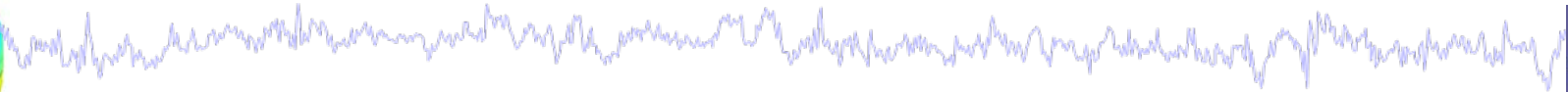


# Advanced IC analysis



## Task 1

Search EEG.event structure

## Task 2

Use *newtimef()* to compare conditions

## Task 3

Plot a RT-sorted component ERP image

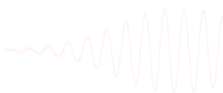
Plot a type-sorted component ERP image

Plot a load-sorted component ERP image

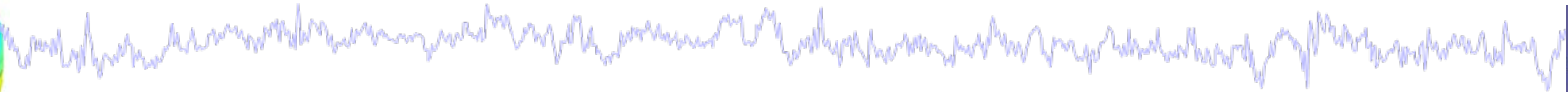
## Task 4

Use outputs from commandline ERP image

**Exercise...**



# Advanced IC analysis



## Task 1

Search EEG.event structure

## Task 2

Use *newtimef()* to compare conditions

## Task 3

Plot a RT-sorted component ERP image

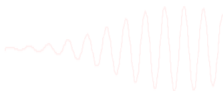
Plot a type-sorted component ERP image

Plot a load-sorted component ERP image

## Task 4

Use outputs from commandline ERP image

## Exercise...



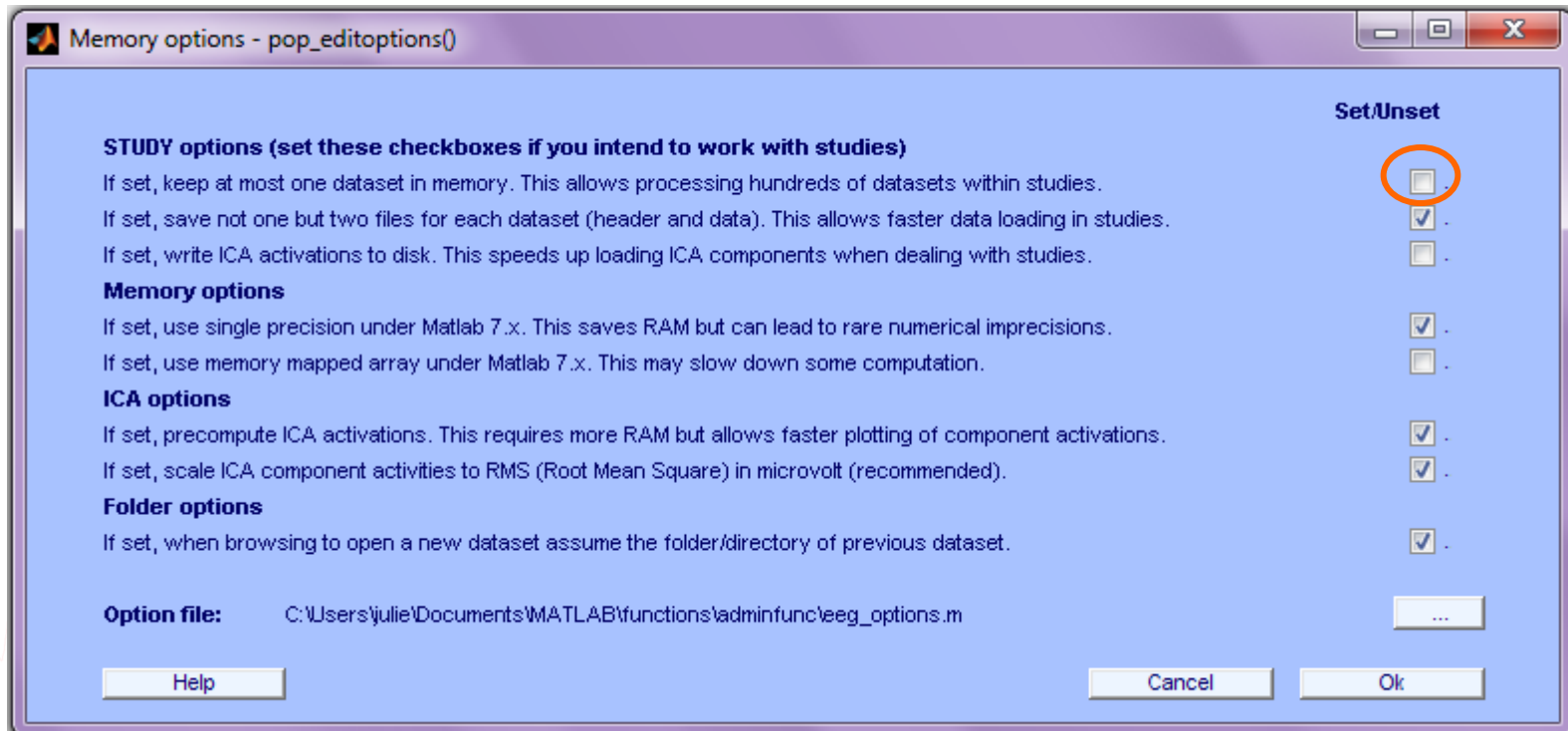
# Set memory options



% you will need memory options to keep more than one dataset in memory at once:

```
pop_editoptions( 'option_storedisk', 0, 'option_savetwofiles', 1,...  
    'option_saveica', 0, 'option_single', 1, 'option_mmapdata', 0,...  
    'option_computeica', 1, 'option_scaleicarms', 1, 'option_rememberfolder', 1);
```

%-----



# The example data: Sternberg working memory



## File

../Data/stern.set

## Data

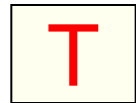
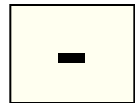
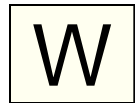
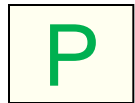
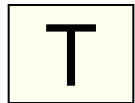
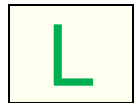
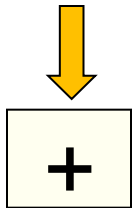
Continuous data (not epoched), ref'd to right mastoid

## Task

3-7 letters to memorize, among 1-5 letters to ignore  
50% chance of probe letter being 'in-set'

## Fixation

(5 sec)



Memorize

Ignore

SOA

(1.4 sec)



## Maintenance

(2-4 sec)

Probe

RT

RESPONSE

*Was this letter in the memorized set?*

# Color-coding for tutorial scripts



**%%%% Color-coding for scripts:**

**% Green text is comments**

**myvariable** (bold, red) = pre-defined variable

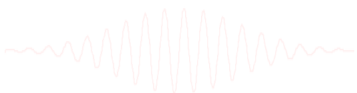
**for ep ... end** = ... (bold, blue) = 'for loop' variable

**if ... end** = (bold, cyan) = 'if loop' statement

**newtimef()** (bold, purple) = function call

**[outdata,outvar,outtrials,...]** (brown, in brackets)

= function output variables



# Search events for specific event type



```
% OBJECTIVES:
```

```
% 1) Find all Memorize letters that were preceded by an ignore letter
```

```
% 2) Find all Memorize letters that were preceded by a memorize letter
```

```
%-----
```

```
% hint: 'memorize' event codes are single letters
```

```
epochidxM = []; % Mem preceded by a mem letter
```

```
epochidxG = []; % Mem preceded by an ignore letter
```

```
for ev = 2:length(EEG.event)
```

```
    if length(EEG.event(ev).type)==1 & length(EEG.event(ev-1).type)==1
```

```
        epochidxM = [epochidxM, ev]; % save this event
```

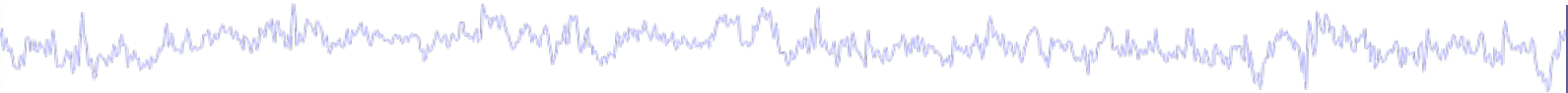
```
    elseif length(EEG.event(ev).type)==1 & EEG.event(ev-1).type(1)=='g'
```

```
        epochidxG = [epochidxG, ev]; % save this event
```

```
    end;
```

```
end;
```

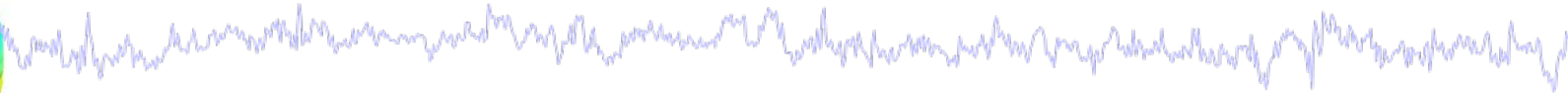
# Epoch on selected events



**% Epoch continuous data around selected events**

```
%-----  
[EEG, indices] = pop_epoch( EEG, [], [-2 2], 'eventindices', epochidxG);  
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, ...  
    'setname', 'Mem after Ignore letter', 'gui', 'off');  
EEG = pop_autorej(EEG, 'nogui', 'on'); % Auto-reject noisy epochs  
[ALLEEG EEG CURRENTSET]=pop_newset(ALLEEG, EEG, CURRENTSET, 'retrieve', 1);  
  
[EEG, indices] = pop_epoch( EEG, [], [-2 2], 'eventindices', epochidxM);  
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, ...  
    'overwrite', 'on', 'setname', 'Mem after Mem letter', 'gui', 'off');  
EEG = pop_autorej(EEG, 'nogui', 'on'); % Auto-reject noisy epochs  
eeglab redraw
```

# Confirm datasets contain expected epochs



```
>> [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET, 'retrieve',1);
```

```
>> EEG.epoch(2) %--- Select several random epochs
```

```
ans =
```

```
event: [4 5 6]
```

```
eventlatency: {[-1.4400e+003] [0] [1.4440e+003]}
```

```
eventload: {[1] [2] [3]}
```

```
eventtype: {'R' 'N' 'Z'}
```

```
eventurevent: {[5] [6] [7]}
```

```
>> [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 2, 'retrieve',2);
```

```
>> EEG.epoch(2)
```

```
ans =
```

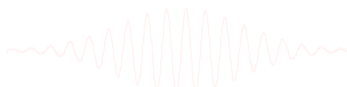
```
event: [4 5 6]
```

```
eventlatency: {[-1.4400e+003] [0] [1.4440e+003]}
```

```
eventload: {[0] [0] [1]}
```

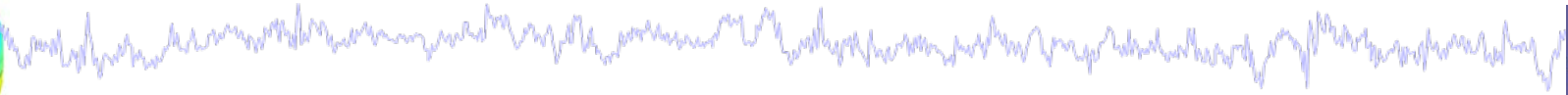
```
eventtype: {'gC' 'Z' 'L'}
```

```
eventurevent: {[15] [16] [17]}
```





# Advanced IC analysis



## Task 1

Search EEG.event structure

## Task 2

Use *newtimef()* to compare conditions

## Task 3

Plot a RT-sorted component ERP image

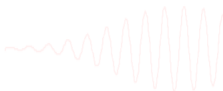
Plot a type-sorted component ERP image

Plot a load-sorted component ERP image

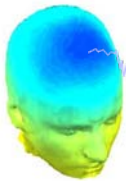
## Task 4

Use outputs from commandline ERP image

## Exercise...



# Get *newtimef()* command from GUI call



Plot component time frequency -- pop\_newtimef()

Component number	4
Sub epoch time limits [min max] (msec)	-2000 1996
Frequency limits [min max] (Hz) or sequence	3 50
Baseline limits [min max] (msec) (0->pre-stim.)	-200 0
Wavelet cycles [min max/fact] or sequence	3 0.5
ERSP color limits [max] (min=-max)	
ITC color limits [max]	
Bootstrap significance level (Ex: 0.01 -> 1%)	.01
Optional newtimef() arguments (see Help)	

Use 200 time points  
Use limits, padding 1  
Use divisive baseline  
Use limits  
 Log spac...  
 No baseline  
 Use FFT  
 see log power (set)  
 plot ITC phase (set)  
 FDR correct (set)

Plot Event Related Spectral Power     Plot Inter Trial Coherence

Help

EEGLAB v7.1.7.18b

File Edit Tools Plot

## #2: Step

Filename:  
Channels  
Frames per  
Epochs  
Events  
Sampling  
Epoch start  
Epoch end  
Reference  
Channel loc  
ICA weight  
Dataset si

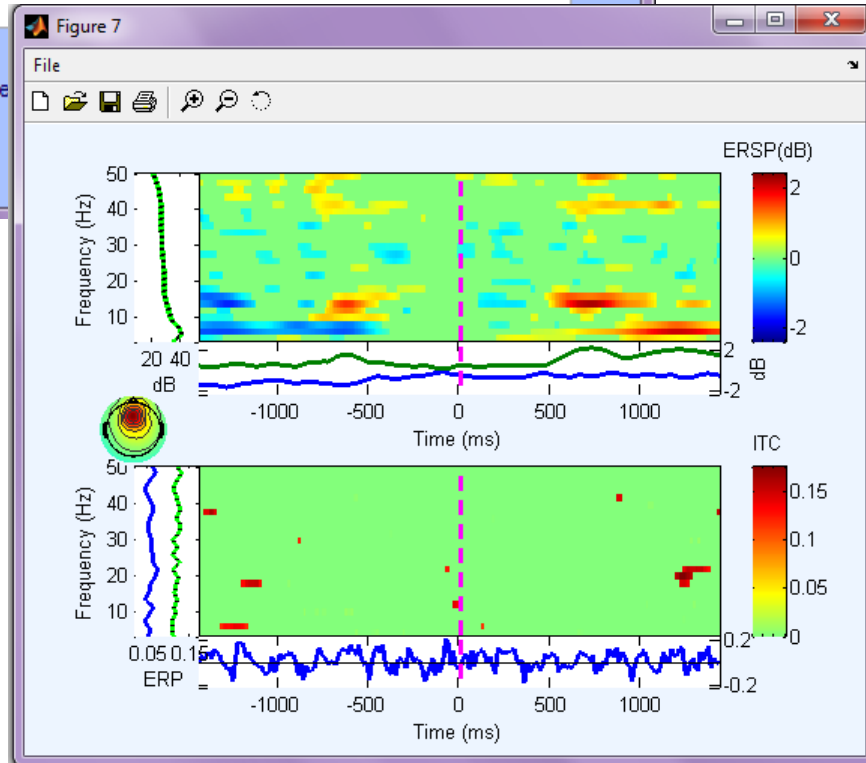
Sum/Compare ERPs

- 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

- Channel time-frequency
- Channel cross-coherence
- Component time-frequency**
- Component cross-coherence



# Use *newtimef()* to compare conditions



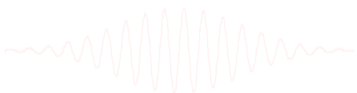
```
>> eegh
```

```
figure; pop_newtimef( EEG, 0, 4, [-2000 1996], [3 0.5] ,...  
  'topovec', EEG.icawinv(:,4), 'elocs', EEG.chanlocs,...  
  'chaninfo', EEG.chaninfo, 'baseline',[-200 0], 'alpha',.01,...  
  'freqs', [3 50], 'plotphase', 'off', 'pdratio', 1);
```

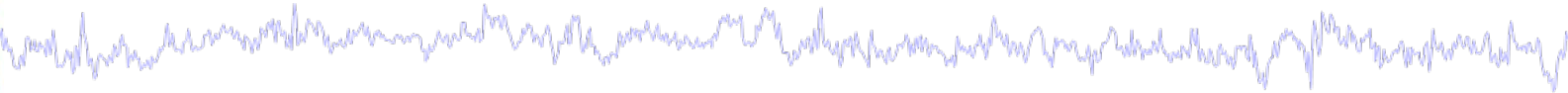
```
>> help newtimef
```

Example using data from two conditions (EEG versus ALLEEG(2)):

```
>> [ersp,itc,powbase,times,freqs,erspboot,itcboot] = ...  
  newtimef( {EEG.data(chan, :, :) ALLEEG(2).data(chan, :, :)} , ...  
  EEG.pnts, [EEG.xmin EEG.xmax]*1000, EEG.srate, cycles);
```



# Task 3: Use *newtimef()* to compare conditions

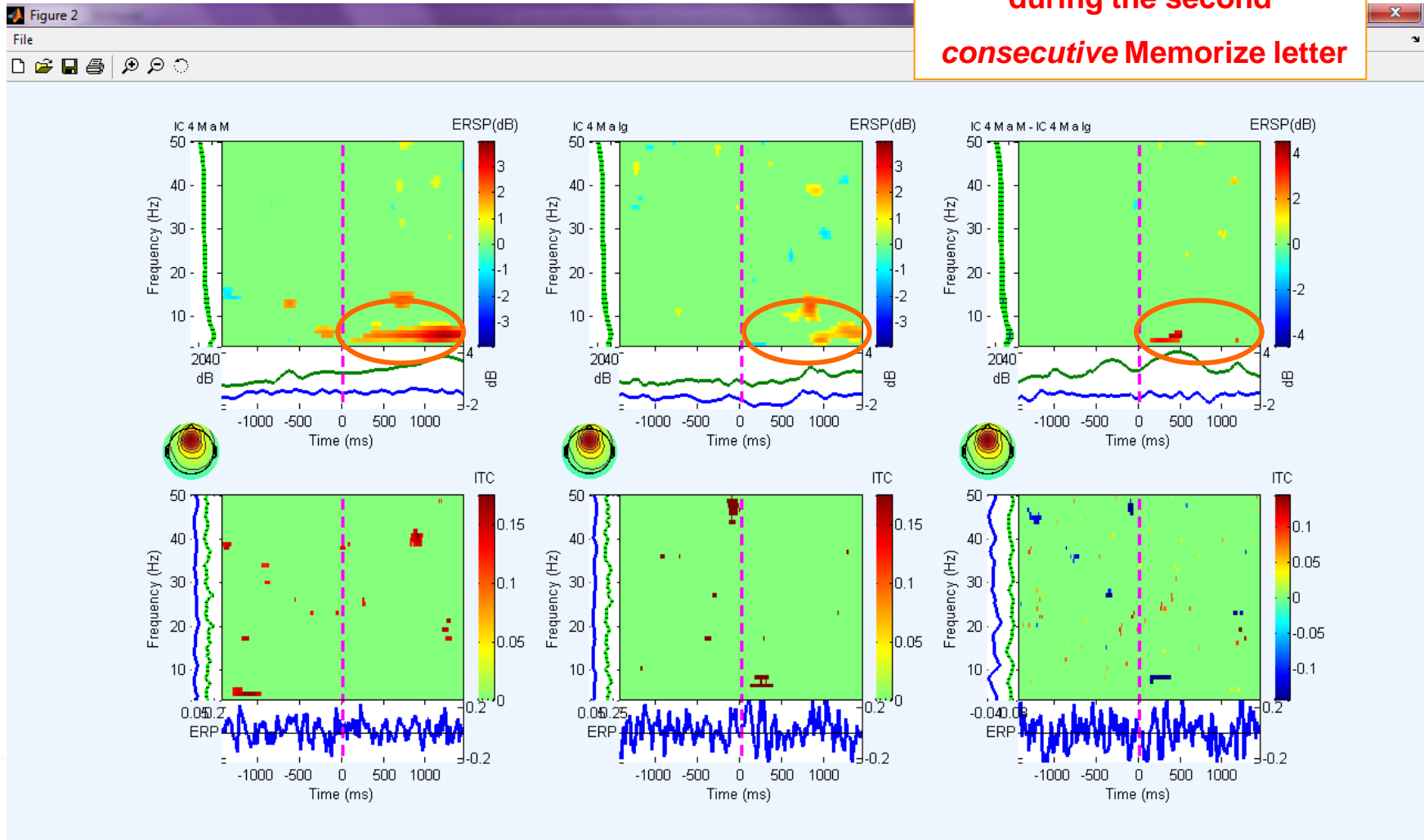


```
% adapt to your script:-----  
  
% data from datasets 1 (mem after mem)  
  
% and 2 (mem after ignore)  
  
ic = 4; % choose a component  
  
[ersp,itc,powbase,times,freqs,erspboot,itcboot] = ...  
newtimef( {ALLEEG(1).icaact(ic,:),ALLEEG(2).icaact(ic,:)}, ...  
data  
EEG.pnts, [EEG.xmin EEG.xmax]*1000, EEG.srate, [3 .5],...  
'type', 'phasecoher', 'topovec', EEG.icawinv(:,ic), ...  
'elocs', EEG.chanlocs, 'chaninfo', EEG.chaninfo, ...  
'title', {['IC ',int2str(ic),' M a M'],...  
condition 1  
['IC ',int2str(ic),' M a Ig']},...  
condition 2  
'baseline',[-200 0], 'alpha',.01,'pdratio', 1,...  
'plotphase','off', 'freqs', [3 50]);
```

# Compare conditions with *newtimef()*



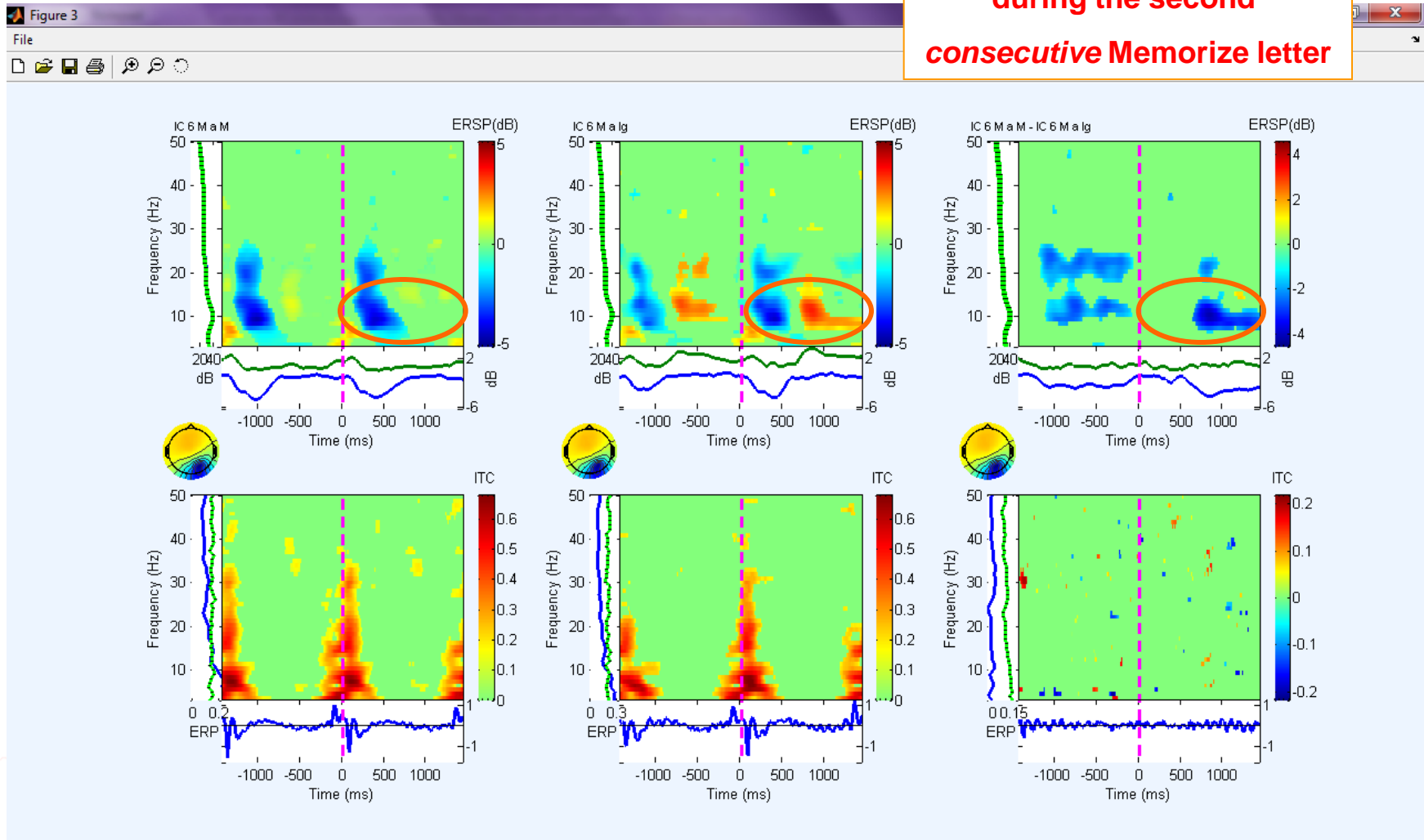
Higher frontal theta  
during the second  
*consecutive* Memorize letter



# Compare conditions with *newtimef()*



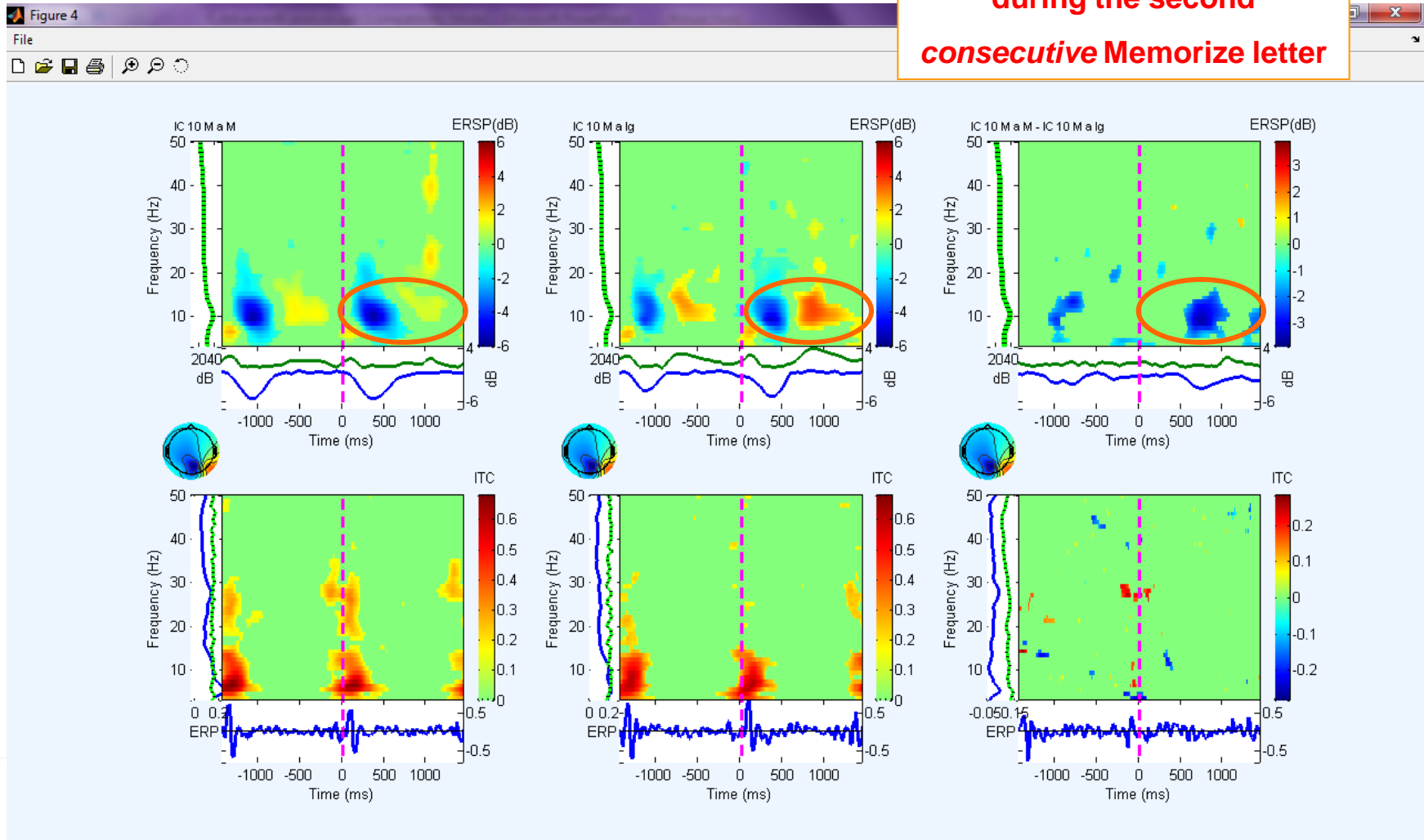
Less *parietal* alpha power  
during the second  
*consecutive* Memorize letter



# Compare conditions with *newtimef()*



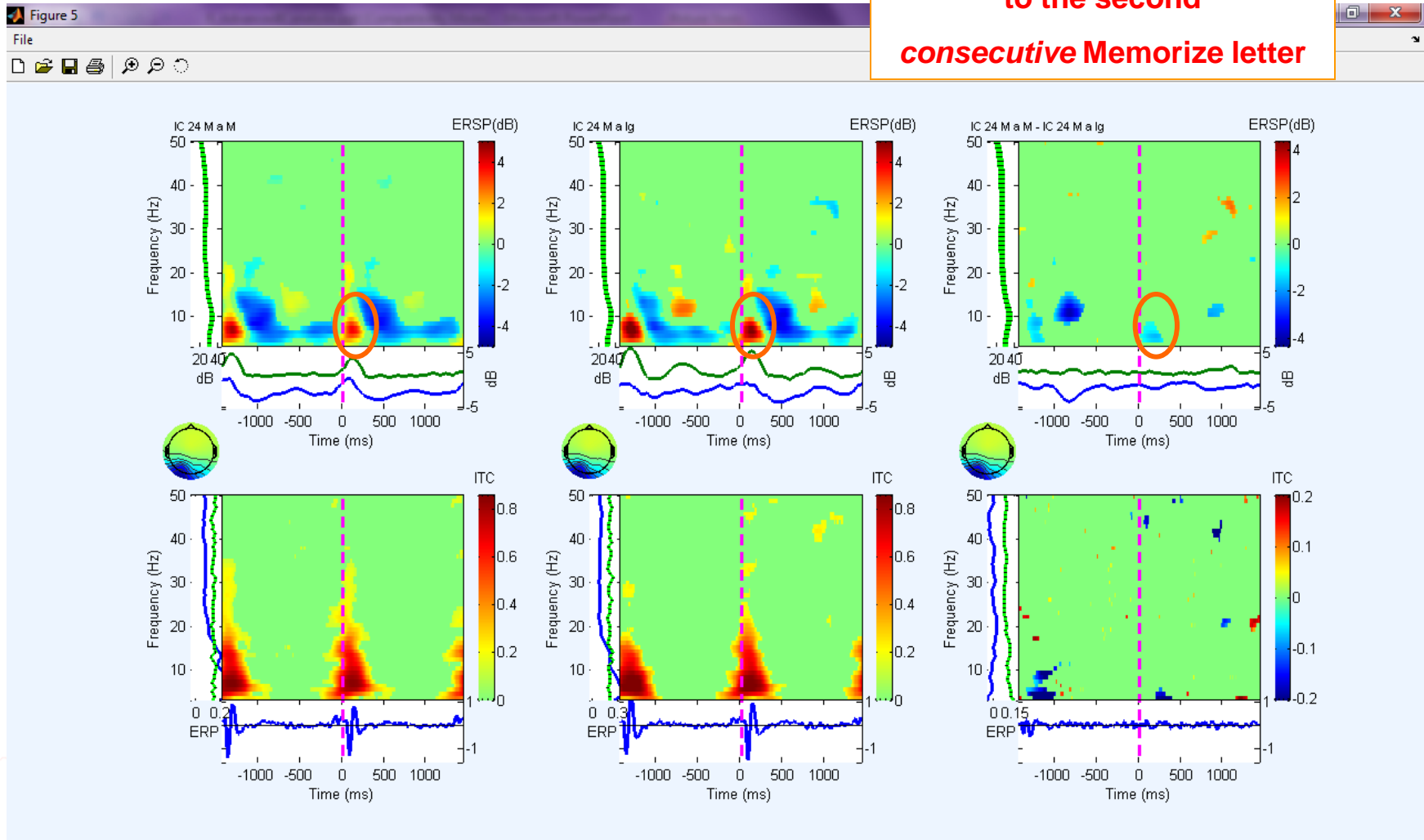
Less *occipital* alpha power  
during the second  
*consecutive* Memorize letter



# Compare conditions with *newtimef()*

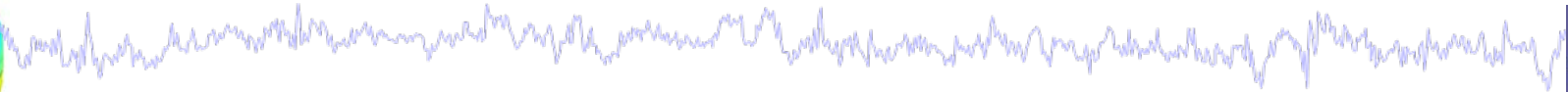


**Less visual evoked potential  
to the second  
consecutive Memorize letter**





# Advanced IC analysis



## Task 1

Search EEG.event structure

## Task 2

Use *newtimef()* to compare conditions

## Task 3

Plot a RT-sorted component ERP image

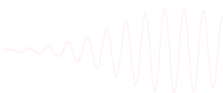
Plot a type-sorted component ERP image

Plot a load-sorted component ERP image

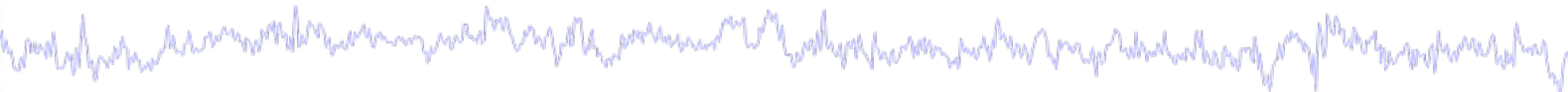
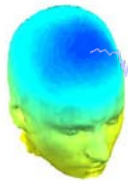
## Task 4

Use outputs from commandline ERP image

## Exercise...



# Sort ERP image by RT



EEGLAB v7.2.7.18b

File Edit Tools **Plot** Study Datasets Help

**#1: Step**

- 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

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

Component ERP image -- pop\_erpimage()

**Component(s)** 17

**Project to channel #**

**Smoothing** 5

**Downsampling** 1

**Time limits (ms)** -3000 3000

**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 Align

latency in' out no

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

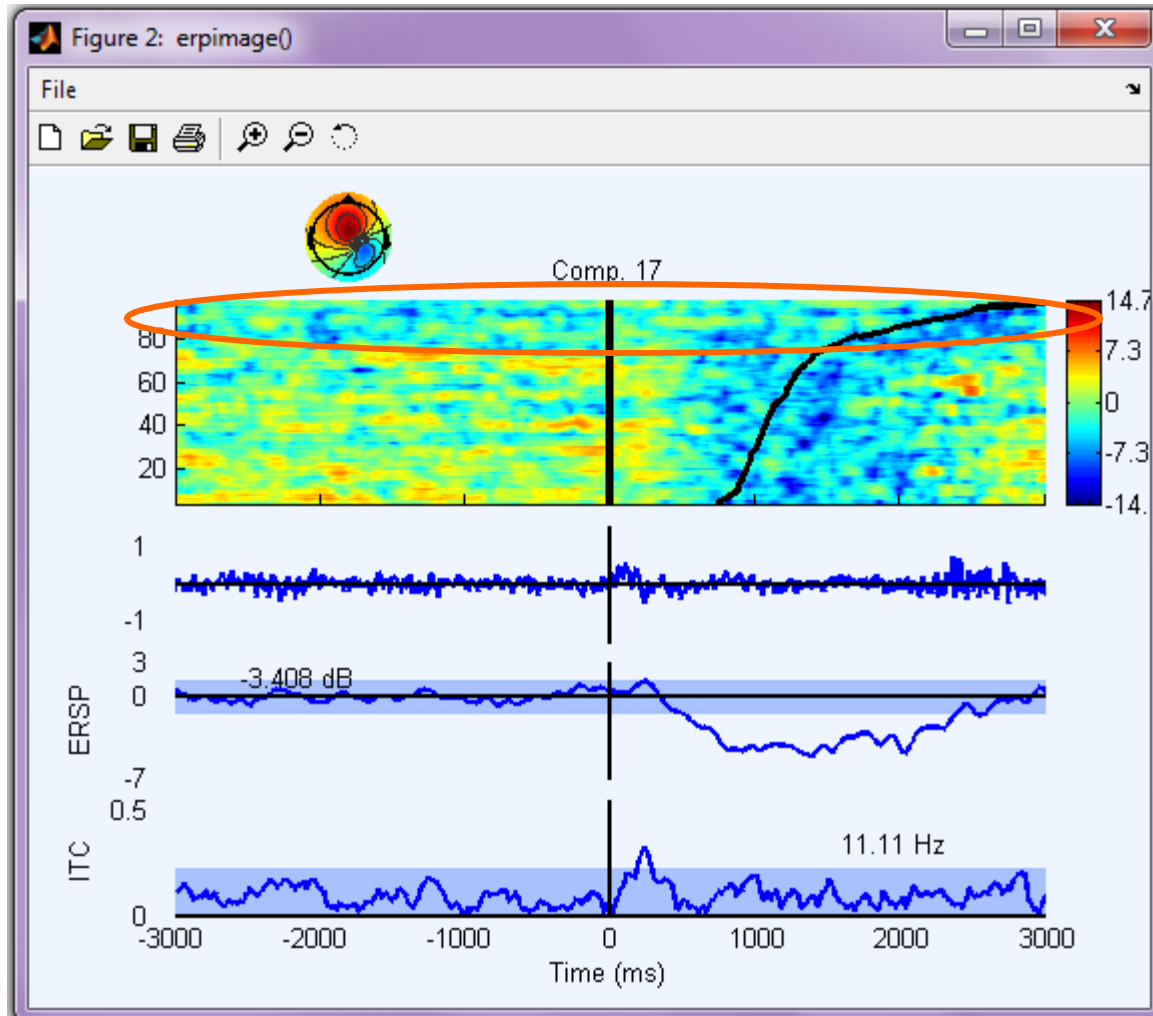
8 12 0.01 (Requires signif.)

**Other options**

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

Help Cancel Ok

# Sort ERP image by RT



# Sort ERP image by response type



Component ERP image -- pop\_erpimage()

Component(s) 3  
Project to channel #  
Smoothing 5  
Downsampling 1  
Time limits (ms) -300 1000

Figure title  
 Plot scalp map  
 Plot ERP  
 Plot colorbar

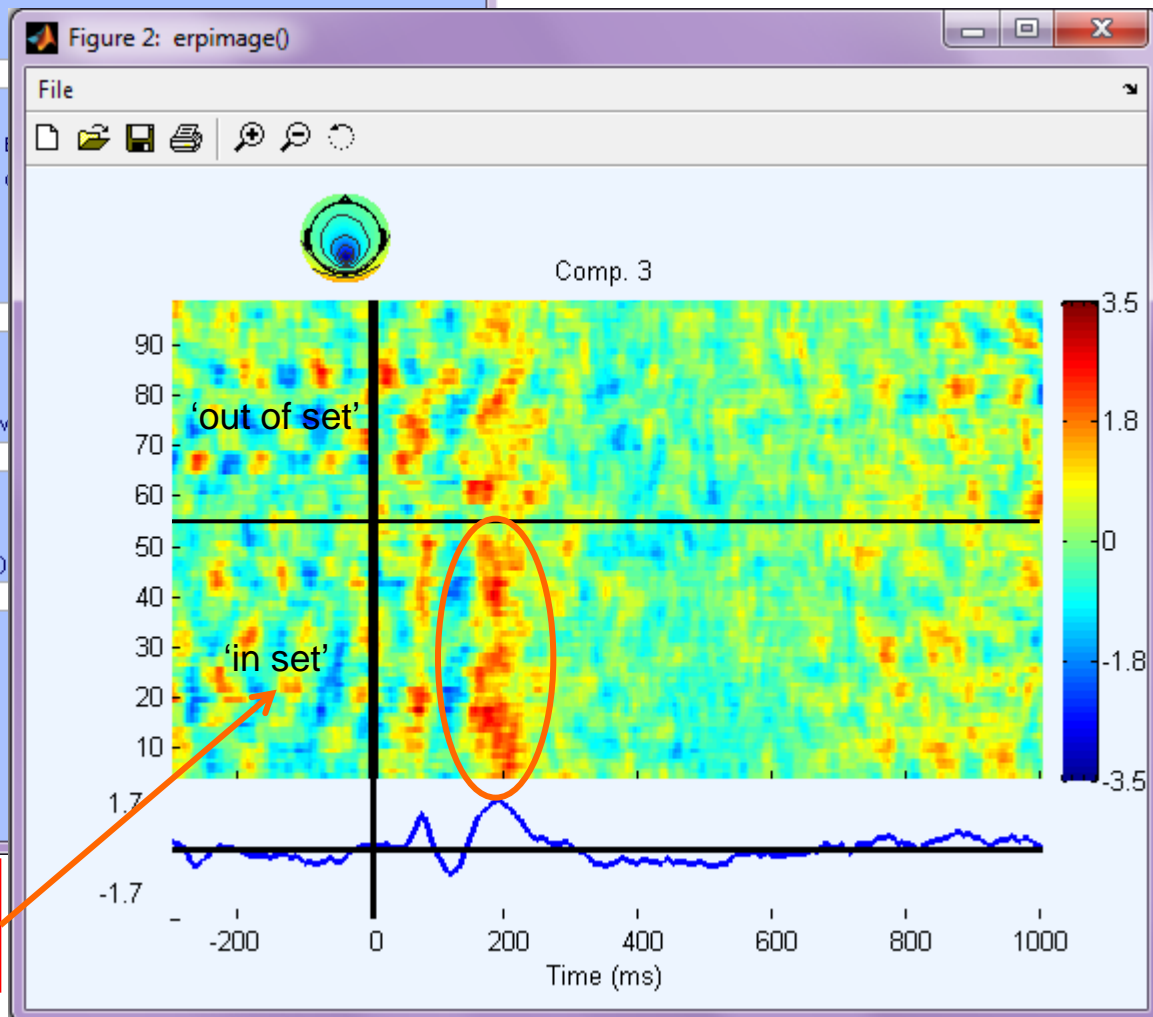
Sort/align trials by epoch event values  
Epoch-sorting field Event type(s) Event time range Rescale  
type in 'out'

Sort trials by phase  
Frequency (Hz | minHz maxHz) Percent low-amp. trials to ignore Window

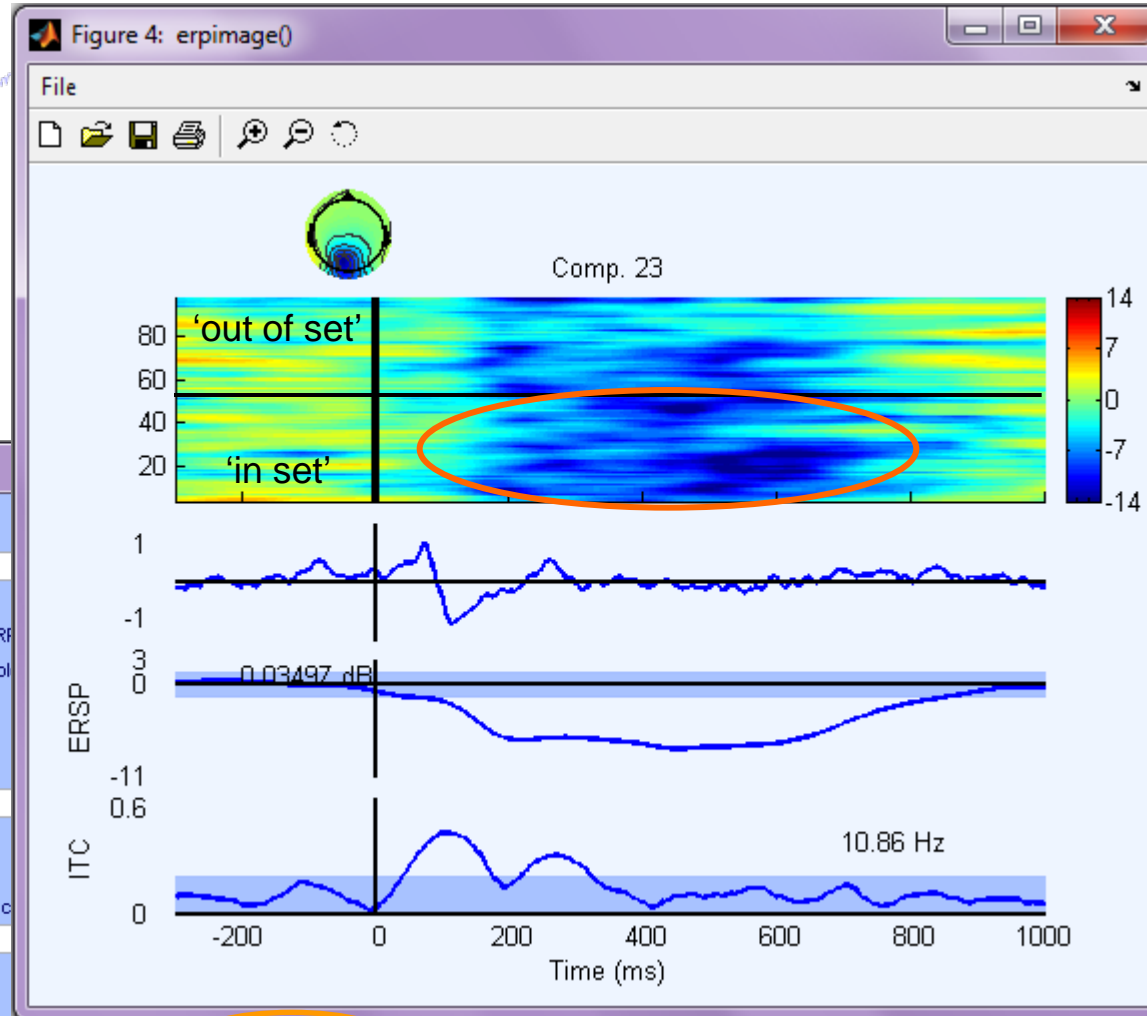
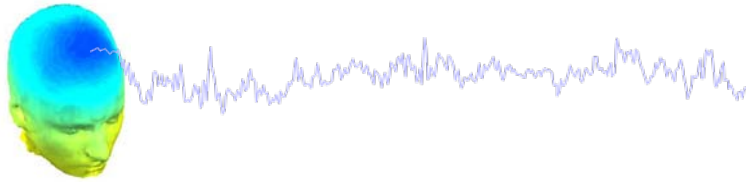
Inter-trial coherence options  
Frequency (Hz | minHz maxHz) Signif. level (<=0.20) Amplitude limits (dB)

Other options  
Plot spectrum (minHz maxHz) Baseline ampl. (dB) Mark times (ms)

Help



# Sort ERP image by response type



Component ERP image -- pop\_erpimage()

Component(s)

Project to channel #

Smoothing

Downsampling

Time limits (ms)

Figure title

Plot scalp map

Plot ERP

Plot colorbar

Sort/align trials by epoch event values

Epoch-sorting field  Event type(s)  Event time range  Rescale

Sort trials by phase

Frequency (Hz | minHz maxHz)  Percent low-amp. trials to ignore  Window c

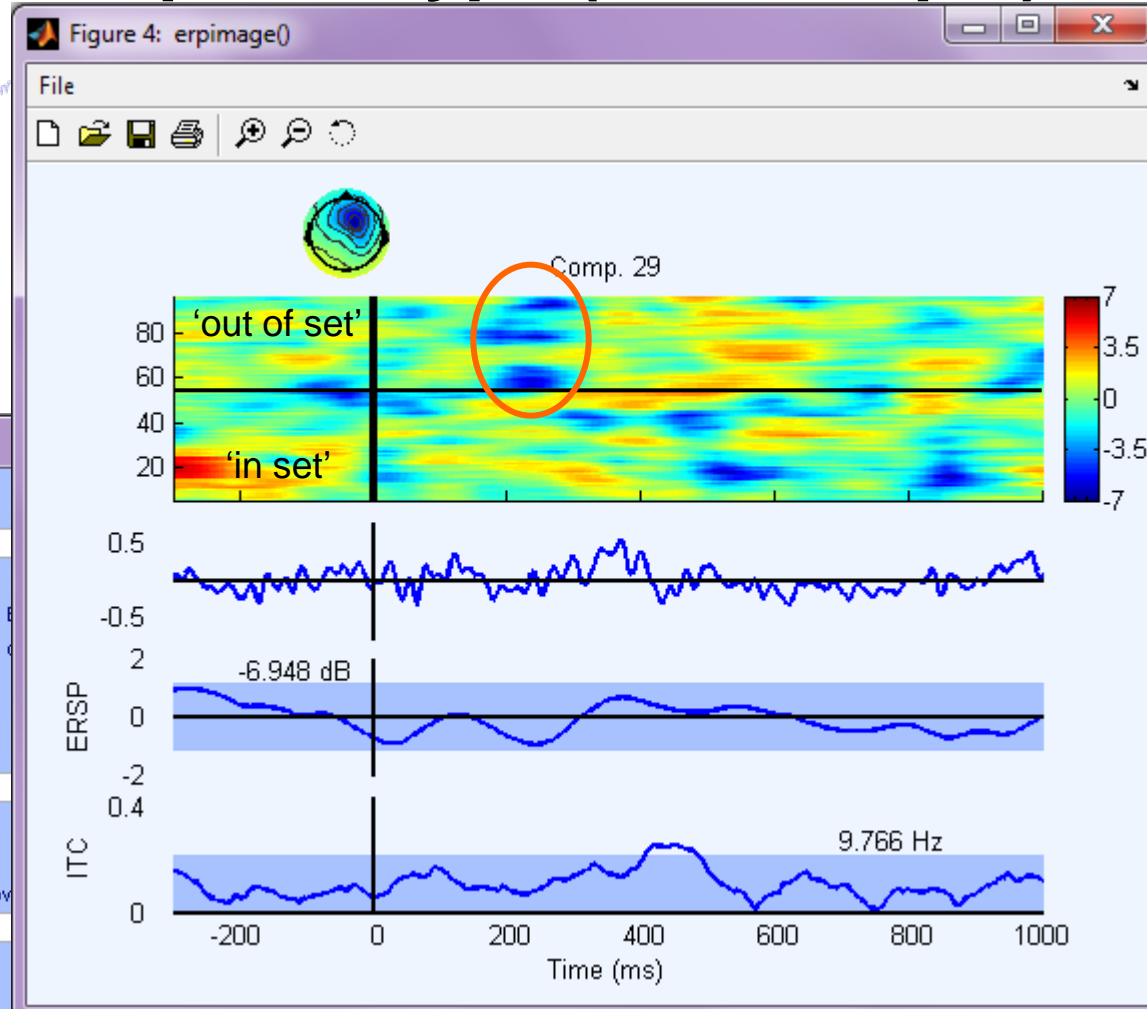
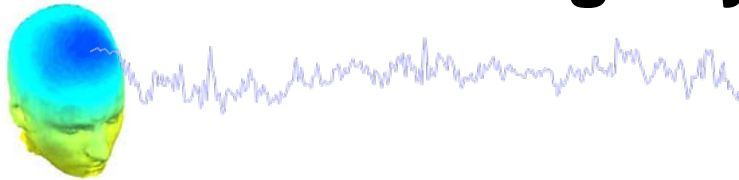
Inter-trial coherence options

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)

# Sort ERP image by response type (2<sup>nd</sup> example)



**Component ERP image -- pop\_erpimage()**

**Component(s)**

**Project to channel #**

**Smoothing**   Plot scalp map

**Downsampling**   Plot ERP

**Time limits (ms)**    Plot colorbar

**Sort/align trials by epoch event values**

Epoch-sorting field:  Event type(s):  Event time range:  Rescale:

**Sort trials by phase**

Frequency (Hz | minHz maxHz):  Percent low-amp. trials to ignore:  Window:

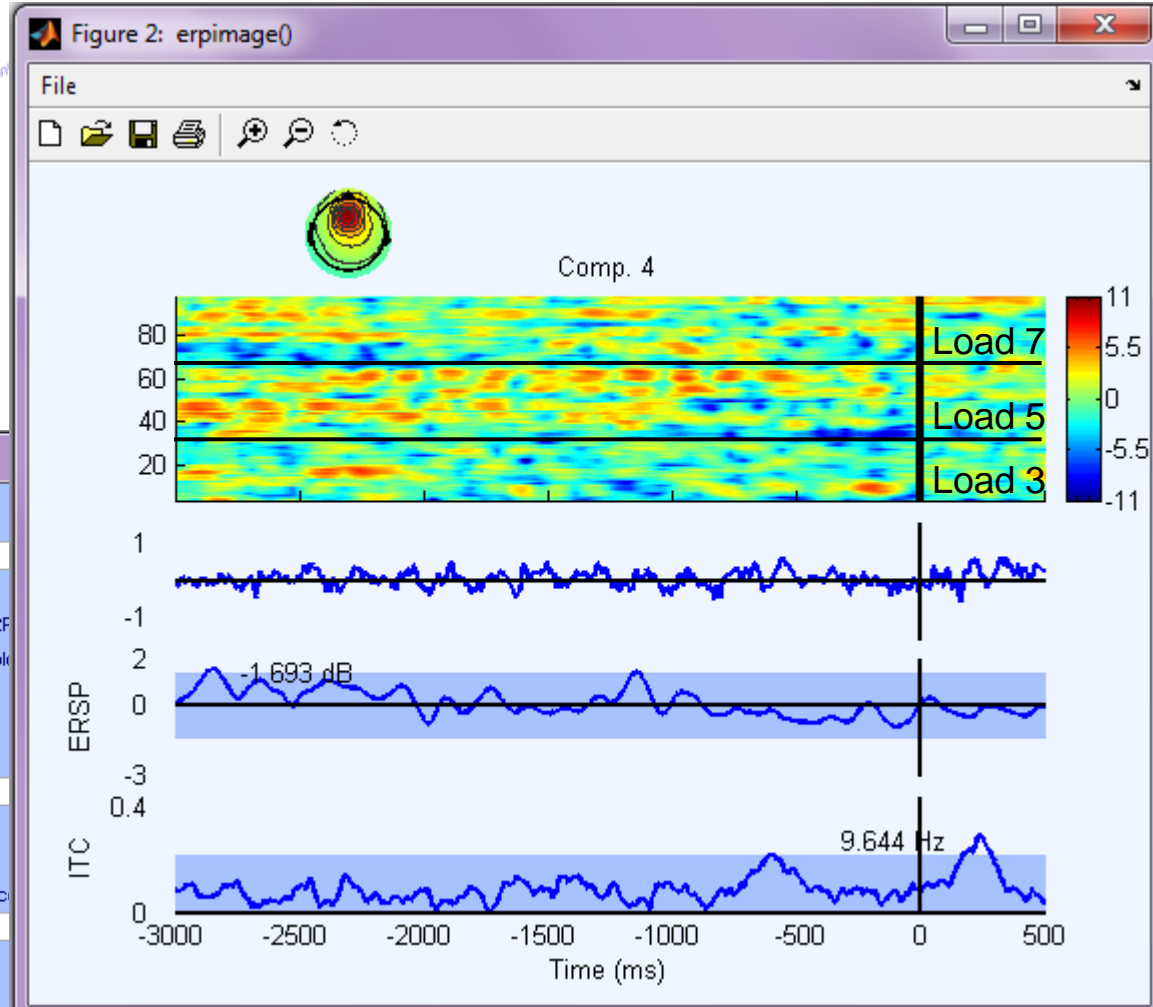
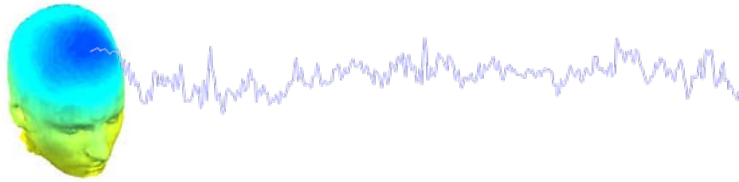
**Inter-trial coherence options**

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)

# Sort ERP image by memory load



Component ERP image -- pop\_erpimage()

Component(s): 4  
Project to channel #:   
Smoothing: 5  
Downsampling: 1  
Time limits (ms): -3000 500

Figure title:   
 Plot scalp map  
 Plot ERP  
 Plot colorbar

Sort/align trials by epoch event values  
Epoch-sorting field: load  
Event type(s): rB' rC' rD' rF' rG' r  
Event time range:   
Rescale: no

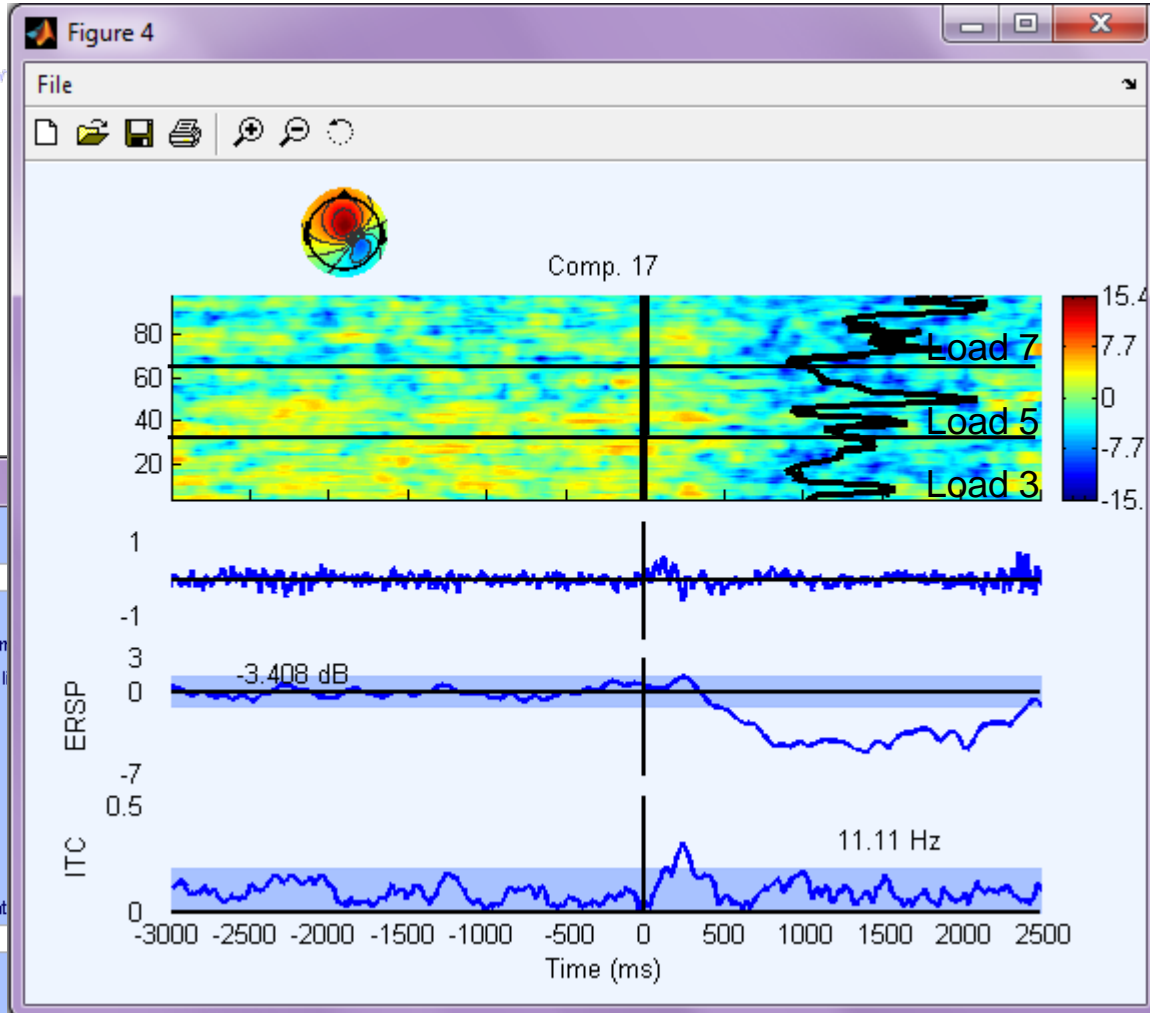
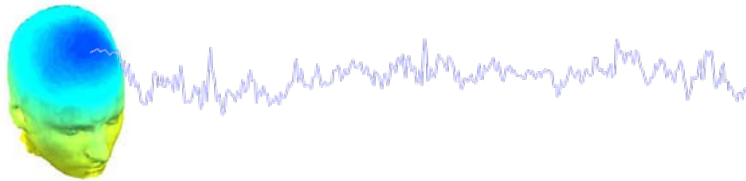
Sort trials by phase  
Frequency (Hz | minHz maxHz):   
Percent low-amp. trials to ignore:   
Window c:

Inter-trial coherence options  
Frequency (Hz | minHz maxHz): 8 12  
Signif. level (<0.20): 0.01  
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

# Sort ERP image by memory load



Component ERP image -- pop\_erpimage()

Component(s)

Project to channel #

Smoothing   Plot scalp map

Downsampling   Plot ERP

Time limits (ms)    Plot colorbar

Figure title

Sort/align trials by epoch event values

Epoch-sorting field  Event type(s)  Event time range  Rescale

Sort trials by phase

Frequency (Hz | minHz maxHz)  Percent low-amp. trials to ignore  Window cent

Inter-trial coherence options

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

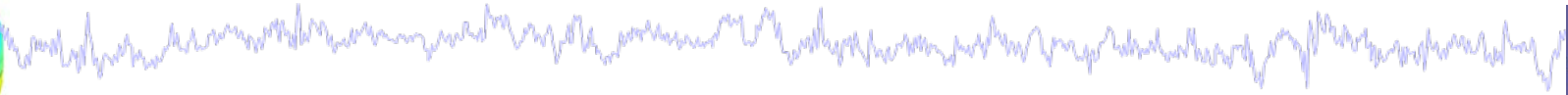
Other options

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

Help Cancel Ok



# Advanced IC analysis



## Task 1

Search EEG.event structure

## Task 2

Use *newtimef()* to compare conditions

## Task 3

Plot a RT-sorted component ERP image

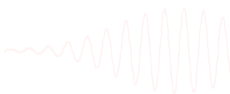
Plot a type-sorted component ERP image

Plot a load-sorted component ERP image

## Task 4

Use outputs from commandline ERP image

**Exercise...**



# ERP image from the commandline



EEGLAB v7.2.7.18b

File Edit Tools **Plot** Study Datasets Help

- #1: Ste
- 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

Component ERP image -- pop\_erpimage()

Component(s)

Project to channel #

Smoothing

Downsampling

Time limits (ms)

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  Align

Don't sort by value

Don't plot values

Sort trials by phase

Frequency (Hz) | minHz maxHz Percent low

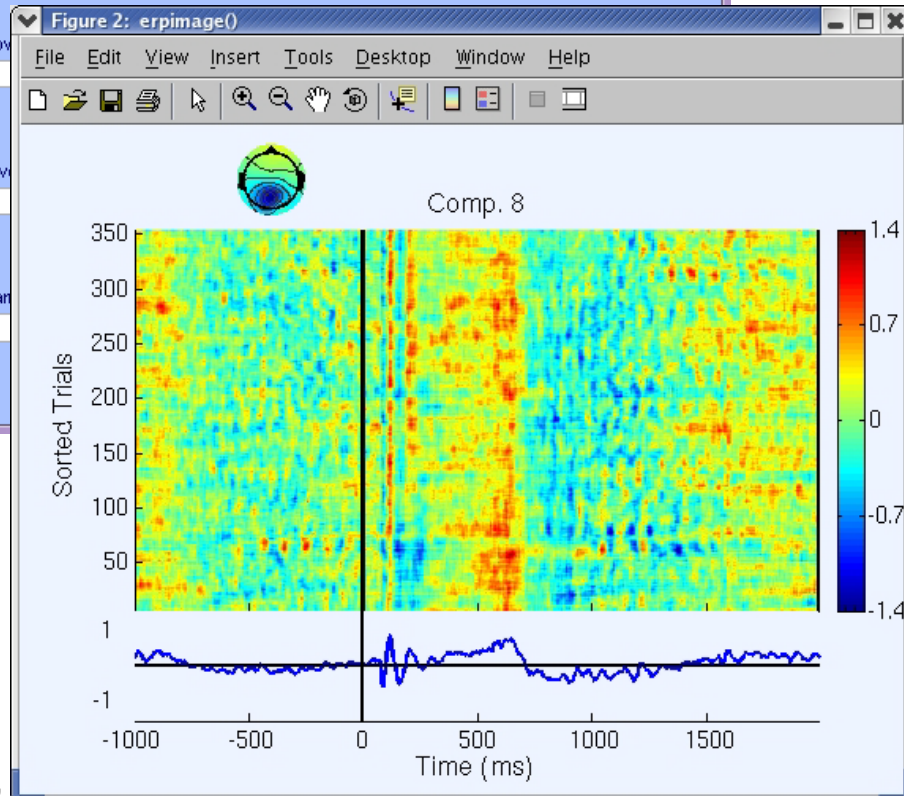
Terminal

File Edit View Terminal Go Help

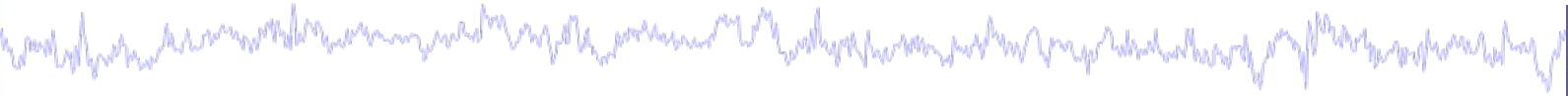
Command executed by pop\_erpimage:

```
erpimage( EEG.icaact([8], :), ones(1, EEG.trials)*EEG.xmax*1000, linspace(EEG.xmin*1000, EEG.xmax*1000, EEG.pnts), 'Comp. 8', 10, 1, 'yerlabel','', 'topo', { EEG.icawinv(:,8) EEG.chanlocs EEG.chaninfo }, 'erp', 'cbar');
```

Plotting input data as 359 epochs of 750 frames sampled at 250.0 Hz.  
Sorting data on input sortvar.  
Smoothing the sorted epochs with a 10-epoch moving window.  
and a decimation factor of 1  
Output data will be 750 frames by 349 smoothed trials.  
Outtrials: 6.00 to 354.00  
The caxis range will be the sym. abs. data range -> [-1.35755,1.35755].  
Data will be plotted between -1000 and 1996 ms.  
Overplotting sorted sortvar on data.  
Plotting the ERP trace below the ERP image  
Plotting a topo map in upper left.  
Done.



# Task 1: Retrieve *erpimage()* call



Command executed by `pop_erpimage`:

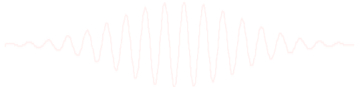
```
erpimage( EEG.icaact([8], :), ones(1, EEG.trials)*EEG.xmax*1000, ...  
          EEG.times, EEG.xmin*1000, EEG.xmax*1000, EEG.pnts), 'Comp. 8', 10, 1, ...  
          'yerplabel', '', 'topo', { EEG.icawinv(:,8) EEG.chanlocs EEG.chaninfo }, ...  
          'erp', 'cbar' );
```

Plotting options/scalp map

Plot ERP and colorbar

To adapt this command to include more *erpimage()* options:

```
>> help erpimage
```



# Use 'help' command to build script



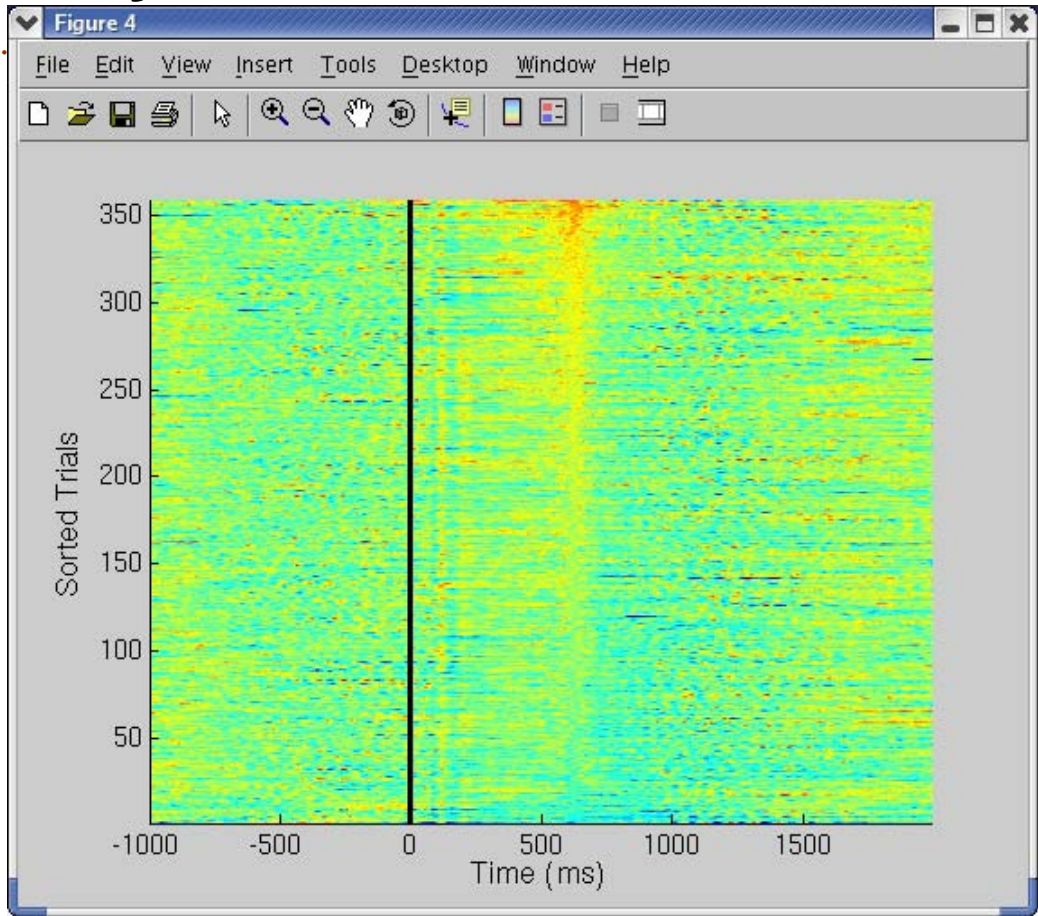
```
Terminal
File Edit View Terminal Go Help
>> help erpimage
erpimage() image a collection of single-trial data epochs, optionally sorted on
and/or aligned to an input sorting variable and smoothed across trials
with a moving-average. (To return event-aligned data without plotting,
use eventlock()). Optionally sort trials on value, amplitude or phase
within a specified latency window. Optionally plot the ERP mean and
std. dev. and moving-window spectral amplitude and inter-trial coherence
at a selected or peak frequency. Click on individual figures parts to
examine them separately and zoom (using axcopy()).
Usage:
>> [outdata,outvar,outtrials,limits,axhndls,erp, ...
      amps,cohers,cohsig,ampsig,outamps,phsangls,phsamp,sortidx,ersig] ...
      = erpimage(data,sortvar,times,'title',avewidth,decimate,...
                 flag1,arg1,flag2,arg2,...);
Necessary inputs:
data [vector or matrix] Single-channel input data to image.
      Formats (1,frames*trials) or (frames,trials)
Optional ordered inputs (with defaults):
sortvar - [vector | []] Variable to sort epochs on (length(sortvar) = nepochs)
          Example: sortvar may be subject response time in each epoch (in ms)
          (default|[]: plot in input order)
times - [vector | []] of latencies (ms) (length(times) = frames)
        Else [startms ntimes srate] Give start latency (ms), time points
        (i.e. frames) per epoch, sampling rate (Hz), (default|[]: 0:nframes-1)
'title' - ['string'] Plot titla (default: none)
avewidth - Number of trials to smooth with a moving-average (may be non-integer)
           (default|0->1)
decimate - Factor to decimate ntrials out by (may be non-integer) (default|0->1)
           If this is large (> sqrt(num. trials)), output this many trials.
Unordered options ('keyword',argument pairs):
```

# ERP image sorted by activation value

```
[outdata,outvar,outtrials,limits,axhndls,...  
erp,amps,cohers,cohsig,ampsig,outamps,...  
phsangls,phsamp,sortidx,ersig] = ...
```

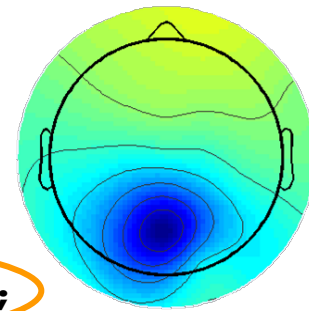
```
erpimage(data,sortvar,times,'title',...  
avewidth,decimate,flag1,arg1,...);
```

```
%%%%% VARIABLES %%%%%%%%%%%%%%%  
comp1 = 8; % Comp number to plot  
data = squeeze(EEG.icaact(comp1,:,:));  
sortvar = []; % no sorting  
startms = 580; % ms  
endms = 620; % ms  
smoothby = 1;  
%%%%% PLOT ERPIMAGE %%%%%%%%%%%%%%%  
figure;
```



```
[outdata,outvar, outtrials,limits, axhndls, erp, ...  
amps, cohers, cohsig, ampsig, outamps, phsangls, ...  
phsamps, sortidx, ersig] ...
```

```
    = erpimage( data, sortvar, EEG.times,  
               '\', smoothby, 1, 'valsort', [startms endms]);
```



# Matlab index definition



```
>> my_numbers = [101,102,103,104,105,106,107,108,109,110];
```

```
my_numbers =
```

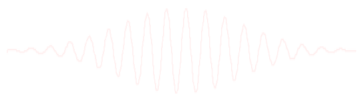
```
101 102 103 104 105 106 107 108 109 110
```

```
>> new_order = [8,2,5,1,10,9,4,6,3,7]; % analogous to sortidx
```

```
>> my_numbers(new_order)
```

```
ans =
```

```
108 102 105 ... 110 109 104 106 103 107
```



# Use sort index to sort a *new* ERP image

Objective: Use sort order (*sortidx*) from 'valsort' of *comp1* to create a new ERP image of another component with the same sort order

```

%%%%% VARIABLES %%%%%%%%%%%
comp1 = 8;
data = squeeze(EEG.icaact(comp1, :, :));
sortvar = [];
startms = 580;
endms = 620;
smoothby = 1;

%%%%% 1st ERPIMAGE %%%%%%%%%%%
figure;
[outdata, outvar, outtrials, limits, axhndls, erp, ...
amps, cohers, cohsig, ampsig, outamps, phsangls, ...
phsamps, sortidx, erpsig] = erpimage(data, sortvar, ...
EEG.times, '', smoothby, 1, 'valsort', [startms endms]);

```

```

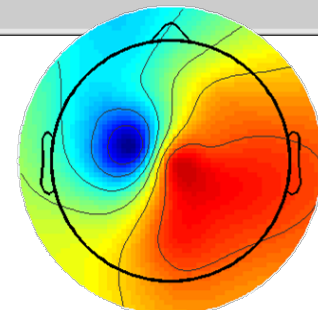
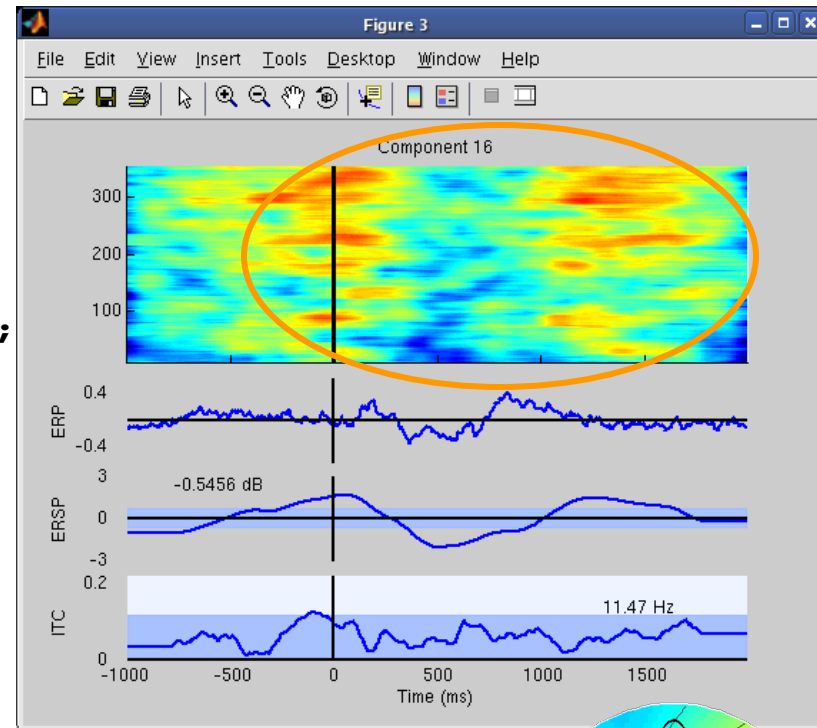
%%%%% 2nd ERPIMAGE %%%%%%%%%%%
%%%%% Sort by previous sort order %%%%%%%%%%%
comp2 = 16;
data2 = squeeze(EEG.icaact(comp2, :, sortidx));
minfrq = 9; % specify freq range for
maxfrq = 12; % amplitude plot
smoothby = 20;

```

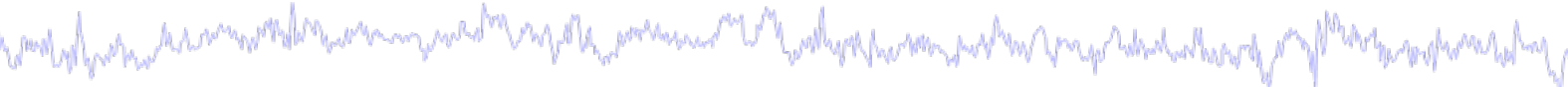
```

figure;
[outdata, outvar, outtrials, limits, axhndls, erp, amps, ...
cohers, cohsig, ampsig, outamps, phsangls, phsamps, sortidx2, erpsig] ...
= erpimage(data2, sortvar, EEG.times, ['Component ', int2str(comp2)], ...
smoothby, 1, 'coher', [minfrq maxfrq .01], 'plotamps');

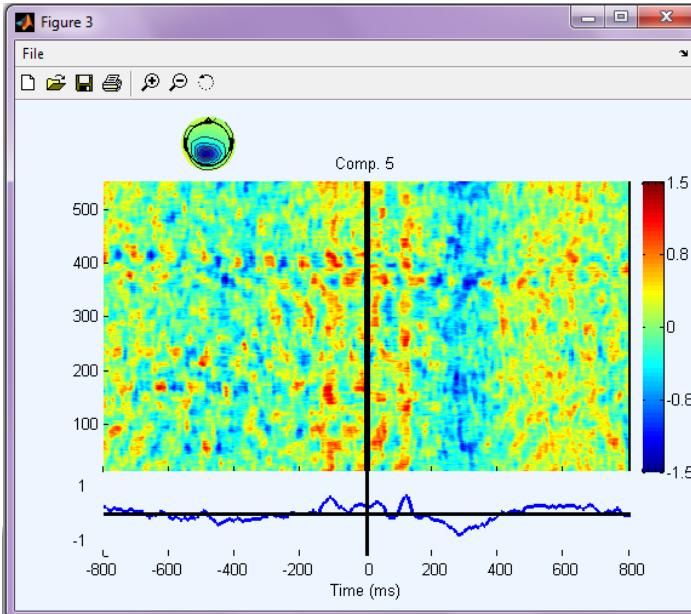
```



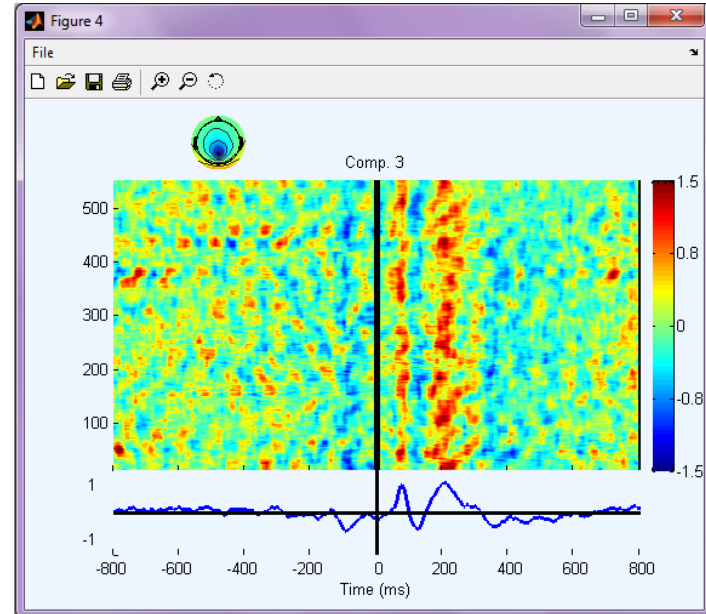
# Phase-sort applied to second IC



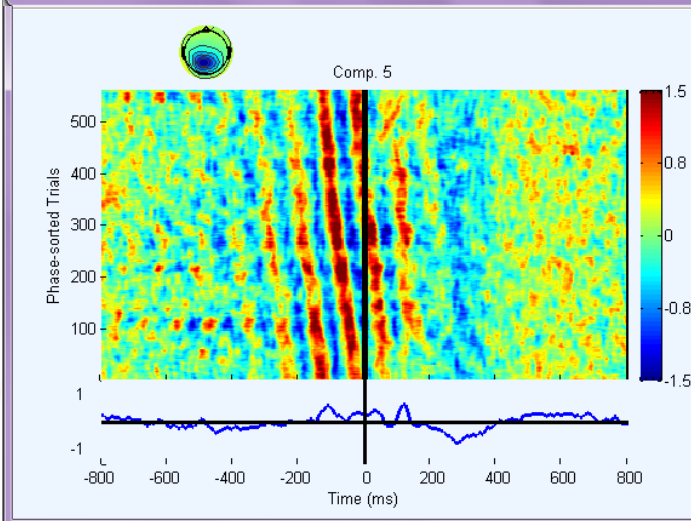
No sort



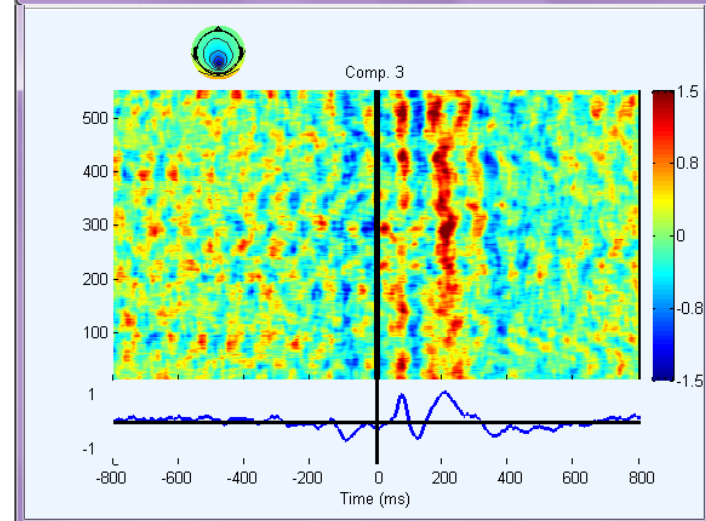
No sort



Phase-sort  
-75 ms  
center

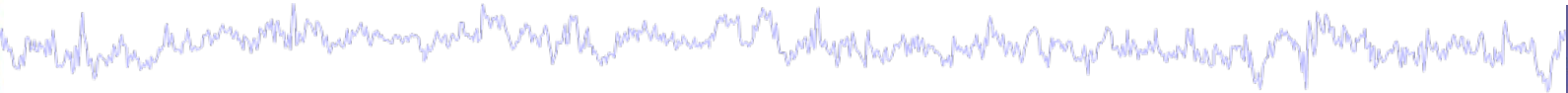


Sorted by  
IC 5  
phase-sort





# Exercise



## • Intermediate:

- Use EEG.event structure to select specific trial types.
  - Epoch on these trials and plot ERSP and/or ERP images
- Plot a two-condition ERSP of a chosen IC (start with loading continuous data, epoching, etc)
- Start with GUI call and adapt a script using 'eegh'
- Collect RTs and include in ERP image plots

## • Advanced:

- From a 'valsort' ERP image, collect **sortidx** output
- Apply sort order to an ERP image of *another* component (try different smoothing factors)

\*\* Example scripts for exercises can be found in  
[.../workshop/Scripts/Tutorial\\_6\\_ICanalysis.m](#)