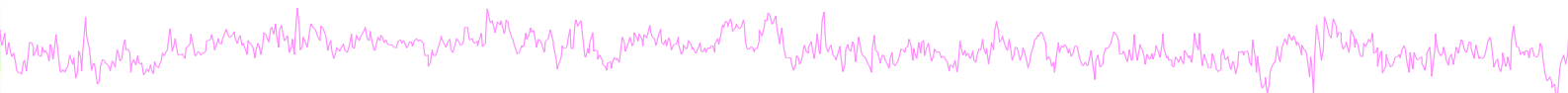
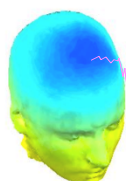
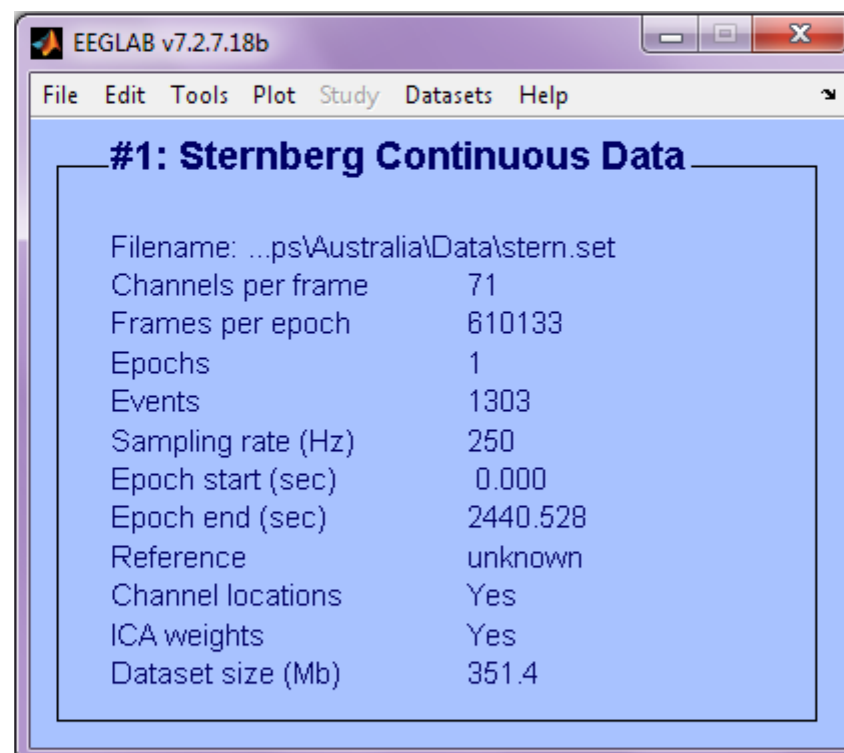
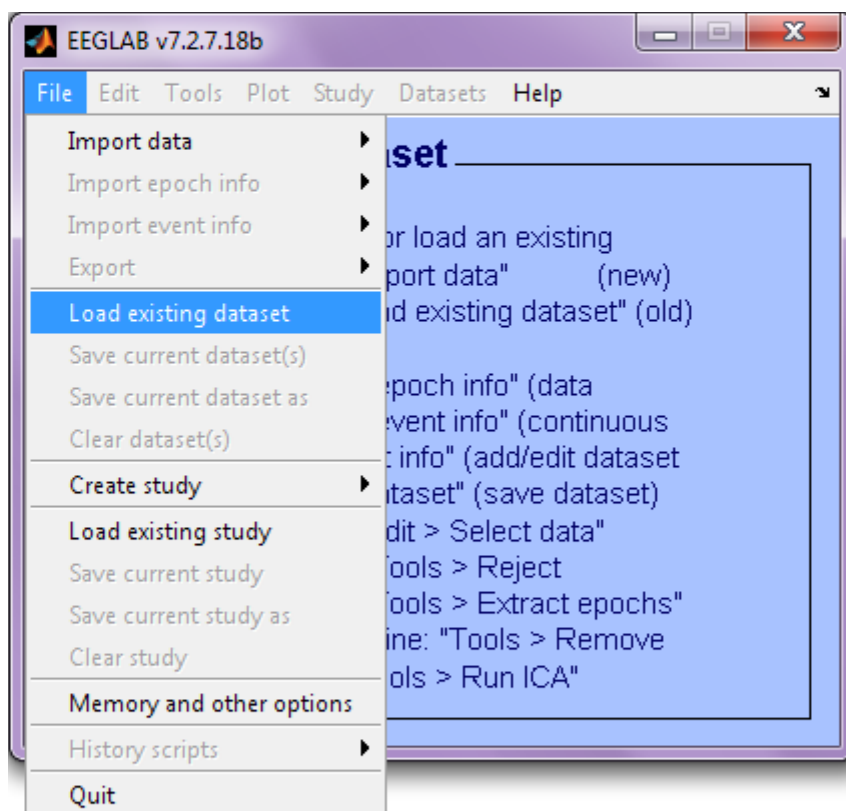


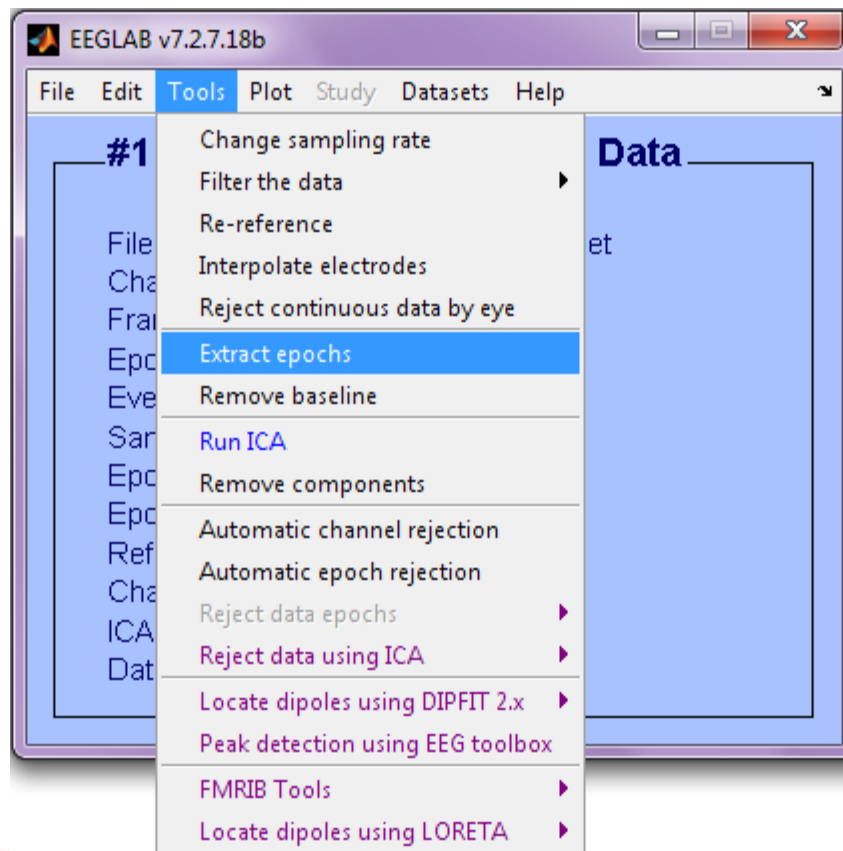
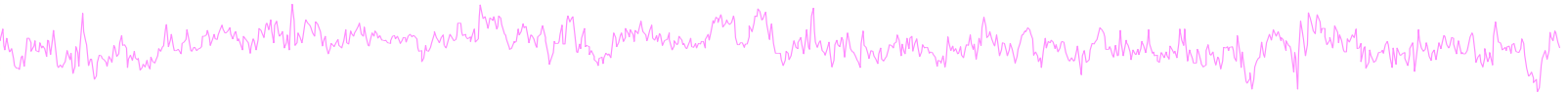
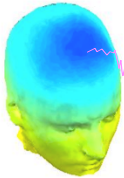
Create a script from 'eegh' output



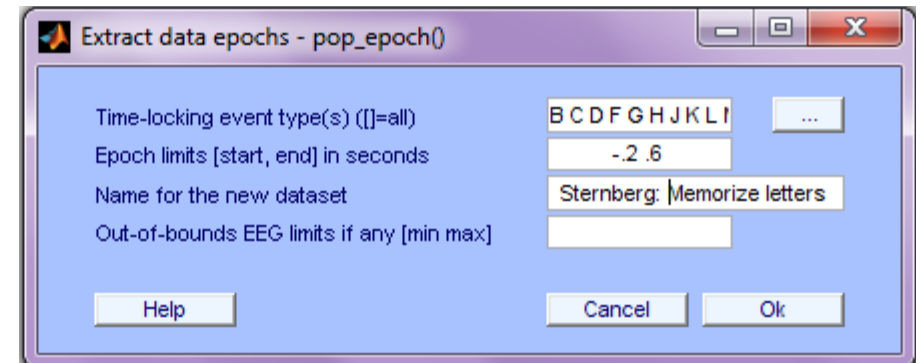
Start by loading a continuous dataset



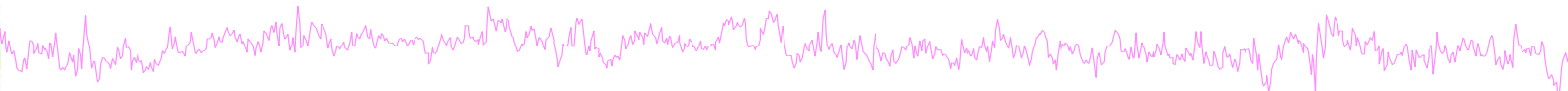
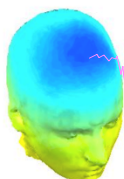
Create a script from 'eegh' output



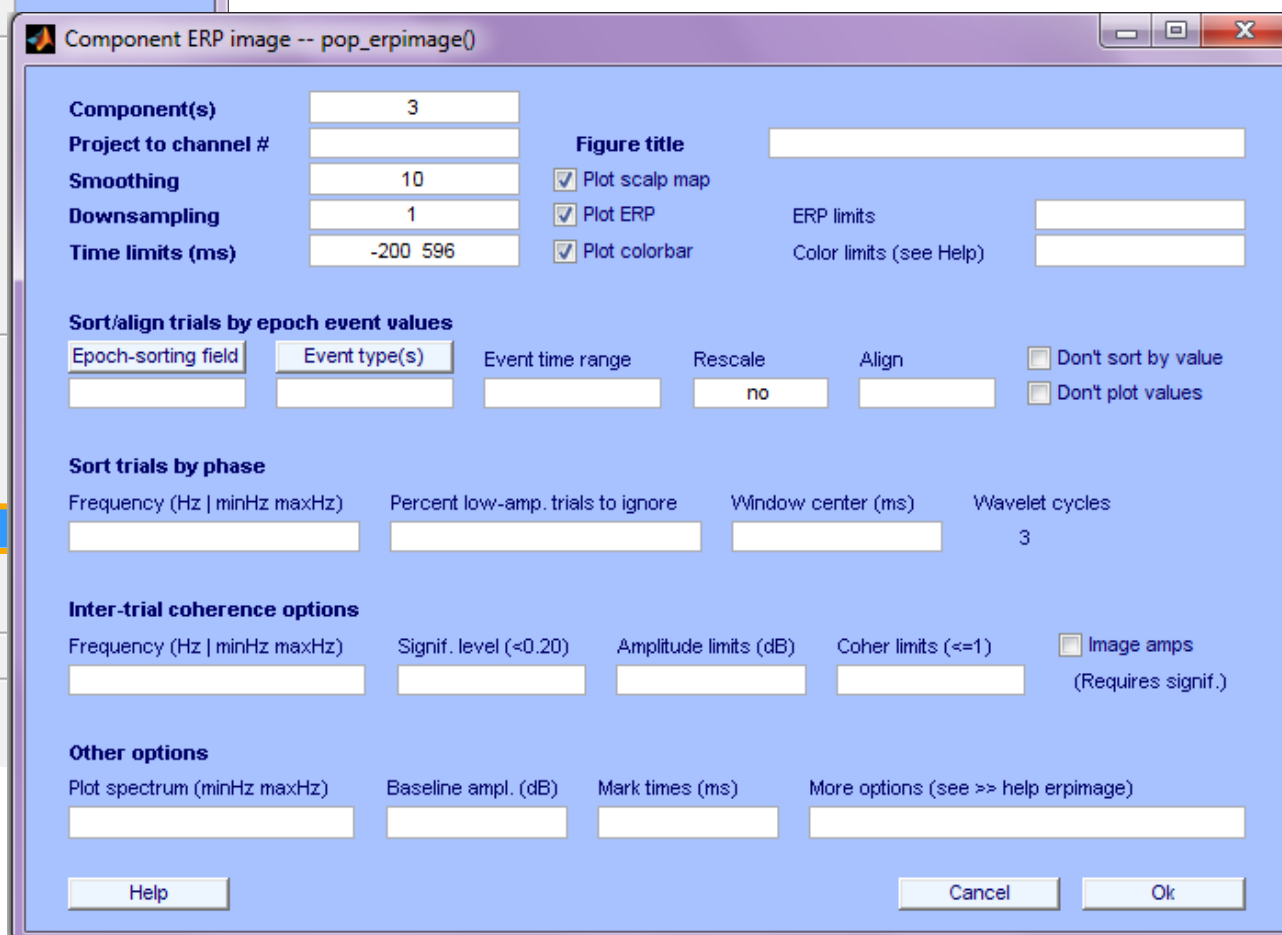
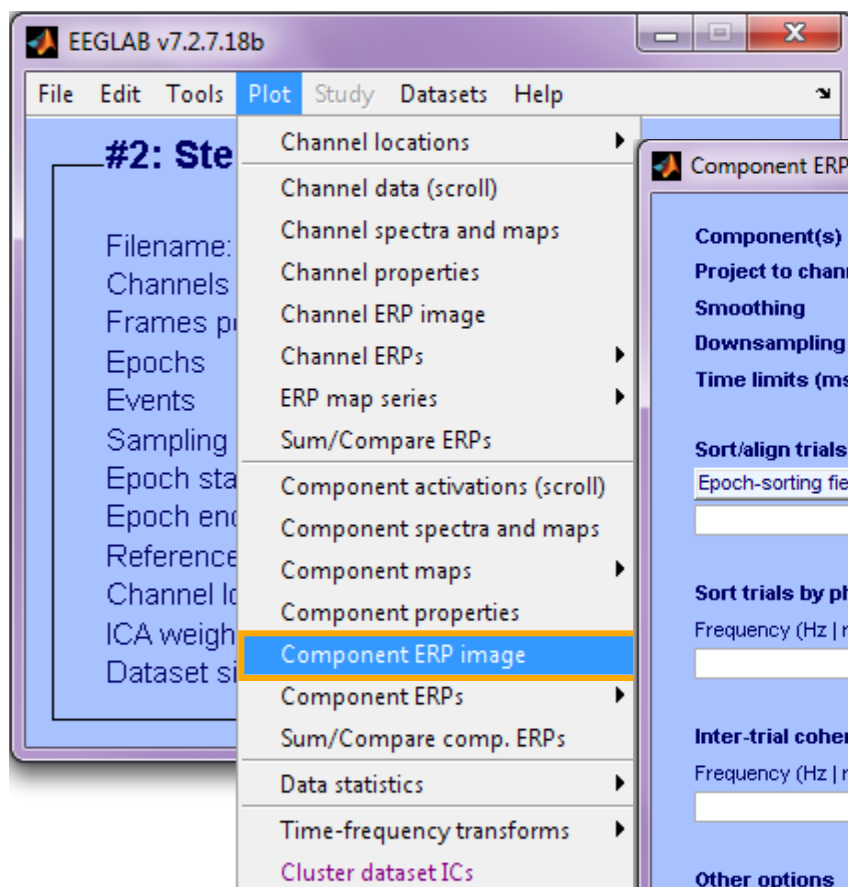
Epoch on Memorize letters



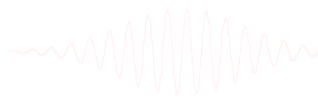
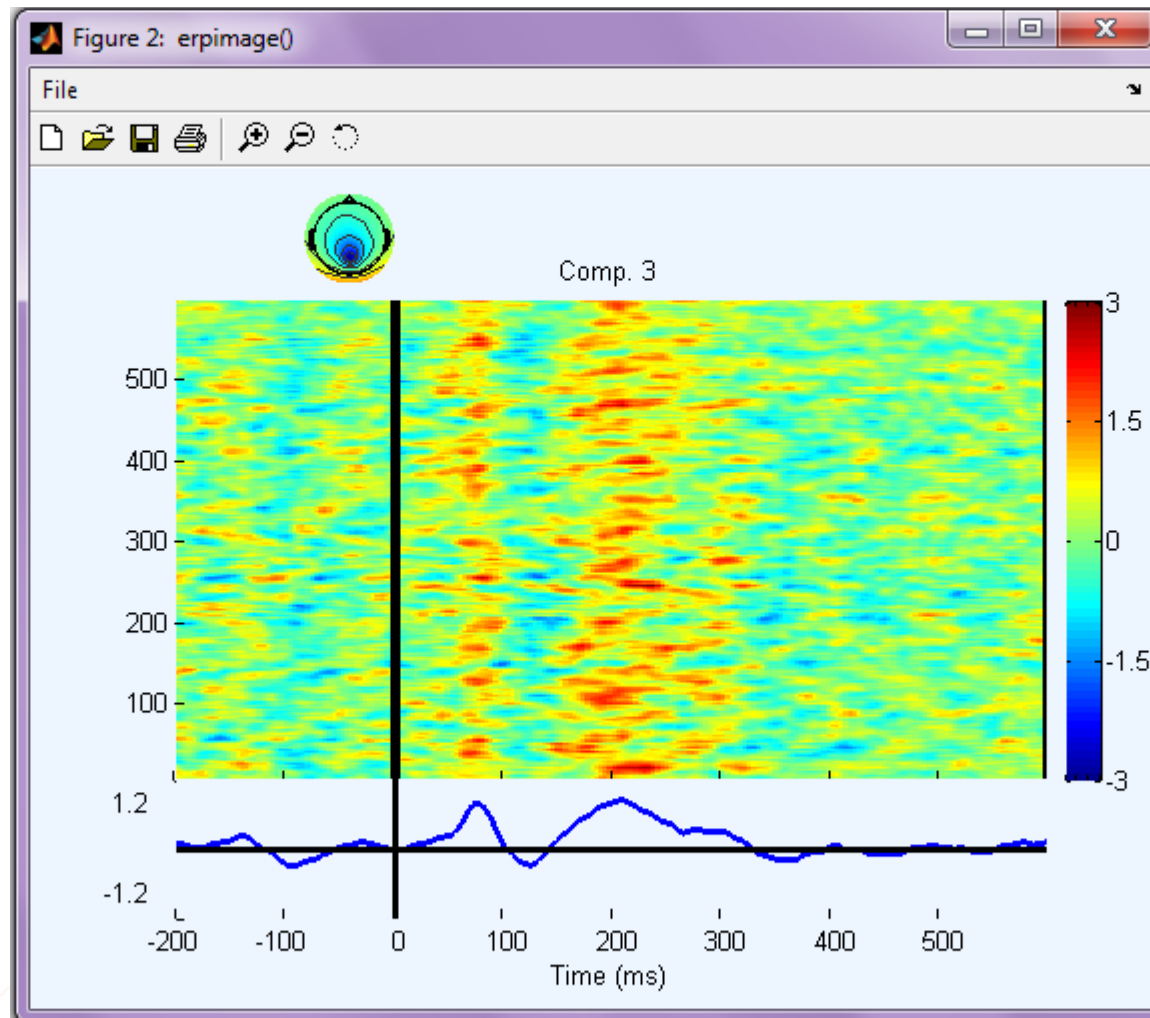
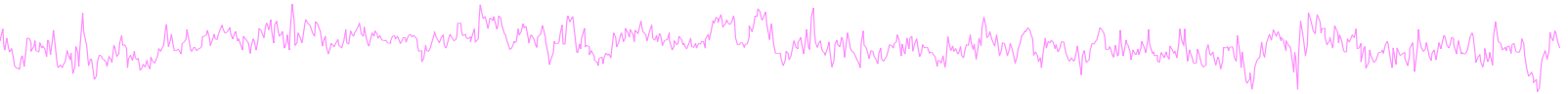
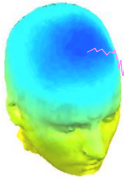
Create a script from 'eegh' output



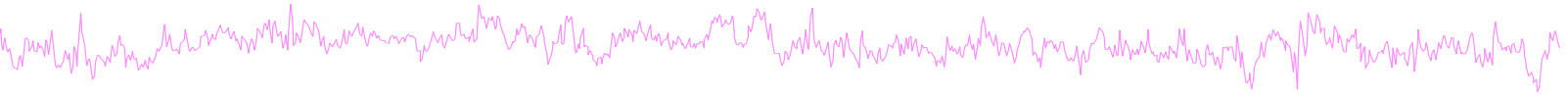
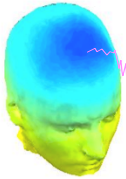
Plot an IC ERP image



Create a script from 'eegh' output

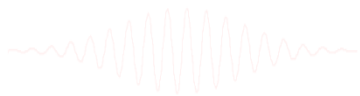


Retrieve commands from eegh

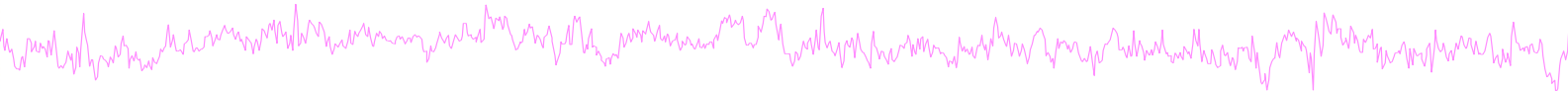
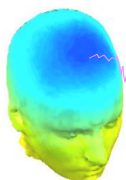


Write a script to do this:

```
>> eegh
```



Parameterize a script



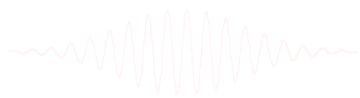
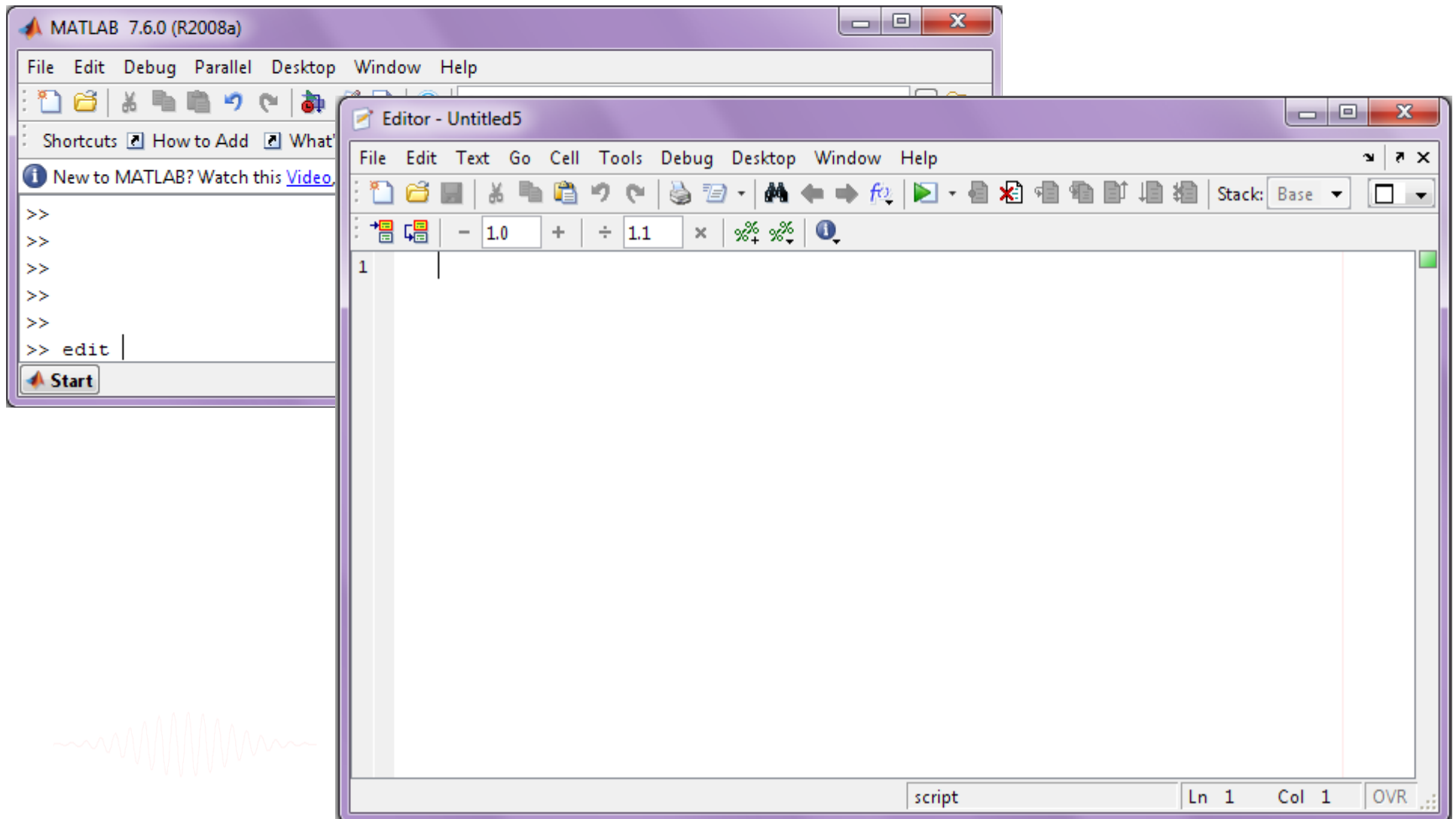
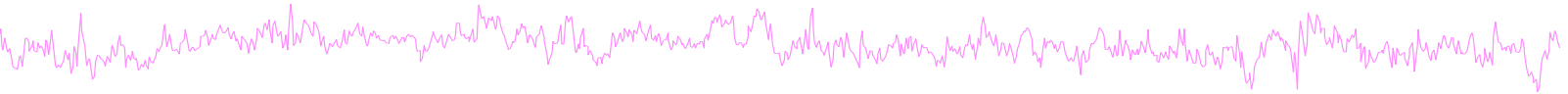
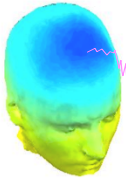
```
>> eegh
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

EEG = pop_loadset('filename', 'stern_125Hz.set', 'filepath', ...
    '\eeglab_data\');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 0);

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

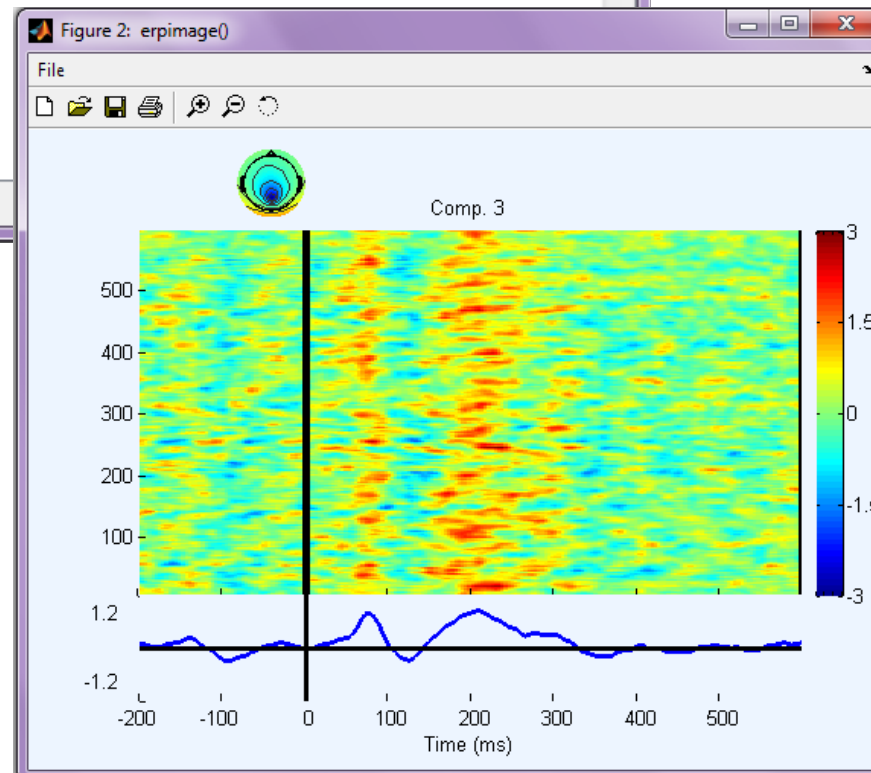
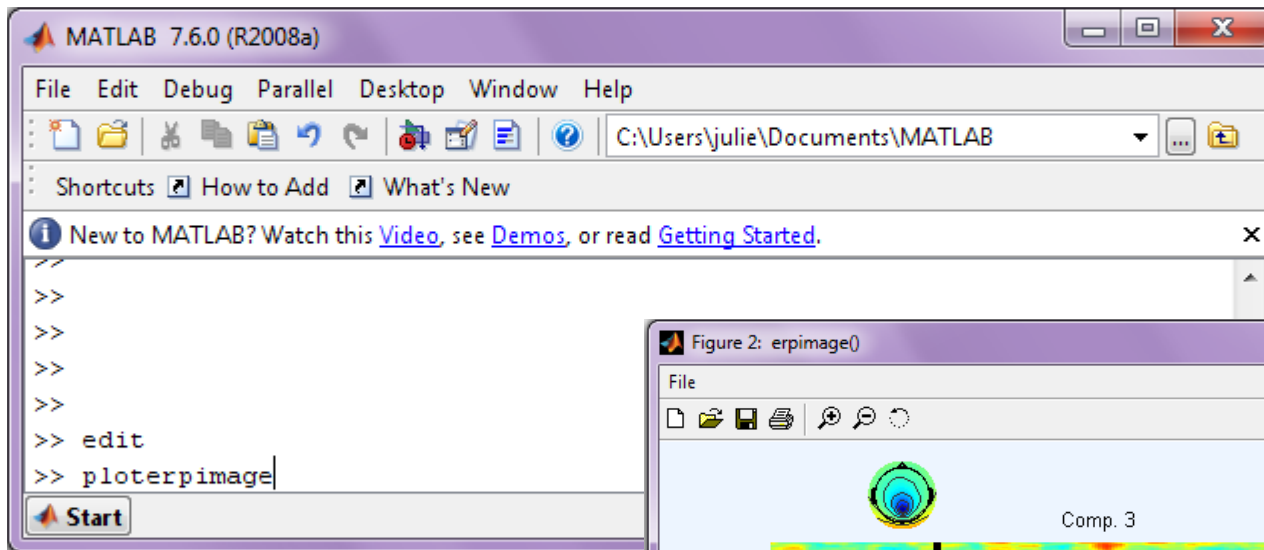
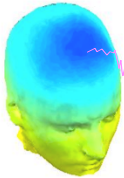
figure; pop_erpimage(EEG,0, [3],[], 'Comp. 3',10,1,{},[],...
    '', 'yerplabel', '', 'erp', 'on', 'cbar', 'on', 'topo',...
    {mean(EEG.icawinv(:, [3]),2) EEG.chanlocs EEG.chaninfo });
```

Create a Matlab script

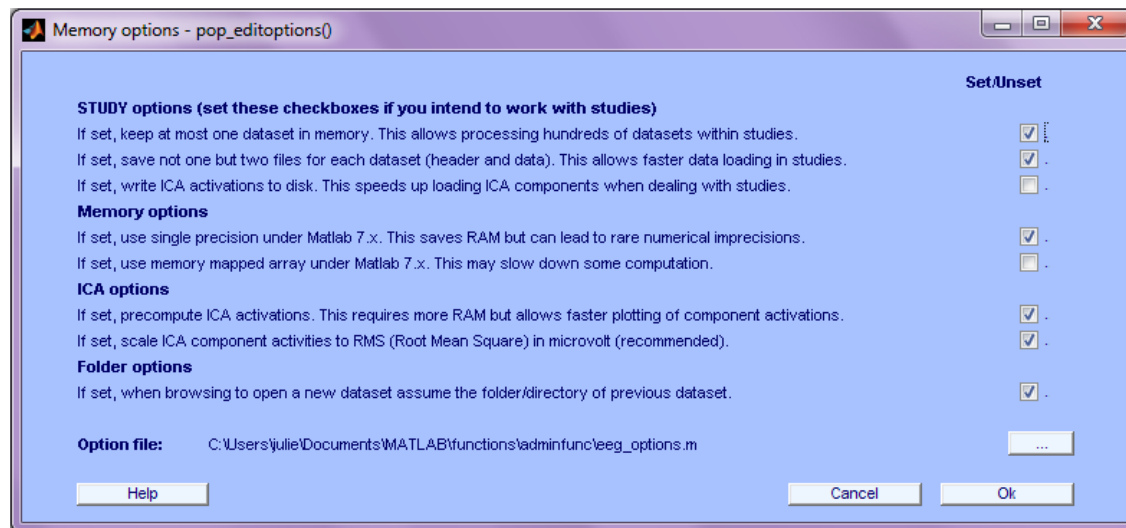
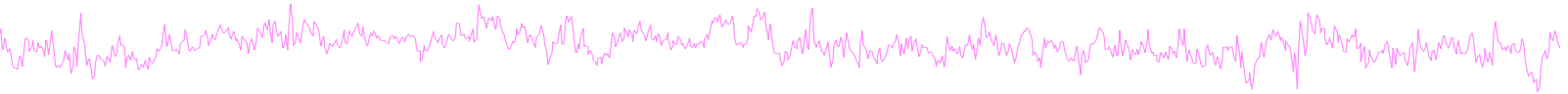
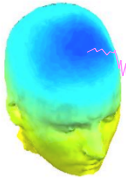


Save as 'ploterpimage.m'

Run your new script



STUDY scripts



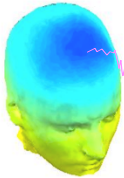
Most important option:

- 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_edioptions( 'option_storedisk', 1);`

Edit dataset info



Create a new STUDY set -- pop_study()

Edit STUDY set information - remember to save changes

STUDY set name:

STUDY set task name:

STUDY set notes:

	dataset filename	browse	subject	session	condition	group	Select by r.v.	
1	C:\Users\julie\Documents\Wor	...	S01		memorize		Comp.: 3 5 ...	Clear
2	C:\Users\julie\Documents\Wor	...	S01		ignore		Comp.: 3 5 ...	Clear
3	C:\Users\julie\Documents\Wor	...	S01		probe		Comp.: 3 5 ...	Clear
4	C:\Users\julie\Documents\Wor	...	S02		memorize		Comp.: 5 6 ...	Clear
5	C:\Users\julie\Documents\Wor	...	S02		ignore		Comp.: 5 6 ...	Clear
6	C:\Users\julie\Documents\Wor	...	S02		probe		Comp.: 5 6 ...	Clear
7	C:\Users\julie\Documents\Wor	...	S03		memorize		Comp.: 6 7 ...	Clear
8	C:\Users\julie\Documents\Wor	...	S03		ignore		Comp.: 6 7 ...	Clear
9	C:\Users\julie\Documents\Wor	...	S03		probe		Comp.: 6 7 ...	Clear
10	C:\Users\julie\Documents\Wor	...	S04		memorize		Comp.: 1 2 ...	Clear

Important note: Removed datasets will not be saved before being deleted from EEGLAB memory

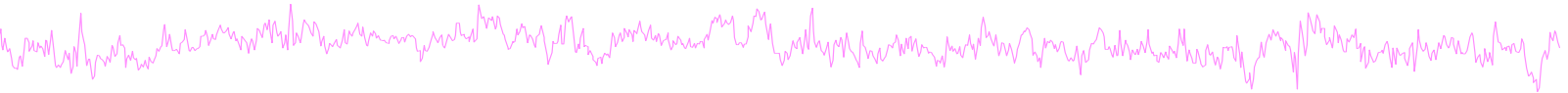
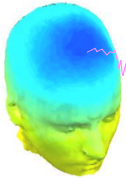
< Page 1 >

☐ Dataset info (condition, group, ...) differs from study info. [set] = Overwrite dataset info.

☐ Delete cluster information (to allow loading new datasets, set new components for clustering, etc.)

Help Cancel Ok

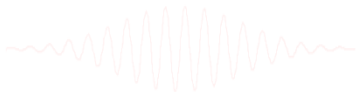
Load dataset info from commandline



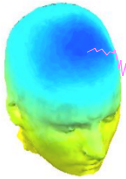
```
% create the STERN STUDY
eeglab
basedir = pwd;
subjs = {'S01' 'S02' 'S03' 'S04' 'S05' 'S06' 'S07' ...
         'S08' 'S09' 'S10' 'S11' 'S12' 'S13'};
commands = {}; % initialize STUDY dataset list

% Now loop through subjects and add to the STUDY
for subj = 1:length(subjs) % for each subject
    fileIgnore = fullfile(basedir, subjs{subj}, 'Ignore.set');
    fileMemorize = fullfile(basedir, subjs{subj}, 'Memorize.set');
    fileProbe = fullfile(basedir, subjs{subj}, 'Probe.set');
    commands = { commands{:} ...
                 {'index' 3*subj-2 'load' fileIgnore 'subject' subjs{subj} 'condition' 'Ignore' } ...
                 {'index' 3*subj-1 'load' fileMemorize 'subject' subjs{subj} 'condition' 'Memorize' } ...
                 {'index' 3*subj 'load' fileProbe 'subject' subjs{subj} 'condition' 'Probe' } };
end;
commands = { commands{:} { 'dipselect', 0.15 } };
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name', 'Sternberg', ...
    'task', 'Sternberg', 'commands', commands, ...
    'updatedat','off', 'filename', fullfile(basedir, 'stern_autogenerated.study'));

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

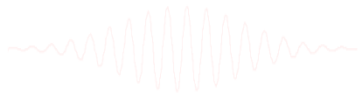


STUDY structure

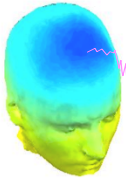


STUDY =

```
    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'}
    setind: [3x13 double]
    etc: [1x1 struct]
preclust: [1x1 struct]
cluster: [1x1 struct]
changrp: [1x71 struct]
    saved: 'yes'
```



Understanding STUDY structure



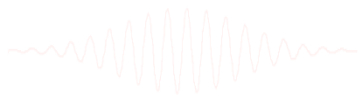
```
>> STUDY.datasetinfo(11) % access dataset 11  
ans =
```

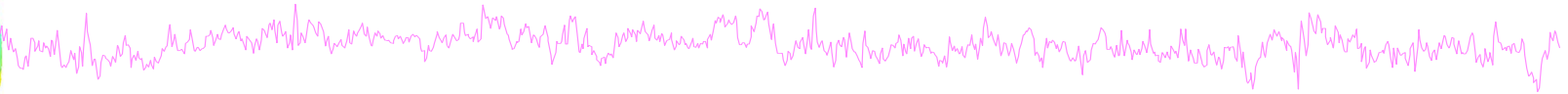
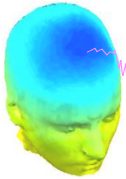
```
    filepath: [1x61 char]  
  filename: 'S04.set'  
   subject: 'S04'  
   session: []  
 condition: ''  
    group: ''  
trialinfo: 1x350 struct  
    index: 11  
    comps: [1x24 double]
```

Subject 4!

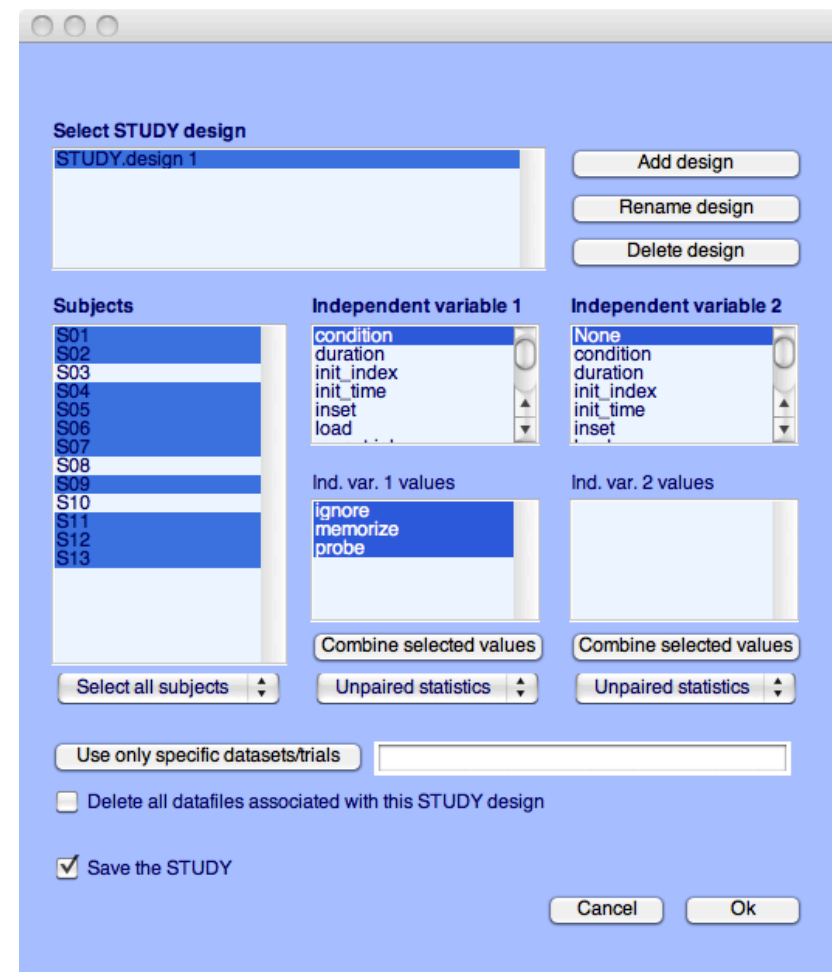
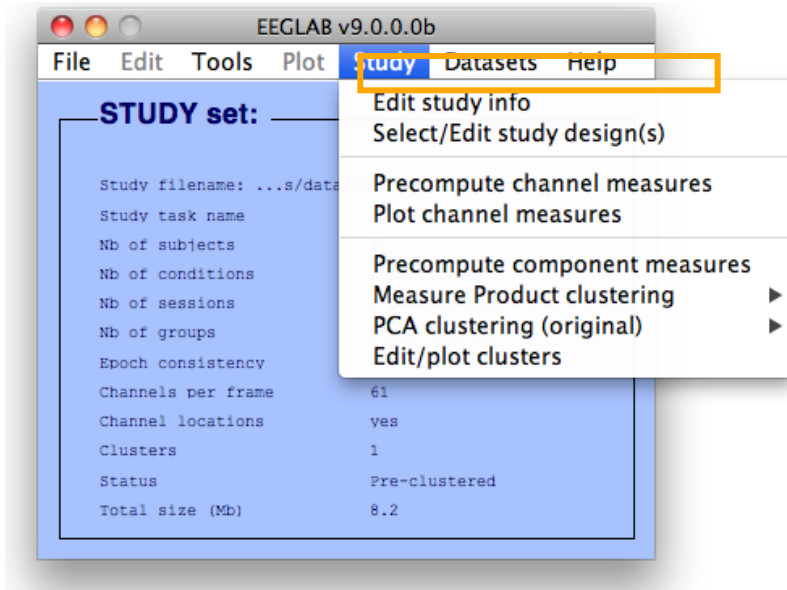
```
>> trialinfo(163) % access trial 163  
ans =
```

```
    stimtype: 'Memorize'  
    latency: 13201  
    duration: 0  
    ...
```



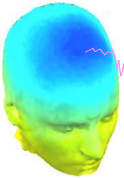


Select subjects



```
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'

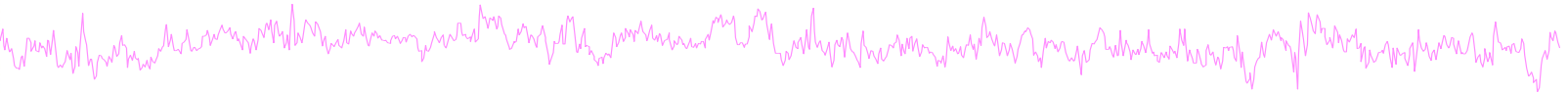
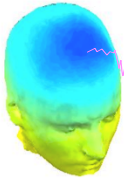
variable: [1x2 struct]
  (1).label : 'condition'
  (1).pairing: 'on'
  (1).value : {'ignore' 'memorize' 'probe'}
  (2).label : ''
  (2).pairing: 'off'
  (2).value : {}

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

include: {}

cells: [1x39 struct]
  dataset : 2
  trials : {[1x267 double]}
  value : {'ignore' ''}
  case : 'S01'
  filebase: '/Volumes/donnees/data/STUDYstern/S01/Ignore'
```

Precompute data measures



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

Select and compute component measures for later clustering -- pop_precomp()

Pre-compute component measures for STUDY 'Sternberg'

☒ Compute ERP/spectrum/ERSP only for components selected by RV (set) or for all components (unset)

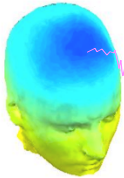
List of measures to precompute

☒ ERPs	Baseline ([min max] in ms)	[-200 0]	
☒ Power spectrum	Spectopo parameters		Test
☒ ERSPs	Time/freq. parameters	'cycles', [3 0.5], 'nfreqs', 100	Test
☒ ITCs			
☒ Scalp maps			

☐ Recompute even if present on disk

```
[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



Select and compute component measures for later clustering -- pop_precl...

Build pre-clustering matrix for STUDY set: Sternberg

Select the cluster to refine by sub-clustering (any existing sub-hierarchy will be overwritten)

ParentCluster 1 (336 ICs)

Note: Only measures that have been precomputed may be used for clustering.

Measures	Dims.	Norm.	Rel. Wt.
☒ spectra	10	☒ 1	
☐ ERPs	10	☒ 1	
☒ dipoles	3	☒ 10	
☐ scalp maps	10	☒ 1	
☒ ERSPs	10	☒ 1	
☐ ITCs	10	☒ 1	
☐ Final dimensions	10		

Help

☐ Use Measure Product clustering

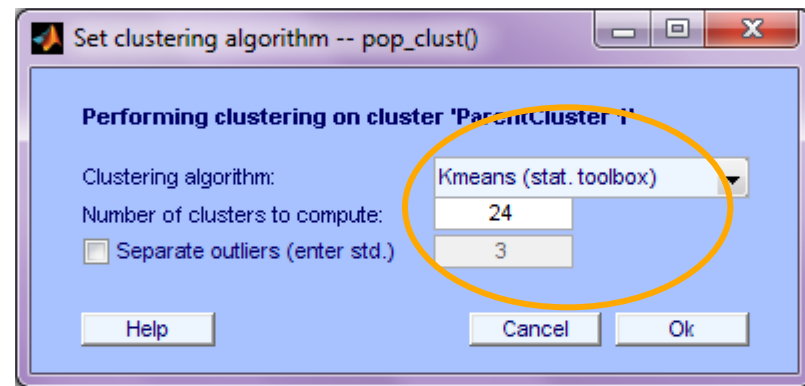
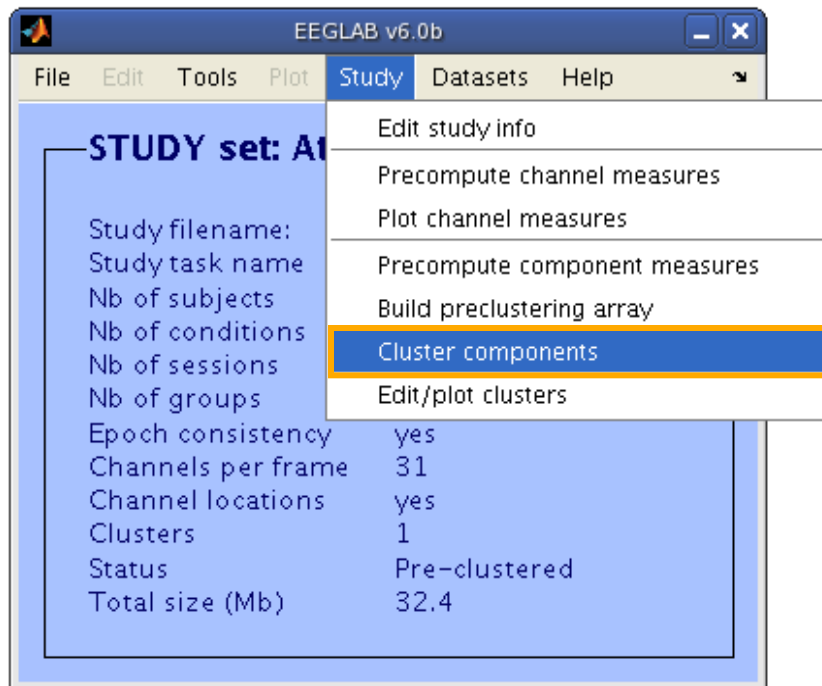
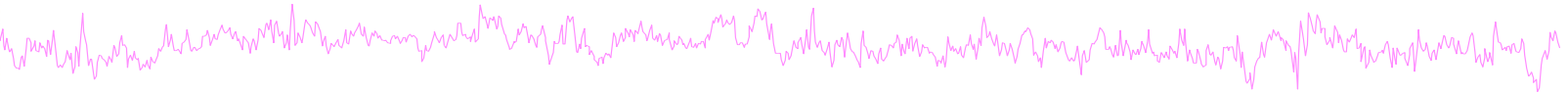
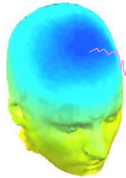
Help Cancel Ok

Additional settings for selected measures:

- spectra:** Freq. range [Hz] 3 25
- ERPs:** Time range [ms]
- dipoles:** Use channel values (dropdown), Absolute values (checked)
- ERSPs:** Time range [ms] 0 600, Freq. range [Hz] 3 30
- ITCs:** Time range [ms], Freq. range [Hz]

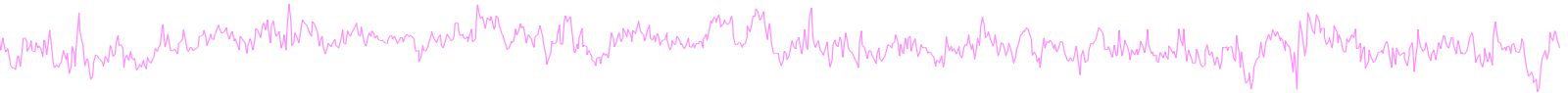
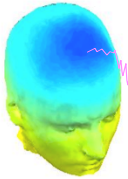
```
parentclust = 1;
[STUDY ALLEEG] = std_preclust(STUDY, ALLEEG, parentclust, {'spec', 'npca', 5, ...
    'norm', 1, 'weight', 1, 'freqrange', [3 25]}, {'erp', 'npca', 6, 'norm', 1, ...
    'weight', 1, 'timewindow', [0 400]}, {'scalp', 'npca', 10, 'norm', 1, 'weight', 1, ...
    'abso', 1}, {'dipoles', '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



```
nclusts = 25; % choose # of clusters to create
[STUDY] = pop_clust(STUDY, ALLEEG, 'algorithm', 'kmeans', 'clus_num', nclusts);
```

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]  
allinds: { [...] [...]; [...] [...]}  
setinds: { [...] [...]; [...] [...]}
```

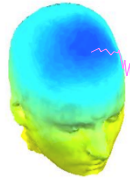
IC indices

dataset indices for ICs

Design IC indices

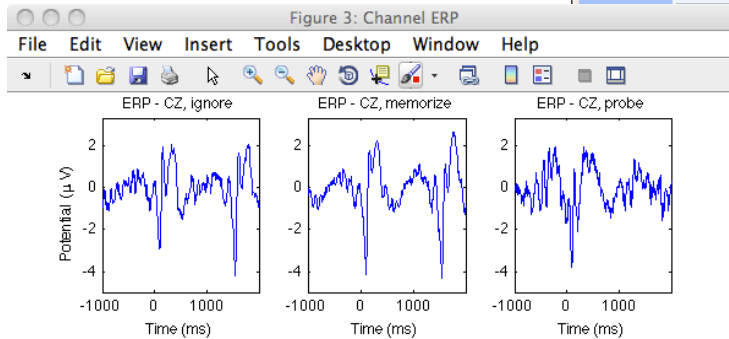
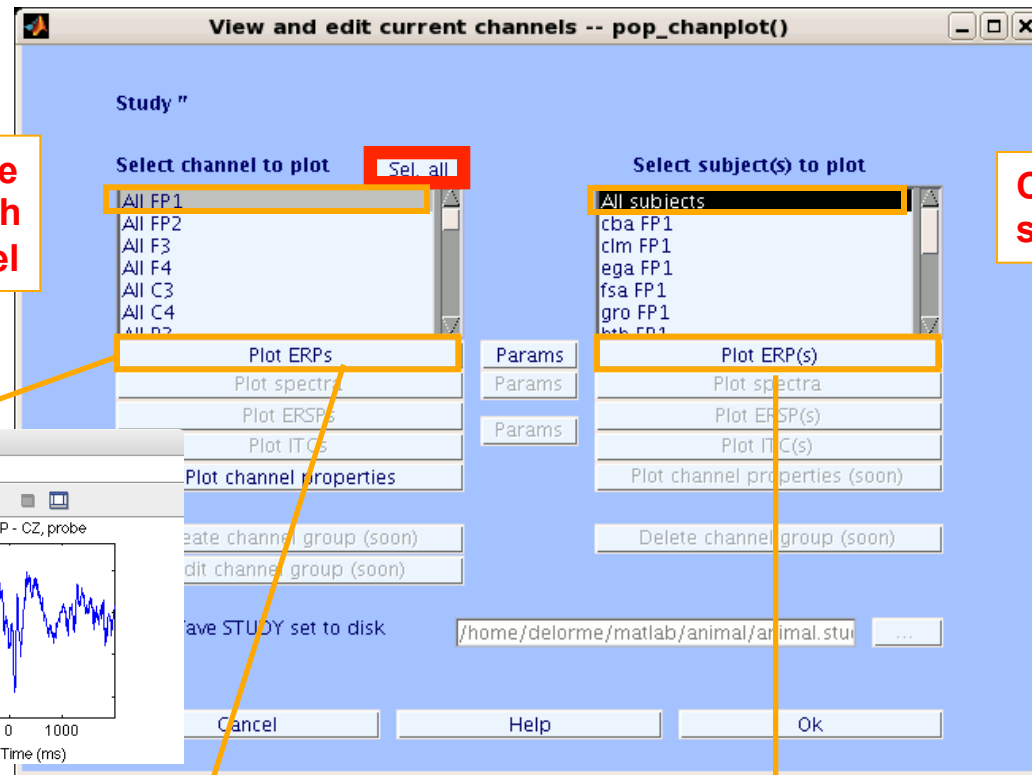
Design indices

6 = cluster index

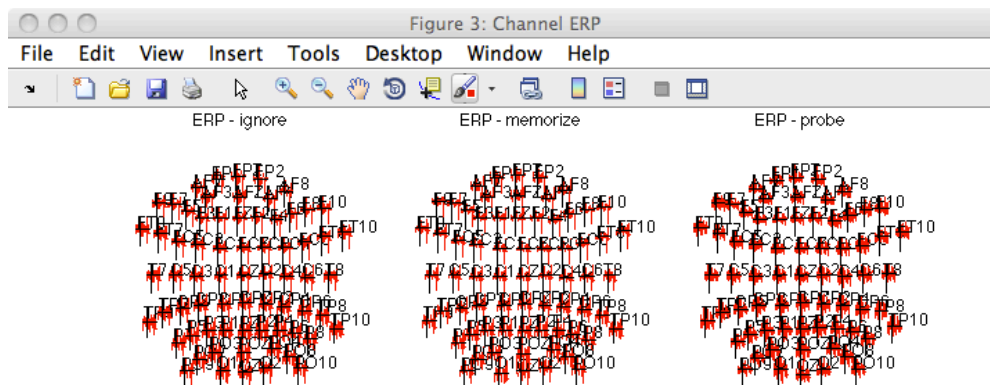


Choose which channel

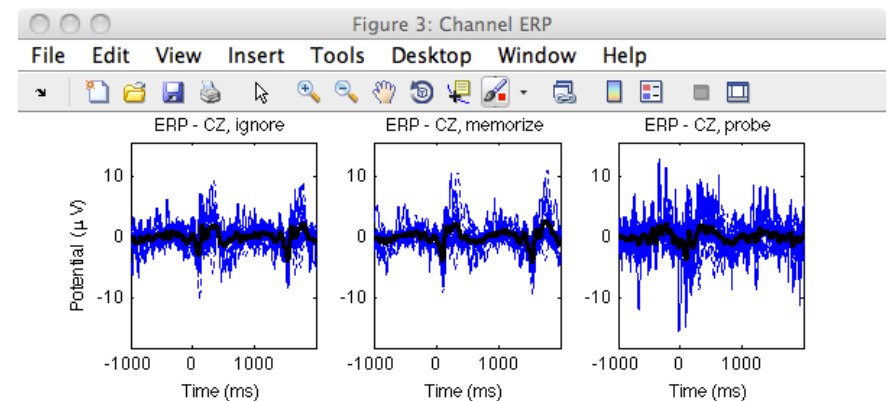
Choose which subject



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

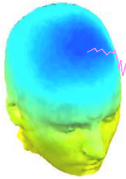


```
STUDY = std_erplot(STUDY,ALLEEG,'channels',{'FP1' ... });
```

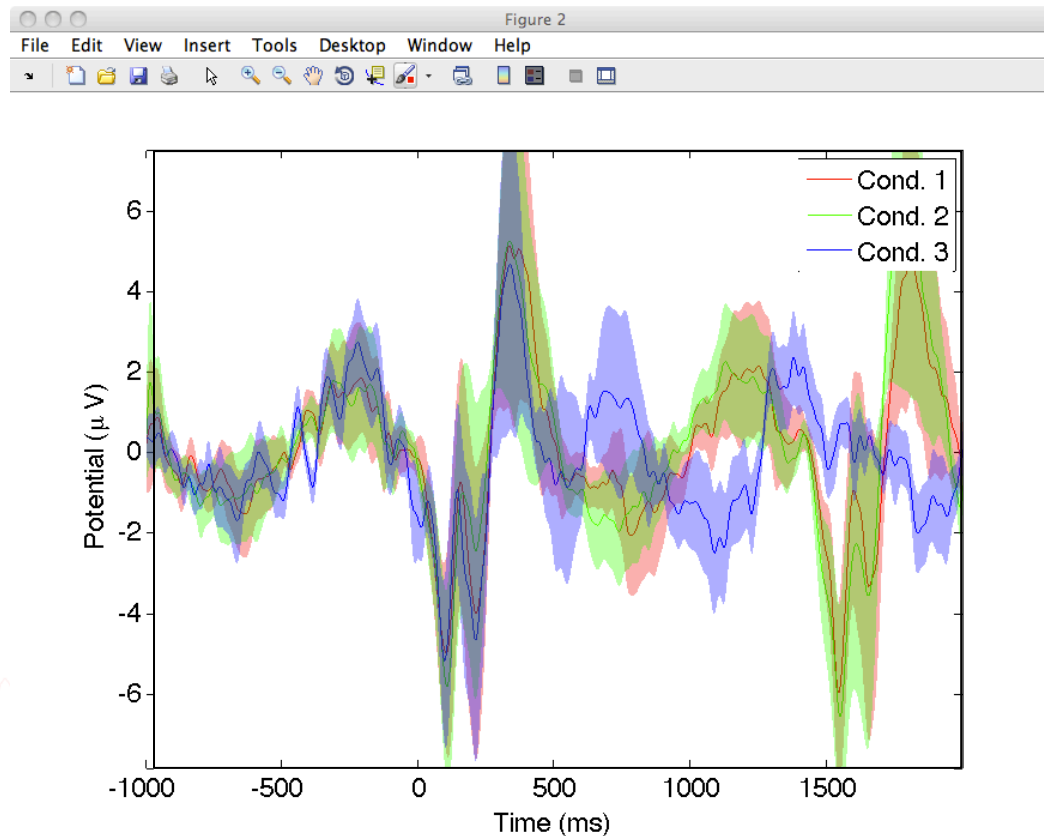


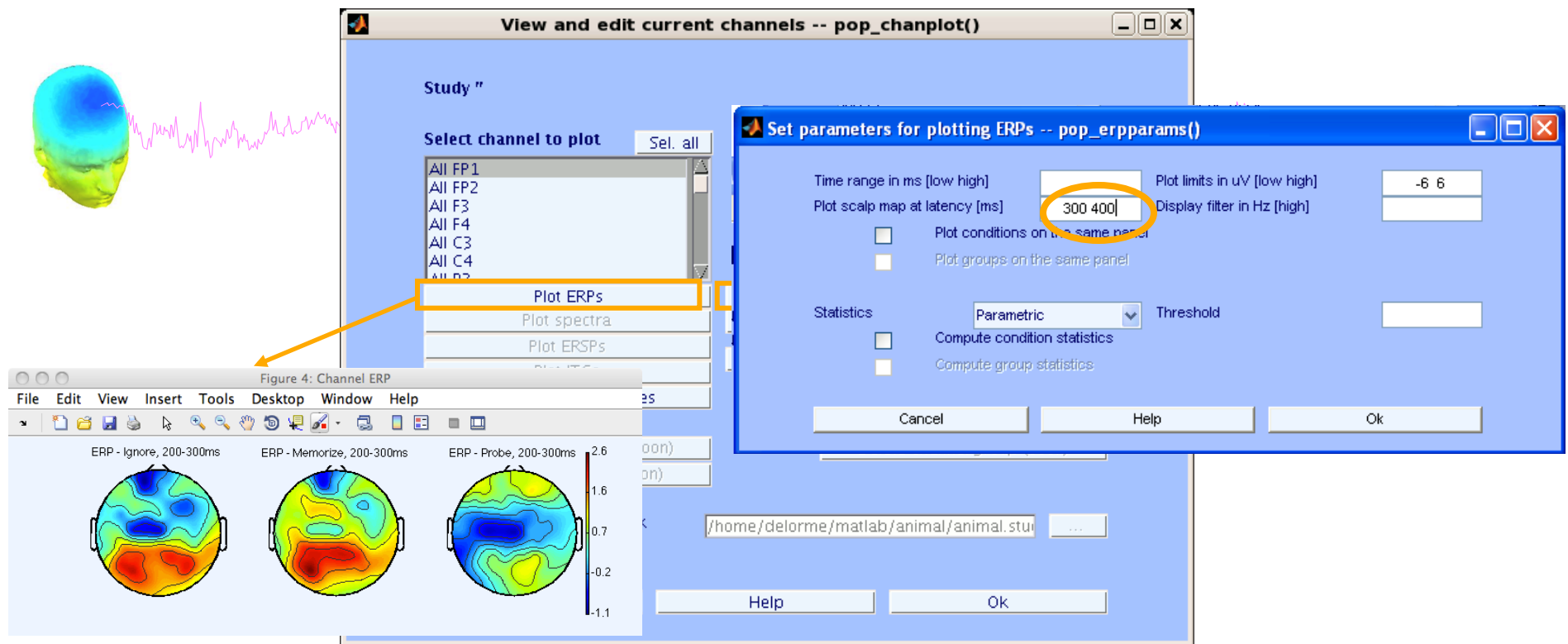
```
STUDY = std_erplot(STUDY,ALLEEG,'channels',{'FP1'}, 'plotsubjects', 'on' );
```

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]

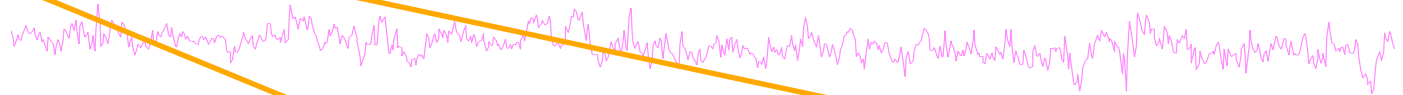
[1x67x13 double]

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

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

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

Exporting text file



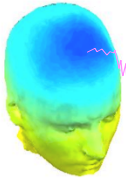
-0.13	-0.4	3.7	-0.9	-1.5	0.23	-0.98	1.8	2.3	-1.4	-2.8	-0.03	3.5
-0.54	-1.3	3.6	-1.1	-1.2	0.62	-0.91	1.6	2.2	-0.98	-7.7	-0.42	3.2
-0.77	-0.06	3.6	-1.4	-1.2	0.78	-0.91	1.2	2.1	-0.66	-0.76	-1	2.5
-0.61	-0.83	3.7	-1.2	-1.2	0.53	-0.88	1.1	1.7	-1.2	-1.8	-1.2	1.6
-0.34	-0.79	3.7	-0.98	-1.2	0.17	-0.72	1	1.4	-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);
```

STUDY Script



```
% create the STERN STUDY
eeglab
basedir = pwd;
subjs = {'S01' 'S02' 'S03' 'S04' 'S05' 'S06' 'S07' ...
        'S08' 'S09' 'S10' 'S11' 'S12' 'S13'};
commands = {}; % initialize STUDY dataset list

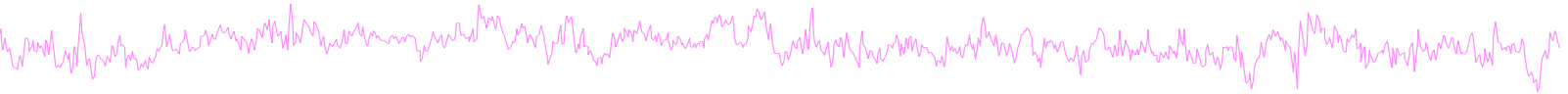
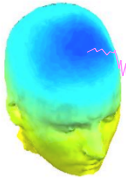
% Now loop through subjects and add to the STUDY
for subj = 1:length(subjs) % for each subject
    fileIgnore = fullfile(basedir, subjs{subj}, 'Ignore.set');
    fileMemorize = fullfile(basedir, subjs{subj}, 'Memorize.set');
    fileProbe = fullfile(basedir, subjs{subj}, 'Probe.set');
    commands = { commands{:} ...
        {'index' 3*subj-2 'load' fileIgnore 'subject' subjs{subj} 'condition' 'Ignore' } ...
        {'index' 3*subj-1 'load' fileMemorize 'subject' subjs{subj} 'condition' 'Memorize' } ...
        {'index' 3*subj 'load' fileProbe 'subject' subjs{subj} 'condition' 'Probe' } };
end;
commands = { commands{:} { 'dipselect', 0.15 } };
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name', 'Sternberg', ...
    'task', 'Sternberg', 'commands', commands, ...
    'updatedat','off', 'filename', fullfile(basedir, 'stern_autogenerated.study'));

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

% compute and plot ERP scalp maps between 200 and 300 ms
[STUDY ALLEEG] = std_precomp(STUDY, ALLEEG, {}, 'interp', 'on', 'allcomps', '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'});

% export the data to text file
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);
```

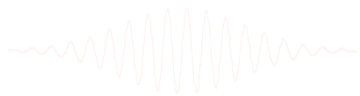
STUDY datafiles



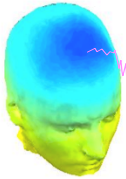
Datafiles (new scheme)

- *design1_S01_ignore.daterp* (ERPs)
- *design1_S01_ignore.datspec* (power spectra)
- *design1_S01_ignore.datersp* (ERSPs)
- *design1_S01_ignore.datitc* (ITCs)

- *design1_S01_ignore.icaerp* (ERPs)
- *design1_S01_ignore.icaspec* (power spectra)
- *design1_S01_ignore.icatopo* (scalp maps)
- *design1_S01_ignore.icaersp* (ERSPs)
- *design1_S01_ignore.icaitc* (ITCs)



Exercises



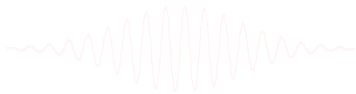
Suggestion for exercises:

Load stern.study in STUDY folder

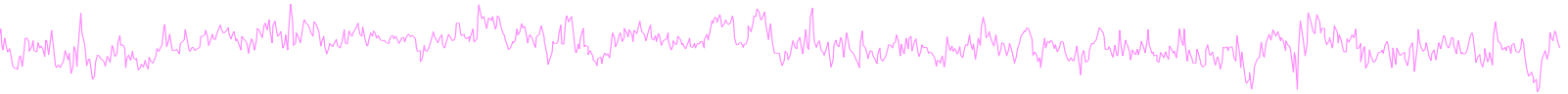
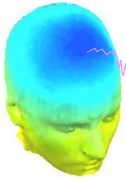
Choose a cluster to investigate. Plot ERP scalp topographies at 100 ms using cluster correction for multiple comparison (Fieldtrip statistics – monte carlo)

Get command line call for plotting ERP.

Modify the script to plot spectrum at 10 Hz instead of ERP.



STUDY datafiles



```
tmp = load('-mat', 'design1_S01_ignore.daterp');  
>> tmp
```

```
tmp =
```

```
chan1: [1x750 double]  
chan2: [1x750 double]  
chan3: [1x750 double]  
chan4: [1x750 double]  
chan5: [1x750 double]  
chan6: [1x750 double]  
chan7: [1x750 double]
```

```
...
```

```
chan69: [1x750 double]  
chan70: [1x750 double]  
chan71: [1x750 double]
```

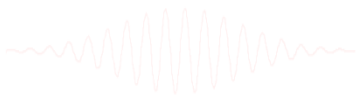
```
labels: {1x71 cell}
```

```
times: [1x750 double]
```

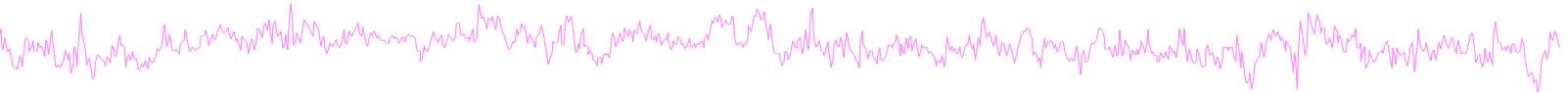
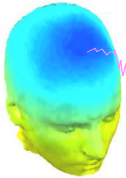
```
datatype: 'ERP'
```

```
parameters: {'rmcomps' {1x1 cell} 'interp' [1x71 struct]}
```

```
datafile: '/Volumes/donnees/data/STUDYstern/S01/Ignore.set'
```



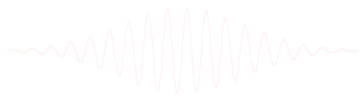
STUDY datafiles



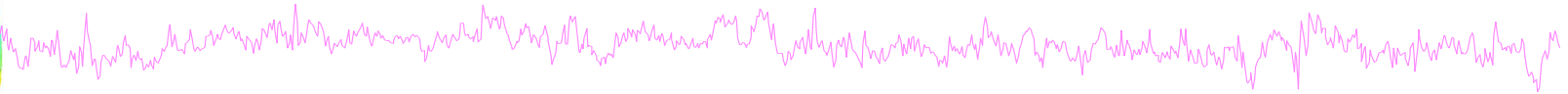
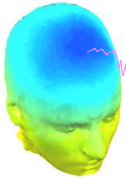
```
>> tmp = load('-mat', 'design1_S01_ignore.datersp');  
>> tmp
```

```
tmp =
```

```
      chan1_ersp: [50x100 single]  
      chan1_erspbase: [1x50 single]  
      chan1_erspboot: []  
      chan2_ersp: [50x100 single]  
      chan2_erspbase: [1x50 single]  
      chan2_erspboot: []  
           freqs: [1x50 double]  
           times: [1x100 double]  
      datatype: 'ERSP'  
      parameters: {1x26 cell}  
      chanlabels: {1x71 cell}  
      datafile: '/Volumes/donnees/data/STUDYstern/S01/Ignore.set'
```



STUDY datafiles



```
>> tmp = load('-mat', 'design1_S01_ignore.icaersp');  
>> tmp
```

```
tmp =
```

```
    comp1_ersp: [50x100 single]  
    comp1_erspbase: [1x50 single]  
    comp1_erspboot: []  
    comp2_ersp: [50x100 single]  
    comp2_erspbase: [1x50 single]  
    comp2_erspboot: []  
        freqs: [1x50 double]  
        times: [1x100 double]  
    datatype: 'ERSP'  
    parameters: {1x26 cell}  
    chanlabels: {1x71 cell}  
    datafile: '/Volumes/donnees/data/STUDYstern/S01/Ignore.set'
```

