Simulaneous Ca^{2+}/fMRI measurements - Manual

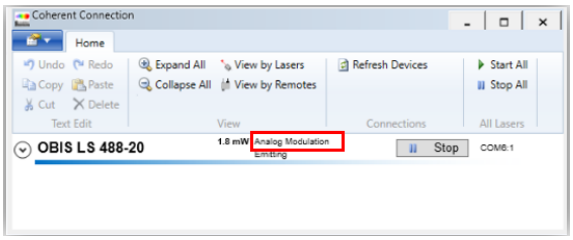
Felix Schlegel, 2017

Institute for Biomedical Engineering, University and ETH Zurich, Switzerland
felix.schlegel@alumni.ethz.ch

All files mentioned in this manual are freely available on github.com/IBT-FMI/calciumfMRI!

1 Laser Setup

Start the control software of your laser (e.g. *Coherent Connection* for Coherent OBIS lasers). Set the emission mode to *Analog Modulation*.

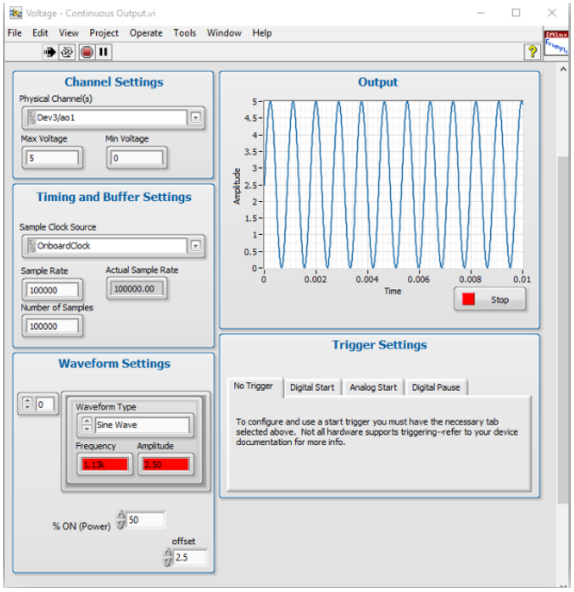


Under analog modulation, the laser power is controlled by an analog signal typically in the range 0 V (no emission) and 5 V (100% emission). Check the manual of your laser for deviations from this standard. Note that 5 V always corresponds to the maximum laser power. Although the software allows reducing the laser power during operation in continuous wave (cw) emission mode, this setting is ignored during analog modulation.

To provide the analog modulation signal, you can use a hardware function generator. Alternatively, a LabView VI is provided, which only requires an analogue output channel on your DAQ box:

Open **LaserModulation.vi**

Select the physical channel of your DAQ device that is connected to the laser's analog input.



Under Waveform Settings, choose Sine Wave. Produce a voltage signal between 0-5 V by setting amplitude = 2.5, and offset = 2.5.

Under Frequency, select a modulation frequency. In our lab, we use a value around 1100 Hz.

Before running the first in-vivo experiment, acquire a noise measurement by setting your laser emission mode to continuous wave (ignoring all the settings on this page), then look at the power spectrum of your data (see Section 4: Data Processing) to determine a frequency range that is not contaminated by any noise in your environment.

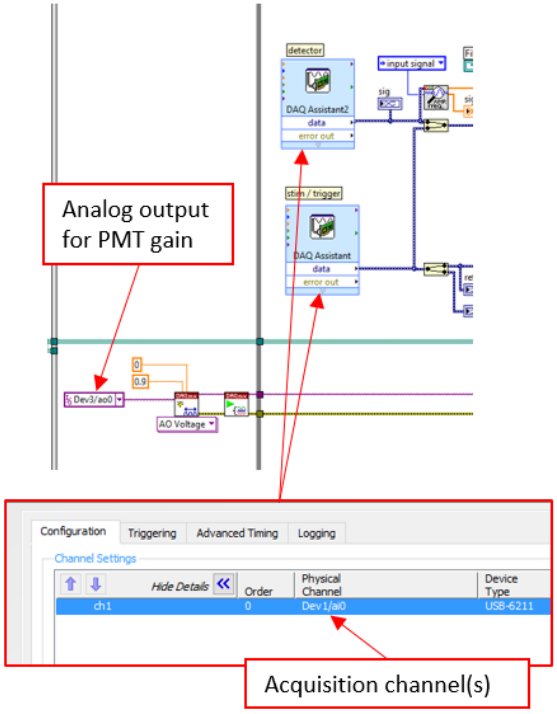
Note, that the values under Sampling Rate and Number of Samples should be a multiple of your

selected frequency. Likewise, as a rule of thumb the sampling frequency in CalciumRecordings.vi (see next section) should be at least 5x the modulation frequency.

2 Calcium Recording Software

Configuration for first time use

Open **CalciumRecordings.vi** and enter the block diagram (CTRL+E). Modify the device names and input/output channels indicated in the screenshot according to your setup (our example VI demonstrates the use of two DAQ devices). At this point, you should have connected two analog inputs (differential configuration) to your DAQ device, one from the PMT amplifier for the calcium signal and one from the TTL output channel of your MR scanner. In addition, one analog output channel is connected to the PMT to control the gain. Alternatively, for stimulus-evoked experiments, you can connect the stimulus output of Stimulator.vi as the second analog input channel to record stimulus events (see next section) and use the MRI TTL output to trigger *Stimulator.vi*. Lastly, choose a default path for the data storage.

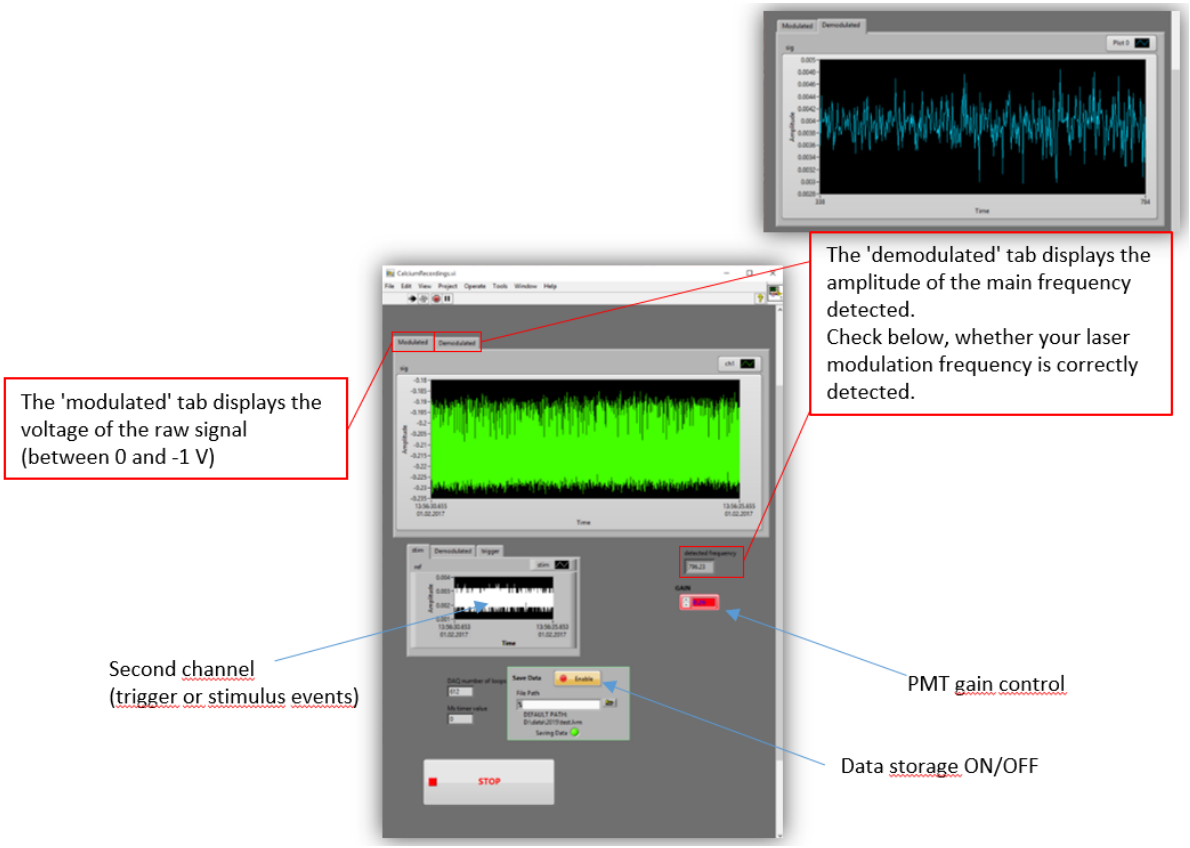


Performing Calcium Recordings

- If the Enable button in the Save Data box is pressed, data will be stored in the default folder in real time. A file *test.lvm* will be c reated each time a recording is started. Previous recordings won't be overwritten but renamed into `backup_XX.lvm`. After each recording, manually retrieve the relevant file from the folder and rename it.

  The following steps are adapted to our PMT (Hamamatsu H10770PA) and might slightly differ for other devices.

- Set the gain initially to a low value (0.2), as the PMT might otherwise shut off automatically due to high current protection.
- Start the VI and observe the signal on the main graph.
- By increasing the gain, the dynamic range in the 'modulated' graph will increase. If the values fall below -1 V, data clipping will occur.
- Increase the gain until you reach an amplitude value of 0.2 in the 'demodulated' graph.
- After adjusting the gain, stop and start the VI again before performing a measurement to create a new file without extreme data fluctuations.



- Once the measurement is complete, press the stop button. The measurement file is automatically created as *test.lvm* in your data folder.

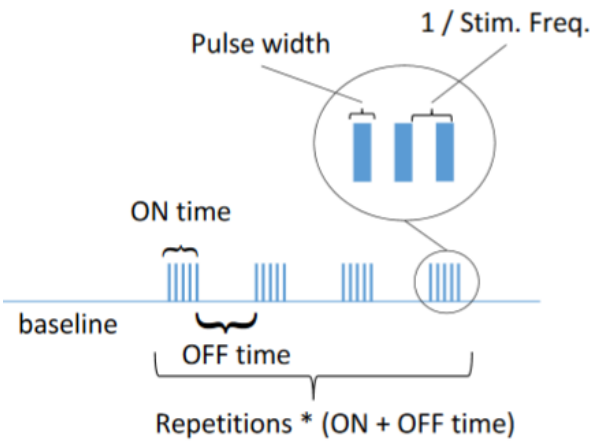
Stimulus Presentation

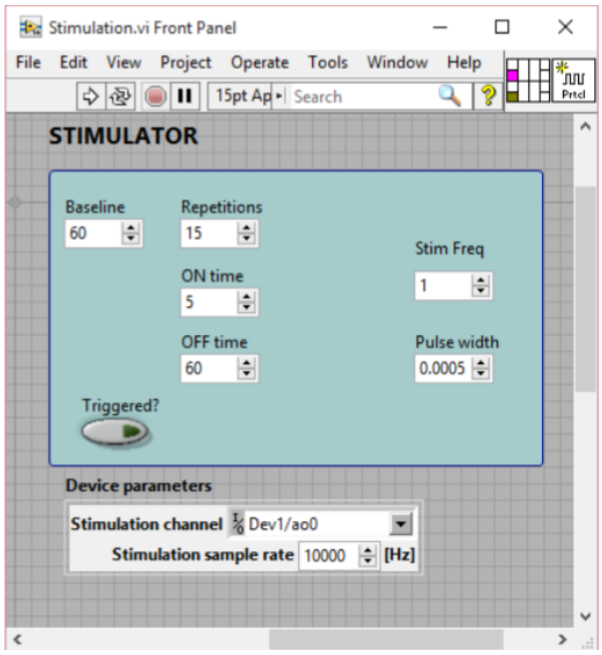
Stimulation.vi produces a TTL output that can be connected to any stimulation device (e.g for electrical paw Stimulation). An array is generated, consisting of zeros (off) and fives (ON). The duration of the array is *Baseline + Repetitions x [ON time + OFF time]* in seconds. During *ON time* of each repetition, 5 V outputs are created according to *Stim Freq.* and *Pulse width* (e.g. 0.5 ms pulses at 1 Hz for 5 seconds, as shown in the screenshot).

Triggered? button: When activated, the stimulation protocol will not start until a trigger input is received (e.g. start of the MRI scan). The trigger input is defined as a drop of the signal from 5V to 0V (trigger event on Bruker systems) on the channel PFI0.

Stimulation channel	Output channel for TTL pulses.
Stimulation sampling rate	Sampling rate of the output. Make sure that this value is high enough to support the pulse width defined above.

To simplify the analysis and alignment of the calcium- and BOLD-data, it is useful to record the stimulus output. For this, connect the signal of the output channel to the 2nd analog input channel defined in *CalciumRecordings.vi* (e.g. using a BNC T-adapter).

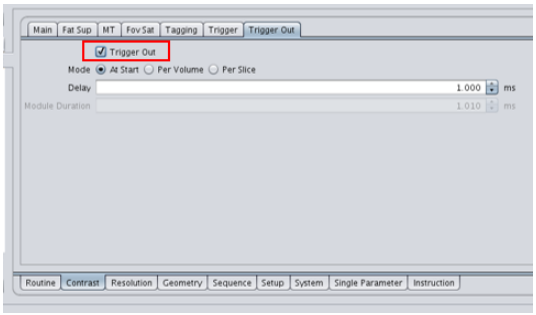




3 Paravision

For MRI consoles running under Bruker ParaVision 6.0.
Ensure that that the trigger output module is on:

- Double click on your EPI scan in the list of scans.
- Click on the ribbon *Contrast*, then *Trigger Out*.
- Ensure that the *Trigger Out* checkbox is checked. It is unchecked by default.



The setting *Mode* is irrelevant as we are only interested in in the first trigger signal. *Delay* can be left at the default value

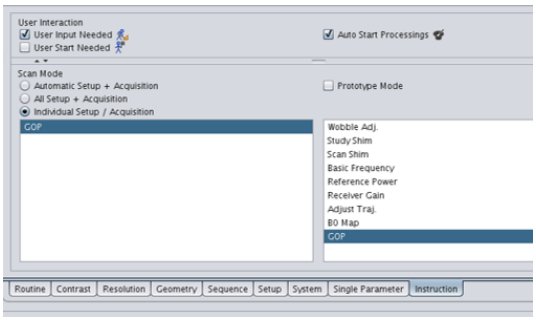
The trigger output module is part of all EPI methods, but may be missing in other methods, such as in the gradient echo FLASH sequence. In such a case it can be added, which however requires basic knowledge of pulse programming on Bruker MRI systems.

Starting a scan for simultaneous acquisition

The pulse sequence will produce a trigger output whenever it is executed, irrespective of whether it is part of the automatic adjustments or of a dummy scan. These false trigger outputs can complicate the alignment with the calcium signal.
To ensure that the first trigger output corresponds to the actual start of the EPI based data acquisition:

- Set the number of dummy scans to zero.
- Under the ribbon *Instruction*, drag *GOP* from the list on the right to the left.

This will skip all adjustment steps such as calculation of the shims and use the previous settings. As anesthetized animals do not move, it is sufficient to run the EPI sequence once with *Automatic Setup + Acquisition* selected, and then use *GOP* for subsequent scans



4 Processing calcium data in Matlab

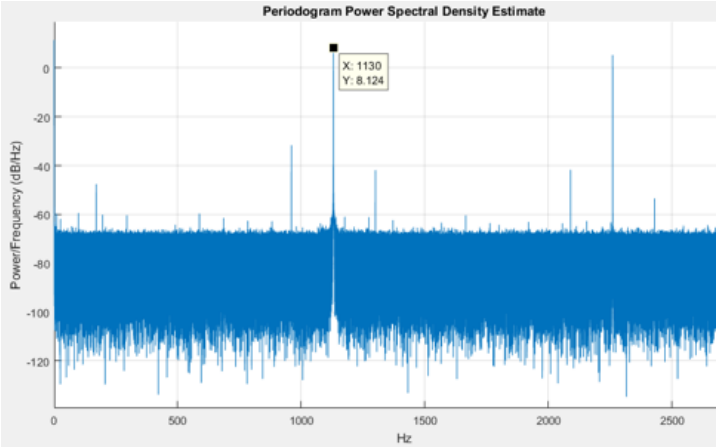
- Open **demodulateSignal.m** in the matlab editor.
  The script is divided into different sections (separated by %%). As you may need to make adjustments to the code or encounter errors, it is best to run each section separately by pressing **CTRL + ENTER**. The script requires a few sub-functions, which are all contained in our github repository.

- Change the Matlab working directory to the folder containing your calcium data created with *CalciumRecordings.vi* by typing:

  cd 'C:\YourCalciumFolder'

Section 1: Data Import

In the command window you will be prompted to enter the name of the *.lvm file and the modulation frequency that you set for the laser. A power spectrum will be displayed where you can also confirm the modulation frequency.

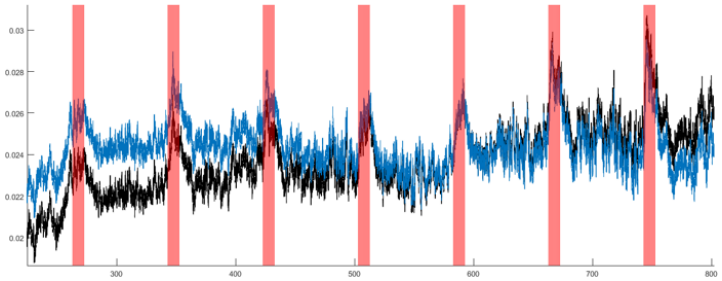


An error message may occur, as the last few data points may not fit into a 50 ms time bin.
This can simply be ignored.
Among others, the following variables are created in the workspace:

data	structure containing the raw data and additional metadata provided by LabView.
ch1	Raw signal of the first analog channel (assumed to be fluorescence signal)
ch2	Raw signal of the second analog channel (assumed to be the stimulus/trigger input)
demodSig	The demodulated signal time course (in 50 ms time resolution).
t_downs	The time axis for demodSig (50 ms time steps).

Section 2: Detrending and plotting

This code section will perform a simple detrending fitting and subtracting a polynomial from the data. You can change the order of the polynomial in the line *opol = 2;*
Lastly, the demodSig time course is plotted prior (black) and after (blue) detrending. Stimulus pulses are indicated by vertical red lines.



Relevant variables created in this section include:

filtered_signal	The detrended demodSig time course.
stim_events	An array containing the time points of stimulus pulses (unit is seconds).

At this point you may want to apply your particular filtering approach, $\Delta F/F$ calculation, averaging of stimulus repetitions etc. which is not covered in the example script as it depends on the particular experimental situation.

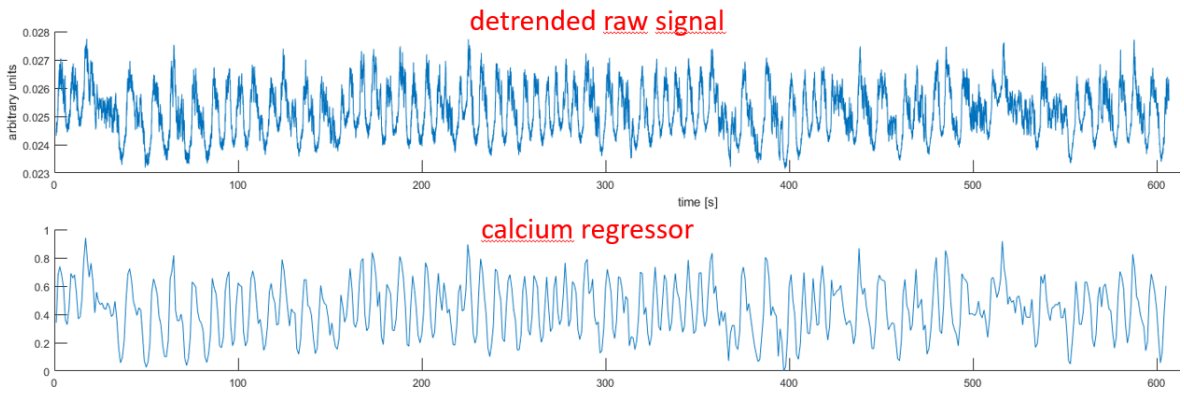
Section 3: Calcium Regressor

The last section will create and store the calcium time course as an ASCII file to be used in AFNI afni.nimh.nih.gov!
Two constants may have to be adjusted in the code:

cutoff_time	The time (in seconds) of the initial EPI volumes that are discarded from analysis (this is typically done in order to allow the MRI signal to reach magnetization equilibrium).
bin_size	Sampling rate of the calcium signal (after demodulation) multiplied with the EPI repetition time. Default value = 20, (20 Hz sampling rate and 1 s TR). Time points of the calcium trace within a bin will be averaged before downsampling to fMRI time resolution.

The *filtered_signal* time course will be cut depending on the first *stim_events* time point and the chosen *cutoff_time* so that it will be aligned with the fMRI time course during the analysis in AFNI.

The signal will then be smoothed by binning, downsampled and rescaled between 0 and 1 (in order to prevent rounding errors). The script will generate plots of the (cropped) detrended raw signal, and the newly generated *calcium_regressor*.



Note that the duration of the calcium regressor is not important. If it is longer or shorter than the fMRI time course, the additional or missing time points will be ignored during analysis in AFNI.

If you notice an artifact in the calcium signal, document the time (in seconds) of its occurrence. This time period can later be censored from the analysis in AFNI.
Two ASCII files will be created on the hard disk:
calcium_regressor.1D and *calcium_regressor_time_axis.1D* which simply contain the corresponding time and data pairs of the entire signal trace.
This is suitable for most resting state and stimulus evoked analyses. Alternatively, only certain parts of the calcium signal could be selected (e.g. stimulus periods) by providing their corresponding time/data pairs. The regressor does not have to be a continuous time series.

5 fMRI analysis in AFNI

Preparation

We assume that the fMRI data has been converted into the AFNI brik/head format and standard preprocessing steps have been applied. These include motion correction, slice-timing correction, coregistration (if required), smoothing and detrending of the data.
For an introduction into AFNI, we recommend going through the tutorial that is provided under afni.nimh.nih.gov/pub/dist/edu/data/CD.expanded/AFNI_data6/FT_analysis/tutorial/
The program 3dDeconvolve will be used for the General Linear Model (GLM) analysis, which requires a single file consisting of time*amplitude pairs for the calcium regressor. To create such a file, use the following console command:

```
1dMarry calcium_regressor_time_axis.1D calcium_regressor.1D \\
> calcium_regressor_married.1D
```

where the first two arguments of 1dMarry are the individual time and amplitude files created with Matlab and the output is a combined file compatible with 3dDeconvolve.

To conveniently execute AFNI programs, all the commands should be written in a tcsh shell script. The script AFNI_analysis in our repository provides a complete example of an analysis pipeline.

GLM analysis

The following command will perform a simple GLM analysis using the calcium signal trace as the only regressor:

```
3dDeconvolve -input CleanedData+orig                              \
             -TR_times 1                                          \
             -num_stimts 1                                        \
             -stim_times_AM1 1 calcium_regressor_married.1D       \
             'SPMG3(0.01)'                                        \
             -stim_label 1 calcium                                \
             -iresp 1 HRF                                         \
             -fitts calcium_model                                 \
             -bucket stats
```

The only two inputs required are *CleanedData+orig*, which is the preprocessed fMRI dataset, and *calcium_regressor_married.1D*, which was created in the previous step. Further parameters are *-TR_times X* defining repetition time in seconds, *SPMG3(0.01)* selecting the SPM basis set as the hemodynamic response function (HRF), consisting of the sum of three gamma variates which allows a certain variability in the response shape. The value in the brackets indicates the stimulus duration in seconds. As we do not want to introduce additional temporal blurring, simply enter a very small value here.

As an option, the parameter *-censorTR X* will ignore a particular time index (unit in TR, not seconds) from analysis, e.g. *–censorTR 100-120* will ignore time index 100-120.

For more options, consult the documentation for 3dDeconvolve under:
afni.nimh.nih.gov/pub/dist/doc/program_help/3dDeconvolve.html!

The following parameters control the outputs generated by the GLM analysis:

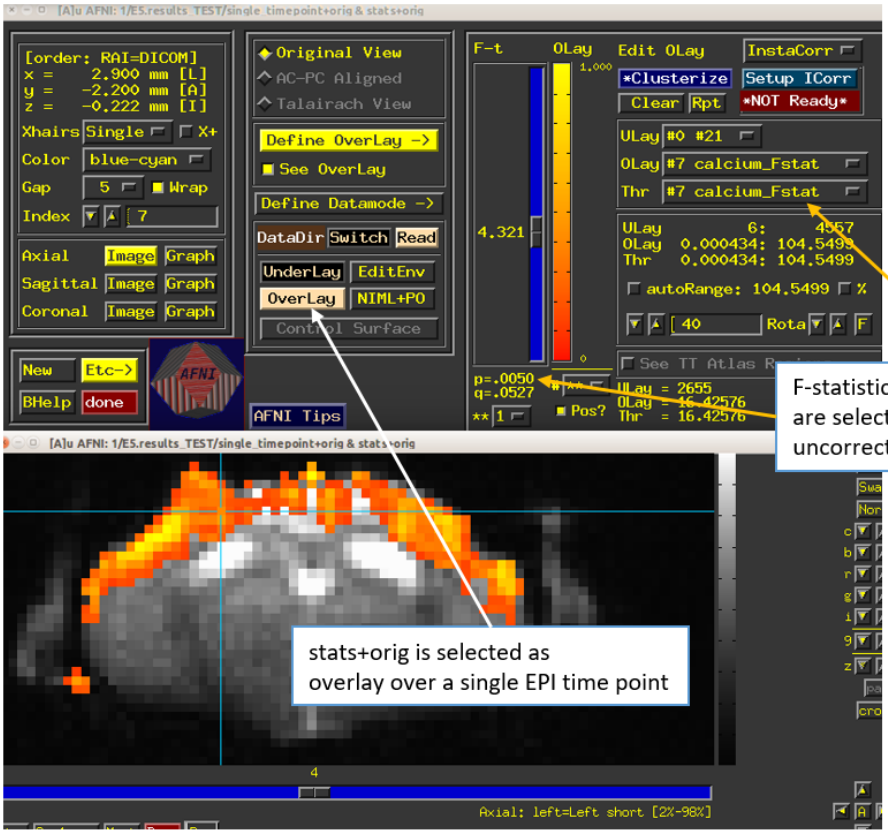
-iresp 1 HRF Twill create the file *HRF+orig*, containing the HRF that was fitted to each voxel as the sum of the three SPM basis functions.

-fitts calcium_model will create the file *calcium_model+orig* which contains the model time series for each voxel (i.e. the calcium regressor convolved with the HRF).

-bucket stats will create the file *stats+orig* which contains all the fit parameters and statistics of the GLM analysis.

Data visualization - Example

The following screenshots show examples of the 3dDeconvolve GLM outputs when displayed in the AFNI GUI.



time course: