Data Editor and Command Line Preprocessing

Initialization on felix/helix

Data directory is here: /data/classes/meg

```
$ . init_course.sh
```

```
module load ctf
Module load python
module load samsrcv3
module load afni
```

```
export PYTHONPATH=pyctf/bin:pyctf/pyctf:$PYTHONPATH
export PATH=pyctf/bin:$PATH
```

Where is my data

- Initially data is stored on squid.nimh.nih.gov, the acquisition computer
- Every night, data is transferred to tako.nimh.nih.gov where it will appear in your data directory organized by date
- Raw data is also transferred to the helix systems
- The MEG core facility DOES NOT provide archival, off-site backup. We do not take responsibility for lost data.

Data Structure

EYZQADGL_asvef_20180608_01.ds/

BadChannels	←simple text file with bad channel names
bad.segments	← simple text file with bad segment times
ClassFile.cls	
DigTrigChannelInfo.txt	
EYZQADGL_asvef_20180608_01.acq	
EYZQADGL_asvef_20180608_01.eeg	
EYZQADGL_asvef_20180608_01.hc	
EYZQADGL_asvef_20180608_01.hist	←dataset acquisition and modification hist
EYZQADGL_asvef_20180608_01.infods	
EYZQADGL_asvef_20180608_01.meg4	←raw data file
EYZQADGL_asvef_20180608_01.newds	
EYZQADGL_asvef_20180608_01.res4	
hz2.ds/	←head localization information
hz.ds/	
MarkerFile.mrk	←marker info
processing.cfg	←filtering info

Sample Dataset

- Five Subjects
- Three Tasks:
  - Continuous Performance Task
  - ASVEF Evoked Response Task
  - Resting state

Continuous Performance Task

- Letters presented one at a time
- Letters appear in either left or right visual field
- Block DNRX: Subjects are asked to respond to every letter except the letter X
- Block DNR: Subjects are asked to passively
- Each letter is presented 8 times
- The letter X is presented 64 times

ASVEF Task

- Randomized sensory stimuli
  - Somatosensory airpuff stimulus to either left or right hand
  - Auditory tone to both ears
  - Checkerboard stimulus in left or right visual field
  - Approximately 100 of each stimulus type

Resting State

- Eyes open
- Fixation

MRI Directory

- MRI with skull but with face removed
- Fiducial markers already placed as tags in the AFNI image

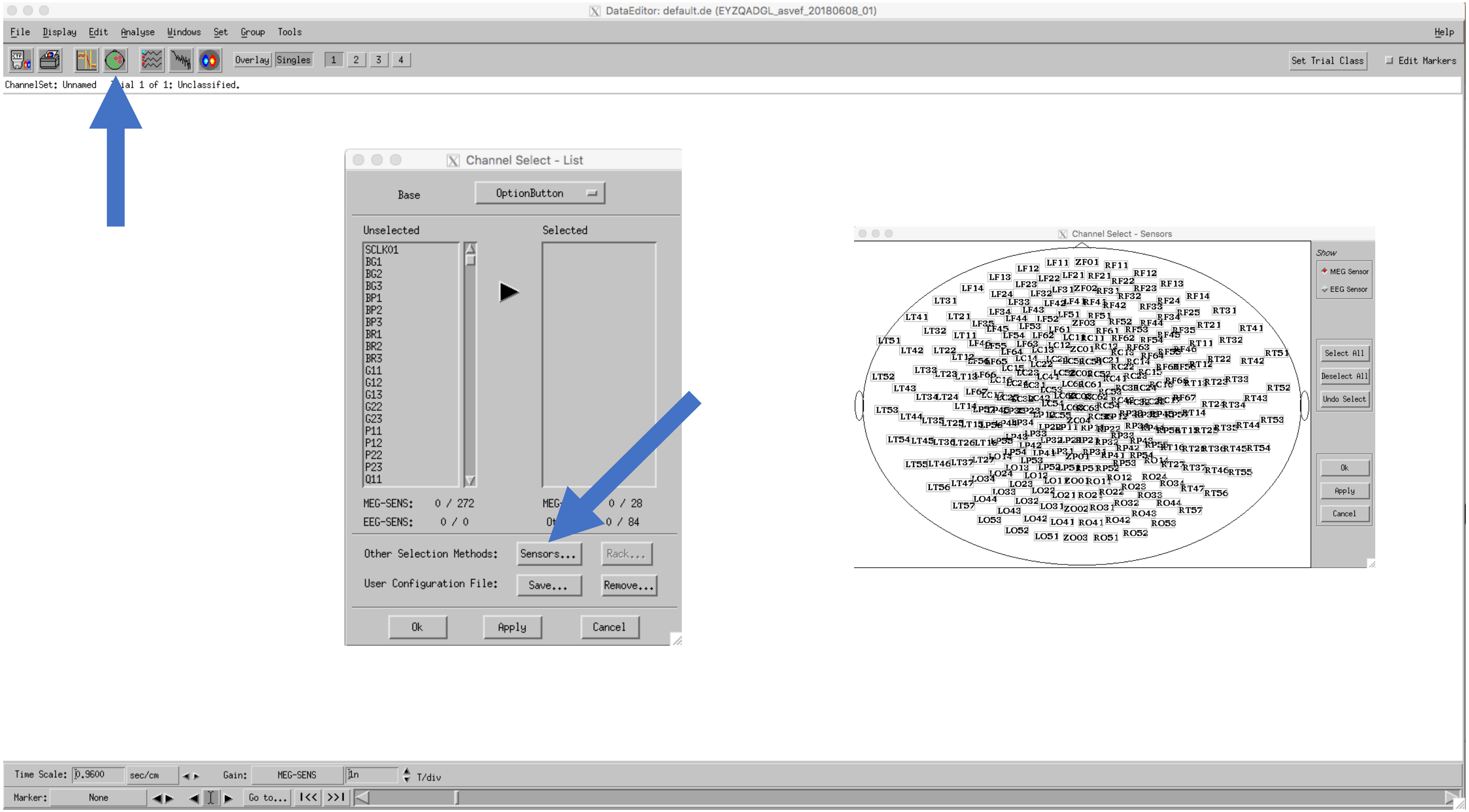
Stimulus delivery channels

- The optical sensor feed is on UADC016
- For the asvef datasets:
  - Left airpuff = UADC001
  - Right airpuff = UADC002
  - Beep = UADC003
- For the cpt dataset
  - UADC005 = response

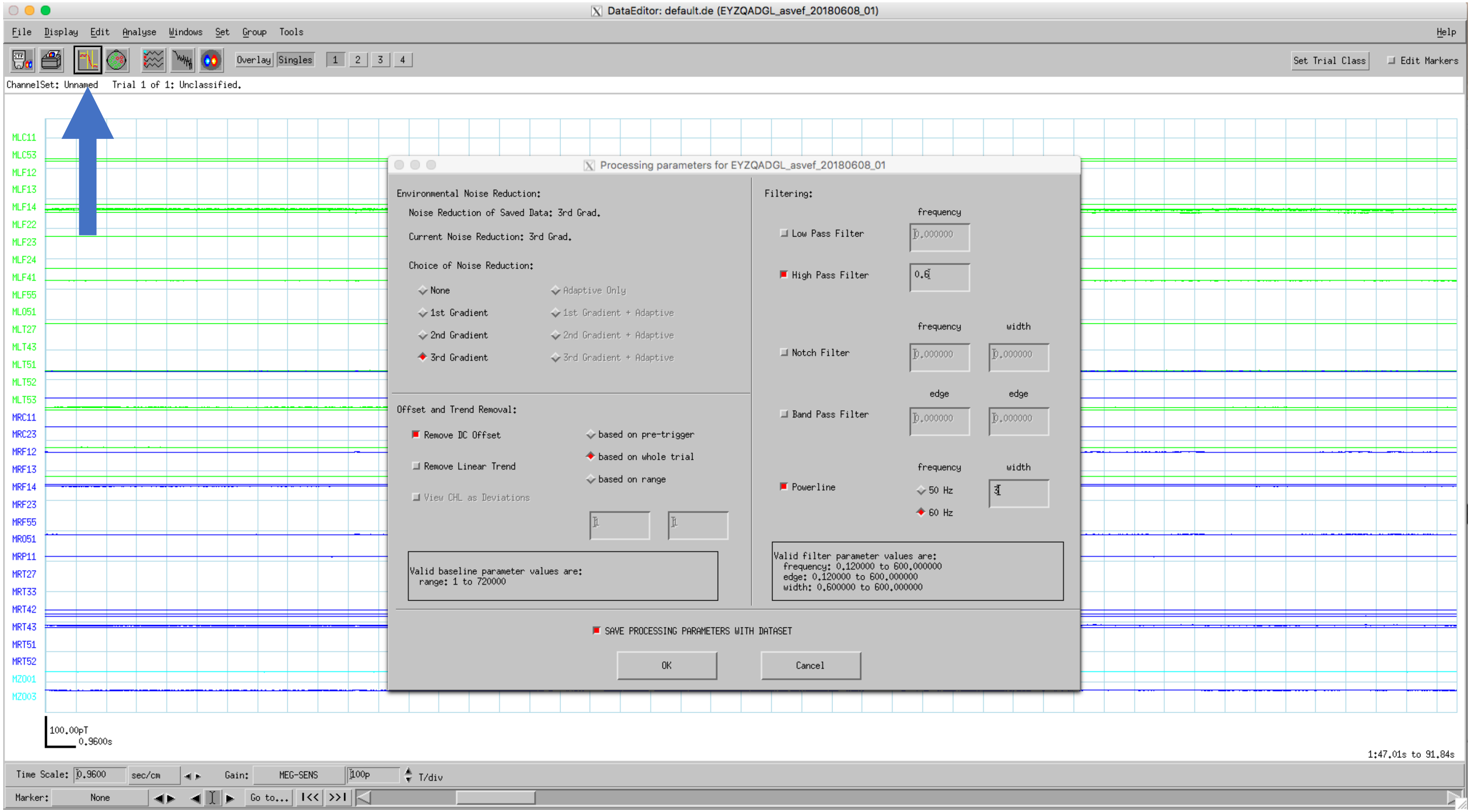
Data Editor

```
$ DataEditor –data EYZQADGL_asvef_20180608_01.ds
```

Selecting Channels



Filtering



ChannelSet: Unnamed Trial 1 of 1: Unclassified.

Processing parameters for EYZQADGL_asvef_20180608_01

Environmental Noise Reduction:

Noise Reduction of Saved Data: 3rd Grad.

Current Noise Reduction: 3rd Grad.

Choice of Noise Reduction:

- ○ None
- ○ 1st Gradient
- ○ 2nd Gradient
- ● 3rd Gradient
- ○ Adaptive Only
- ○ 1st Gradient + Adaptive
- ○ 2nd Gradient + Adaptive
- ○ 3rd Gradient + Adaptive

Offset and Trend Removal:

- ☒ Remove DC Offset
- □ Remove Linear Trend
- □ View CHL as Deviations
- ○ based on pre-trigger
- ● based on whole trial
- ○ based on range

1 1

```
Valid baseline parameter values are:
  range: 1 to 720000
```

Filtering:

	frequency	width
□ Low Pass Filter	0.000000	
☒ High Pass Filter	0.6	
□ Notch Filter	0.000000	0.000000

	edge	edge
□ Band Pass Filter	0.000000	0.000000

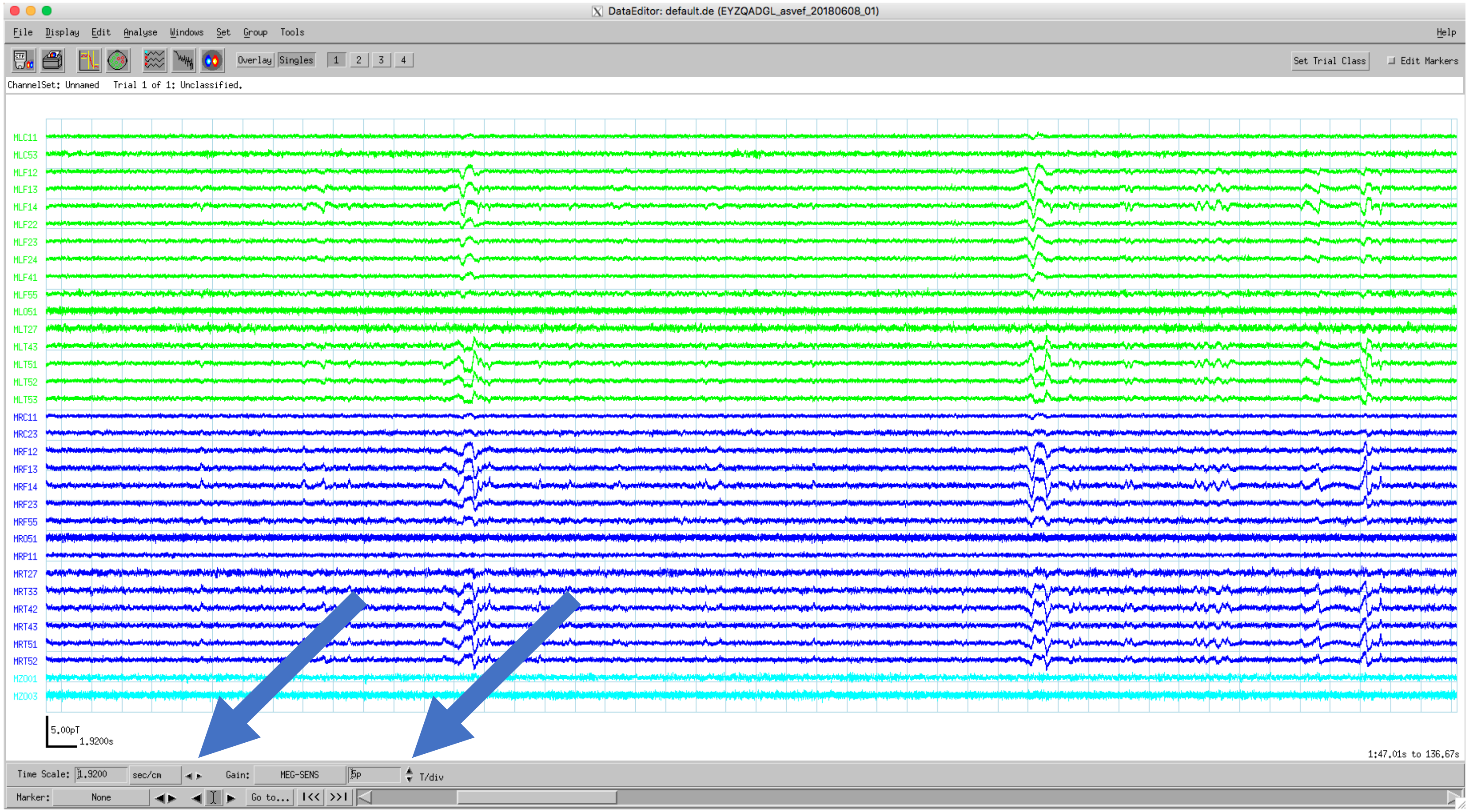
	frequency	width
☒ Powerline	○ 50 Hz ● 60 Hz	3

```
Valid filter parameter values are:
  frequency: 0.120000 to 600.000000
  edge: 0.120000 to 600.000000
  width: 0.600000 to 600.000000
```

☒ SAVE PROCESSING PARAMETERS WITH DATASET

OK Cancel

Setting Scale



DataEditor: default.de (EYZQADGL_asvef_20180608_01)

ChannelSet: Unnamed   Trial 1 of 1: Unclassified.

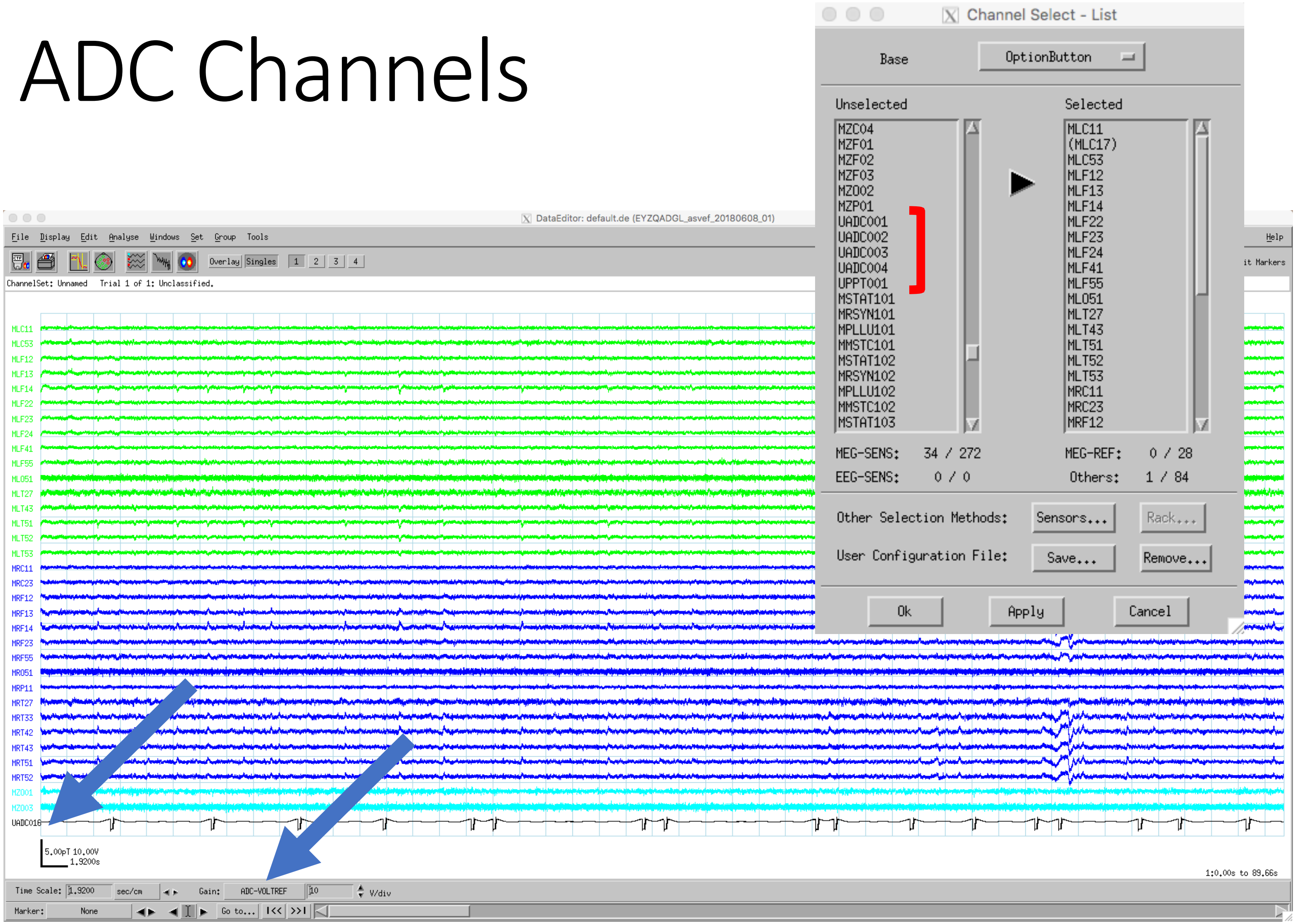
Time Scale: 1.9200 sec/cm

Gain: MEG-SENS 5p T/div

Marker: None

1:47.01s to 136.67s

ADC Channels



DataEditor: default.de (EYZQADGL_asvef_20180608_01)

ChannelSet: Unnamed   Trial 1 of 1: Unclassified.

Time Scale: 1.9200 sec/cm   Gain: ADC-VOLTREF 10 V/div

Marker: None

Channel Select - List

Base: OptionButton

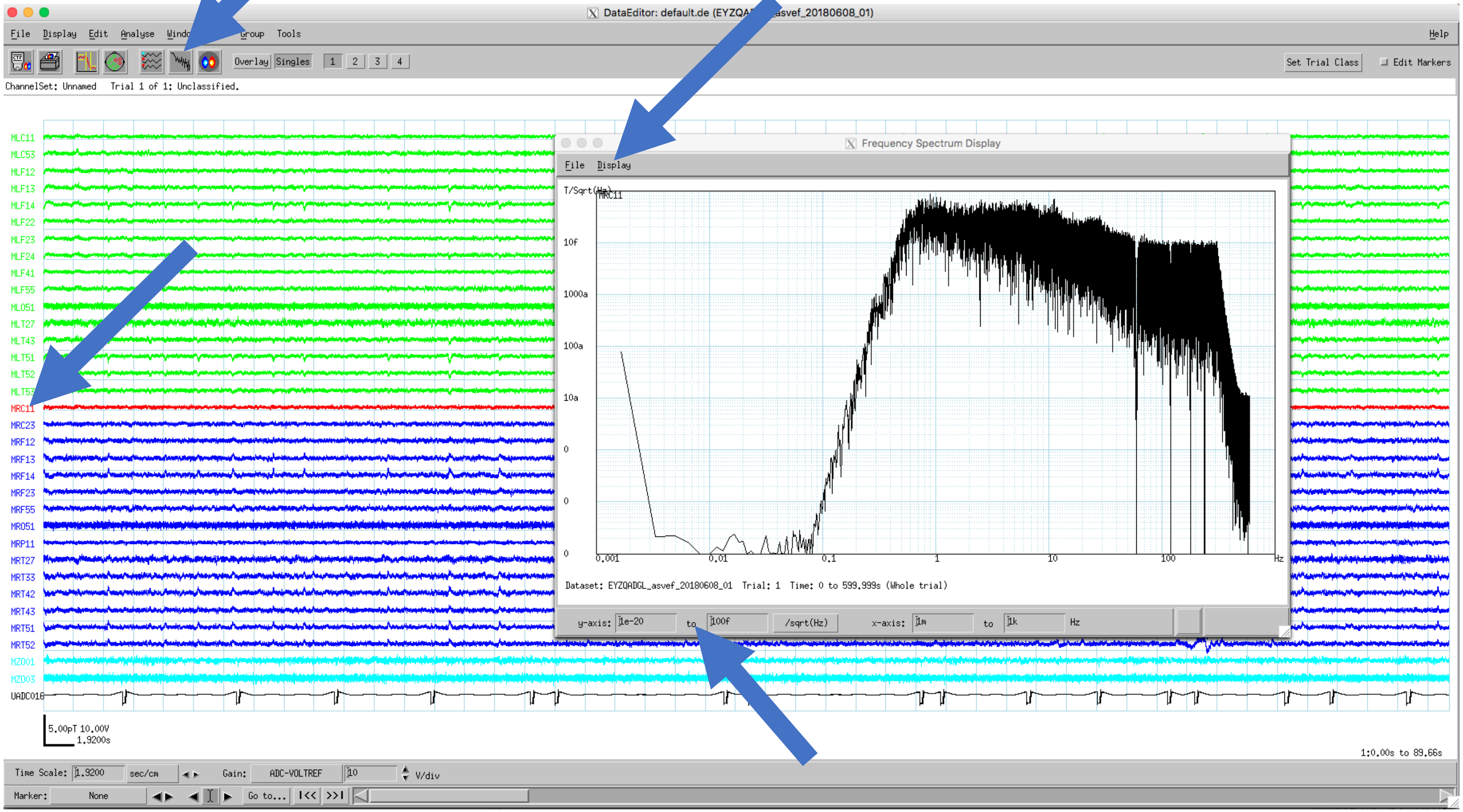
Unselected:
MZC04
MZF01
MZF02
MZF03
MZO02
MZP01
UADC001
UADC002
UADC003
UADC004
UPPT001
MSTAT101
MRSYN101
MPLLU101
MMSTC101
MSTAT102
MRSYN102
MPLLU102
MMSTC102
MSTAT103

Selected:
MLC11
(MLC17)
MLC53
MLF12
MLF13
MLF14
MLF22
MLF23
MLF24
MLF41
MLF55
MLO51
MLT27
MLT43
MLT51
MLT52
MLT53
MRC11
MRC23
MRF12

MEG-SENS: 34 / 272   MEG-REF: 0 / 28

EEG-SENS: 0 / 0   Others: 1 / 84

Frequency Spectra



ChannelSet: Unnamed   Trial 1 of 1: Unclassified.

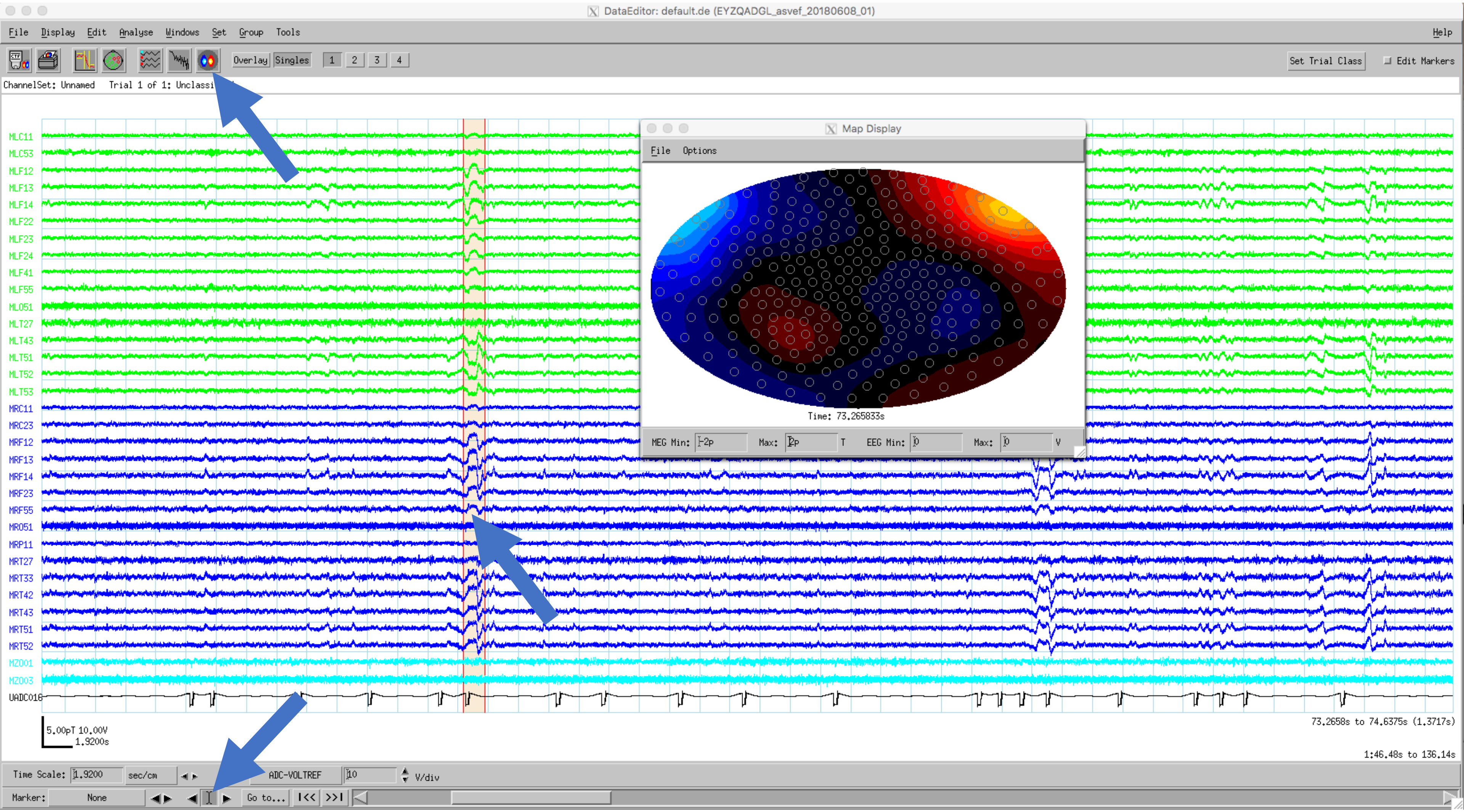
Dataset: EYZQADGL_asvef_20180608_01 Trial: 1 Time: 0 to 599.999s (Whole trial)

y-axis: 1e-20 to 100f /sqrt(Hz)   x-axis: 1m to 1k Hz

Time Scale: 1.9200 sec/cm   Gain: ADC-VOLTREF 10 V/div

Marker: None

Artifacts


DataEditor: default.de (EYZQADGL_asvef_20180608_01)

ChannelSet: Unnamed   Trial 1 of 1: Unclassi

Map Display

Time: 73.265833s

MEG Min: -2p   Max: 2p T   EEG Min: 0   Max: 0 V

5.00pT 10.00V
1.9200s

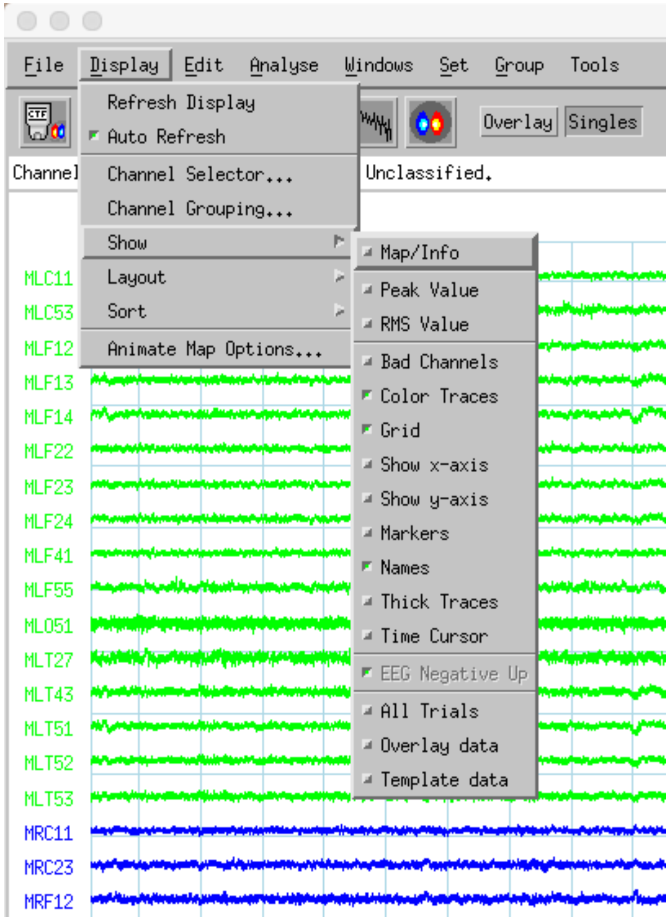
73.2658s to 74.6375s (1.3717s)

1:46.48s to 136.14s

Time Scale: 1.9200 sec/cm   ADC-VOLTREF 10 V/div

Marker: None

Display options



Add markers: Thresholding

```
$thresholdDetect2 –help
            -m <marker>
            -i
            -np
            -a <amplThresh>
            -d <derivThresh>
            -dt <deadtime>
            -c <channel>
```

```
$ thresholdDetect2 -np -m leftpuff -a .5 -d .5 -dt 1
            -c UADC001 EYZQADGL_asvef_20180608_01.ds
```

Add markers: Parsing

```
$ parsemarks –help
            -m <marker>
            -q <query>
```

Example Query: “name == ‘resp’ and inwindow(-.5, 0, ‘stim’)
i.e. collect all responses where the stim marker is at least 500 ms before the response marker

```
$ addMarker –help
            -f
            -n <marker>
            -p <textfile>
```

Filtering

```
$ newDs –help
        -f
        -includeBadChannels
        -includeBadSegments
        -includeBad
        -filter <filename>
        -time <start> <end>

$ newDs –f –filter processing.cfg
        EYZQADGL_asvef_20180608_01.ds
        EYZQADGL_asvef_20180608_01-f.ds
```

Processing.cfg

```
// Processing configuration.
// Defaults for normal analysis.

// PROCESSING PARAMETERS
processing
{
      // balance: order, adapted
      // (adapted=0 -> not adapted)
      // (adapted=1 -> adapted)
      balance:       3,0                                          ← Third order gradient balancing
      // lowpass: enable, filterOrder, fc
      lowpass:       0,4,0.00000000000000000                      ← Low-pass filter

      // highpass: enable, filterOrder, fc
      highpass:      1,4,0.50000000000000000                      ← High-pass filter
      // bandreject: enable, filterOrder, fc1, fc2
      bandreject:    0,4,0.00000000000000000,0.00000000000000000
      // bandpass: enable, filterOrder, fc1, fc2
      bandpass:      0,4,0.00000000000000000,0.00000000000000000
      // bandreject: enable, filterOrder, fc1, fc2
      bandreject:    1,2,59.50000000000000000,60.50000000000000000          ← Powerline filter
      // bandreject: enable, filterOrder, fc1, fc2
      bandreject:    1,2,119.50000000000000000,120.50000000000000000        ← Powerline harmonics
      // bandreject: enable, filterOrder, fc1, fc2
      bandreject:    1,2,179.50000000000000000,180.50000000000000000
      // bandreject: enable, filterOrder, fc1, fc2
      bandreject:    1,2,239.50000000000000000,240.50000000000000000
      // offset: enable, baselineSelection, startPt, endPt
      // (baseline=0 --> use pretrigger data)                               ← Removing DC offset
      // (baseline=1 --> use from startPt to endPt)
      // (baseline=2 --> use whole trial)
      // (baseline+=10 --> do trend removal)

      offset: 1,2,1,1
}
```

Better yet – use a script!

```
$ doThresh EYZQADGL_asvef_20180608_01.ds
```

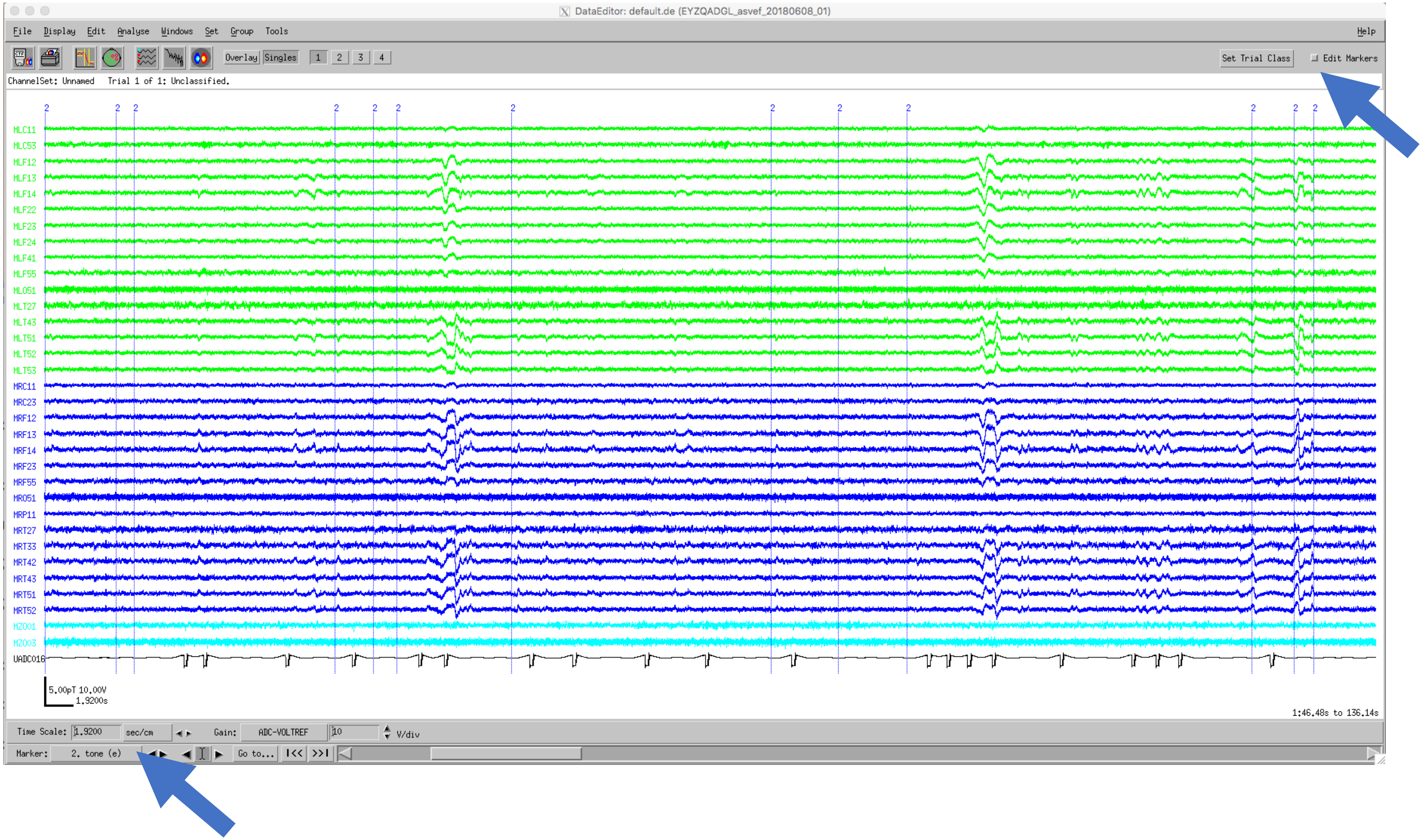
All initial thresholding, parsing markers, filtering

Tom Holroyd will help you write specific scripts for your task.

Output files from ePrime, Presentation, etc. can be parsed for identifying markers

More efficient approach – use the active pixel on the Propixx!

Markers



DataEditor: default.de (EYZQADGL_asvef_20180608_01)

ChannelSet: Unnamed   Trial 1 of 1: Unclassified.

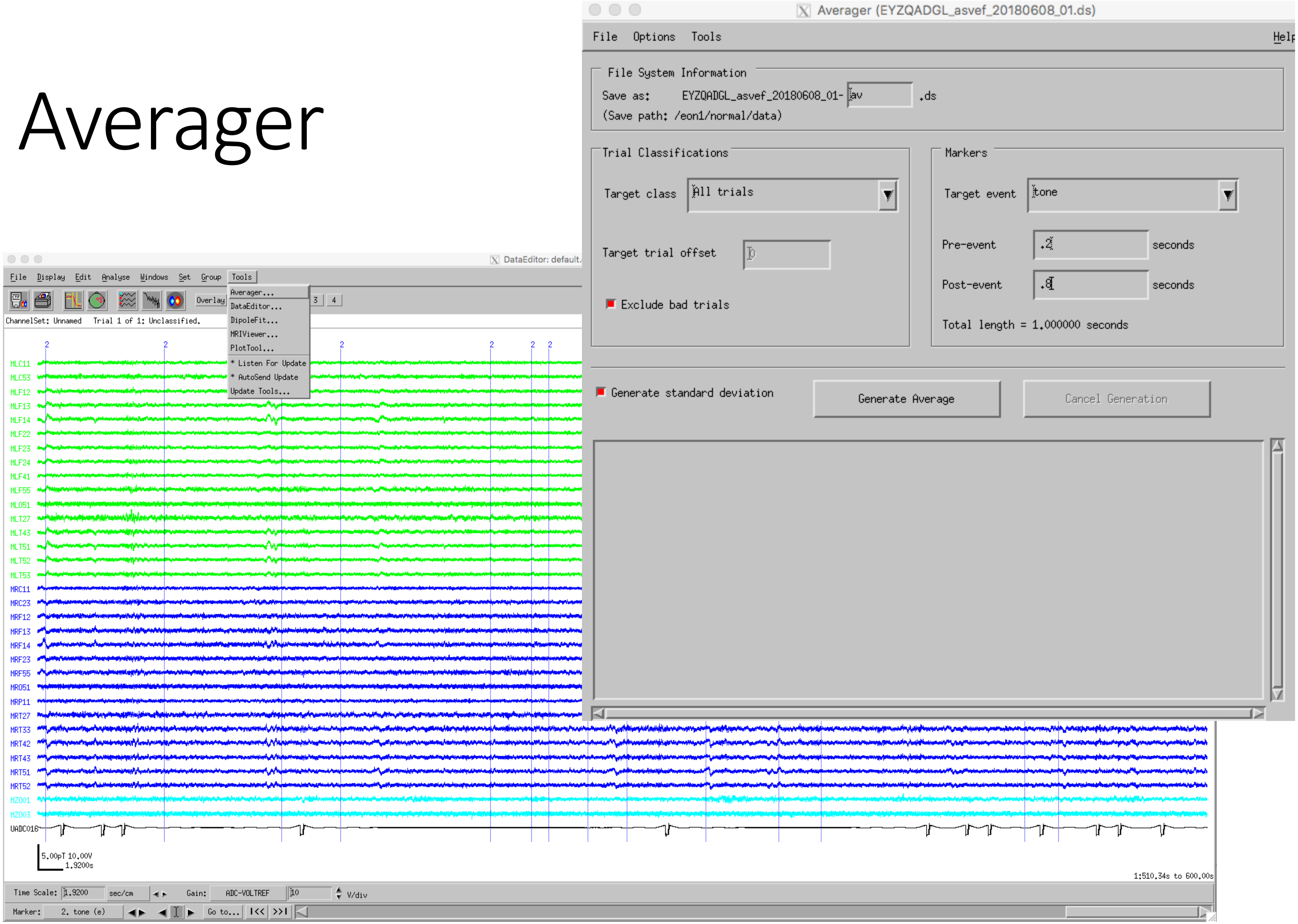
1:46.48s to 136.14s

Time Scale: 1.9200 sec/cm

Gain: ADC-VOLTREF 10 V/div

Marker: 2. tone (e)

Averager



Averager (EYZQADGL_asvef_20180608_01.ds)

File System Information

Save as: EYZQADGL_asvef_20180608_01- av .ds
(Save path: /eon1/normal/data)

Trial Classifications

Target class: All trials

Target trial offset: 0

☒ Exclude bad trials

Markers

Target event: tone

Pre-event: .2 seconds

Post-event: .8 seconds

Total length = 1.000000 seconds

☒ Generate standard deviation

DataEditor: default.

ChannelSet: Unnamed Trial 1 of 1: Unclassified.

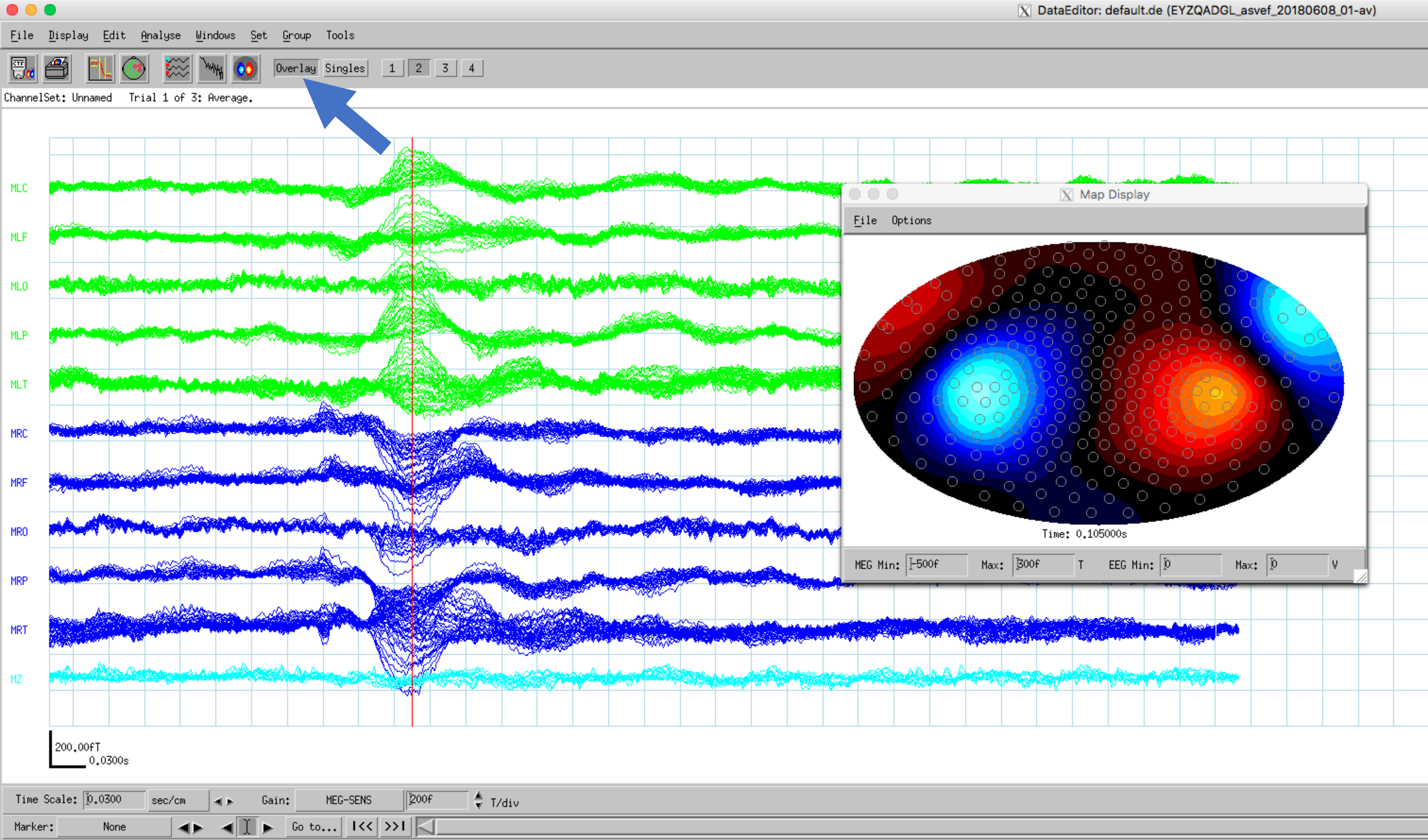
5.00pT 10.00V 1.9200s

1:510.34s to 600.00s

Time Scale: 1.9200 sec/cm Gain: ADC-VOLTREF 10 V/div

Marker: 2. tone (e)

ERF to Tone Stimulus



DataEditor: default.de (EYZQADGL_asvef_20180608_01-av)

ChannelSet: Unnamed Trial 1 of 3: Average.

Map Display

Time: 0.105000s

MEG Min: -500f Max: 300f T EEG Min: 0 Max: 0 V

200.00fT
0.0300s

Time Scale: 0.0300 sec/cm Gain: MEG-SENS 200f T/div

Marker: None

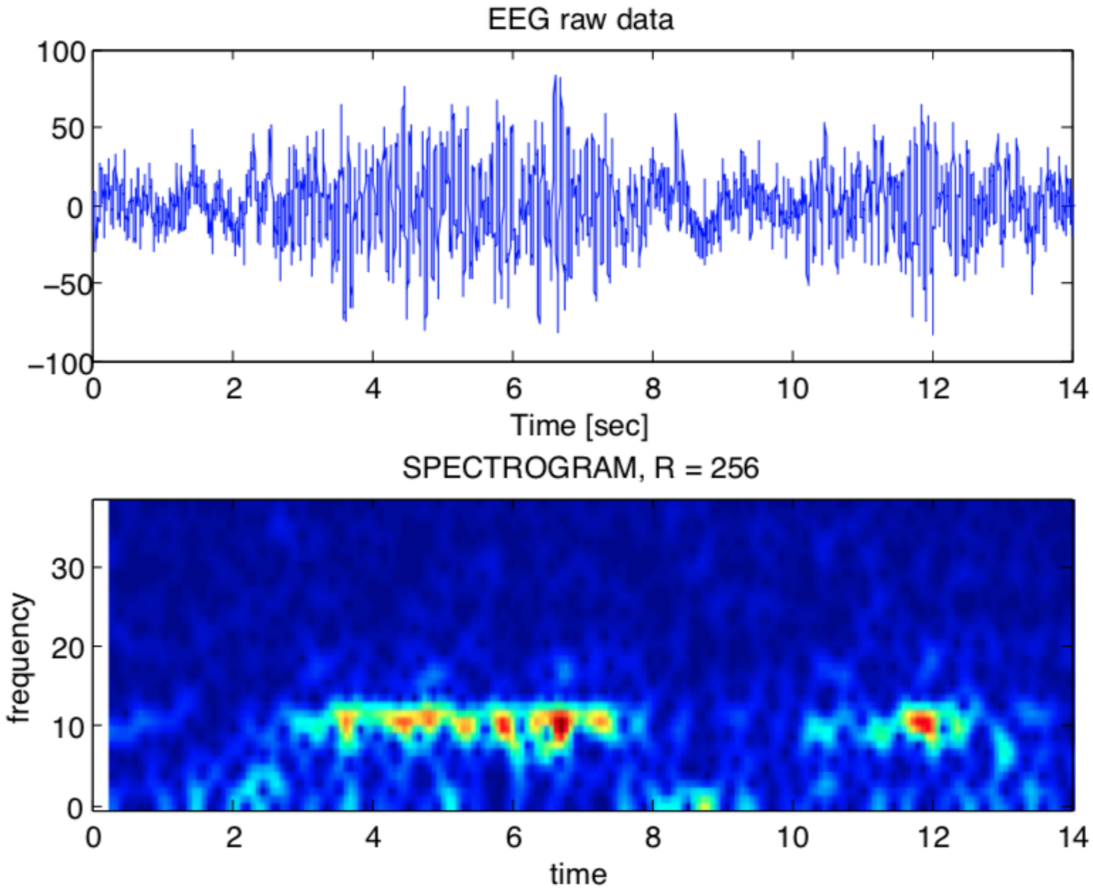
Pyctf

- Python toolbox for interacting with CTF Datasets
- Authored by Tom Holroyd
- Command line tools, as well as modules for reading and writing CTF datasets
- Current implementation is in Python2, Python3 module is in process, as is functionality to deal with new .ds datasets

Pyctf – command line tools

- fiddist – reports the distance between the fiducial markers in both .ds datasets and AFNI tags
- Several tools to determine head movement
- Several tools to repair datasets
- StockwellDs – makes time frequency plots

How do I know where and at what frequency to look? Time-Frequency Analysis



Niko.busch@charite.de

StockwellDs

-m marker

-t “t0 t1”

-b “lo hi”

-c channel

-n

-B “t0 t1”

-o prefix

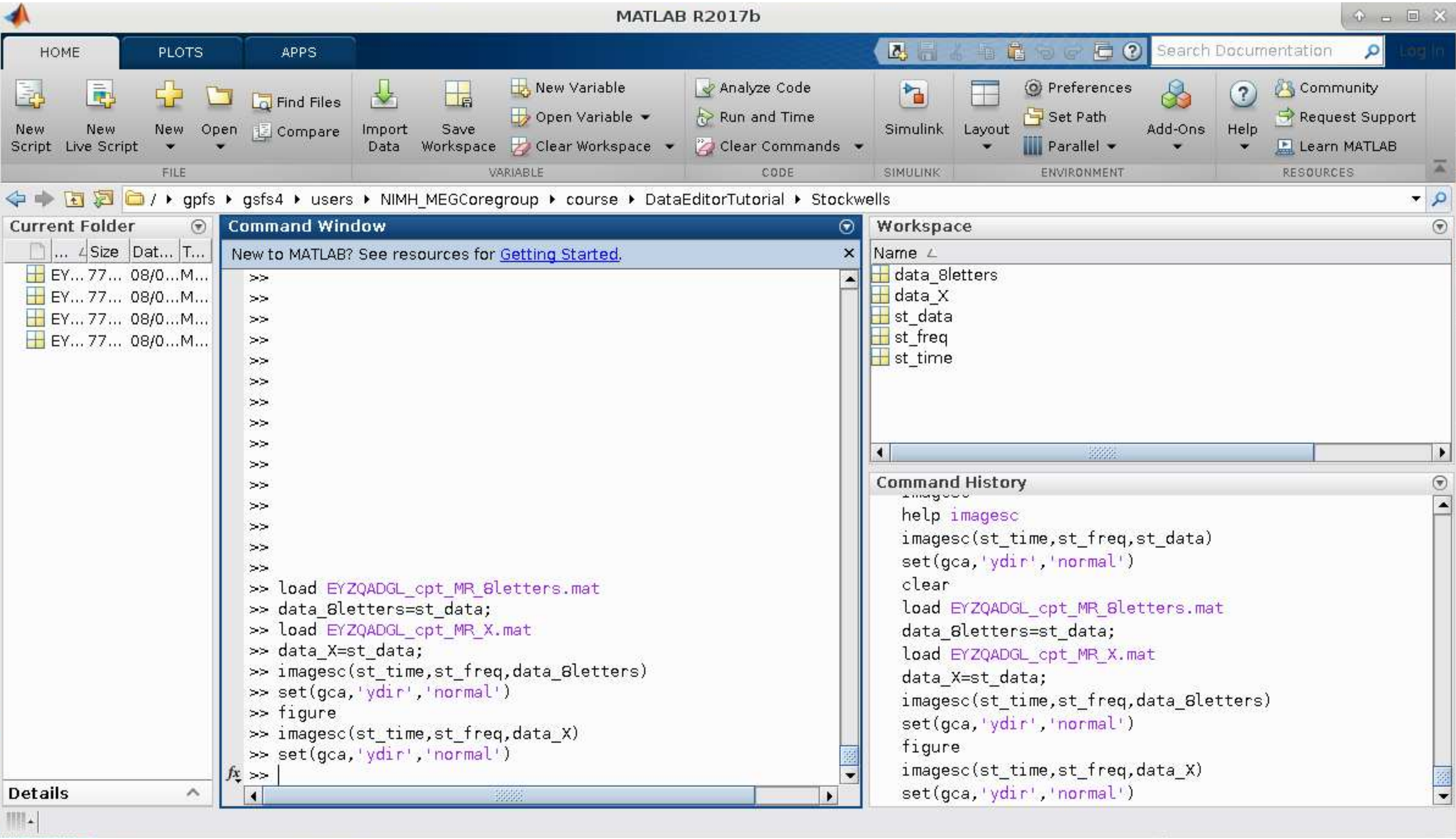
--mat matfile

StockwellDs

```
$ StockwellDs.py -d EYZQADGL_cpt_20180608_01.ds
        -m X -t “-0.3 0.7” -b “0 80” -c ML
        -n -B “0.2 0.3” –mat EYZQADGL_cpt_MR_S -v
```

Or -o to produce an afni brick
AFNI brick stockwells can be operated on using AFNI routines, or viewed using:

```
$disptfbrik.py
```



Workspace:

```
data_8letters
data_X
st_data
st_freq
st_time
```

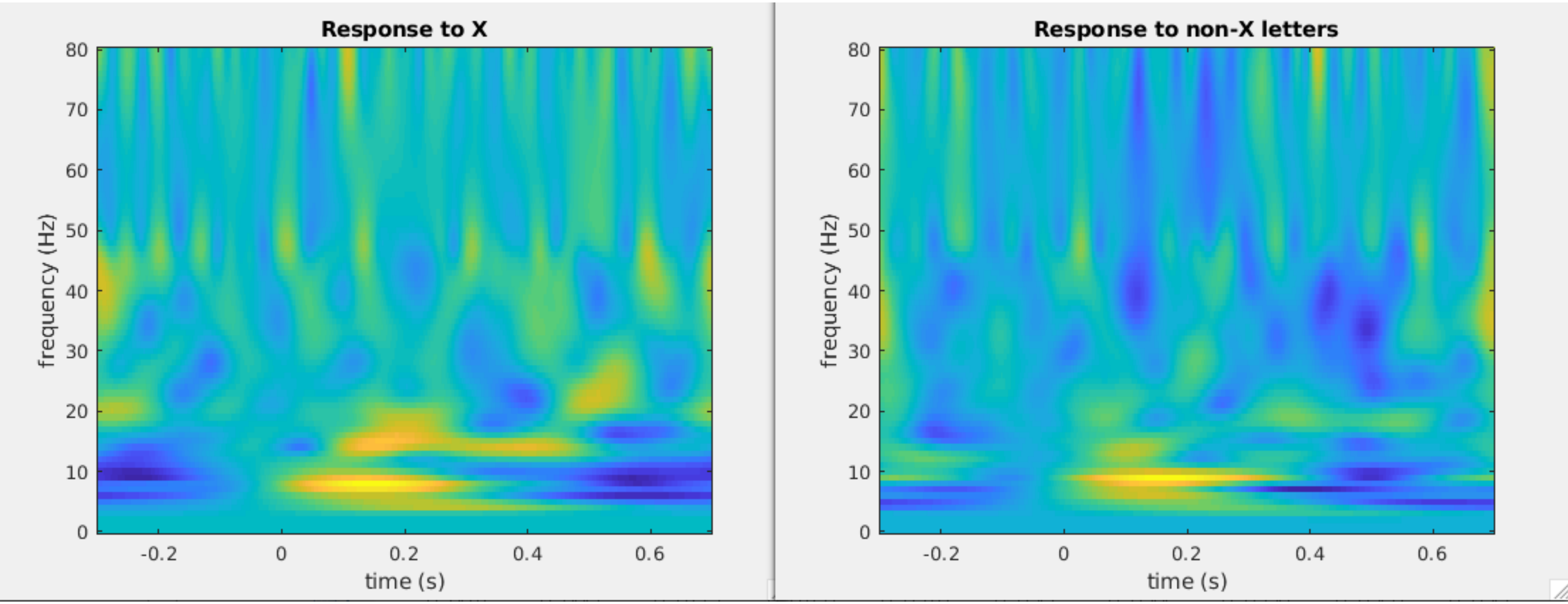
Command Window:

```
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>> load EYZQADGL_cpt_MR_8letters.mat
>> data_8letters=st_data;
>> load EYZQADGL_cpt_MR_X.mat
>> data_X=st_data;
>> imagesc(st_time,st_freq,data_8letters)
>> set(gca,'ydir','normal')
>> figure
>> imagesc(st_time,st_freq,data_X)
>> set(gca,'ydir','normal')
fx >> 
```

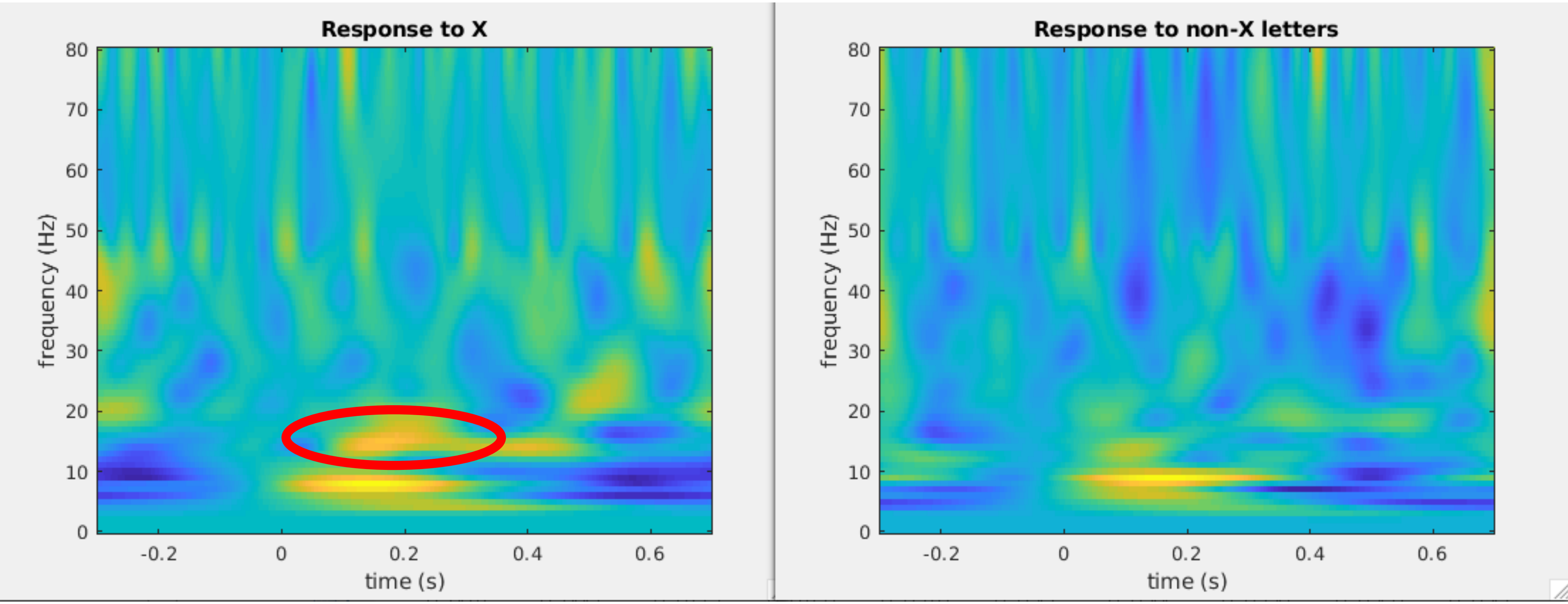
Command History:

```
imagesc
help imagesc
imagesc(st_time,st_freq,st_data)
set(gca,'ydir','normal')
clear
load EYZQADGL_cpt_MR_8letters.mat
data_8letters=st_data;
load EYZQADGL_cpt_MR_X.mat
data_X=st_data;
imagesc(st_time,st_freq,data_8letters)
set(gca,'ydir','normal')
figure
imagesc(st_time,st_freq,data_X)
set(gca,'ydir','normal')
```

Single Subject, response to X and non-X letters



Single Subject, response to X and non-X letters


Response to X
frequency (Hz)
time (s)
Response to non-X letters
frequency (Hz)
time (s)