023/07/20 22:59:59 | Success: added outflow_CalculateRestPower__Copy357 (R7KSWE) to Pipeline Builder.
2023/07/20 22:59:59 | Module Refresh: Base 32 | User 4 | Pipeline 2
2023/07/20 22:59:10 | Module Refresh: Base 32 | User 4 | Pipeline 1
2023/07/20 22:59:10 | Module Refresh: Base 32 | User 4 | Pipeline 1
2023/07/20 22:59:10 | 1 new modules detected.
2023/07/20 22:59:10 | No deleted modules detected.
2023/07/20 22:59:10 | Module Refresh: Base 32 | User 3 | Pipeline 1
2023/07/20 22:59:10 | Updating Module List
```

this.

Step 1. Click on `Refresh` to user modules. This will load the new custom power module.

Step 2. Next, select the new module and add it to the pipeline builder by clicking on `Add Module`.

Step 3. Select the new power module so we can add a custom name.

Step 4. Add a custom name to the module that is more descriptive.

Step 5. Select the new output folder for the power results and select "Update".

8. Now return to the builder tab and click on the "Execute (Loop)" button to run the analysis. The results will be stored in the 'results_custombands' folder.

8.4 Calculating Connectivity

The vHTP function eeg_htpGraphPhaseBcm (Brain connectivity matrix) is a comprehensive function that conducts phase-based connectivity analysis on an EEG dataset. It calculates several common phase-based connectivity measures like DWPLI, IPSC, etc between all pairs of EEG channels.

It takes the input EEG data, which is expected to be epoched into trials. It loops through all channel pairs, extracts the signal for those pairs, and calculates the connectivity measures over a range of frequencies.

The output is the connectivity matrices for each measure and frequency, as well as graph theory measures calculated on the connectivity matrices.

Let's run a custom version of this connectivity function to calculate only gamma connectivity and perform the operation on parallel processing to speed it up.

1. You can start right at the builder tab. Remove the power module but keep the import SET module.
2. Find the "Calculate Connectivity" module and click on "Copy and Edit Module". This will create a copy of the connectivity module that you can edit.
3. Adjust the bands to only include gamma bands. If you do not have a GPU, change the gpuOn parameter to 0. If you do have a GPU, you can leave it at 1.
4. This time we will also edit the fname to change the module name in list boxes. This is optional but can be helpful to distinguish between different connectivity modules.

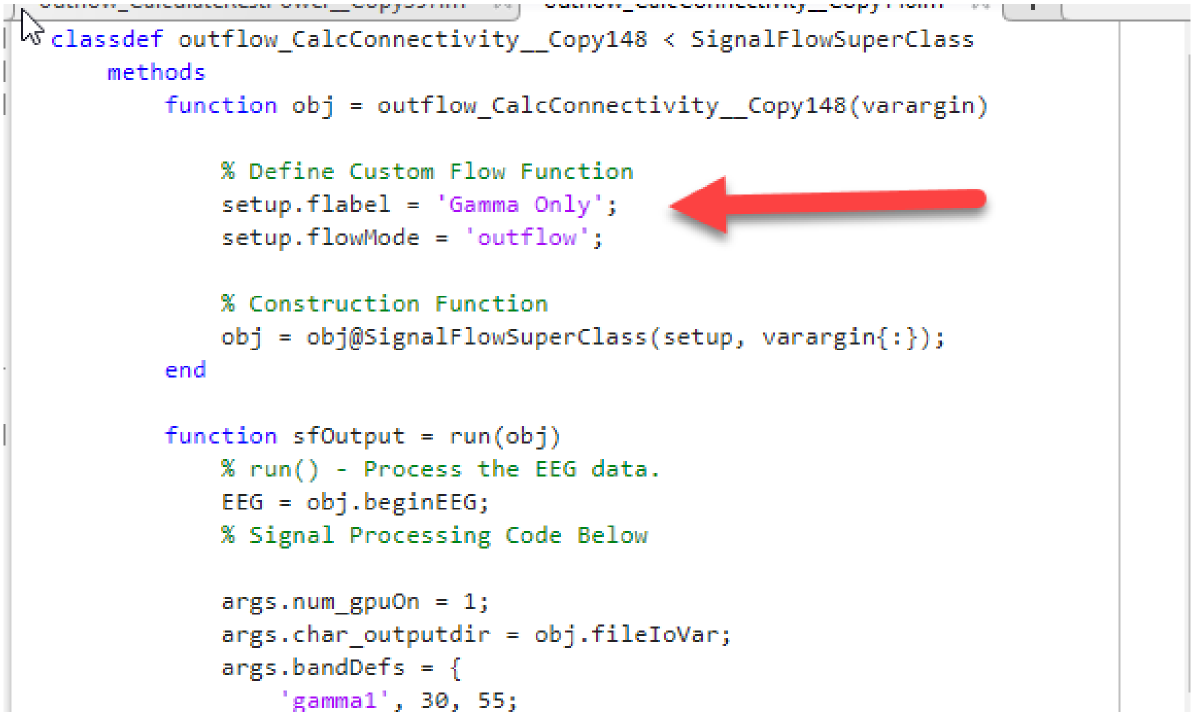
4. Save and close the editor and return to the SignalFlow application.
5. At this point, you could update the description and pick an alternative results folder. Make sure to press "Update" to save your changes.
6. Now return to the execute tab and click on the "Execute (Parallel)" button to run the analysis. The results will be stored in the 'results' folder.

8.5 Summary:

In this tutorial, we have shown how to use **SignalFlowEEG** to analyze mouse EEG data. We have shown how to import SET files, calculate spectral power, and calculate connectivity.

The raw data files containing the values of the results can be opened across most statistical software including MATLAB.

We will next work on loading and visualizing this data in R.



```
classdef outflow_CalcConnectivity__Copy148 < SignalFlowSuperClass
    methods
        function obj = outflow_CalcConnectivity__Copy148(varargin)

            % Define Custom Flow Function
            setup.flabel = 'Gamma Only';
            setup.flowMode = 'outflow';

            % Construction Function
            obj = obj@SignalFlowSuperClass(setup, varargin{:});
        end

        function sfOutput = run(obj)
            % run() - Process the EEG data.
            EEG = obj.beginEEG;
            % Signal Processing Code Below

            args.num_gpuOn = 1;
            args.char_outputdir = obj.fileIoVar;
            args.bandDefs = {
                'gamma1', 30, 55;
```

Figure 8.10: Custom Module Name