

**Instruction document (User guide)**

Ver 1.0

# NeuroNetFlow

**Update date:** 2026.01.18

**Authors:** Seokbeen Lim, Marin Nycklemoe, Paul H. Min

**Institution:** Mayo Clinic

# Table of Contents

<b>1. License &amp; Citation .....</b>	1
○ 1.1. MIT License.....	1
○ 1.2. How to Cite .....	1
<b>2. Introduction.....</b>	2
<b>3. System requirements &amp; installation .....</b>	3
○ 3.1. System requirements .....	3
○ 3.2. Installation guide.....	6
○ 3.3. Verified test environments.....	7
<b>4. Software architecture &amp; description of modules .....</b>	8
○ 4.1. Software architecture .....	8
○ 4.2. Functional description.....	9
- 4.2.1. Main control interface .....	9
- 4.2.2. Module 1: ImageProcess .....	9
- 4.2.3. Module 2: AnalysisData .....	13
4.2.3.1. Respiratory signal processing .....	13
4.2.3.2. CSF dynamics analysis .....	14
4.2.3.3. Cardiac signal processing .....	16
4.2.3.4. Integrated visualization.....	16
- 4.2.4. Output and feature export.....	17
○ 4.3. Exported features for downstream statistical analysis .....	17
<b>5. Step-by-Step Workflow .....</b>	26
○ 5.1. Check the directory .....	26
○ 5.2. Run for the main control window .....	26
○ 5.3. Load image dataset .....	27
○ 5.4. Conduct segmenting ROI.....	28
○ 5.5. Check the signal, conduct sudo mask correction, and calibrate the signal ..	30
○ 5.6. Load the physiology signal and save the dataset for analyzing it.....	37
○ 5.7. Load preprocessed data for analyzing CSF signal with aligned physiological data.....	39
<b>6. Additional function .....</b>	47
○ 6.1. Pre-defined mask .....	47
<b>7. Software metadata .....</b>	49
<b>8. Reference .....</b>	50

## **1. License & Citation**

### **1.1. MIT License**

Copyright (c) 2026 [Seokbeen Lim / Mayo Clinic]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### **1.2. How to Cite**

Please cite this software as follows:

APA style:

Lim, S., Nycklemoe, M. E., Jacobson, D. N., Kim, D. C., Rossman, P. J., Karki, P., Lowe, V. J., Cogswell, P. M., & Min, P. H. (2026). NeuroNetFlow: A MATLAB-based software toolbox for analyzing flow dynamics based on phase-contrast MRI. *SoftwareX*, [Volume], [Article Number]. [https://doi.org/\[DOI\\_Number\]](https://doi.org/[DOI_Number])

IEEE style:

S. Lim et al., "NeuroNetFlow: A MATLAB-based software toolbox for analyzing flow dynamics based on phase-contrast MRI," *SoftwareX*, vol. [Volume], Art. no. [Article Number], Jan. 2026. doi: [DOI\_Number].

JAMDA (AMA) style:

Lim S, Nycklemoe ME, Jacobson DN, et al. NeuroNetFlow: a MATLAB-based software toolbox for analyzing flow dynamics based on phase-contrast MRI. *SoftwareX*. 2026;[Volume]:[Article Number]. doi:[DOI Number]

## 2. Introduction

### ***NeuroNetFlow: A MATLAB-based software toolbox for analyzing flow dynamics based on phase-contrast MRI***

Cerebrospinal fluid (CSF) constitutes an essential element of the central nervous system's physiological framework<sup>1</sup>. Beyond serving as a mechanical buffer and providing buoyancy, CSF is involved in sustaining chemical homeostasis and purging metabolic waste from the brain<sup>1-4</sup>. Growing evidence suggests a correlation between CSF circulation and neurological decline, such as the glymphatic system, which has prompted ongoing research to define circulation mechanisms and refine the in vivo measurement of CSF dynamics<sup>2, 4</sup>. Historically, CSF flow was understood as being primarily driven by cardiac-related pulsatility, where the interplay of arterial pressure and intracranial compliance results in rhythmic oscillations<sup>5, 6</sup>. Yet, recent studies increasingly emphasize that respiration is a key driver of CSF circulation via venous return<sup>7-12</sup>. This is especially pertinent in humans, as breathing varies individually and can be intentionally altered, providing a possible non-invasive method to impact CSF flow<sup>6, 10, 13</sup>.

Advances in high-resolution velocity-encoding MRI now enable the direct quantification of both cardiac and respiratory influences on CSF flow, offering a precise means to investigate respiratory modulation<sup>6-8, 10, 11, 13</sup>. This approach allows for the detailed assessment of CSF velocity waveforms within key structures such as the foramen magnum, cerebral aqueduct, and lateral ventricles<sup>6</sup>. Through ROI-based analysis and temporal integration, these velocity measurements are transformed into meaningful physiological indices, including flow rate (ml/s) and volume displacement (ml)<sup>6, 14, 15</sup>. However, the integration of PC-MRI data with physiological signals like respiration is still hampered by the scarcity of suitable open-source toolboxes. Moreover, while various CSF segmentation algorithms exist<sup>7, 13-15</sup>, the lack of an accessible GUI environment often requires researchers to engage in time-consuming manual coding or masking. The *NeuroNetFlow* toolbox was developed to overcome these challenges, offering an open-source GUI that substantially reduces the time and effort required for comprehensive CSF dynamics analysis.

*NeuroNetFlow* isolates voxels associated with cardiac cycles and masks them using a previously established algorithm, operating on the premise that CSF flow incorporates cardiac oscillation components<sup>14, 15</sup>. This framework allows users to intuitively validate CSF ROI segmentation through a semi-automatic interactive mask editing feature, which facilitates the precise addition or removal of voxels corresponding to actual CSF signals within a GUI environment. Furthermore, the toolbox includes calibration and alignment functions that synchronize CSF data with external physiological signals, such as PPG and respiration. Building on empirical evidence that CSF dynamics also contain respiratory oscillation components, *NeuroNetFlow* provides an additional dedicated function to extract and mask these specific signals, enabling a comprehensive analysis of cardio-respiratory-related CSF movement<sup>6</sup>. Through a predefined analysis pipeline, users can evaluate preprocessed CSF signals alongside

aligned PPG and respiratory data. Utilizing this software, we quantified CSF characteristics—specifically CSF mean speed, CSF displacement, and CSF net flow—in participants with and without long-term breathing training<sup>6</sup>. Our previous findings indicate that sustained respiratory training modulates breathing patterns, resulting in enhanced daily CSF dynamics and suggesting a potential for non-invasive respiratory control<sup>6</sup>. *NeuroNetFlow* thus offers an intuitive, quantitative approach for PC-MRI researchers, with the potential to evolve into a generalized analysis pipeline through research community-driven algorithmic improvements in the future.

### **3. System requirements & installation**

#### **3.1. System requirements**

Operating system: macOS (Apple Silicon, Sequoia 15.7.2 or later), Windows 10/11

Software Environment: MATLAB R2023a or later.

Hardware requirements:

- Memory (RAM): 8 GB minimum (16 GB or higher recommended).
- Processor (CPU): Intel Core i5 or later. Apple M2/M3 series.
- Storage: At least xxx of free space for software installation; additional space required for PC MRI data.

Display & GUI requirements:

- Screen Resolution: 1920 x 1080 (Full HD) or higher recommended for optimal GUI layout precise ROI segmentation.
- Laptop Compatibility: The software is fully compatible with laptop environments (e.g., 13-inch to 16-inch models). The GUI is designed to be responsive; however, users on smaller screens should ensure that MATLAB's display scaling is set appropriately to view all interface elements.
- Pointing Device: A standard mouse or trackpad is required for interactive coordinate selection.

Data compatibility:

- Imaging data: NIfTI (.nii) format (Magnitude and Phase .nii file) from Philips scanner.
- Physiological Data: Compatible with .log file from Philips scanner containing time-stamped respiratory belt, ppg, and system marker signals.

#### **Software Components and Required Files**

To ensure full functionality of the **NeuroNetFlow toolbox**, the following files must be present in the MATLAB path. These dependencies were identified using the `matlab.codetools.requiredFilesAndProducts` utility to ensure all necessary components for the App Designer-based architecture are included.

- Main Control Interface
  - File Name: Breathing\_PCMRI\_GUI\_Ver1.mlapp
  - Role: This is the central control interface of the software. It adopts a multi-window architecture to manage the entire workflow, including:
    - Data loading and parameter configuration.

- Execution of downstream analysis modules for PC-MRI-based breathing-related CSF dynamics.
- Management and invocation of specialized sub-GUI windows.
- Associated Sub-GUI Windows

The main GUI dynamically calls the following specialized sub-windows depending on the user's analysis tasks:

<b>File Name</b>	<b>Description &amp; Functionality</b>
Analysis_further_window.mlapp	Advanced parameter tuning and extended post-processing analyses.
Analysis_signals_window.mlapp	Time- and frequency-domain analysis of physiological signals (respiration, CSF flow, and velocity).
CSF_onecycle_Window.mlapp	Analysis of CSF dynamics focused on a single respiratory cycle.
Calibration_window.mlapp	Static tissue-based velocity calibration to correct phase errors.
Calibration_window_overall.mlapp	Global calibration inspection and validation across the full ROI or volume.
MagnitudeImgPlot.mlapp	Visualization of magnitude images and detailed ROI inspection.
Open_image_slide.mlapp	Interactive multi-slice image browsing and visualization tool.

- External Functions and Resources

In addition to the GUI files, the following resources are essential for data parsing and interface styling:

- ReadPhilipsScanPhysLog.m: A critical utility function designed to parse Philips MRI physiology log files (SCANPHYSLOG) and extract high-resolution respiratory signals.
- Logo\_for\_GUI\_window.png: A graphical asset used for the software's branding and interface identity.

- Summary Statement

All listed files have been validated through the MATLAB Dependency Analyzer. This ensures that the complete set of functions, GUI layouts, and external resources required to execute the analysis workflow are self-contained within the repository.

### 3.2 Installation guide

#### Step 1: Download the Source Code

You can obtain the software by either cloning the repository using Git or downloading it as a ZIP file.

- **Option A:** Using Git (Recommended for updates) Open your terminal or command prompt and run:

Bash

```
git clone https://github.com/SeokbeenLim/NeuroNetFlow.git
```

- **Option B:** Manual Download

1. Visit the repository: <https://github.com/SeokbeenLim/NeuroNetFlow.git>
2. Click the green "**Code**" button and select "**Download ZIP**".
3. Extract the contents to your preferred working directory.

#### Step 2: Set Up MATLAB Path

To ensure that the main GUI can access all associated sub-windows and utility functions, you must add the project folders to the MATLAB search path.

1. Launch MATLAB.
2. In the Command Window, navigate to the folder where you extracted the files.
3. Run the following commands to include all subdirectories:

MATLAB

```
% Add the NeuroNetFlow folder and its subfolders to the path  
addpath(genpath(pwd));  
savepath;
```

*(Alternatively, you can right-click the root folder in the MATLAB Current Folder browser and select **Selected Folders and Subfolders** under the "Add to Path" menu.)*

#### Step 3: Launch the Application

Once the path is set, you can start the software by running the main App Designer file.

1. Locate `Breathing_PCMRI_GUI_Ver1.mlapp` in the file browser.
2. Double-click the file or type the following in the Command Window:

MATLAB

```
Breathing_PCMRI_GUI_Ver1
```

3. The main control interface will appear, and you can begin your analysis.

### 3.3. Verified test environments

The software has been developed and extensively tested across multiple hardware platforms and operating systems to ensure stability, performance, and reproducibility. All core functionalities, including GUI-based image processing, signal synchronization, and feature extraction modules, were validated under the following environments.

## Software

- **MATLAB:** R2023b or later
  - Verified with built-in App Designer components and signal/image processing toolboxes.
  - No third-party MATLAB toolboxes beyond standard distributions are required.

## Hardware and Operating Systems

Platform	Operating System	Processor	Memory	Verification Status
Apple MacBook / Mac Studio	macOS	Apple Silicon (M-series cores)	64 GB / 128 GB or higher	Fully tested
Windows Laptop	Windows 11	13th Gen Intel® Core™ i5-1350P (1.90 GHz)	32 GB RAM (31.6 GB usable)	Fully tested

## Notes

- The software was primarily developed and optimized on **Apple Silicon-based macOS systems**, demonstrating stable performance for large PC-MRI datasets and high-resolution physiological signals.
- Cross-platform compatibility was confirmed on **Windows 11**, with identical computational outputs and GUI behavior.
- Performance may scale with available memory, particularly for batch processing of multi-slice or multi-subject datasets.

## 4. Software architecture & description of modules

### 4.1. Software architecture

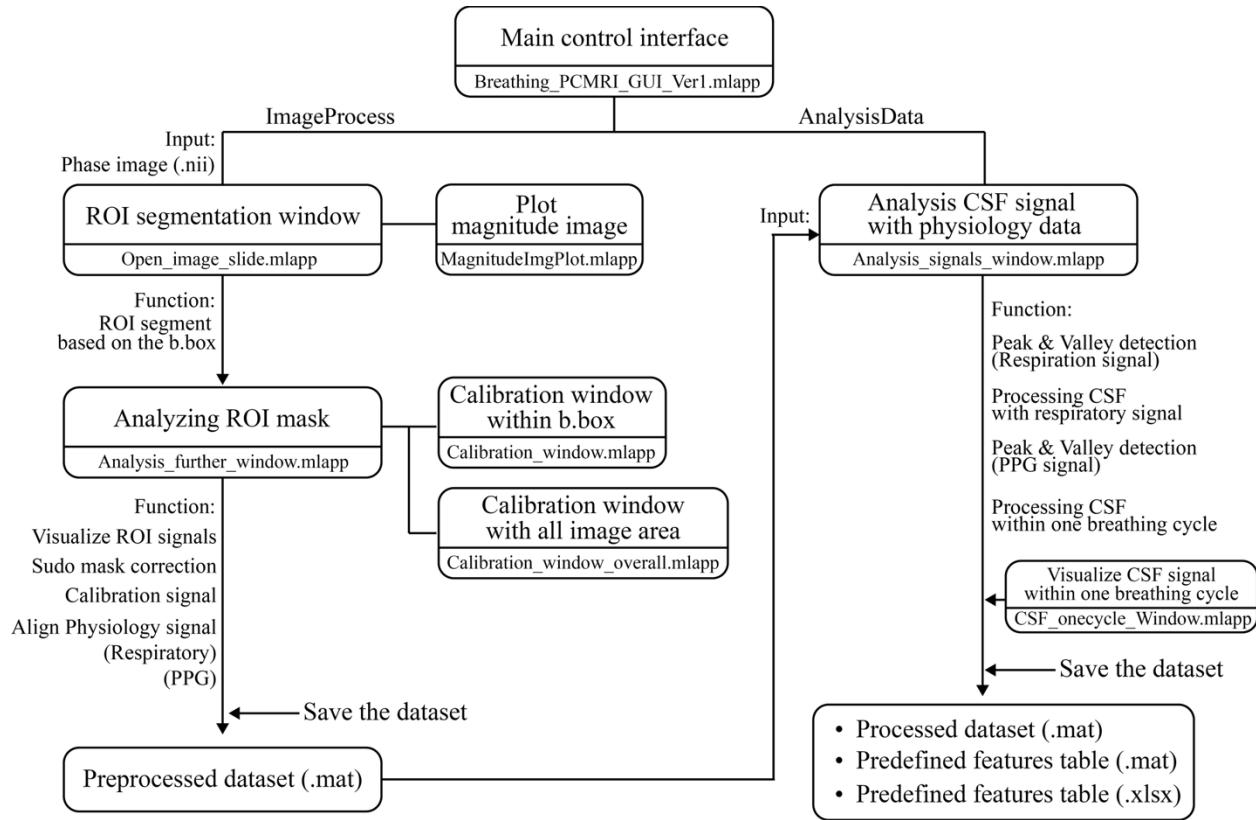


Figure 1. Software architecture.

The *NeuroNetFlow* toolbox is a MATLAB-based software framework designed for end-to-end analysis of cerebrospinal fluid (CSF) flow dynamics using phase-contrast MRI (PC-MRI) in conjunction with physiological signals.

The software architecture is organized into three major components: (1) the Main Control Interface, (2) the Image Processing module, and (3) the Signal Analysis module (Figure 1).

The Main Control Interface serves as the central orchestration layer of the toolbox. It enables users to sequentially execute image processing and signal analysis pipelines through a unified graphical user interface (GUI). During execution, the interface provides real-time status updates and detailed log messages, allowing users to monitor processing steps, parameter selections, and data flow across modules.

The Image Processing module focuses on PC-MRI image handling and ROI generation, while the Signal Analysis module operates on preprocessed temporal signals to extract quantitative CSF and physiological features. The modular design allows each component to function independently while maintaining seamless data exchange through standardized intermediate

outputs. This architecture supports reproducibility, extensibility, and interactive user control throughout the analysis workflow.

## 4.2. Functional description

This section describes the two core functional pillars of the *NeuroNetFlow* toolbox: the Image Processing pipeline and the Signal Analysis pipeline, both controlled through the Main Control Interface.

### 4.2.1. Main control interface

The Main Control Interface provides a unified environment for managing data loading, parameter configuration, module execution, and result visualization. Users can initiate image processing and signal analysis workflows independently or sequentially.

A dedicated logging panel displays execution messages, warnings, and processing summaries in real time, enabling transparent tracking of the analysis pipeline and facilitating debugging and reproducibility.

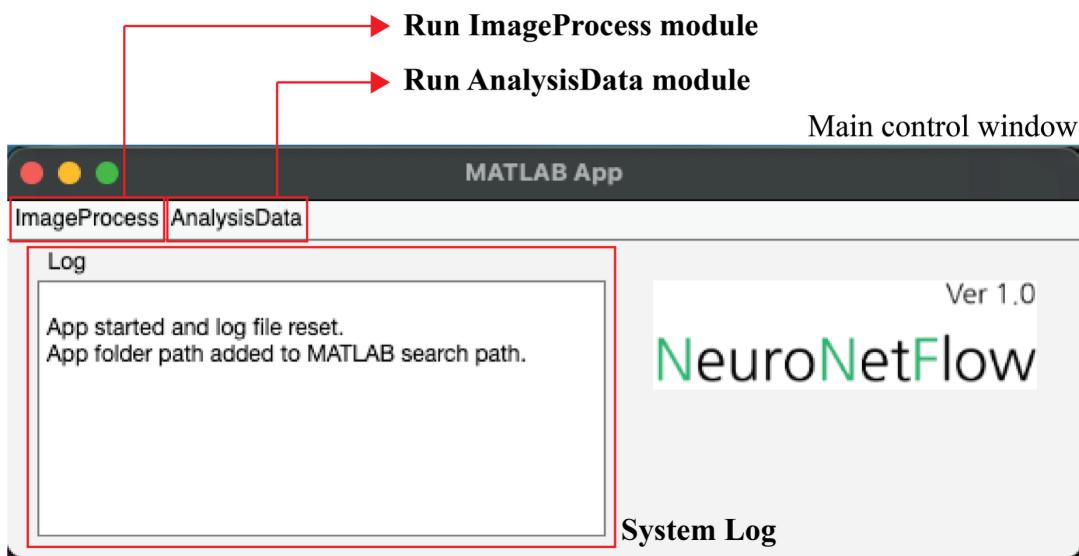


Figure 2. Functional description: Main control window

### 4.2.2. Module 1: ImageProcess

The ImageProcess module is responsible for PC-MRI image preprocessing and CSF region definition. It accepts PC-MRI phase images as input and applies automated CSF segmentation algorithms to identify candidate CSF regions.

Following automated segmentation, the detected CSF regions are converted into binary masks. The toolbox provides interactive editing tools that allow users to manually refine these masks. During this process, users can visualize both voxel-wise temporal signals and ROI-averaged

signals, enabling informed decisions when adding or removing voxels from the CSF mask through semi-manual editing.

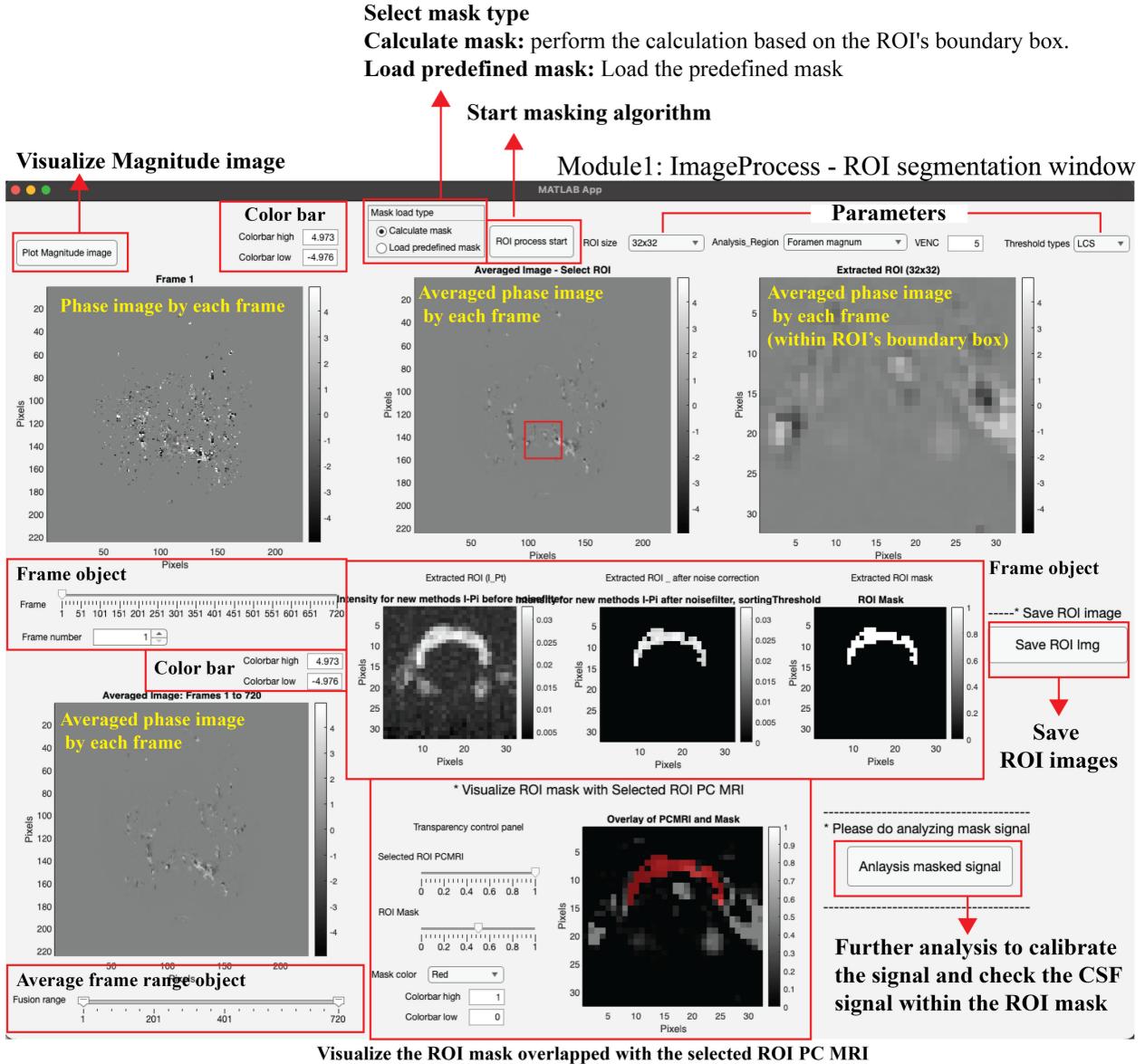


Figure 3. Functional description: Module1-ImageProcess, ROI segmentation window

To correct background phase offsets, the toolbox includes a calibration function that allows users to manually define stationary tissue regions presumed to contain B0 inhomogeneity effects. These tissue regions are used to estimate and remove background phase contributions from the entire PC-MRI dataset, improving the accuracy of CSF velocity estimation.

In addition, the ImageProcess module supports loading external physiological recordings (e.g., respiratory belt signals and PPG signals) and performs temporal alignment between PC-MRI data and physiological signals. The aligned and calibrated outputs are stored as preprocessed

datasets that serve as standardized inputs for subsequent signal analysis.

#### 4.2.3. Module 2: AnalysisData

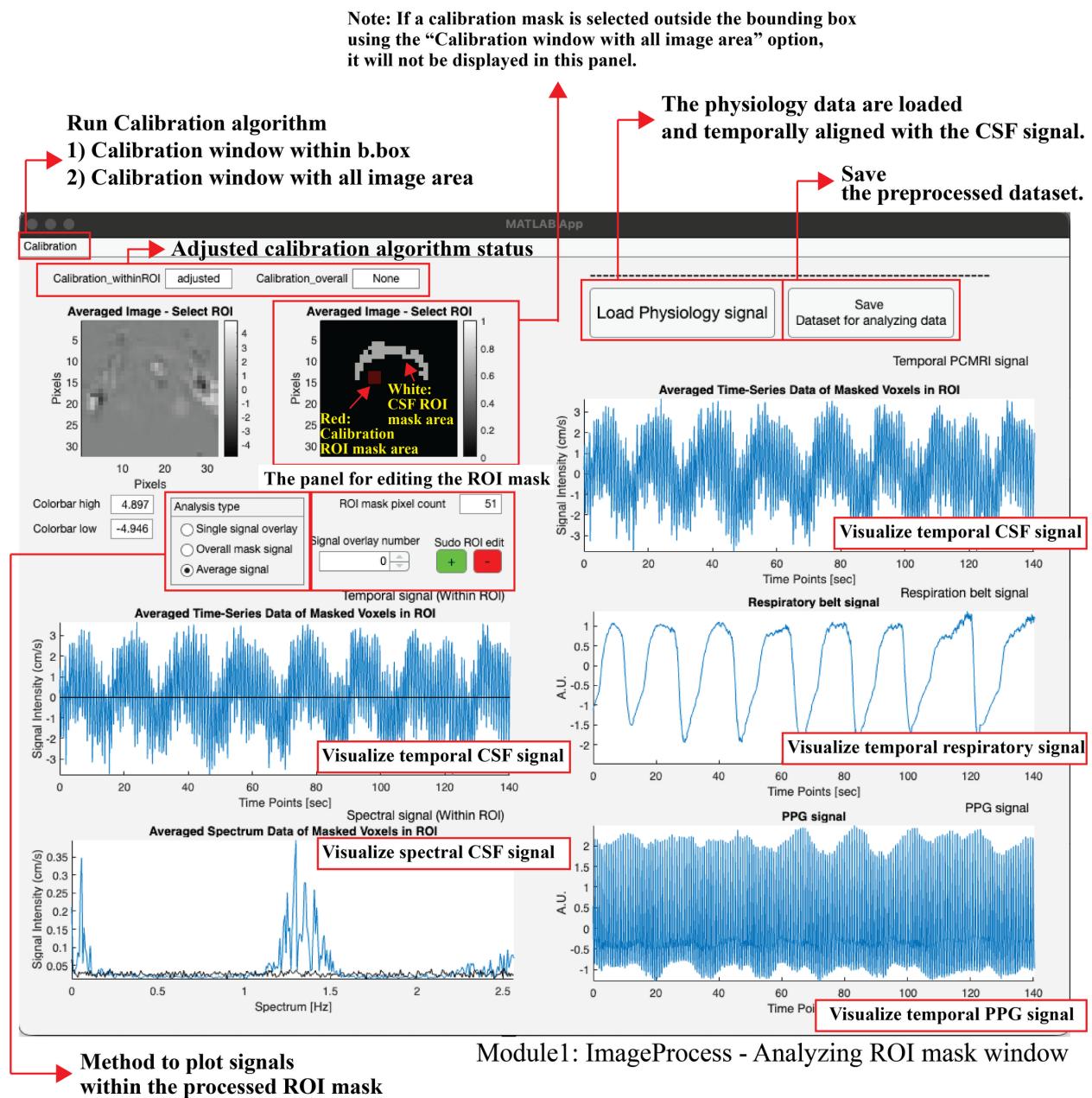
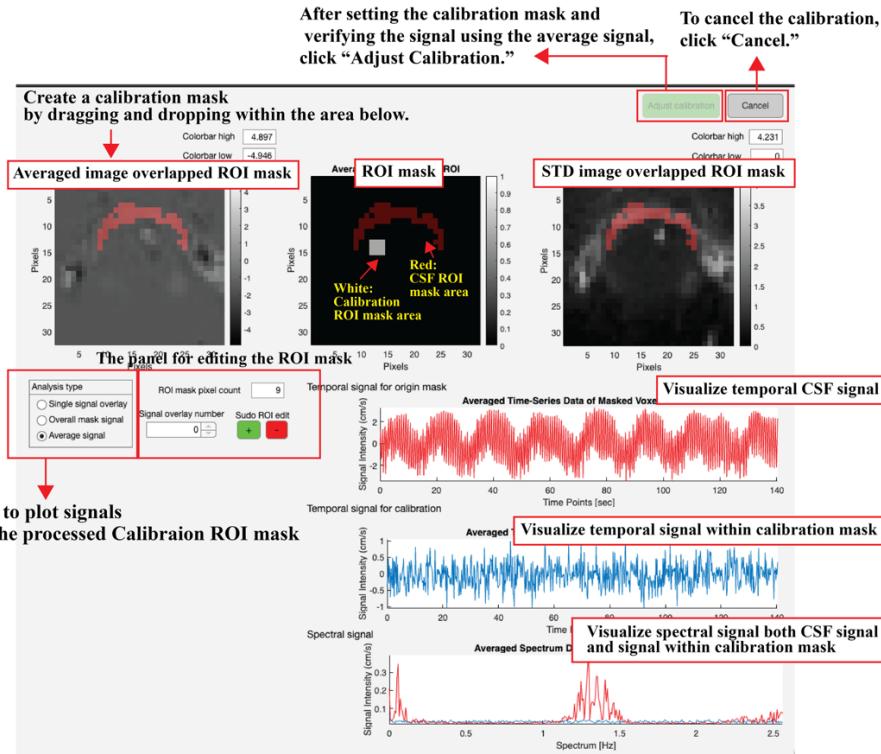
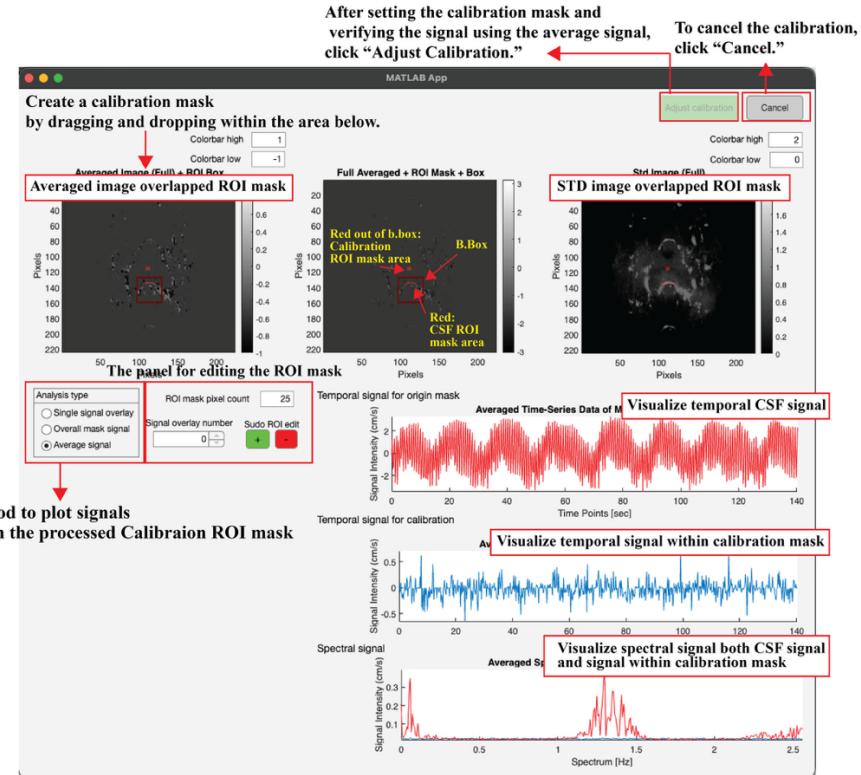


Figure 4. Functional description: Module1-ImageProcess, Analyzing ROI mask window



Module1: ImageProcess - Calibration window within b.box

Figure 5. Functional description: Module1-ImageProcess, Calibration window within b.box



Module1: ImageProcess - Calibration window with all image area

Figure 6. Functional description: Module1-ImageProcess, Calibration window with all image area

Module1: ImageProcess - Plot magnitude image window

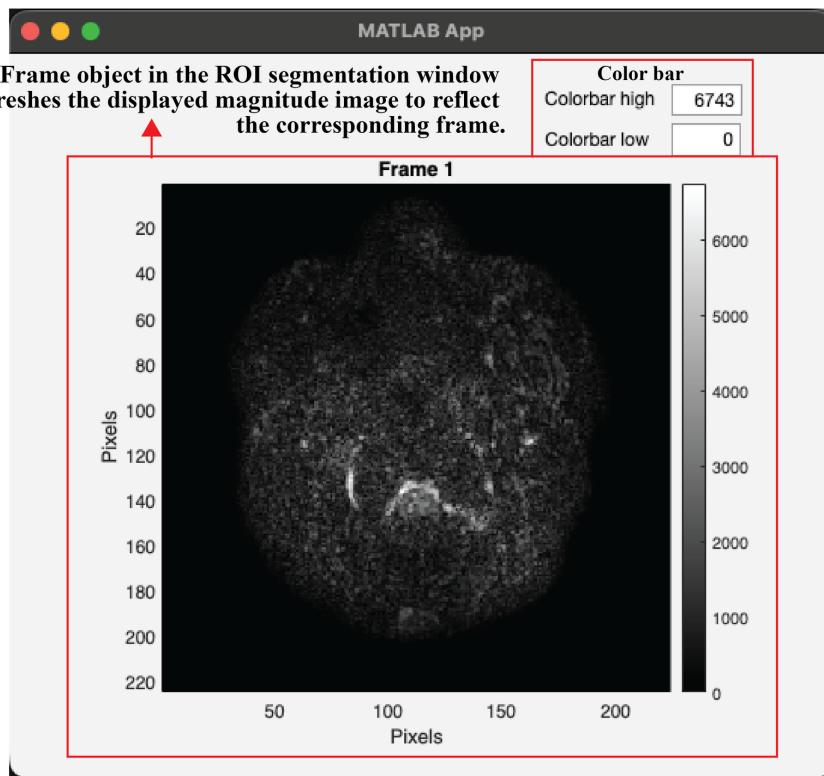


Figure 7. Functional description: Module1-ImageProcess, Plot magnitude image window

#### 4.2.3. Module 2: AnalysisData

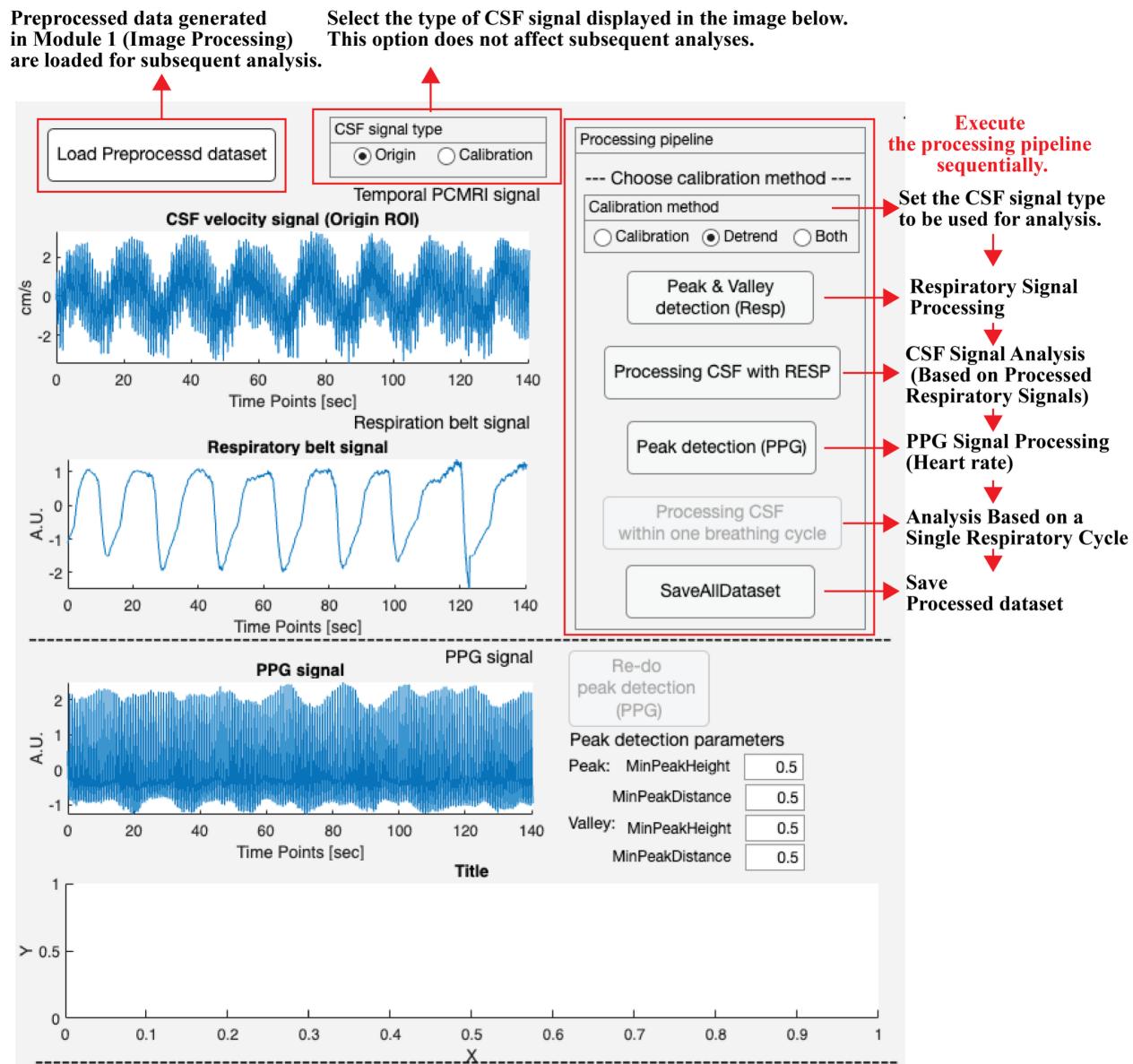
The AnalysisData module performs systematic signal processing and feature extraction using the preprocessed outputs generated by the ImageProcess module.

##### 4.2.3.1. Respiratory signal processing

Respiratory peak and valley detection is implemented using a two-stage procedure.

In the first stage, the toolbox assists users in identifying an optimal frequency band for filtering the raw respiratory signal. Because raw respiratory recordings often exhibit irregular patterns and noise, direct peak detection may be unreliable. Users can iteratively adjust band-pass filter parameters to obtain a cleaned respiratory waveform that preserves respiratory trends while suppressing noise.

Once a satisfactory filtered signal is obtained, the second stage performs peak and valley detection on the refined respiratory signal. Detected peaks and valleys are used as temporal landmarks for downstream CSF analysis. Detection parameters can be iteratively adjusted, and if automated detection remains suboptimal due to irregular breathing patterns, the toolbox provides manual editing functionality. Users can add, remove, or modify detected peaks and valleys interactively using mouse-based controls.



Module2: AnalysisData - Load data and processing pipeline part

Figure 8. Functional description: Module2-AnalysData, Load data and processing pipeline part

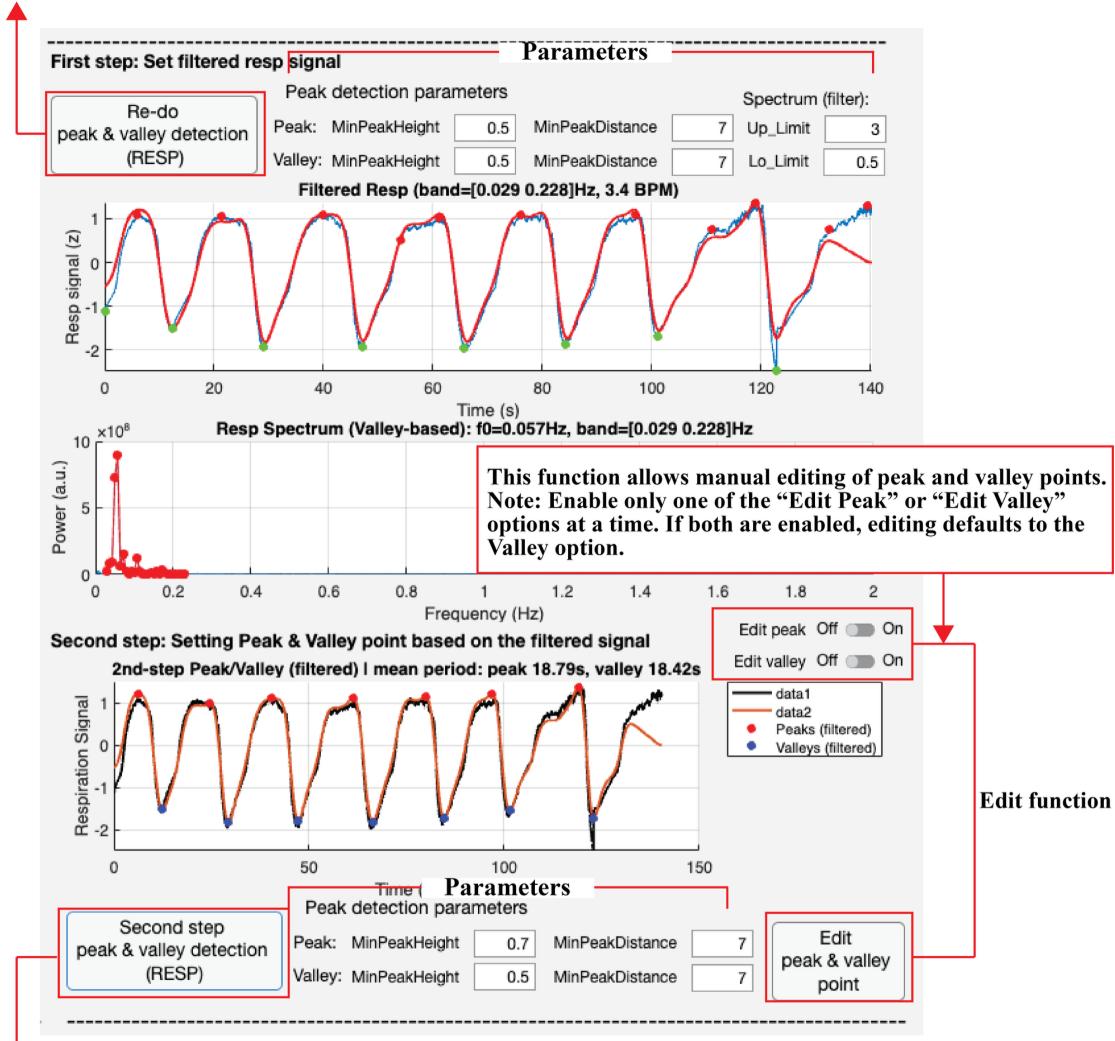
#### 4.2.3.2. CSF dynamics analysis

Using the finalized respiratory landmarks, the toolbox converts CSF velocity signals derived from PC-MRI into flow rate (ml/s) and computes volume displacement (ml) through temporal integration.

CSF dynamics are further analyzed by separating inhalation and exhalation phases, allowing computation of phase-specific integrals and summary statistics to characterize respiration-dependent CSF flow behavior.

Spectral analysis is performed on CSF velocity signals to quantify respiratory and cardiac contributions. The toolbox extracts the area under the curve (AUC) for predefined respiratory and cardiac frequency bands, along with additional spectral descriptors such as power variability and half-power bandwidth.

**Peak and valley detection is re-performed using the updated parameters.**



Module2: AnalysisData - Respiratory signal processing part

- 1) Perform the second step.
- 2) Peak and valley detection is re-performed using the updated parameters.

The respiratory signal processing pipeline consists of two sequential steps.

In the first step, a filtered respiratory signal is generated from the raw respiratory signal using initially detected peak and valley points. Valley points are used to define the spectral range of the respiratory signal, and band-limited filtering is performed based on this defined frequency range.

In the second step, peaks and valleys are re-detected in the filtered respiratory signal. During this stage, users may manually refine peak and valley locations using the editing function: left mouse clicks add points, whereas right mouse clicks remove points.

Figure 9. Functional description: Module2-AnalysisData, Respiratory signal processing part

#### 4.2.3.3. Cardiac signal processing

PPG signals are processed using a pipeline analogous to respiratory analysis. Peaks are detected to compute inter-beat intervals (IBIs), which are subsequently converted into heart rate (BPM). Temporal heart rate dynamics are visualized, and low-frequency (LF, 0.04-0.15 Hz) and high-frequency (HF, 0.15-0.4 Hz) components are derived for autonomic characterization.

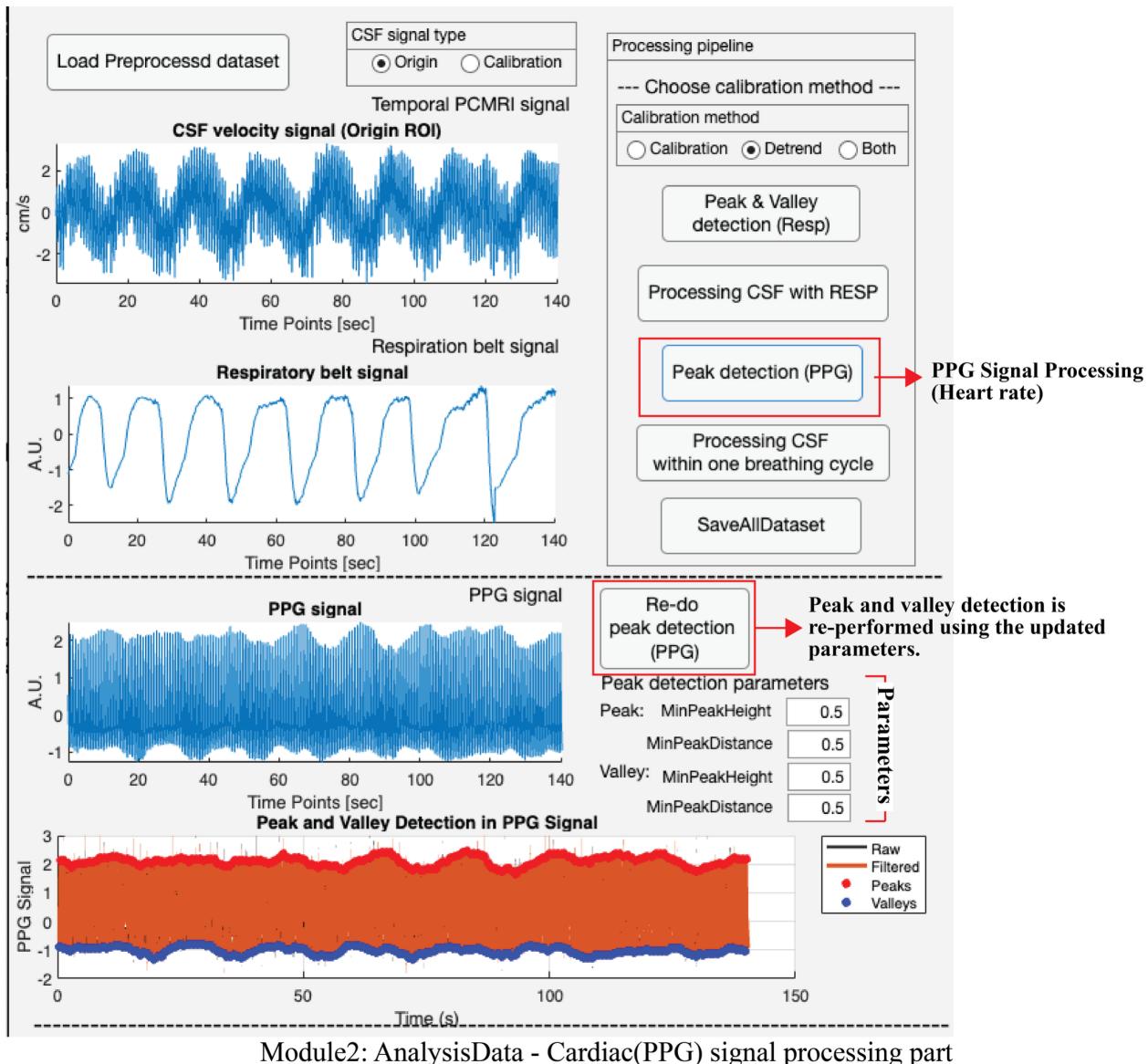


Figure 10. Functional description: Module2-AnalysisData, Cardiac(PPG) signal processing part

#### 4.2.3.4. Integrated visualization

Upon completion of signal processing, the toolbox presents synchronized visualizations of respiratory signals, CSF flow rate (raw, respiratory-filtered, and cardiac-filtered), volume

displacement (raw and filtered), and heart rate signals. This integrated display allows users to inspect CSF dynamics and physiological interactions within a single temporal framework.

Additionally, *NeuroNetFlow* provides functionality to normalize individual respiratory cycles to a 0–100% breathing phase. Signals are averaged across cycles to generate representative respiratory-cycle-resolved profiles, from which extrema values and their relative timing within the breathing cycle are extracted for further analysis.

#### **4.2.4. Output and feature export**

All processed signals and extracted features can be saved in MATLAB .mat format. Predefined quantitative features are also automatically summarized into structured tables and exported as .xlsx files, enabling downstream statistical analysis across participants.

The extracted features include temporal CSF velocity and flow metrics, respiratory- and cardiac-resolved integrals, spectral characteristics, and breathing-cycle-normalized descriptors. These outputs are designed to facilitate group-level statistical analysis of CSF dynamics and their coupling with physiological signals.

### **4.3. Exported features for downstream statistical analysis**

*NeuroNetFlow* exports a standardized set of quantitative features derived from preprocessed and aligned PC-MRI CSF signals and physiological recordings (respiratory belt and PPG). These features are saved per participant and can be aggregated across participants into a single spreadsheet (e.g., .xlsx) to support group-level statistical analyses (e.g., between-group comparisons, correlation/regression modeling, and longitudinal evaluation). Each feature is computed from a clearly defined processed signal stage (temporal, spectral, respiratory landmark-based, or breathing-cycle normalized), enabling reproducible and interpretable downstream analyses.

#### **A) Temporal PC-MRI velocity features (CSF velocity; temporal domain)**

Mean\_velocity: Mean CSF velocity over time – Temporal PC-MRI velocity signal

Sum\_velocity: Sum (temporal accumulation) of CSF velocity values – Temporal PC-MRI velocity signal

Mean\_absolute\_velocity: Mean of absolute CSF velocity values – Temporal PC-MRI velocity signal

SD\_of\_velocity: Standard deviation of CSF velocity – Temporal PC-MRI velocity signal

SD\_of\_absolute\_velocity: Standard deviation of absolute CSF velocity – Temporal PC-MRI velocity signal

CI\_with\_velocity: Confidence interval (CI) of mean CSF velocity – Temporal PC-MRI velocity signal

## CI\_with\_absolute\_velocity: Confidence interval (CI) of mean absolute CSF velocity – Temporal PC-MRI velocity signal

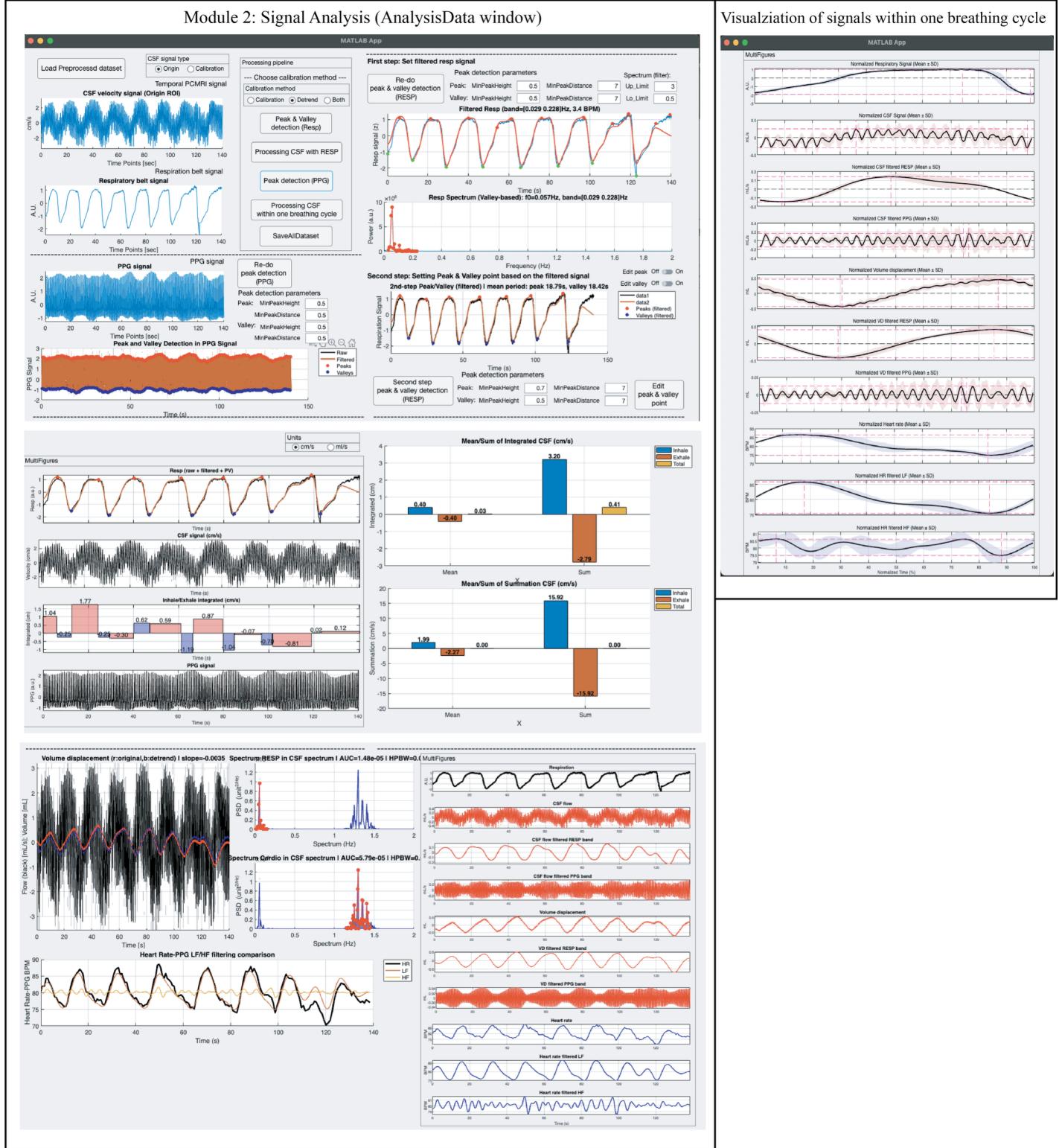


Figure 11. Functional description: Module2-AnalysisData, The entire window view

## A-1) Peak/valley summary (velocity)

Mean\_Peak\_points\_of\_CSF\_velocity: Mean of detected peak amplitudes in CSF velocity – Temporal PC-MRI velocity signal (peak-detected)

SD\_of\_Peak\_points\_of\_CSF\_velocity: Standard deviation of detected velocity peaks – Temporal PC-MRI velocity signal (peak-detected)

CI\_with\_Peak\_points\_of\_CSF\_velocity: Confidence interval of detected velocity peaks – Temporal PC-MRI velocity signal (peak-detected)

Maximum\_velocity\_from\_Traditionalcode: Maximum CSF velocity computed using a conventional (“Traditionalcode”) method – Temporal PC-MRI velocity signal

Mean\_Valley\_points\_of\_CSF\_velocity: Mean of detected valley amplitudes in CSF velocity – Temporal PC-MRI velocity signal (valley-detected)

SD\_of\_Valley\_points\_of\_CSF\_velocity: Standard deviation of detected velocity valleys – Temporal PC-MRI velocity signal (valley-detected)

CI\_with\_Valley\_points\_of\_CSF\_velocity: Confidence interval of detected velocity valleys – Temporal PC-MRI velocity signal (valley-detected)

Minimum\_velocity\_from\_Traditionalcode: Minimum CSF velocity computed using a conventional (“Traditionalcode”) method – Temporal PC-MRI velocity signal

## A-2) Peak–valley composite descriptors (velocity)

Mean\_Difference\_between\_Peak\_Valley: Sum of mean peak and mean valley (MeanPeak + MeanValley) – Temporal PC-MRI velocity peaks/valleys

SD\_Difference\_between\_Peak\_Valley: Sum of SD of peaks and SD of valleys (SDPeak + SDValley) – Temporal PC-MRI velocity peaks/valleys

CI\_with\_Difference\_between\_Peak\_Valley: Sum of CI of peaks and CI of valleys (CIPeak + CIValley) – Temporal PC-MRI velocity peaks/valleys

## B) Temporal PC-MRI flow-rate features (velocity → flow rate; temporal domain)

Mean\_Flow\_rate: Mean CSF flow rate – Temporal CSF flow-rate signal (ml/s)

Sum\_Flow\_rate: Sum of CSF flow rate values – Temporal CSF flow-rate signal (ml/s)

Mean\_absolute\_Flow\_rate: Mean absolute CSF flow rate – Temporal CSF flow-rate signal (ml/s)

SD\_of\_Flow\_rate: Standard deviation of CSF flow rate – Temporal CSF flow-rate signal (ml/s)

SD\_of\_absolute\_Flow\_rate: Standard deviation of absolute CSF flow rate – Temporal CSF flow-rate signal (ml/s)

CI\_with\_Flow\_rate: Confidence interval (CI) of mean CSF flow rate – Temporal CSF flow-rate signal (ml/s)

CI\_with\_absolute\_Flow\_rate: Confidence interval (CI) of mean absolute CSF flow rate – Temporal CSF flow-rate signal (ml/s)

### B-1) Peak/valley summary (flow rate)

Mean\_Peak\_points\_of\_CSF\_Flow\_rate: Mean of detected peak amplitudes in CSF flow rate – Temporal CSF flow-rate signal (peak-detected)

SD\_of\_Peak\_points\_of\_CSF\_Flow\_rate: Standard deviation of detected flow-rate peaks – Temporal CSF flow-rate signal (peak-detected)

CI\_with\_Peak\_points\_of\_CSF\_Flow\_rate: Confidence interval of detected flow-rate peaks – Temporal CSF flow-rate signal (peak-detected)

Maximum\_Flow\_rate\_from\_Traditionalcode: Maximum CSF flow rate computed using a conventional (“Traditionalcode”) method – Temporal CSF flow-rate signal

Mean\_Valley\_points\_of\_CSF\_Flow\_rate: Mean of detected valley amplitudes in CSF flow rate – Temporal CSF flow-rate signal (valley-detected)

SD\_of\_Valley\_points\_of\_CSF\_Flow\_rate: Standard deviation of detected flow-rate valleys – Temporal CSF flow-rate signal (valley-detected)

CI\_with\_Valley\_points\_of\_CSF\_Flow\_rate: Confidence interval of detected flow-rate valleys – Temporal CSF flow-rate signal (valley-detected)

Minimum\_flow\_rate\_from\_Traditionalcode: Minimum CSF flow rate computed using a conventional (“Traditionalcode”) method – Temporal CSF flow-rate signal

### B-2) Peak–valley composite descriptors (flow rate)

Mean\_Diff\_between\_Peak\_valley\_Flow\_rate: Sum of mean peak and mean valley (MeanPeak + MeanValley) – Temporal flow-rate peaks/valleys

SD\_Diff\_between\_Peak\_valley\_Flow\_rate: Sum of SD of peaks and SD of valleys (SDPeak + SDValley) – Temporal flow-rate peaks/valleys

CI\_with\_Diff\_between\_Peak\_valley\_flow\_rate: Sum of CI of peaks and CI of valleys (CIPeak + CIValley) – Temporal flow-rate peaks/valleys

### C) Respiration-phase-resolved integrated flow features (inhale/exhale segmentation)

**(Respiratory peaks/valleys define inhale/exhale intervals; flow rate is integrated within each interval.)**

Inhale\_mean\_of\_Integrated\_CSF\_Flow\_rate: Mean integrated flow rate during inhale segments – Integrated flow rate per inhale segment (PC-MRI + respiratory landmarks)

Exhale\_mean\_of\_Integrated\_CSF\_Flow\_rate: Mean integrated flow rate during exhale segments – Integrated flow rate per exhale segment (PC-MRI + respiratory landmarks)

Total\_mean\_of\_Integrated\_CSF\_Flow\_rate: Mean integrated flow rate across inhale + exhale segments – Integrated flow rate (PC-MRI + respiratory landmarks)

Inhale\_sum\_of\_Integrated\_CSF\_Flow\_rate: Sum of integrated flow rate across inhale segments – Integrated flow rate per inhale segment (PC-MRI + respiratory landmarks)

Exhale\_sum\_of\_Integrated\_CSF\_Flow\_rate: Sum of integrated flow rate across exhale segments – Integrated flow rate per exhale segment (PC-MRI + respiratory landmarks)

Total\_sum\_of\_Integrated\_CSF\_Flow\_rate: Sum of integrated flow rate across inhale + exhale segments – Integrated flow rate (PC-MRI + respiratory landmarks)

#### **D) Volume displacement stability / detrending diagnostic**

Slope\_before\_detrend\_of\_volume\_displacement: Linear slope of the integrated flow-rate signal prior to detrending (used in volume displacement computation) – Pre-detrend volume-displacement precursor signal (integrated flow rate)

#### **E) Spectral PC-MRI features (respiratory/cardiac band quantification)**

**(Computed from the frequency spectrum of the CSF velocity signal.)**

##### **E-1) Respiratory-band spectrum features**

AUC\_of\_RESP: Area under the curve (AUC) within the predefined respiratory frequency band – CSF velocity spectrum (respiratory band)

SD\_of\_predefine\_RESP\_spectrum: Standard deviation of spectral power values within the respiratory band – CSF velocity spectrum (respiratory band)

HPBW\_range\_of\_predefine\_RESP\_spectrum: Half-power bandwidth range around the dominant respiratory-band peak – CSF velocity spectrum (respiratory band)

##### **E-2) Cardiac-band spectrum features**

AUC\_of\_Cardiac: Area under the curve (AUC) within the predefined cardiac frequency band – CSF velocity spectrum (cardiac band)

SD\_of\_predefine\_Cardiac\_spectrum: Standard deviation of spectral power values within the cardiac band – CSF velocity spectrum (cardiac band)

**HPBW\_range\_of\_predefine\_Cardiac\_spectrum:** Half-power bandwidth range around the dominant cardiac-band peak – CSF velocity spectrum (cardiac band)

### **E-3) Cross-band ratio**

**Ratio\_AUC\_RESP\_PPG:** Ratio of respiratory-band AUC to cardiac-band AUC (as implemented in the toolbox) – CSF velocity spectrum (respiratory vs cardiac bands)

## **F) Respiratory timing features (from detected respiratory peaks/valleys)**

**Mean\_Peak\_time\_interval:** Mean peak-to-peak interval (average breathing period based on peaks) – Respiratory signal (peak-detected)

**Mean\_Valley\_time\_interval:** Mean valley-to-valley interval (average breathing period based on valleys) – Respiratory signal (valley-detected)

**SD\_Peak\_time\_interval:** Standard deviation of peak-to-peak intervals – Respiratory signal (peak-detected)

**SD\_Valley\_time\_interval:** Standard deviation of valley-to-valley intervals – Respiratory signal (valley-detected)

## **G) One-breathing-cycle normalized features (0–100% cycle; cycle-averaged patterns)**

**(Each respiratory cycle is normalized to 0–100%, signals are averaged across cycles, and extrema timing/value are extracted.)**

### **G-1) Maxima timing/value**

**Max\_norm\_TimeP\_RESP:** Breathing-cycle percentage at which respiratory signal reaches its maximum – Cycle-normalized respiratory signal

**Max\_norm\_TimeP\_Flow\_rate:** Breathing-cycle percentage of maximum flow rate – Cycle-normalized CSF flow rate

**Max\_norm\_TimeP\_value\_of\_Flow\_rate:** Maximum flow rate value – Cycle-normalized CSF flow rate (ml/s)

**Max\_norm\_TimeP\_Flow\_rate\_f\_RESP:** Breathing-cycle percentage of maximum flow rate after respiratory-band filtering – Cycle-normalized respiratory-filtered flow rate

**Max\_norm\_TimeP\_value\_of\_Flow\_rate\_f\_RESP:** Maximum respiratory-filtered flow rate value – Cycle-normalized respiratory-filtered flow rate (ml/s)

**Max\_norm\_TimeP\_Flow\_rate\_f\_Cardiac:** Breathing-cycle percentage of maximum flow rate after cardiac-band filtering – Cycle-normalized cardiac-filtered flow rate

Max\_norm\_TimeP\_value\_of\_Flow\_rate\_f\_Cardiac: Maximum cardiac-filtered flow rate value – Cycle-normalized cardiac-filtered flow rate (ml/s)

Max\_norm\_TimeP\_volume\_displacement: Breathing-cycle percentage of maximum volume displacement – Cycle-normalized volume displacement

Max\_norm\_TimeP\_value\_of\_volume\_displacement: Maximum volume displacement value – Cycle-normalized volume displacement (ml)

Max\_norm\_TimeP\_volume\_displacement\_f\_RESP: Breathing-cycle percentage of maximum respiratory-filtered volume displacement – Cycle-normalized respiratory-filtered volume displacement

Max\_norm\_TimeP\_value\_of\_volume\_displacement\_f\_RESP: Maximum respiratory-filtered volume displacement value – Cycle-normalized respiratory-filtered volume displacement (ml)

Max\_norm\_TimeP\_volume\_displacement\_f\_Cardiac: Breathing-cycle percentage of maximum cardiac-filtered volume displacement – Cycle-normalized cardiac-filtered volume displacement

Max\_norm\_TimeP\_value\_of\_volume\_displacement\_f\_Cardiac: Maximum cardiac-filtered volume displacement value – Cycle-normalized cardiac-filtered volume displacement (ml)

Max\_norm\_TimeP\_HR: Breathing-cycle percentage of maximum heart rate – Cycle-normalized heart rate

Max\_norm\_TimeP\_value\_of\_HR: Maximum heart rate value – Cycle-normalized heart rate (BPM)

Max\_norm\_TimeP\_HR\_f\_LF\_band: Breathing-cycle percentage of maximum LF-filtered heart rate – Cycle-normalized LF-filtered HR (0.04–0.15 Hz)

Max\_norm\_TimeP\_value\_of\_HR\_f\_LF\_band: Maximum LF-filtered heart rate value – Cycle-normalized LF-filtered HR (BPM)

Max\_norm\_TimeP\_HR\_f\_HF\_band: Breathing-cycle percentage of maximum HF-filtered heart rate – Cycle-normalized HF-filtered HR (0.15–0.4 Hz)

Max\_norm\_TimeP\_value\_of\_HR\_f\_HF\_band: Maximum HF-filtered heart rate value – Cycle-normalized HF-filtered HR (BPM)

## G-2) Minima timing/value

Min\_norm\_TimeP\_RESP: Breathing-cycle percentage at which respiratory signal reaches its minimum – Cycle-normalized respiratory signal

Min\_norm\_TimeP\_Flow\_rate: Breathing-cycle percentage of minimum flow rate – Cycle-normalized CSF flow rate

Min\_norm\_TimeP\_value\_of\_Flow\_rate: Minimum flow rate value – Cycle-normalized CSF flow rate (ml/s)

Min\_norm\_TimeP\_Flow\_rate\_f\_RESP: Breathing-cycle percentage of minimum flow rate after respiratory-band filtering – Cycle-normalized respiratory-filtered flow rate

Min\_norm\_TimeP\_value\_of\_Flow\_rate\_f\_RESP: Minimum respiratory-filtered flow rate value – Cycle-normalized respiratory-filtered flow rate (ml/s)

Min\_norm\_TimeP\_Flow\_rate\_f\_Cardiac: Breathing-cycle percentage of minimum flow rate after cardiac-band filtering – Cycle-normalized cardiac-filtered flow rate

Min\_norm\_TimeP\_value\_of\_Flow\_rate\_f\_Cardiac: Minimum cardiac-filtered flow rate value – Cycle-normalized cardiac-filtered flow rate (ml/s)

Min\_norm\_TimeP\_volume\_displacement: Breathing-cycle percentage of minimum volume displacement – Cycle-normalized volume displacement

Min\_norm\_TimeP\_value\_of\_volume\_displacement: Minimum volume displacement value – Cycle-normalized volume displacement (ml)

Min\_norm\_TimeP\_volume\_displacement\_f\_RESP: Breathing-cycle percentage of minimum respiratory-filtered volume displacement – Cycle-normalized respiratory-filtered volume displacement

Min\_norm\_TimeP\_value\_of\_volume\_displacement\_f\_RESP: Minimum respiratory-filtered volume displacement value – Cycle-normalized respiratory-filtered volume displacement (ml)

Min\_norm\_TimeP\_volume\_displacement\_f\_Cardiac: Breathing-cycle percentage of minimum cardiac-filtered volume displacement – Cycle-normalized cardiac-filtered volume displacement

Min\_norm\_TimeP\_value\_of\_volume\_displacement\_f\_Cardiac: Minimum cardiac-filtered volume displacement value – Cycle-normalized cardiac-filtered volume displacement (ml)

Min\_norm\_TimeP\_HR: Breathing-cycle percentage of minimum heart rate – Cycle-normalized heart rate

Min\_norm\_TimeP\_value\_of\_HR: Minimum heart rate value – Cycle-normalized heart rate (BPM)

Min\_norm\_TimeP\_HR\_f\_LF\_band: Breathing-cycle percentage of minimum LF-filtered heart rate – Cycle-normalized LF-filtered HR (0.04–0.15 Hz)

Min\_norm\_TimeP\_value\_of\_HR\_f\_LF\_band: Minimum LF-filtered heart rate value – Cycle-normalized LF-filtered HR (BPM)

Min\_norm\_TimeP\_HR\_f\_HF\_band: Breathing-cycle percentage of minimum HF-filtered heart rate – Cycle-normalized HF-filtered HR (0.15–0.4 Hz)

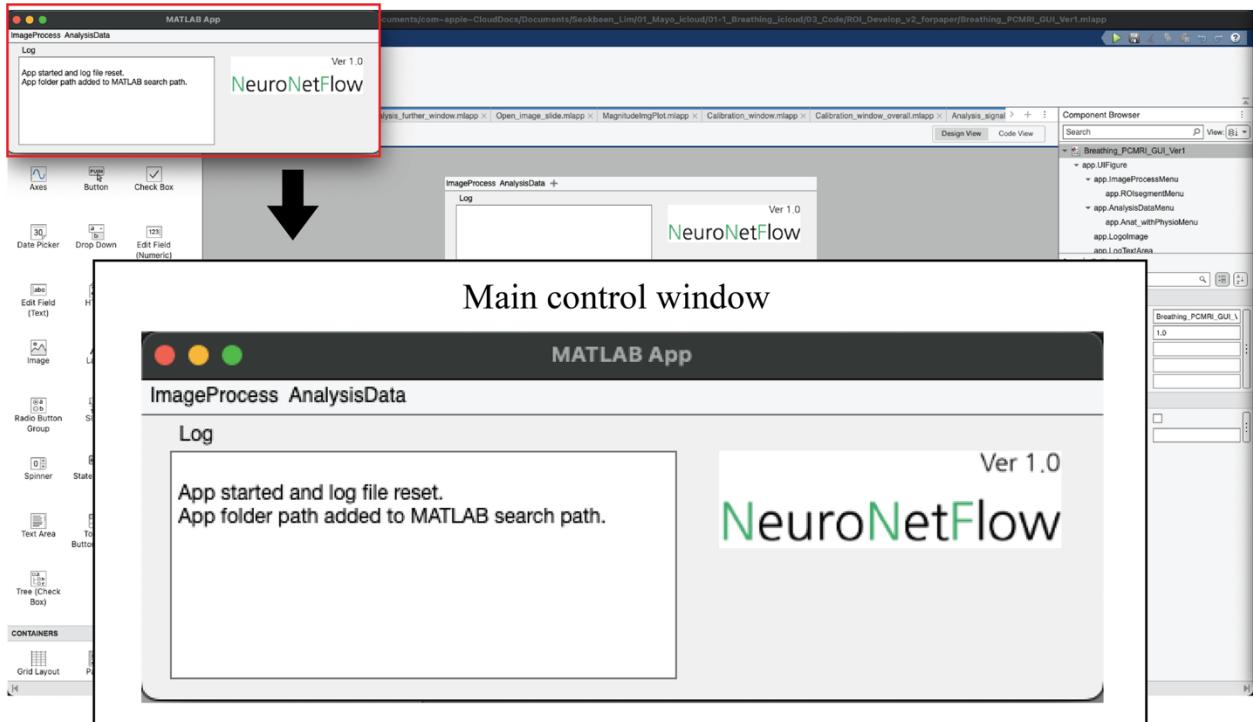
Min\_norm\_TimeP\_value\_of\_HR\_f\_HF\_band: Minimum HF-filtered heart rate value – Cycle-normalized HF-filtered HR (BPM)

## 5. Step-by-Step Workflow

### 5.1. Check the directory

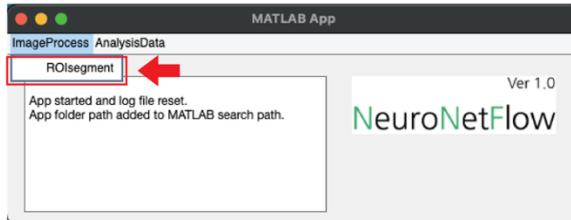
Name	Date Modified	Size	Kind
Analysis_further_window.mlapp	Today at 12:01AM	233 KB	MATLAB App
Analysis_signals_window.mlapp	Today at 12:01AM	445 KB	MATLAB App
Breathing_PCMRI_GUI_Ver1.mlapp	Today at 12:01AM	38 KB	MATLAB App
Calibration_window_overall.mlapp	Today at 12:01AM	186 KB	MATLAB App
Calibration_window.mlapp	Today at 12:01AM	131 KB	MATLAB App
CSF_onecycle_Window.mlapp	Today at 12:01AM	23 KB	MATLAB App
> example	Today at 12:01AM	--	Folder
LICENSE	Today at 12:01AM	1 KB	Plain Text
MagnitudeImgPlot.mlapp	Today at 12:01AM	29 KB	MATLAB App
Open_image_slide.mlapp	Today at 12:01AM	219 KB	MATLAB App
README.md	Today at 12:01AM	101 bytes	Markdown File
ReadPhilipsScanPhysLog.m	Today at 12:01AM	17 KB	MATLAB Code
> src	Today at 12:01AM	--	Folder

### 5.2. Run for the main control window



### 5.3. Load image dataset

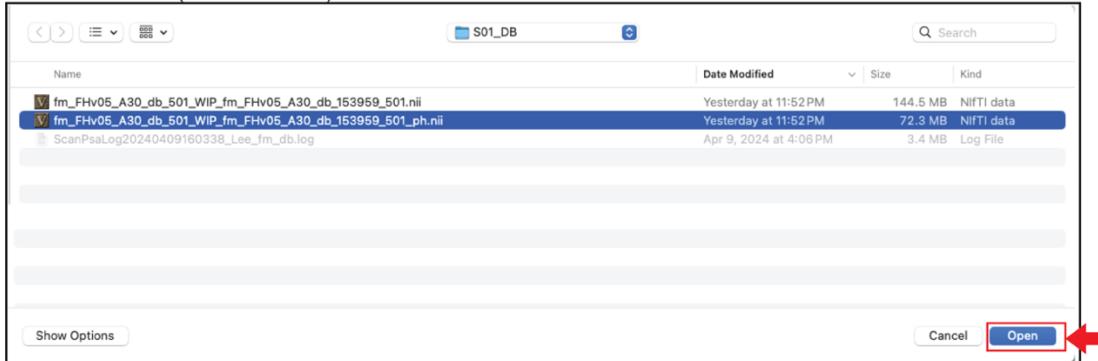
1. Click ROIsegment



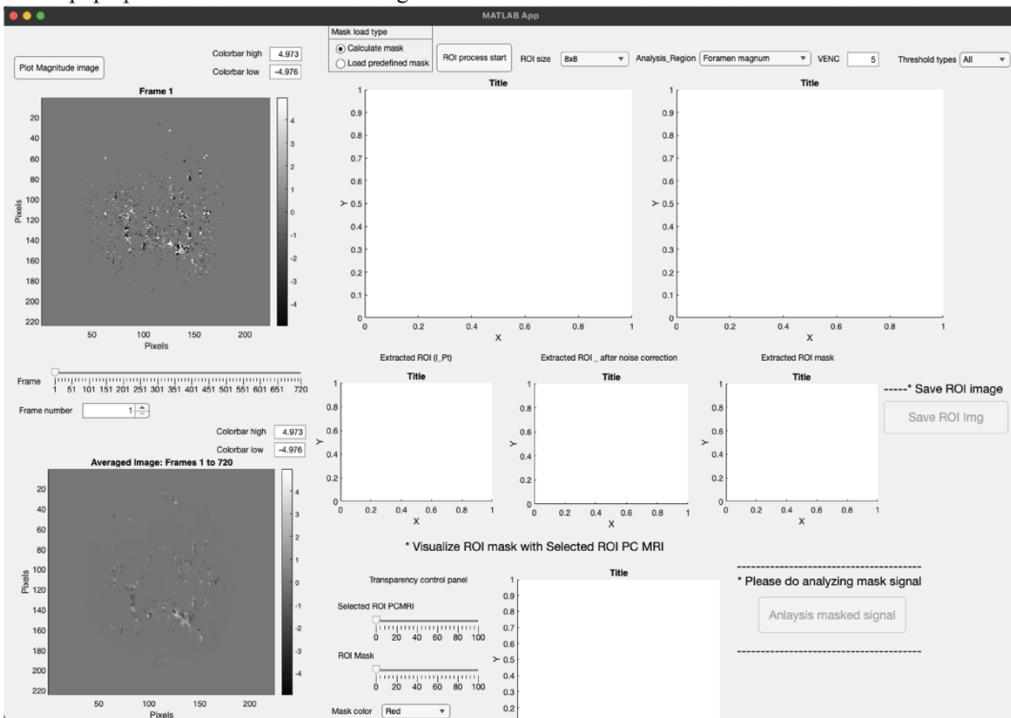
2. Find the directory that saved the data



3. Select the data (Phase .nii data)

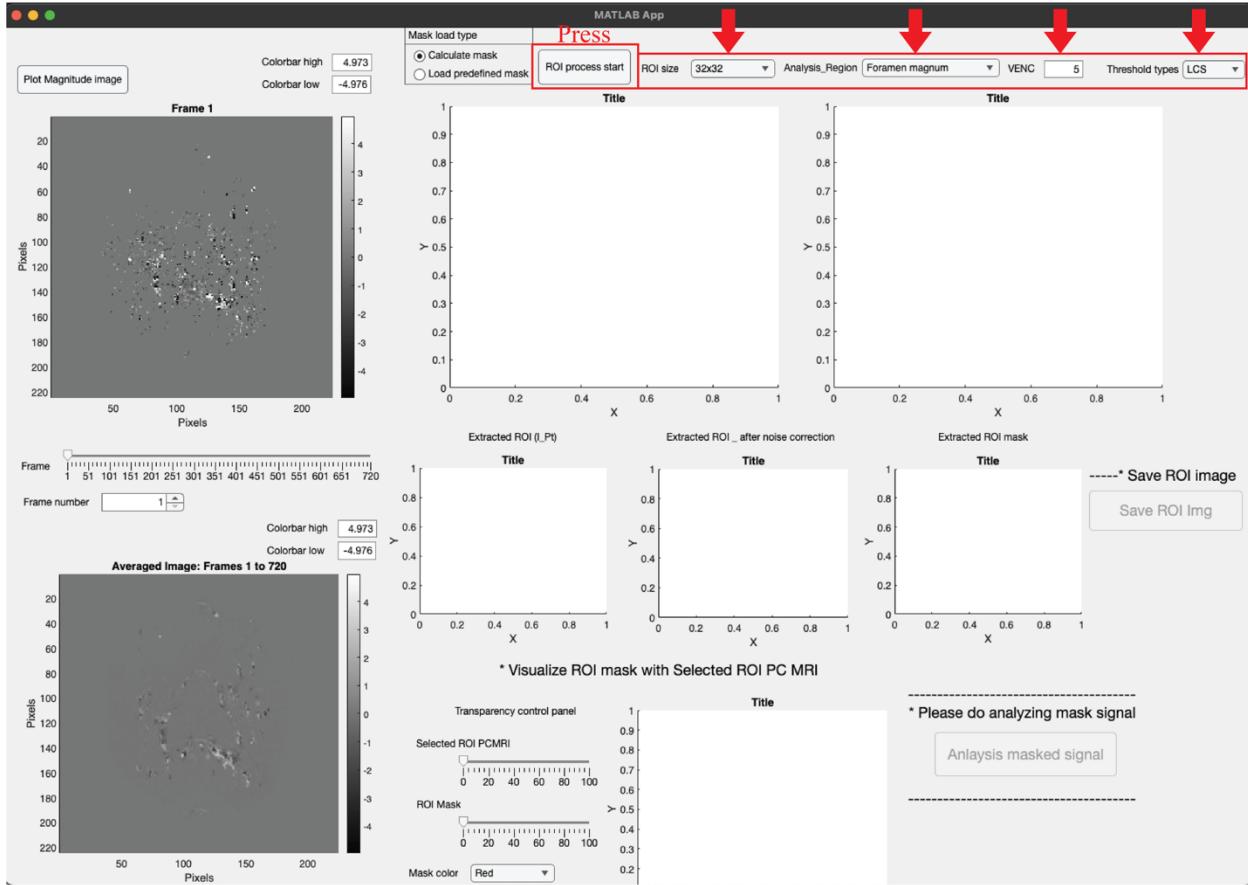


4. Start to pop up the window for the ROI segment

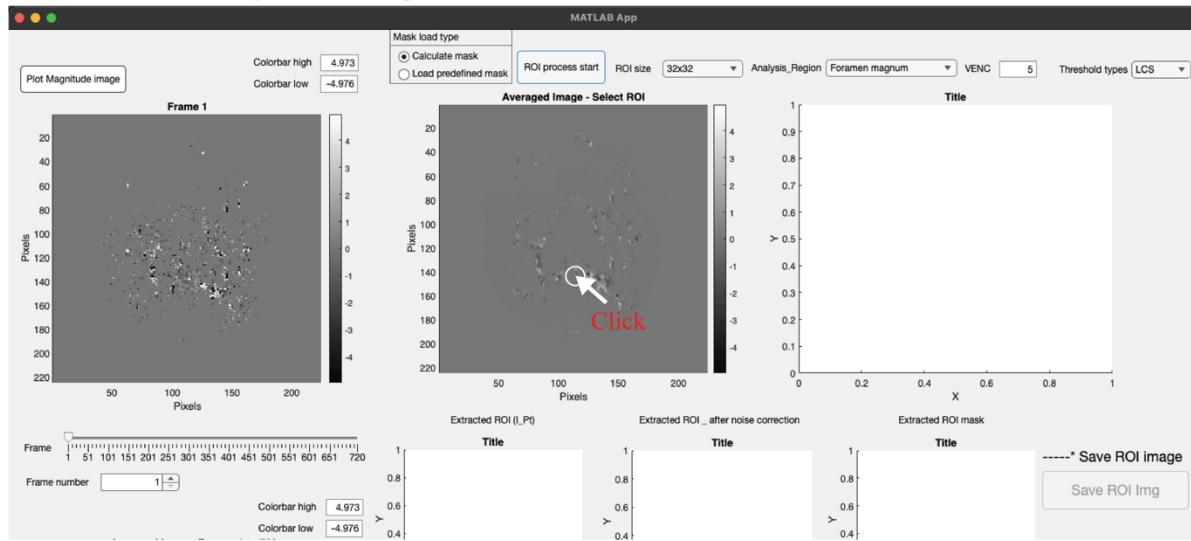


## 5.4. Conduct segmenting ROI

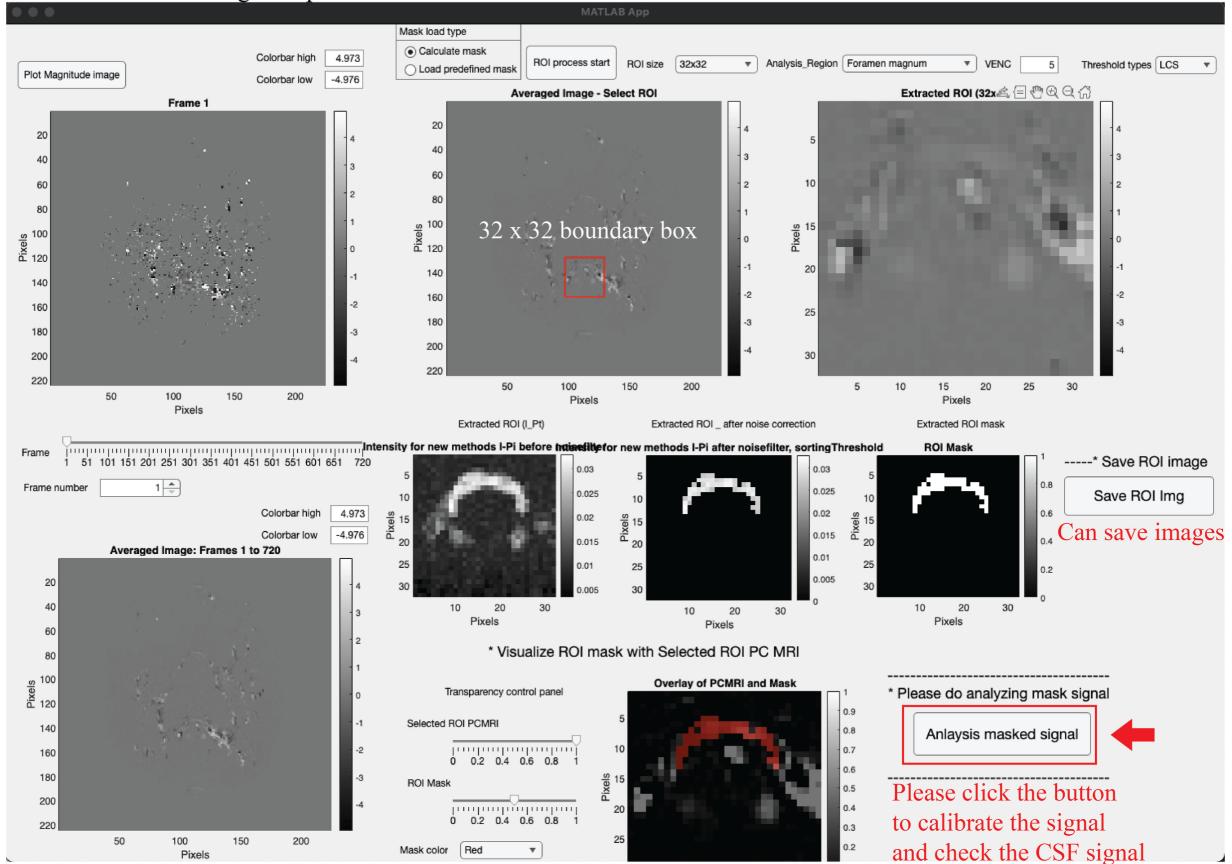
1. Select parameters and press “ROI process start” button



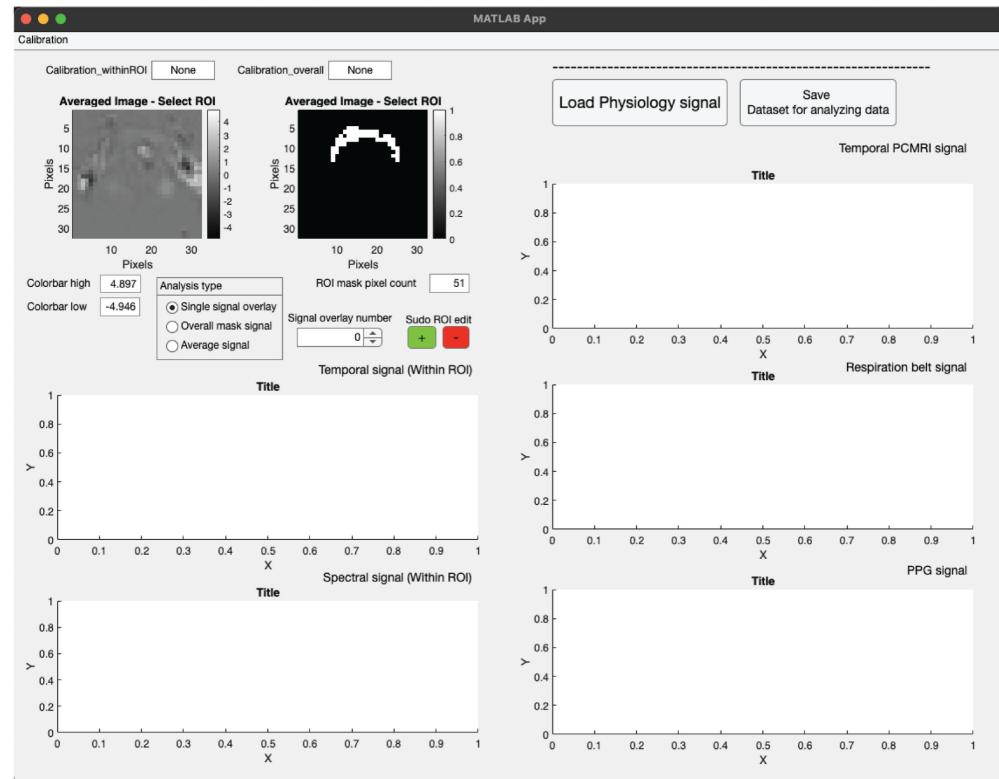
2. Please click wherever you want to segment



### 3. Coudct the ROI segment process and create a mask

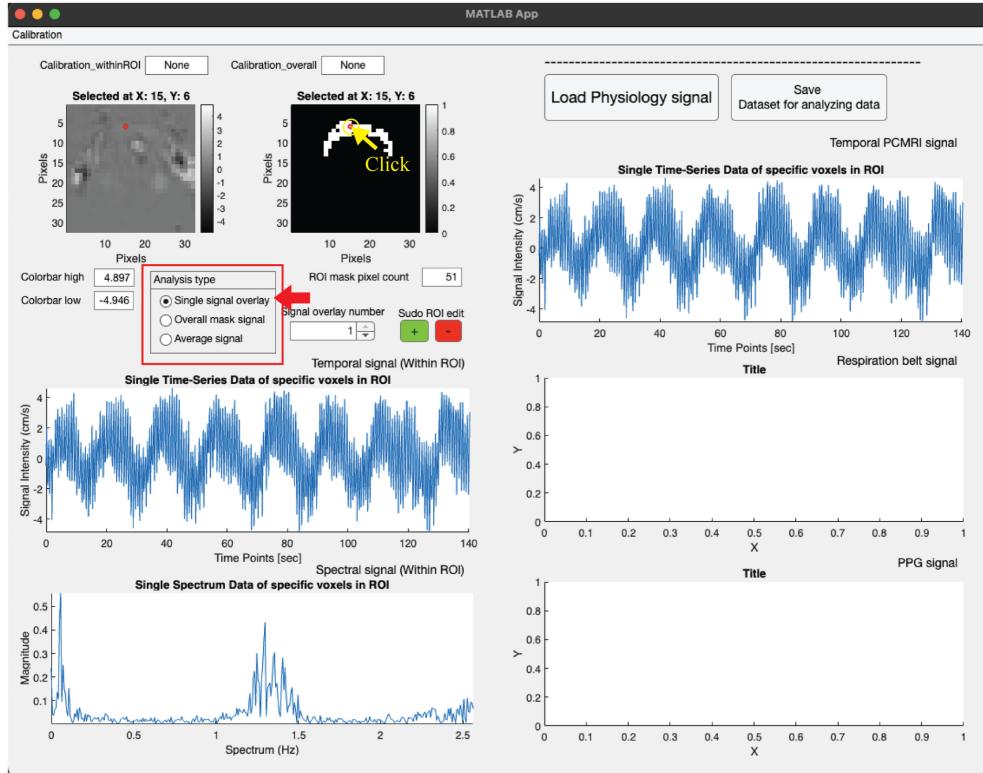


### 4. Start to pop up further analysis window

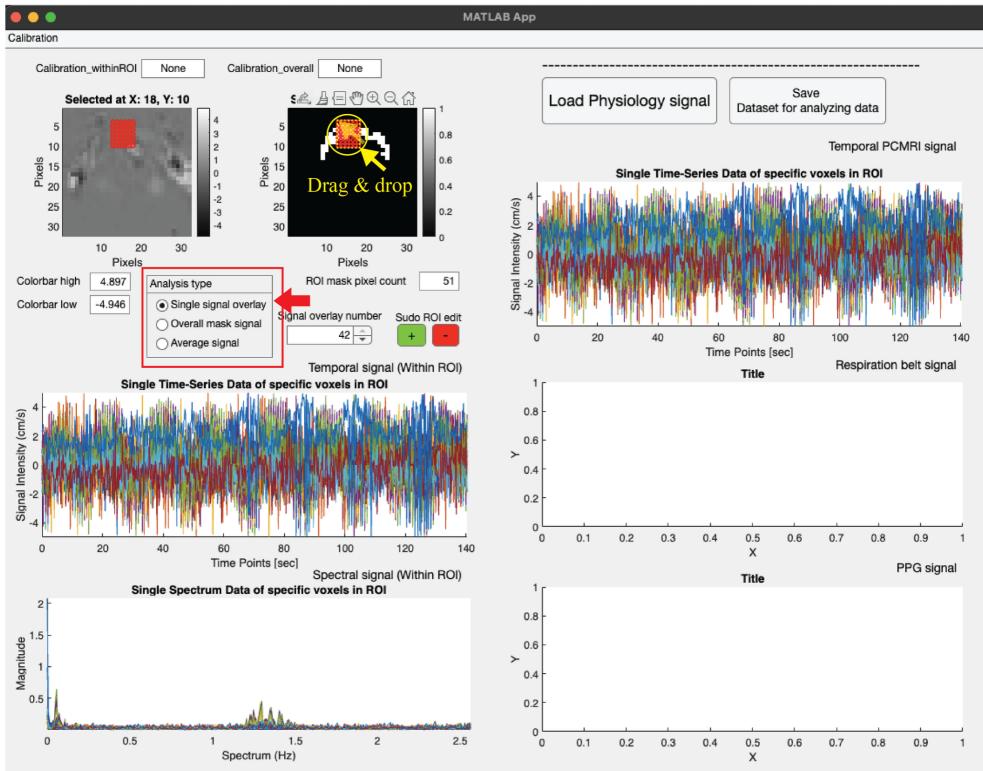


## 5.5. Check the signal, conduct sudo mask correction, and calibrate the signal

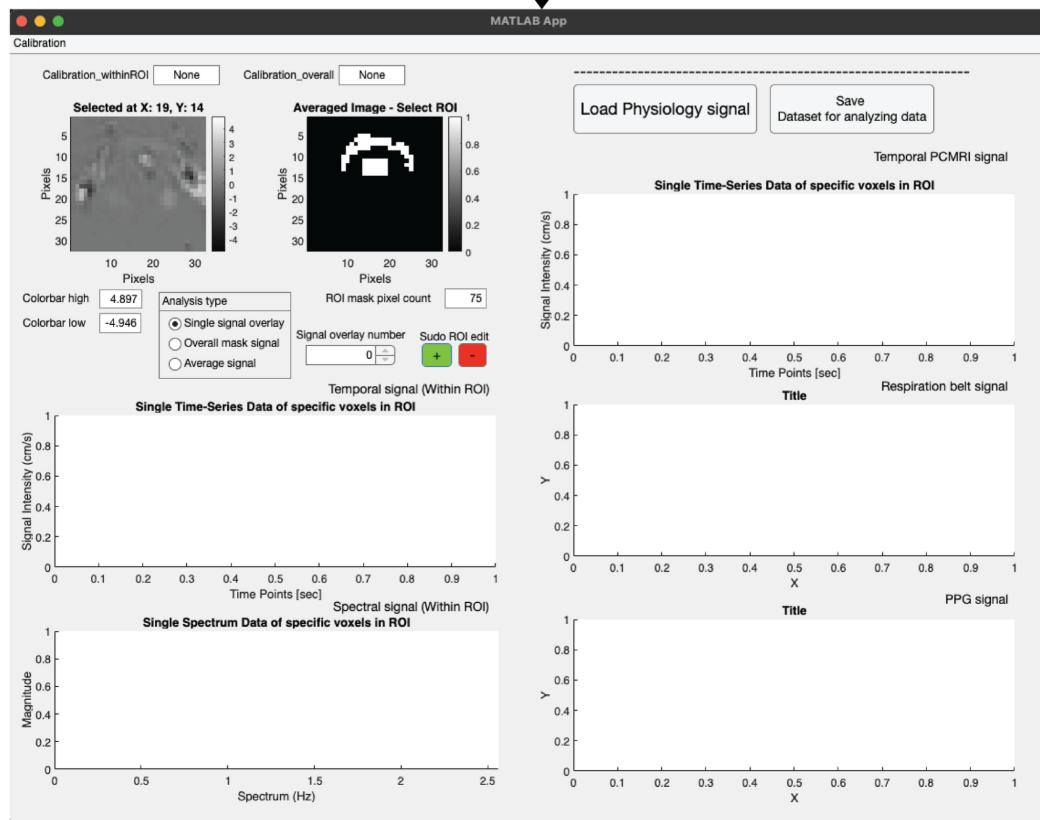
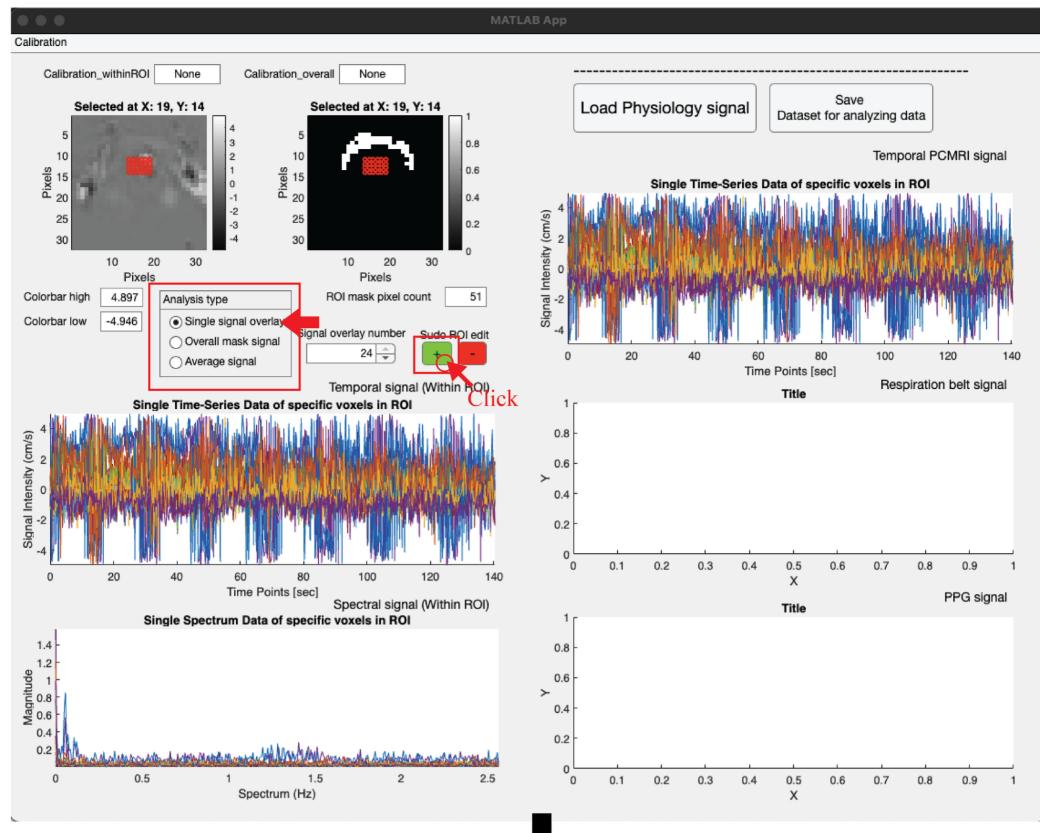
- When you click the voxel in the single signal overlay function, you can see the temporal & spectral signals at that voxel



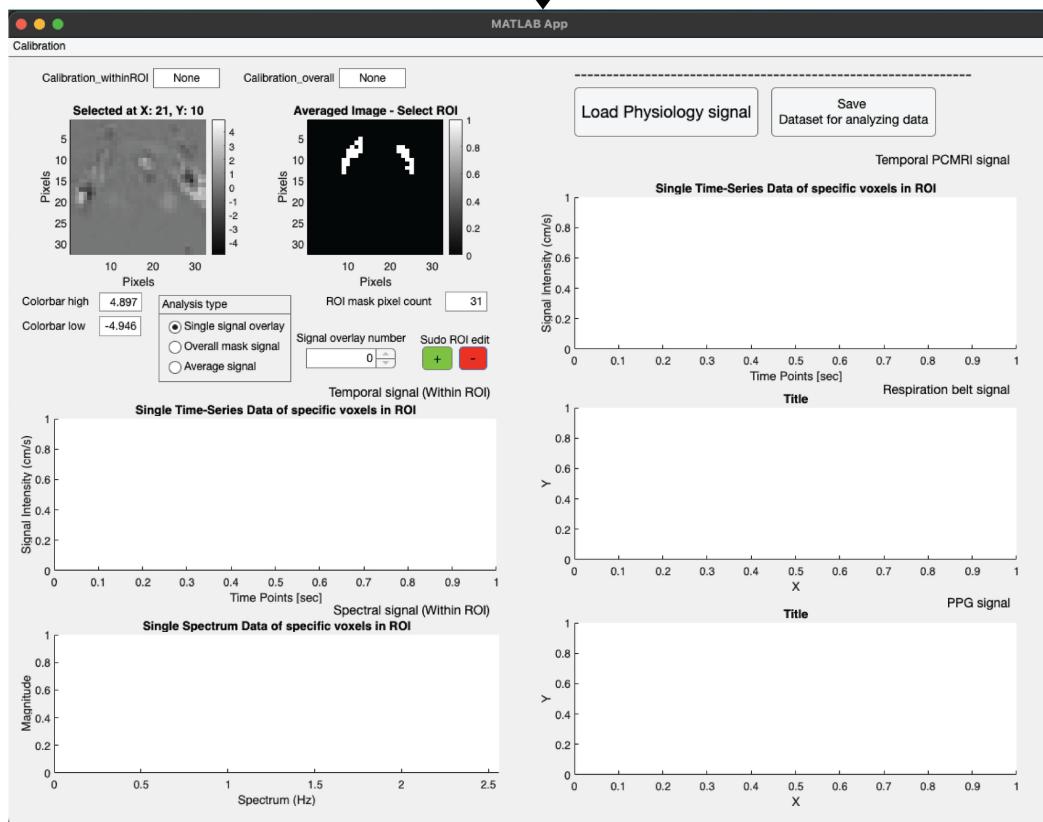
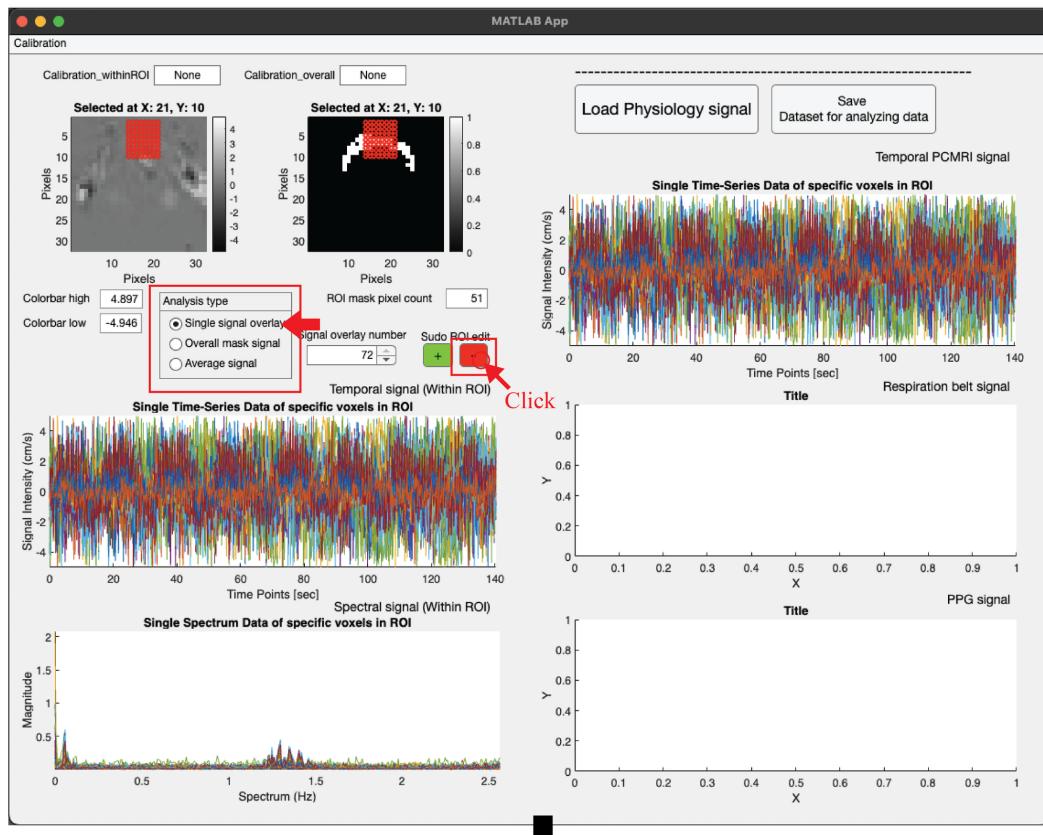
- When you drag & drop the voxels in the single signal overlay function, you can see the multiple temporal & spectral signals at those voxels.



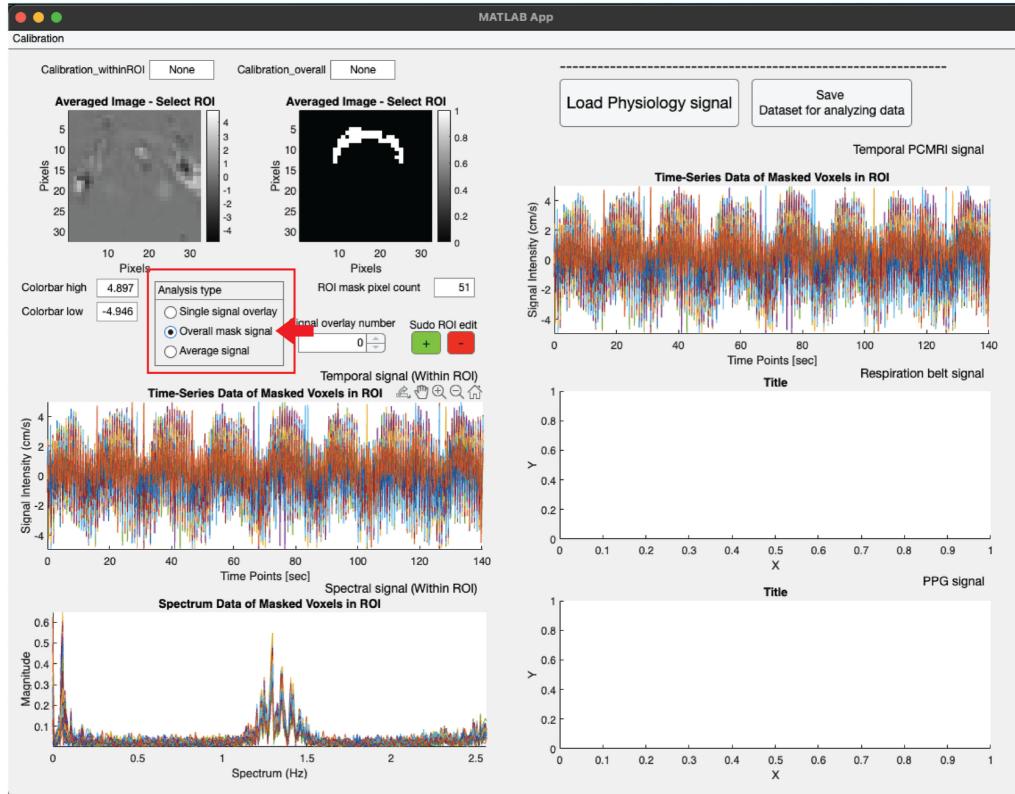
3. If you press the “+” green button while there are selected voxels, you can add those voxels as an ROI mask area.



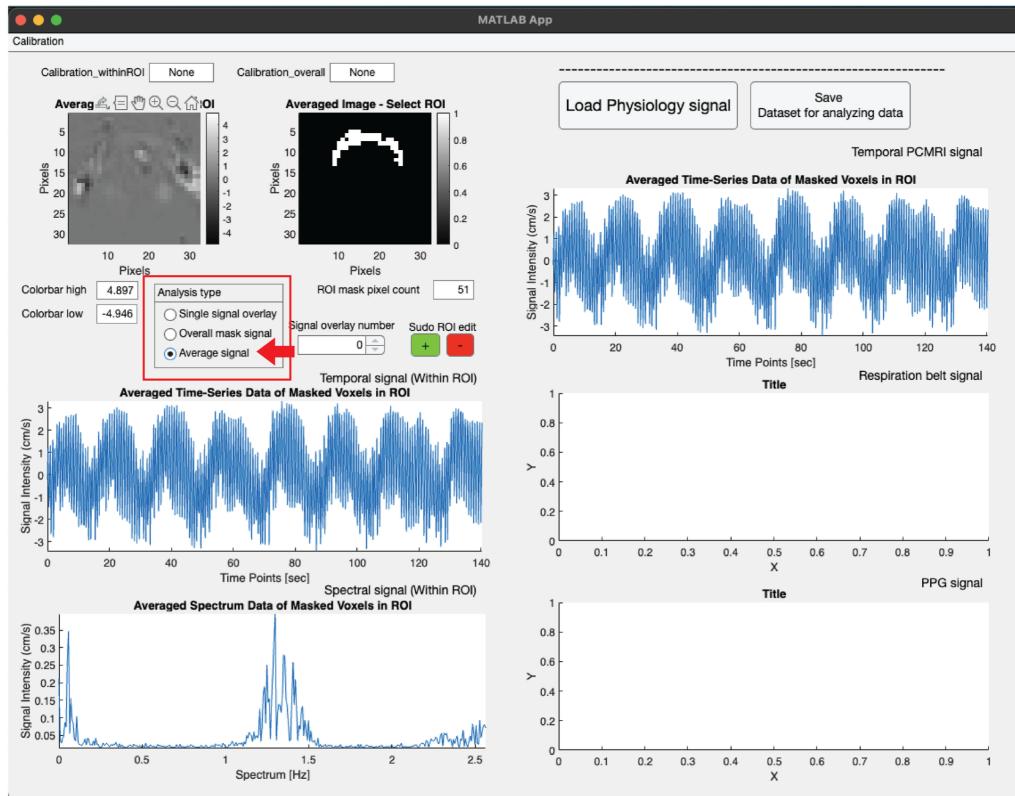
4. If you press the “–” red button while there are selected voxels, you can get rid of those voxels from an ROI mask area.



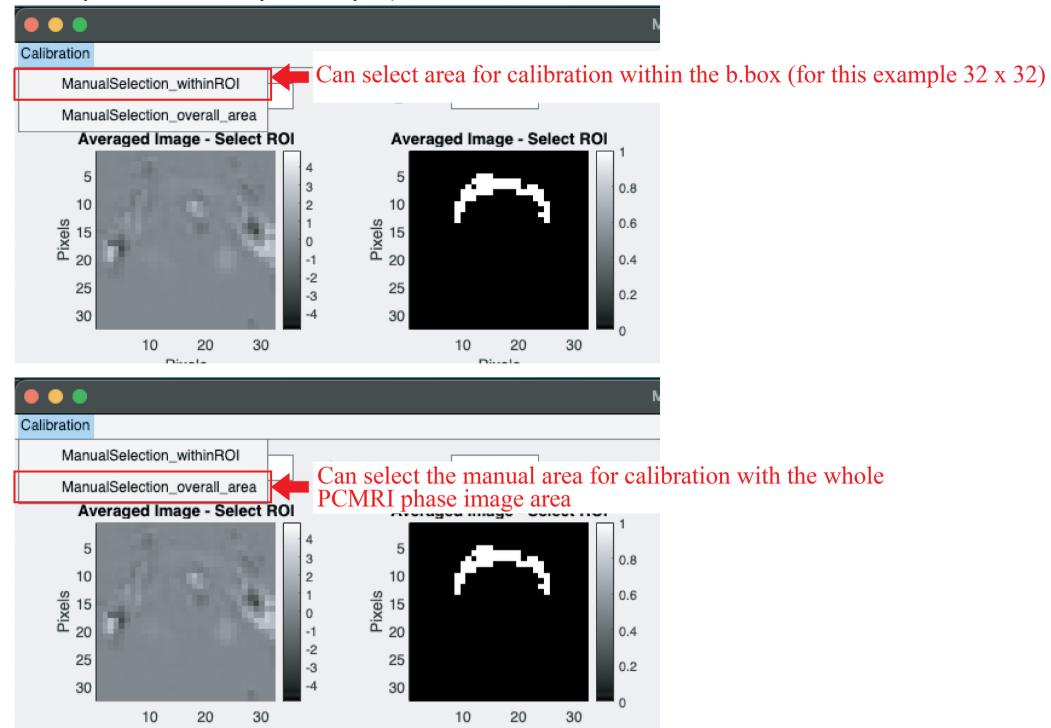
5. If you change the Analysis type to “Overall mask signal”, you can see the temporal & spectral signals of voxels within the ROI mask



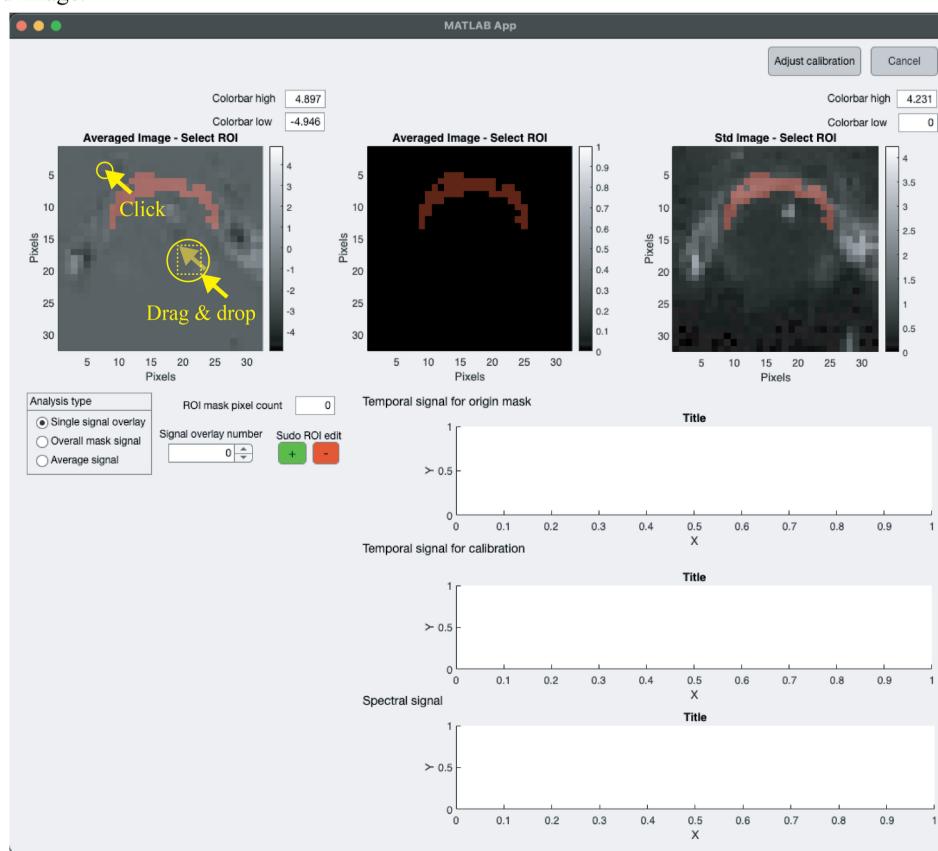
6. If you change the Analysis type to “Average signal”, you can see the averaged temporal & spectral signals of voxels within the ROI mask



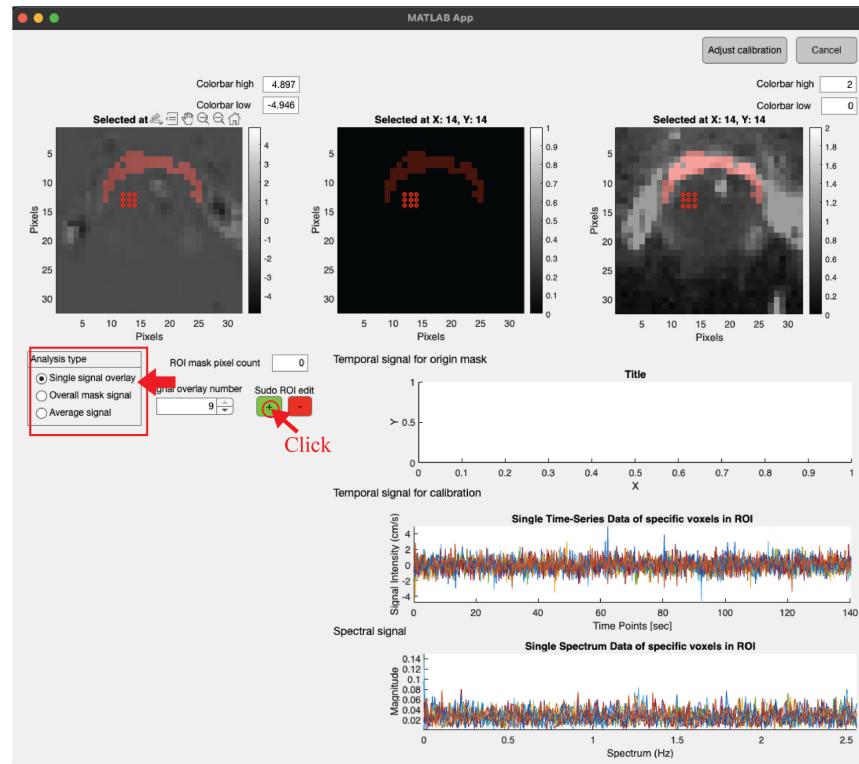
7. Select calibration method; 1) within b.box 2) whole PCMRI phase image area  
 (Note: Please choose only one method for your analysis)



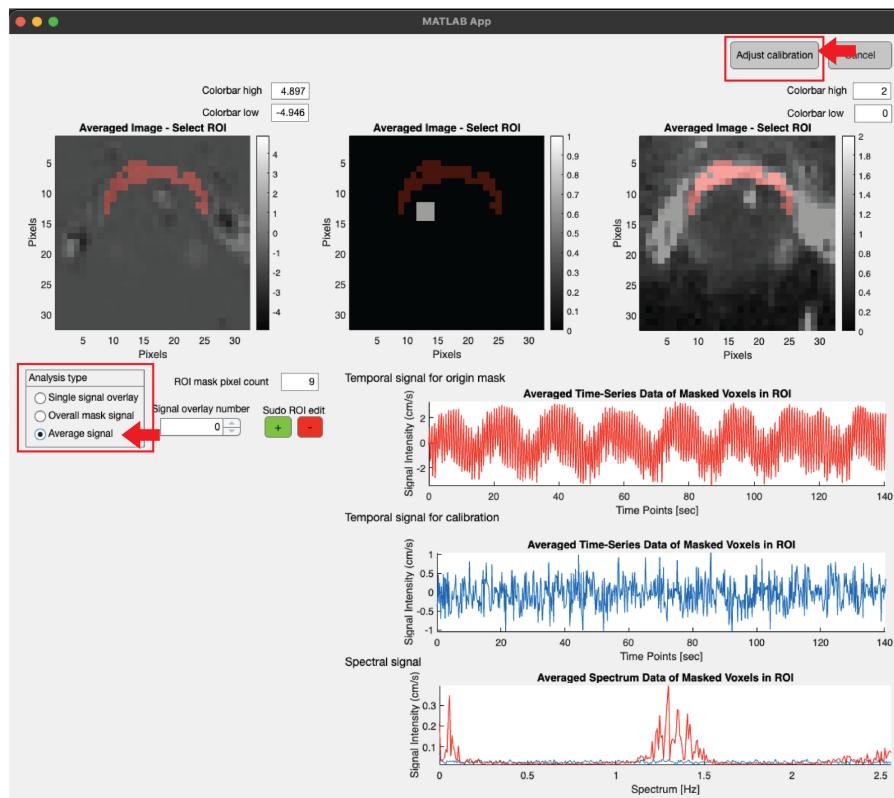
8. For ManualSelection\_withinROI, you can see three axes, which represent each Average image, ROI mask, and STD images. Please click or drag & drop voxels on Average image axes in the left for calibration by referring to the Averaged image and Std image.



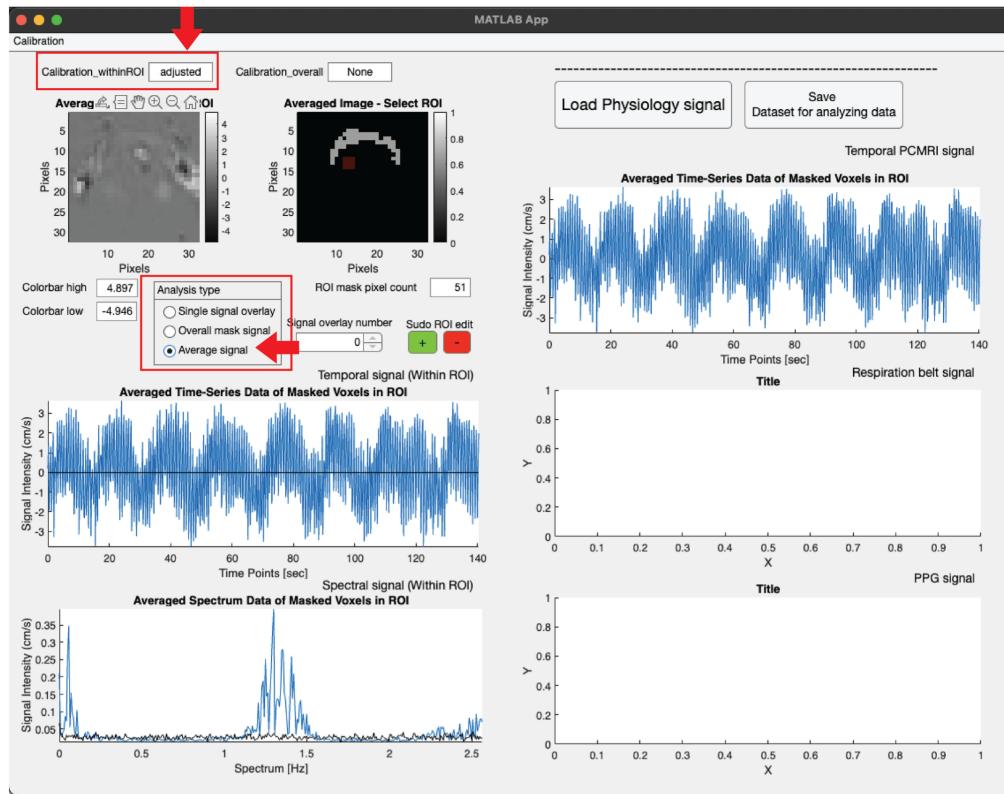
9. If you press the "+" green button while there are selected voxels, you can add those voxels as a calibration mask area.



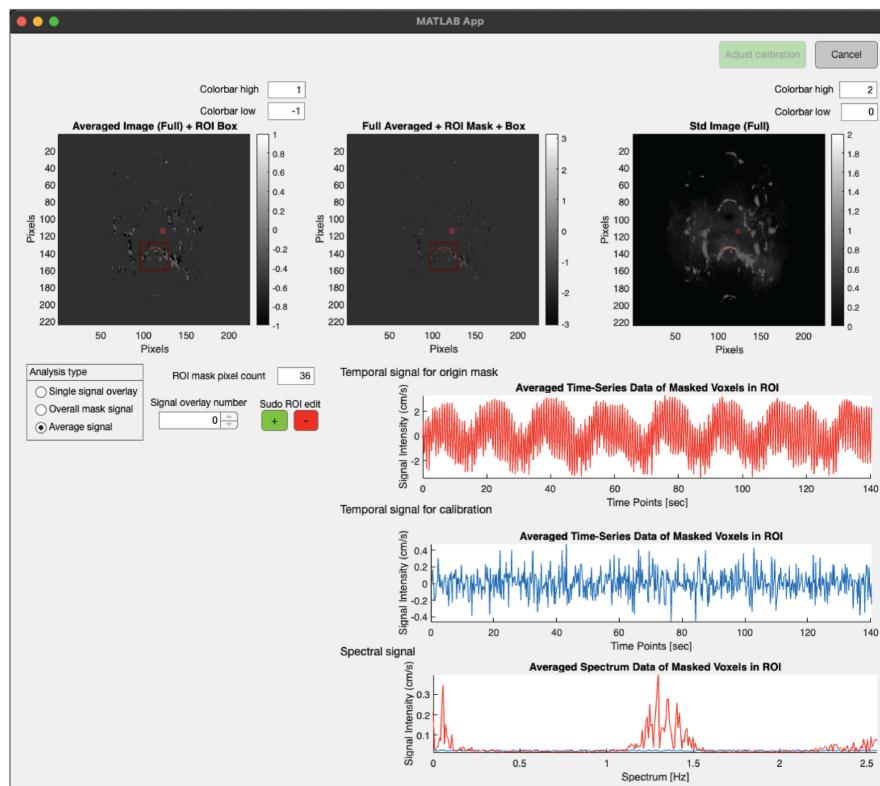
10. Please change the Analysis type to "Average signal" to press the "Adjust calibration" button. You can see the red temporal signal within the ROI masks and the blue temporal signal within the calibration mask at the first and second axes. You can also see both spectrum signals in the bottom axes.



11. If calibration was adjusted well, you can see "adjusted" in the box where you set it. After that, please change the Analysis type to "Average signal" again. You can see a black signal close to zero in the temporal signal axes, which means the signal in the calibration mask was calibrated, and the overall CSF signal was calibrated.

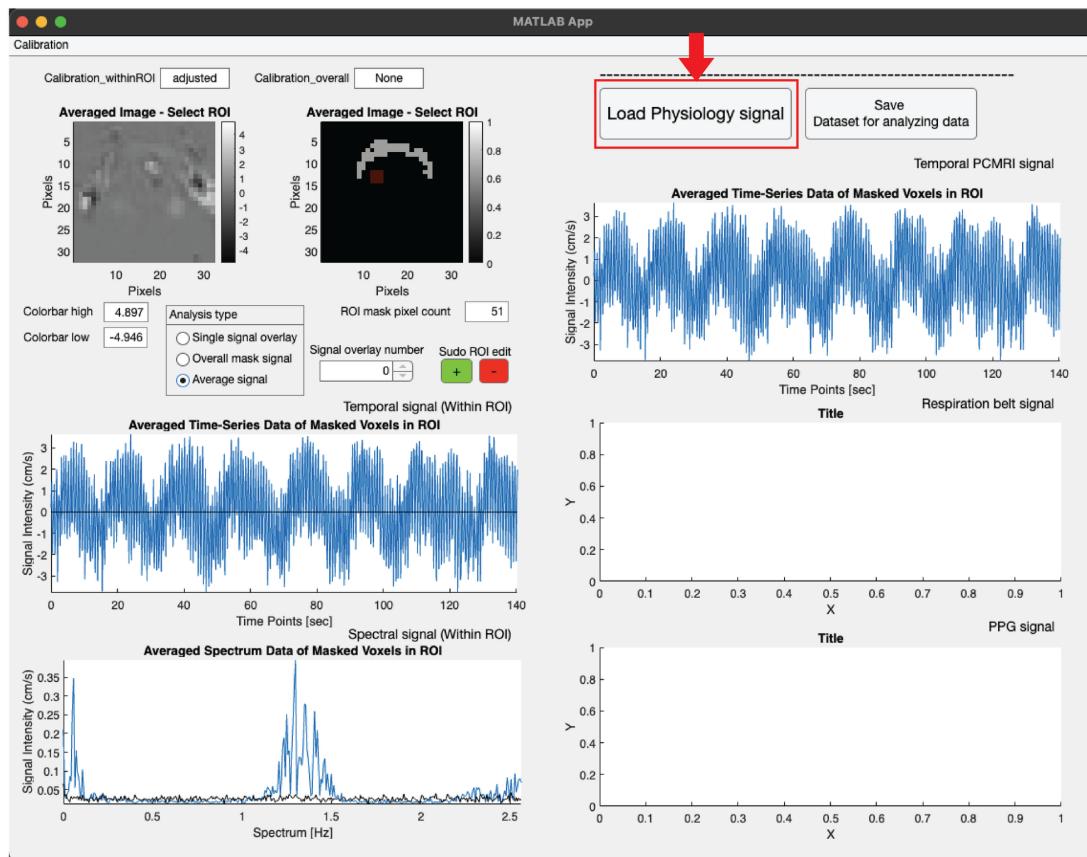


12. For ManualSelection\_overall\_area, the procedure is the same as the previous one.



## 5.6. Load the physiology signal and save the dataset for analyzing it

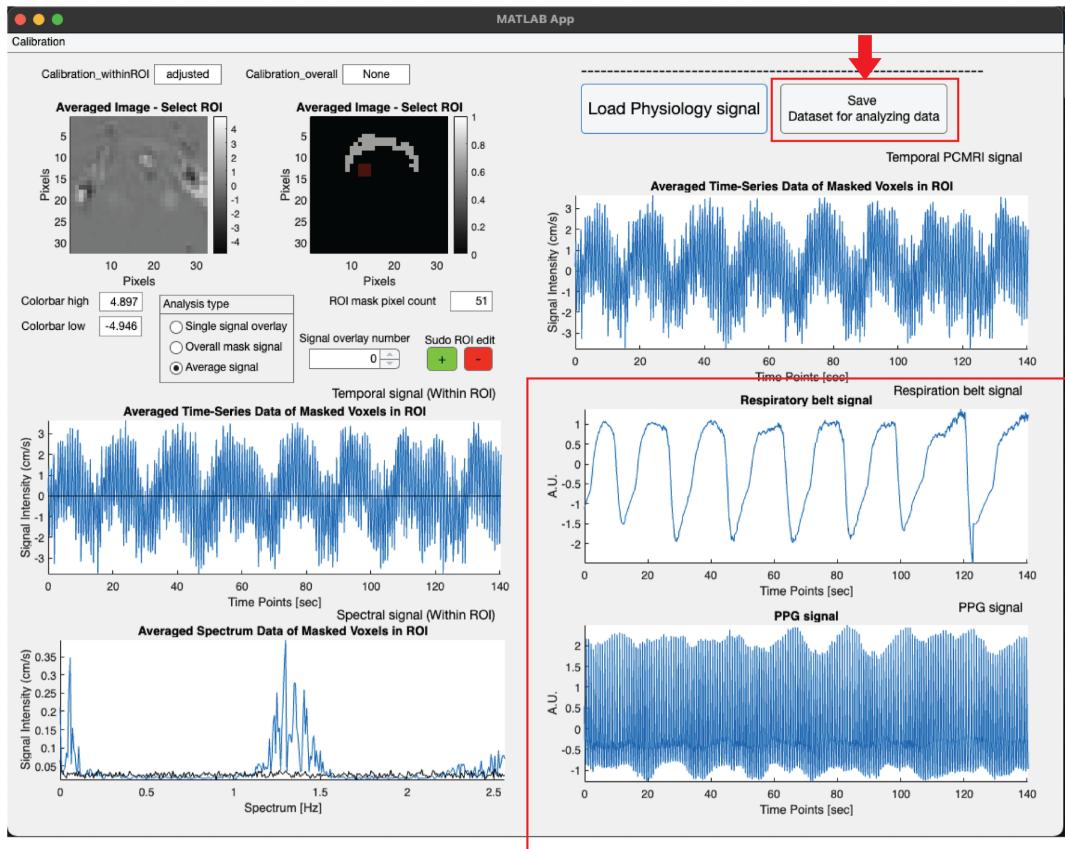
1. If you finish processing until calibration, please press the "Load Physiology signal" button.



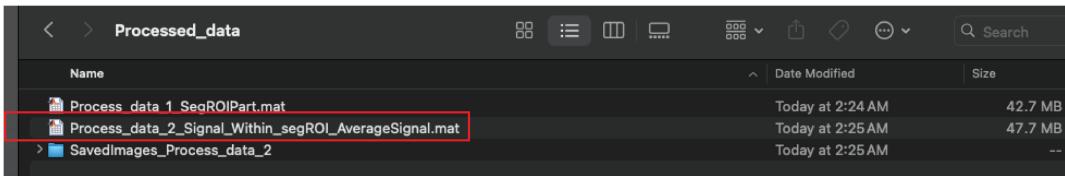
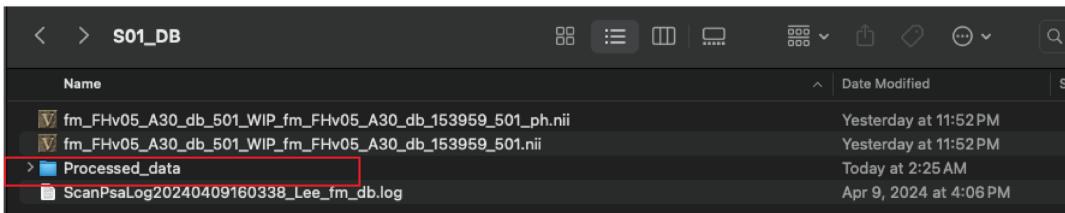
2. Find the physiology data (.log) and press “Okay” button.



3. Please check if the respiratory signal and the PPG signal show well on those axes. After that, please press the "Save dataset for analyzing data" button.

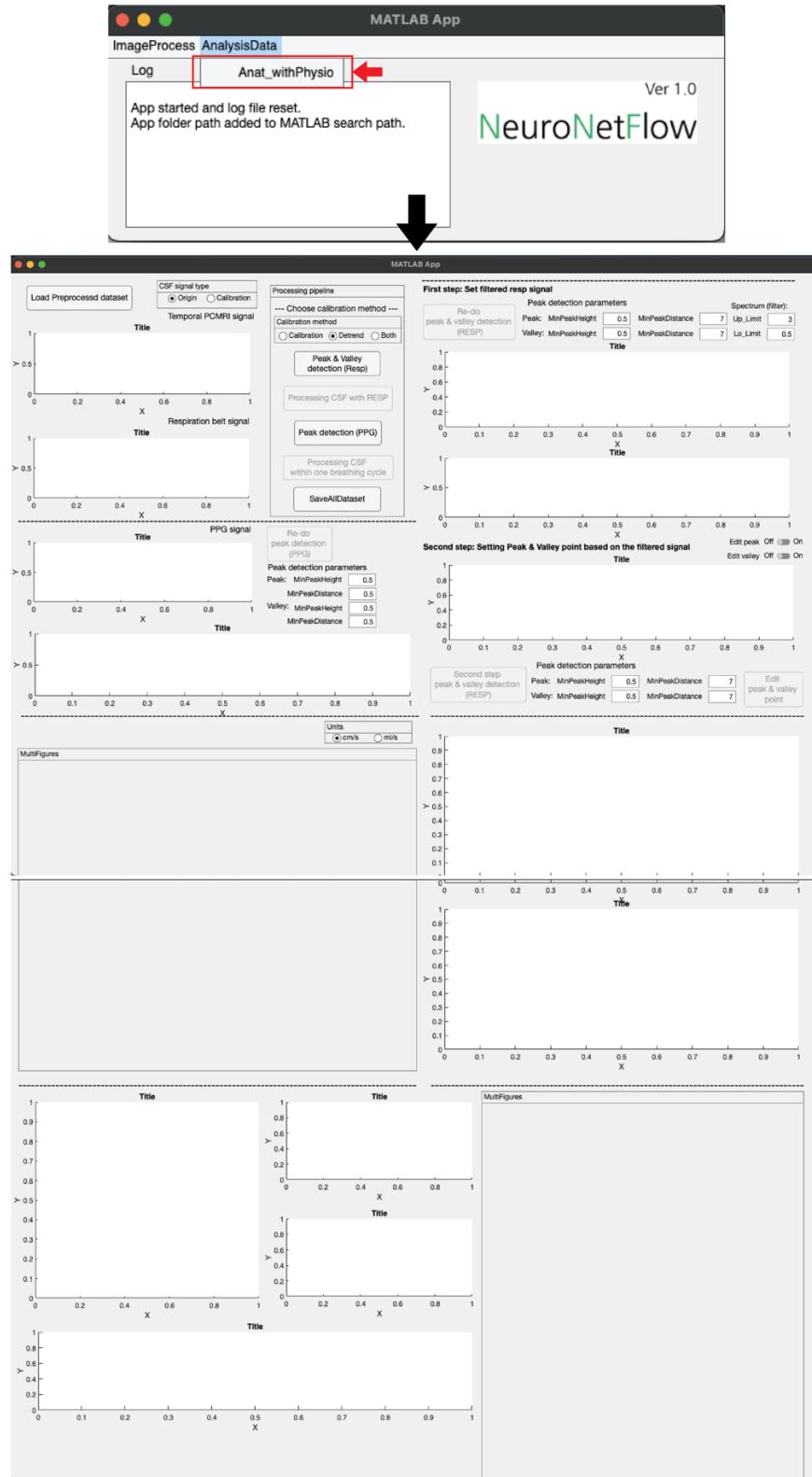


4. Please check if the directory "Processed\_data" has been created in the directory where the data you selected is located. Also, please check if "Process\_data\_2\_Signal\_Within\_segROI\_AverageSignal.mat" has been created in the "Processed\_data", because this data is the input data for analyzing the signal.

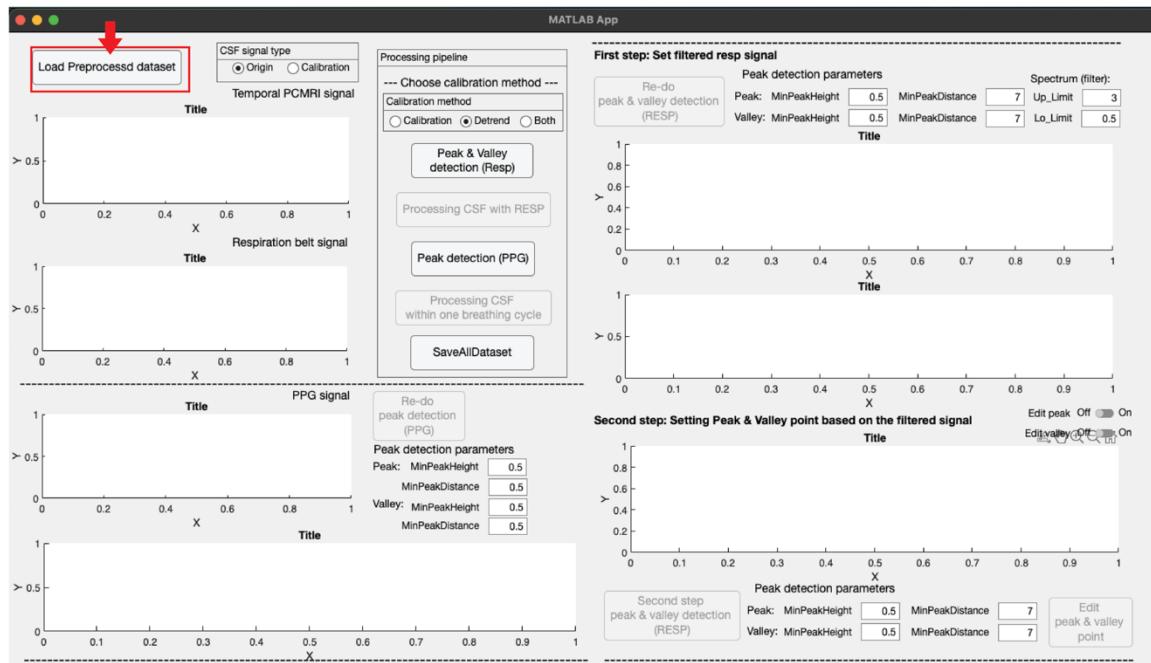


## 5.7. Load preprocessed data for analyzing CSF signal with aligned physiological data

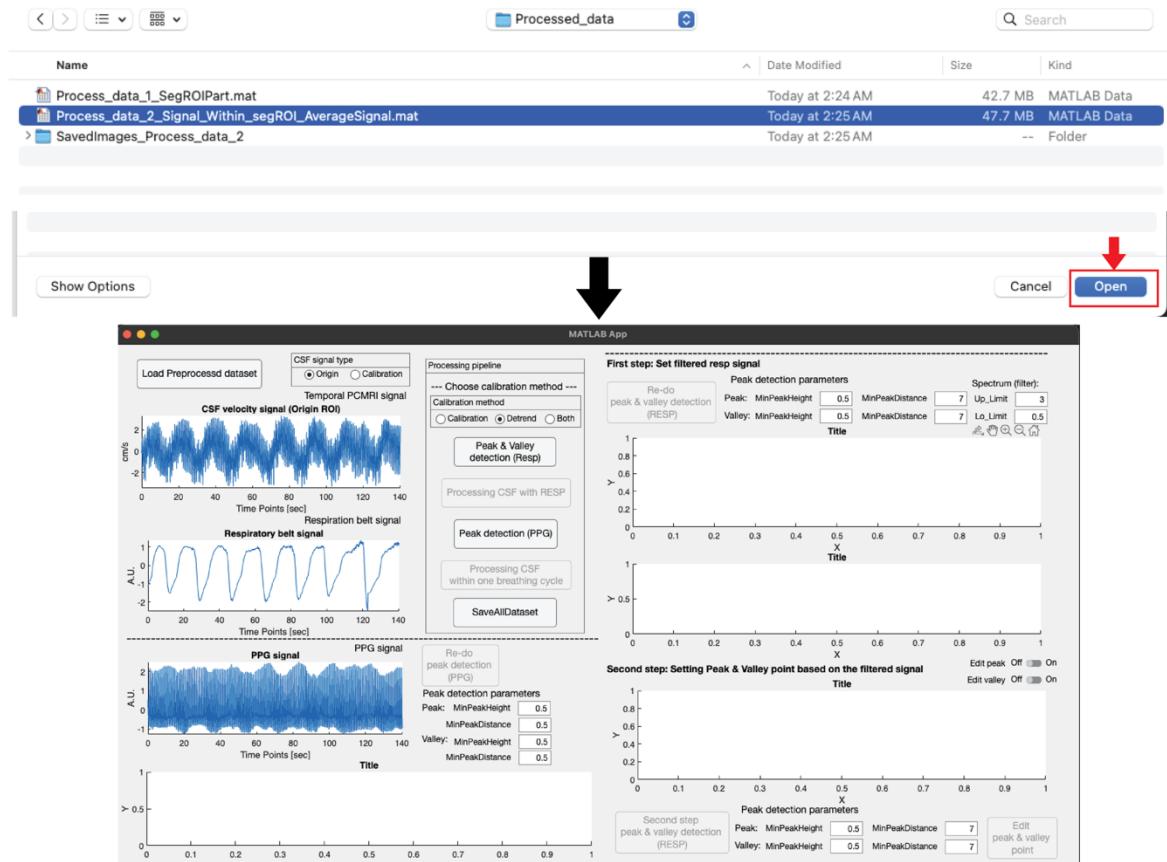
1. Press the “Anat\_withPhysio” button to run the analysis GUI environment.



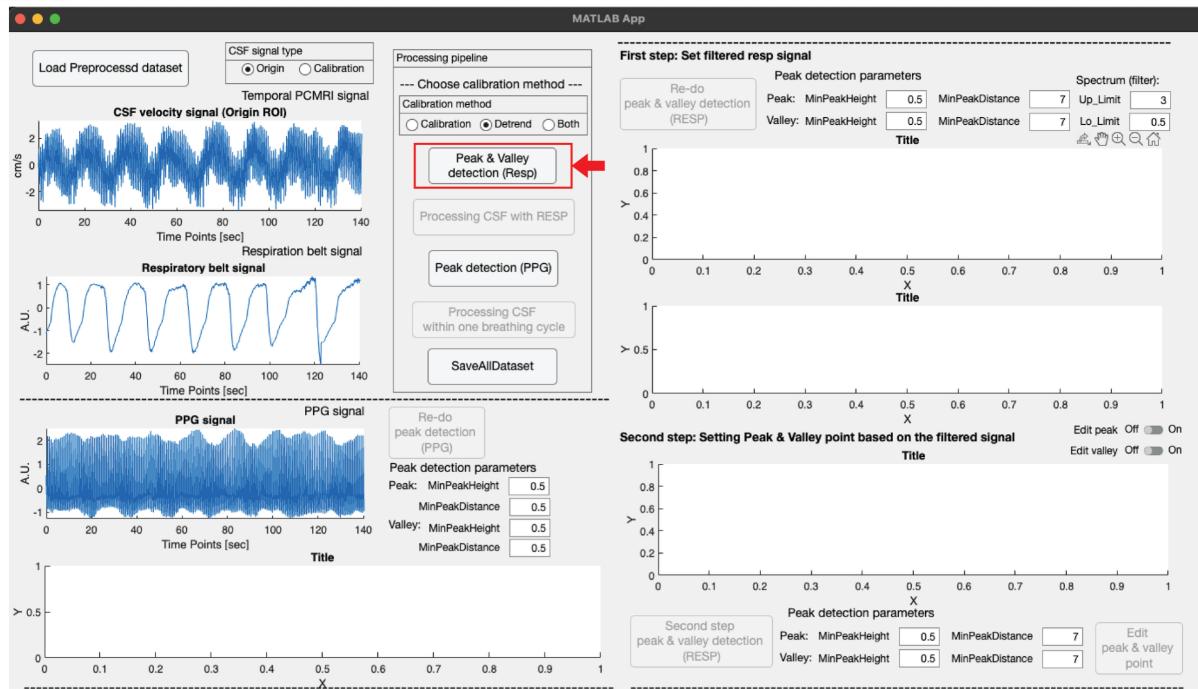
2. Press the “Anat\_withPhysio” button to run the analysis GUI environment.



3. Find and select the preprocessed data “Process\_data\_2\_Signal\_Within\_segROI\_AverageSignal.mat”.  
(Note: Please choose the “Process\_data\_2” not the “Process\_data\_1”)

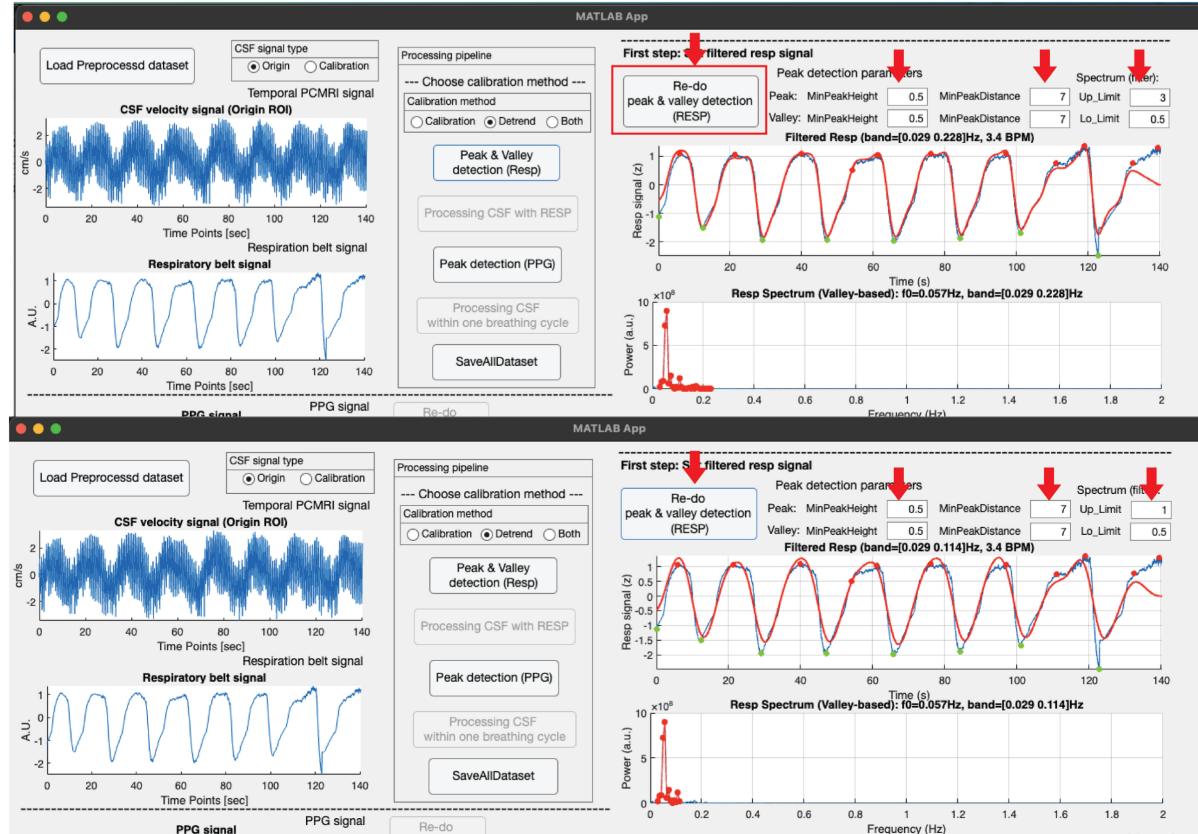


4. Set the CSF signal type and choose the data type you want to analyze, and press the “Peak & Valley detection (Resp)” button.

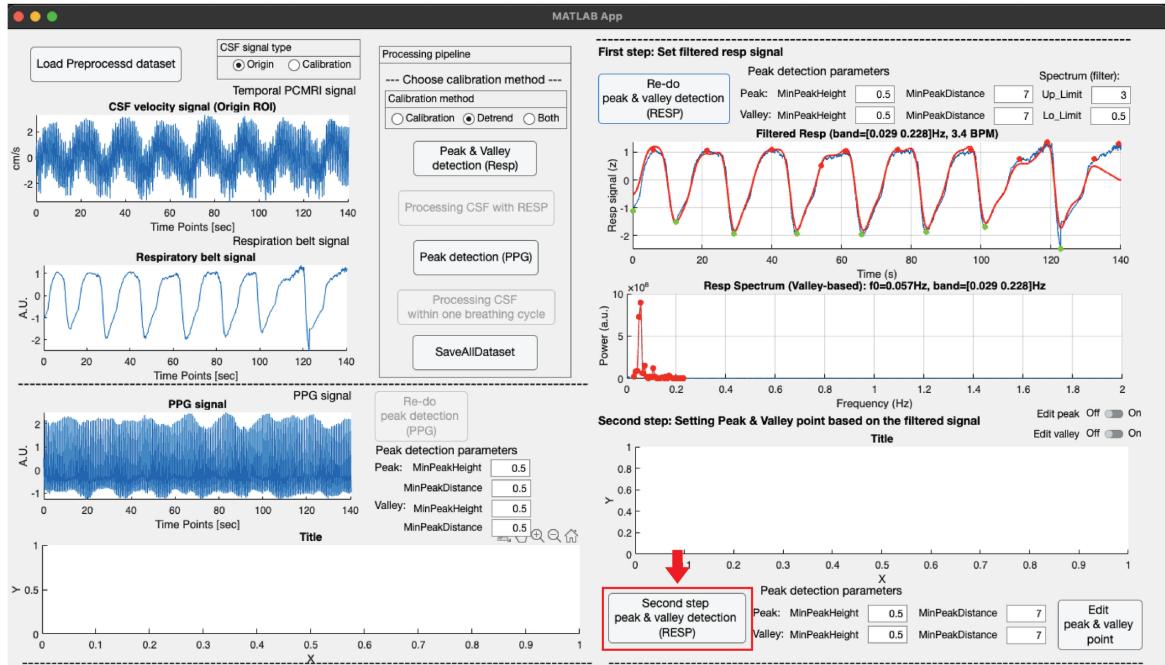


5. First step for respiratory signal analysis with peak & valley detection would be to find a good quality of filtered respiratory signal. This program attempts to fit a specific frequency spectrum point to the respiration rate based on the detected valley point location. Spectrum Up\_Limit, Lo\_Limit define the spectrum bandwidth to filter the raw respiratory signal based on the spectrum point.

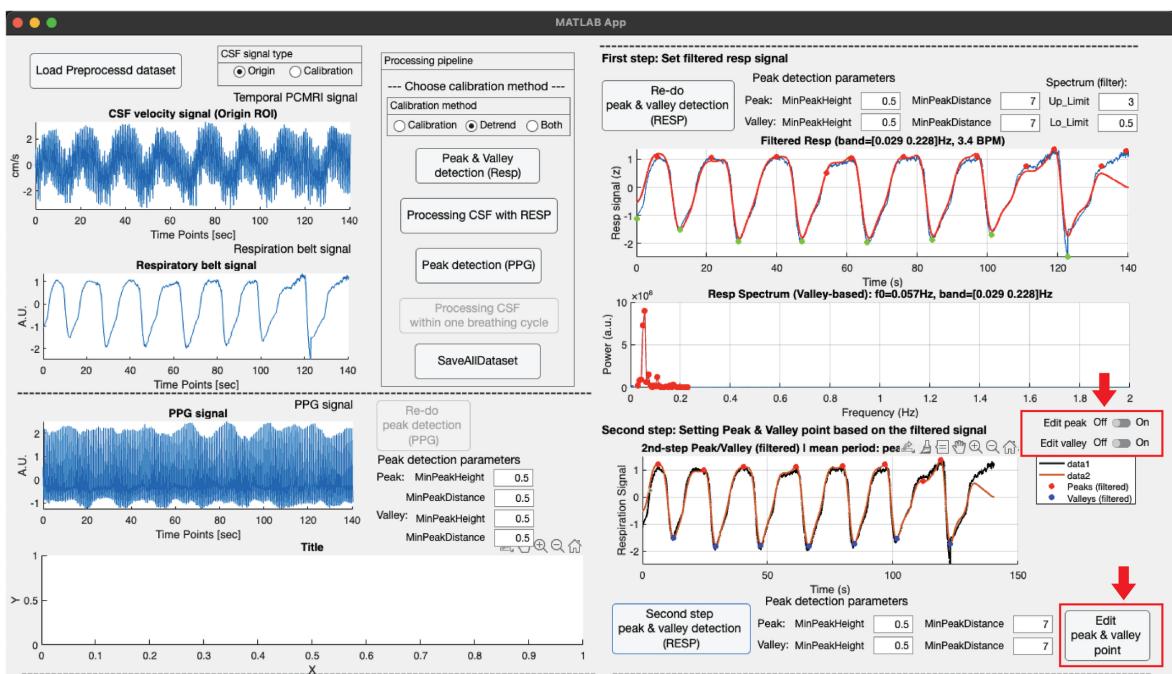
You can redo for peak & valley detection with modified parameters



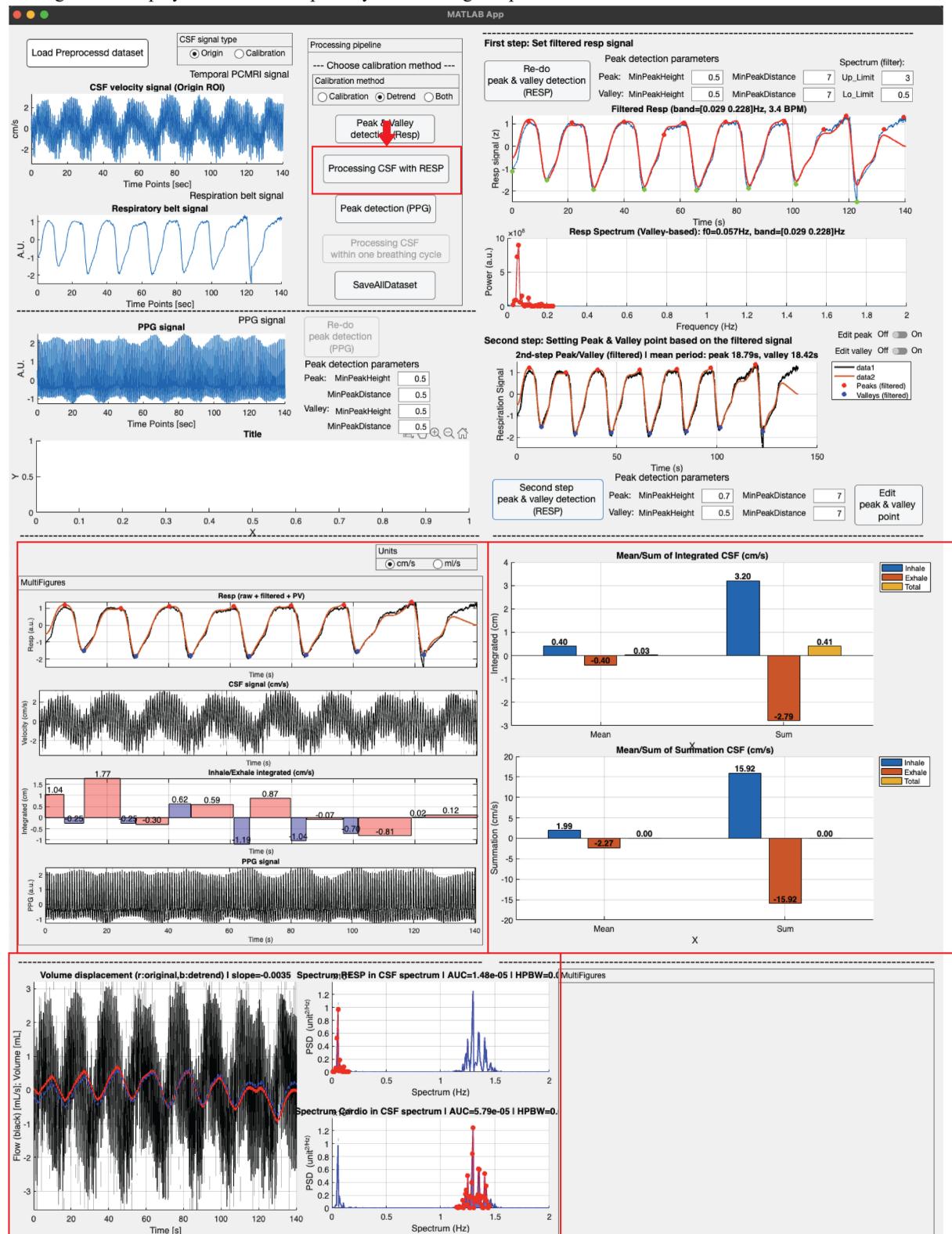
6. If you finish setting the first step of respiration signal processing, please press the "Second step peak & valley detection (RESP)" button. This step is for setting the location of the peak & valley points based on the filtered respiratory signal. Please redo this process with the modified parameter next to the button when the detected peak & valley locations are incorrect.



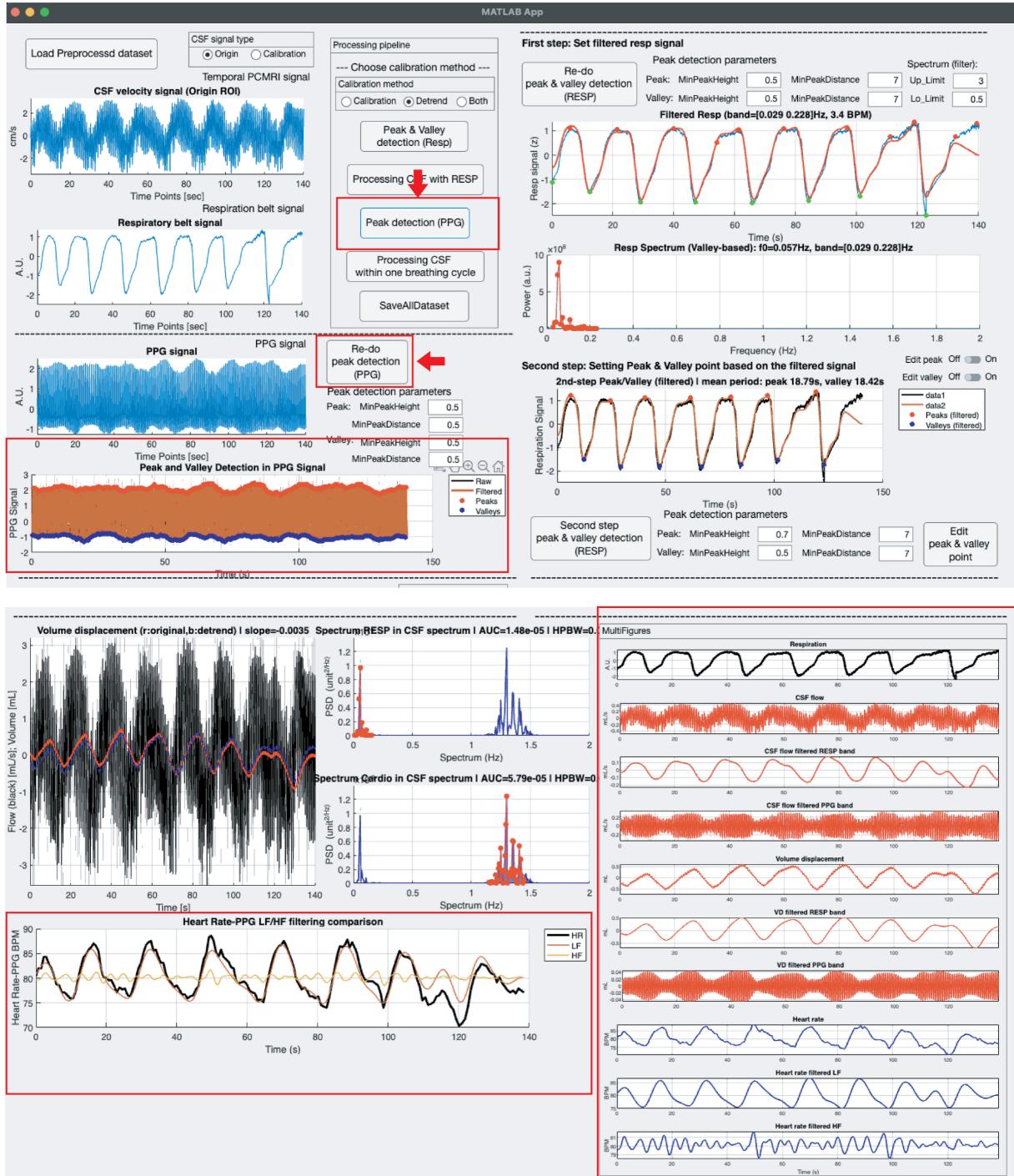
7. This algorithm also provides manual editing of the location of peaks & valleys using the "Edit peak & valley point" button. If you press that button, the Editing function will be activated. After that, you need to swipe the slide object with each "edit peak" or "edit valley" upper the axes to modify the location of the point. Please note that only one slide should be swiped during editing the location of points. If you turn on one function (for example, edit peak or edit valley), you can add a peak point by left-clicking the mouse, and you can remove a peak point by right-clicking the mouse. If you finish editing, please turn off the function and press the "Edit peak & valley point" button again; it will be deactivated.



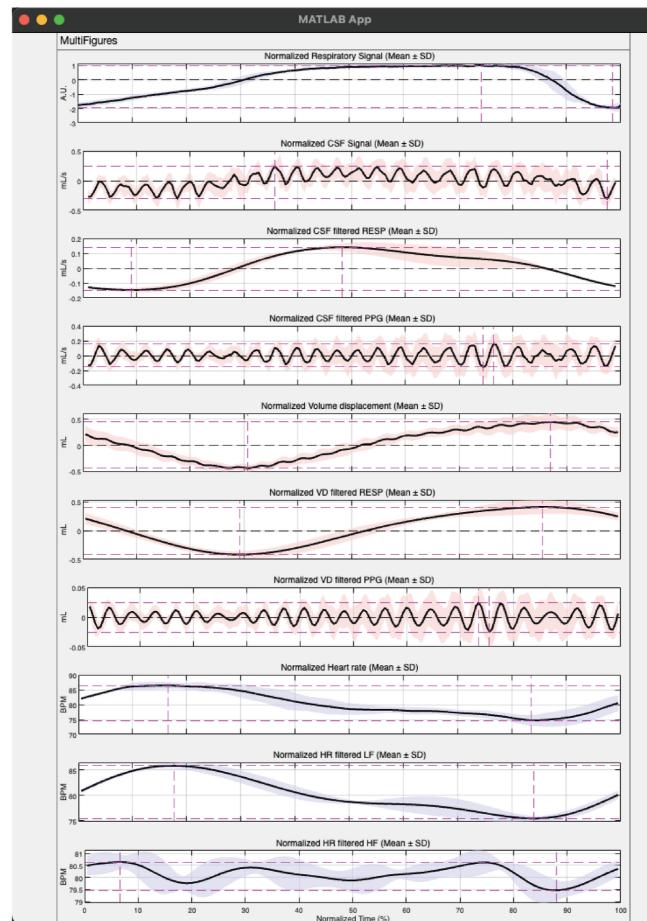
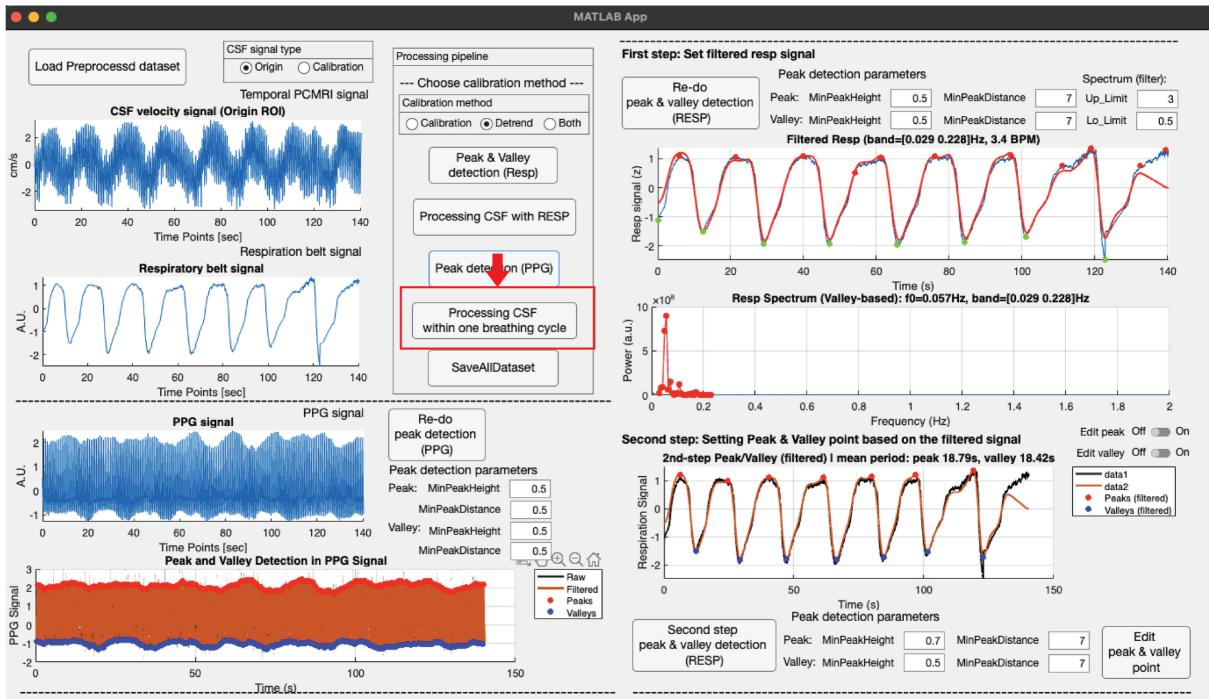
8. If you finish setting the second step of respiration signal processing, please press the "Processing CSF with RESP" button. This step performs CSF signal processing using well-detected peak & valley locations from the previous procedure. You can confirm the visualization results of CSF signal processing, as shown below. The processed CSF signal would be able to change from velocity unit (cm/s) to flow unit (ml/s). Also, the algorithm calculates volume displacement from the flow signal and displays AUC in the respiratory and PPG signal spectra.



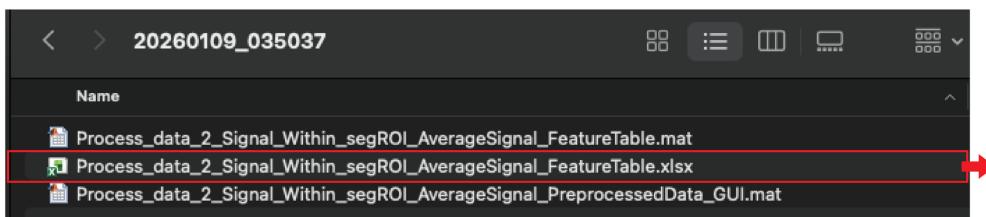
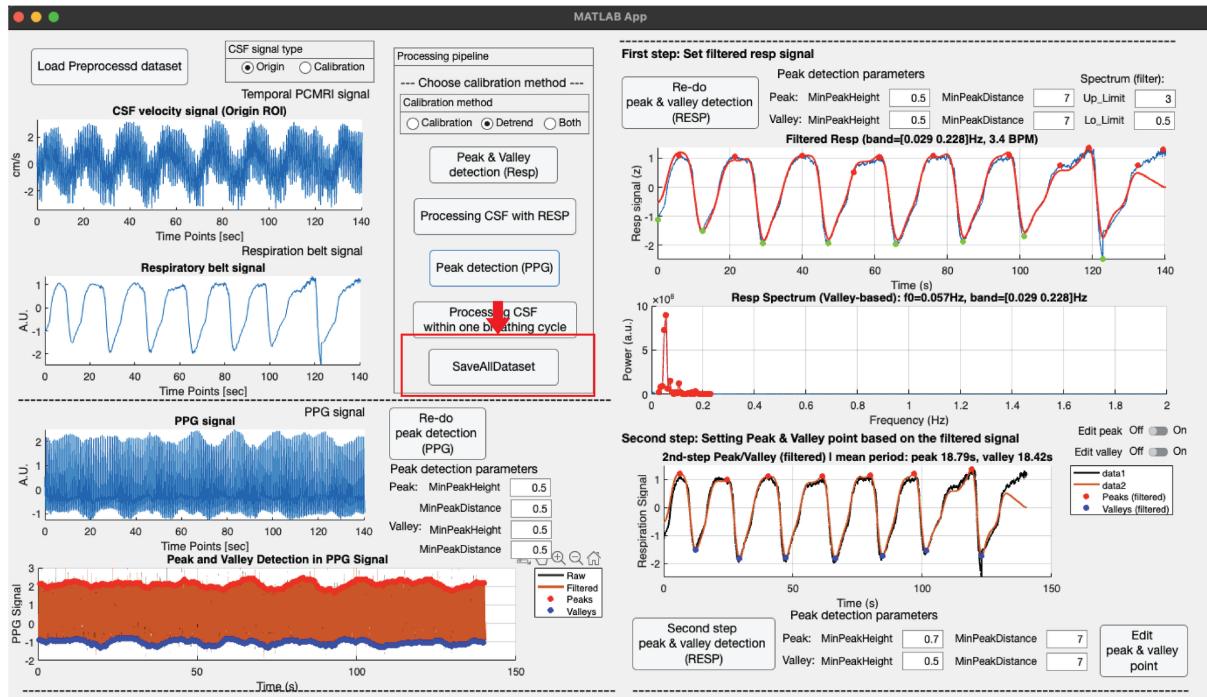
9. After that, you can press the "Peak detection (PPG)" button to obtain the Heart rate signal from PPG processing. Also, redo this process with modified parameters when the detected peak & valley locations are incorrect. If you find good parameters, check the results at the bottom of the window. The heart rate signal from the processed PPG would show the filtered signal in the low-frequency and high-frequency bands, which represent sympathetic and parasympathetic nerve system activity, respectively, in heart rate variability (HRV) analysis. You can also confirm multiple figures, which include all processed signals. These figures show the entire temporal signal trend in a single plot.



10. Please press the "Processing CSF within one breathing cycle" button to conduct further analysis of CSF signals within one breathing cycle. The algorithm automatically segments CSF signals by respiratory cycle and shows the overall averaged signal tendency within one breathing cycle in a separate window.



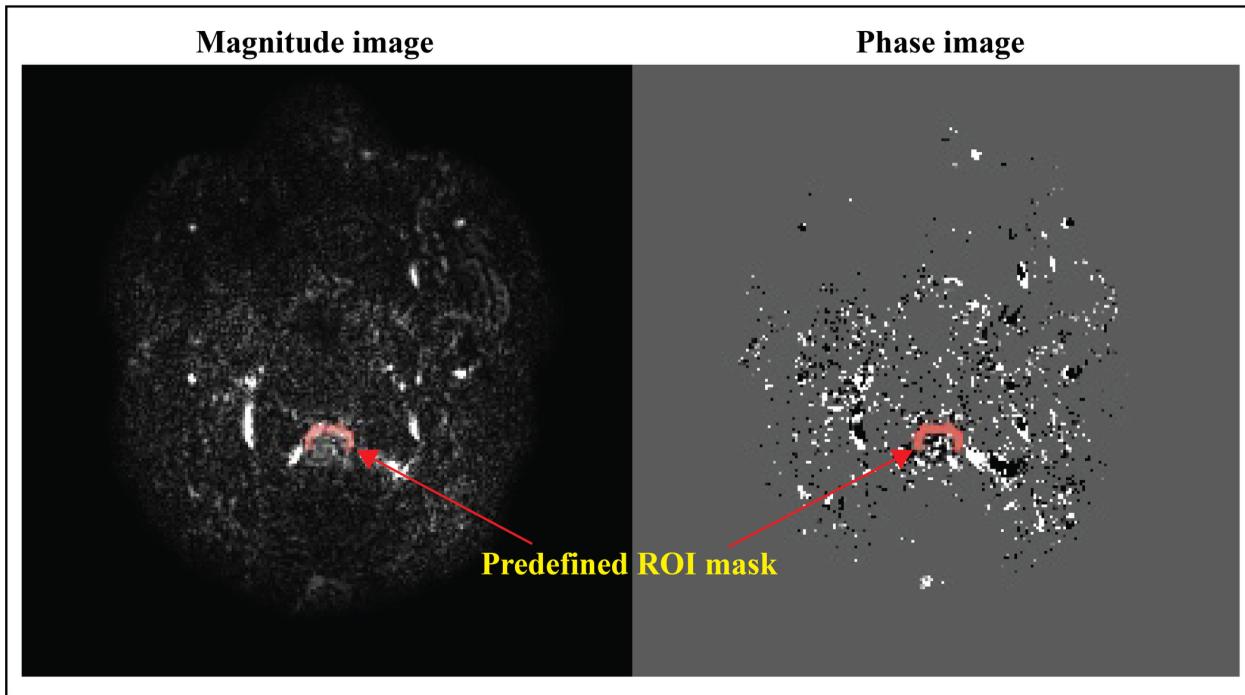
11. Please press the "SaveAllDataset" button when you finish all the pipelines for analyzing CSF signals. Please check if the "Process\_data\_2\_Signal\_Within\_segROI\_AverageSignal\_CSF\_GUI\_Output" has been created in the "Processed\_data". In there, you can see the directory named the time that you created. Please check the three different files in that directory. "PreprocessedData\_GUI.mat" has all processed data. "FeatureTable.xlsx or .mat" has the table structure that predefined feature values and labels. Please analyze all participants you recruited as follows, using this toolbox instruction, and gather all feature tables for those participants. After that, you can conduct a statistical analysis based on those datasets.



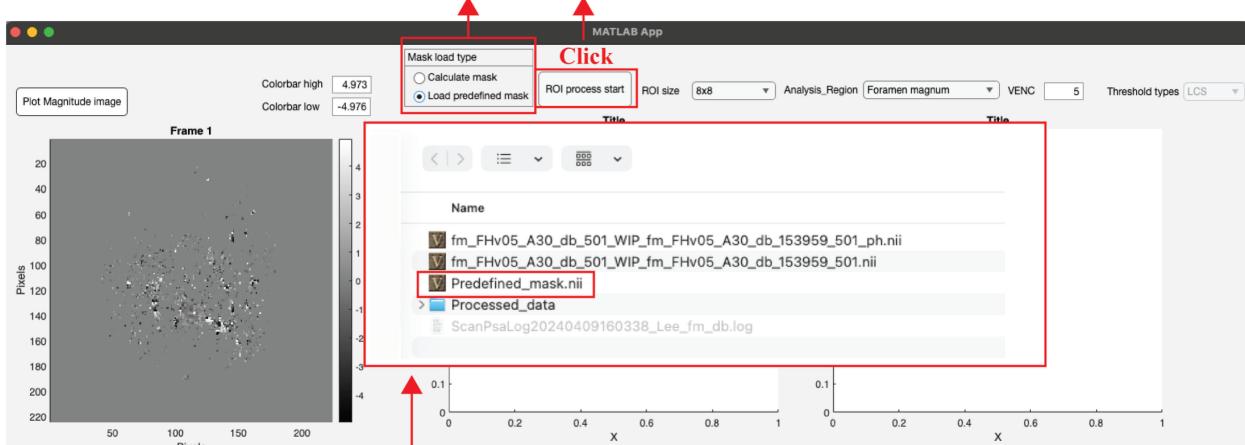
## 6. Additional function

### 6.1. Pre-defined mask

The mask can be defined in advance using external software such as AFNI.



To load a predefined mask, select the “Load predefined mask” radio button and then click “ROI Process Start.”

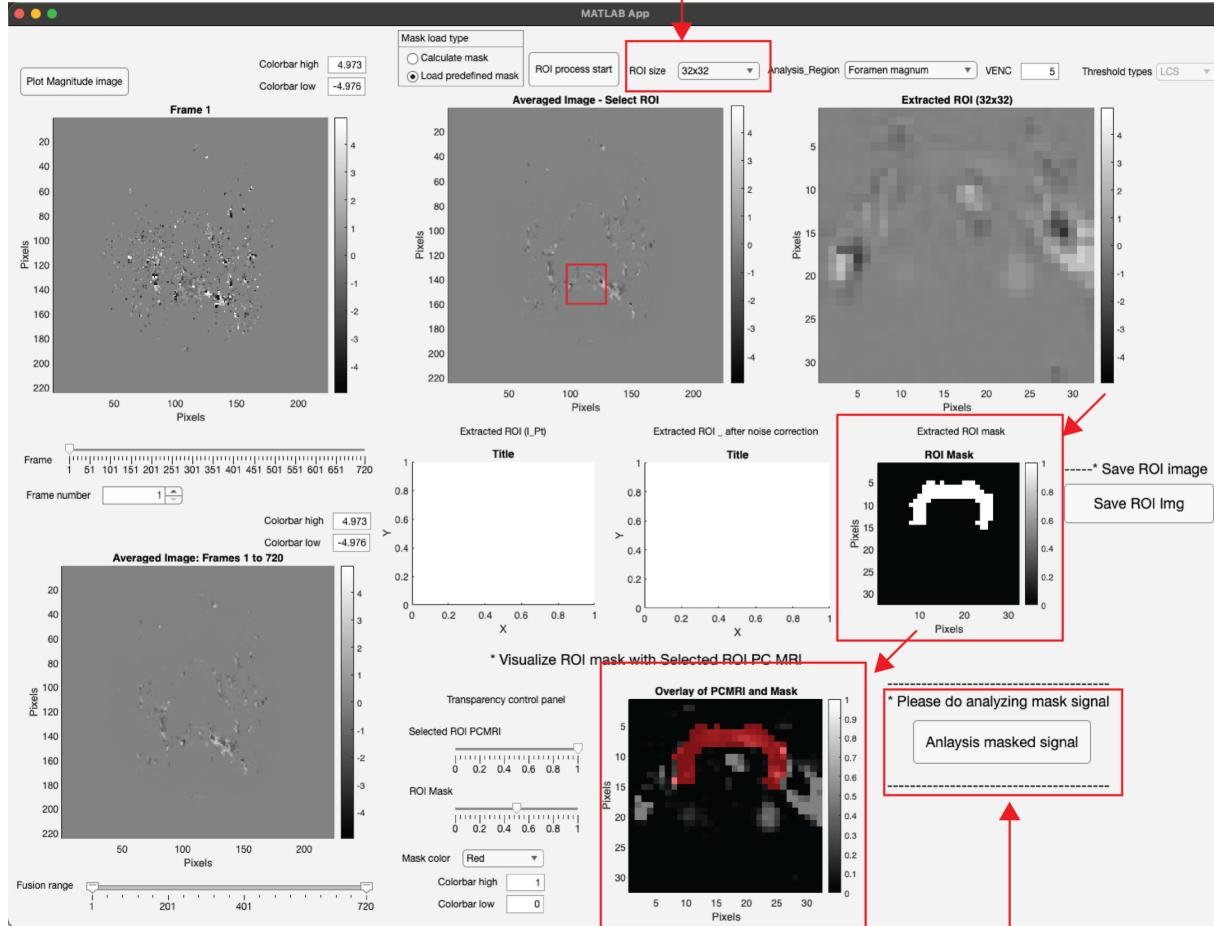


When you click “Start ROI Process,” a file explorer opens, allowing the user to locate and load a predefined mask from the specified path.

**In the Boundary Box options, select the desired box size and click the target location in the Average image.**  
This procedure follows the same interaction used in the Calculate Mask process.

A bounding box is then generated, and the predefined mask is loaded within this region. Because the automatic ROI calculation algorithm is not executed when a predefined mask is used, the ROI mask is loaded directly without additional computation.

Subsequent Calibration and ROI pseudo-editing steps are identical to those performed in the Calculate Mask process.



Once the predefined mask is successfully loaded,  
perform the Calibration and Pseudo ROI mask editing steps,  
load the physiology signals, and save the resulting data.  
This workflow is identical to the Calculate Mask process.

## 7. Software metadata

Nr	Code metadata description	<i>Metadata</i>
C1	Current code version	<i>V1.0</i>
C2	Permanent link to code/repository used for this code version	<a href="https://github.com/SeokbeenLim/NeuroNetFlow.git">https://github.com/SeokbeenLim/NeuroNetFlow.git</a>
C3	Permanent link to reproducible capsule	<i>N/A</i>
C4	Legal code license	<i>MIT License</i>
C5	Code versioning system used	<i>Git</i>
C6	Software code languages, tools and services used	<i>MATLAB</i>
C7	Compilation requirements, operating environments and dependencies	<i>MATLAB R2023b or later; compatible with Windows 10/11 and macOS.</i> Note: This version is specifically optimized for and compatible only with phase-contrast MRI data acquired from Philips scanners.
C8	If available, link to developer documentation/manual	
C9	Support email for questions	<a href="mailto:Lim.Seokbeen@mayo.edu">Lim.Seokbeen@mayo.edu</a> , <a href="mailto:min.paul@mayo.edu">min.paul@mayo.edu</a>

## Reference

1. Lun, MP, Monuki, ES, Lehtinen, MK. Development and functions of the choroid plexus–cerebrospinal fluid system. *Nat Rev Neurosci* 2015;16(8):445-457.
2. Cai, Y, Zhang, Y, Leng, S, et al. The relationship between inflammation, impaired glymphatic system, and neurodegenerative disorders: A vicious cycle. *Neurobiology of Disease* 2024;192:106426.
3. Hladky, SB, Barrand, MA. Regulation of brain fluid volumes and pressures: basic principles, intracranial hypertension, ventriculomegaly and hydrocephalus. *Fluids and Barriers of the CNS* 2024;21(1):57.
4. Yoon, J-H, Jin, H, Kim, HJ, et al. Nasopharyngeal lymphatic plexus is a hub for cerebrospinal fluid drainage. *Nature* 2024;625(7996):768-777.
5. Alberto, D, Alessandro, M. Respiration: A New Mechanism for CSF Circulation? *The Journal of Neuroscience* 2017;37(30):7076.
6. Lim, S, Cogswell, PM, Jacobson, DN, et al. Human cerebrospinal fluid net flow enhanced by respiration during the awake state. *Nature Communications* 2025.
7. Aktas, G, Kollmeier, JM, Joseph, AA, et al. Spinal CSF flow in response to forced thoracic and abdominal respiration. *Fluids and Barriers of the CNS* 2019;16(1):10.
8. Liu, P, Monnier, H, Owashi, K, et al. The Effects of Free Breathing on Cerebral Venous Flow: A Real-Time Phase Contrast MRI Study in Healthy Adults. *The Journal of Neuroscience* 2024;44(3):e0965232023.
9. Liu, P, Owashi, K, Monnier, H, et al. Synchronous assessment of CSF and cerebral arteriovenous flow interactions across ultra-low, low, respiratory, and cardiac frequencies using real-time phase-contrast MRI. *Neuroimage* 2025;321:121490.
10. Liu, P, Owashi, K, Monnier, H, et al. Cardiac and respiratory activities induce temporal changes in cerebral blood volume, balanced by a mirror CSF volume displacement in the spinal canal. *Neuroimage* 2025;305:120988.
11. Sahoo, P, Kollmeier, JM, Wenkel, N, et al. CSF and venous blood flow from childhood to adulthood studied by real-time phase-contrast MRI. *Child's Nervous System* 2024;40(5):1377-1388.
12. Steffi, D-K, Arun, AJ, Klaus-Dietmar, M, et al. Inspiration Is the Major Regulator of Human CSF Flow. *The Journal of Neuroscience* 2015;35(6):2485.
13. Yildiz, S, Grinstead, J, Hildebrand, A, et al. Immediate impact of yogic breathing on pulsatile cerebrospinal fluid dynamics. *Sci Rep-Uk* 2022;12(1):10894.
14. Liu, P. Acquisition and processing of phase contrast Magnetic Resonance Imaging for the quantification of cerebral blood and cerebrospinal fluid flow under respiratory influence

Acquisition et traitement de l'imagerie par Résonance Magnétique en contraste de phase pour la quantification des écoulements cérébraux du sang et du Liquide Cérébro-Spinal sous influence respiratoire. Université de Picardie Jules Verne; 2021.

15. Liu, P, Fall, S, Baledent, O. Flow 2.0 -a flexible, scalable, cross-platform post-processing software for realtime phase contrast sequences. 2022.