

---

# Processing Signal Viewer

## User Documentation

---

Authors: Quintess Barnhoorn, Loes Erven, Kayla Gericke, Mignon Hagemeijer, Nick van der Linden, Peter van Olmen, Sander van 't Westeinde

<b>Getting Started</b>	<b>2</b>
Running the program	2
Running on Windows	2
Running with simulated data	2
Running with an EEG cap	2
Running on Linux	3
Running with simulated data	3
Running with an EEG cap	3
Android	4
Adding the buffer to Android	4
Running on Android	4
<b>User Interface</b>	<b>5</b>
Sidebar	6
Pre-processing	6
Spatial Filter	6
Adapt Filter	7
Spectral Filter	7
Tabs	7
The Time Tab	7
The Frequency Tab	8
The 50 Hz Tab	8
<b>Common Errors</b>	<b>9</b>
BufferClient is not found	9
Android keyboard is unable to type	9
Slow loading data	9

# Getting Started

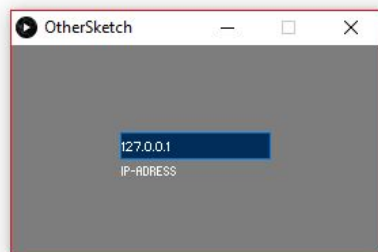
## Running the program

### Running on Windows

When you want to use the 32-bit version, which is much smaller than the 64-bit version, you need to have Java 8.

### Running with simulated data

1. Start the proxy by running the debug\_quickstart.bat file
2. Open the EEGSignalViewer by running EEGSignalViewer.exe
3. There will open a popup window with IP-Address as shown in the picture below. Here, Enter the IP address of the device on which the proxy is running. If you are running it on the same device this should be the default IP address that has already been entered.



4. Click on enter and the program should be running from this point. You can now start selecting preprocessing and filter options and switch domains by clicking or tapping on the boxes.

### Running with an EEG cap

- Start a buffer by running dataAcq/startJavaBuffer.bat
- Start the appropriate acquisition driver for your device:
  - \* TMSi Mobita: dataAcq/startMobita.bat
  - \* Emotiv Epoc: dataAcq/startEmotiv.bat
  - \* Biosemi Active 2: dataAcq/startBiosemi.bat
  - \* Interaxon Muse: dataAcq/startMuse.bat
- Open the EEGSignalViewer by running the EEGSignalViewer.exe

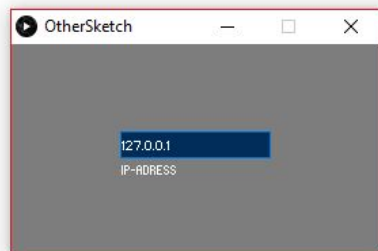
- Enter the IP address of the device on which the proxy is running. If you are running it on the same device this should be the default IP address that has already entered.

## Running on Linux

Linux has four versions; which version you want to use depends on your processor and operating system. There are ARM and x86 versions. Each of these has a 32-bit and 64-bit version. In most cases you are fine running the 64-bit variant.

### Running with simulated data

1. Start the proxy by running the `debug_quickstart.sh`. If it asks how to run it choose run in terminal.
2. Open the EEGSignalViewer by running `./EEGSignalViewer`. From a file manager a simple double click should suffice, but if running from a terminal do not forget the `./` part, otherwise it will not run.
3. A popup window asking for an IP address will open as shown in the picture below. Here, Enter the IP address of the device on which the bufferClient is running. If you are running it on the same device this should be the default IP address that has already been entered.



4. Click on enter and the program should start running. You can now start selecting preprocessing and filter options and switch domains by clicking or tabbing on the boxes.

### Running with an EEG cap

- Start a buffer by running `dataAcq/startJavaBuffer.sh`
- start appropriate acquisition driver for your device:
  - \* TMSi Mobita: `dataAcq/startMobita.sh`
  - \* Emotiv Epoc: `dataAcq/startEmotiv.sh`
  - \* Biosemi Active 2: `dataAcq/startBiosemi.sh`
  - \* Interaxon Muse: `dataAcq/startMuse.sh`

- Open the EEGSignalViewer by running `./EEGSignalViewer`. From a file manager a simple double click should suffice, but if running from a terminal do not forget the `./` part, otherwise it will not run.
- Enter the IP address of the device on which the proxy is running. If you are running it on the same device this could in theory be the default IP address that is already entered.

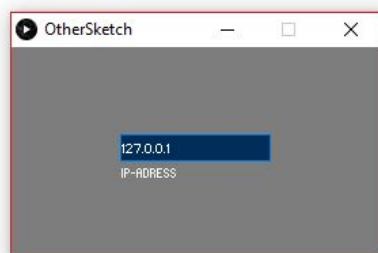
## Android

### Adding the buffer to Android

1. Install the Android APK app on your device (preferably somewhere you can easily find it later)
2. On your device, open the APK, it will download immediately
3. Go to the app, and select both server and Clients
4. Select *SignalProxyThreadke*

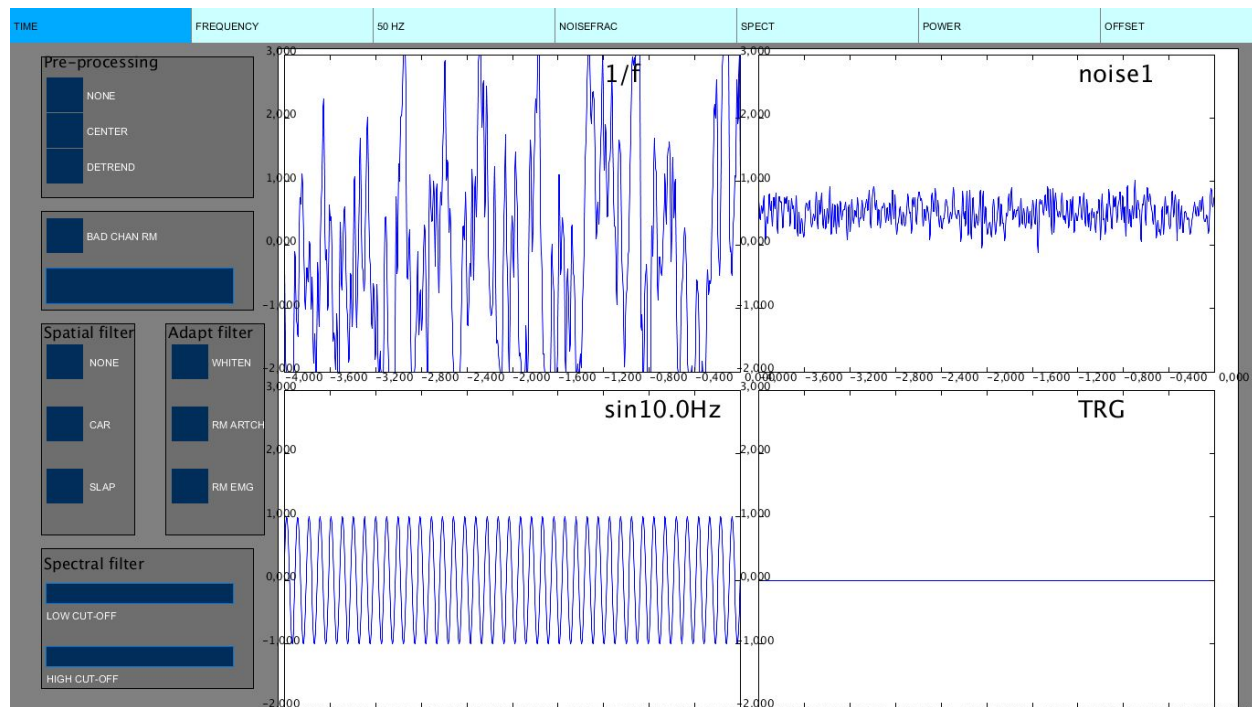
### Running on Android

1. Make sure you have the buffer APK on your phone. This can be done by following the instruction in the adding the buffer to android section.
2. Start the bufferClient by opening the bufferClient app
3. Open the EEGSignalViewer app
4. There will open a popup window with IP-Address as shown in the picture below. Here, Enter the IP address of where the bufferClient is running. If you are running it on the same device this could in theory be the default IP address that is already entered. It is possible to run the buffer on a different device from where you are running the viewer. In this case enter the IP address of that device. When doing this make sure you are on the same network.



5. Click on enter and the program should be running from this point. You can now start selecting preprocessing and filter options and switch domains by clicking or tabbing on the boxes.

# User Interface

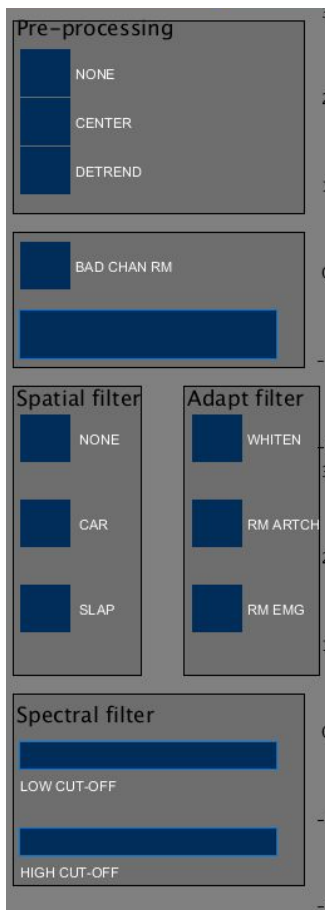


The picture above shows in the white part the graphs, displaying the data the signal viewer receives from the buffer. The number of graphs depends on the number of channels the buffer receives. In this display 4 different graphs are shown. The names in the upper right corner are the names of the channels (1/f, noise1, sin10.0Hz, TRG). If no names are known, these names are just numbered from 1 to n, where n is the number of channels.

On the right gray side different selectable filter and preprocessing options are displayed. The square boxes are selectable boxes, that change colour if selected, and the rectangles are text fields.

The light blue bars are the other possible viewtypes. Click or tap there to switch between two different domains, for instance between time and frequency. The slightly darker blue indicates which view type has been selected. By default, the data is displayed in the time domain.

## Sidebar



The following functionalities have been implemented in the current version:

- Pre-processing :
  - None
  - Center
- Spatial filter:
  - Car
- Spectral filter:
  - Low cut-off
  - High cut-off

The rest of the buttons and functionalities have not been implemented yet.

## Pre-processing

- None: The data is not preprocessed, and displayed as-is.
- Center: Centers the data around 0.
- Detrend: Not implemented in this version.
- Bad Chan Rm: Not implemented in this version.

## Spatial Filter

- None: No filter is used.
- CAR: Common average referencing is used, meaning that the common average of electrodes is subtracted from the input signal of each electrode.
- SLAP: not implemented in this version.

## Adapt Filter

- Whiten: not implemented in this version
- Rm ArtCh: Not implemented in this version
- Rm EMG: Not implemented in this version

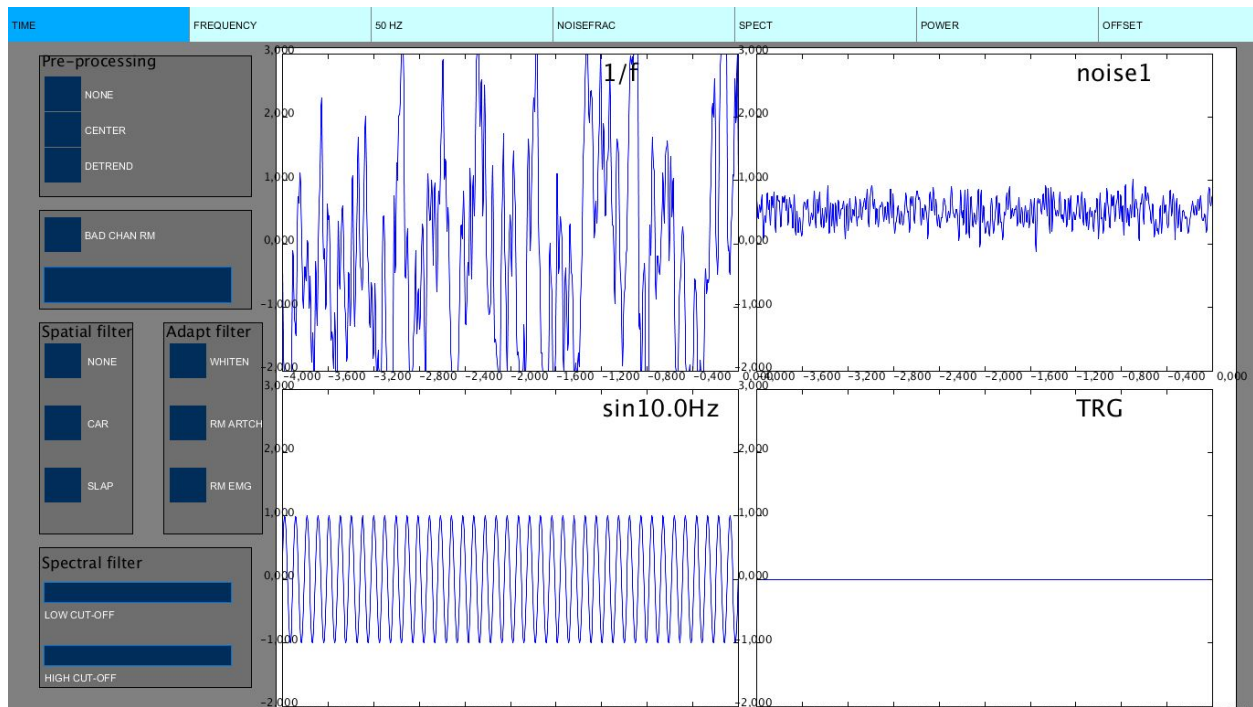
## Spectral Filter

- Low cut-off: This is a textfield where a number can be entered that will become the lower bound for the frequencies. All frequencies below this lower bound will be filtered out.
- High cut-off: This is a textfield where a number can be entered that will become the upper bound for the frequencies. All frequencies above this upper bound will be filtered out.

## Tabs

Currently, the Time and Frequency domains have been fully implemented. The 50 Hz tab has been partially implemented, but will not display data. The NoiseFrec, Spectrogram, Power and Offset domains have not been implemented in this version, and will show nonsensical graphs if selected.

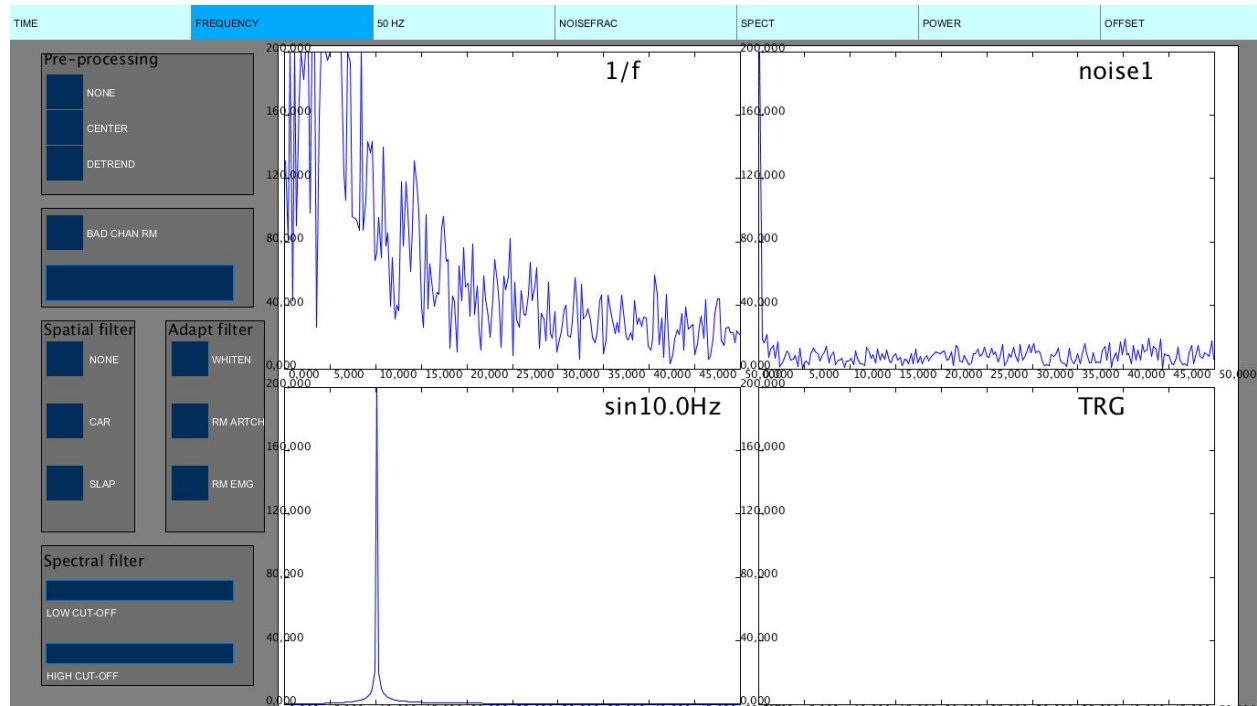
## The Time Tab



When this tab is selected the data is shown in the time domain. On the x-axis the time from -4 to 0 seconds are shown. On the y-axis the power of the signal in  $\mu\text{V}$  is shown.

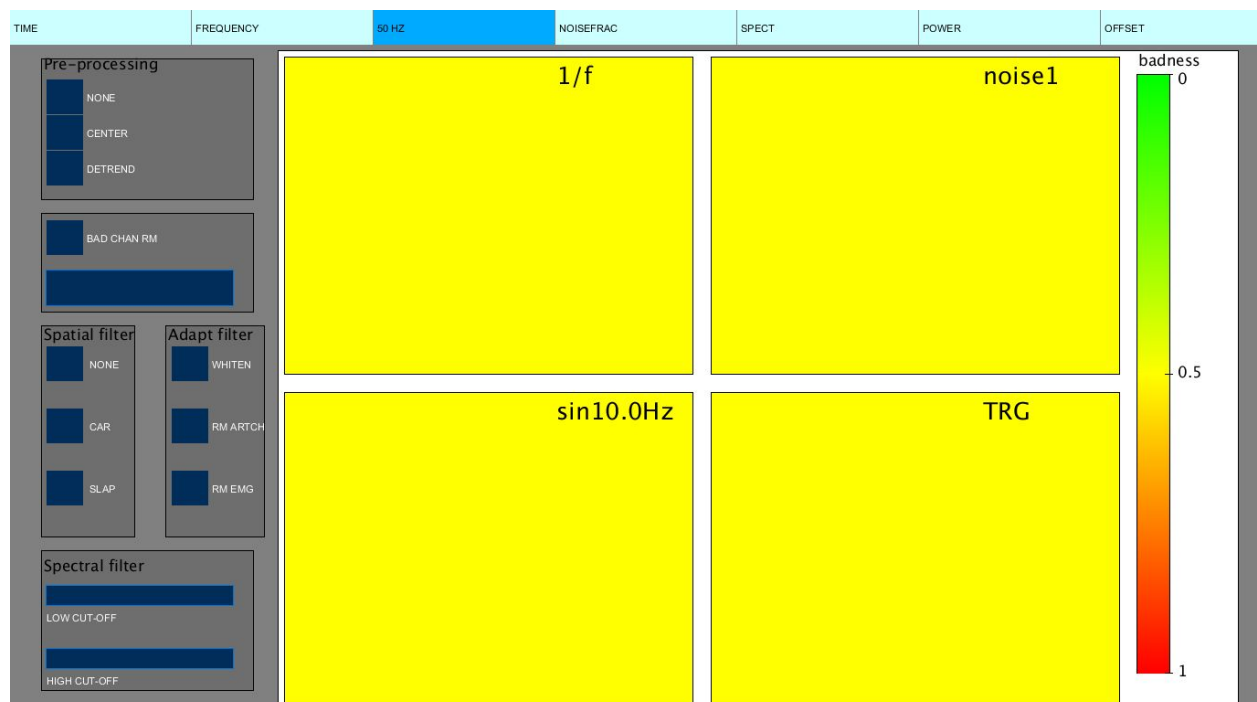


## The Frequency Tab



When this tab is selected the data is shown in the frequency domain. On the x-axis the frequencies are shown in Hz. On the y-axis the power at each frequency in  $\mu\text{V}$  is shown.

## The 50 Hz Tab



This functionality has not been fully implemented yet, and only shows the legend and yellow graphs.

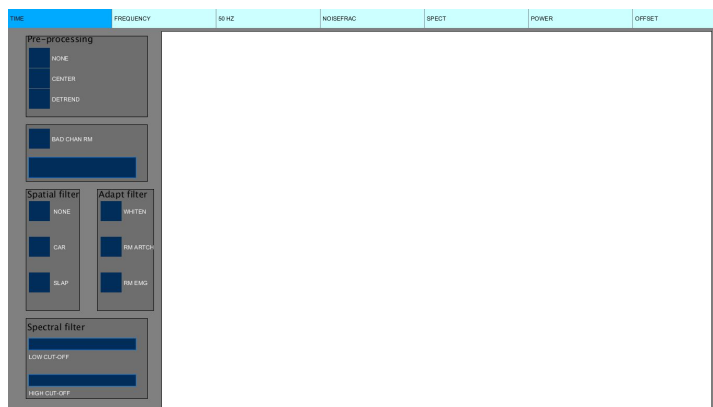
## Common Errors

### BufferClient is not found

**Symptoms:** No graphs appear, and the program freezes.

**Cause:** This error occurs when the buffer is not running or not found at the given IP address.

**Solution:** This is solved by closing the problem and (re)starting the buffer and/or checking the IP address, making sure the right address is entered.



### Android keyboard is unable to type

**Symptoms:** Not being able to type into the text fields.

**Cause:** When the keyboard in the android version is clicked away with the back button, android still recognizes it as being toggled. If it is then toggled again, android thinks it has not been toggled.

**Solution:** If the keyboard is then clicked away again with the back button, android correctly recognizes the keyboard as not being toggled.

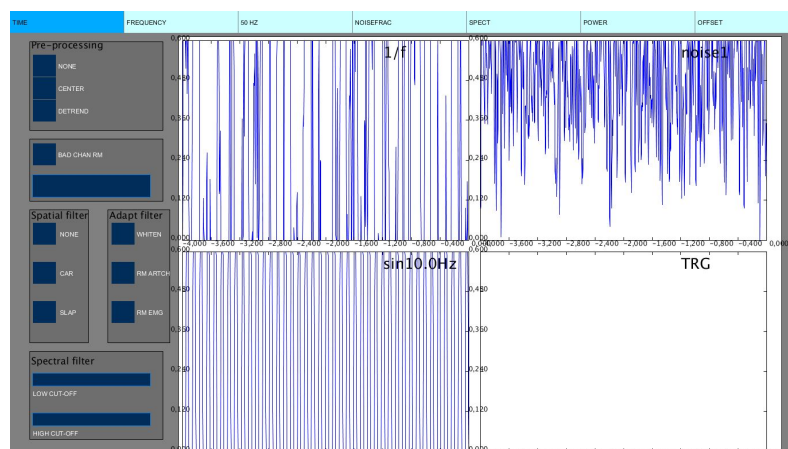
### Slow loading data

**Symptoms:** When starting the data the graphs are not displayed correctly.

**Cause:** It takes some time after starting before the buffer has given enough data and all the statistics are stabilized.

**Solution:** Either by patiently waiting until all the data is there, by switching to a different view or by selecting a preprocessing option.

Before:



After switching between views or selecting a preprocessing option:

