Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

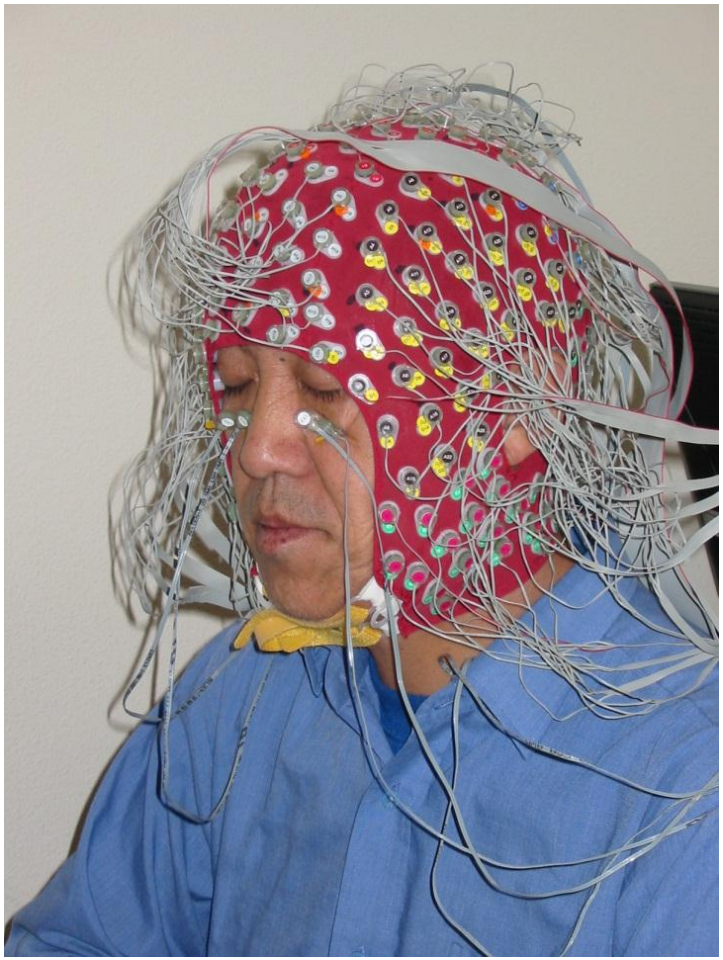
Examine raw data

Reject bad channels

Reject large artifact time points

Run ICA

Dense-array EEG



Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

Examine raw data

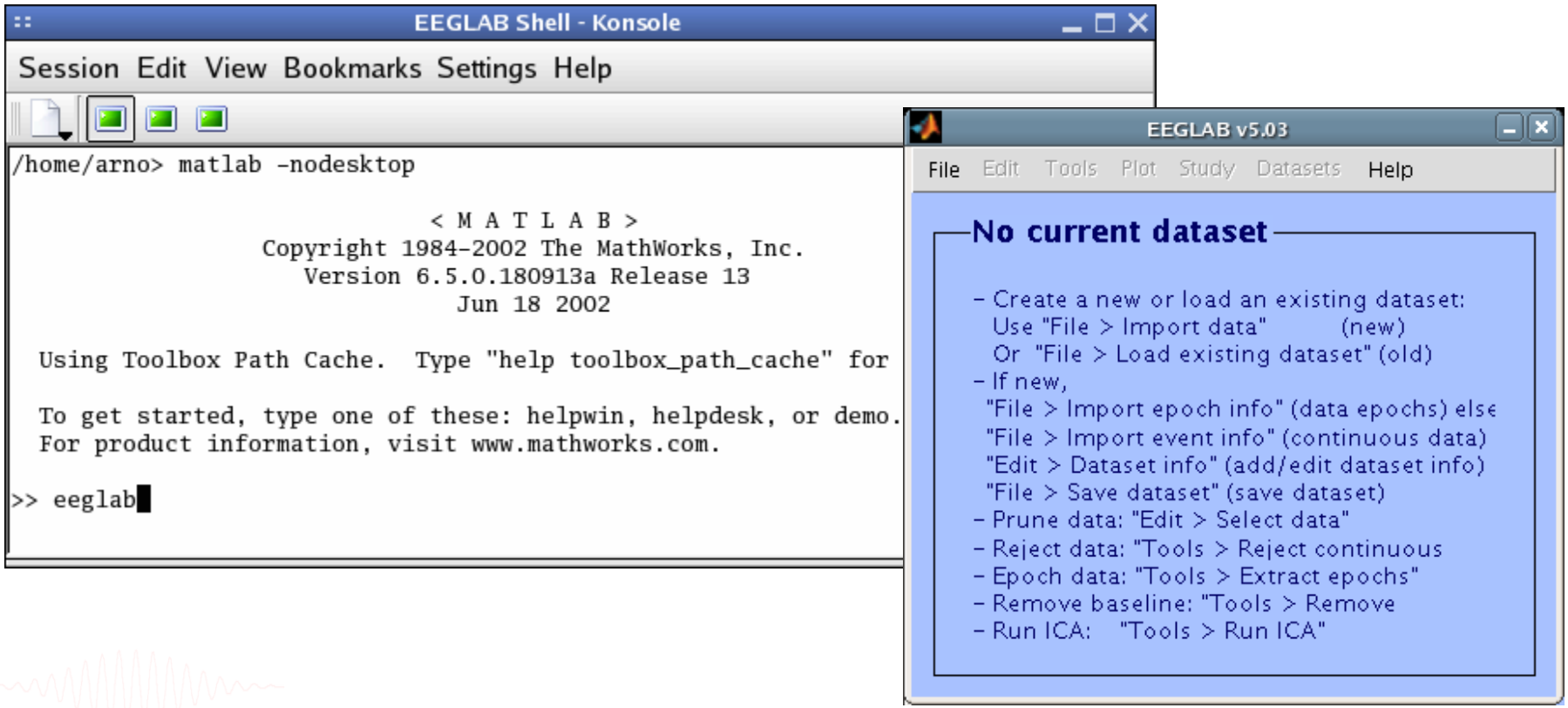
Reject bad channels

Reject large artifact time points

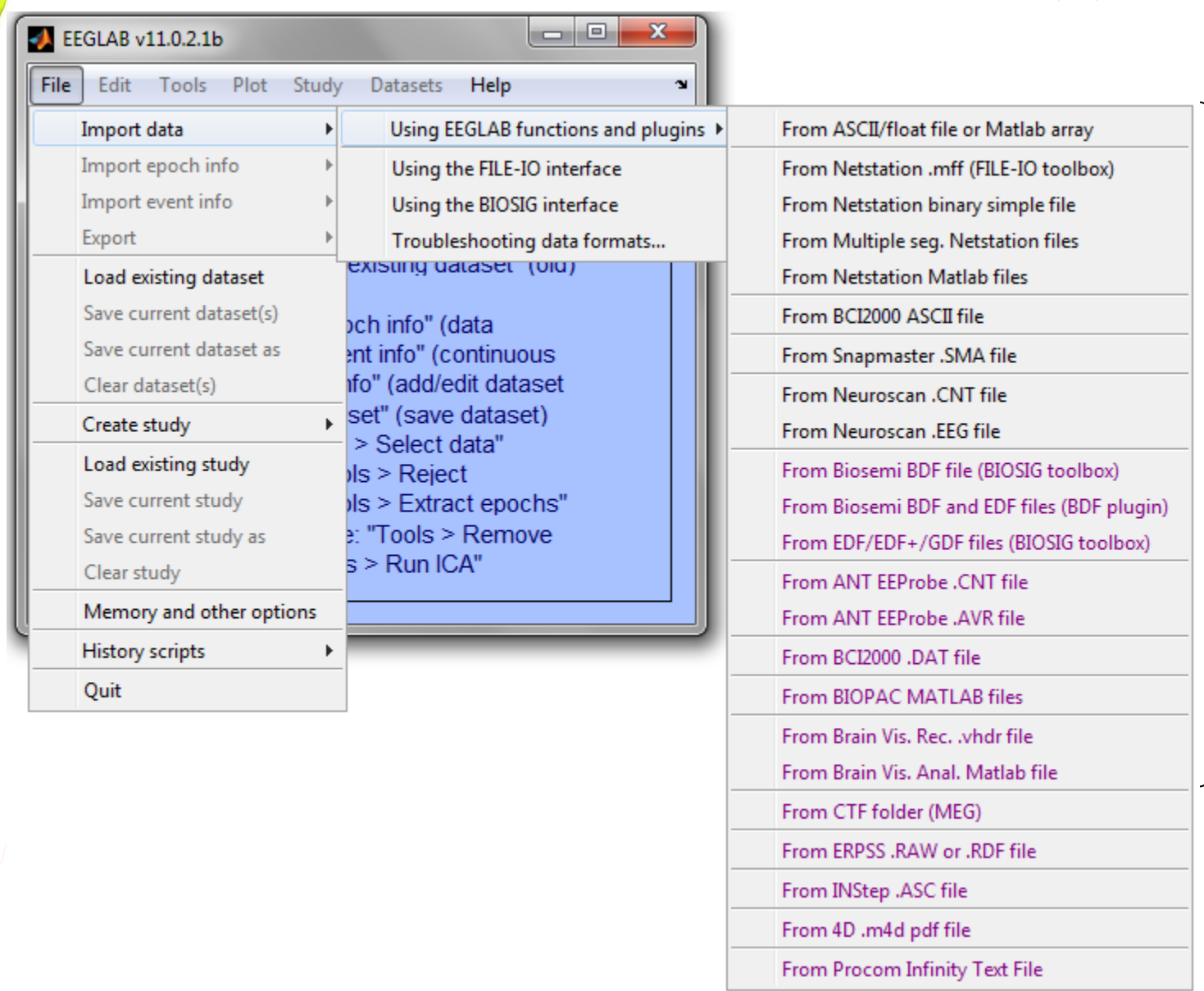
Run ICA

EEGLAB Matlab toolbox

main graphic interface

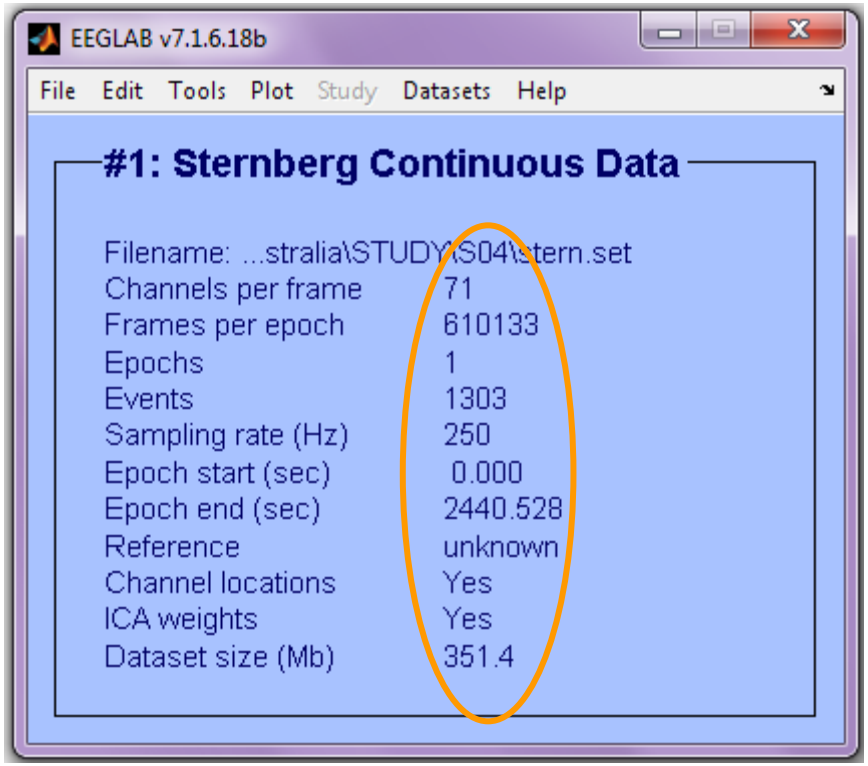


Importing a dataset



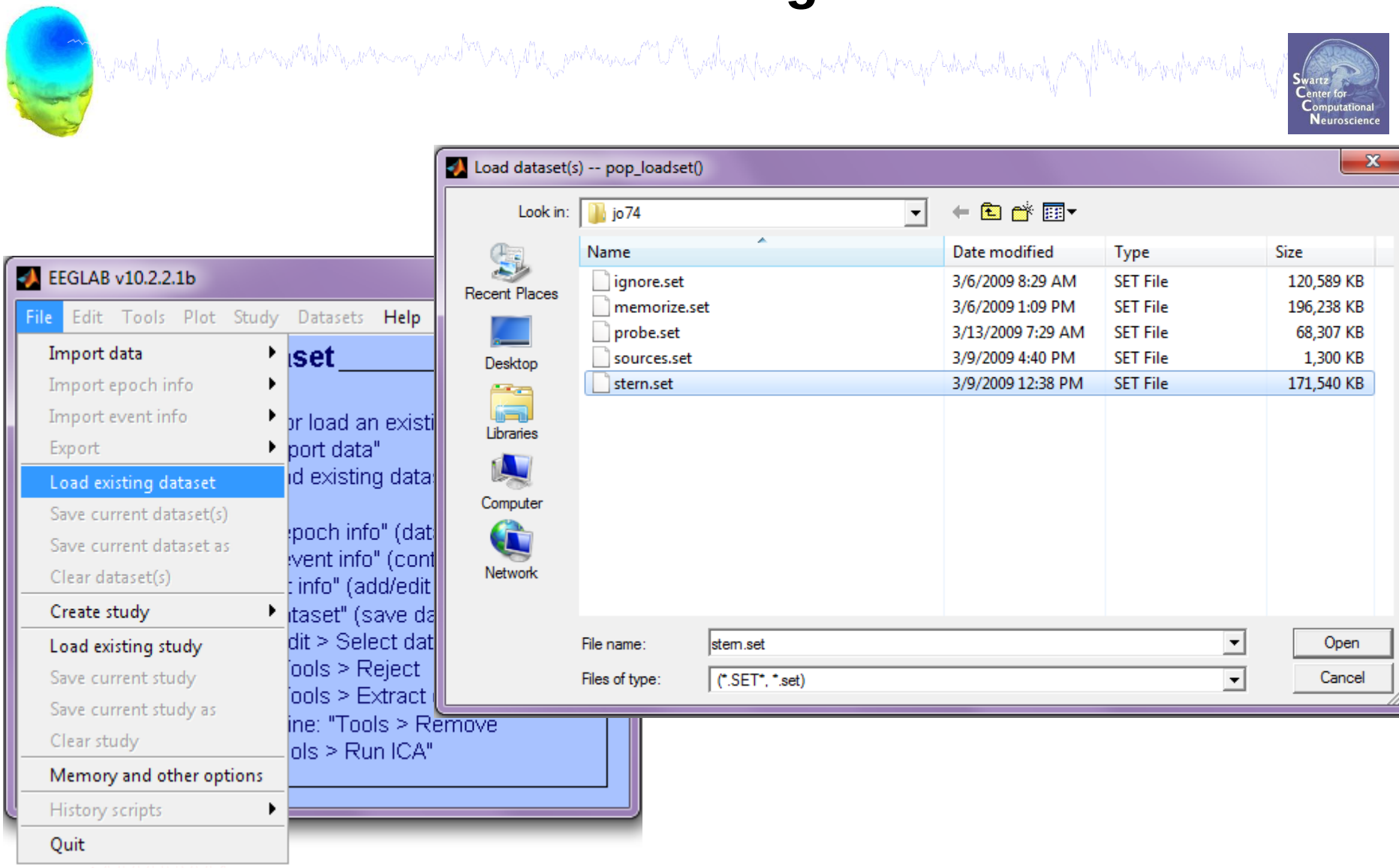
EEGLAB supports many different raw data formats

Imported EEG data



EEGLAB GUI displays dataset basics

Load an existing dataset



Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

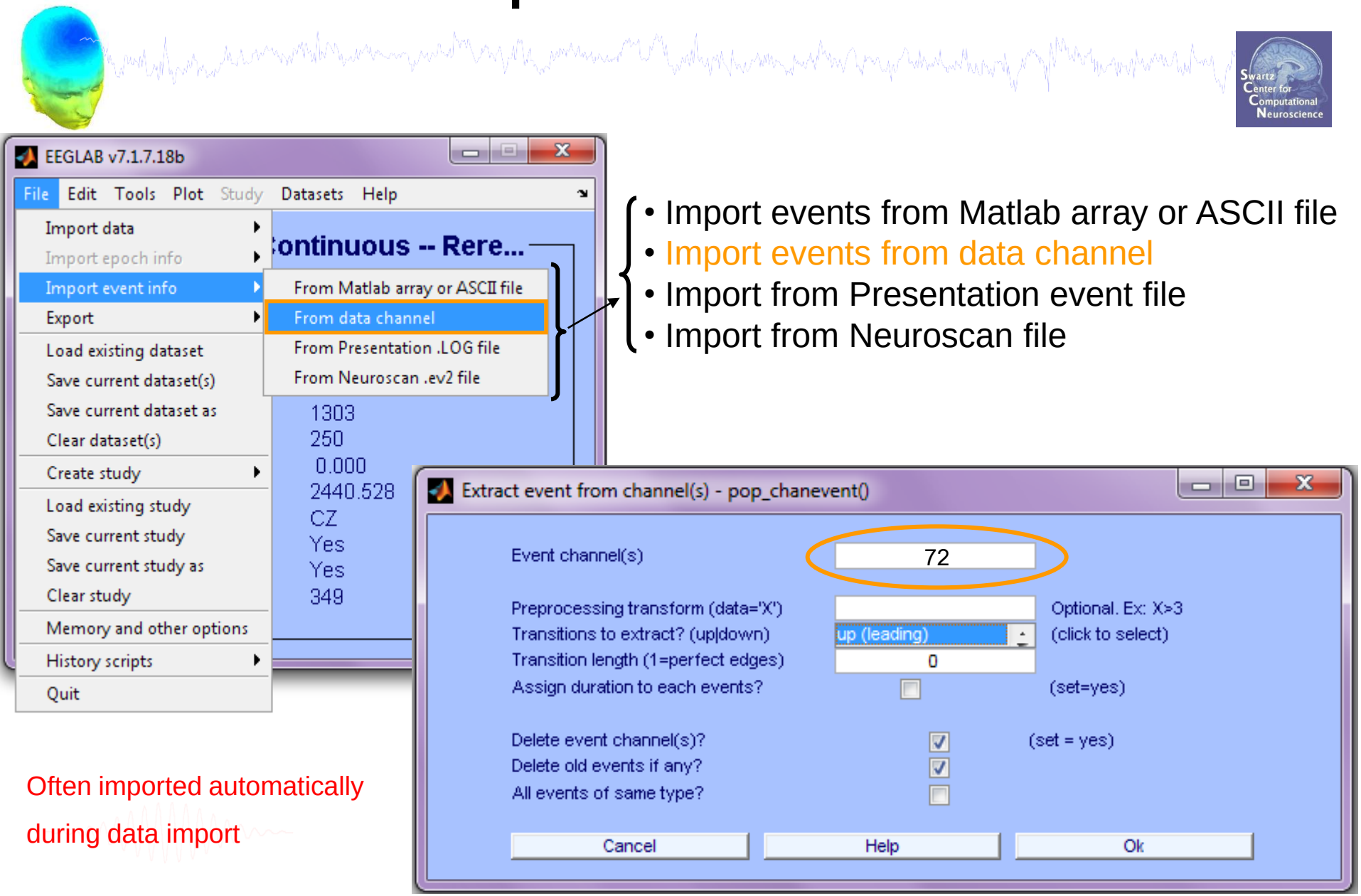
Examine raw data

Reject bad channels

Reject large artifact time points

Run ICA

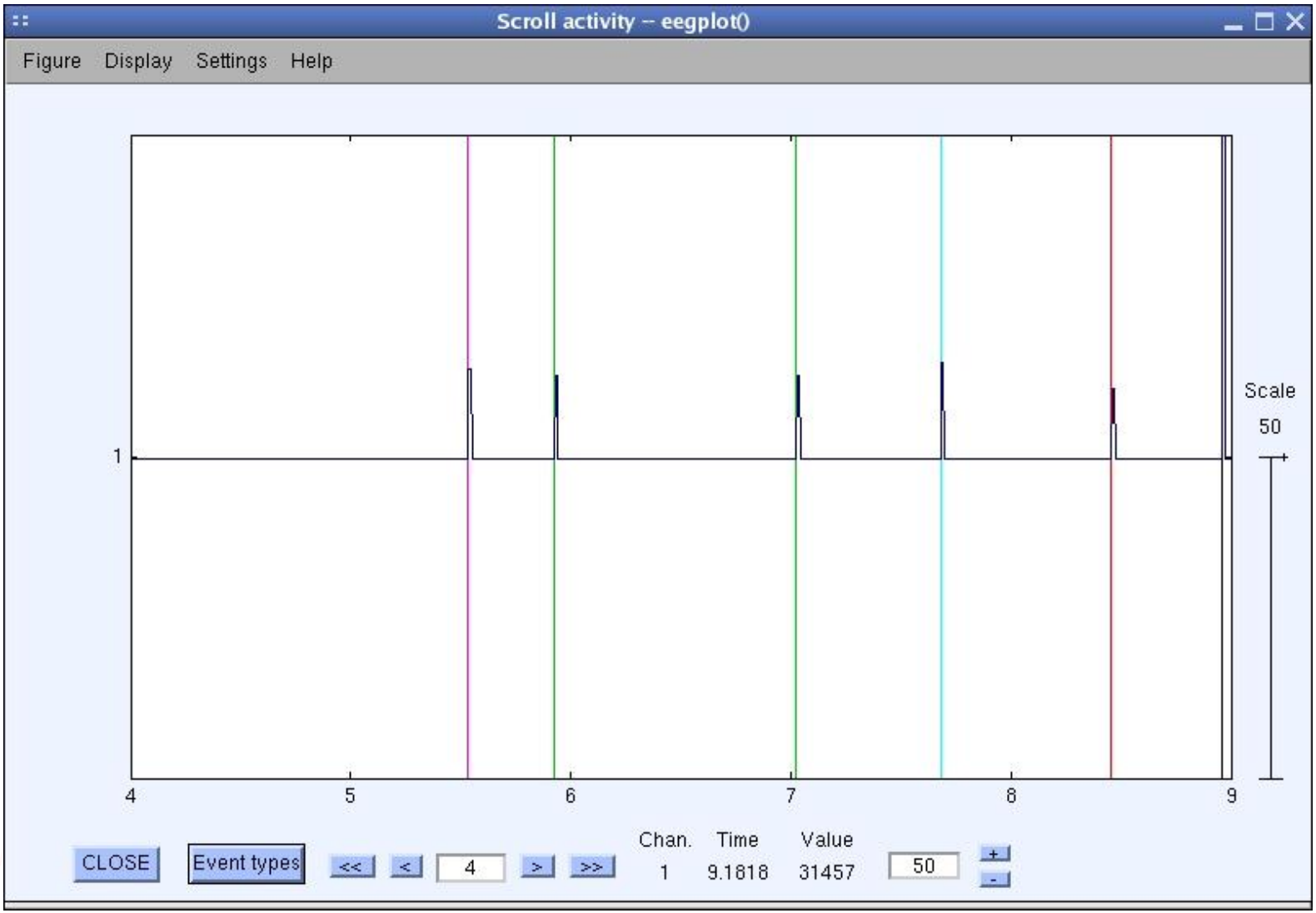
Import data events



- Import events from Matlab array or ASCII file
- Import events from data channel
- Import from Presentation event file
- Import from Neuroscan file

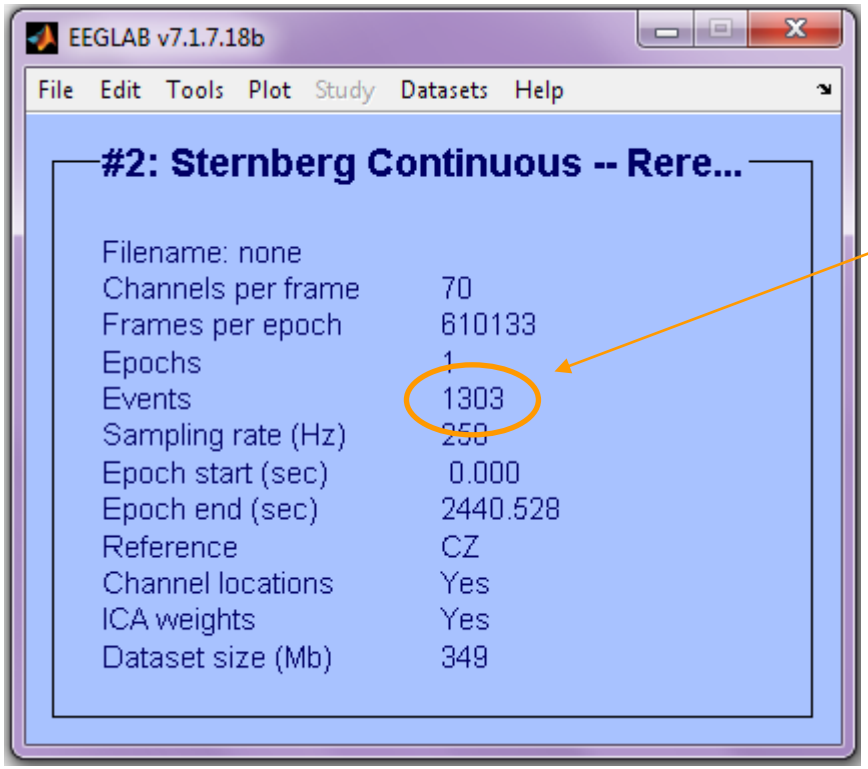
Often imported automatically during data import

Appearance of an event channel in raw data



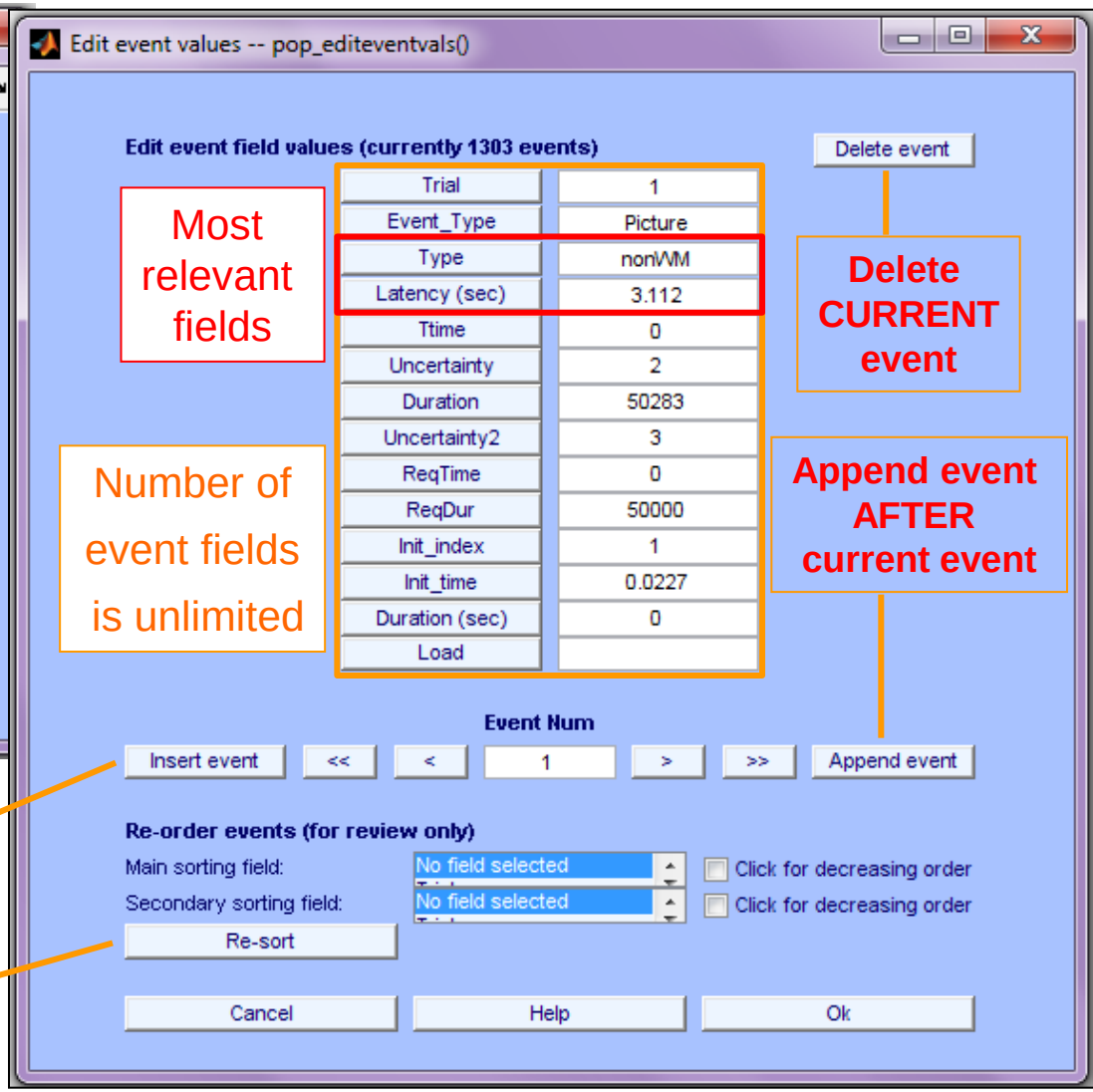
Imported data events

```
>> EEG.event

ans =

1x1303 struct array with fields:
    Trial
    Event_Type
    type
    latency
    TTime
    Uncertainty
    Duration
    Uncertainty2
    ReqTime
    ReqDur
    init_index
    init_time
    urevent
    duration
    load
    rt
```



If event import was successful, you will see an appropriate number here

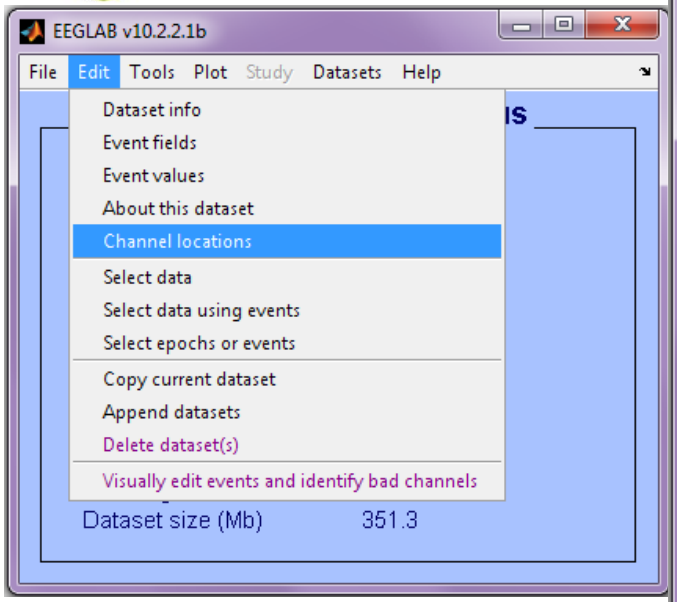
Review event values

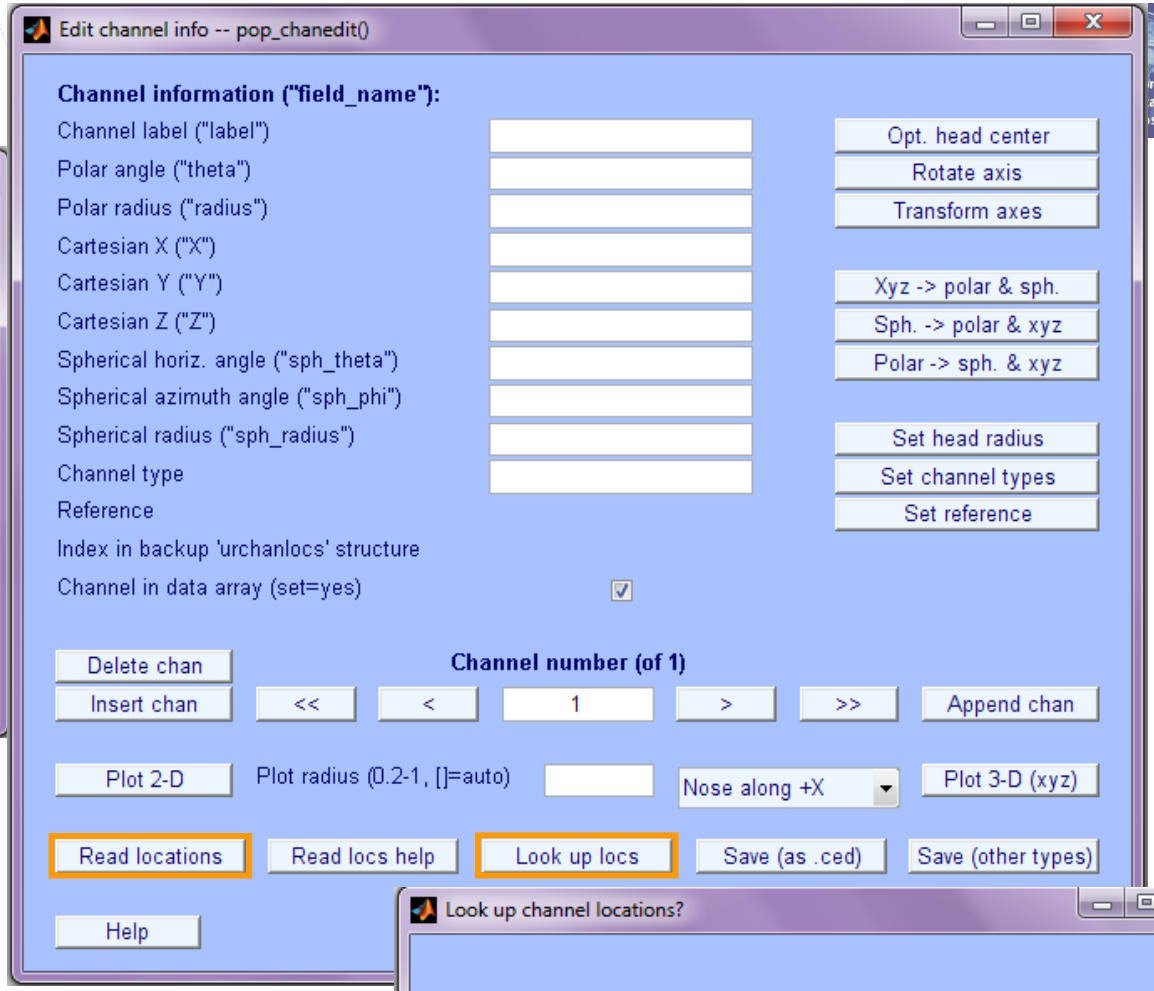


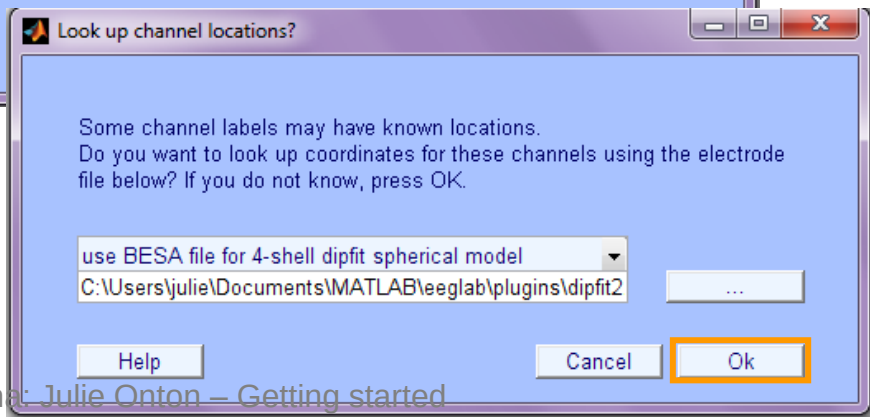
Insert event BEFORE current event

To resort: first select Main sorting field

Import channel locations

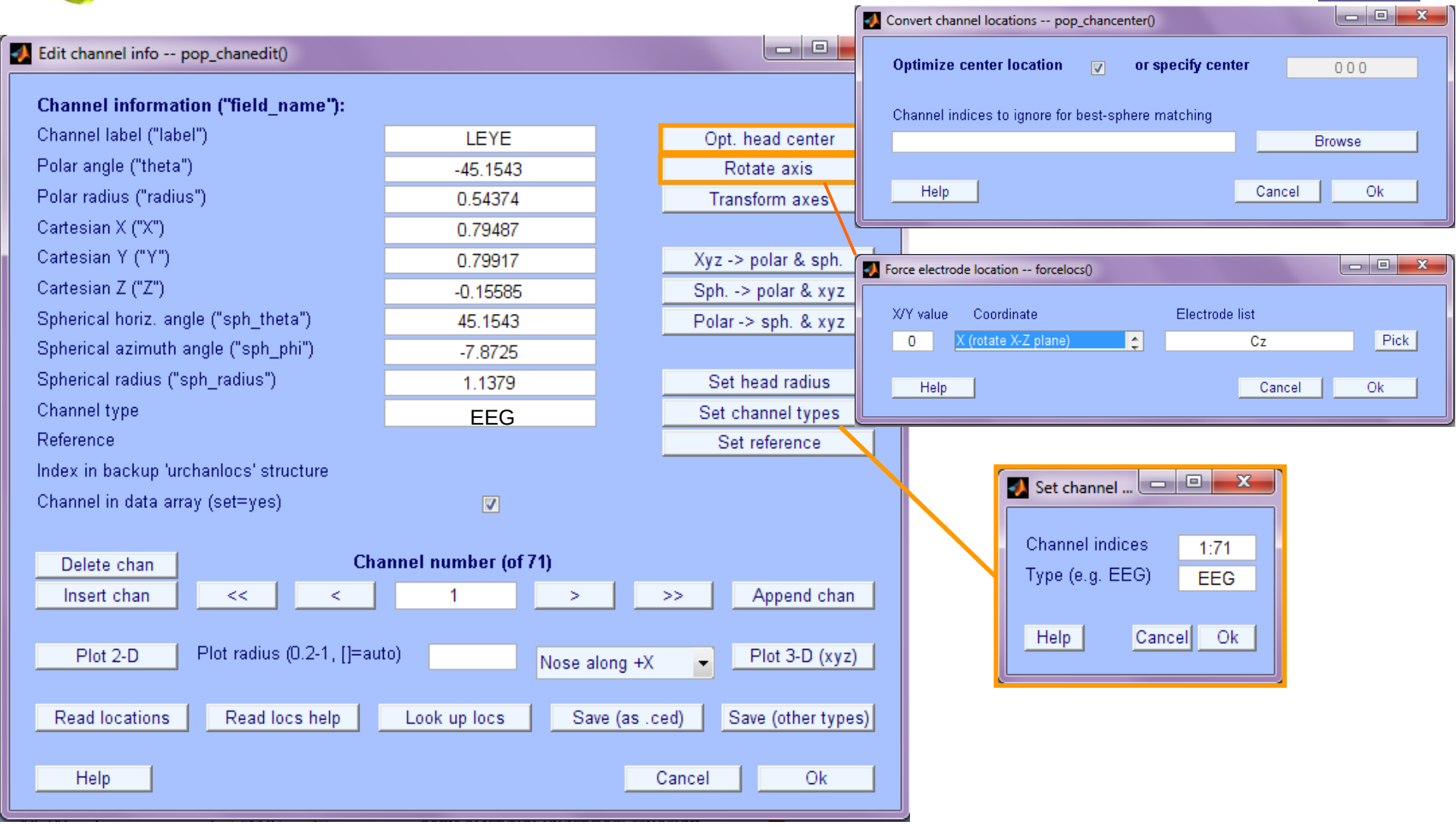






Several file formats supported (Polhemus, BESA, El Guide …)

Import channel locations



Edit channel info -- pop_chanedit()

Channel information ("field_name"):

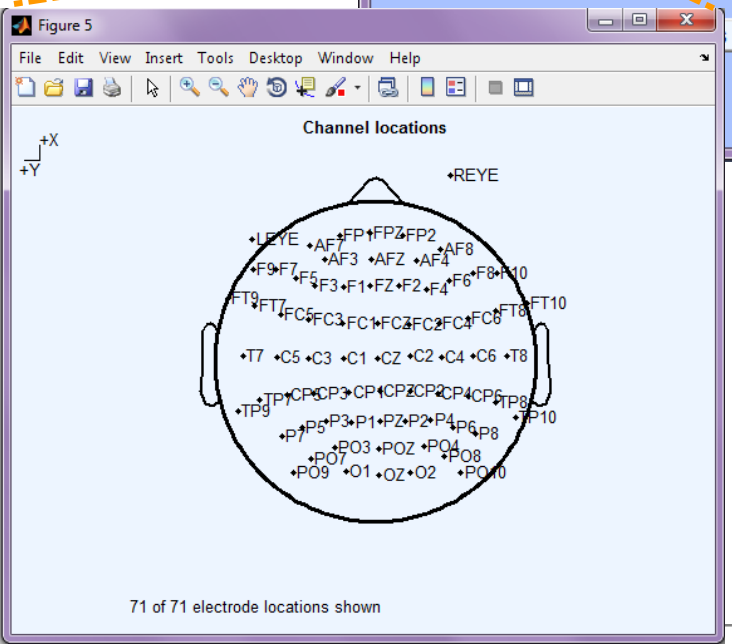
Channel label ("label")	LEYE	Opt. head center
Polar angle ("theta")	-45.1543	Rotate axis
Polar radius ("radius")	0.54374	Transform axes
Cartesian X ("X")	0.79487	
Cartesian Y ("Y")	0.79917	Xyz -> polar & sph.
Cartesian Z ("Z")	-0.15585	Sph. -> polar & xyz
Spherical horiz. angle ("sph_theta")	45.1543	Polar -> sph. & xyz
Spherical azimuth angle ("sph_phi")	-7.8725	
Spherical radius ("sph_radius")	1.1379	Set head radius
Channel type		Set channel types
Reference		Set reference
Index in backup 'urchanlocs' structure		
Channel in data array (set=yes)	☑	

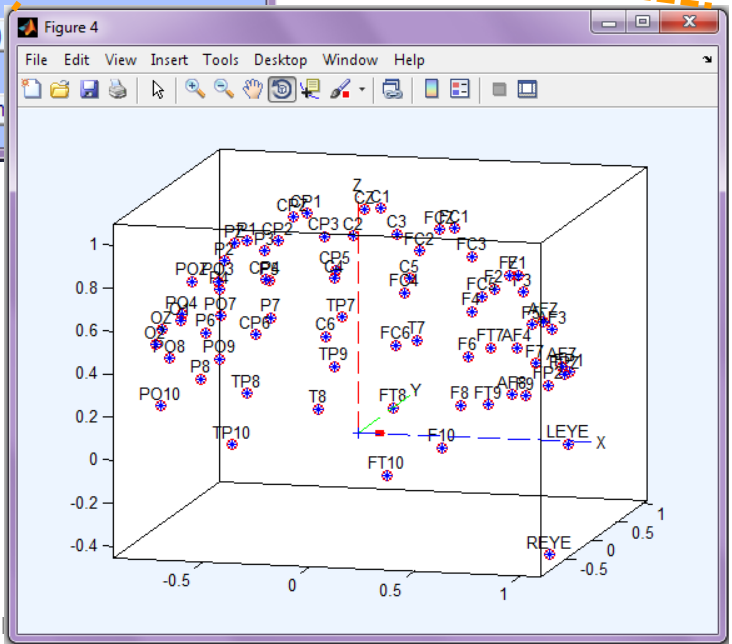
Delete chan | Insert chan | << | < | 1 | > | >> | Append chan

Channel number (of 71)

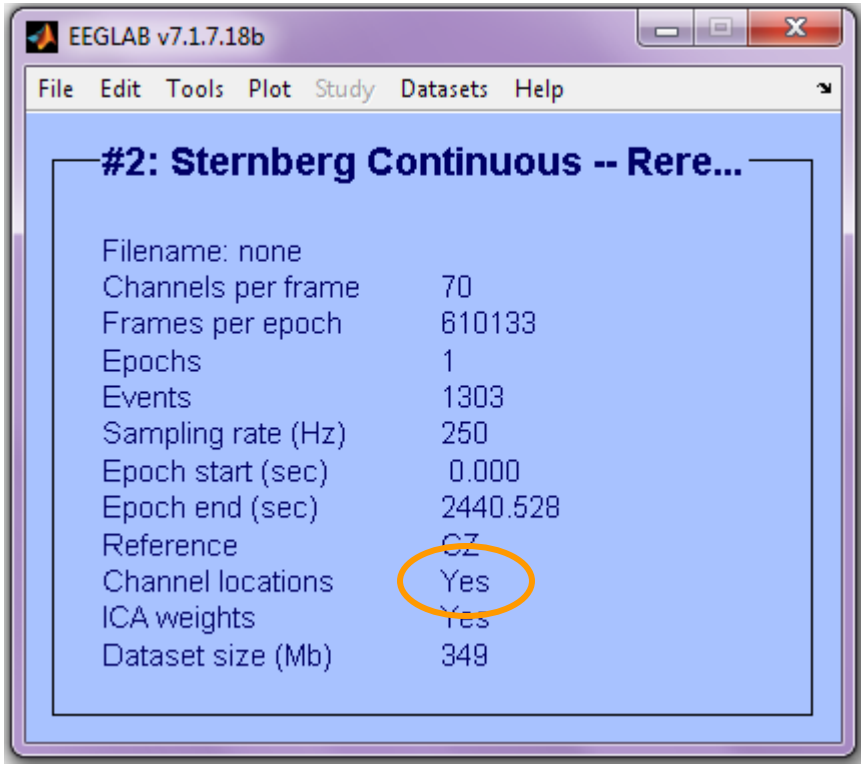
Plot 2-D | Plot radius (0.2-1, []=auto) | Nose along +X | Plot 3-D (xyz)

help | Look up locs | Save (as .ced) | Can





Imported channel locations



Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

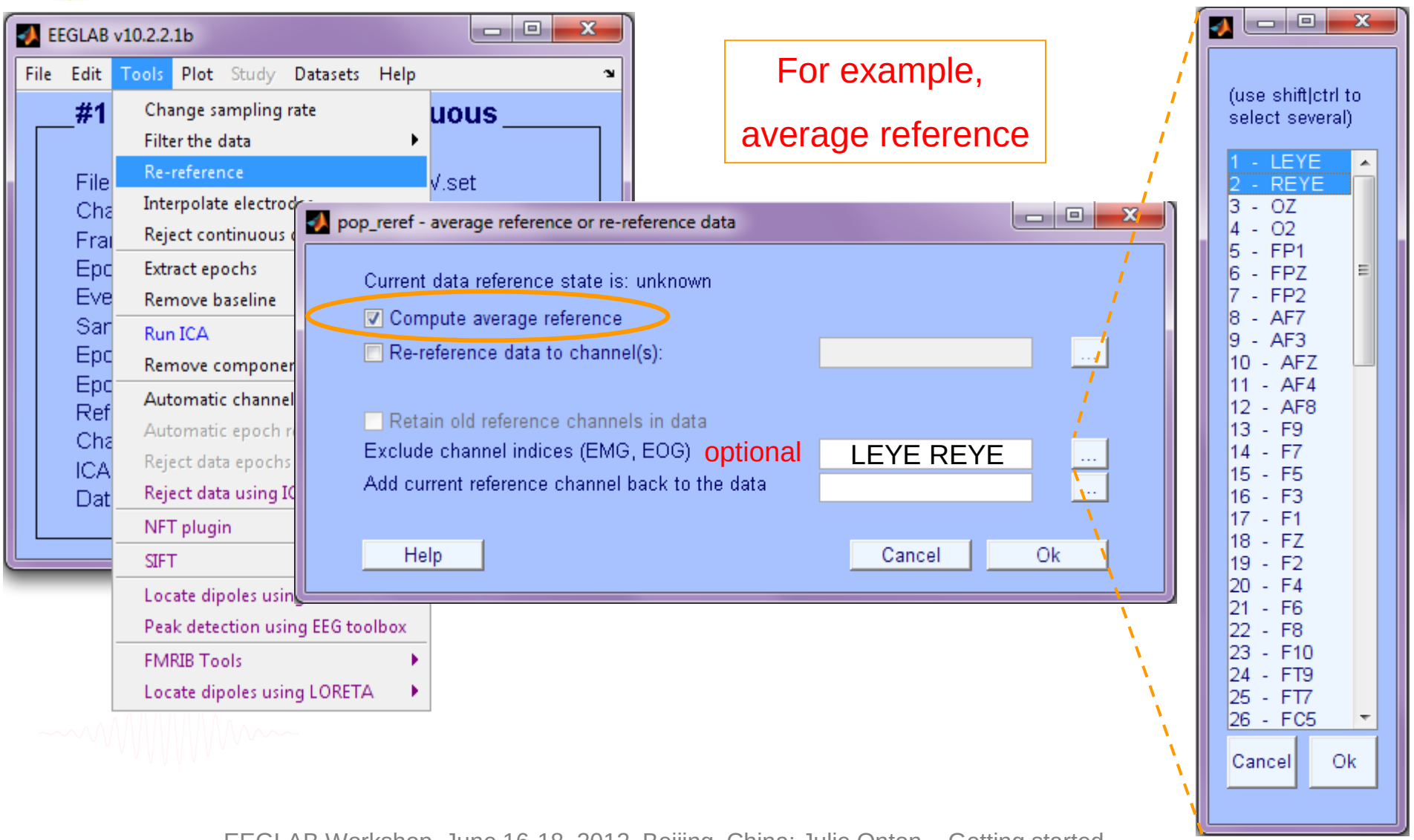
Examine raw data

Reject bad channels

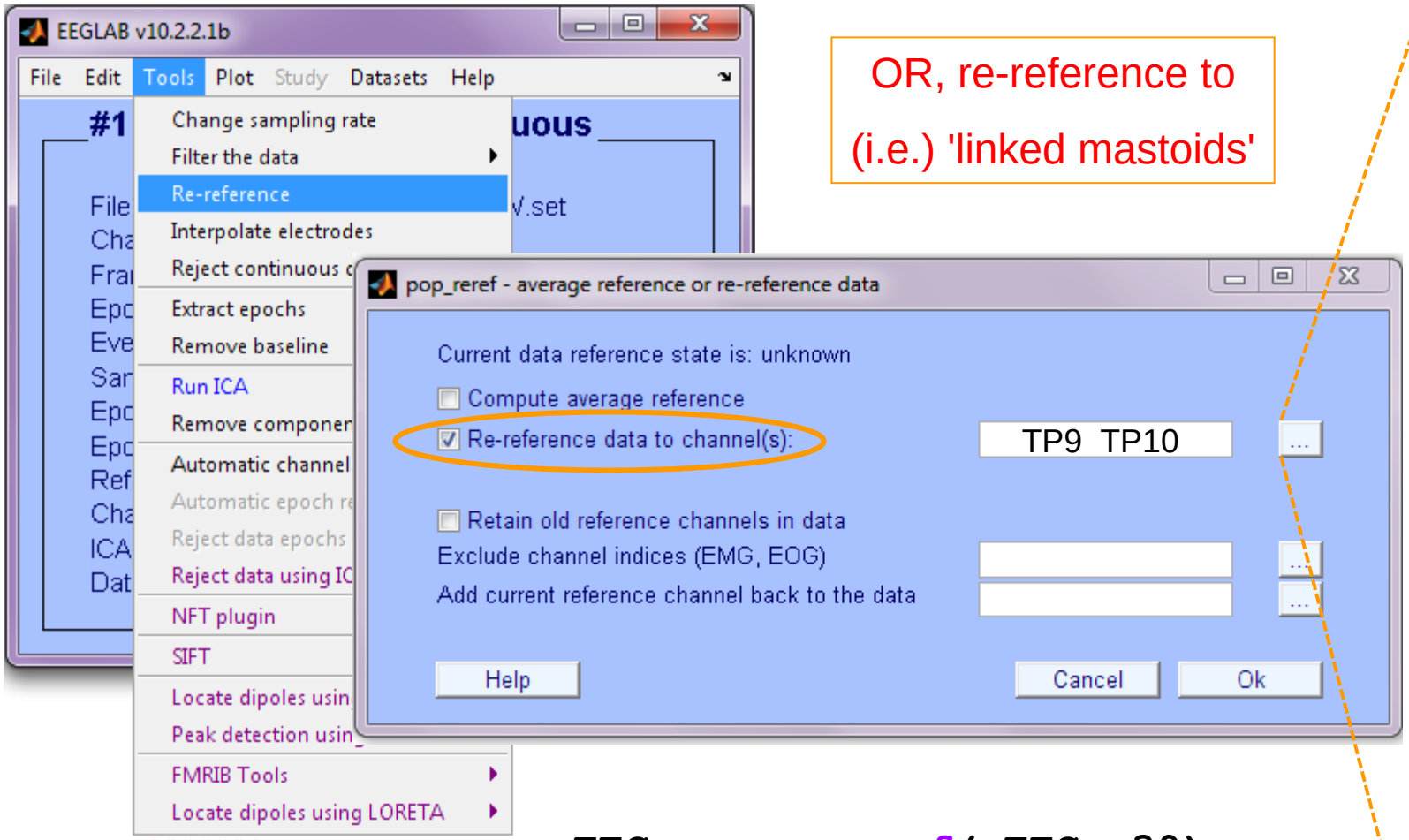
Reject large artifact time points

Run ICA

Re-reference data (if necessary/desired)

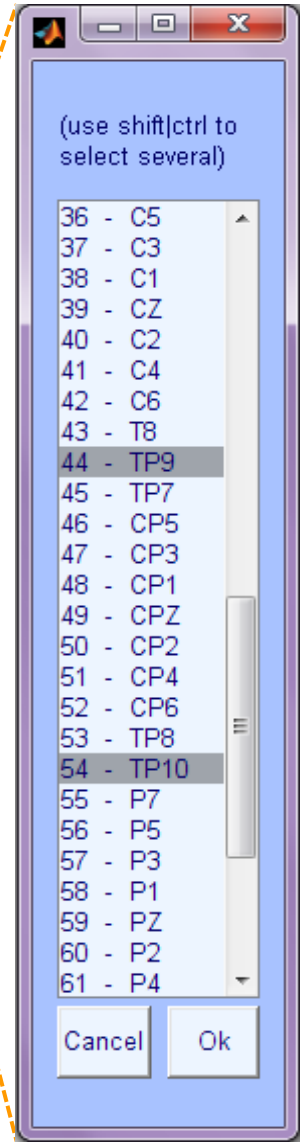


Re-reference data (if necessary/desired)



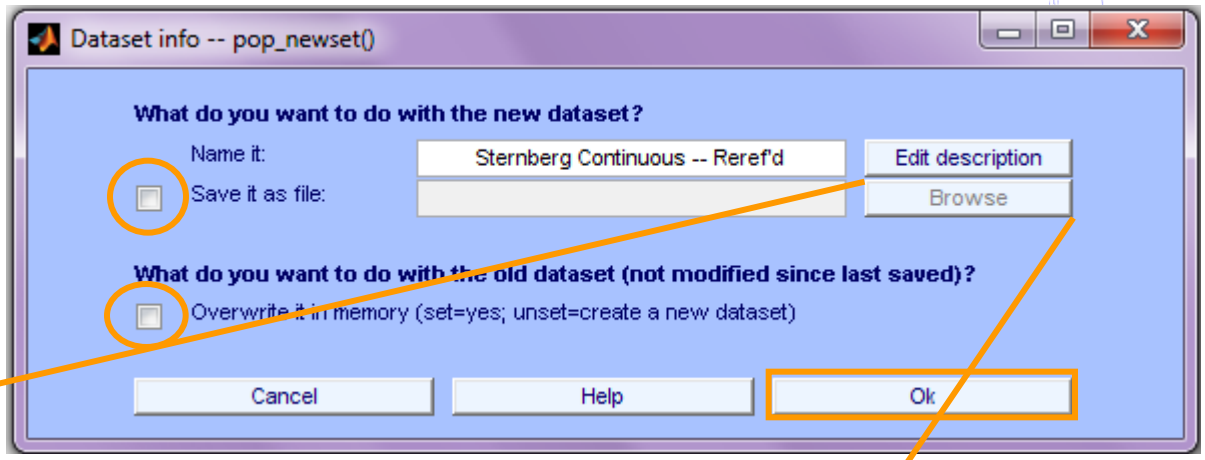
OR, re-reference to

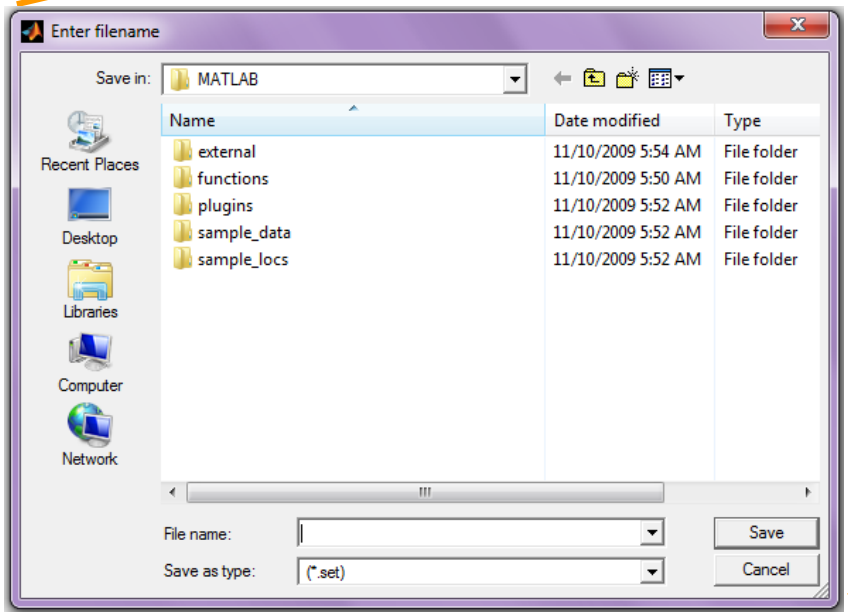
(i.e.) 'linked mastoids'

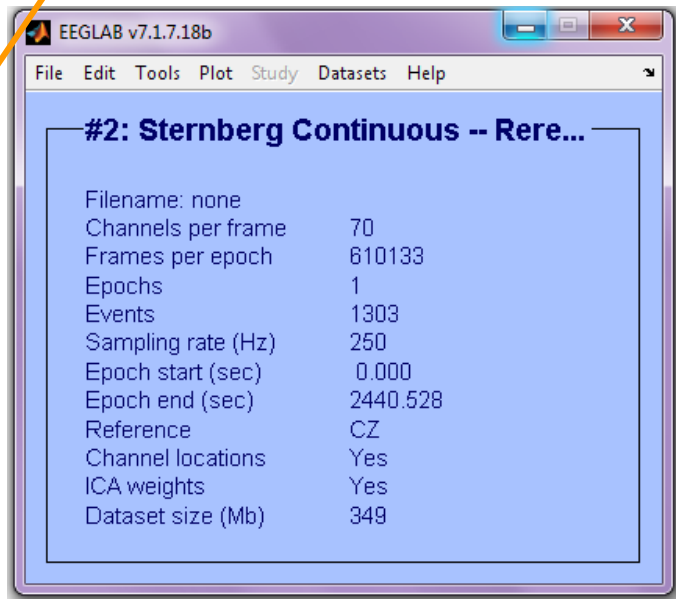



```
EEG = pop_reref( EEG, 39);
```

Save new dataset, keep old one

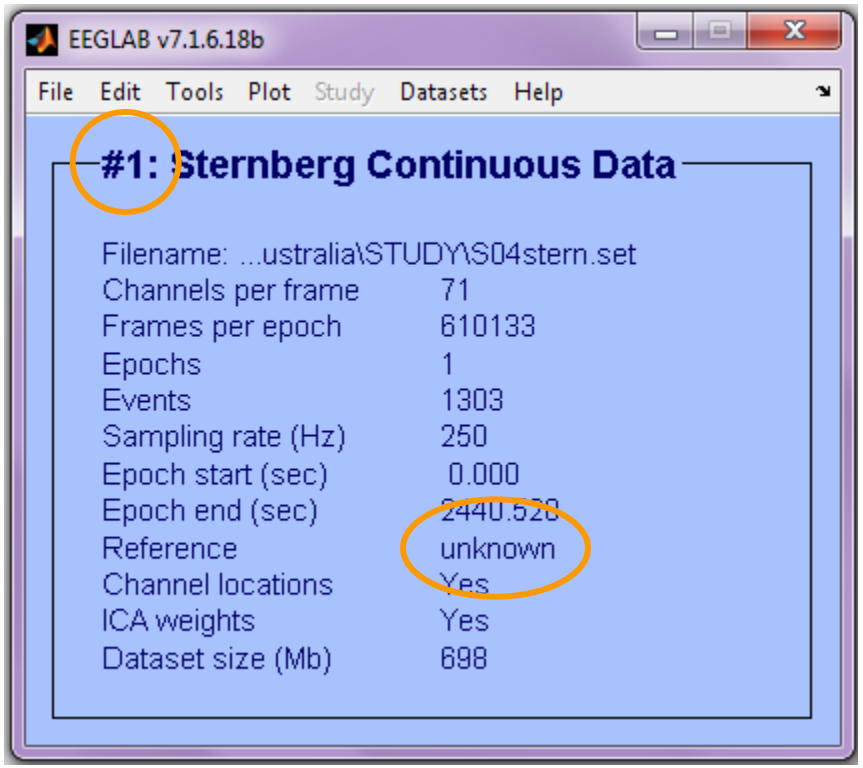


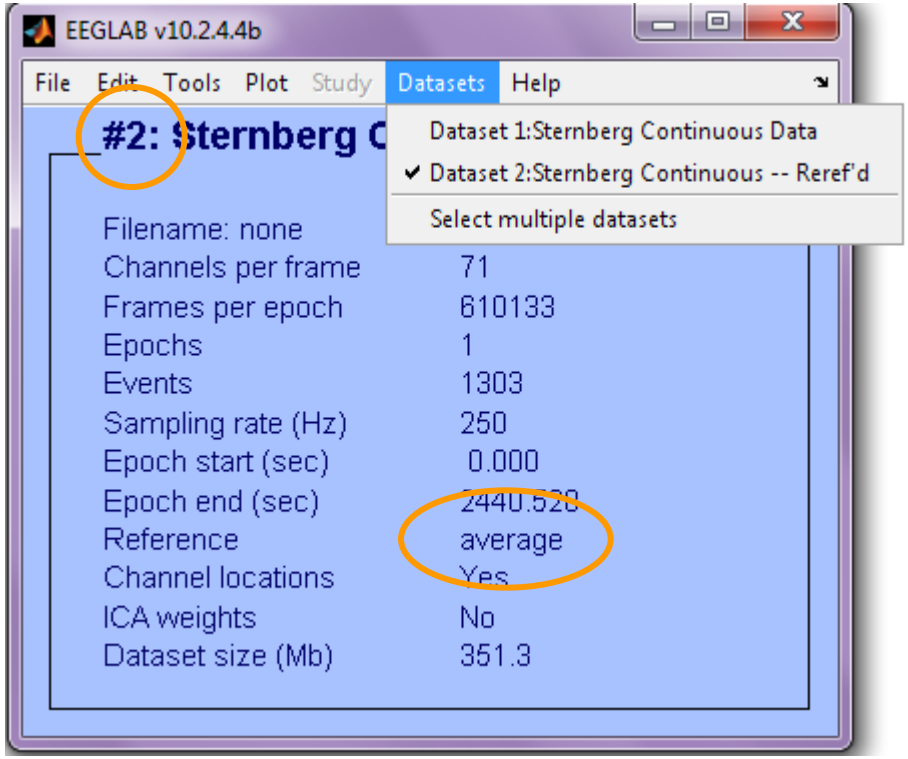





```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG,EEG, 1, 'setname',...
'Sternberg Continuous -- Reref''d');
```

Multiple active datasets (ALLEEG)





Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

Examine raw data

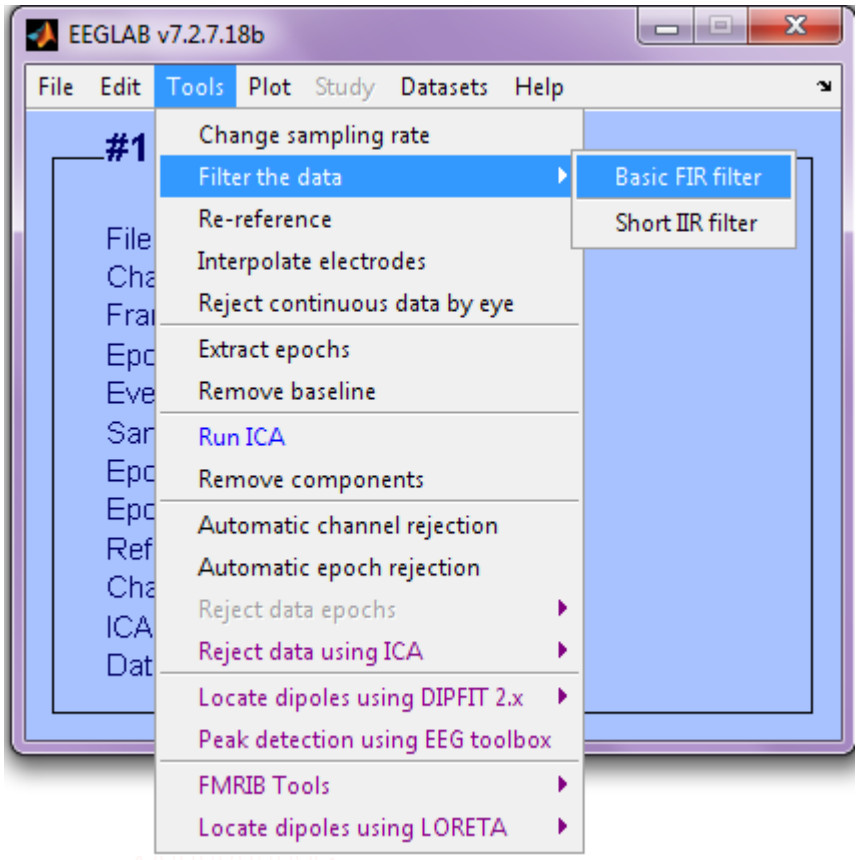
Reject bad channels

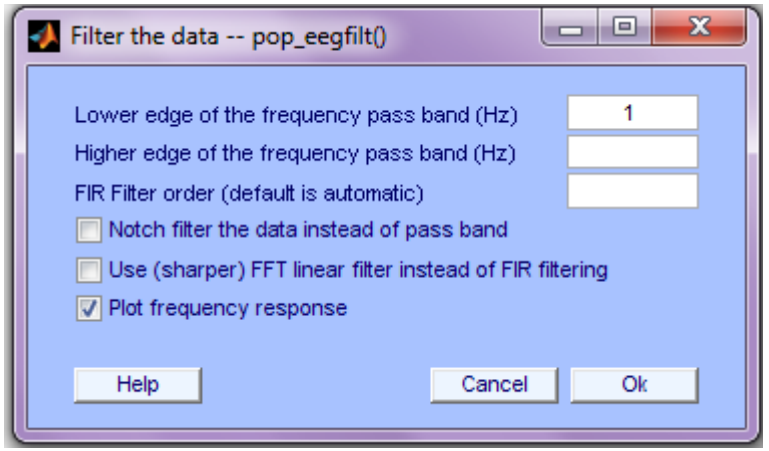
Reject large artifact time points

Run ICA

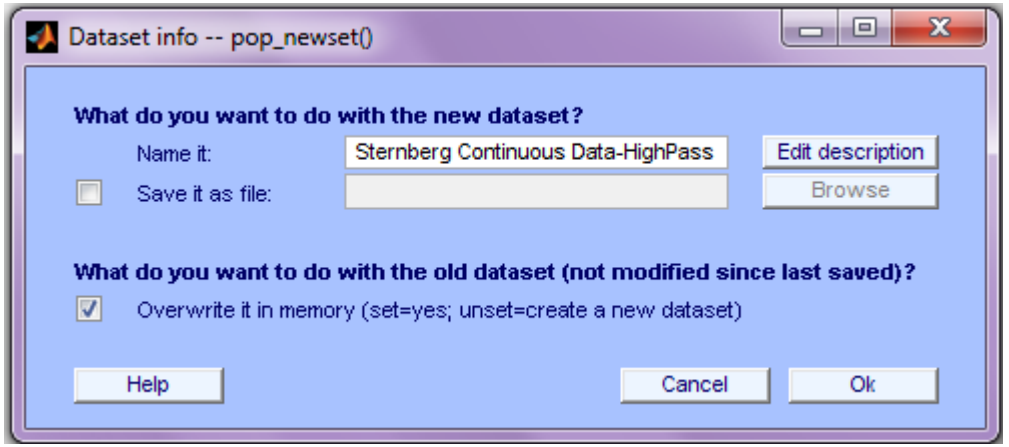
Filter the data (if necessary/desired)

Lower cut off frequencies require longer stretches of continuous data





High-pass needed for ICA



Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

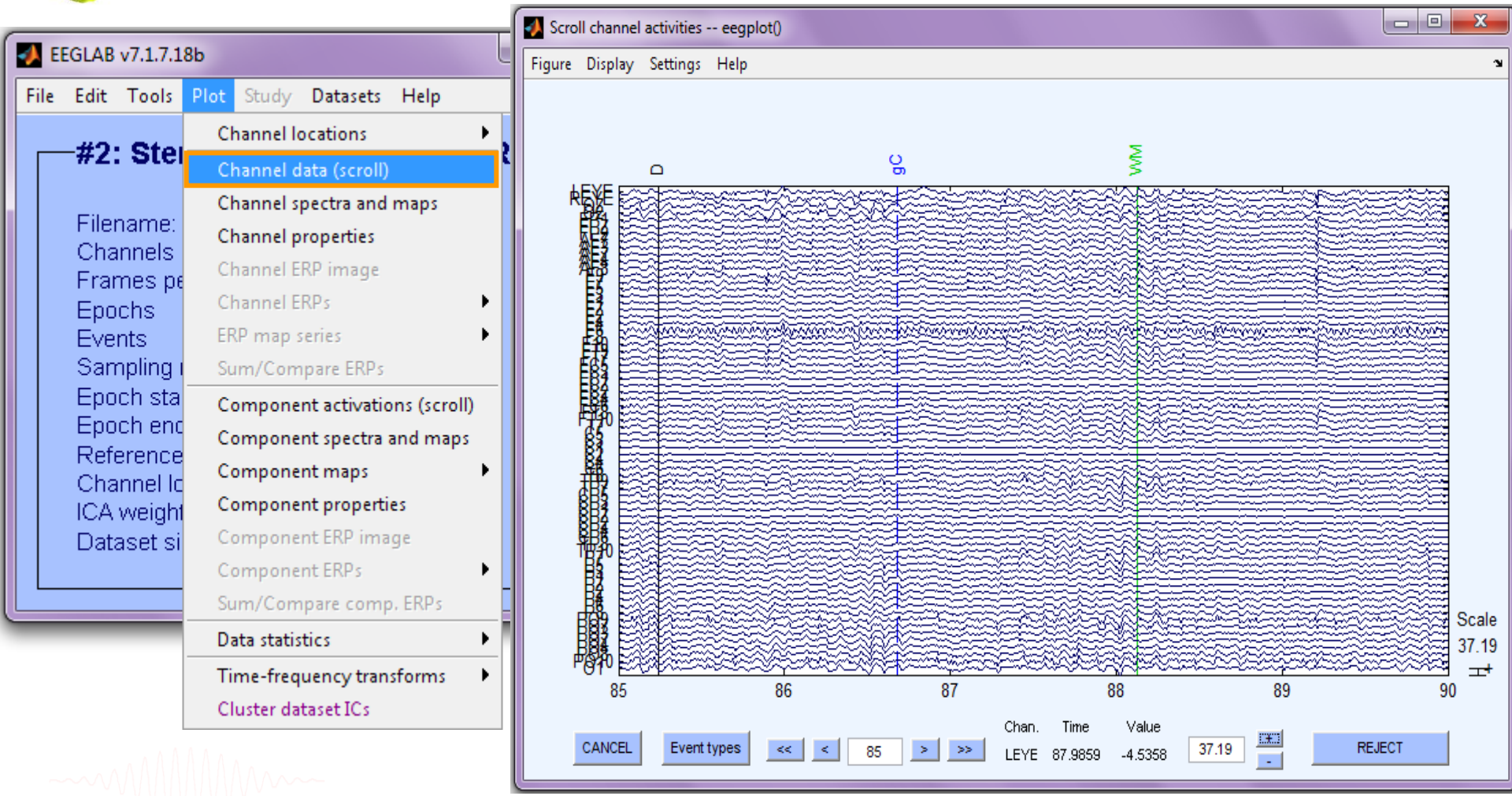
Examine raw data

Reject bad channels

Reject large artifact time points

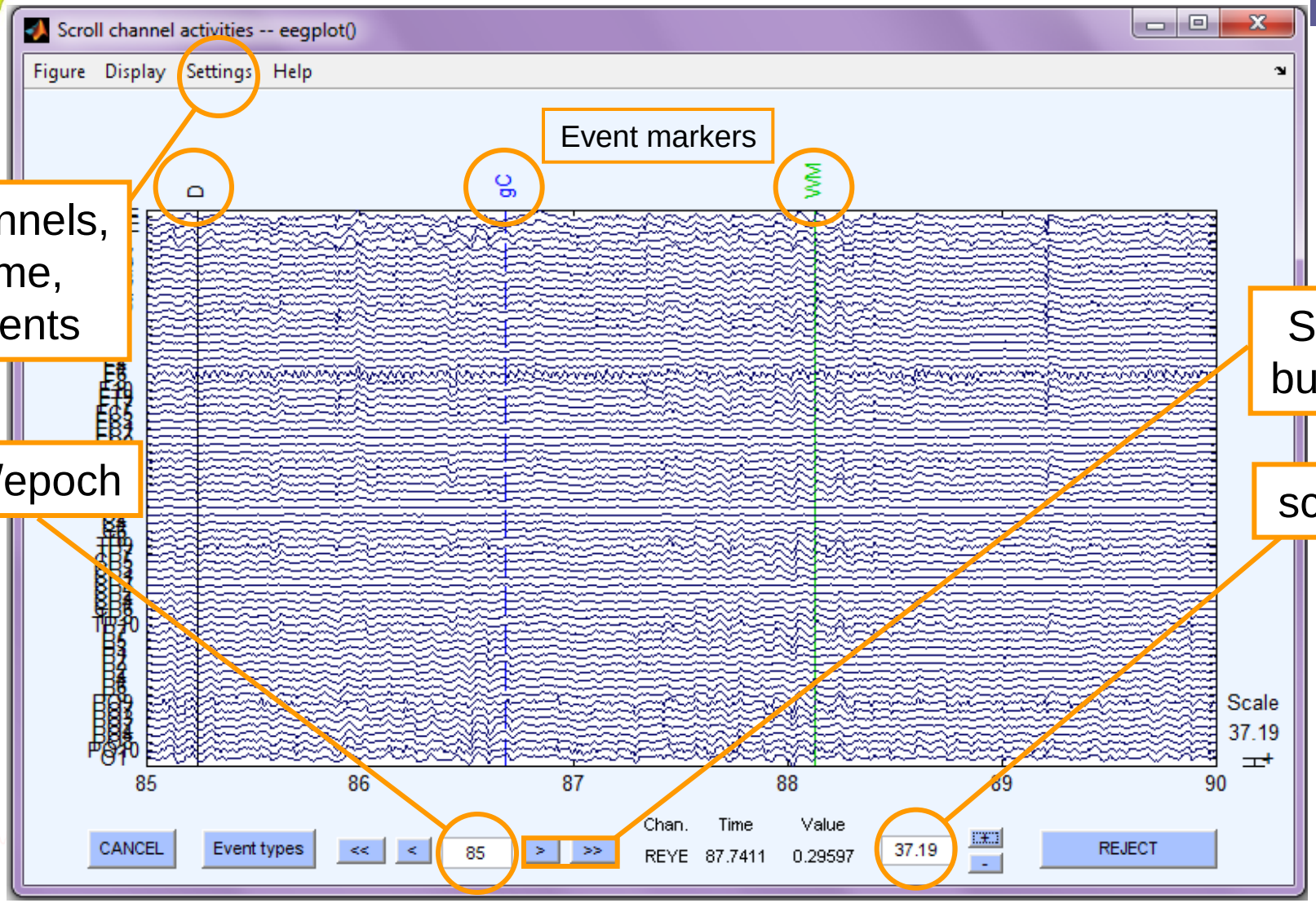
Run ICA

Scroll channel data




```
>> pop_eegplot(EEG,1,1,1);
```

Scroll channel data



Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

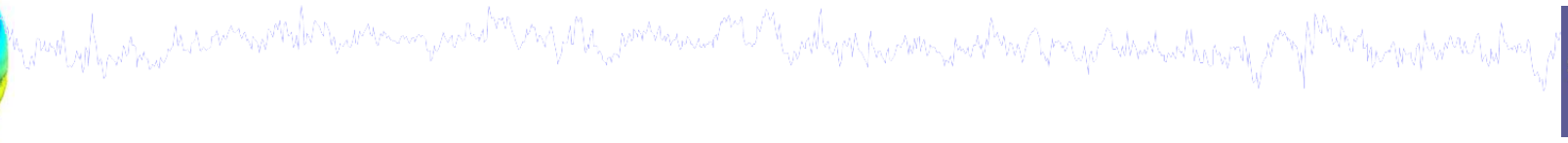
Examine raw data

Reject bad channels

Reject large artifact time points

Run ICA

Remove channel



Swartz Center for Computational Neuroscience

Scroll channel activities -- eegplot()

Figure Display Settings Help

1) Identify bad channel

gJ gC

LEYE REYE O2 EB1 EB7 EB2 AF3 AF7 AF4 AF8 E8 E7 E17 E6 E1 E5 E9 E10 FT9 FT7 FC5 FC1 FC6 FT8 FT10 T7 C5 C1 C2 C6 T8 TP9 CP5 CP1 CP2 CP6 TP10 P7 P3 P1 P2 P4 P8 PO9 PO7 PO3 PO4 PO8 PO10 O1

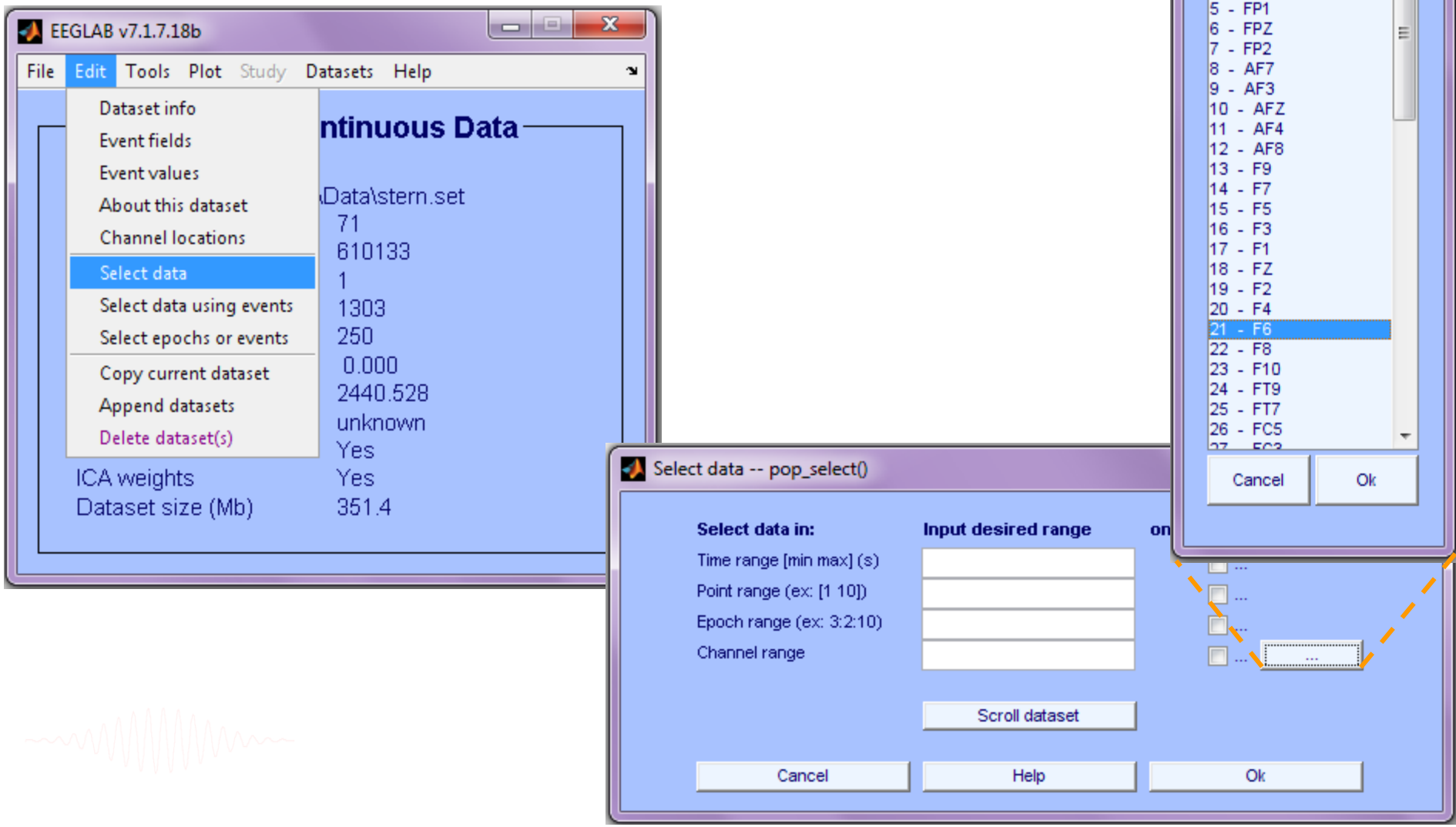
449 450 451 452 453 454

Scale 35

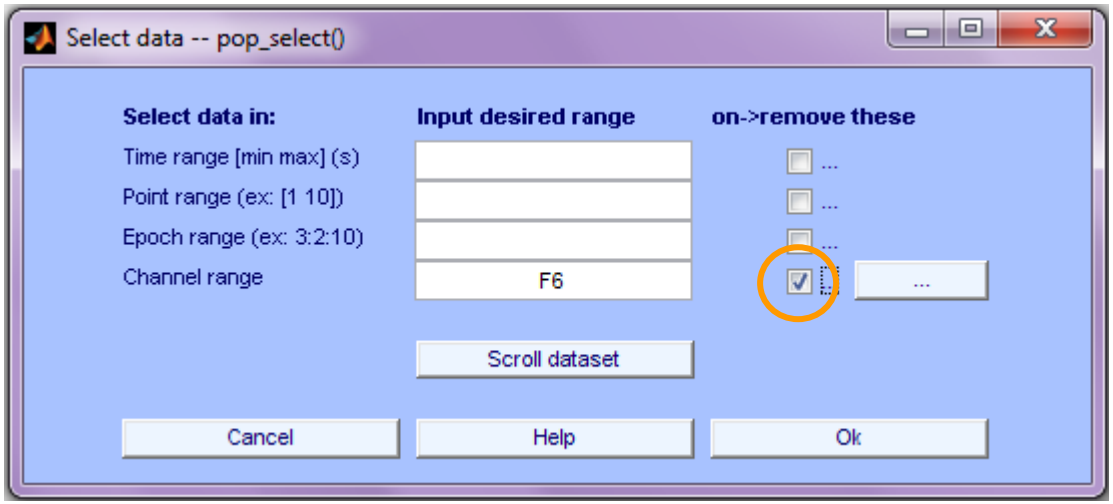
Chan. Time Value

CANCEL Event types << < 449 > >> O1 451.0988 3.6619 35 + - REJECT

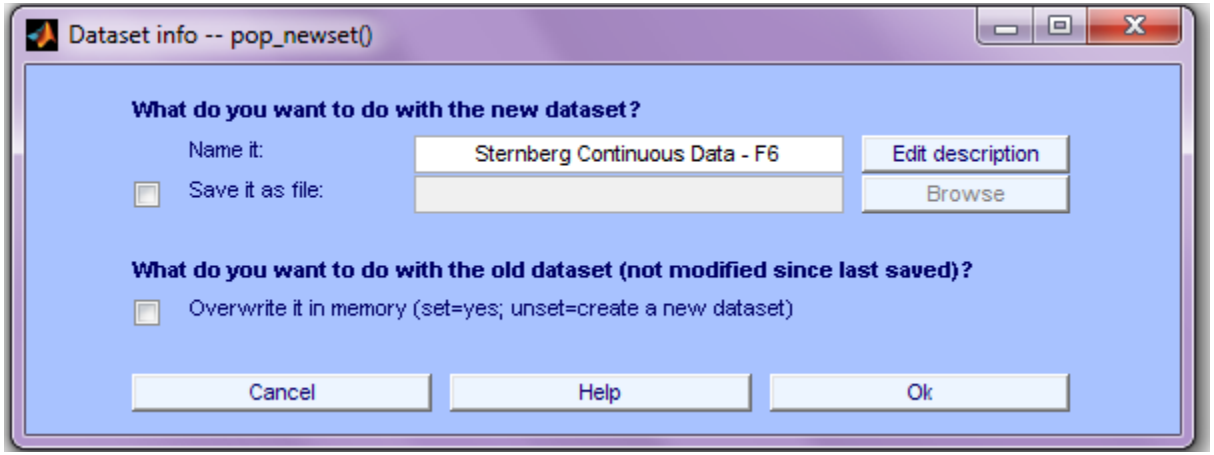
Remove channel(s)



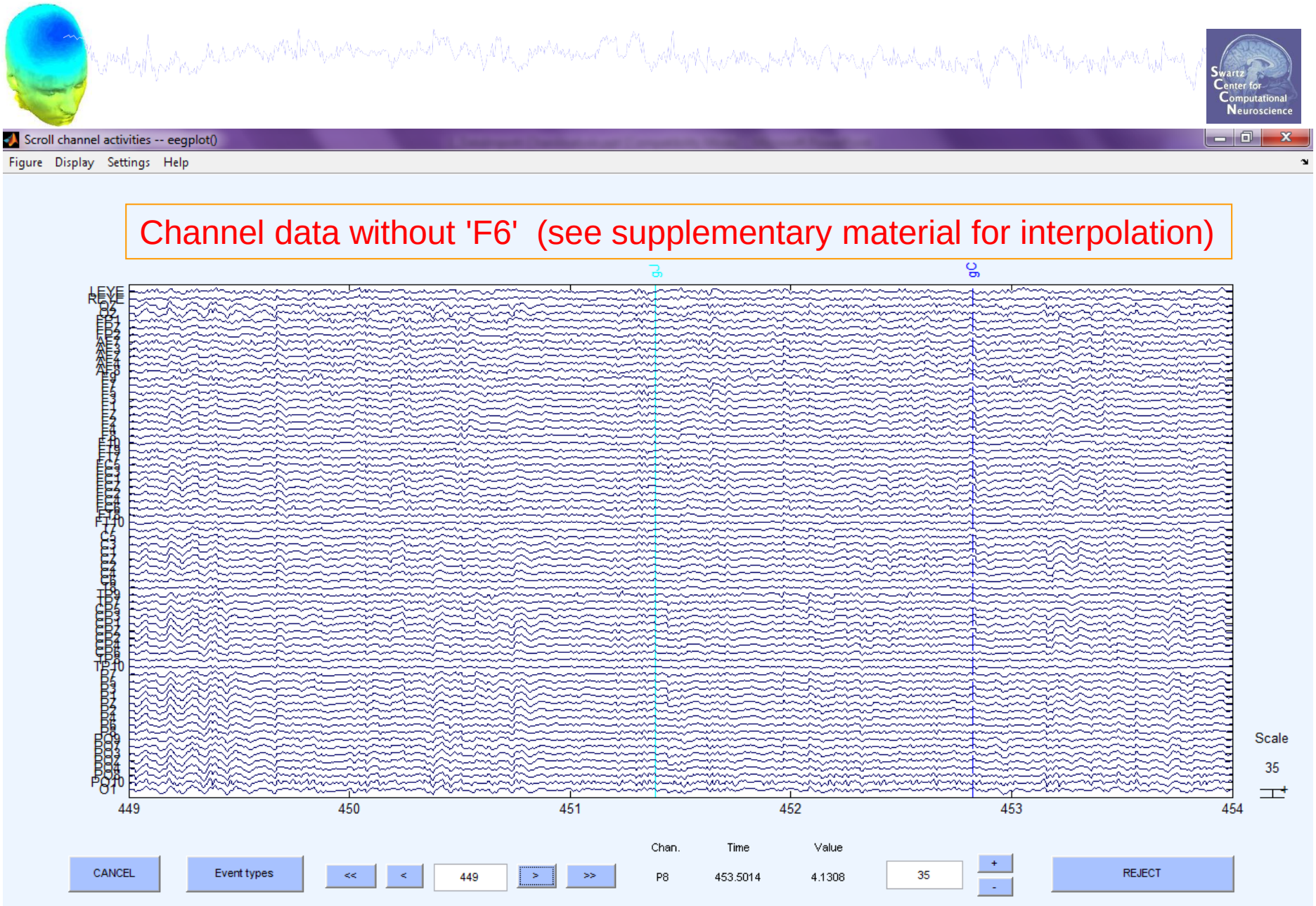
Removing channel(s)



If not checked, will result in dataset with one channel



Channel removed



Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

Examine raw data

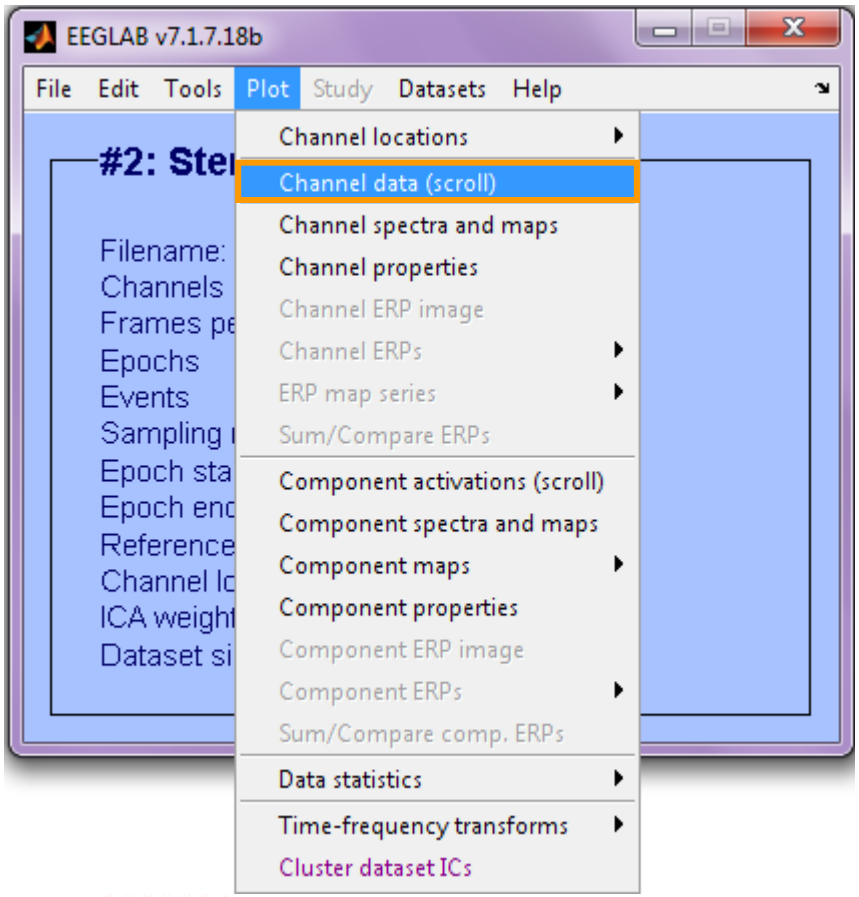
Reject bad channels

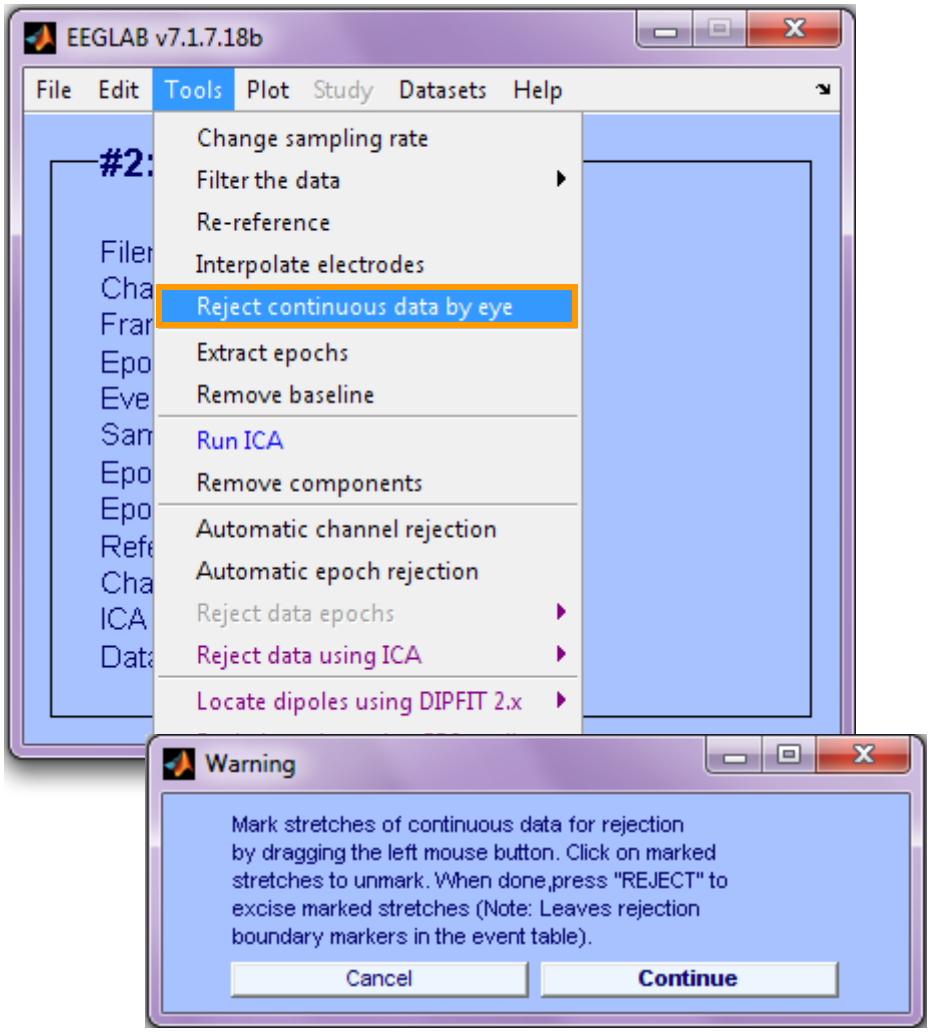
Reject large artifact time points

Run ICA

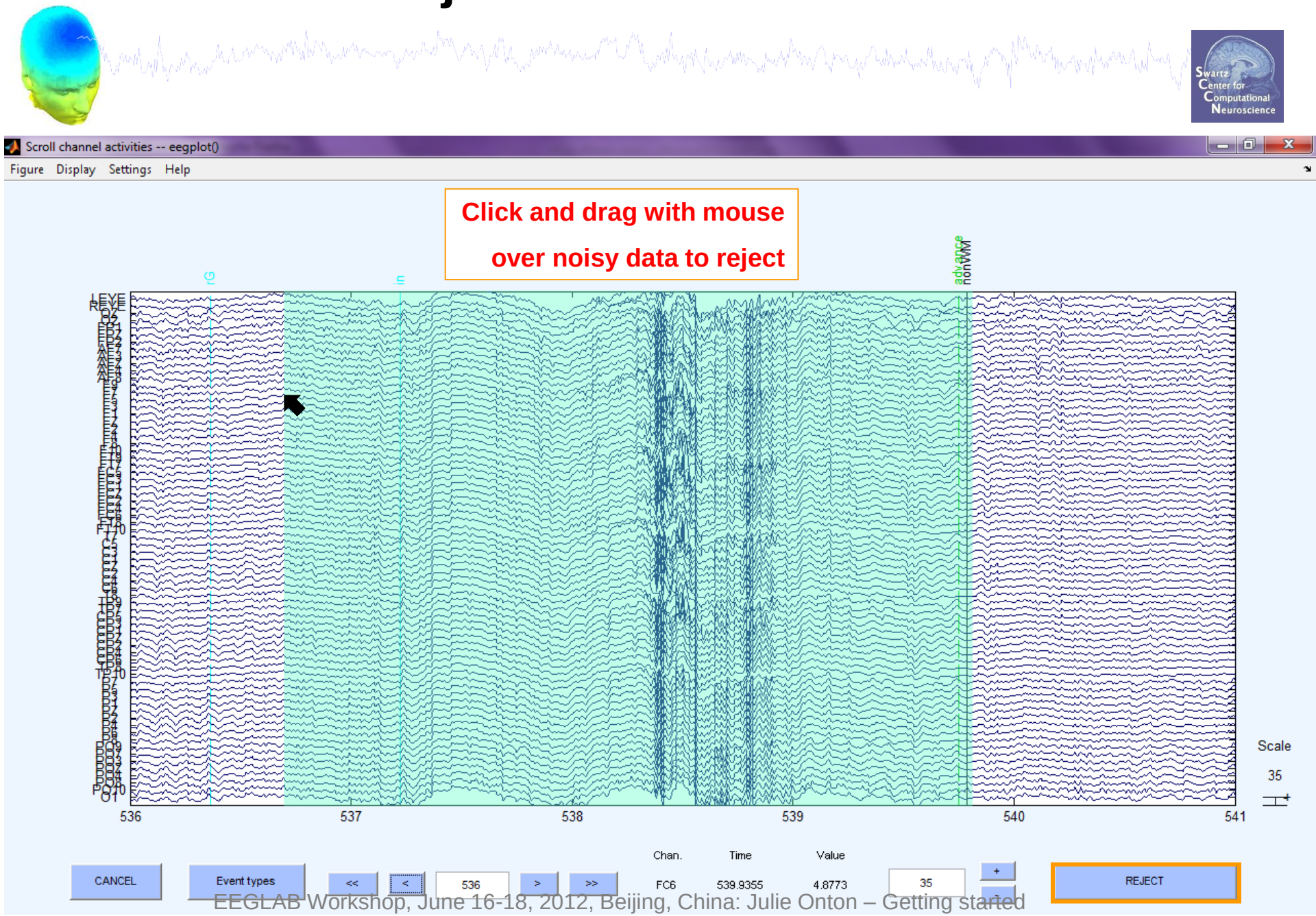
Reject continuous data

Equivalent





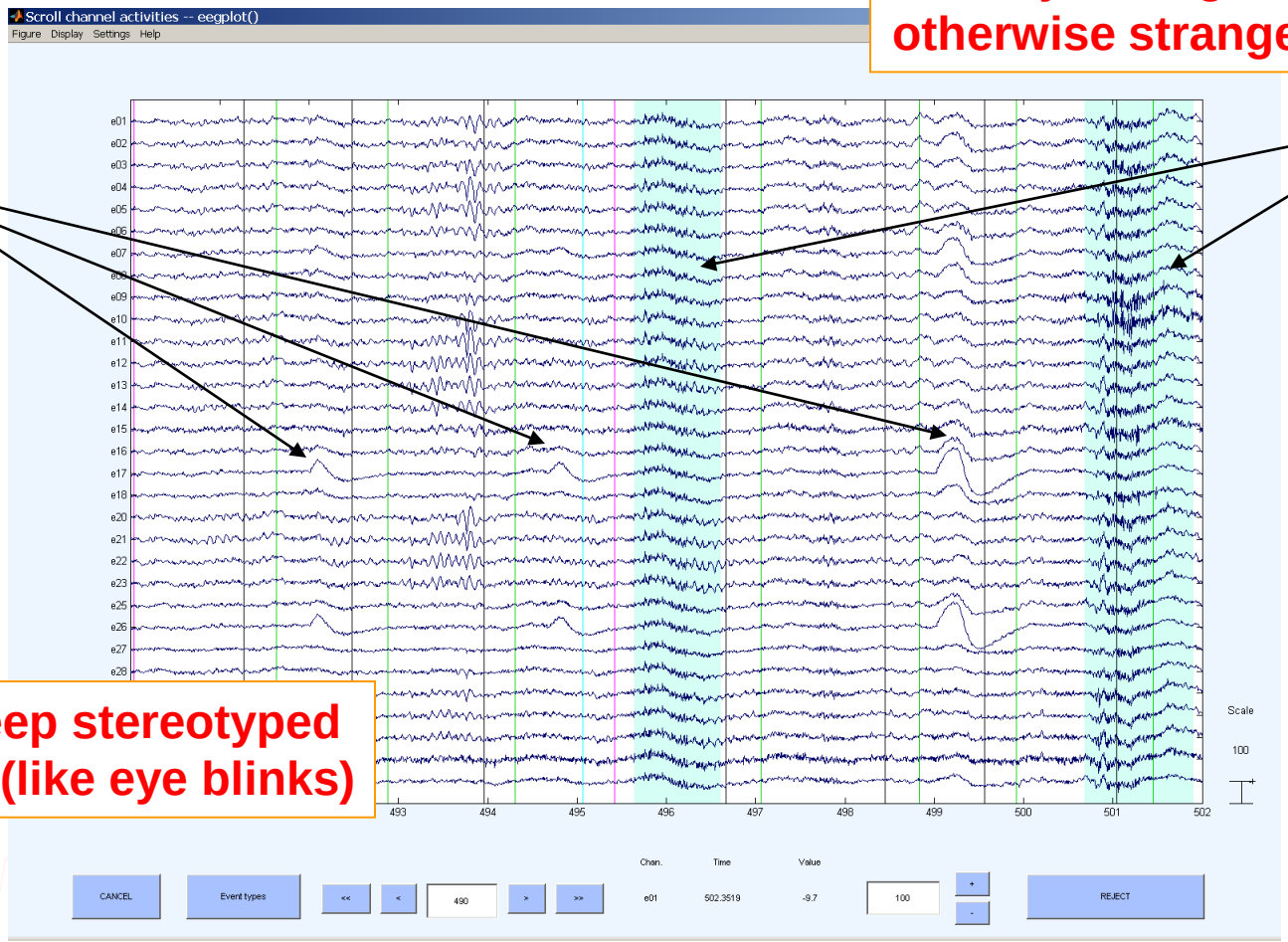
Reject continuous data



Rejecting data for ICA

To prepare data for ICA:

Reject large muscle or otherwise strange events...



... but keep stereotyped artifacts (like eye blinks)

Pre-processing pipeline

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

Examine raw data

Reject bad channels

Reject large artifact time points

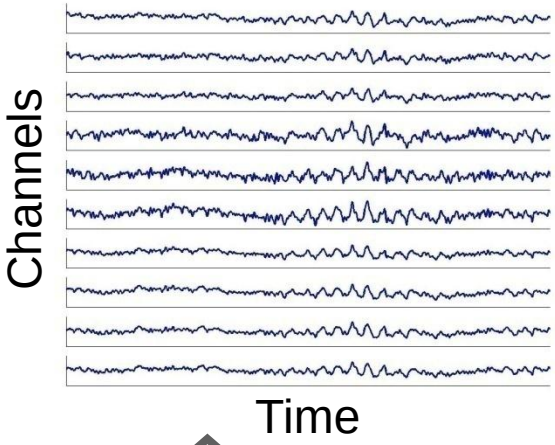
Run ICA

Independent Component Analysis

x = scalp EEG

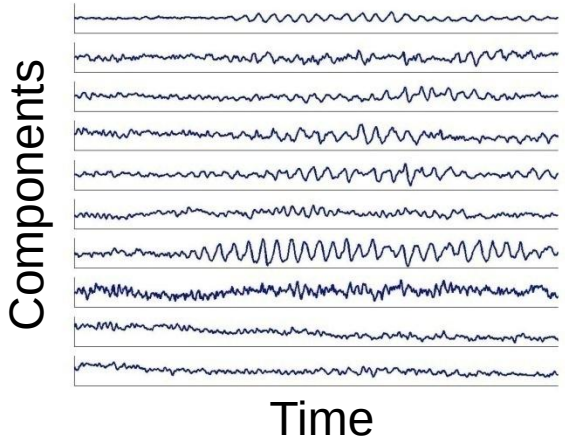
W = unmixing matrix

u = sources



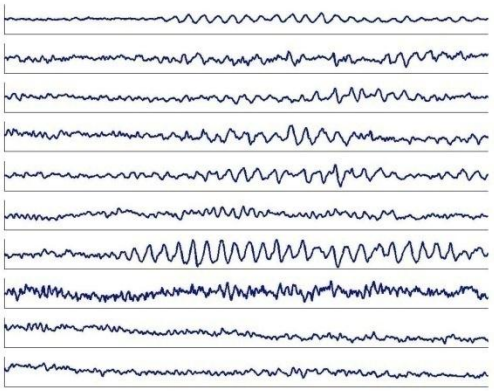
W*x = u

ICA



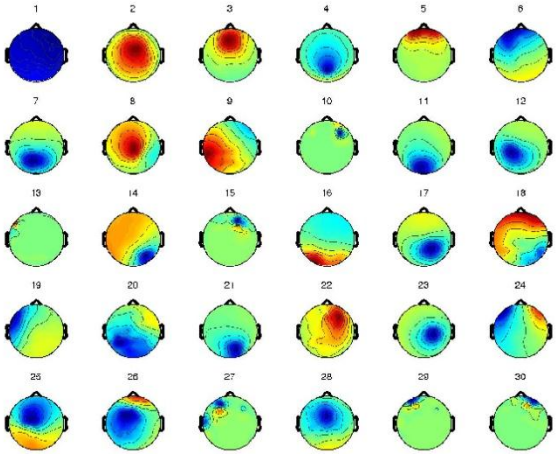
x = W^{-1}*u

u = sources



*

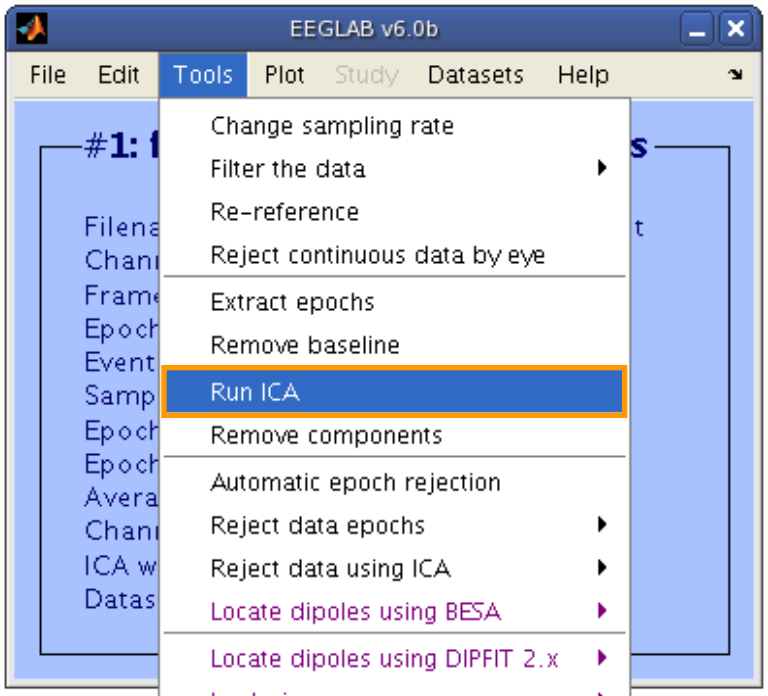
W^{-1} (scalp projections)



“Secrets” to a good ICA decomposition

- Garbage in… garbage out (it’s not magic)
- Remove large, non-stereotyped artifacts
- Do you have enough data? (based mostly on time, not frames)
  * ~30 min of data for 60-70 channels, ~60 min for > 200 channels
- High-pass filter to remove slow drifts
  * low-pass/ notch filters usually unnecessary
- Remove bad channels
- Data must be in double precision (not single)

Runica options



Option	Default	Comments
'extended'	0	1 is recommended to find sub-gaussians
'stop'	1e-7	final weight change → stop
'lrate'	determined from data	too small → too long… too large → wts blow up
'maxsteps'	512	more channels → more steps
'pca'	0 or EEG.nbchan	Decompose only a principal data subspace

Other algorithms: binica,sobi,acsobiro

Runica progress…

Swartz Center for Computational Neuroscience

Press Button to interupt runica

Interupt

csh

```
Input data size [33,133175] = 33 channels, 133175 frames/nFinding 33 ICA components using extended ICA.
Kurtosis will be calculated initially every 1 blocks using 6000 data points.
Decomposing 122 frames per ICA weight ((1089)^2 = 133175 weights, Initial learning rate will be 0.001, block size
Learning rate will be multiplied by 0.98 whenever angledelta >= 60 deg.
More than 32 channels: default stopping weight change 1E-7
Training will end when wchange < 1e-07 or after 512 steps.
Online bias adjustment will be used.
Removing mean of each channel ...
Final training data range: -171.806 to 179.094
Computing the sphering matrix...
Starting weights are the identity matrix ...
Sphering the data ...
Beginning ICA training ... first training step may be slow ...
step 1 - lrate 0.001000, wchange 16.85061324, angledelta  0.0 deg
step 2 - lrate 0.001000, wchange 0.26760405, angledelta  0.0 deg
step 3 - lrate 0.001000, wchange 0.79058323, angledelta 104.0 deg
step 4 - lrate 0.000980, wchange 0.66700031, angledelta 147.2 deg
step 5 - lrate 0.000960, wchange 0.62849071, angledelta 146.5 deg
step 6 - lrate 0.000941, wchange 0.73967955, angledelta 150.7 deg
step 7 - lrate 0.000922, wchange 0.73727229, angledelta 151.6 deg
step 8 - lrate 0.000904, wchange 0.74051387, angledelta 137.9 deg
step 9 - lrate 0.000886, wchange 0.74536137, angledelta 156.0 deg
step 10 - lrate 0.000868, wchange 0.72101402, angledelta 143.7 deg
step 11 - lrate 0.000851, wchange 0.14690114, angledelta 102.5 deg
step 12 - lrate 0.000834, wchange 0.11822100, angledelta 114.3 deg
step 13 - lrate 0.000817, wchange 0.75552966, angledelta 100.6 deg
step 14 - lrate 0.000801, wchange 0.26739750, angledelta 109.1 deg
step 15 - lrate 0.000785, wchange 0.12123251, angledelta 94.2 deg
step 16 - lrate 0.000769, wchange 0.10285606, angledelta 110.7 deg
step 17 - lrate 0.000754, wchange 0.09770499, angledelta 118.6 deg
step 18 - lrate 0.000739, wchange 0.09544428, angledelta 117.1 deg
```

csh

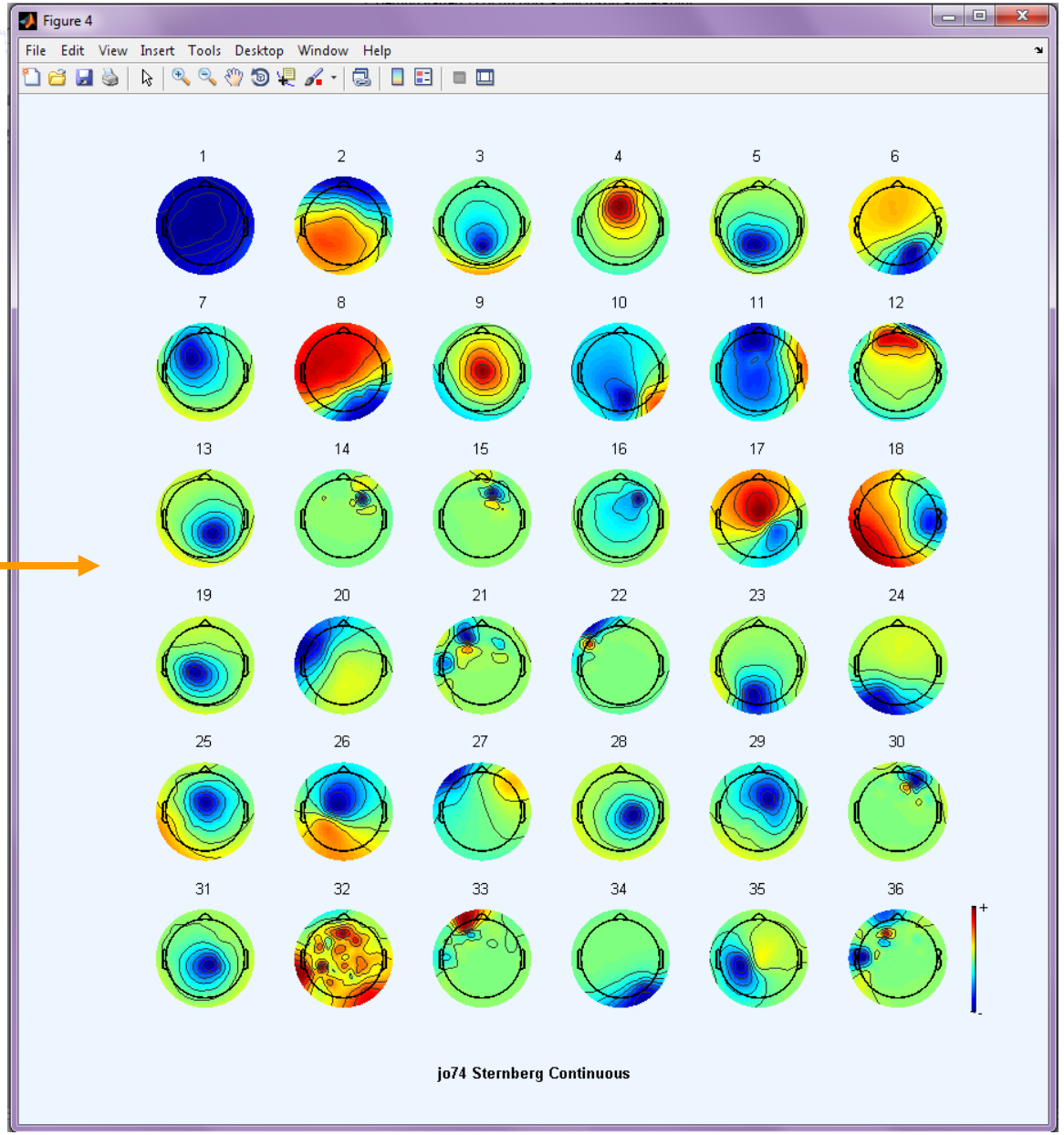
```
step 241 - lrate 0.000002, wchange 0.00000082, angledelta 101.5 deg
step 242 - lrate 0.000001, wchange 0.00000061, angledelta 96.1 deg
step 243 - lrate 0.000001, wchange 0.00000057, angledelta 97.5 deg
step 244 - lrate 0.000001, wchange 0.00000054, angledelta 93.7 deg
step 245 - lrate 0.000001, wchange 0.00000055, angledelta 100.3 deg
step 246 - lrate 0.000001, wchange 0.00000047, angledelta 96.9 deg
step 247 - lrate 0.000001, wchange 0.00000046, angledelta 91.3 deg
step 248 - lrate 0.000001, wchange 0.00000045, angledelta 101.5 deg
step 249 - lrate 0.000001, wchange 0.00000041, angledelta 103.1 deg
step 250 - lrate 0.000001, wchange 0.00000036, angledelta 95.5 deg
step 251 - lrate 0.000001, wchange 0.00000033, angledelta 92.1 deg
step 252 - lrate 0.000001, wchange 0.00000029, angledelta 97.4 deg
step 253 - lrate 0.000001, wchange 0.00000030, angledelta 95.8 deg
step 254 - lrate 0.000001, wchange 0.00000023, angledelta 94.2 deg
step 255 - lrate 0.000001, wchange 0.00000023, angledelta 97.6 deg
step 256 - lrate 0.000001, wchange 0.00000023, angledelta 97.1 deg
step 257 - lrate 0.000001, wchange 0.00000021, angledelta 92.0 deg
step 258 - lrate 0.000001, wchange 0.00000020, angledelta 99.1 deg
step 259 - lrate 0.000001, wchange 0.00000019, angledelta 95.0 deg
step 260 - lrate 0.000001, wchange 0.00000015, angledelta 98.3 deg
step 261 - lrate 0.000001, wchange 0.00000014, angledelta 99.0 deg
step 262 - lrate 0.000001, wchange 0.00000014, angledelta 94.3 deg
step 263 - lrate 0.000001, wchange 0.00000013, angledelta 95.4 deg
step 264 - lrate 0.000001, wchange 0.00000012, angledelta 94.1 deg
step 265 - lrate 0.000001, wchange 0.00000011, angledelta 96.1 deg
step 266 - lrate 0.000001, wchange 0.00000010, angledelta 94.8 deg
step 267 - lrate 0.000001, wchange 0.00000010, angledelta 94.5 deg
step 268 - lrate 0.000001, wchange 0.00000010, angledelta 97.7 deg
step 269 - lrate 0.000001, wchange 0.00000008, angledelta 95.1 deg
Sorting components in descending order of mean projected variance ...
Permuting the activation wave forms ...
>>
>>
```

ICA weights in EEG structure

Terminal

File Edit View Terminal Tabs Help

```
>> EEG

EEG =

            setname: 'faces_4 continuous'
           filename: 'faces_4.set'
           filepath: '/home/julie/workshop06/'
            subject: ''
              group: ''
          condition: ''
            session: []
           comments: [15x48 char]
             nbchan: 33
             trials: 1
               pnts: 133175
              srate: 250
               xmin: 0
               xmax: 532.6960
              times: []
               data: [33x133175 single]
             icaact: [33x133175 single]
            icawinv: [33x33 double]
          icasphere: [33x33 double]
         icaweights: [33x33 double]
       icachansind: [1x33 double]
           chanlocs: [1x33 struct]
         urchanlocs: []
           chaninfo: [1x1 struct]
                ref: 'common'
              event: [1x731 struct]
            urevent: [1x731 struct]
  eventdescription: {[]  []}
              epoch: []
  epochdescription: {}
             reject: [1x1 struct]
              stats: [1x1 struct]
           specdata: []
         specicaact: []
         splinefile: ''
      icasplinefile: ''
             dipfit: [1x1 struct]
            history: [1x1633 char]
              saved: 'no'
                etc: []

>>
```



Pre-processing pipeline (review)

Swartz Center for Computational Neuroscience

Collect high-density EEG data (>30 chan)

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

Examine raw data

Reject bad channels

Reject large artifact time points

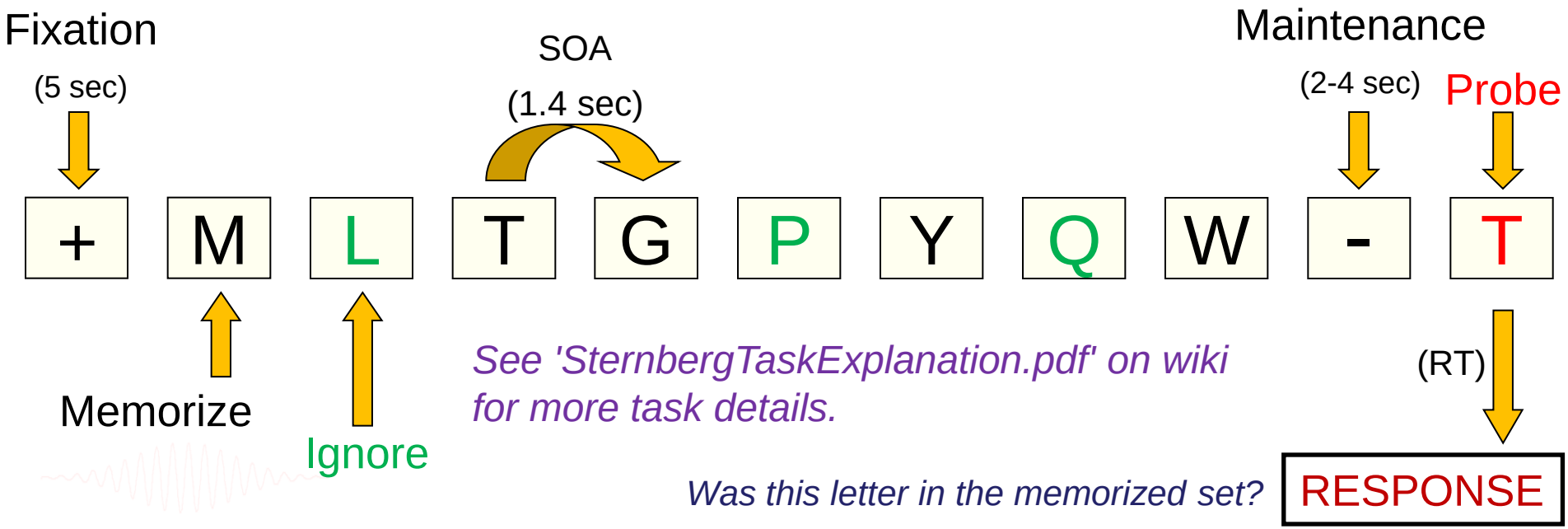
Run ICA

The example data: Sternberg working memory

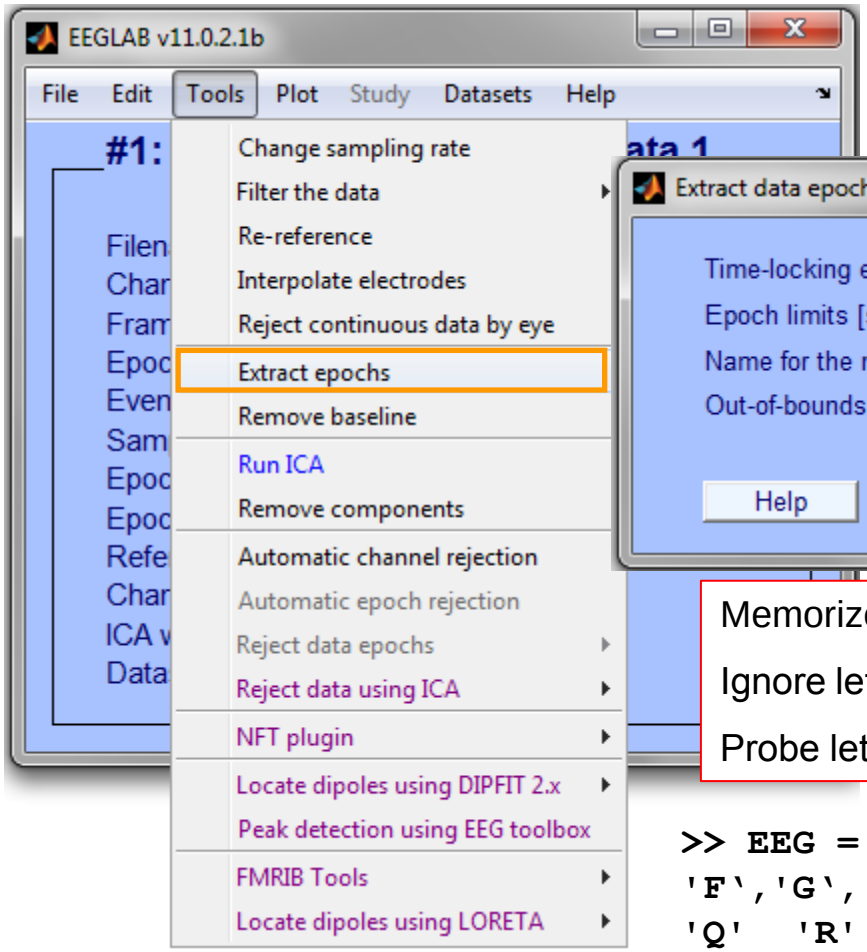
File .../SampleData/**stern.set**

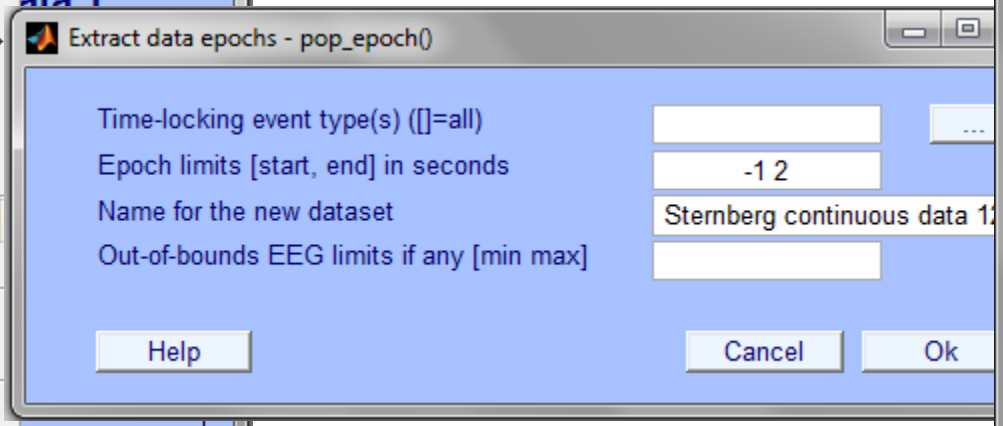
Data Continuous data (not epoched), ref'd to right mastoid

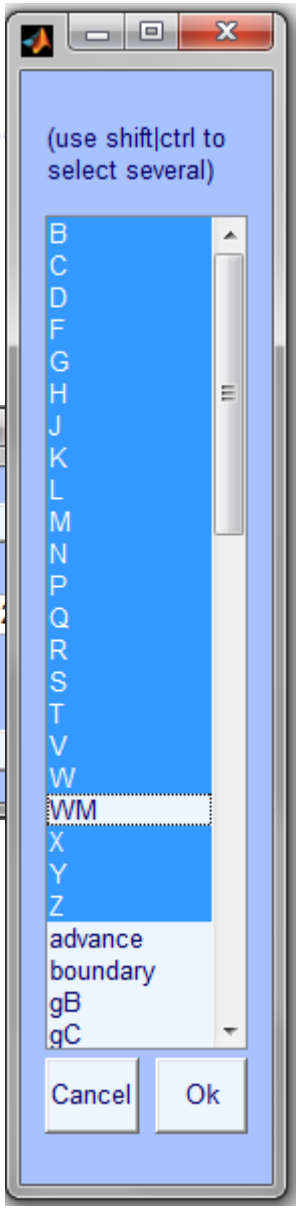
Task between **3** and **7** letters to **memorize (colored black)**,
between **1** and **5** letters to **ignore (colored green)**,
8 letters presented during each trial
50% chance of **probe** letter being 'in-set'



Epoch on EEG.event type







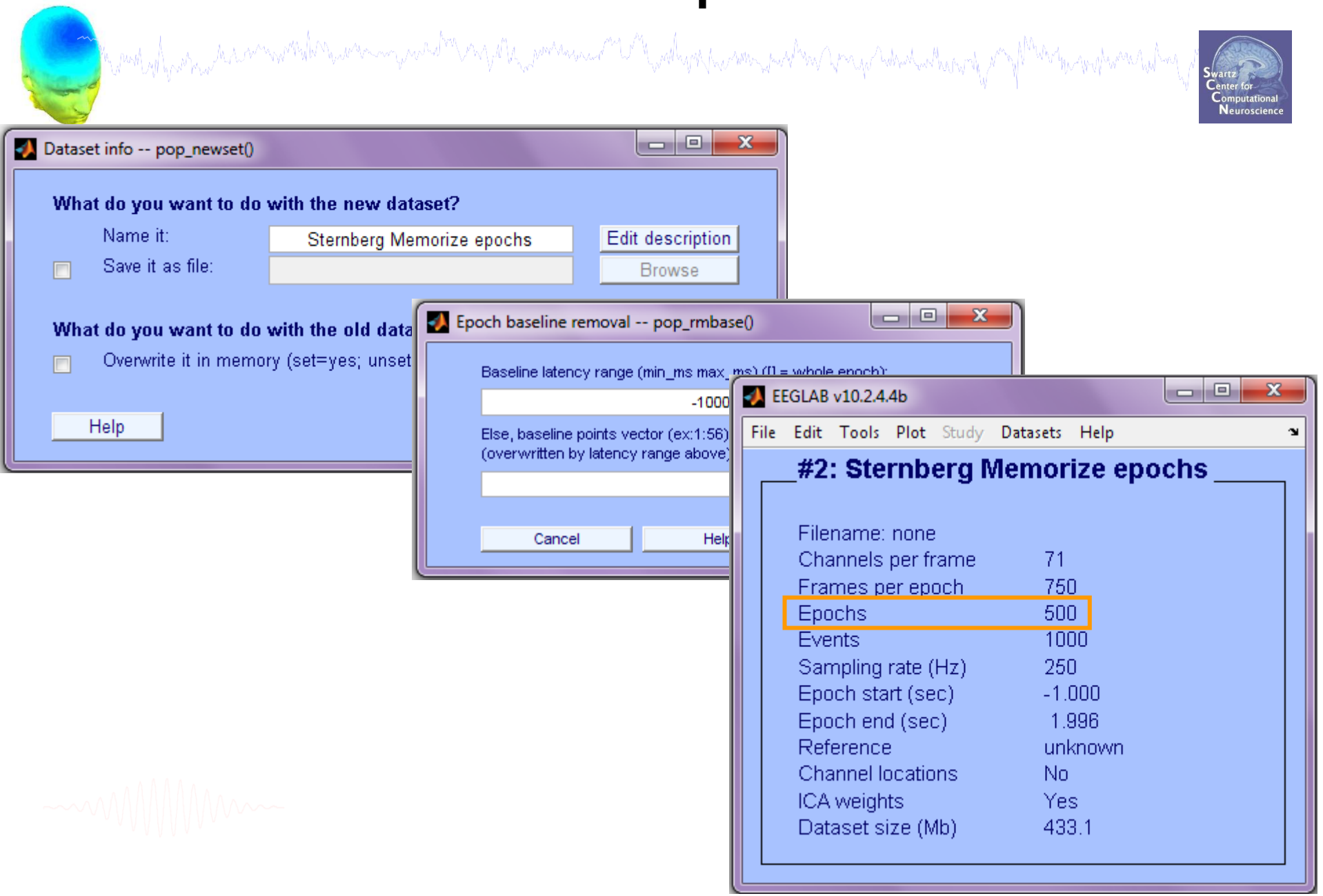
Memorize letters: capital letters

Ignore letters: ‘g’ preceding capital letter (e.g., ‘gB’)

Probe letters: ‘r’ preceding capital letter (e.g., ‘rB’)

```
>> EEG = pop_epoch( EEG, {'B','C','D',...
'F','G','H','J','K','L','M','N','P'...
'Q'  'R'  'S'  'T'  'V'  'W'...
'X'  'Y'  'Z'  }, [-1  2], 'newname',...
'Sternberg Memorize letter epochs',...
'epochinfo', 'yes');
```

Extract epochs



Exercise

- **ALL**

  -Load stern.set (continuous data)

  -Do not save your changes under the same filename!

- **Novice**

  -Scroll channel data and explore plotting options under 'Settings'.

  -Reject noisy time points by visual inspection

  -Import standard channel locations

  -Practice preprocessing steps described in this lecture

- **Intermediate / Advanced** (requires supplementary material)

  -Remove a channel and then replace it by interpolation

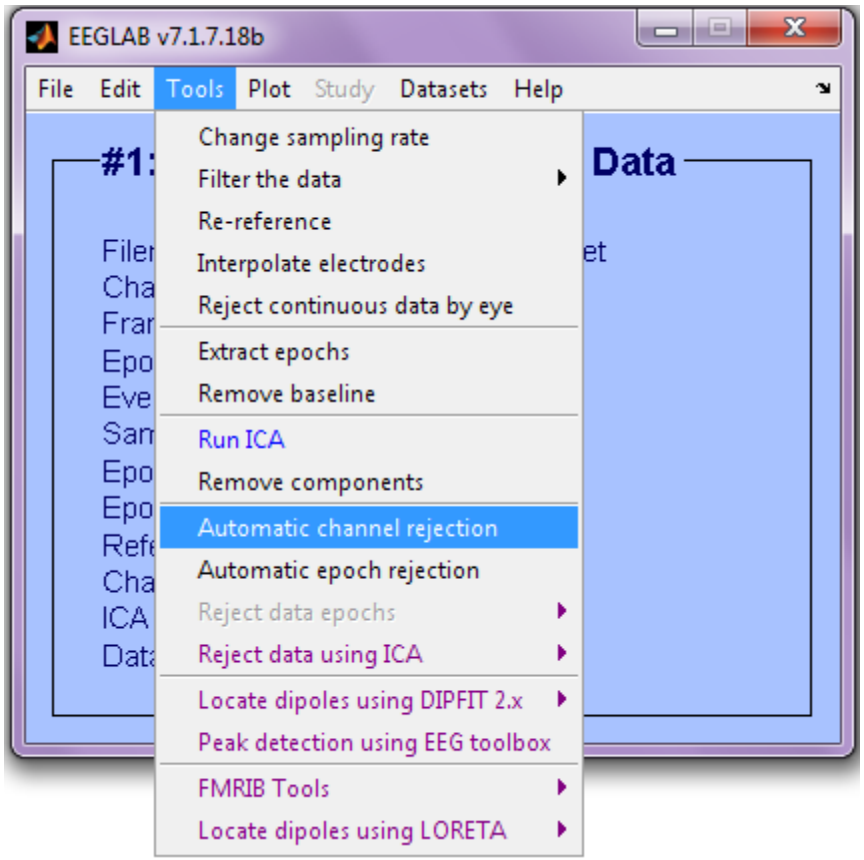
  -Compare this signal with the original when you do this with a 'clean' channel

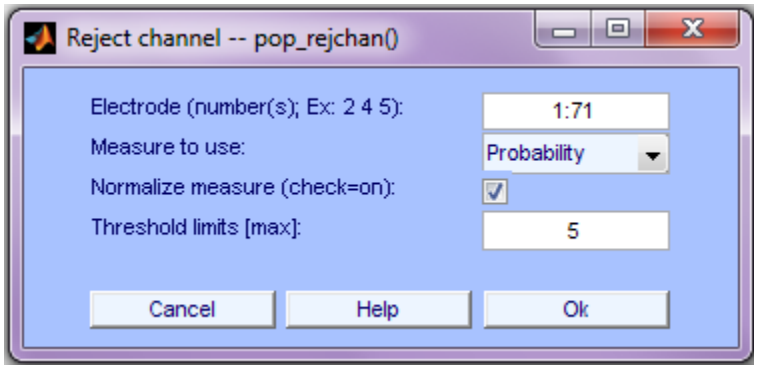
  -Epoch data even of interest, plot Channel ERPs from Plot menu

  -Try different filter methods and cut-offs, compare results

Supplementary lessons

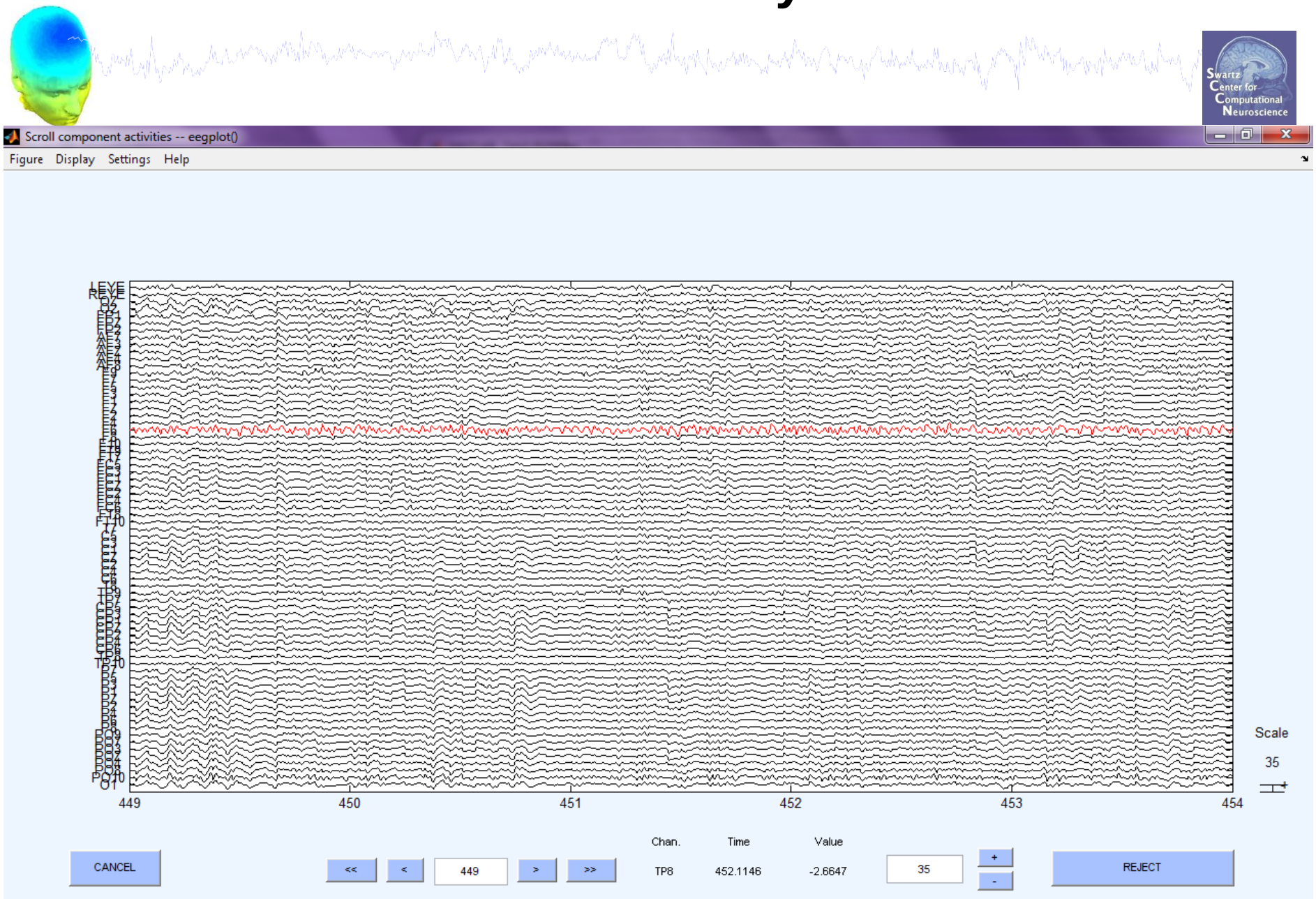
Auto-detection of noisy channels



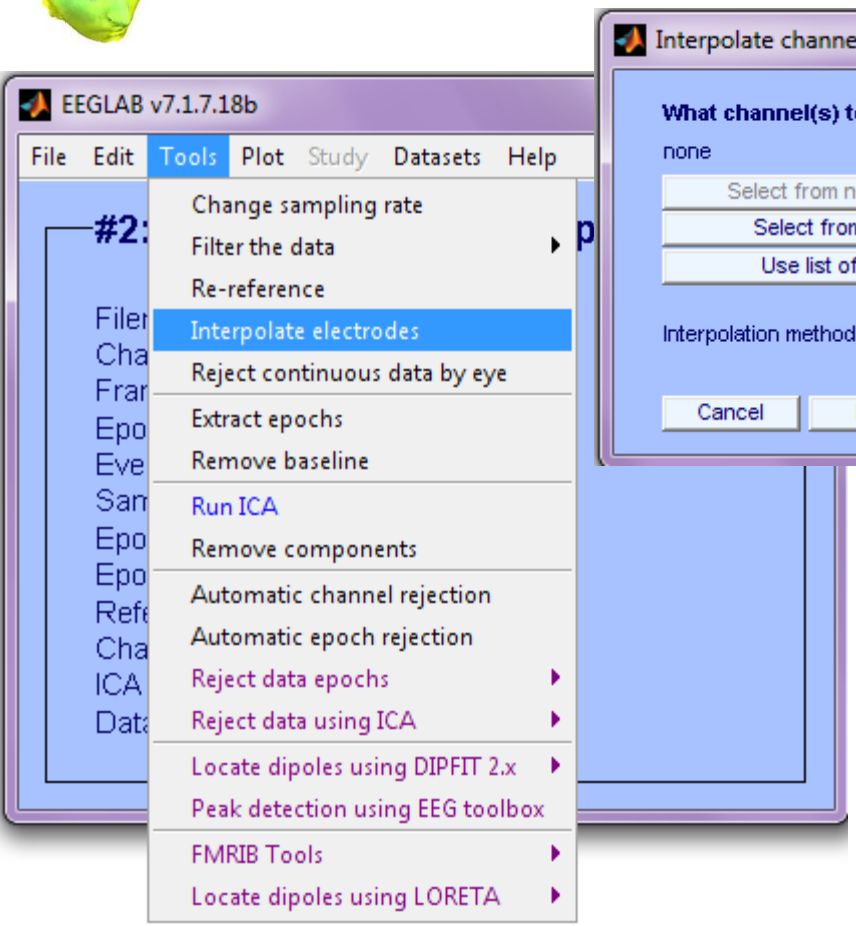


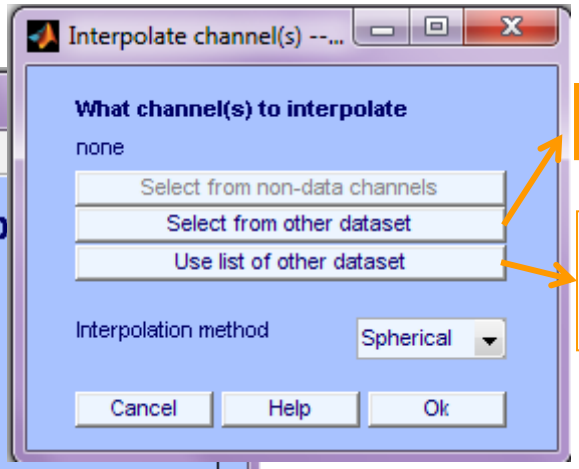

```
>> EEG = pop_rejchan(EEG,  'elec',[1:71] , 'threshold',5,...
 'norm', 'on', 'measure', 'prob');
```

Auto-detected noisy channel



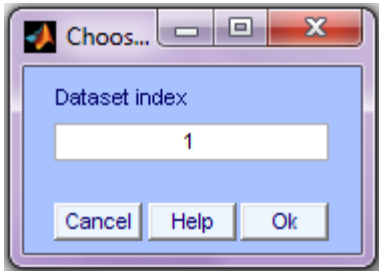
Interpolate bad channel

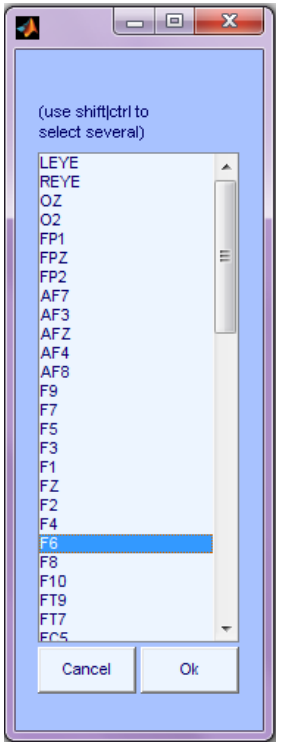


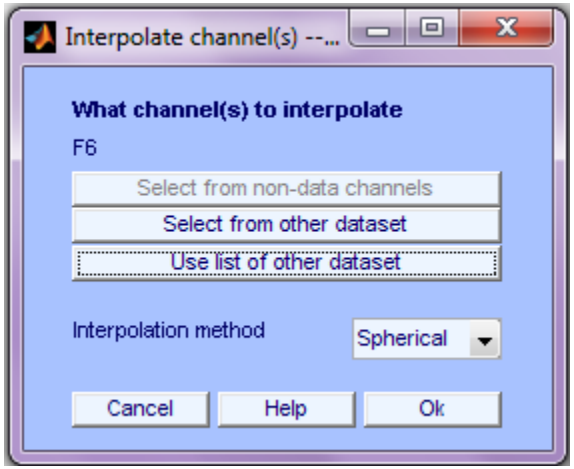


Choose a channel from other dataset

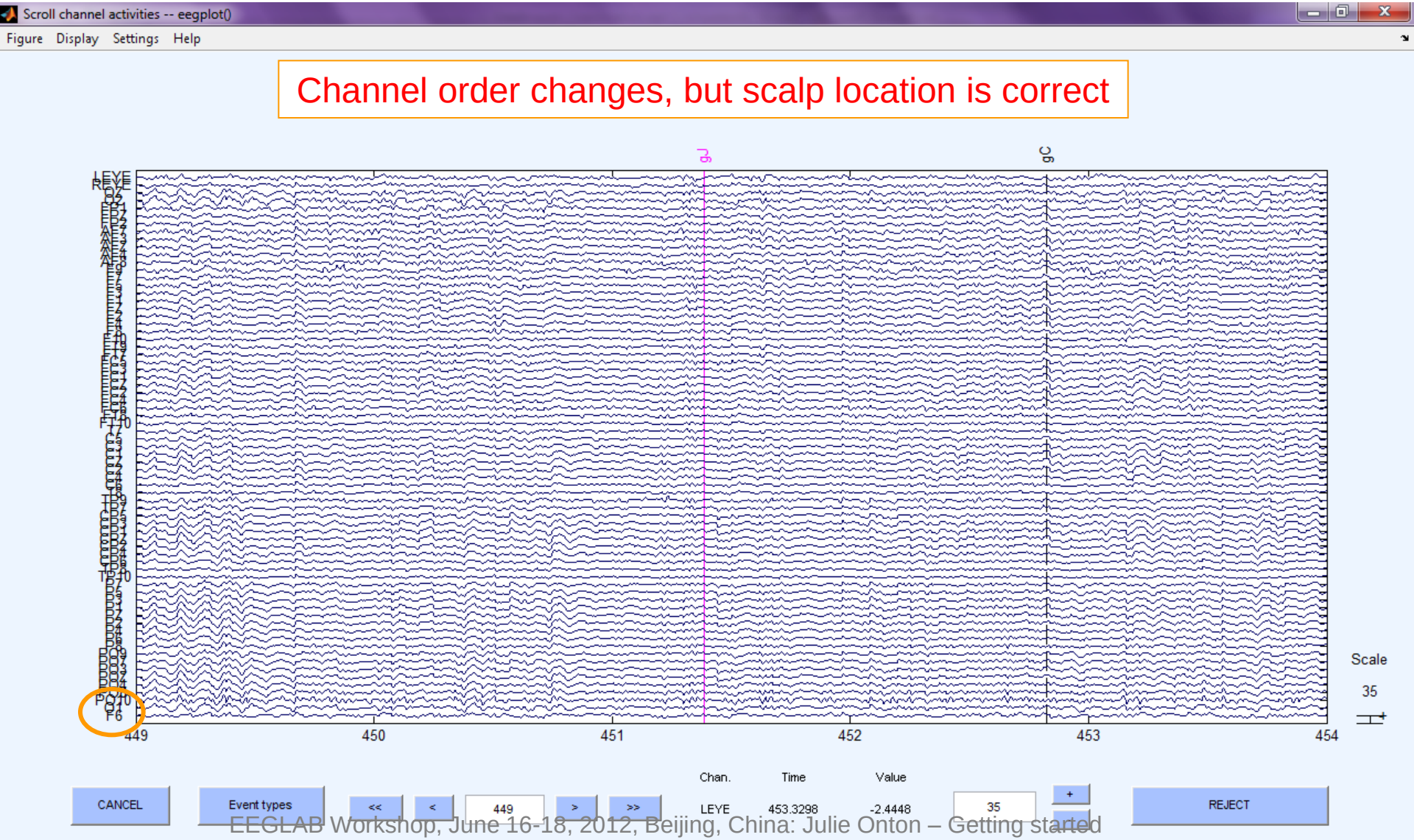
Auto-select deleted channel from other dataset



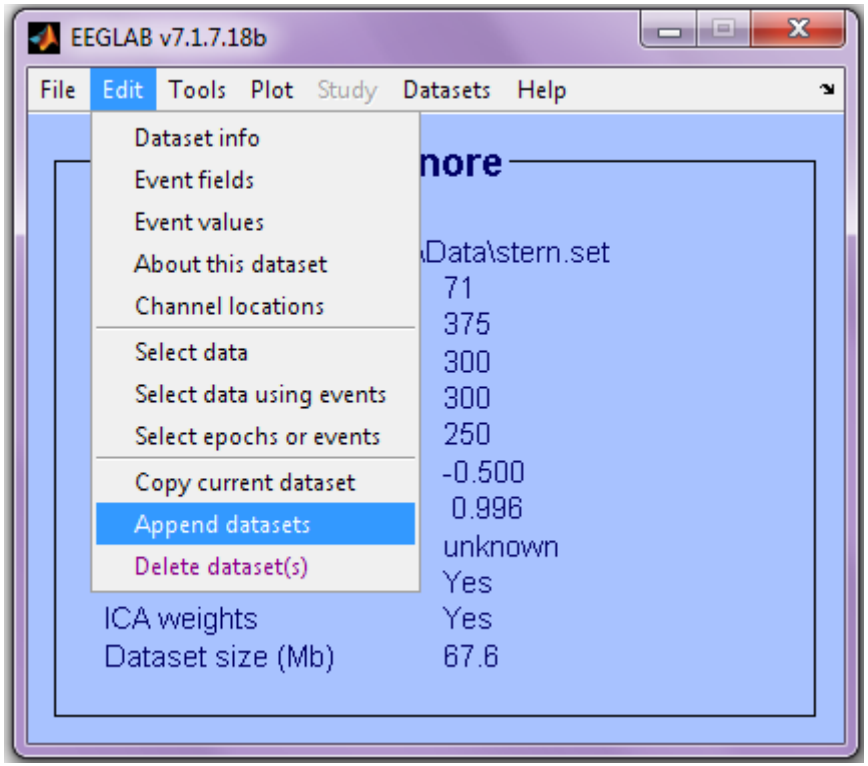


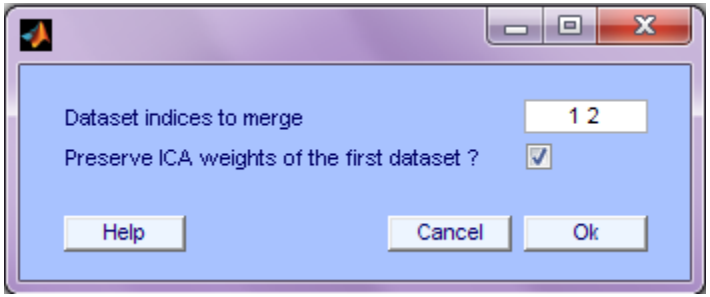


Interpolated channel

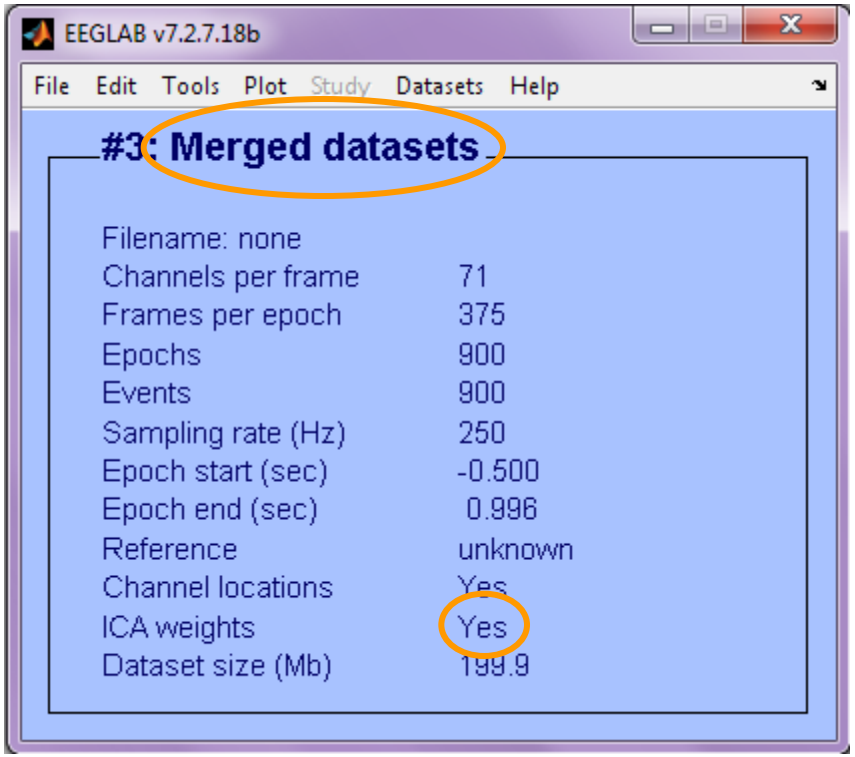


Merge (append) datasets

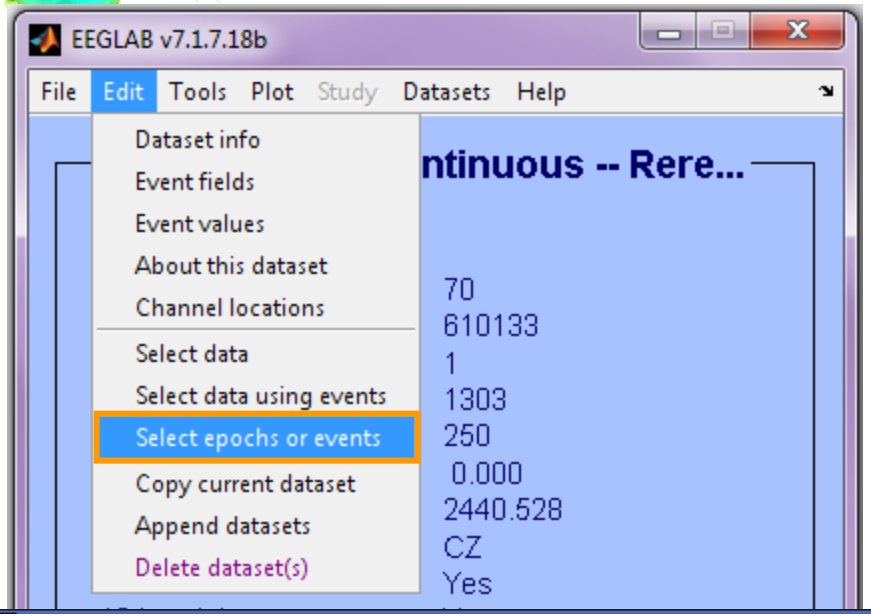




Merged datasets



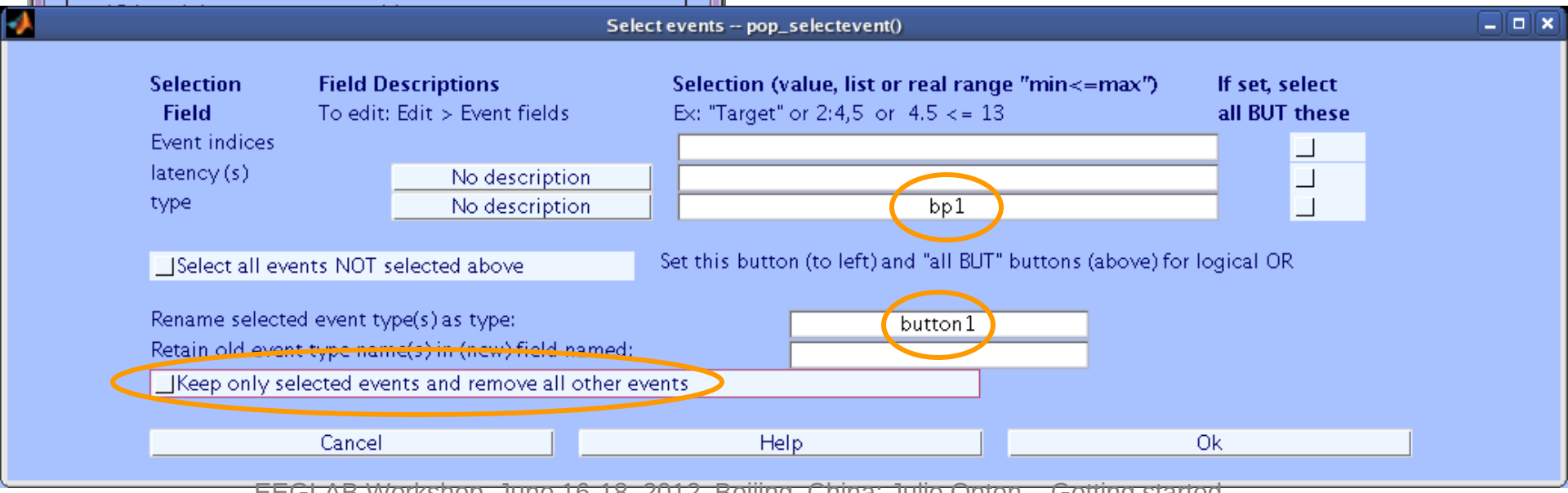
Renaming events



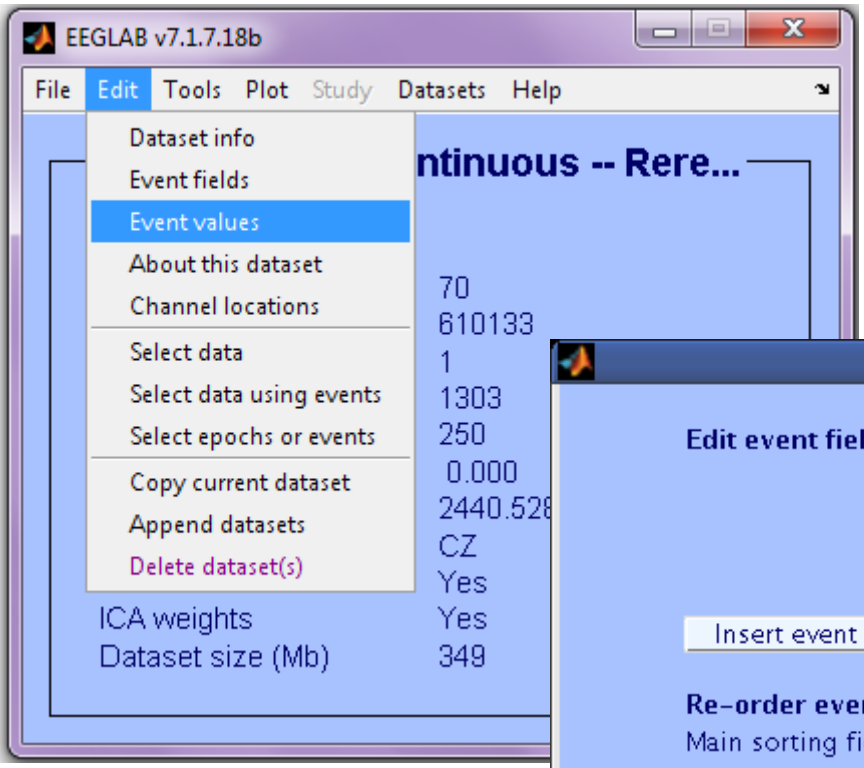
1) input original 'type' code

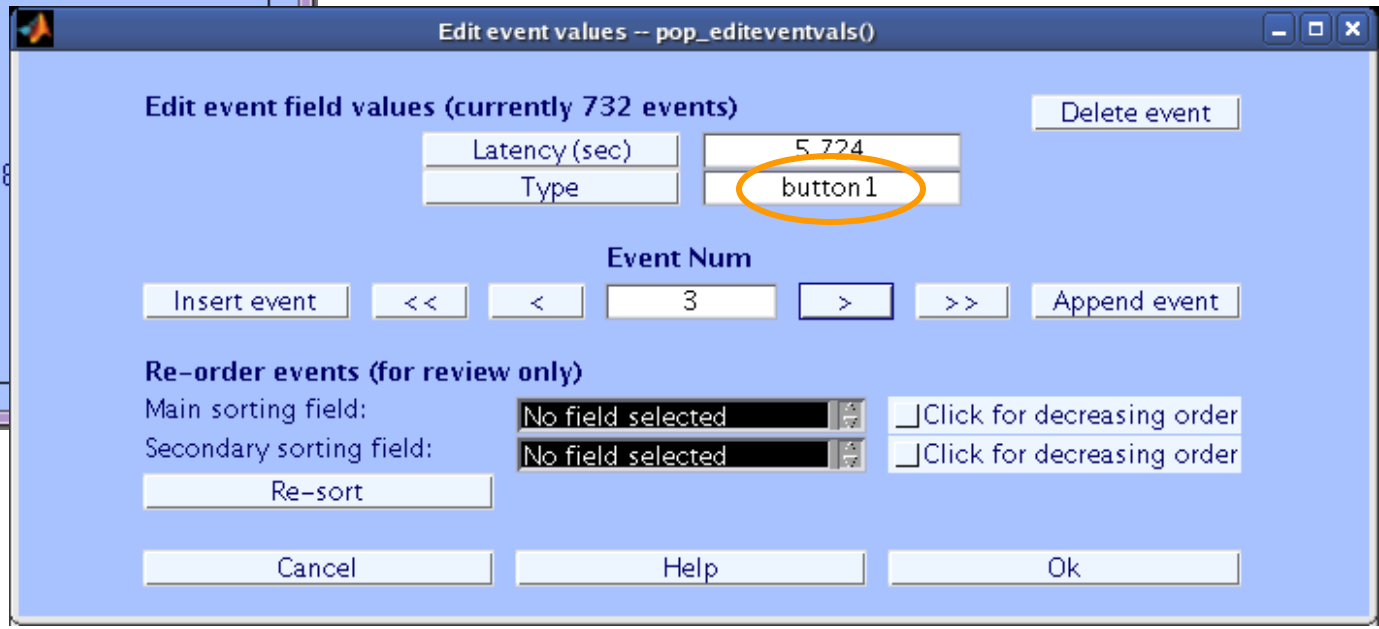
2) input new 'type' code

3) Keep/delete all other events

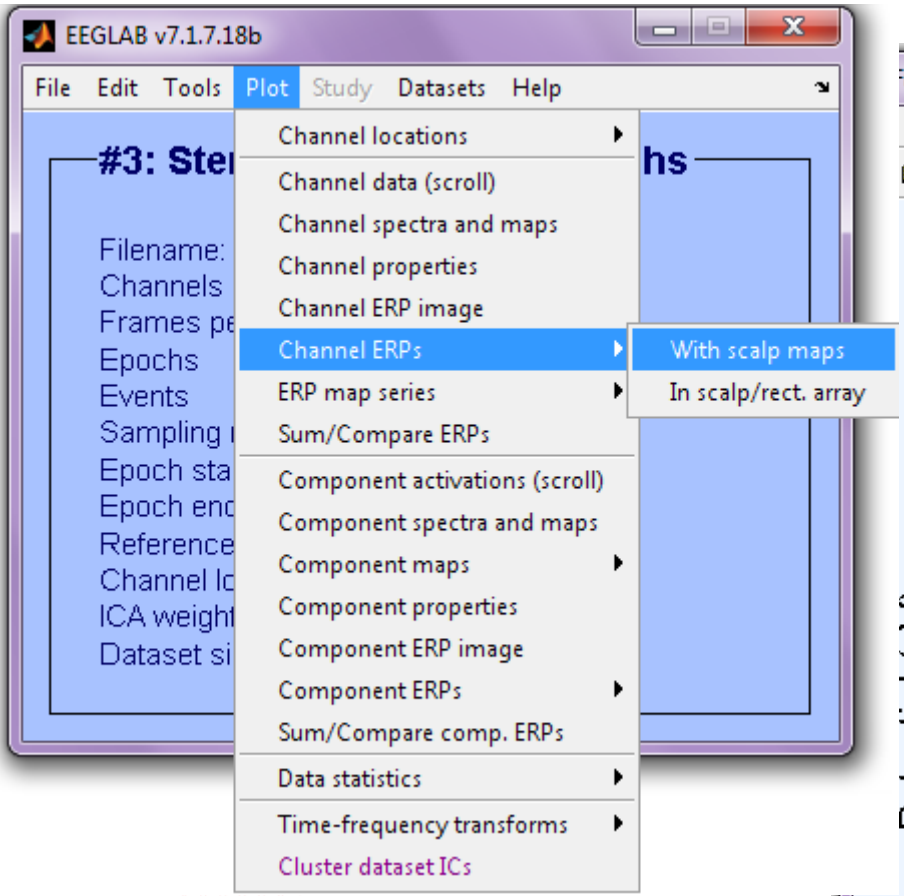


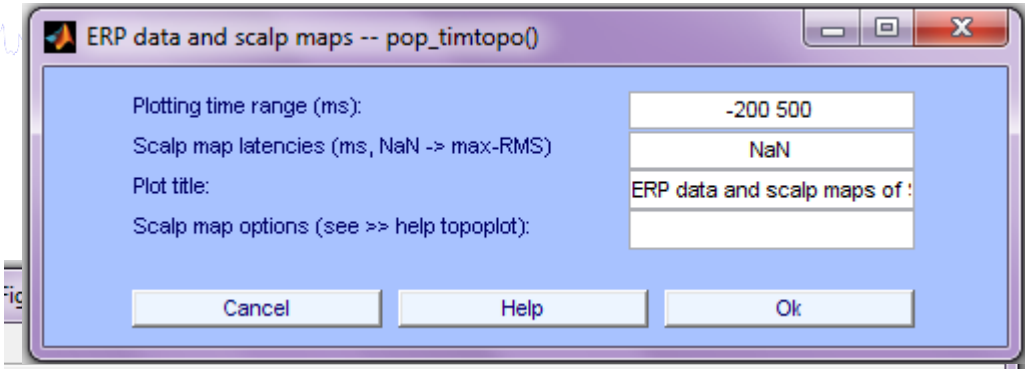
Renaming events

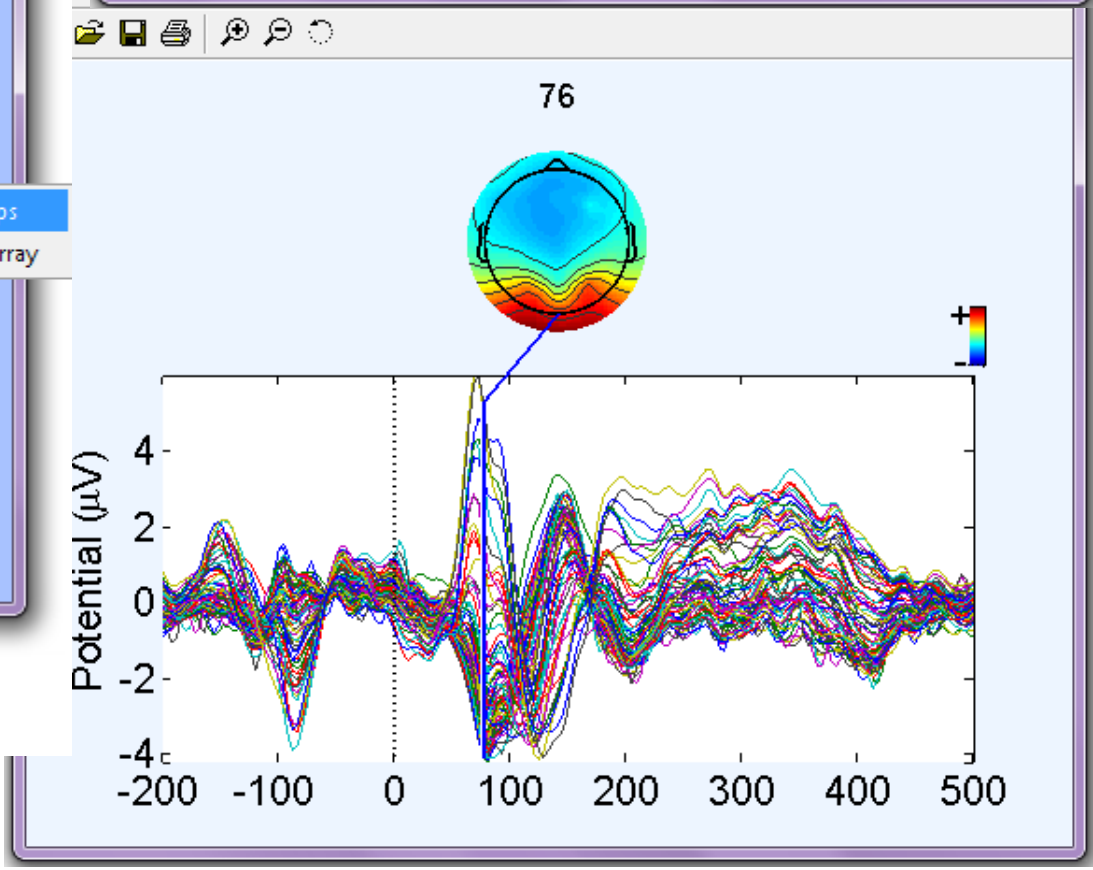




Analysis of channel ERPs

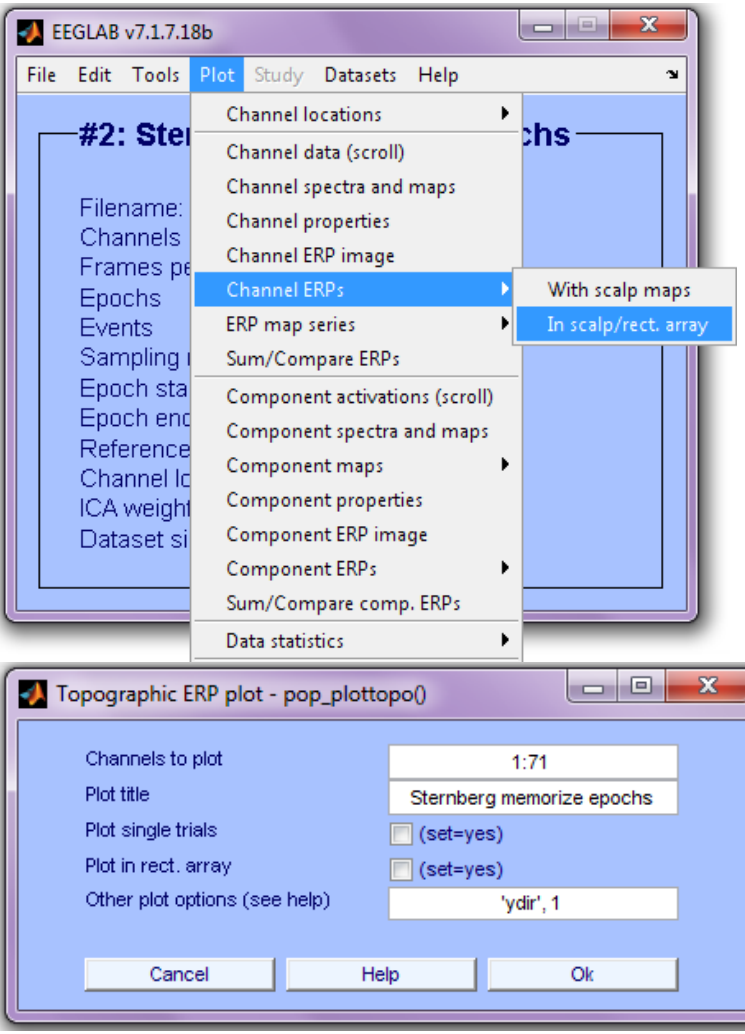


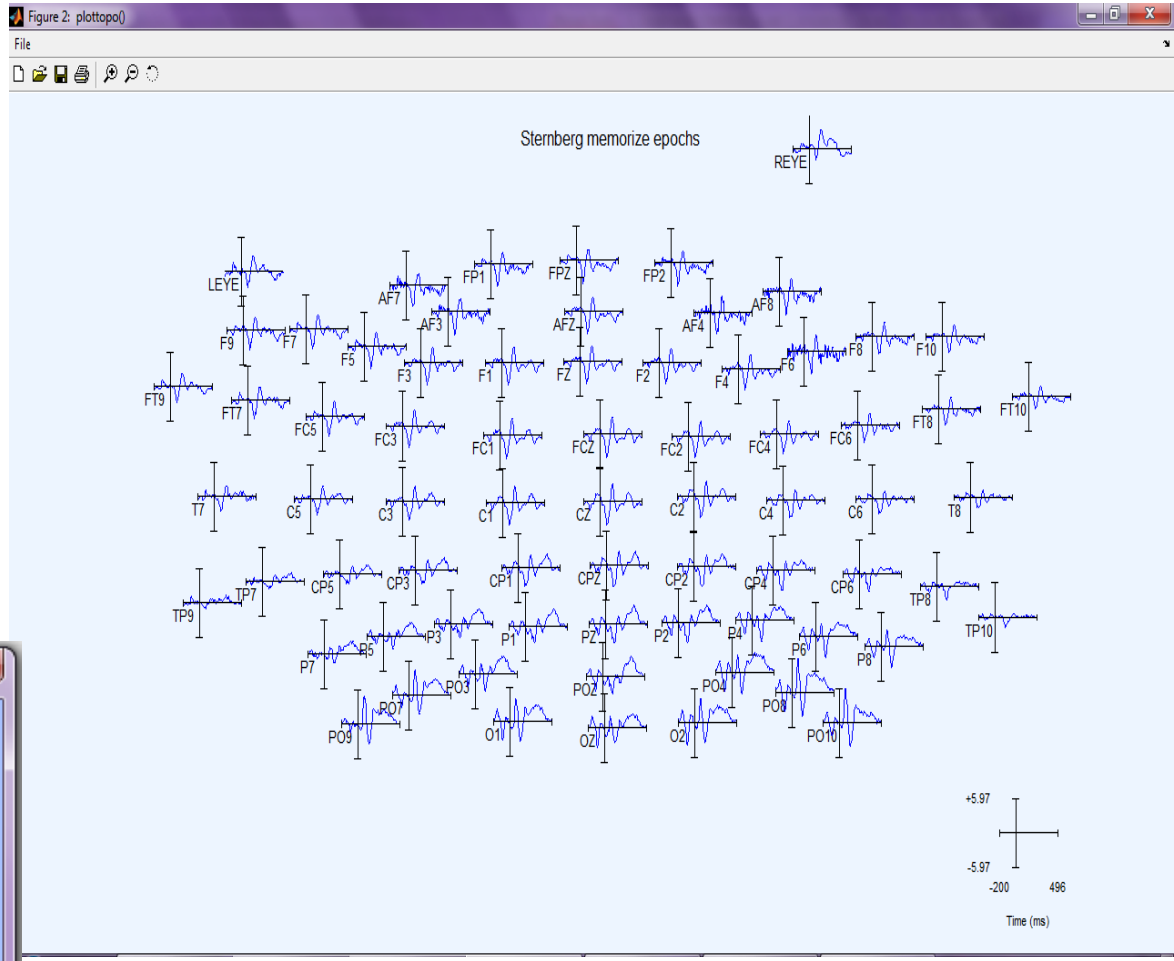




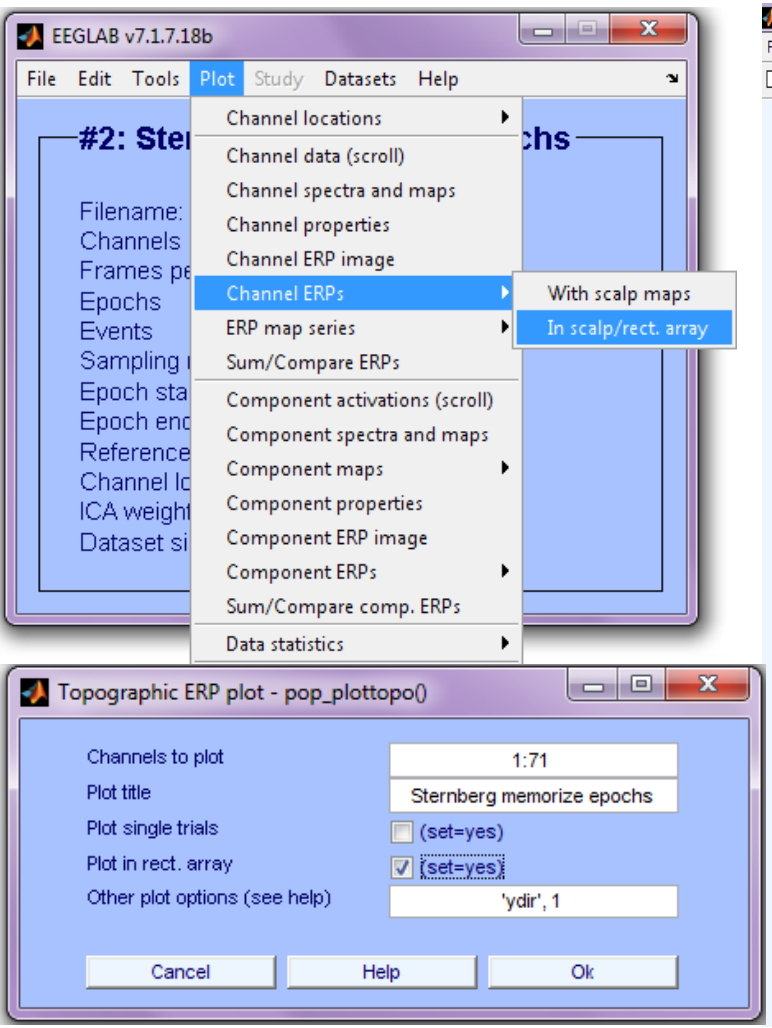

```
>> pop_timtopo(EEG,[-200 500],[NaN],'ERP data and scalp maps');
```

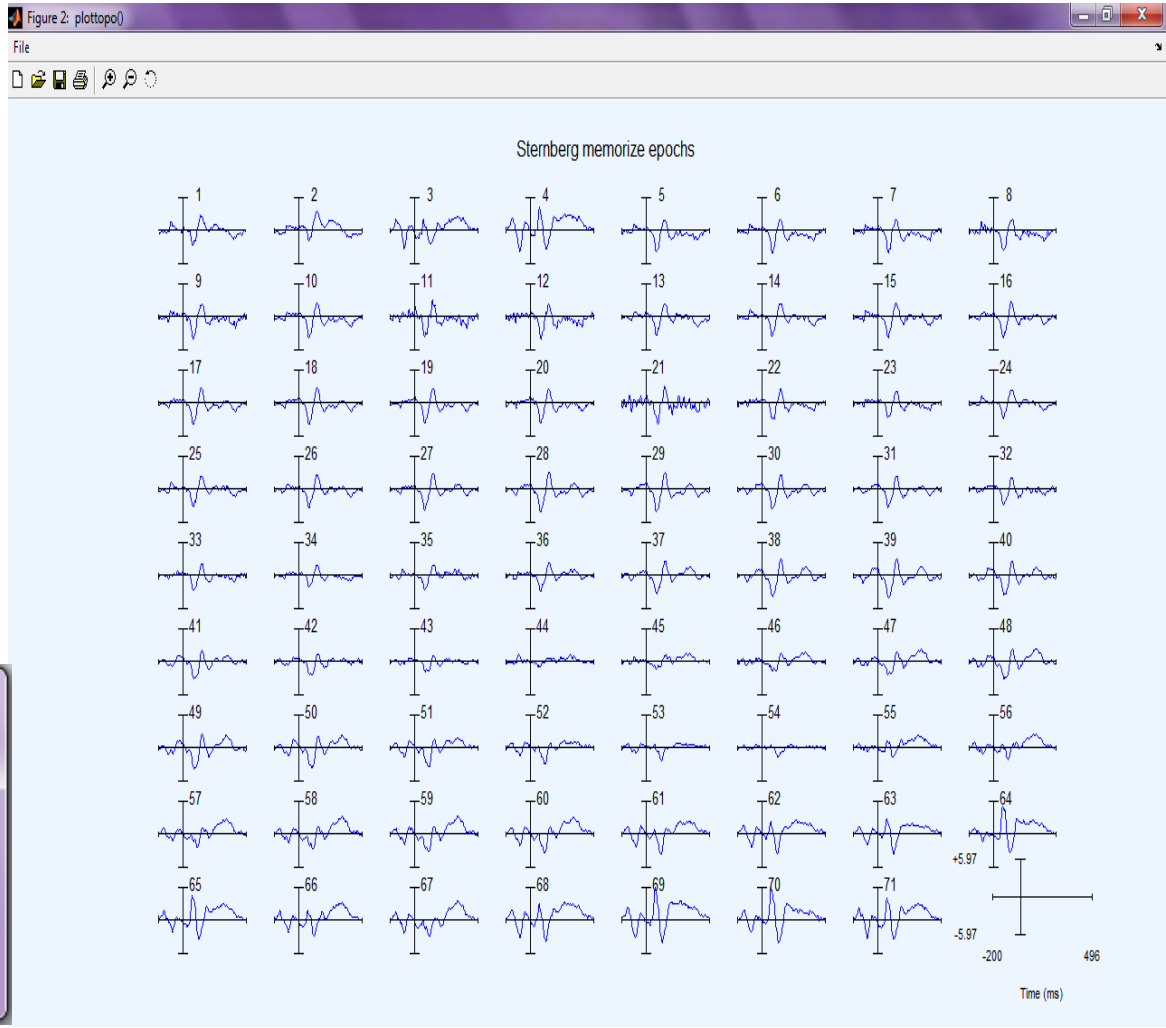
Analysis of channel ERPs



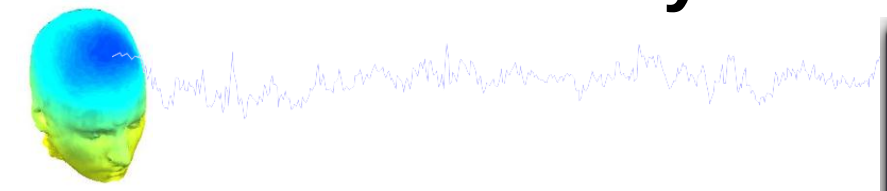


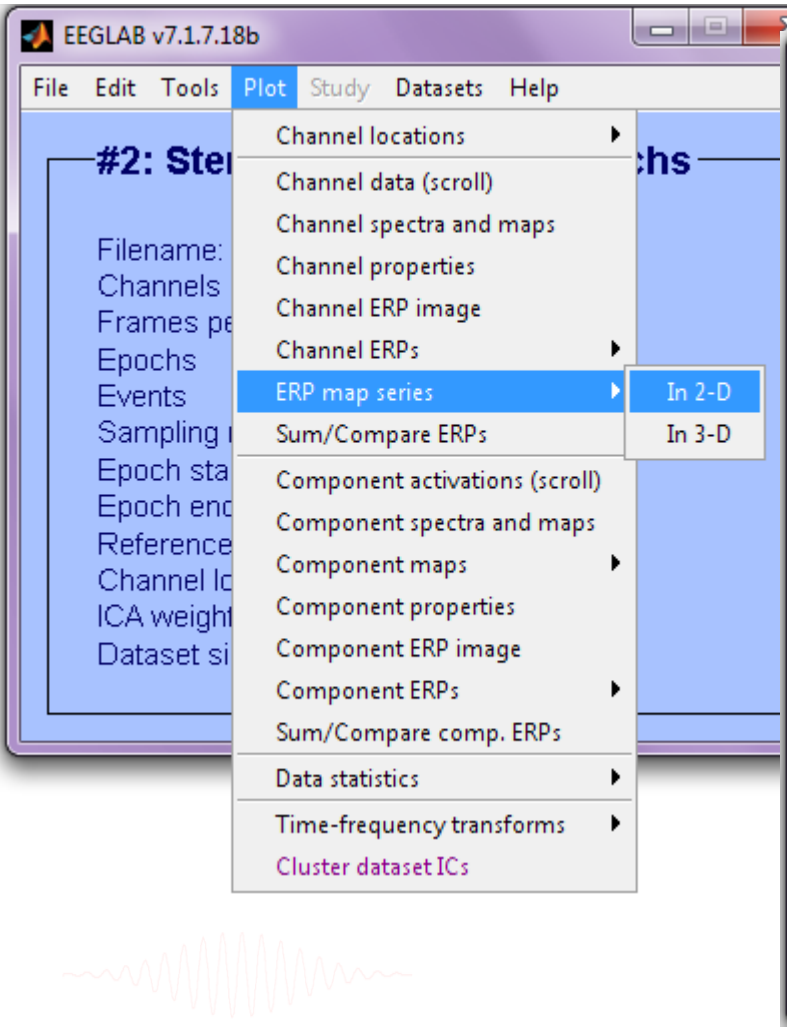
Channel ERP in rectangular array

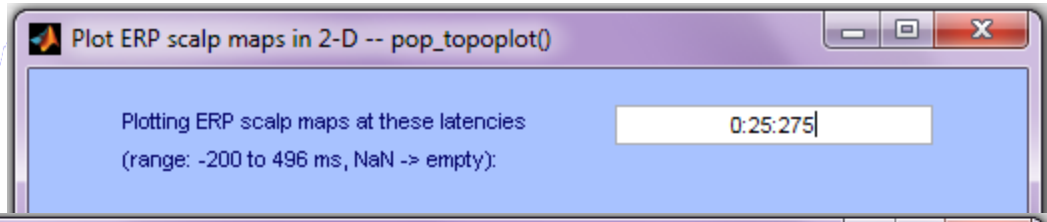


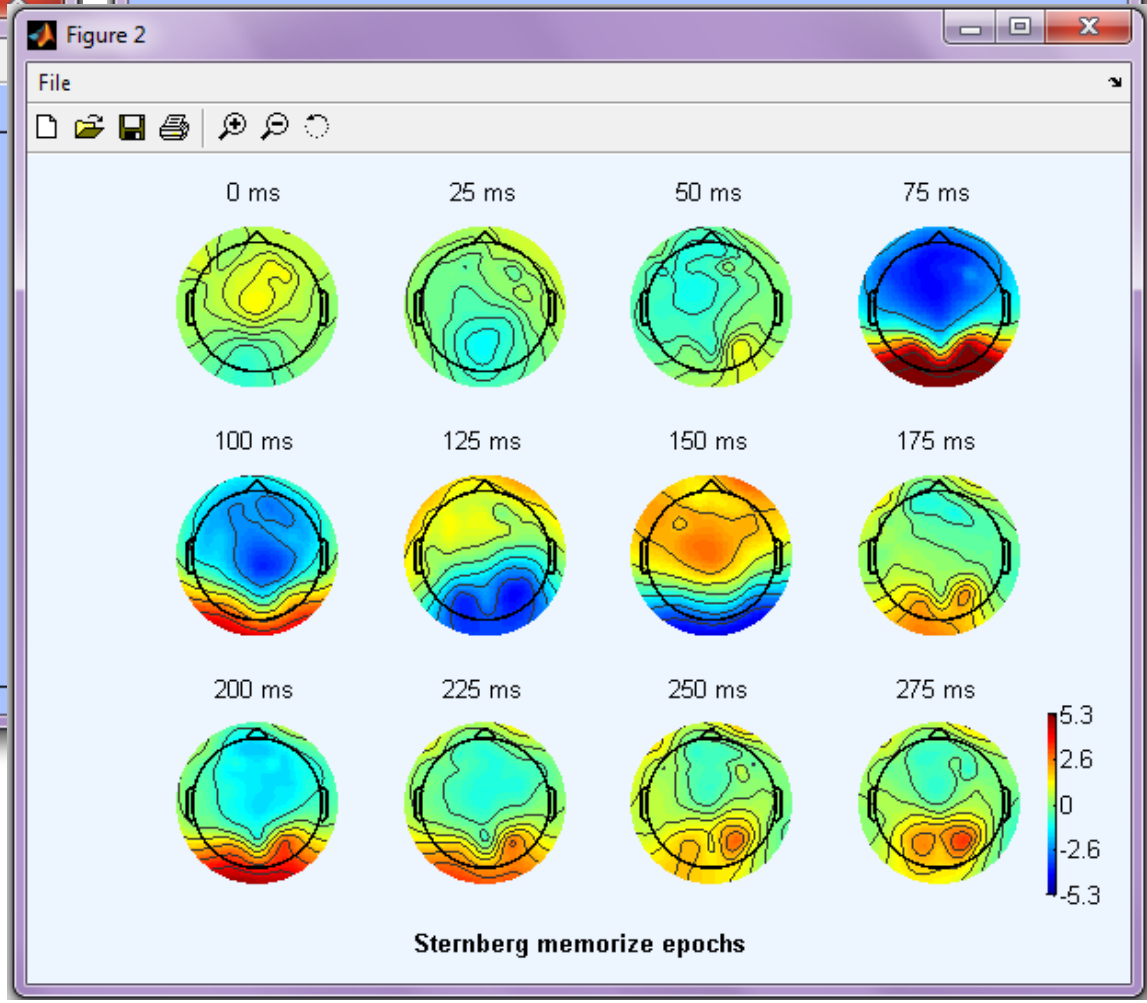


Analysis of channel ERPs



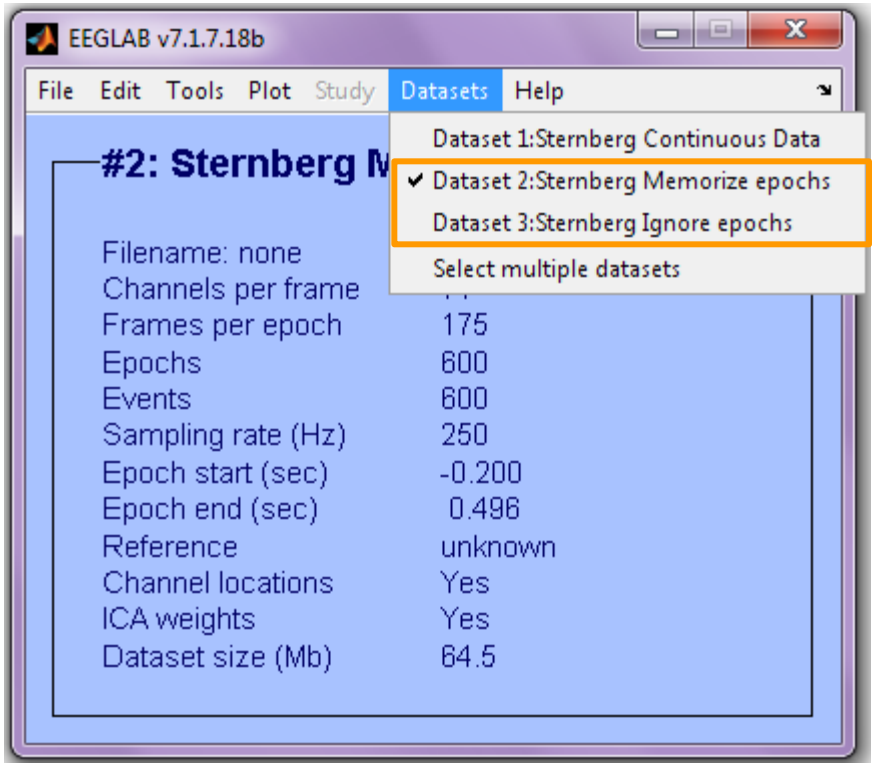






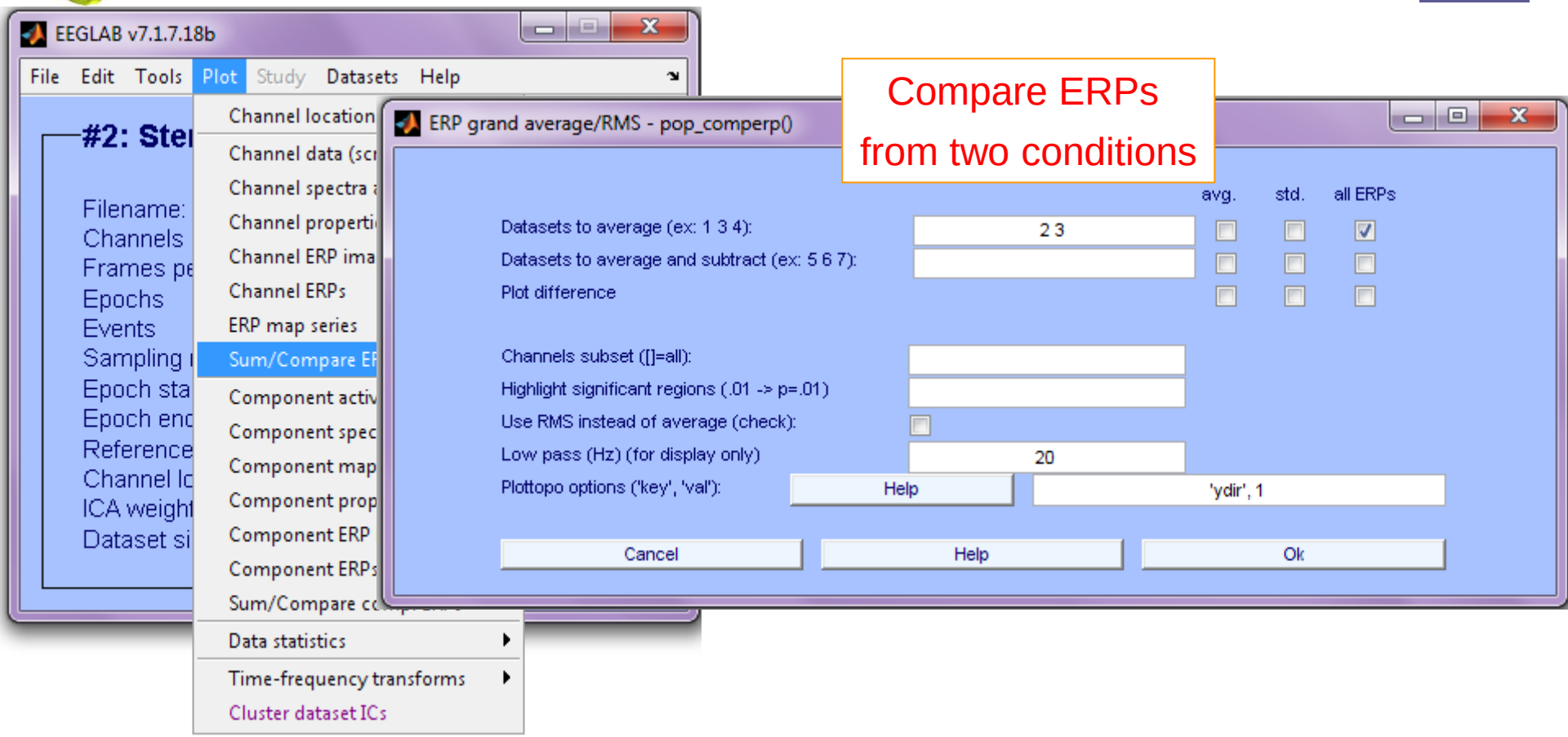

```
pop_topoplot(EEG,1,[0:25:275],'Memorize',[3 4],0,'electrodes','off');
```

Compare ERPs across conditions



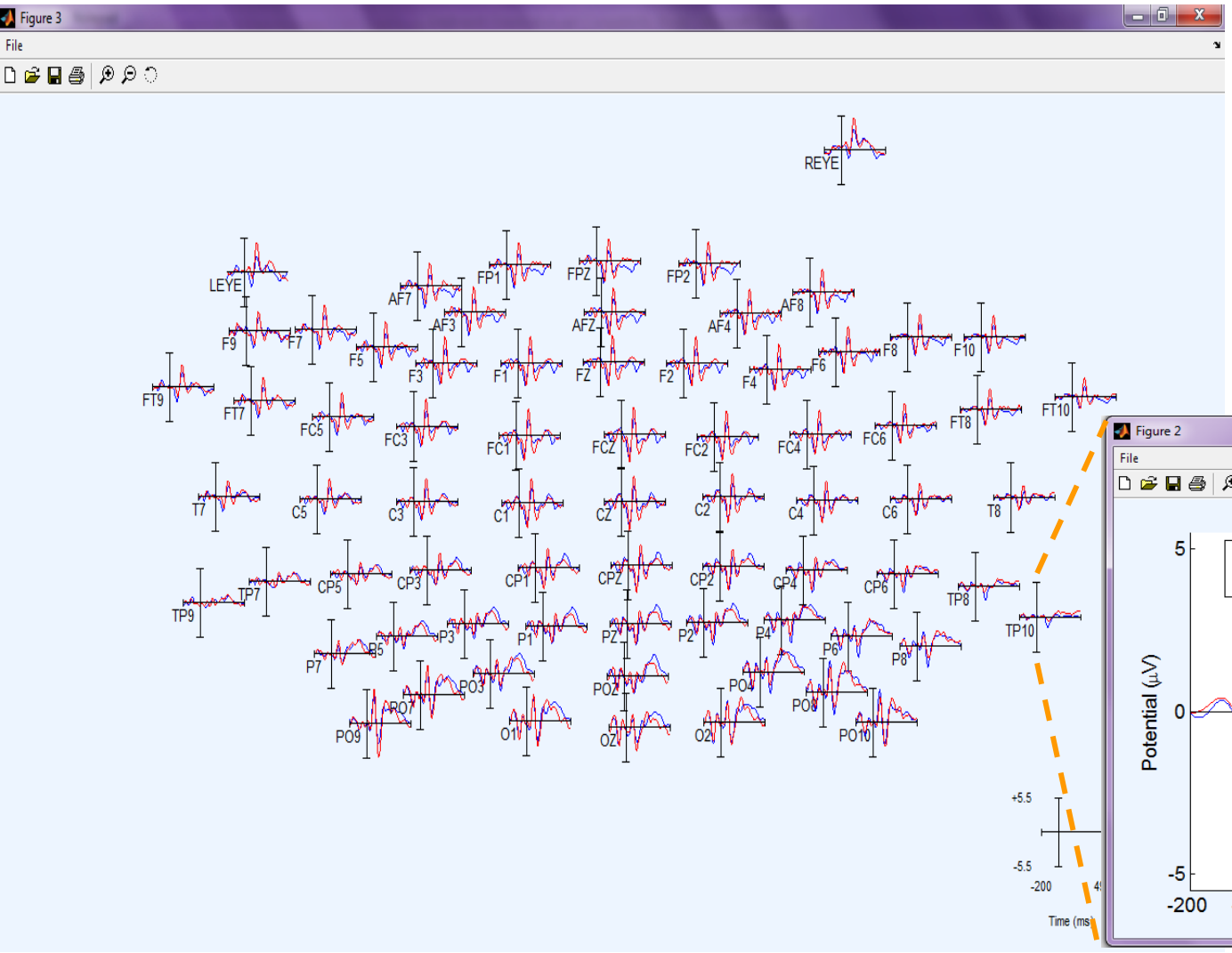
How do 'Memorize' and 'Ignore' ERPs differ?

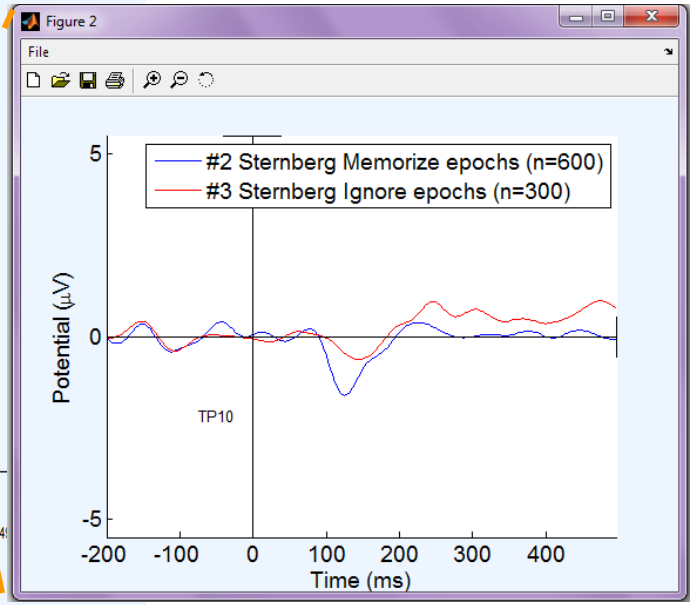
Compare ERPs across conditions




```
>>pop_comperp(ALLEEG,1,[2 3],[],'addavg','off','addstd','off', ...
  'addall','on','diffavg','off','diffstd','off','lowpass',20, ...
  'tplotopt',{'ydir',1});
```

Compare ERPs across conditions



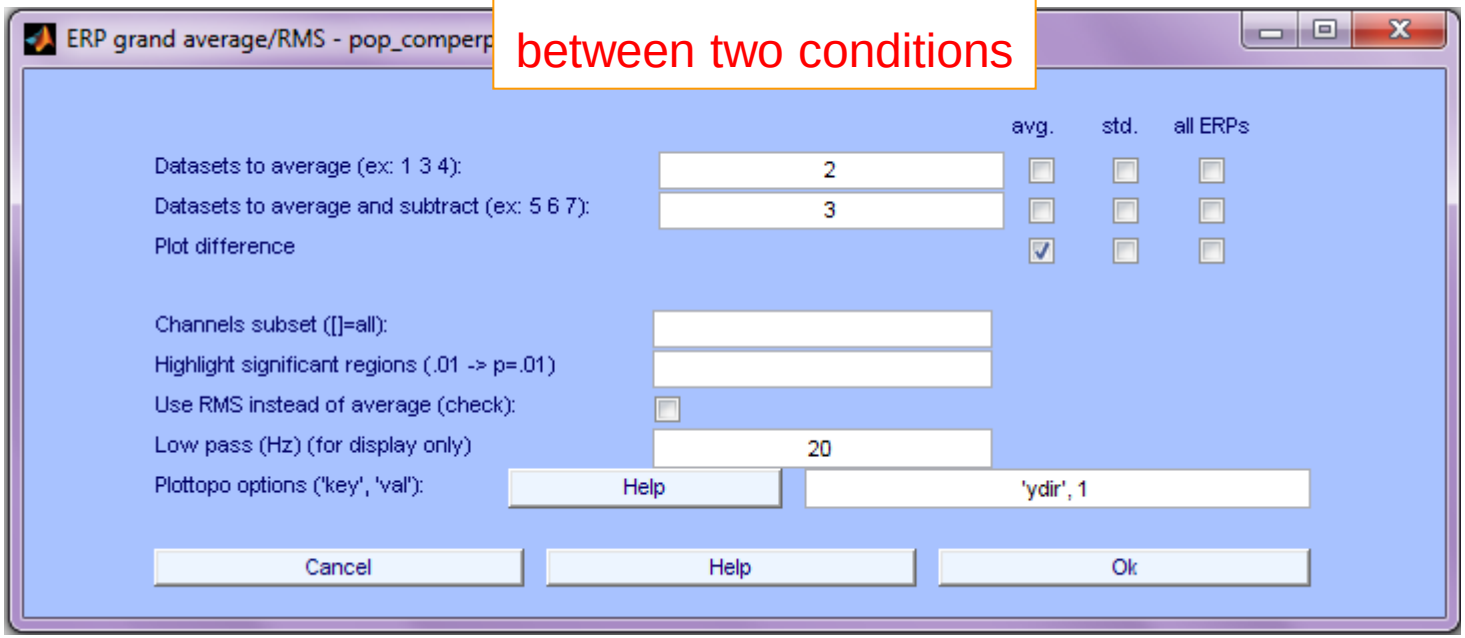


Click on an axis

to see larger image

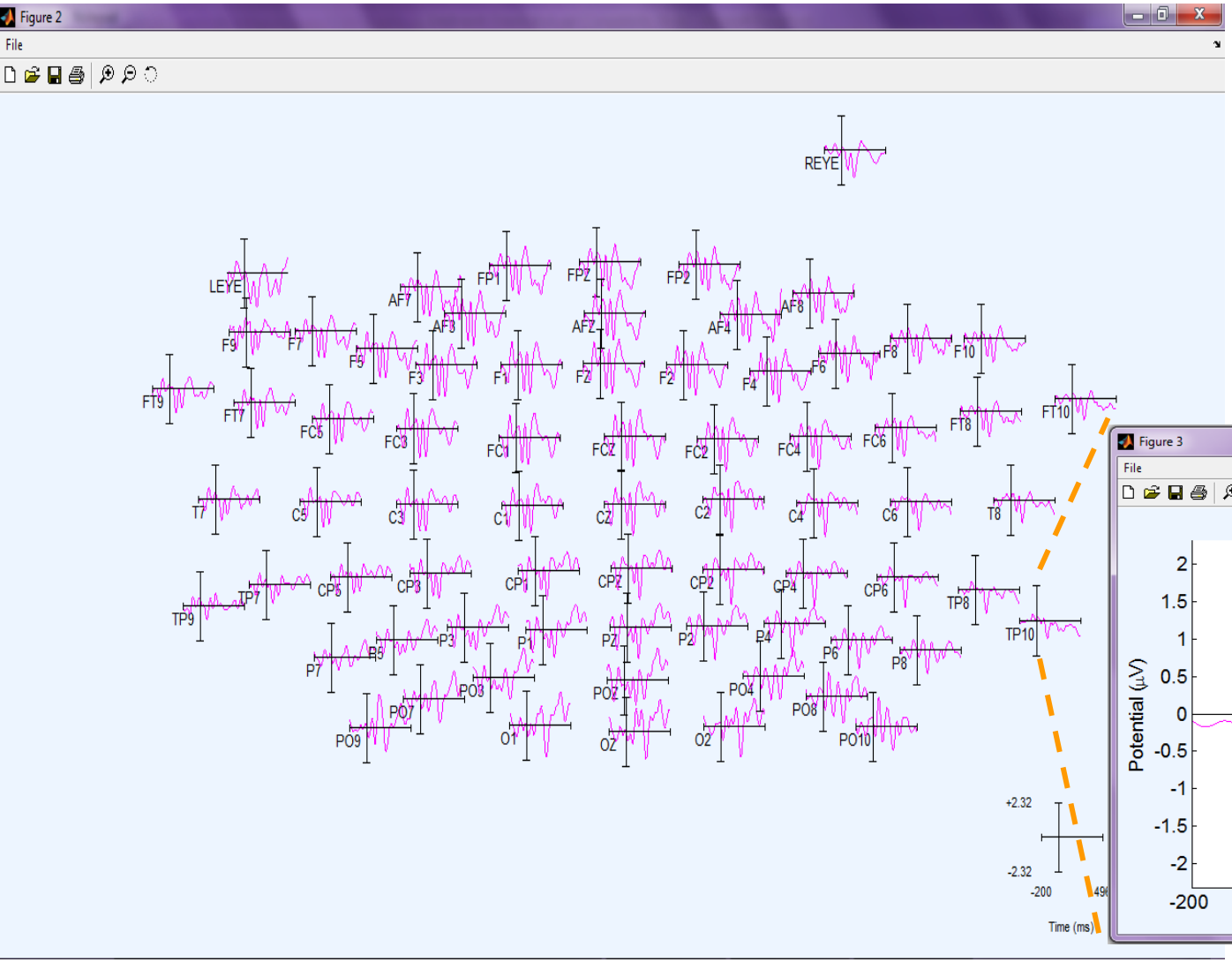
Analysis of ERP *differences*

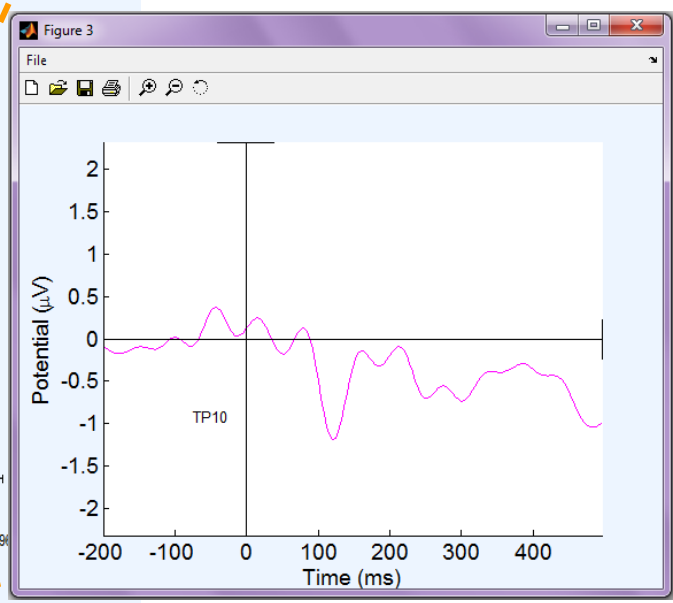
Plot *difference* between two conditions




```
>> pop_comperp(ALLEEG,1, 2, 3,'addavg','off',…
'addstd','off', 'diffavg','on','diffstd','off', …
'lowpass',20, 'tplotopt',{'ydir',1});
```

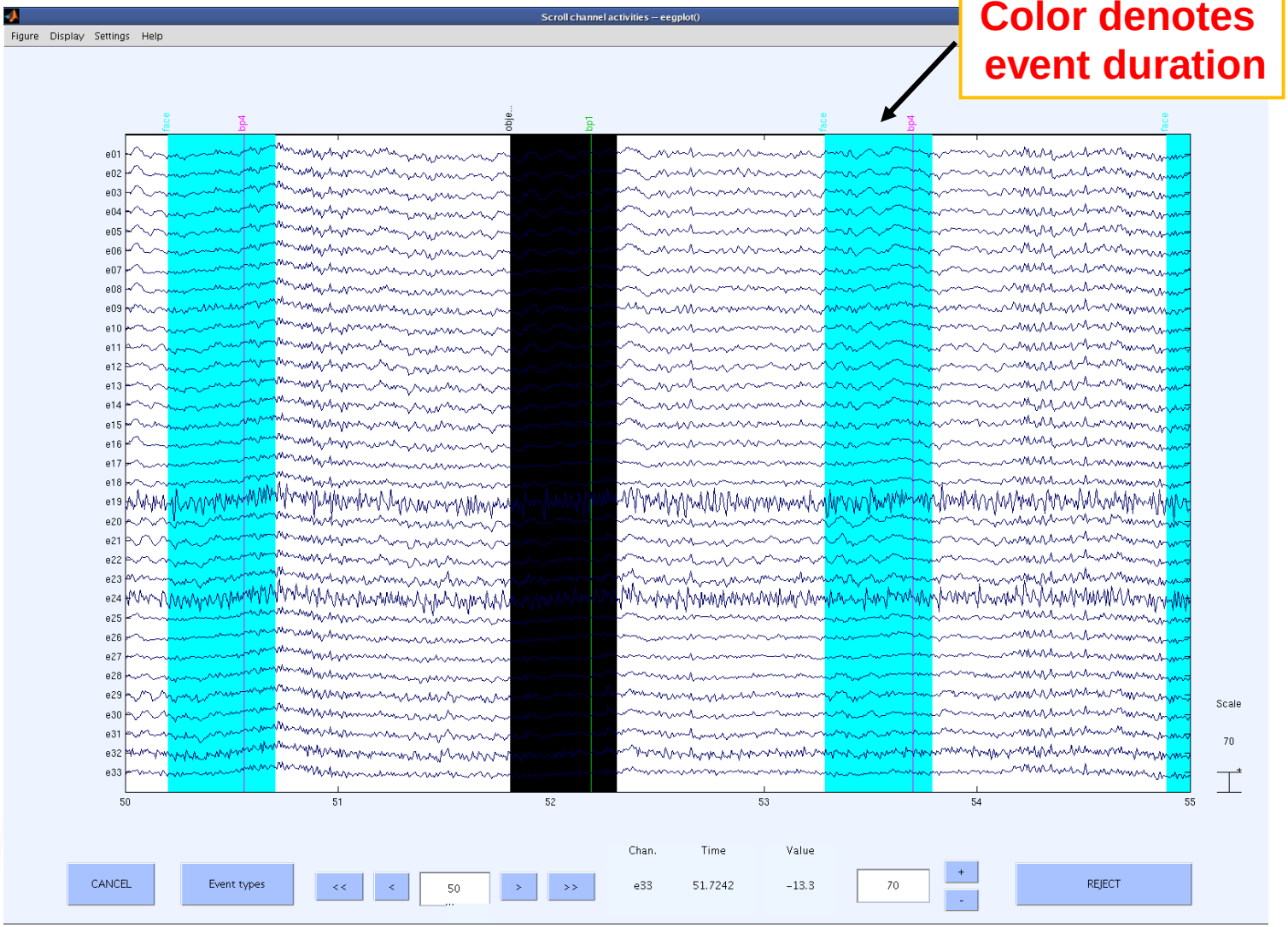
Analysis of ERP differences



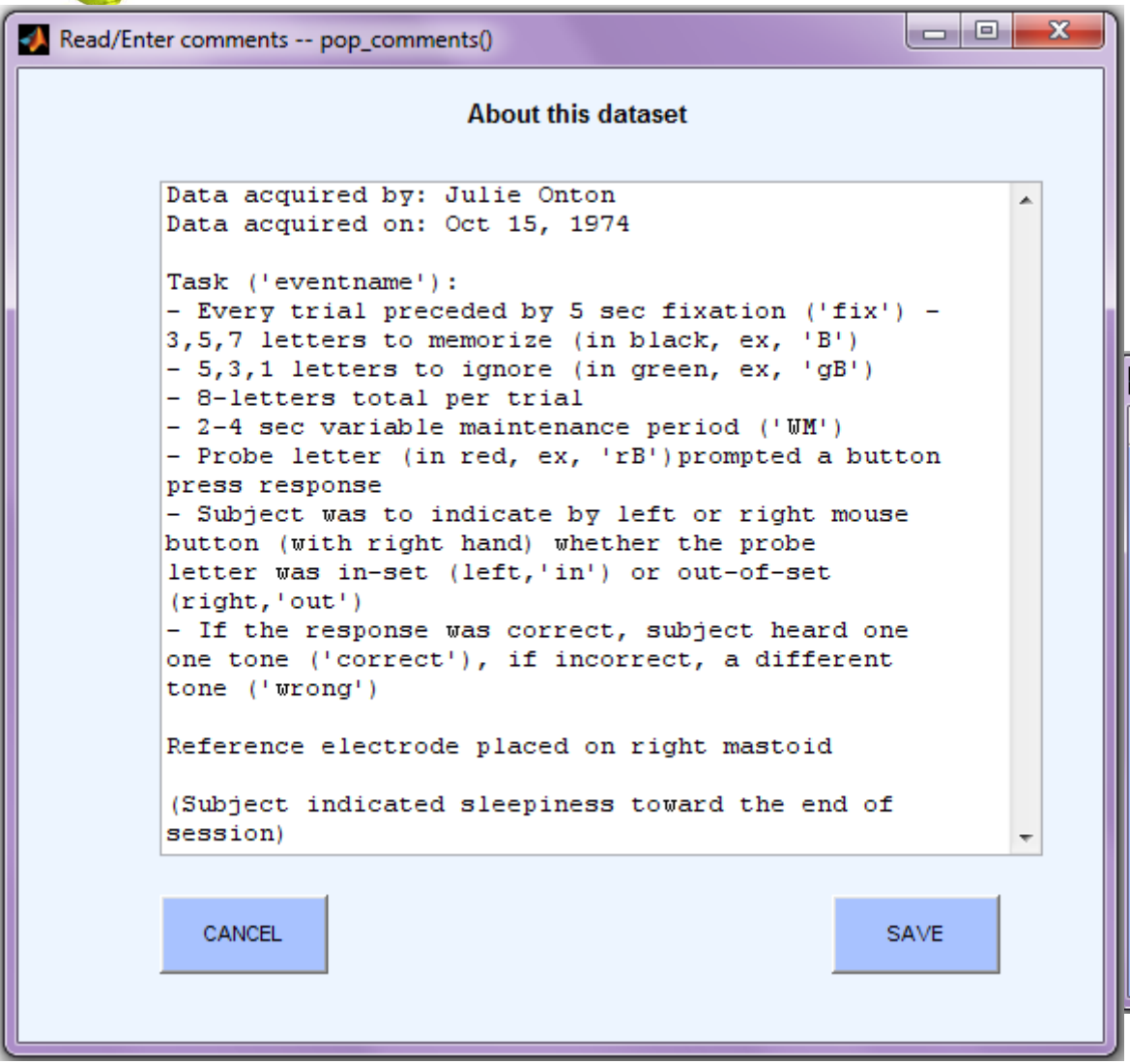


ERP
difference
between
2 conditions

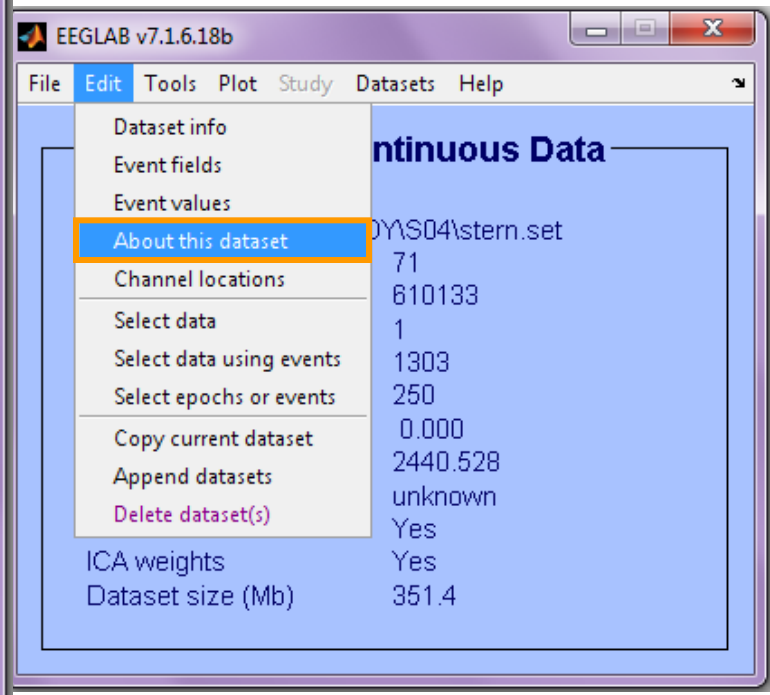
Event durations



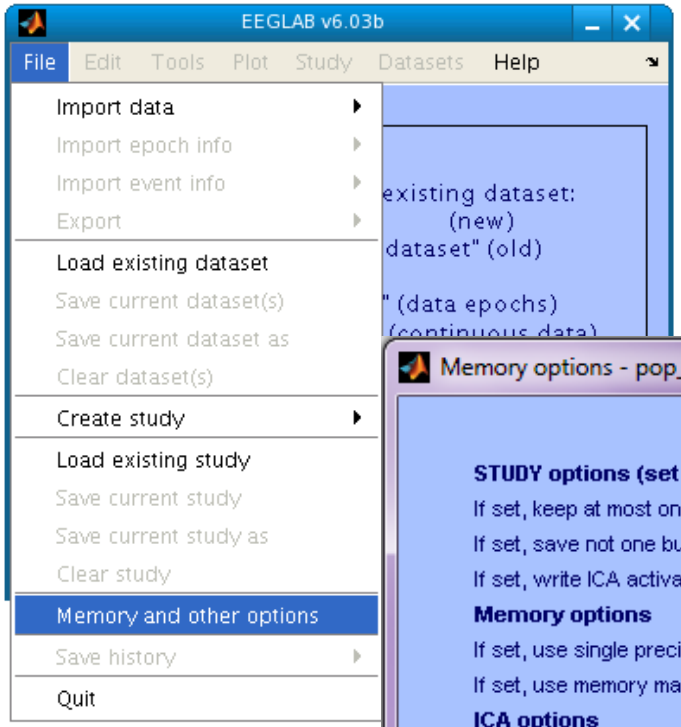
Comments in EEGLAB structure




```
>> EEG.comments
```



Memory options



Set when loading a STUDY