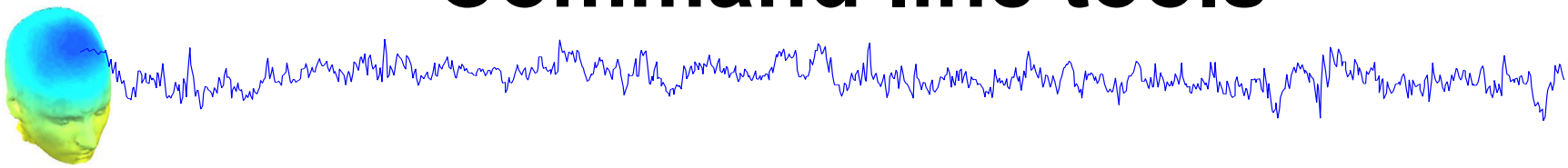
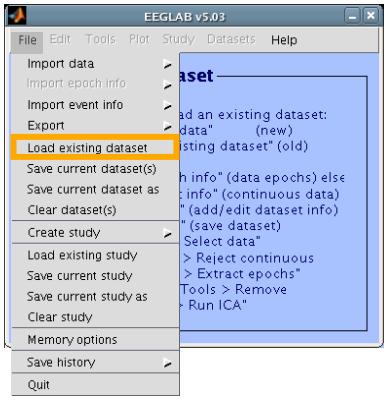
Command line tools

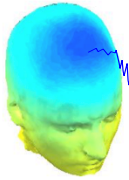


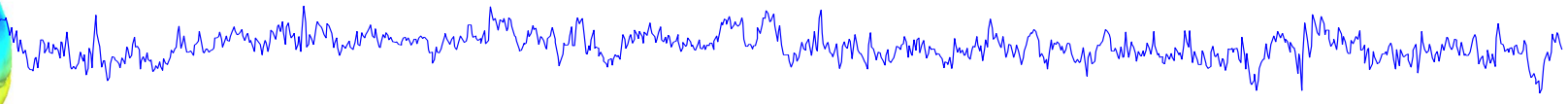


(Menus write both dataset and global history)

- Automated processing on groups of subjects (possibly on several processors).
- Richer options for plotting and processing functions (time-frequency decompositions, …)
- Selecting data/epoch based on event context
- Custom processing…

Using EEGLAB history for basic scripting





Task 1

Create a script from ‘eegh’ output

Task 2

Adapt your script with variables

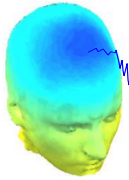
Task 3

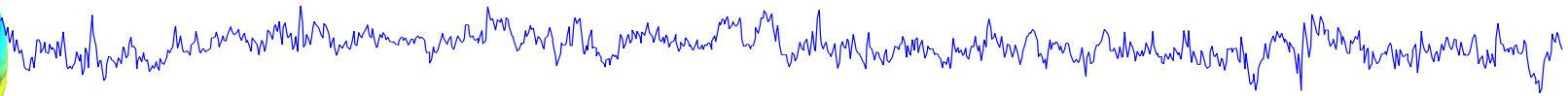
Create a Matlab function

Task 4

Exercise...

Using EEGLAB history for basic scripting





Task 1

Create a script from ‘eegh’ output

Task 2

Adapt your script with variables

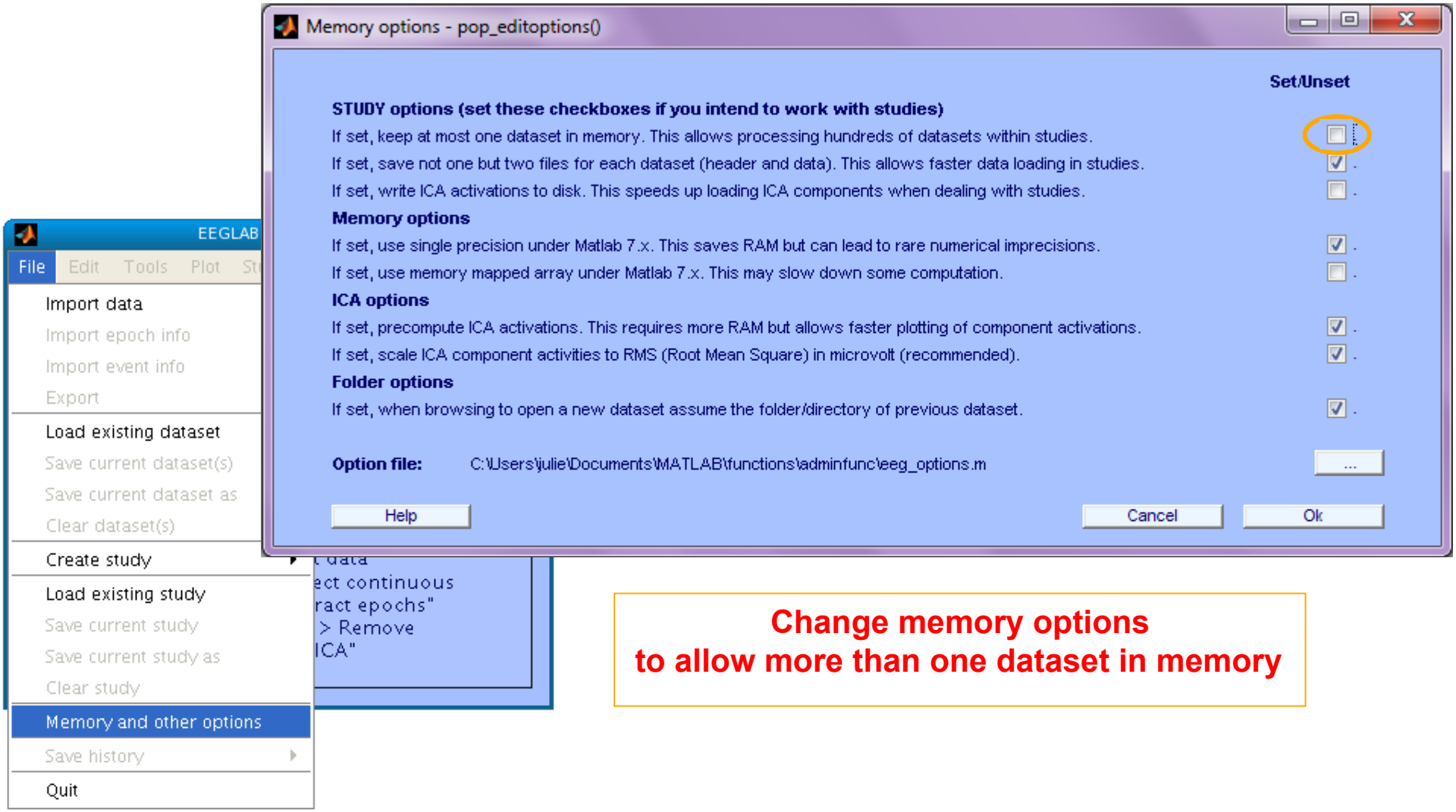
Task 3

Create a Matlab function

Task 4

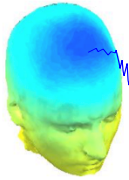
Exercise…

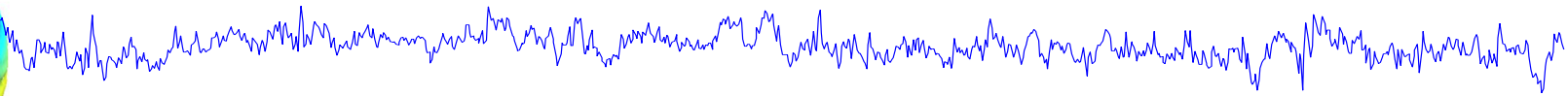
Memory options



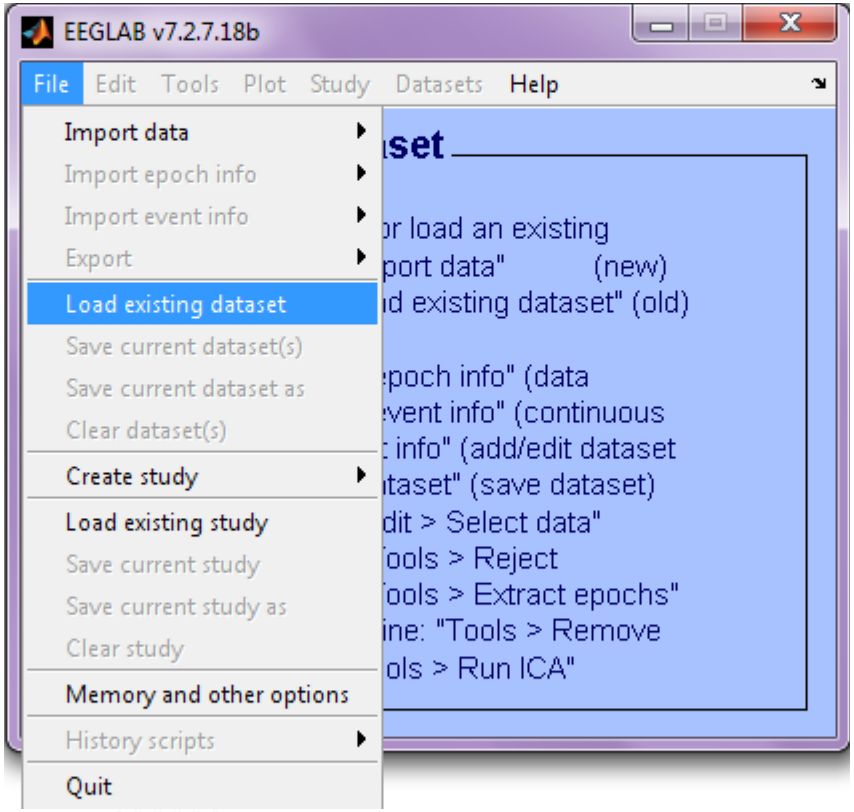
Change memory options
to allow more than one dataset in memory

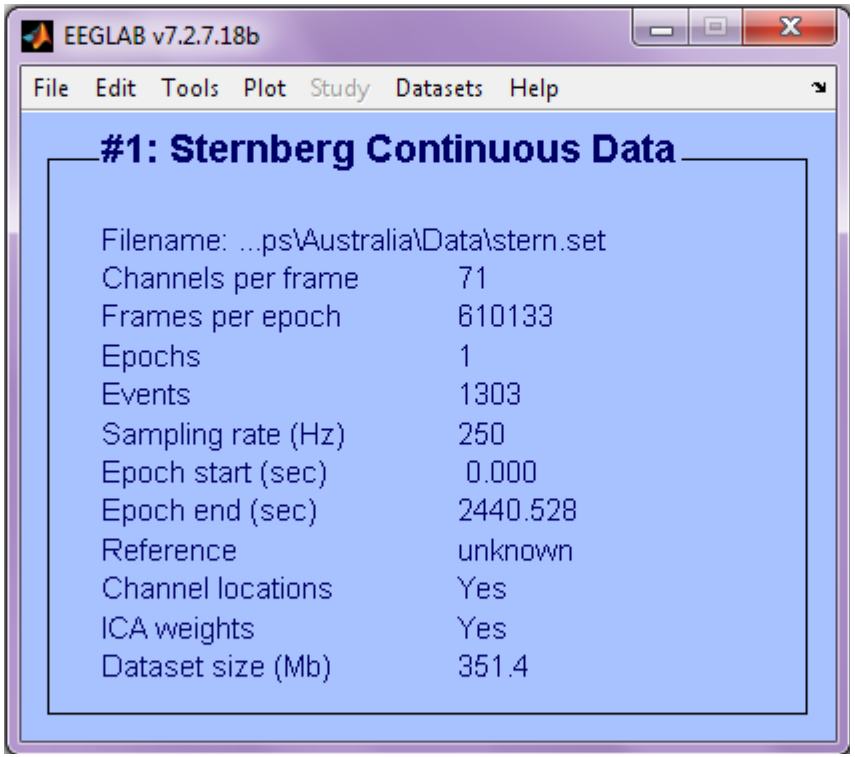
Create a script from ‘eegh’ output



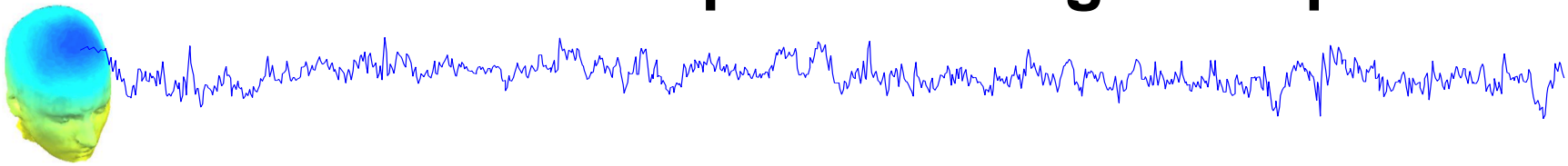


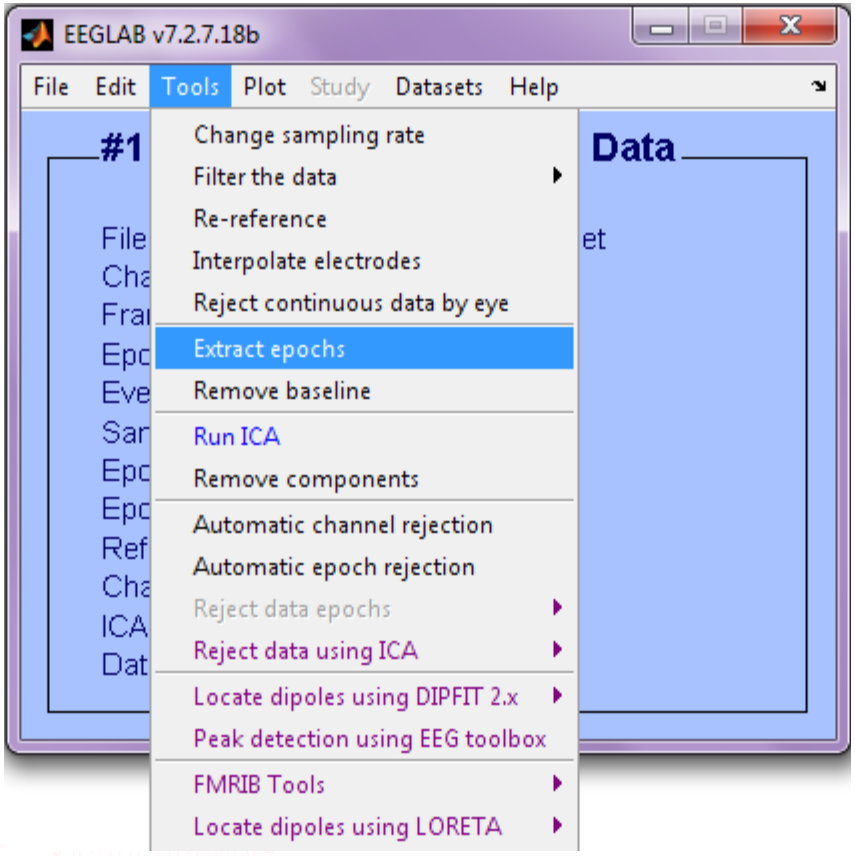
Start by loading a continuous dataset



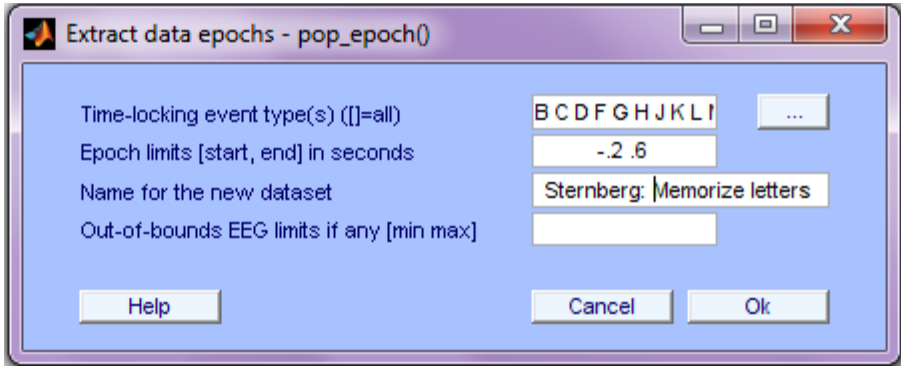


Create a script from ‘eegh’ output

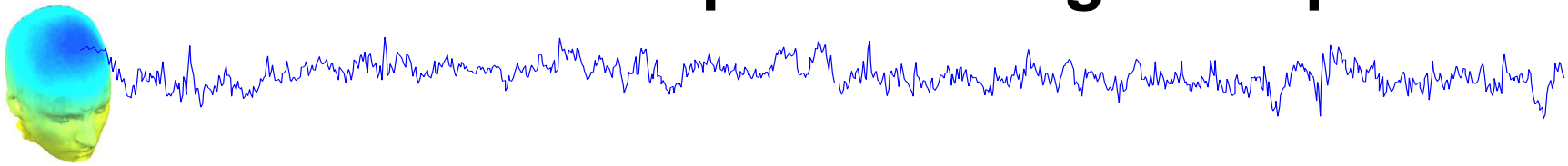




Epoch on Memorize letters



Create a script from ‘eegh’ output



Plot an IC ERP image

EEGLAB v7.2.7.18b

File Edit Tools Plot Study Datasets Help

#2: Ste

Filename:
Channels
Frames pr
Epochs
Events
Sampling
Epoch sta
Epoch en
Reference
Channel lo
ICA weigh
Dataset si

- Channel locations
- Channel data (scroll)
- Channel spectra and maps
- Channel properties
- Channel ERP image
- Channel ERPs
- ERP map series
- Sum/Compare ERPs
- Component activations (scroll)
- Component spectra and maps
- Component maps
- Component properties
- Component ERP image
- Component ERPs
- Sum/Compare comp. ERPs
- Data statistics
- Time-frequency transforms
- Cluster dataset ICs

Component ERP image -- pop_erpimage()

Component(s) 3
Project to channel #
Smoothing 10
Downsampling 1
Time limits (ms) -200 596

Figure title
Plot scalp map
Plot ERP
Plot colorbar
ERP limits
Color limits (see Help)

Sort/align trials by epoch event values
Epoch-sorting field | Event type(s) | Event time range | Rescale (no) | Align | Don't sort by value | Don't plot values

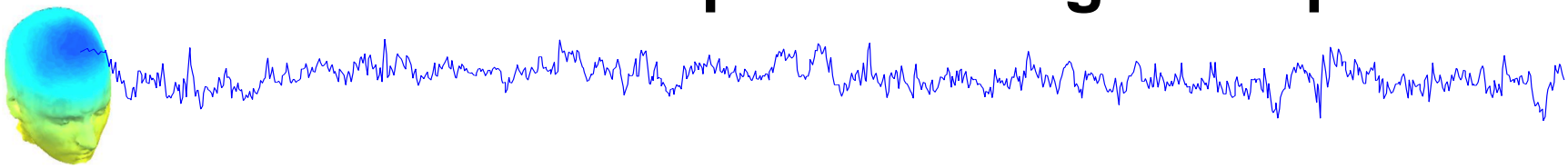
Sort trials by phase
Frequency (Hz | minHz maxHz) | Percent low-amp. trials to ignore | Window center (ms) | Wavelet cycles (3)

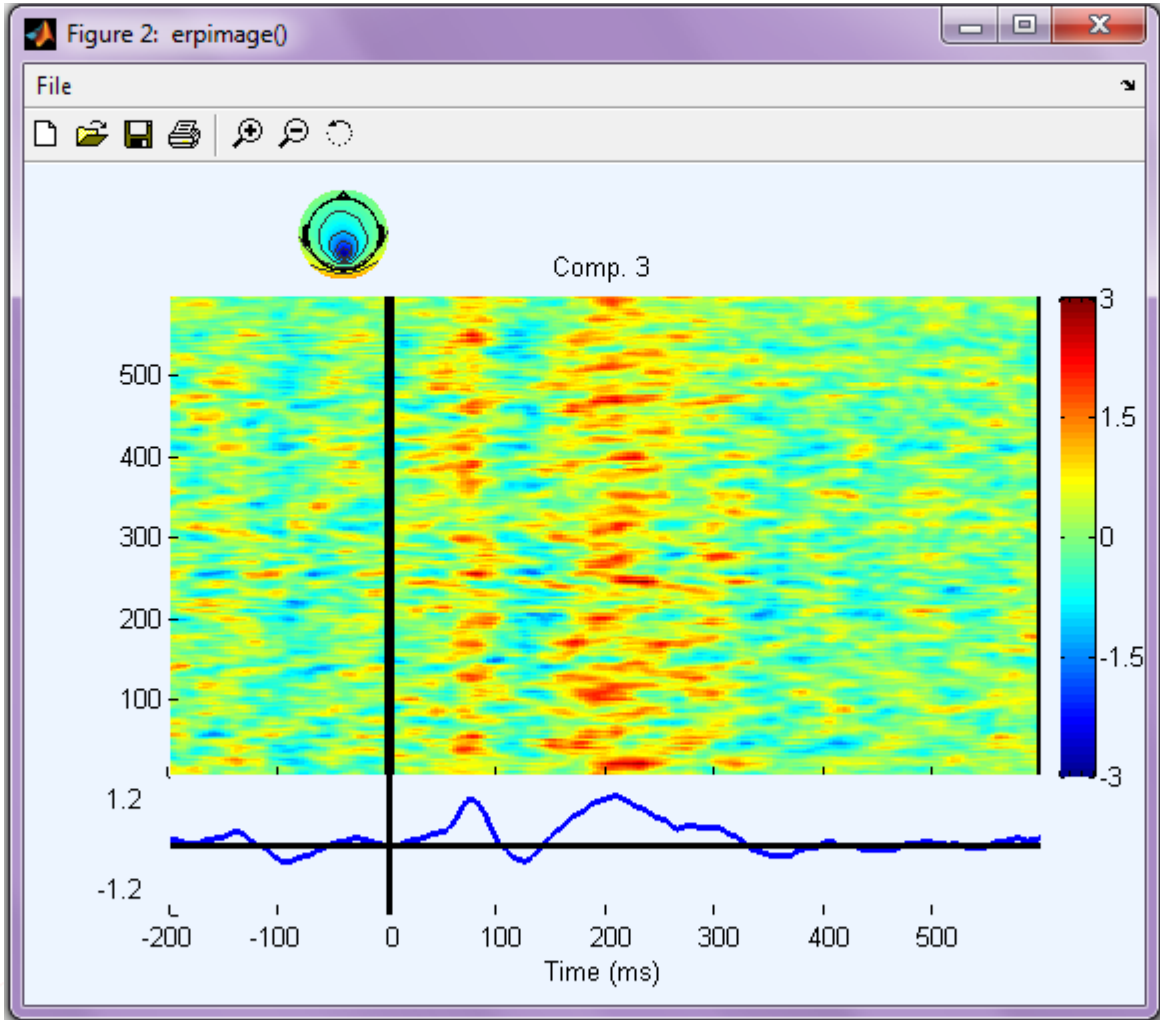
Inter-trial coherence options
Frequency (Hz | minHz maxHz) | Signif. level (<0.20) | Amplitude limits (dB) | Coher limits (<=1) | Image amps (Requires signif.)

Other options
Plot spectrum (minHz maxHz) | Baseline ampl. (dB) | Mark times (ms) | More options (see >> help erpimage)

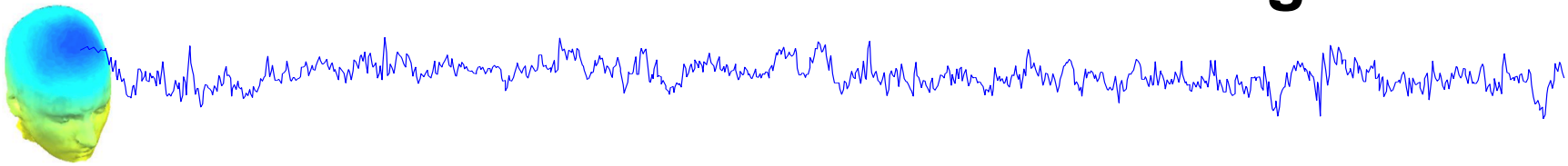
Help | Cancel | Ok

Create a script from ‘eegh’ output





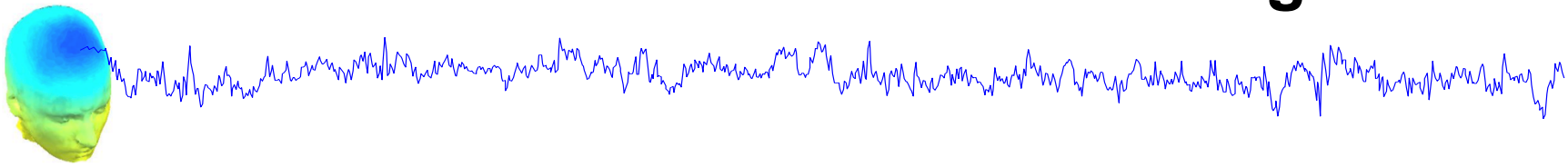
Retrieve commands from eegh



Write a script to do this:

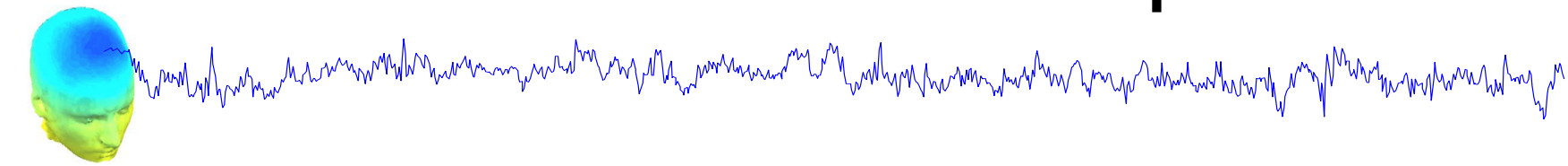
```
>> eegh
```

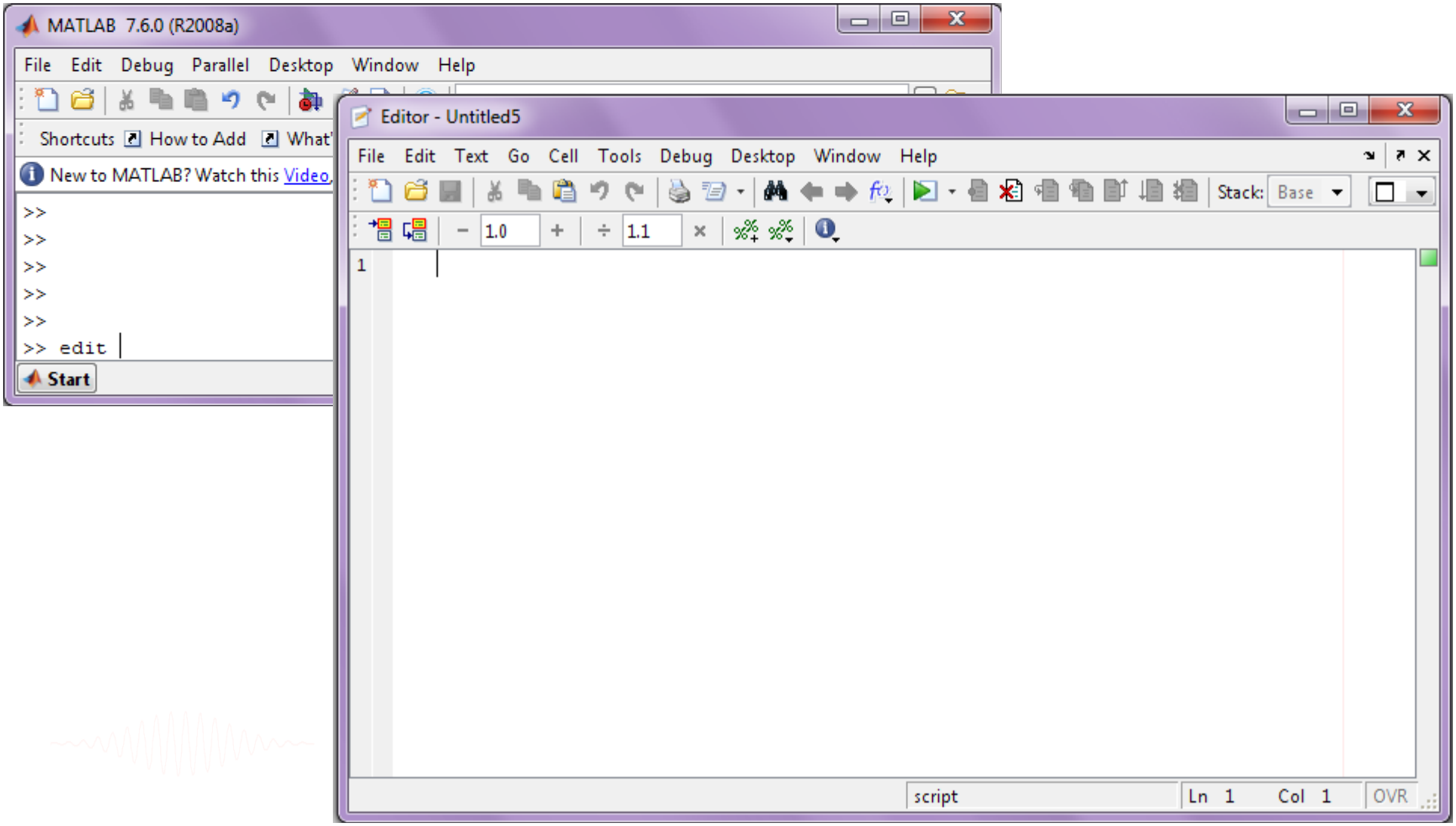
Retrieve commands from eegh



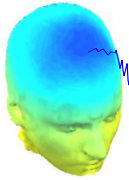
```
>> eegh
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

EEG = pop_loadset('filename', 'stern_125Hz.set');
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, 0);

EEG = pop_epoch( EEG, {'B'  'C'  'D' ... }, [-0.2 0.6], 'newname',
'Memorize epochs', 'epochinfo', 'yes');
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, 1);
EEG = pop_rmbase( EEG, [-200 0]);
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);

figure; pop_erpimage(EEG,0, [3],[],'Comp. 3',10,1,{},
[],'','yerplabel', '', 'erp', 'on', 'cbar', 'on','topo',
{mean(EEG.icawinv(:,[3]),2) EEG.chanlocs EEG.chaninfo });
```

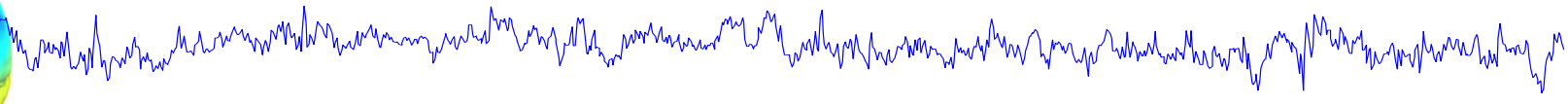
Create a Matlab script





Create a Matlab script





Copy and paste from Matlab window:

Editor - Untitled5*

File Edit Text Go Cell Tools Debug Desktop Window Help

Stack: Base

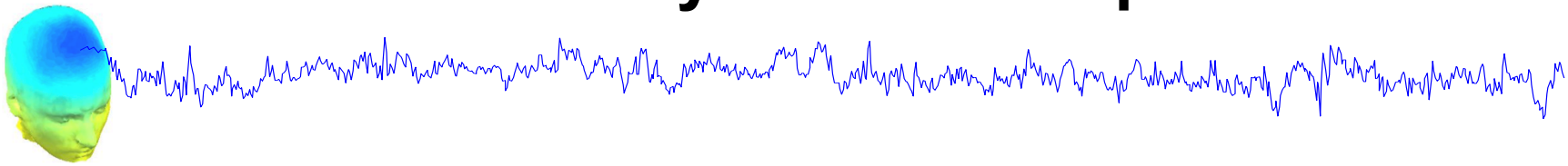
1.0 1.1

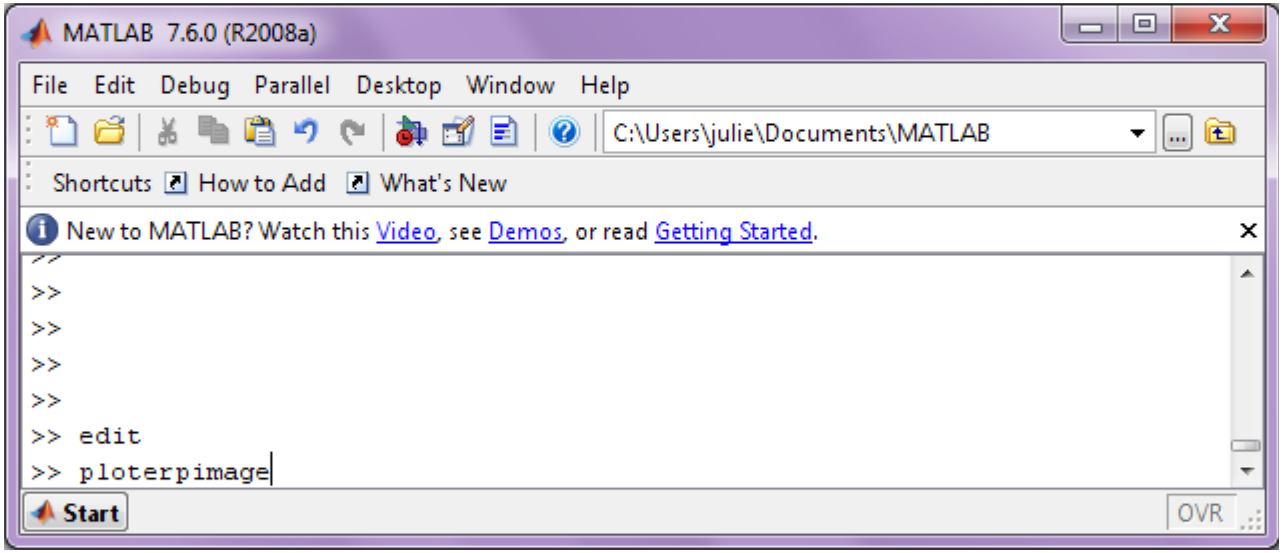
```
1    [ALLEEG EEG CURRENTSET ALLCOM] = eeglab;
2    EEG = pop_loadset( 'filename', 'stern.set', 'filepath', 'C:\\Users\\julie\\Docum
3    [ALLEEG, EEG, CURRENTSET] = eeg_store( ALLEEG, EEG, 0 );
4    EEG = pop_epoch( EEG, {  'B'  'C'  'D'  'F'  'G'  'H'  'J'  'K'  'L'  'M'  'N'
5    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, 'gui', 'off');
6    EEG = pop_rmbase( EEG, [-200    0]);
7    [ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
8    figure; pop_erpimage(EEG,0, [3],[],'Comp. 3',10,1,{},[],'' , 'yerplabel','', 'er
```

script Ln 7 Col 51 OVR

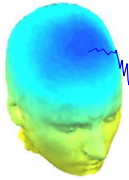
Save as ‘ploterpimage.m‘
In MATLAB folder

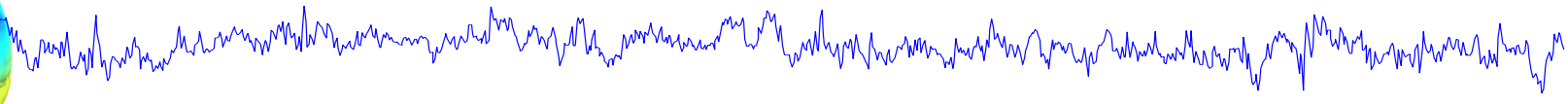
Run your new script



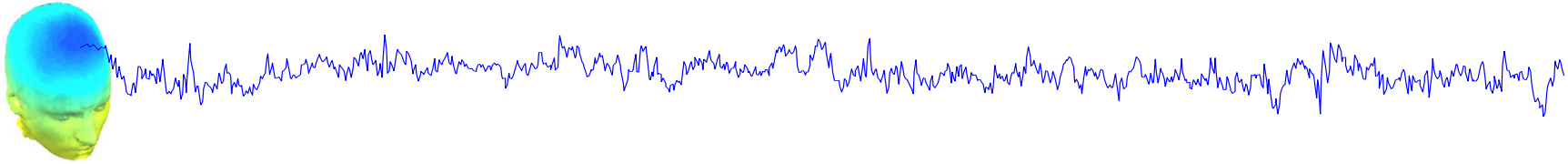


Exercise page 1



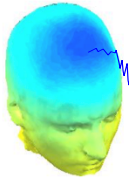


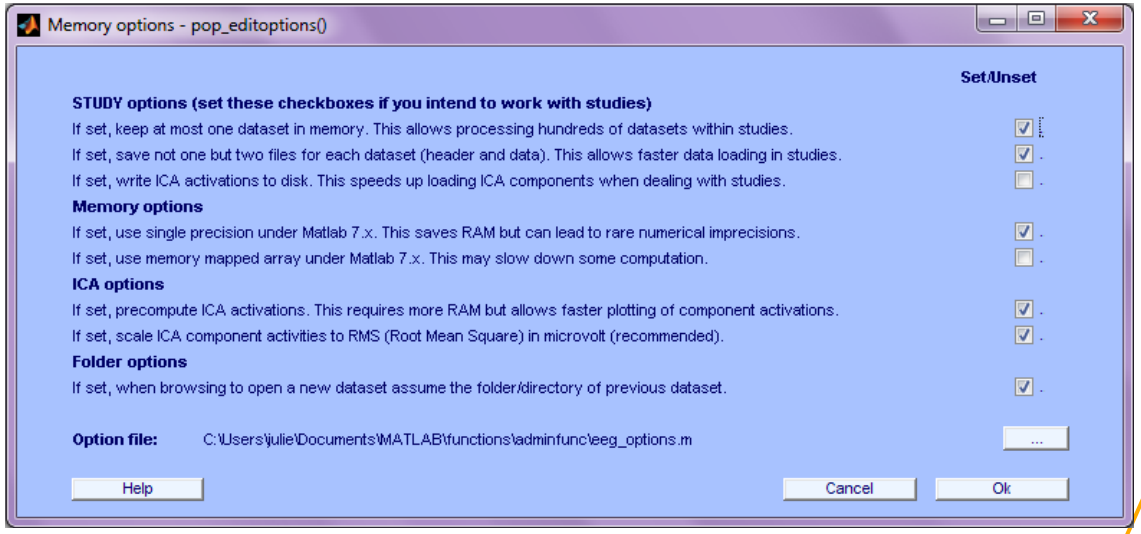
```
>> eeglab

% load dataset,
% epoch on 'memorize letter' B, C, etc...
% plot erpimage for component 3

>> eegh

% open Matlab editor

>> edit

% copy & paste eegh results into a new
% file and save it (ploterpimage.m)

>> clear
>> close all
>> ploterpimage
>> eeglab redraw
```



Advanced Scripting in EEGLAB

STUDY scripts



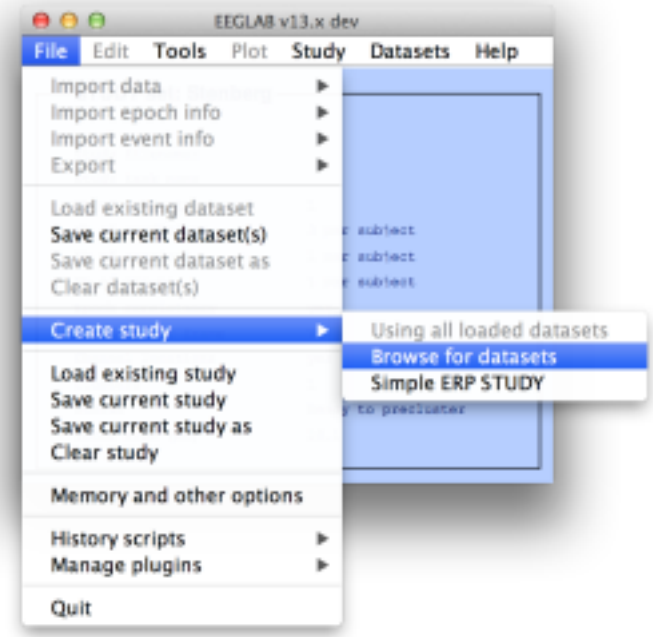


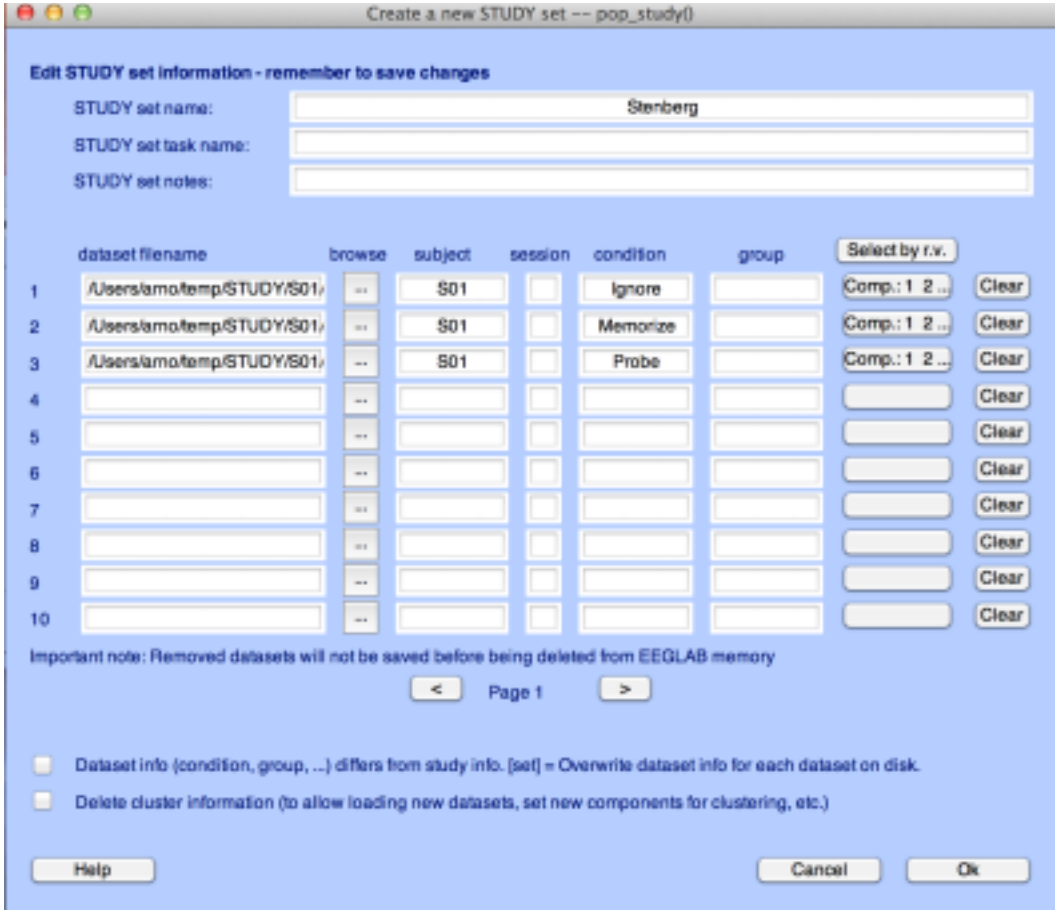
Most important option:

- Allows only one dataset to be loaded at once.
- Most STUDYs are too big to have all data loaded at once.

```
% Set memory options:
pop_editoptions( 'option_storedisk', 1);
```

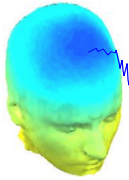
Edit dataset info

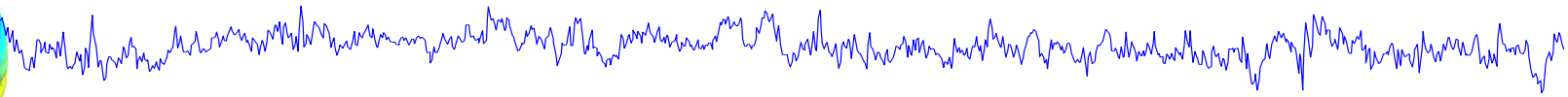





```
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name','Stenberg','commands',{{'index' 1
'load' '/data/STUDY/S01/Ignore.set'} {'index' 2 'load' '/data/S01/Memorize.set'}
{'index' 3 'load' '/data/S01/Probe.set'} {'index' 1 'subject' 'S01'} {'index' 2
'subject' 'S01'} {'index' 3 'subject' 'S01'} {'index' 1 'condition' 'Ignore'} {'index' 2
'condition' 'Memorize'} {'index' 3 'condition' 'Probe'}},'updatedat','off' );
```

Looking a the function that create STUDY

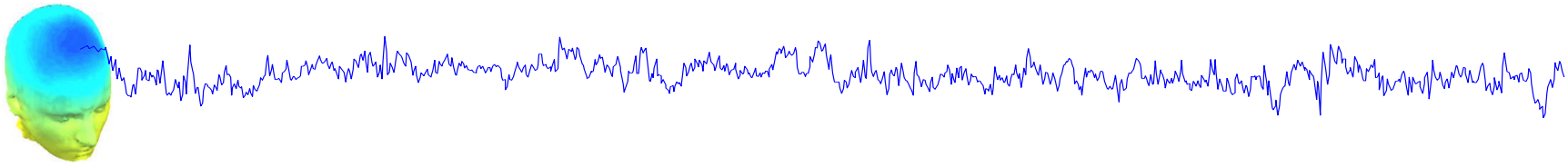




```
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name','Stenberg','commands',
     {{'index' 1 'load' '/data/STUDY/S01/Ignore.set'} …
     {'index' 2 'load' '/data/S01/Memorize.set'} …
     {'index' 3 'load' '/data/S01/Probe.set'} …
     {'index' 1 'subject' 'S01'} …
     {'index' 2 'subject' 'S01'} …
     {'index' 3 'subject' 'S01'} …
     {'index' 1 'condition' 'Ignore'} …
     {'index' 2 'condition' 'Memorize'} …
     {'index' 3 'condition' 'Probe'}},'updatedat','off' );
```

```
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name','Stenberg','commands',
     {{'index' 1 'load' '/data/STUDY/S01/Ignore.set' 'subject' 'S01' 'condition' 'Ignore'} …
     {'index' 2 'load' '/data/S01/Memorize.set' 'subject' 'S01' 'condition' 'Memorize'} …
     {'index' 3 'load' '/data/S01/Probe.set' 'subject' 'S01' 'condition' 'Probe'} …
     },'updatedat','off' );
```

Exercice



```
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name','Stenberg','commands',
     {{'index' 1 'load' '/data/STUDY/S01/Ignore.set' 'subject' 'S01' 'condition' 'Ignore'} …
     {'index' 2 'load' '/data/S01/Memorize.set' 'subject' 'S01' 'condition' 'Memorize'} …
     {'index' 3 'load' '/data/S01/Probe.set' 'subject' 'S01' 'condition' 'Probe'} …
     },'updatedat','off' );
```

If not present, add it by hand because some dataset might not have it

1- Start EEGLAB and import the 3 datasets for Subject 1 (Ignore.set, Memorize.set and Probe.set) in a STUDY (menu Tools > Create STUDY > Browse for datasets)

2- Look in the history

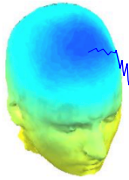
3- Copy to a script, add “eeglab redraw” at the end of your script

4- Restart Matlab, execute the script, look at your STUDY info and design (menu *STUDY > Edit STUDY* info and *STUDY > Select/Edit STUDY design*)

5- Modify the script to import subject 1 to 4

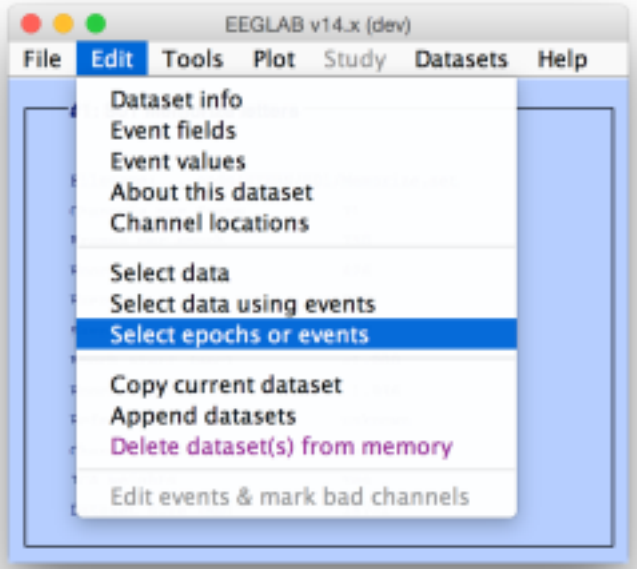
6- Restart Matlab, execute the script, look at your STUDY info and design

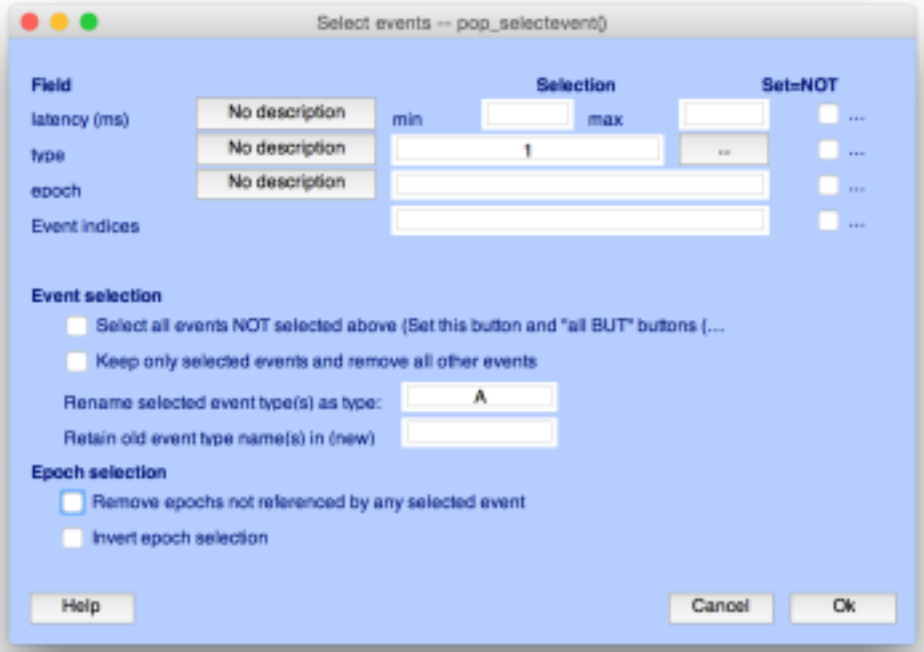
Redefining events



Adjusting latencies

```
for iEvent = 1:length(EEG.event)
    % shift by 16 samples (or 53.3ms at 200Hz) due to filter delay
    EEG.event(iEvent).latency = EEG.event(iEvent).latency + 16;
end;
EEG.saved = 'no';
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG);
eeglab redraw
```



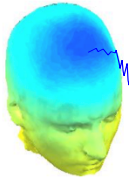


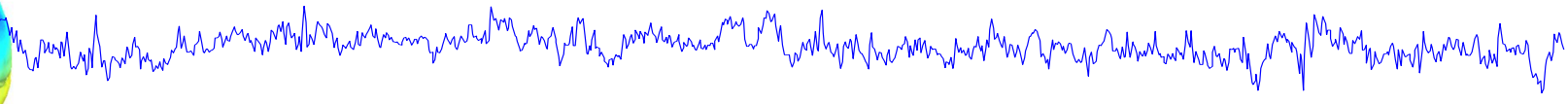
Renaming events

```
for iDat = 1:length(ALLEEG)
    EEG = ALLEEG(iDat);
    % rename events
    EEG = pop_selectevent( EEG, 'type',1,'renametype','A','deleteevents','off');
    EEG = pop_selectevent( EEG, 'type',2,'renametype','B','deleteevents','off');
    EEG = pop_selectevent( EEG, 'type',3,'renametype','C','deleteevents','off');
    EEG = pop_selectevent( EEG, 'type',4,'renametype','D','deleteevents','off');
    EEG = pop_selectevent( EEG, 'type',8,'renametype','rt','deleteevents','off');
    EEG.saved = 'no';
    [ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, iDat);
end;
eeglab redraw
```


```
EEG.event(4)
        type: 'square'
    position: 2
     latency: 424
     urevent: 1
       epoch: 2

EEG.event(1)
        type: 'square'
    position: 2
     latency: 129
     urevent: 1
       epoch: 1

EEG.urevent(1)
        type: 'square'
    position: 2
     latency: 129
```

Redefining events



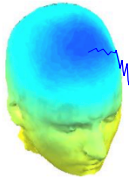


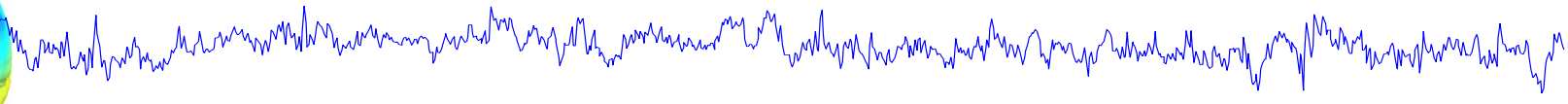
```
for iDat = 1:length(ALLEEG)
    TMPEEG = eeg_checkset(ALLEEG(iDat), 'loaddata'); % load data

    % 'B'  'C'  'D'  'F'  'G'  'H'  'J'  'K'  'L' ... -> Memorize
    % 'gB' 'gC' 'gD' 'gF' 'gG' 'gH' 'gJ' 'gK' 'gL' ... -> Ignore
    for iEvent = 1:length(TMPEEG.event)
          prevEvent = TMPEEG.event(iEvent).urevent-2;
          if prevEvent > 2 && TMPEEG.urevent(prevEvent).type(1) == 'g'
             TMPEEG.event(iEvent).prevEvent = 'ignore';
          else TMPEEG.event(iEvent).prevEvent = 'memorize';
          end;
    end;

    TMPEEG.saved = 'no'; % tag as not saved
    ALLEEG(iDat) = pop_saveset(TMPEEG, 'savemode', 'resave'); % resave data
end;

STUDY = std_maketrialinfo(STUDY, ALLEEG); % update STUDY
STUDY.saved = 'no';
[STUDY EEG] = pop_savestudy( STUDY, EEG, 'savemode', 'resave'); % resave STUDY
```

Precomputed files need to be recomputed after changing events.


Edit STUDY design -- pop_studydesign()
Select STUDY design
New
Rename
Delete
Design Matrix
memorize vs ignore
Design 2
Design 3
Design 4
Resave STUDY
Edit selected design
Independent variables
New
Import
Edit
Delete
Categorical variable: condition - Values (probe - ignore
Categorical variable: prevEvent - Values (ignore - memo
Subjects
S07
S08
S09
S10
S11
S12
S13
Delete all pre-computed datafiles for this STUDY design
Web help
Cancel
Ok
Add variable
Select independent variable
prevEvent
reqtime
rt
stimulus
ttime
type
uncertainty1
Categorical variable
Select variable values
ignore
memorize
Combine selected values
Cancel
Ok
Design Matrix: Design 4
Select subject
Sort by
Value on Click
S01
Design Matrix
condition-probe
condition-ignore&memorize
prevEvent-ignore
prevEvent-memorize
constant
Trials
100
200
300
400
500
600
700
1
2
3
4
5

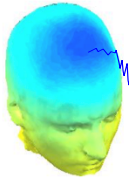
Load dataset info from commandline

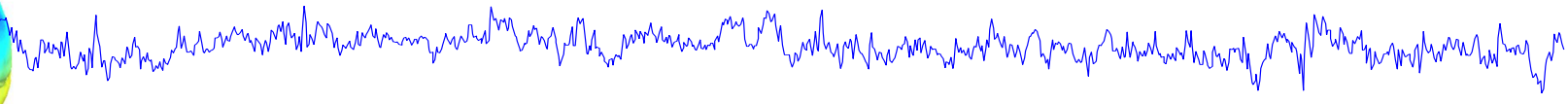




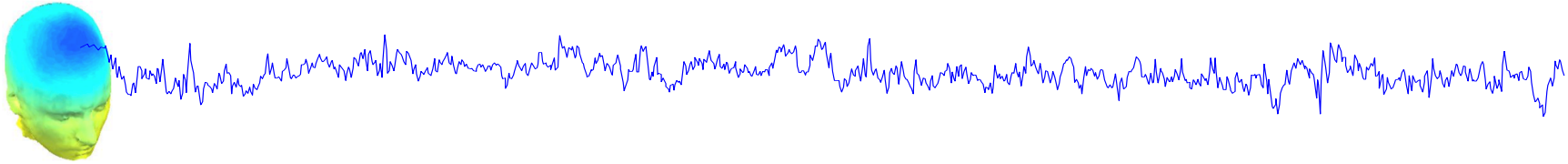
```
% Create Stern STUDY
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;
pop_editoptions( 'option_storedisk', 1);
subjects = {'S01' 'S02' 'S03' 'S04' 'S05' 'S06' 'S07' 'S08' 'S09' 'S10' 'S11' 'S12'};
filepath = '/Users/arno/temp/STUDY'; % XXXXX Change path here XXXXX
if ~exist(filepath), error('You need to change the path to the STUDY'); end;
commands = {}; % initialize STUDY dataset list

% Loop through all of the subjects in the study to create the dataset
for loopnum = 1:length(subjects) %for each subject
    IgnoreFile   = fullfile(filepath, subjects{loopnum}, 'Ignore.set');
    MemorizeFile = fullfile(filepath, subjects{loopnum}, 'Memorize.set');
    ProbeFile    = fullfile(filepath, subjects{loopnum}, 'Probe.set');
    commands = {commands{:} ...
        {'index' 3*loopnum-2 'load' IgnoreFile   'subject' subjects{loopnum} 'condition' 'Ignore'} ...
        {'index' 3*loopnum-1 'load' MemorizeFile 'subject' subjects{loopnum} 'condition' 'Memorize'} ...
        {'index' 3*loopnum   'load' ProbeFile    'subject' subjects{loopnum} 'condition' 'Probe'}};
end;
% Uncomment the line below to select ICA components with less than 15% residual variance
% commands = {commands{:} {'dipselect', 0.15}};
[STUDY, ALLEEG] = std_editset(STUDY, ALLEEG, 'name','Sternberg','commands',commands,'updatedat','on');

% Update workspace variables and redraw EEGLAB
CURRENTSTUDY = 1; EEG = ALLEEG; CURRENTSET = [1:length(EEG)];
[STUDY, ALLEEG] = std_checkset(STUDY, ALLEEG);
eeglab redraw
```

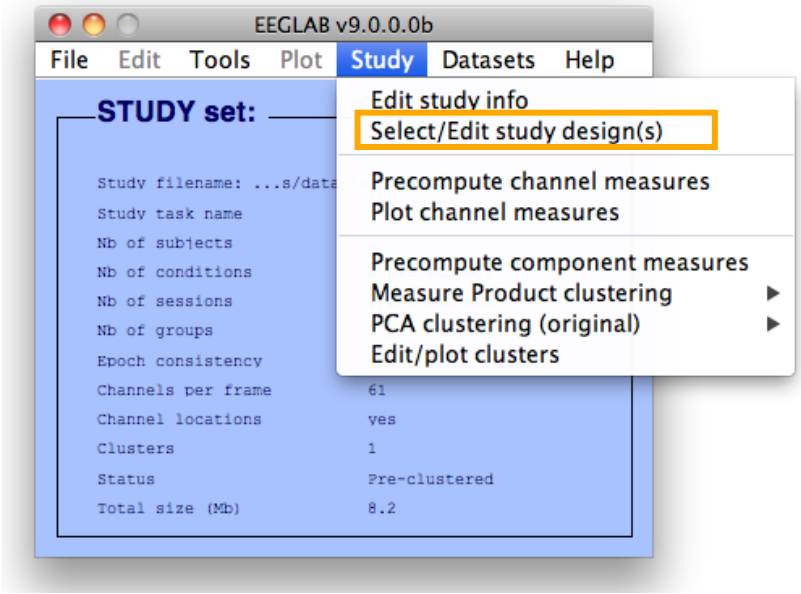
STUDY structure



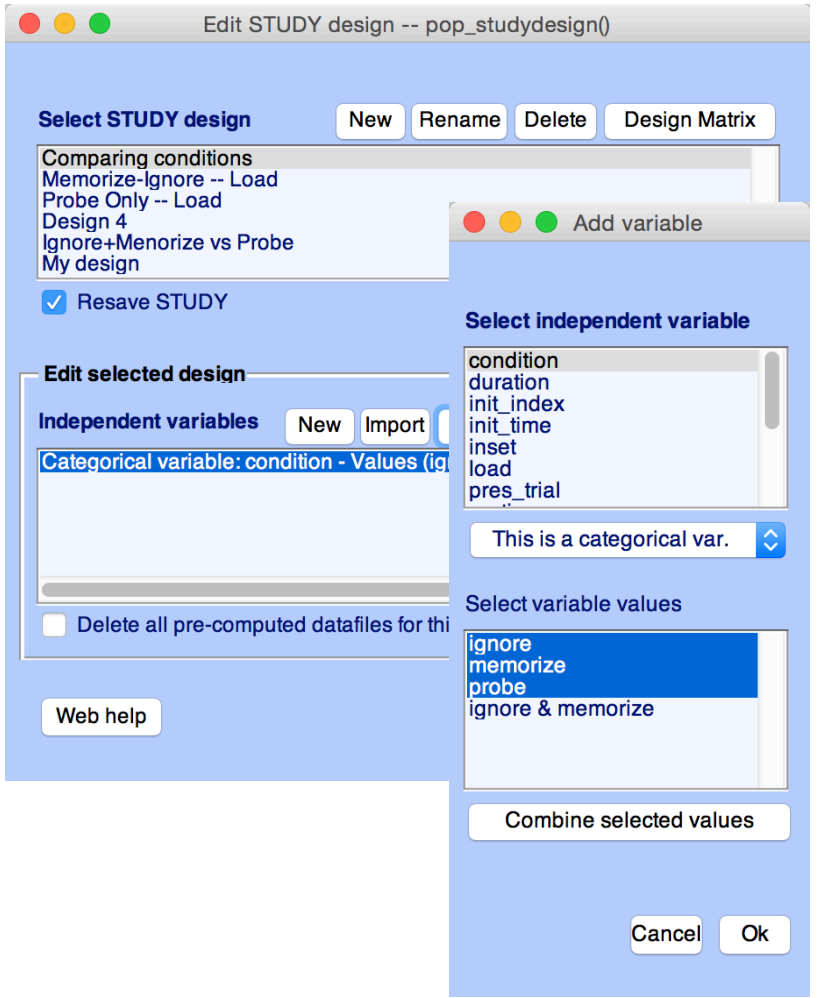


```
STUDY =
            name: 'Sternberg'
            task: 'Sternberg'
     datasetinfo: [1x39 struct]
           notes: ''
        filename: 'stern.study'
        filepath: 'C:\Users\julie\Documents\Workshops\Finland\STUDY'
         history: [1x7332 char]
         subject: {1x13 cell}
           group: {''}
         session: []
       condition: {'ignore'  'memorize'  'probe'}
          design: [1x1 struct]
             etc: [1x1 struct]
        preclust: [1x1 struct]
         cluster: [1x1 struct]
         changrp: [1x71 struct]
           saved: 'yes'
```



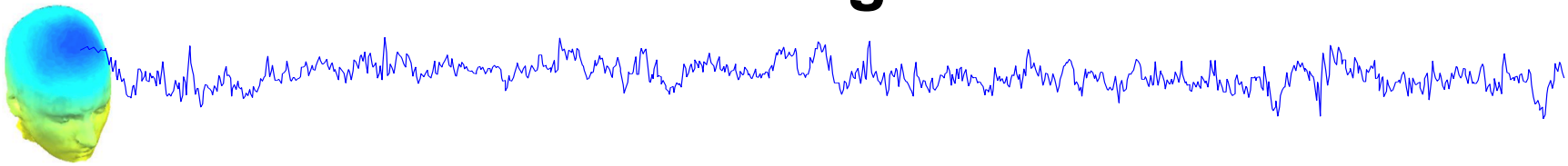


Select subjects




```
STUDY = std_makedesign(STUDY, ALLEEG, 3,
'variable1','condition',
'variable2','',
'name','Design 3',
'values1',{'ignore' 'memorize' 'probe'},
'subjselect',{'S02' 'S03'},
'dataselect',{'condition' {'probe'}});
```

STUDY design structure

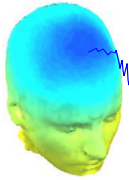


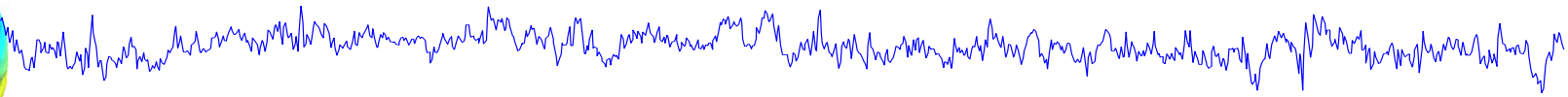
```
STUDY.design(1)

ans =

           name: 'Design 1 - compare letter types'
       variable: [1x2 struct]
          cases: [1x1 struct]
        include: {}
          cells: [1x39 struct]
```

Exploding the contents of each of these sub-structures, we obtain

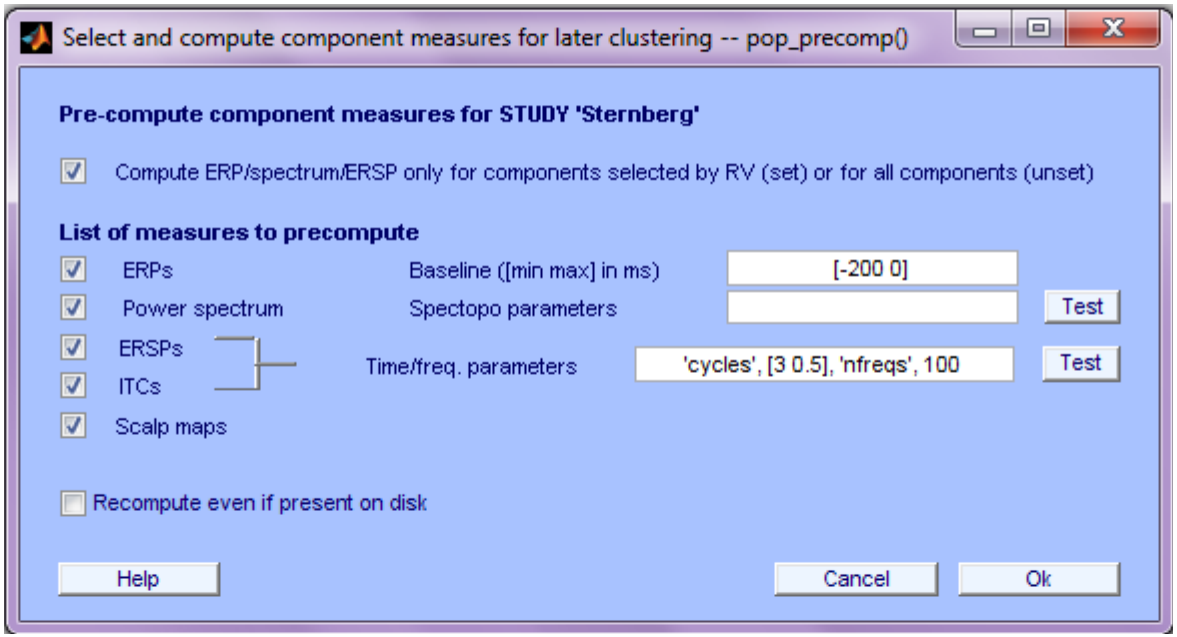
```
           name: 'Design 1 - light and audio all subjects'

       variable: [1x2 struct]
            (1).label   : 'condition'
            (1).pairing: 'on'
            (1).value   : {'ignore'  'memorize'  'probe'}
            (2).label   : ''
            (2).pairing: 'off'
            (2).value   : {}

         cases: [1x1 struct]
            label: 'subject'
            value: {'S01'  'S02'  'S03'  'S04'  'S05'  'S06' }
```

Precompute data measures



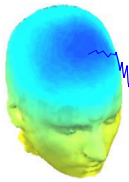


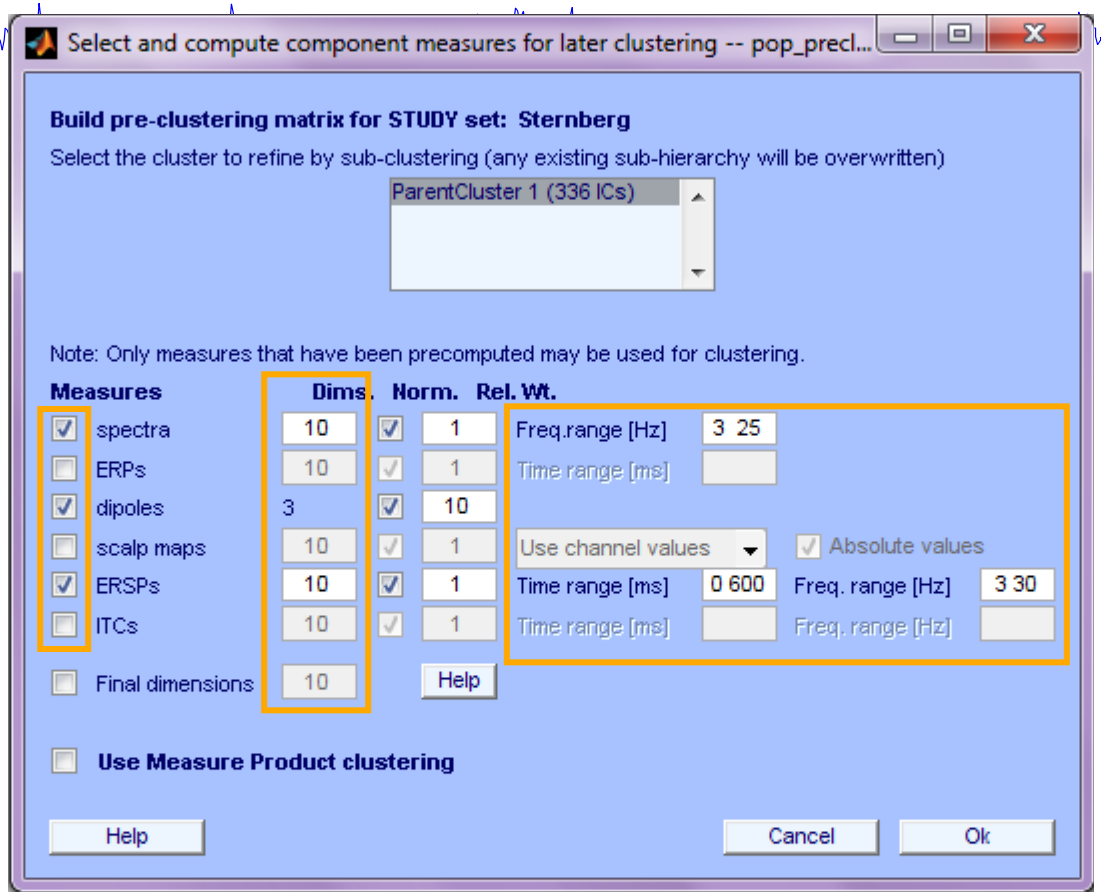
TIP: Compute all measures so you can test different combinations for clustering




```
[STUDY ALLEEG] = std_precomp(STUDY, ALLEEG, 'components', 'erp', ...
'on', 'rmbase',[-200 0] , 'scalp', 'on', 'spec', 'on', ...
'specparams',{}, 'ersp', 'on', 'erspparams', {'cycles' [3 0.5] ...
'nfreqs',100, 'freqs',[3 70] }, 'itc', 'on');
```

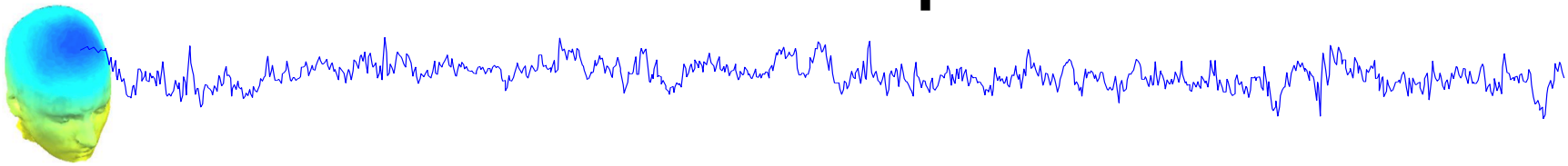
Precluster the data

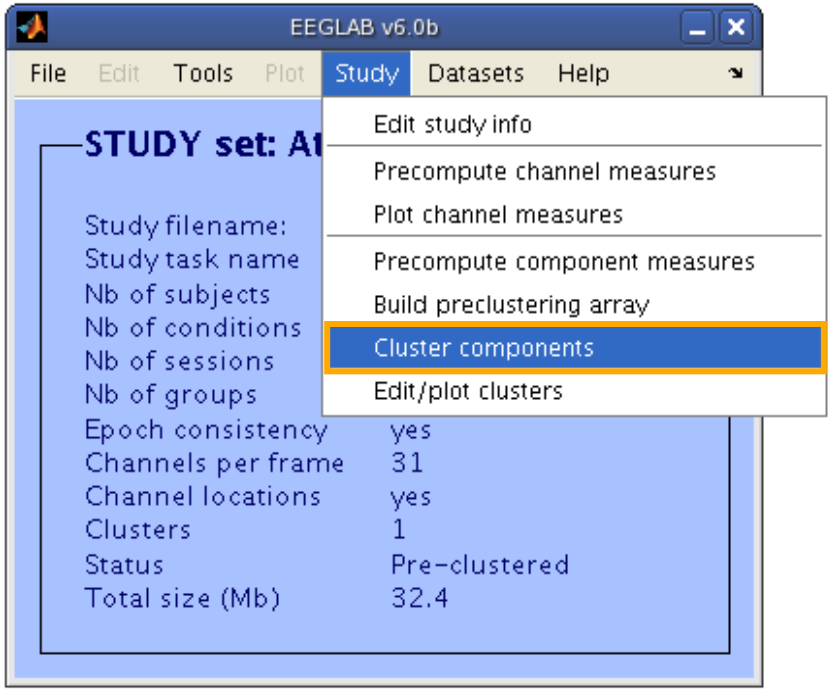


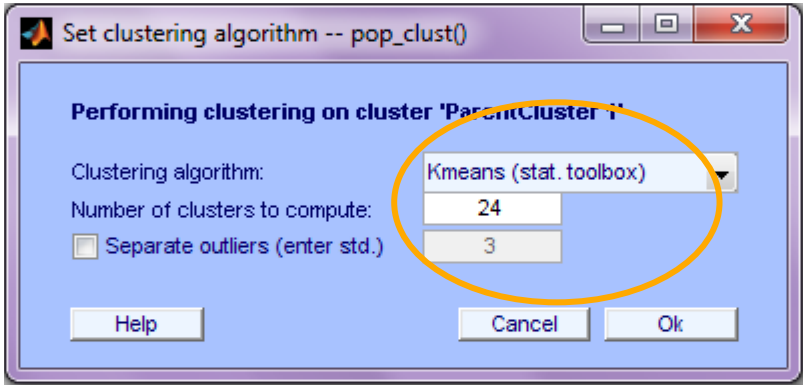



```
[STUDY ALLEEG] = std_preclust(STUDY, ALLEEG, 1,{'spec','npca',5,…
   'norm',1,'weight',1,'freqrange',[3 25]},{'erp','npca',6,'norm',1,…
   'weight',1, 'timewindow',[0 400]},{'scalp','npca',10,'norm',1,'weight',1,…
   'abso',1},{'dipoles','norm',1,'weight',10},{'ersp','npca',20,…
   'freqrange',[3 30] ,'timewindow',[0 600],'norm',1,'weight',1},{'itc',…
   'npca',6,'freqrange',[3 30],'timewindow',[0 400] , 'norm',1, 'weight',1});
```

Cluster components

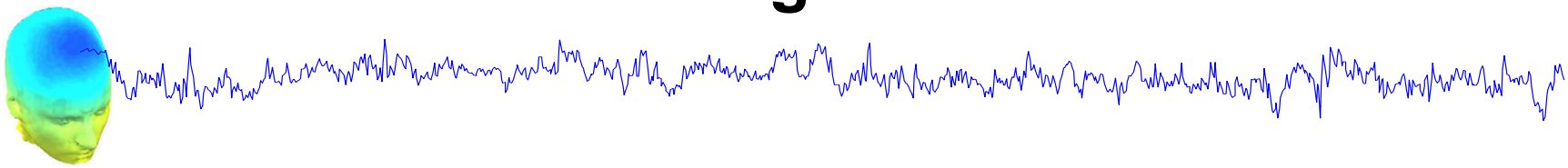







```
[STUDY] = pop_clust(STUDY, ALLEEG,'algorithm','kmeans','clus_num',24);
```

Understanding STUDY structure



26 = # of clusters

```
>> STUDY.cluster
1x26 struct array with fields:
    parent
    name
    child
    comps
    sets
    algorithm
    preclust
    dipole
    allinds
    setinds
```

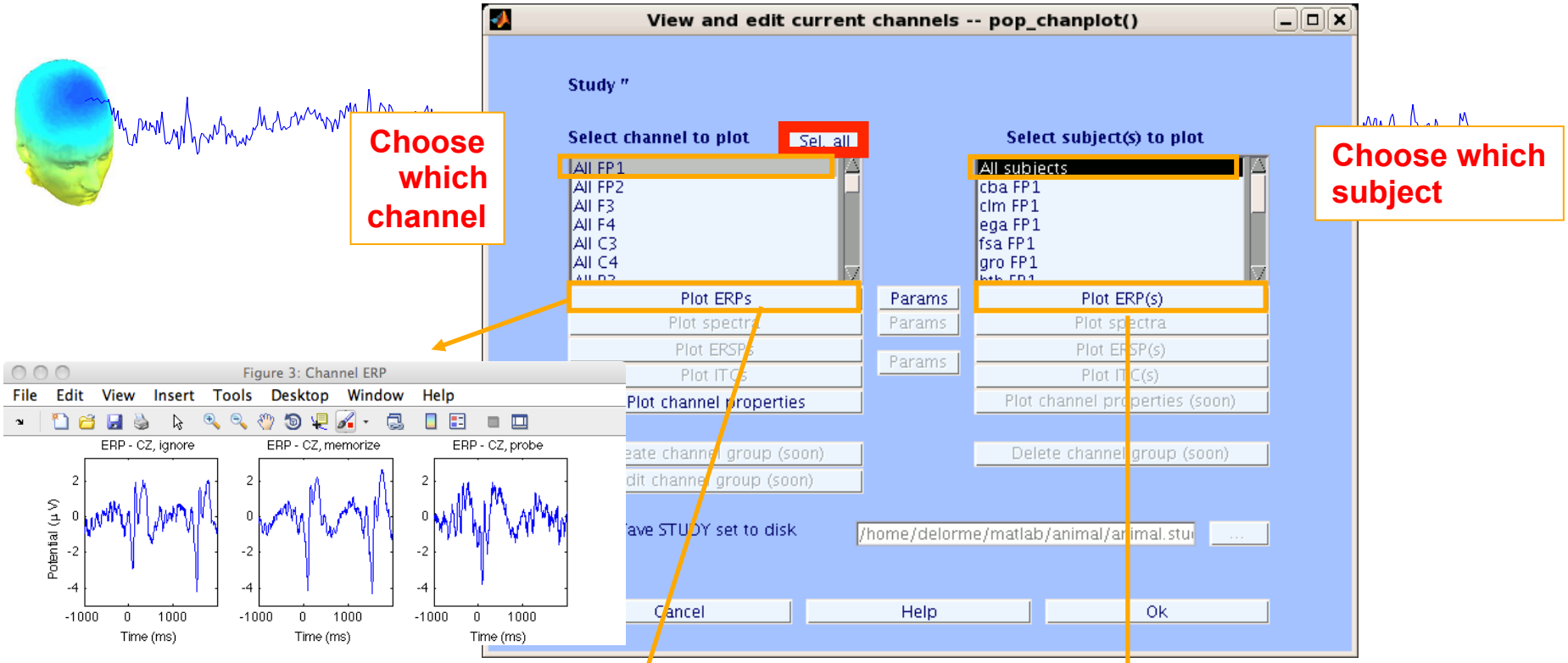
One cluster:

```
>> STUDY.cluster(6)
ans =
       parent: {'ParentCluster 1'}
         name: 'Cls 6'
        child: []
        comps: [35 7 12 35 10 23 7 30 4 …]
         sets: [1 2 3 4 5 6 7 8 9 10 1 2 …]
    algorithm: {'Kmeans'  [24]}
     preclust: [1x1 struct]
       dipole: [1x1 struct]
```

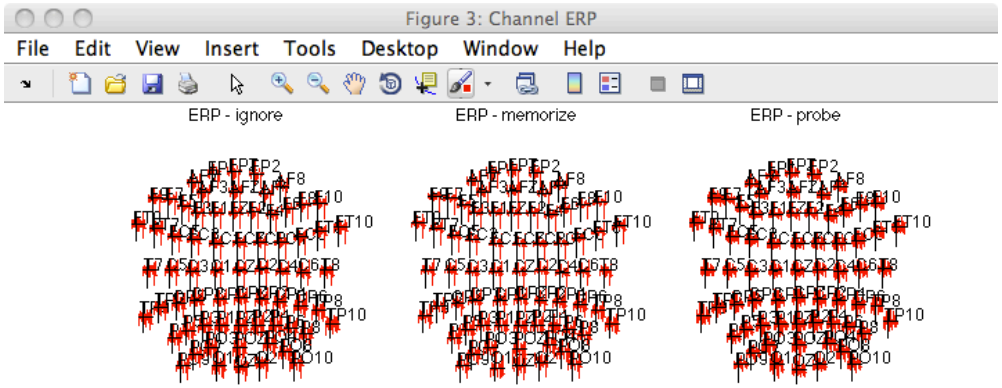
6 = cluster index

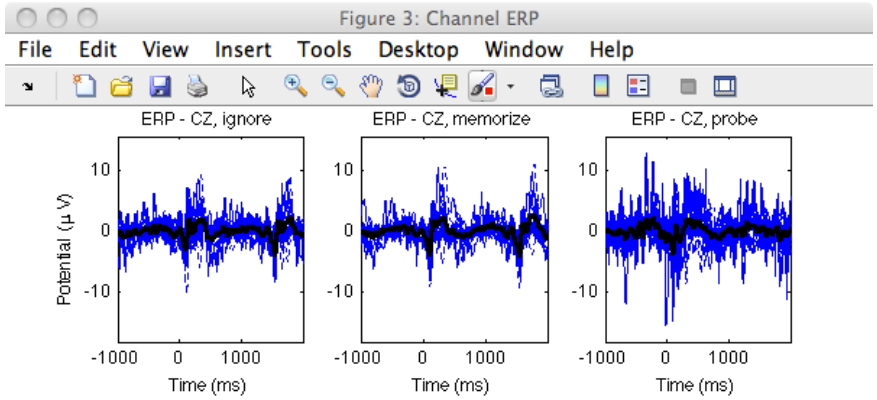
IC indices

dataset indices for ICs



STUDY = std_erpplot(STUDY,ALLEEG,'channels',{'FP1'});

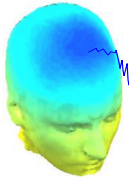


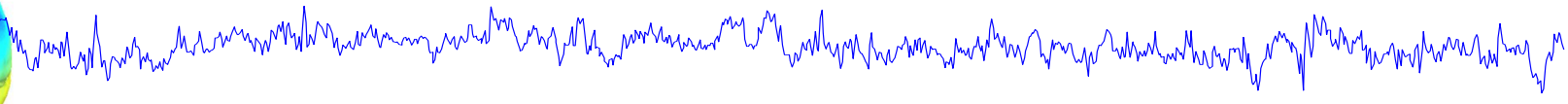


STUDY = std_erpplot(STUDY,ALLEEG,'channels',{'FP1' … });

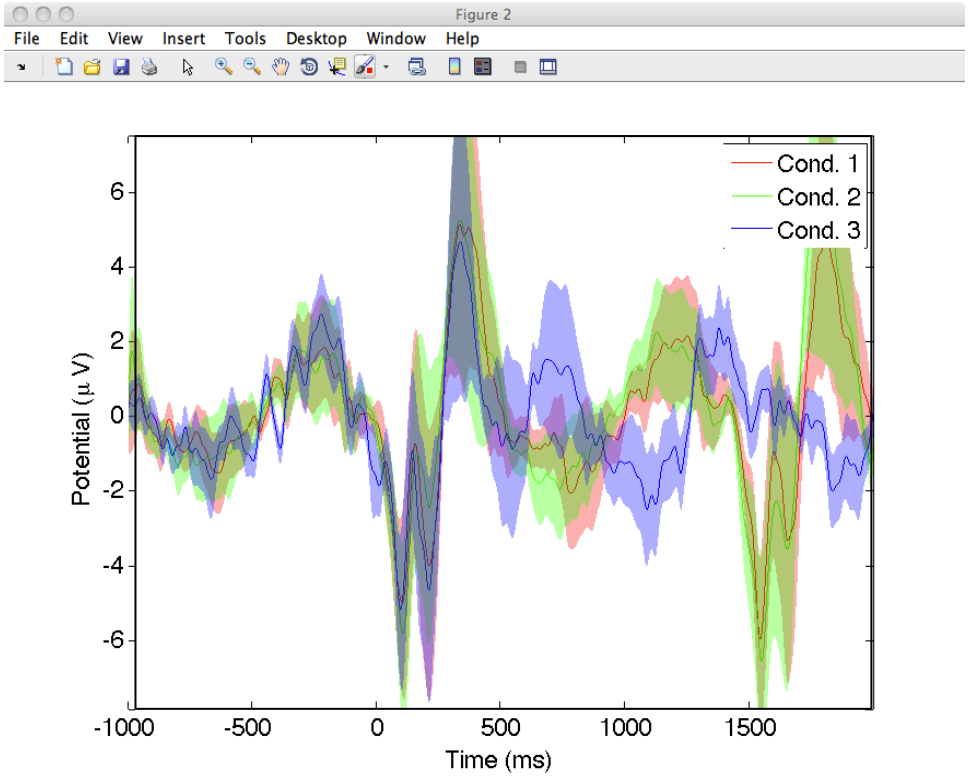
STUDY = std_erpplot(STUDY,ALLEEG,'channels',{'FP1'}, 'plotsubjects', 'on');

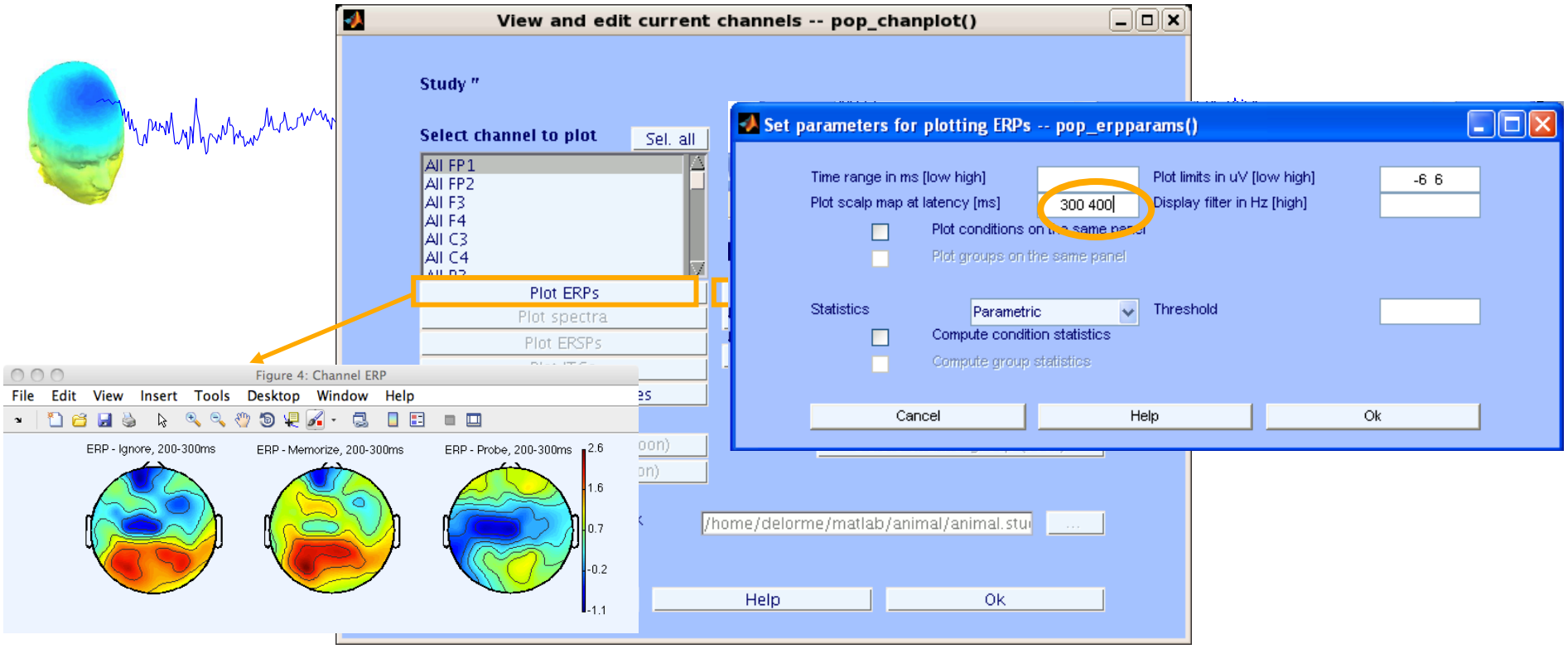
Advanced plotting features





```
STUDY = std_erpplot(STUDY,ALLEEG,'channels',{'CZ'});
std_plotcurve(STUDY.changrp(39).erptimes,
   STUDY.changrp(39).erpdata,  'plotconditions',
   'together', 'plotstderr', 'on', 'filter', 30);
```





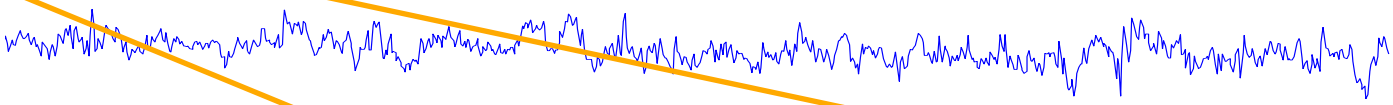

```
STUDY = std_erpplot(STUDY,ALLEEG, 'topotime',[200 300] , 'channels',{'OZ' 'O2' 'FP1' 'FPZ' 'FP2'});
[STUDY erpdata ] = std_erpplot(STUDY,ALLEEG, , 'topotime',[200 300] , 'channels',{'OZ' 'O2'});
```

Exporting to excell file

```
[1x67x13 double]
[1x67x13 double]
[1x67x13 double]
```

```
xlswrite('myxlsfile',squeeze(erpdata{1}),1);
xlswrite('myxlsfile',squeeze(erpdata{2}),2);
xlswrite('myxlsfile',squeeze(erpdata{3}),3);
```

Exporting text file



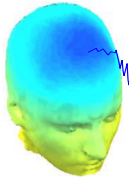
-0.13	-0.4	3.7	-0.9	-1.5	0.23	-0.98	1.8	2.3	-1.4	-2.8	-0.03	3.5
-0.54	-1.3	3.6	-1.1	-1.2	0.62	-0.91	1.6	2.2	-0.98	-7.7	-0.42	3.2
-0.77	-0.06	3.6	-1.4	-1.2	0.78	-0.91	1.2	2.1	-0.66	-0.76	-1	2.5
-0.61	-0.83	3.7	-1.2	-1.2	0.53	-0.88	1.1	1.7	-1.2	-1.8	-1.2	1.6
-0.34	-0.79	3.7	-0.98	-1.2	0.17	-0.72	1	1.4	-1.7	-2.3	-0.72	1.4
-0.27	-0.42	3.2	-0.69	-1.4	-0.04	-0.29	0.97	0.81	-2.5	-1.5	-0.38	1.7
0.097	-0.58	3.2	-0.61	-1.2	-0.32	0.36	0.47	2.1	-0.96	-2.8	0.89	2.4
0.43	-0.04	2.3	-0.47	-0.87	-0.37	0.21	0.83	3.1	-0.53	-0.85	1.2	3.4
0.21	-0.54	2.4	-0.07	-0.05	-0.08	-0.08	1	3.3	-0.42	-3.7	0.92	3.8
-0.1	-1.1	2.7	-0.33	-0.28	0.48	-0.5	1.2	3.3	-0.53	-2	0.36	4
-0.51	-2.2	2.9	-0.59	-0.23	1.3	-0.72	1.4	3.3	-0.14	-16	-0.05	3.9

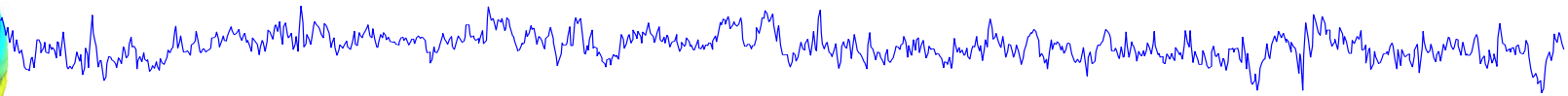
```
dlmwrite('erpfile.txt',squeeze(erpdata{1}),'delimiter', '\t', 'precision', 2);

dlmwrite('erpfile.txt',squeeze(erpdata{2}),'-append', 'roffset', 1,
         'delimiter', '\t', 'precision', 2);

dlmwrite('erpfile.txt',squeeze(erpdata{2}),'-append', 'roffset', 1,
         'delimiter', '\t', 'precision', 2);
```

Exercice: run STUDY Script

```
% Create Stern STUDY
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;
pop_editoptions( 'option_storedisk', 1);
subjects = {'S01' 'S02' 'S03' 'S04' 'S05' 'S06' 'S07' 'S08' 'S09' 'S10' 'S11' 'S12'};
filepath = '/Users/arno/temp/STUDY'; % XXXXX Change path here XXXXX
if ~exist(filepath), error('You need to change the path to the STUDY'); end;
commands = {}; % initialize STUDY dataset list

% Loop through all of the subjects in the study to create the dataset
for loopnum = 1:length(subjects) %for each subject
    IgnoreFile   = fullfile(filepath, subjects{loopnum}, 'Ignore.set');
    MemorizeFile = fullfile(filepath, subjects{loopnum}, 'Memorize.set');
    ProbeFile    = fullfile(filepath, subjects{loopnum}, 'Probe.set');
    commands = {commands{:} ...
        {'index' 3*loopnum-2 'load' IgnoreFile   'subject' subjects{loopnum} 'condition' 'Ignore'} ...
        {'index' 3*loopnum-1 'load' MemorizeFile 'subject' subjects{loopnum} 'condition' 'Memorize'} ...
        {'index' 3*loopnum   'load' ProbeFile    'subject' subjects{loopnum} 'condition' 'Probe'}};
end;
% Uncomment the line below to select ICA components with less than 15% residual variance
% commands = {commands{:} {'dipselect', 0.15}};
[STUDY, ALLEEG] = std_editset(STUDY, ALLEEG, 'name','Sternberg','commands',commands,'updatedat','on');

% Update workspace variables and redraw EEGLAB
CURRENTSTUDY = 1; EEG = ALLEEG; CURRENTSET = [1:length(EEG)];
[STUDY, ALLEEG] = std_checkset(STUDY, ALLEEG);
eeglab redraw

[STUDY ALLEEG] = std_precomp(STUDY, ALLEEG, {},'rmicacomps','on','interp','on','recompute','on','erp','on');
STUDY = pop_erpparams(STUDY, 'topotime',[200 300] );
[STUDY erpdata] = std_erpplot(STUDY,ALLEEG,'channels',{'LEYE' 'REYE' 'OZ' 'O2' 'FP1' 'FPZ' 'FP2' 'AF7' ...
            'AF3' 'AFZ' 'AF4' 'AF8' 'F9' 'F7' 'F5' 'F3' 'F1' 'FZ' 'F2' 'F4' 'F6' 'F8' 'F10' 'FT9' ...
            'FT7' 'FC5' 'FC3' 'FC1' 'FCZ' 'FC2' 'FC4' 'FC6' 'FT8' 'FT10' 'T7' 'C5' 'C3' 'C1' 'CZ' ...
            'C2' 'C4' 'C6' 'T8' 'TP9' 'TP7' 'CP5' 'CP3' 'CP1' 'CPZ' 'CP2' 'CP4' 'CP6' 'TP8' 'TP10' ...
            'P7' 'P5' 'P3' 'P1' 'PZ' 'P2' 'P4' 'P6' 'P8' 'PO9' 'PO7' 'PO3' 'POZ' 'PO4' 'PO8' 'PO10' 'O1'});

dlmwrite('erpfile.txt',squeeze(erpdata{1}),'delimiter', '\t', 'precision', 2);
dlmwrite('erpfile.txt',squeeze(erpdata{2}),'-append', 'roffset', 1, 'delimiter', '\t', 'precision', 2);
dlmwrite('erpfile.txt',squeeze(erpdata{2}),'-append', 'roffset', 1, 'delimiter', '\t', 'precision', 2);
```

Advanced scripting: EEG pipeline





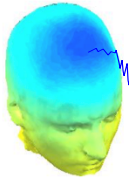
```
sInfo = [];
sInfo(end+1).file = 'rawdata/S01.raw';
sInfo(end).name = 'S01';
sInfo(end).bad_channels = { 'E1' };
```

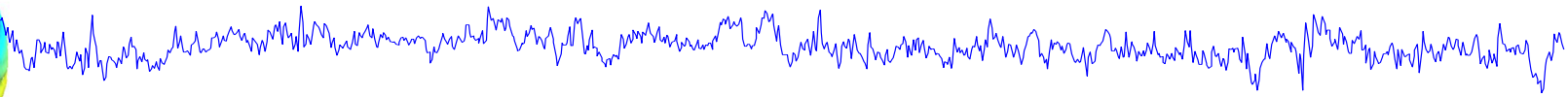
Raw data file

Subject name

Subject name

Advanced scripting: EEG pipeline





```
sInfo = [];
sInfo(end+1).file = 'rawdata/S01.raw';
sInfo(end).name = 'S01';
sInfo(end).bad_channels = { 'E1' };
sInfo(end).bad_data  = [726 1495;6098 6831;13245 14057;15715 16399;22756 2445
```

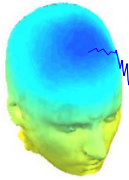
Raw data file

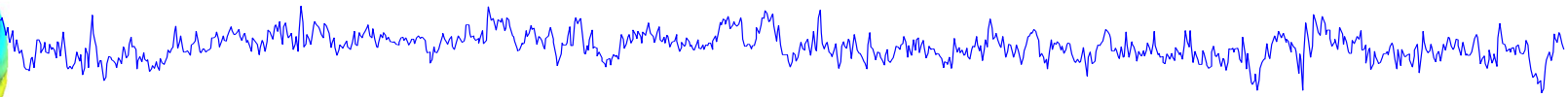
Subject name

Subject name

Copy the output from the eeg_eegrej function in the history

Advanced scripting: EEG pipeline





```
sInfo = [];
sInfo(end+1).file = 'rawdata/S01.raw';
sInfo(end).name = 'S01';
sInfo(end).bad_channels = { 'E1' };
sInfo(end).bad_data  = [726 1495;6098 6831;13245 14057;15715 16399;22756 2445
sInfo(end).bad_comps = [1.6681  1.9870 0.3979 0.4444 -0.2274 -0.1433 -0.2626
                        1.1917 -1.4838 0.7469 -1.1599 0.4773 -0.3257  0.3074
```

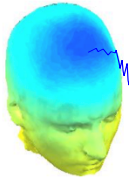
Raw data file

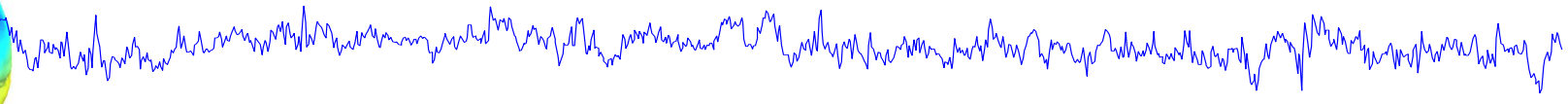
Subject name

Subject name

Copy transposed columns of the inverse weight matrix
EEG.icawinv for your selected artifact components

Advanced scripting: EEG pipeline



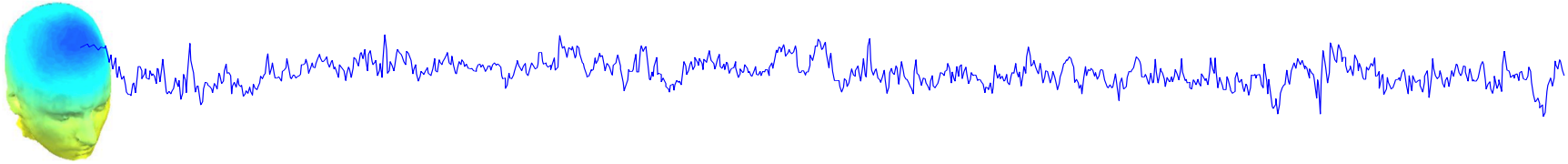


datainfo.m file

```
sInfo = [];
sInfo(end+1).file = 'rawdata/S01.raw';
sInfo(end).name = 'S01';
sInfo(end).bad_channels = { 'E1' };
sInfo(end).bad_data  = [726 1495;6098 6831;13245 14057;15715 16399;22756 2445
sInfo(end).bad_comps = [1.6681  1.9870 0.3979 0.4444 -0.2274 -0.1433 -0.2626
                        1.1917 -1.4838 0.7469 -1.1599 0.4773 -0.3257  0.3074

sInfo(end+1).file = 'rawdata/S02.raw';
sInfo(end).name = 'S02';
sInfo(end).bad_channels = { };
sInfo(end).bad_data  = [41661 43713;24000 24833;44878 46501;48706 49210;51190
sInfo(end).bad_comps = [0.6960 -0.8637  0.9087 -0.8028  0.4873 -0.2142  0.273
                       -0.0875 -0.4056 -0.0287 -0.3870  0.0600 -0.3716  0.342
                        2.1928  1.5712  0.8622  0.3215 -0.0357 -0.3125 -0.226

sInfo(end+1).file = 'rawdata/S03.raw';
sInfo(end).name = 'S03';
sInfo(end).bad_channels = { 'E10' 'E19' 'E20' 'E29' };
sInfo(end).bad_data  = [1 10449;19808 21815;25678 27254;29257 30010;34023 360
sInfo(end).bad_comps = [ 1.8583  2.0468 -0.0516  0.3159 -0.4256 -0.2770 -0.36
                         1.2189 -0.7385  1.2464 -0.8913  0.5475 -0.3971  0.29
                        -0.1248 -0.1358 -0.1954 -0.2533 -0.1555 -0.2313 -0.03
```

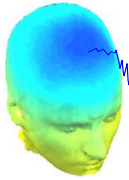
```
datainfo;
pop_editoptions( 'option_storedisk', 1);
Folder = 'ds_eeg_preprocessed_data';
if ~exist(outputfolder), mkdir(outputfolder); end;

for iSubj = 1:length(sInfo)

    % load dataset
    EEG = pop_biosig( sInfo(iSubj).file);
    EEG.setname = sInfo(iSubj).name;

    % proprocess data
    EEG = pop_iirfilt( EEG, 0.5, 0, [], 0, 0); % high pass filtering
    EEG = pop_iirfilt( EEG, 0, 55, [], 0, 0); % low pass filtering
    EEG = pop_select(EEG, 'nochannel', sInfo(iSubj).bad_channels); % remove bad channels
    EEG = pop_reref( EEG, []); % average reference (optional)
    EEG = eeg_eegrej( EEG, sInfo(iSubj).bad_data); % remove bad portions of data

    % run ICA
    EEG = pop_runica(EEG, 'pca', EEG.nbchan-1);

    % tag bad components
    % EEG = pop_findmatchingrejcomps(EEG, matchcomps',sInfo(iSubj).bad_comps,'corrthresh',0.92);
    if ~isempty(sInfo(iSubj).bad_comps)
        [corr,indx,indy] = matcorr(EEG.icawinv', sInfo(iSubj).bad_comps);
        if ~all(abs(corr(1:size(sInfo(iSubj).bad_comps,1))) > 0.92),
              error('Correlation too low'); end;
        EEG.reject.gcompreject(indx(1:size(sInfo(iSubj).bad_comps,1))) = 1;
    end;

    % extract data epochs
    EEG = pop_epoch(EEG, { 'S1' 'S2' });

    % save dataset
    EEG.saved = 'no';
    EEG = pop_saveset( EEG, 'filepath', folder, 'filename', [ sInfo(iSubj).name '.set' ]);
end;
```

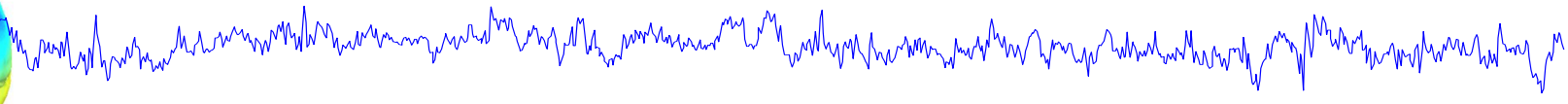
Create STUDY



```
datainfo;
pop_editoptions( 'option_storedisk', 1);
outputEEGFolder = 'preprocessed_data';
studyCommand    = {};

% generate STUDY commands
for iSubject = 1:length(sInfo)
    fileName  = fullfile(outputEEGFolder, [ sInfo(iSubject).name '.set' ]);
    studyCommand = [ studyCommand { 'index' iSubject 'load' fileName 'subject' ...
           sInfo(iSubject).name } ];
end;

% create data
[STUDY ALLEEG] = std_editset( [], [], 'name', 'test', 'commands', studyCommand, ...
         'updatedat','off', 'filename', 'test.study', 'resave', 'on');
STUDY = std_makedesign(STUDY, ALLEEG, 1, 'variable1','','variable2','','name', 'Design 1', ...
    'pairing1','on','pairing2','on','delfiles','off','defaultdesign','off','subjselect', ...
    {sInfo.name},'filepath','studyfiles');

% update workspace variables and redraw EEGLAB
CURRENTSTUDY = 1; EEG = ALLEEG; CURRENTSET = [1:length(EEG)];
[STUDY, ALLEEG] = std_checkset(STUDY, ALLEEG);
eeglab redraw

% precompute and plot data
[STUDY ALLEEG] = std_precomp(STUDY, ALLEEG, {},'interp','on','recompute','on','erp', 'on');
[STUDY erp] = std_erpplot(STUDY,ALLEEG, 'channels',{allchanlocs.labels}, 'topotime',[200 300]);
```

Exercice: build your own pipeline





Suggestion for exercise

1. Load SimpleOddball.set dataset
2. High pass filter at 1Hz (menu Tools > Filter)
3. Reject bad portion of data by hand (menu Tools > Reject continuous…)
4. Re-reference to average ref. (optional) (menu Tools > Re-reference)
5. Build your *datainfo.m* file using EEGLAB history (see scripting lecture)
6. Run ICA (use PCA if necessary); select bad ICA components
7. Epoch data on Oddball (type 2) and Standard (type 1) – save dataset
8. Add components to you *datainfo.m* file
9. Create a STUDY with this single file
10. Compare the ERP for Oddball (type 2) and Standard (type 1) and use single-trial statistics with cluster correction for multiple comparisons
11. Build a script that creates the STUDY and perform the same analysis
12. Save the figure at the end of the script in eps or jpg format (“print –deps file” command or “print –djpg file” command).
13. Run the full pipeline (dataset processing and STUDY processing)
14. Change the filtering in the pipeline (step 2) and observe effects