1- Open EEGLAB

Tip: to figure out how to make scripts in eeglab. First run a command and then type eegh in command line. This will give you the command structure of what you have just run and how the commands are meant to be structured

Preprocessing steps: 2-8 (9)

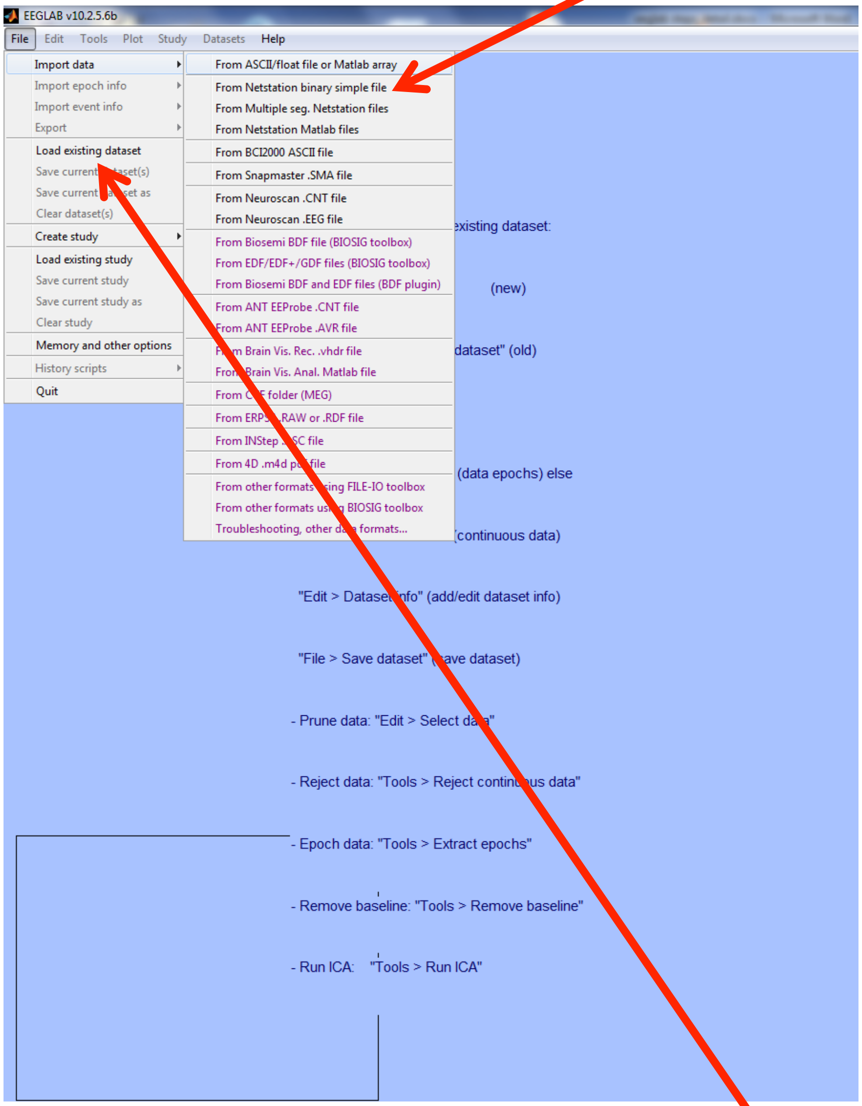
Epoching and ICA: 10-12

Averaging and Stats: 13-16

2- Import data

a. File>Import data >From Netstation binary simple file

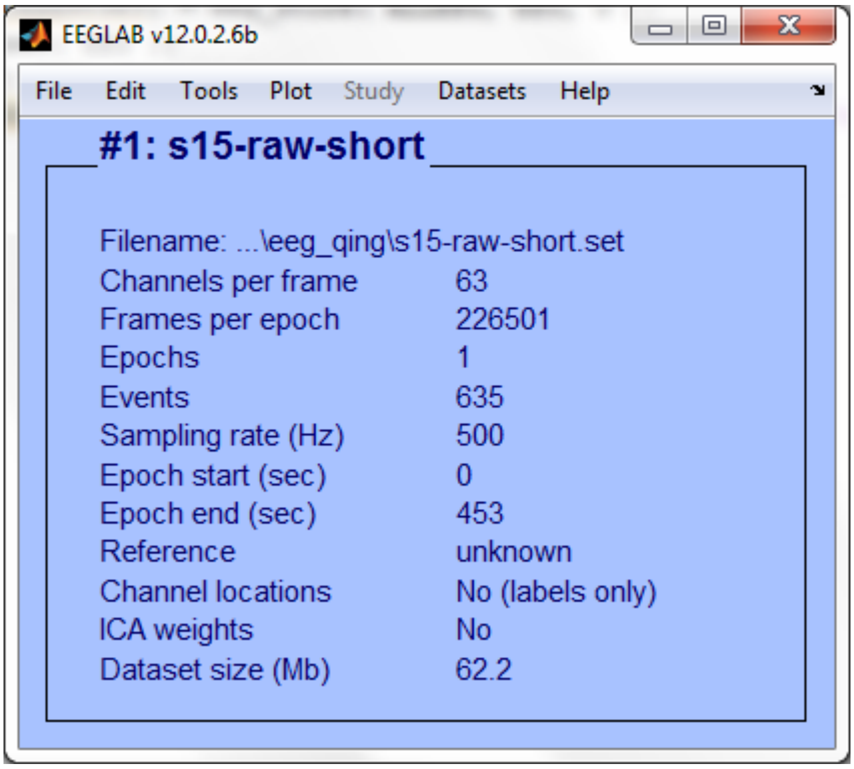
   i. Load corresponding Netstation file



EEGLAB v10.2.5.6b
File
Edit
Tools
Plot
Study
Datasets
Help
Import data
Import epoch info
Import event info
Export
Load existing dataset
Save current dataset(s)
Save current dataset as
Clear dataset(s)
Create study
Load existing study
Save current study
Save current study as
Clear study
Memory and other options
History scripts
Quit
From ASCII/float file or Matlab array
From Netstation binary simple file
From Multiple seg. Netstation files
From Netstation Matlab files
From BCI2000 ASCII file
From Snapmaster .SMA file
From Neuroscan .CNT file
From Neuroscan .EEG file
From Biosemi BDF file (BIOSIG toolbox)
From EDF/EDF+/GDF files (BIOSIG toolbox)
From Biosemi BDF and EDF files (BDF plugin)
From ANT EEProbe .CNT file
From ANT EEProbe .AVR file
From Brain Vis. Rec. .vhdr file
From Brain Vis. Anal. Matlab file
From CTF folder (MEG)
From ERPSS .RAW or .RDF file
From INStep .ASC file
From 4D .m4d pdf file
From other formats using FILE-IO toolbox
From other formats using BIOSIG toolbox
Troubleshooting, other data formats...
existing dataset"
(new)
dataset" (old)
(data epochs) else
(continuous data)
"Edit > Dataset info" (add/edit dataset info)
"File > Save dataset" (save dataset)
- Prune data: "Edit > Select data"
- Reject data: "Tools > Reject continuous data"
- Epoch data: "Tools > Extract epochs"
- Remove baseline: "Tools > Remove baseline"
- Run ICA: "Tools > Run ICA"

Note: to load an existing data set:

b. The information that pops up in EEGlab tells you about the size of the file, number of events, number of channels etc.



#1: s15-raw-short

```
Filename: ...\eeg_qing\s15-raw-short.set
Channels per frame          63
Frames per epoch            226501
Epochs                      1
Events                      635
Sampling rate (Hz)          500
Epoch start (sec)           0
Epoch end (sec)             453
Reference                   unknown
Channel locations           No (labels only)
ICA weights                 No
Dataset size (Mb)           62.2
```

c. Save data so it is easier to have the data stored as a .set file in eeglab – to reopen

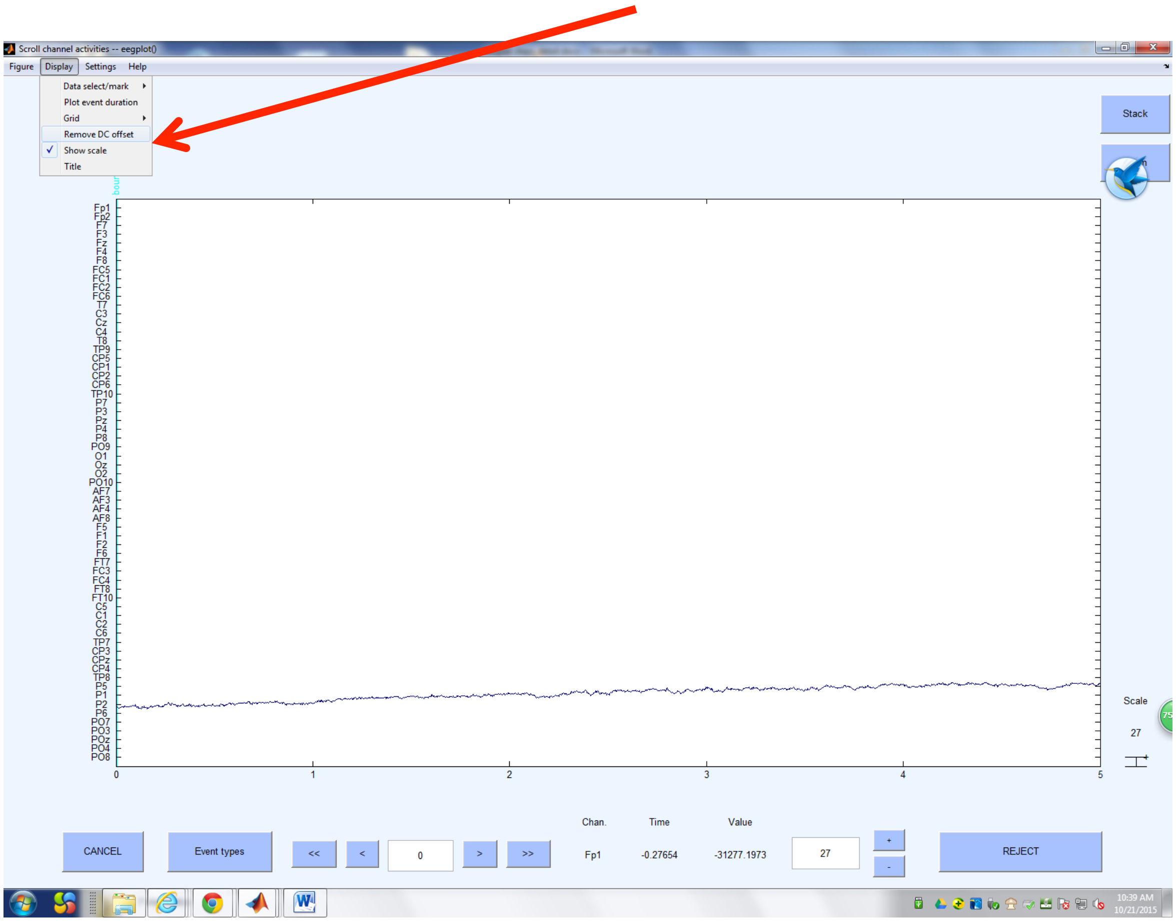
d. SAVE ("ss_expt_name_scroll")

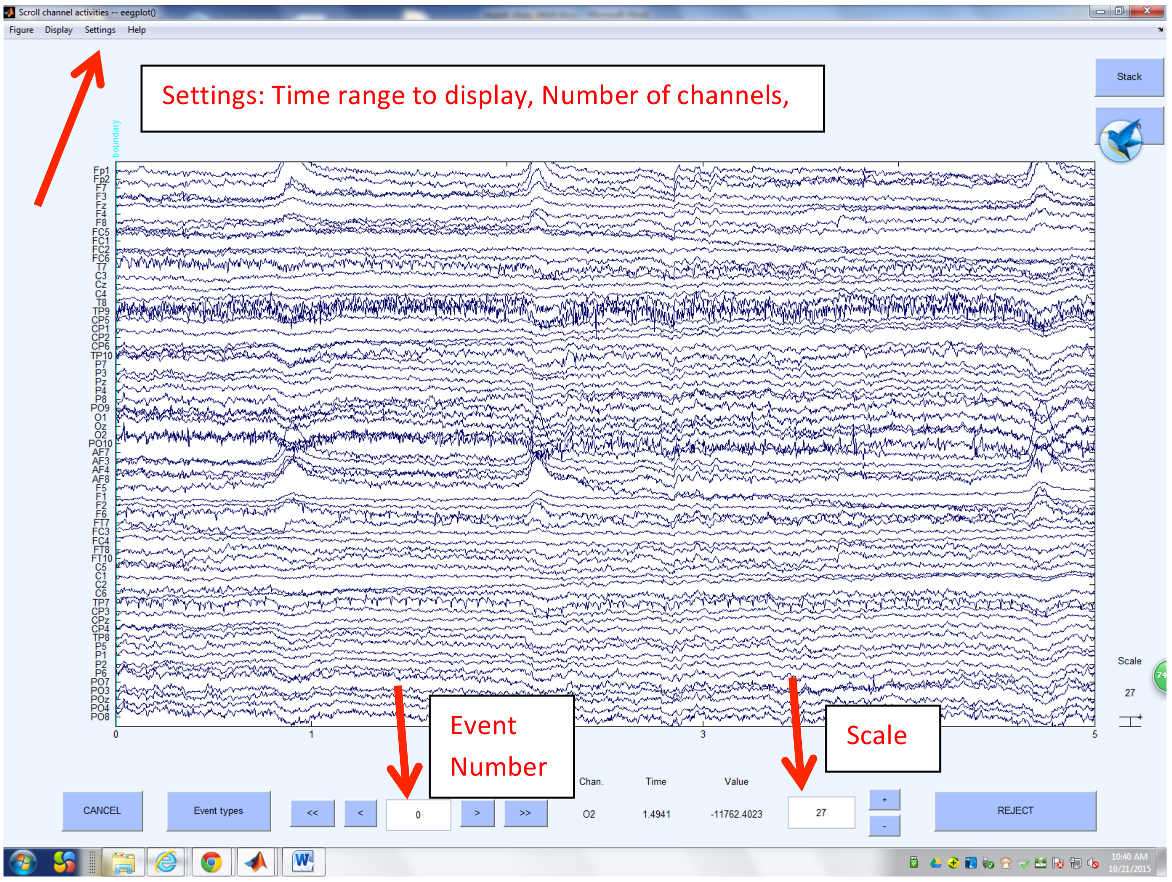
3- Plot and remove artefacts (optional)

a. It is sometimes a good idea to scroll through the entire raw data – to make sure everything went ok during recording. While this can be a bit time consuming, it will give you more familiarity as to whether there are any odd channels that may need interpolating later on, or large artefacts that should be removed from any further analysis.

b. To plot: Plot>Channel data (scroll)

c. Remove DC offset: Display>Remove DC offset




Scroll channel activities -- eegplot()
Figure Display Settings Help
Settings: Time range to display, Number of channels,
Stack
Event Number
Scale
27
CANCEL
Event types
<<
<
0
>
>>
Chan.
Time
Value
O2
1.4941
-11762.4023
27
+
-
REJECT

d. To move through the plot use the arrow keys

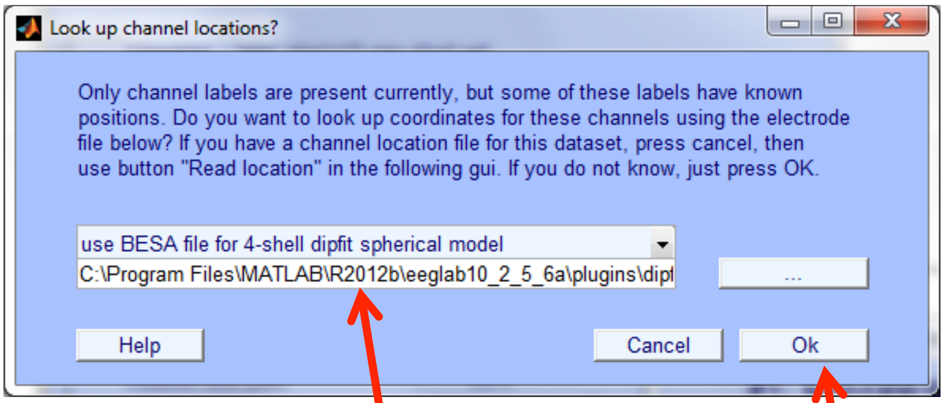
e. SAVE (“ss_expt_name_scroll”)

4- Extract triggers (if applicable)

a. If you want to remove triggers (i.e. remove incorrect trials from analysis), or insert different labels for triggers then it is easier to do this at this early stage.
b. To extract triggers (EEG.event) – then run a program selecting the triggers you want – the important information to keep is the type and latency of the trigger.
c. Add event triggers into file: file>importeventfile>from matlab array: type in "EEG.event" into array
d. SAVE ("ss_expt_name_trig")

5- Import channel location file

a. Need to insert channel information. This is stored in a text document with electrode label and x/y co-ordinates (location) of the electrodes (e.g. Cz is 0,0)
b. Edit> Channel locations



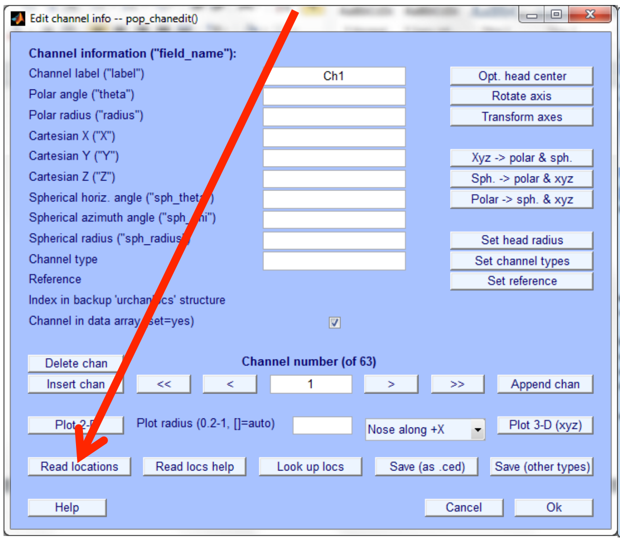
Look up channel locations?

Only channel labels are present currently, but some of these labels have known positions. Do you want to look up coordinates for these channels using the electrode file below? If you have a channel location file for this dataset, press cancel, then use button "Read location" in the following gui. If you do not know, just press OK.

use BESA file for 4-shell dipfit spherical model
C:\Program Files\MATLAB\R2012b\eeglab10_2_5_6a\plugins\dipf

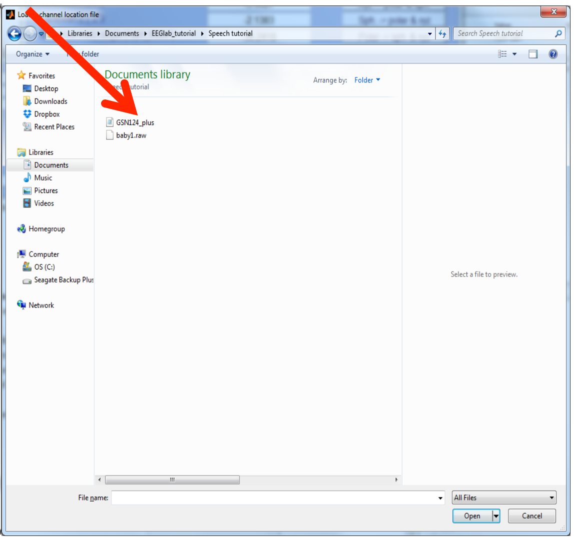
i. Make sure BESA spherical file is selected> Press Ok

c. Read locations> select: GSN124_plus.loc



Edit channel info -- pop_chanedit()

Channel information ("field_name"):
Channel label ("label") Ch1
Polar angle ("theta")
Polar radius ("radius")
Cartesian X ("X")
Cartesian Y ("Y")
Cartesian Z ("Z")
Spherical horiz. angle ("sph_theta")
Spherical azimuth angle ("sph_phi")
Spherical radius ("sph_radius")
Channel type
Reference
Index in backup 'urchanlocs' structure
Channel in data array (set=yes) ✓
Channel number (of 63) 1
Plot radius (0.2-1, []=auto)
Nose along +X



Load channel location file

Libraries ▸ Documents ▸ EEGlab_tutorial ▸ Speech tutorial

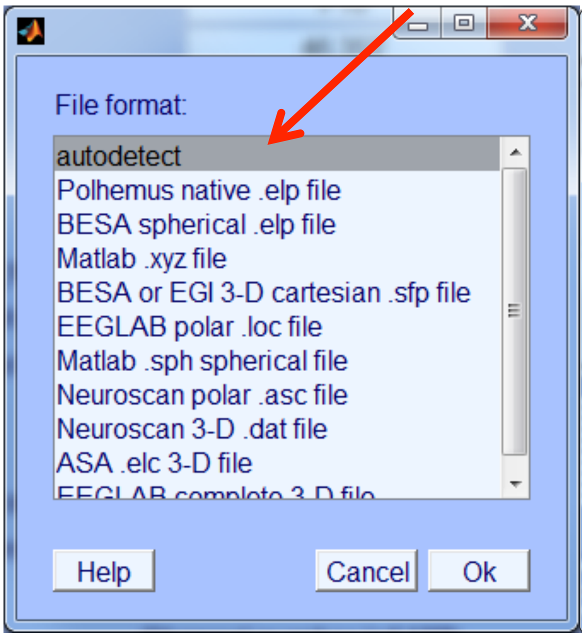
Documents library
Speech tutorial
Arrange by: Folder

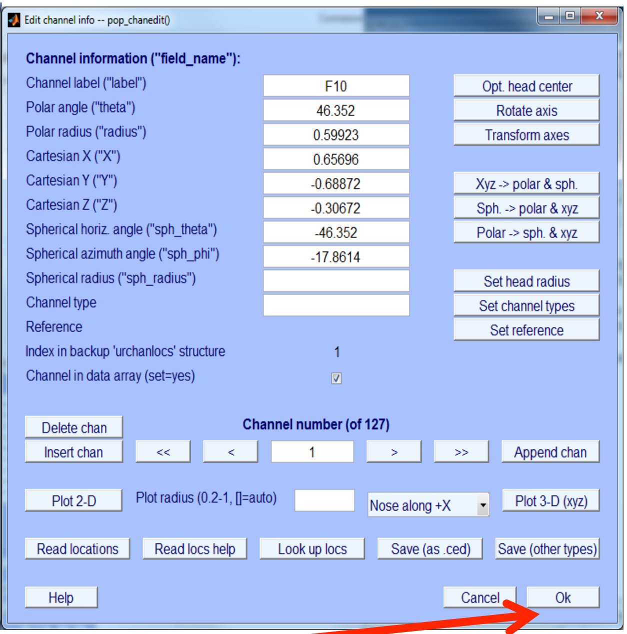
GSN124_plus
baby1.raw

Select a file to preview.

File name:
All Files

i. Select: autodetect> Press OK




Edit channel info -- pop_chanedit()

Channel information ("field_name"):

Channel label ("label")	F10
Polar angle ("theta")	46.352
Polar radius ("radius")	0.59923
Cartesian X ("X")	0.65696
Cartesian Y ("Y")	-0.68872
Cartesian Z ("Z")	-0.30672
Spherical horiz. angle ("sph_theta")	-46.352
Spherical azimuth angle ("sph_phi")	-17.8614
Spherical radius ("sph_radius")	
Channel type	
Reference	
Index in backup 'urchanlocs' structure	1
Channel in data array (set=yes)	✓

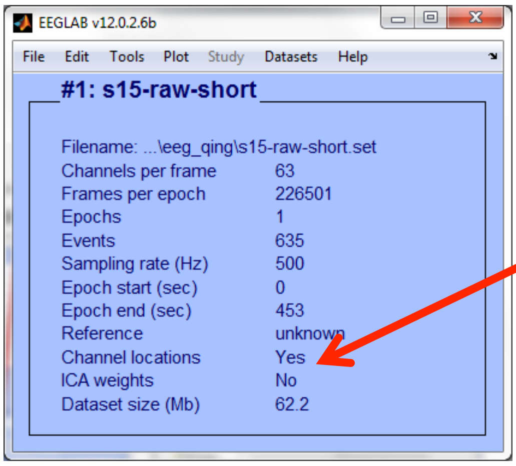
Channel number (of 127): 1

Plot radius (0.2-1, []=auto)

Nose along +X

d. Press Ok on Edit chan info window

e. In data set file – Channel Locations should now have changed to a "yes"


EEGLAB v12.0.2.6b

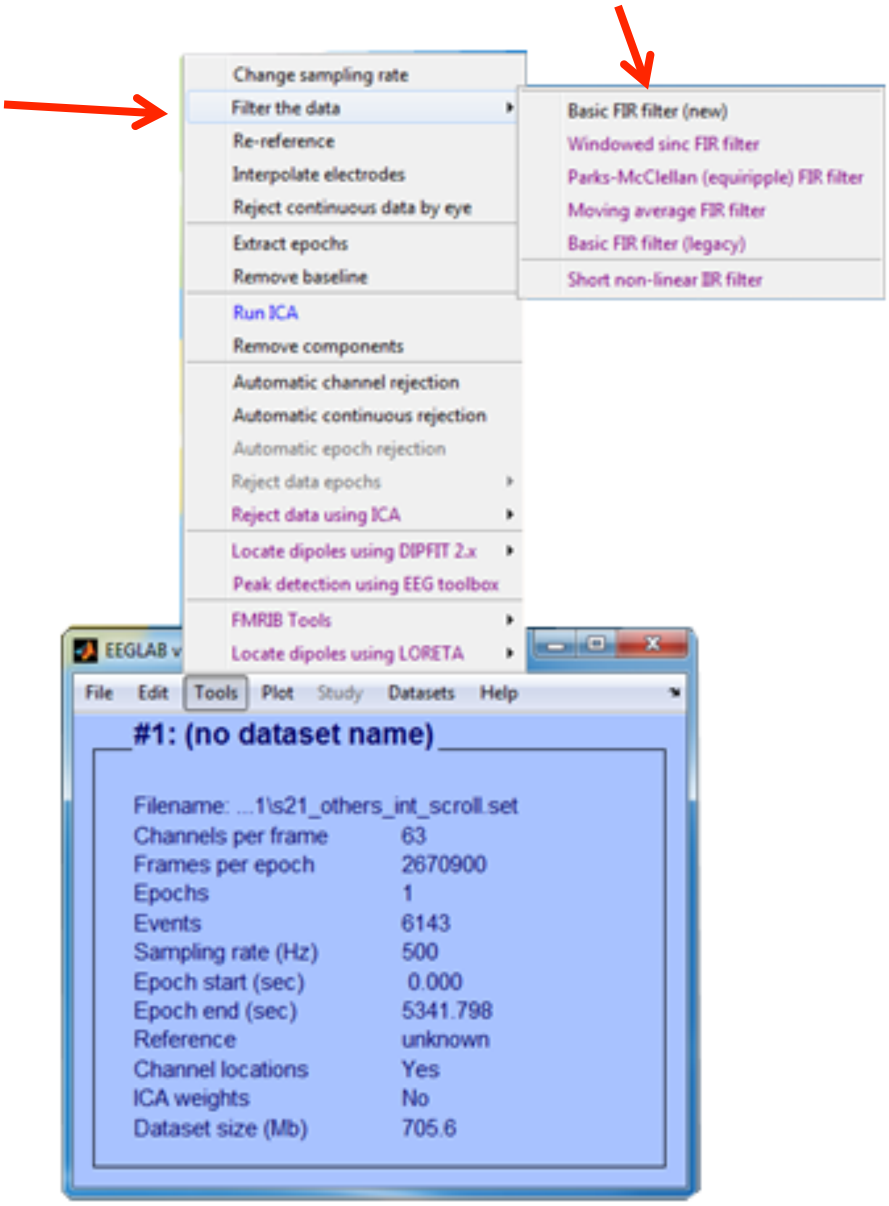
#1: s15-raw-short

Filename: ...\eeg_qing\s15-raw-short.set	
Channels per frame	63
Frames per epoch	226501
Epochs	1
Events	635
Sampling rate (Hz)	500
Epoch start (sec)	0
Epoch end (sec)	453
Reference	unknown
Channel locations	Yes
ICA weights	No
Dataset size (Mb)	62.2

f. SAVE ("ss_expt_name_chan")

6- Filters (use FIR)

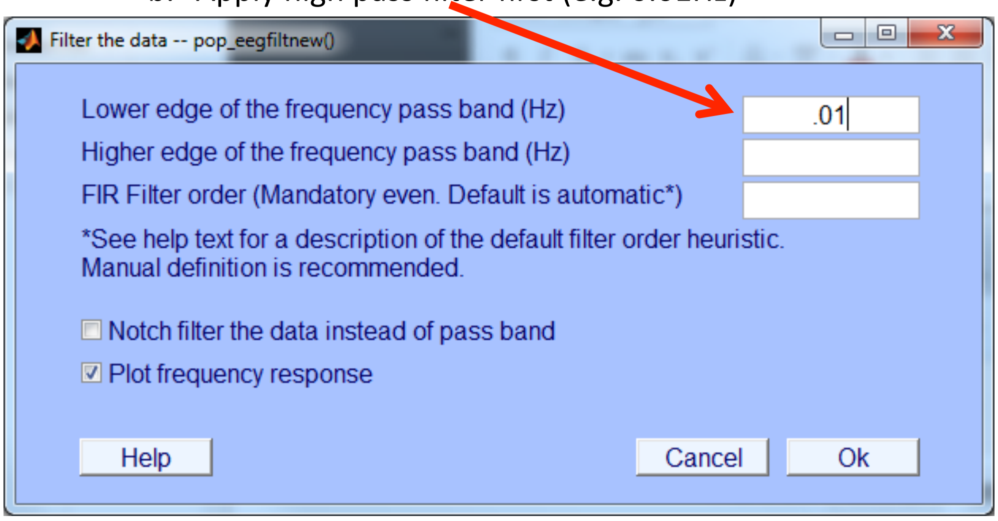
a. Tools>Filter the data > Basic (FIR)


Change sampling rate
Filter the data
Re-reference
Interpolate electrodes
Reject continuous data by eye
Extract epochs
Remove baseline
Run ICA
Remove components
Automatic channel rejection
Automatic continuous rejection
Automatic epoch rejection
Reject data epochs
Reject data using ICA
Locate dipoles using DIPFIT 2.x
Peak detection using EEG toolbox
FMRIB Tools
Locate dipoles using LORETA
Basic FIR filter (new)
Windowed sinc FIR filter
Parks-McClellan (equiripple) FIR filter
Moving average FIR filter
Basic FIR filter (legacy)
Short non-linear IIR filter
EEGLAB v
File
Edit
Tools
Plot
Study
Datasets
Help

#1: (no dataset name)

Filename: ...1\s21_others_int_scroll.set	
Channels per frame	63
Frames per epoch	2670900
Epochs	1
Events	6143
Sampling rate (Hz)	500
Epoch start (sec)	0.000
Epoch end (sec)	5341.798
Reference	unknown
Channel locations	Yes
ICA weights	No
Dataset size (Mb)	705.6

b. Apply high pass filter first (e.g. 0.01Hz)

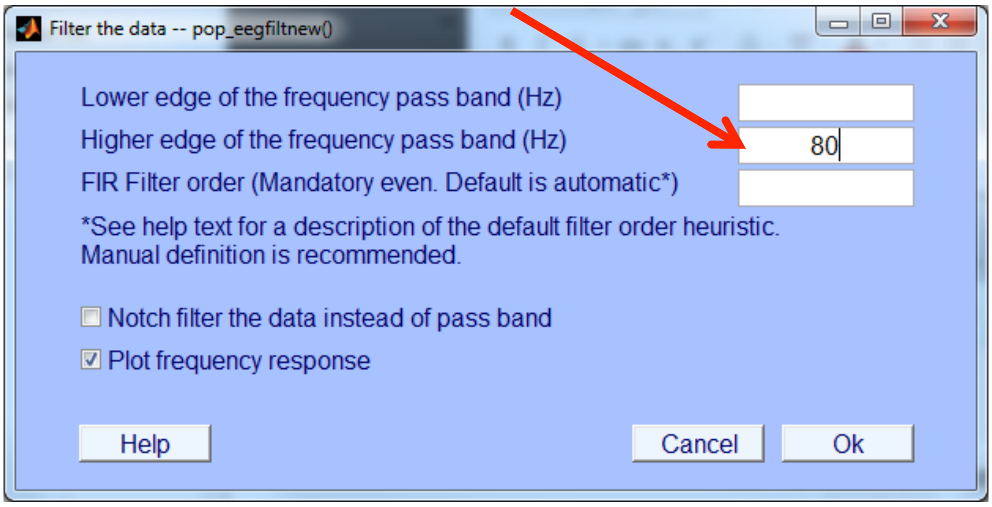


Lower edge of the frequency pass band (Hz) .01
Higher edge of the frequency pass band (Hz)
FIR Filter order (Mandatory even. Default is automatic*)
*See help text for a description of the default filter order heuristic.
Manual definition is recommended.

☐ Notch filter the data instead of pass band
☑ Plot frequency response

c. Once the data has been high-pass filtered

d. Apply low pass filter next (e.g. 80Hz)



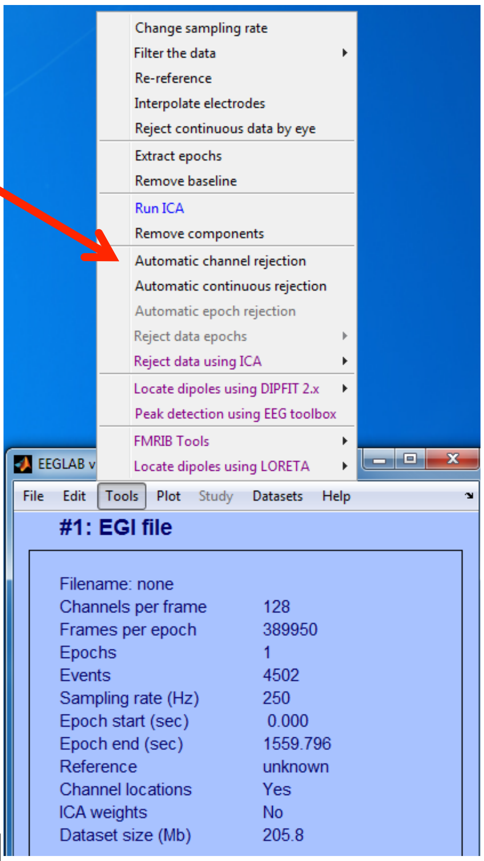
Lower edge of the frequency pass band (Hz)
Higher edge of the frequency pass band (Hz) 80
FIR Filter order (Mandatory even. Default is automatic*)
*See help text for a description of the default filter order heuristic.
Manual definition is recommended.
☐ Notch filter the data instead of pass band
☑ Plot frequency response

e. Note: filtering can distort data or clean it up, Luck has a great chapter on how and when to use filters

f. SAVE ("ss_expt_name_filt")

7- Identify Bad Channels

a. Tools>Automatic channel rejection



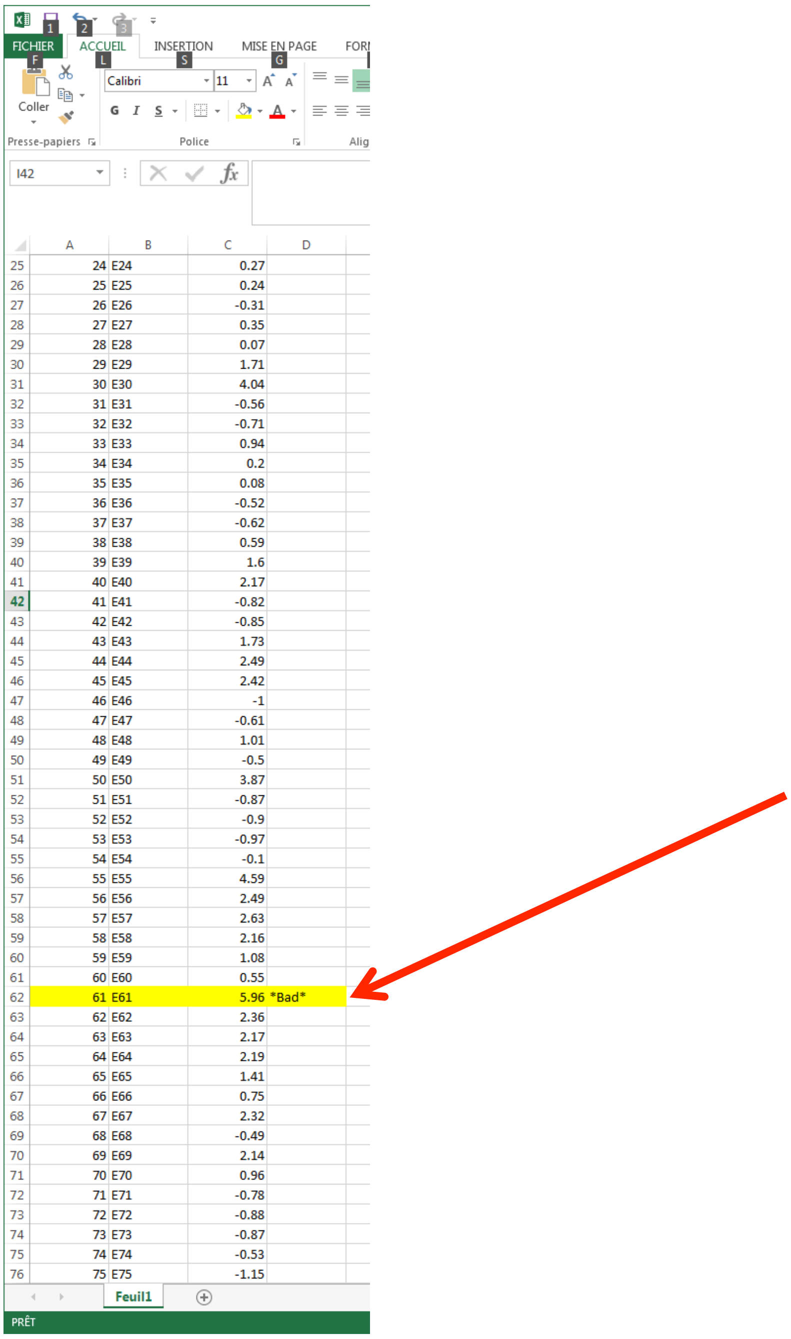
#1: EGI file

```
Filename: none
Channels per frame       128
Frames per epoch         389950
Epochs                   1
Events                   4502
Sampling rate (Hz)       250
Epoch start (sec)         0.000
Epoch end (sec)          1559.796
Reference                unknown
Channel locations        Yes
ICA weights              No
Dataset size (Mb)        205.8
```

]

b. The Output of this can be copied into an Excel file and saved.

c. Highlight any channel labelled as bad and take note of the electrode number



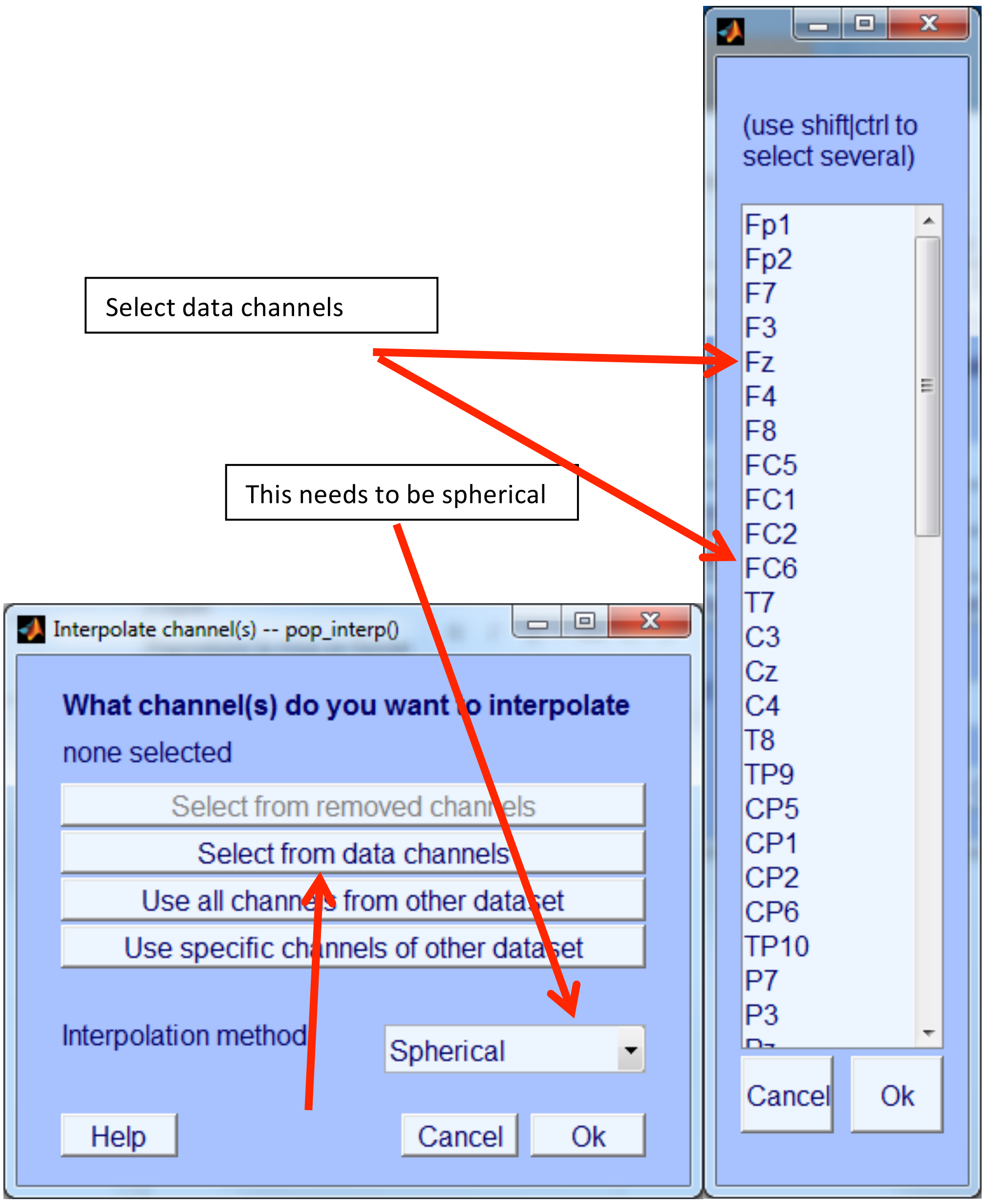
	A	B	C	D
25	24	E24	0.27	
26	25	E25	0.24	
27	26	E26	-0.31	
28	27	E27	0.35	
29	28	E28	0.07	
30	29	E29	1.71	
31	30	E30	4.04	
32	31	E31	-0.56	
33	32	E32	-0.71	
34	33	E33	0.94	
35	34	E34	0.2	
36	35	E35	0.08	
37	36	E36	-0.52	
38	37	E37	-0.62	
39	38	E38	0.59	
40	39	E39	1.6	
41	40	E40	2.17	
42	41	E41	-0.82	
43	42	E42	-0.85	
44	43	E43	1.73	
45	44	E44	2.49	
46	45	E45	2.42	
47	46	E46	-1	
48	47	E47	-0.61	
49	48	E48	1.01	
50	49	E49	-0.5	
51	50	E50	3.87	
52	51	E51	-0.87	
53	52	E52	-0.9	
54	53	E53	-0.97	
55	54	E54	-0.1	
56	55	E55	4.59	
57	56	E56	2.49	
58	57	E57	2.63	
59	58	E58	2.16	
60	59	E59	1.08	
61	60	E60	0.55	
62	61	E61	5.96	*Bad*
63	62	E62	2.36	
64	63	E63	2.17	
65	64	E64	2.19	
66	65	E65	1.41	
67	66	E66	0.75	
68	67	E67	2.32	
69	68	E68	-0.49	
70	69	E69	2.14	
71	70	E70	0.96	
72	71	E71	-0.78	
73	72	E72	-0.88	
74	73	E73	-0.87	
75	74	E74	-0.53	
76	75	E75	-1.15	

d. OR: Edit>Select Data>Channel Data (I AM UP TO HERE!!)

8- Interpolation (as a preprocessing step - do this once you have run through 1 to 12 once) Note: you can also interpolate after epoch and ica'ing but will have to re-reference after ICA.

a. When channels are bad/noisy/flat and are contributing to epoch rejection then you need to interpolate the electrode. Interpolation takes an average of the signal from surrounding electrodes so that the bad signal becomes an interpolation of the signal from the surrounding electrodes

b. If this is done before the ICA (i.e. before epoch extraction) the data entered into the ICA is clean
c. Tools>Interpolate electrodes
d. Select from data channels
   i. pick channels you want to interpolate


Select data channels

This needs to be spherical

Interpolate channel(s) -- pop_interp()

What channel(s) do you want to interpolate

none selected

Interpolation method: Spherical

(use shift|ctrl to select several)

Fp1, Fp2, F7, F3, Fz, F4, F8, FC5, FC1, FC2, FC6, T7, C3, Cz, C4, T8, TP9, CP5, CP1, CP2, CP6, TP10, P7, P3

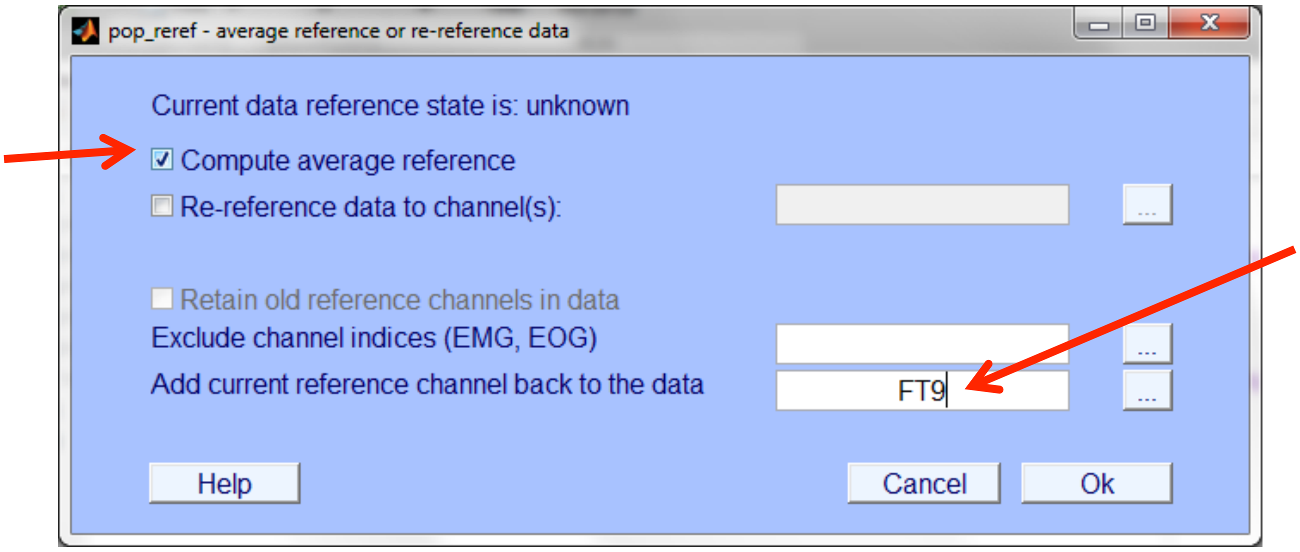
Select from removed channels
Select from data channels
Use all channels from other dataset
Use specific channels of other dataset
Help
Cancel
Ok
Cancel
Ok

e. Interpolation method: spherical method
f. SAVE ("ss_expt_name_interp")

9- Rereference

a. Tools>Re-reference

b. For an average reference:
   i. Click “compute average reference”
   ii. In box for “add current reference channel back to the data” type in your reference channel (e.g. Cz)


pop_reref - average reference or re-reference data

Current data reference state is: unknown

[✓] Compute average reference

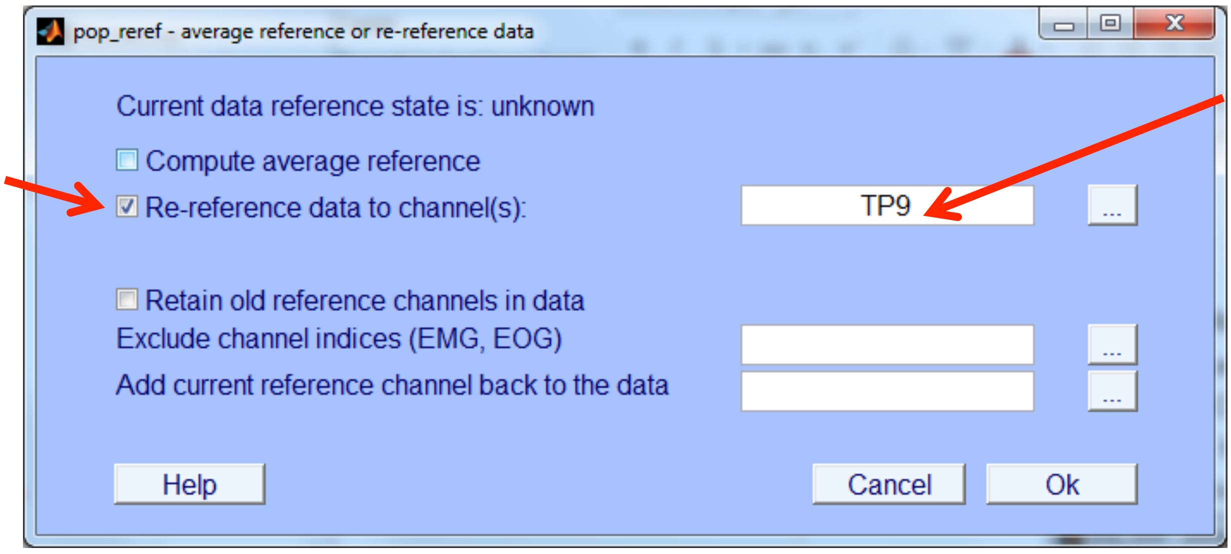
[ ] Re-reference data to channel(s):

[ ] Retain old reference channels in data

Exclude channel indices (EMG, EOG)

Add current reference channel back to the data: FT9

c. For a channel reference (e.g mastoids: TP9 and TP10):
   i. Click “re-reference data to channel” and select channels you want to reference to
   ii. In box for “add current reference channel back to the data” type in your reference channel (e.g. Cz)


pop_reref - average reference or re-reference data

Current data reference state is: unknown

[ ] Compute average reference

[✓] Re-reference data to channel(s): TP9

[ ] Retain old reference channels in data

Exclude channel indices (EMG, EOG)

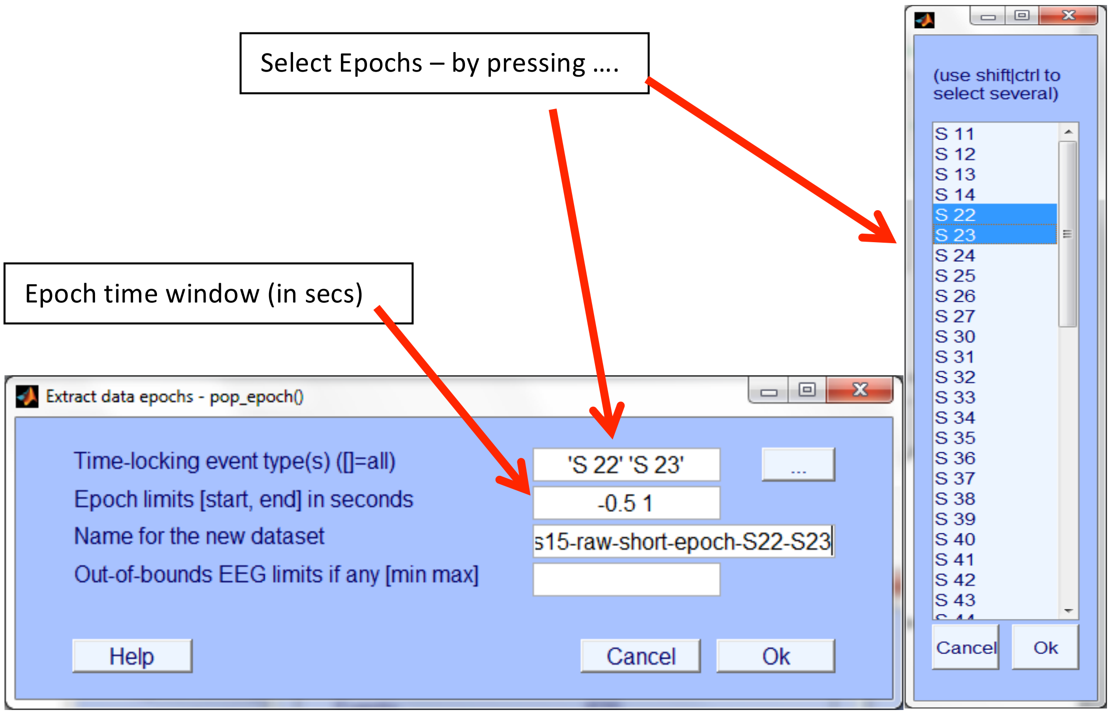
Add current reference channel back to the data

d. SAVE (“ss_expt_name_reref”)

10- Extract data epochs

a. Once pre-processed data (Steps 1-9), use the final pre-processed data (“ss_seq_omit_interp”) to extract data epochs. Need to decide the triggers (time locking events), the size of the epoch window which includes pre-stimulus and post-stimulus (in seconds e.g. -.5 to 1), the baseline (in milliseconds e.g. -100 to 0)

b. Tools>Extract Epochs



Select Epochs – by pressing ….

Epoch time window (in secs)

Extract data epochs - pop_epoch()

Time-locking event type(s) ([]=all)	'S 22' 'S 23'
Epoch limits [start, end] in seconds	-0.5 1
Name for the new dataset	s15-raw-short-epoch-S22-S23
Out-of-bounds EEG limits if any [min max]	

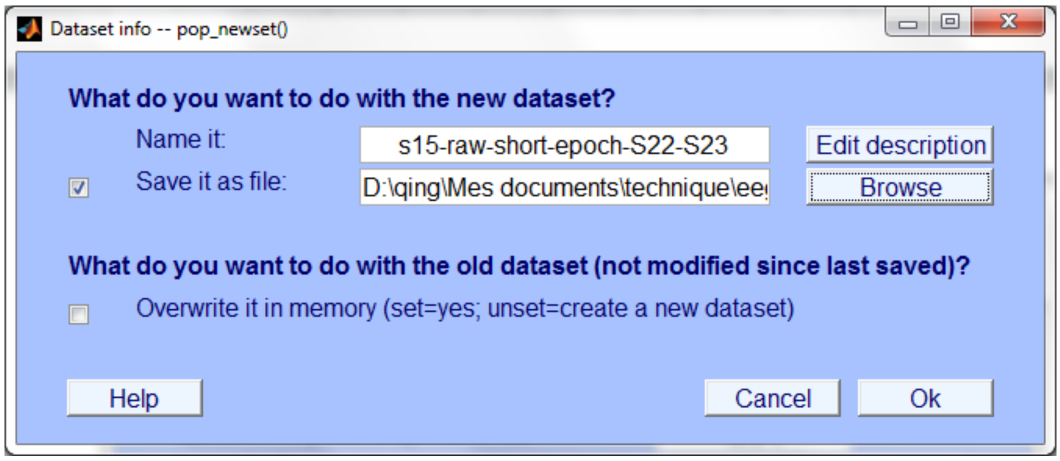
(use shift\|ctrl to select several)

S 11, S 12, S 13, S 14, S 22 (selected), S 23 (selected), S 24, S 25, S 26, S 27, S 30, S 31, S 32, S 33, S 34, S 35, S 36, S 37, S 38, S 39, S 40, S 41, S 42, S 43

i. Time-locking event: select triggers

ii. Epoch limits: epoch time window [start time, end time]

iii. Save dataset ("ss_expt_name_epoch_name")



Dataset info -- pop_newset()

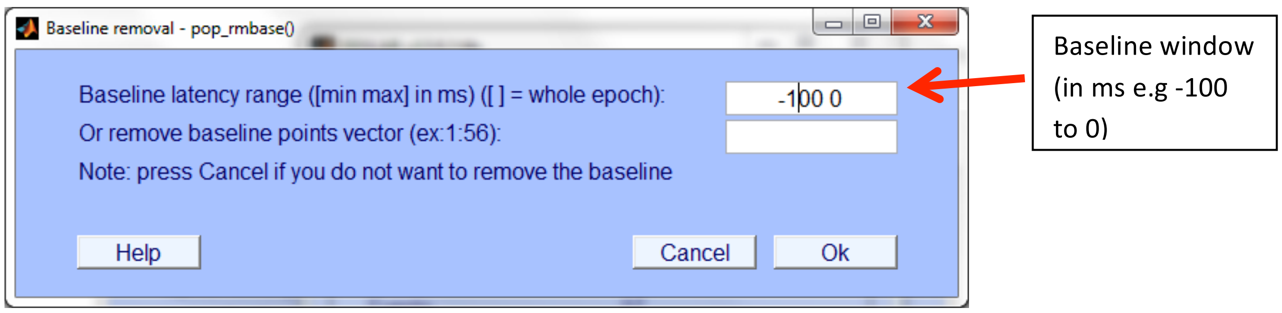
What do you want to do with the new dataset?

Name it:	s15-raw-short-epoch-S22-S23
[✓] Save it as file:	D:\qing\Mes documents\technique\ee

What do you want to do with the old dataset (not modified since last saved)?

[ ] Overwrite it in memory (set=yes; unset=create a new dataset)

iv. Popup: baseline removal



Baseline removal - pop_rmbase()

Baseline latency range ([min max] in ms) ([ ] = whole epoch):	-100 0
Or remove baseline points vector (ex:1:56):	

Note: press Cancel if you do not want to remove the baseline

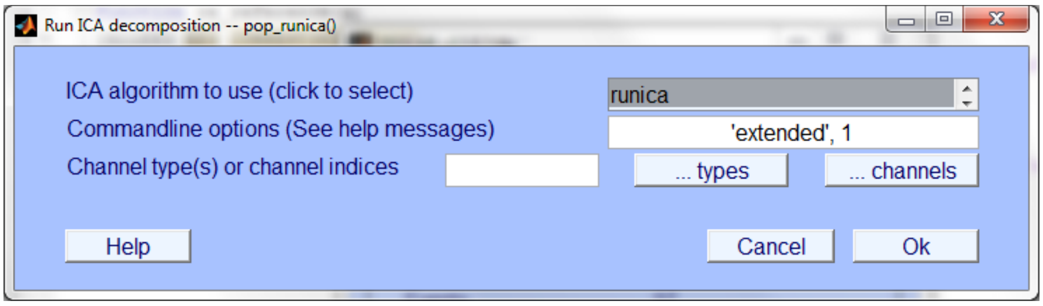
Baseline window (in ms e.g -100 to 0)

1. Baseline latency range [start time for baseline, end time for baseline] (note in ms)>Ok
2. SAVE: File> save current data set ("ss_expt_name_epoch_name_base")

c. MAKE SURE YOU SAVE THE EPOCHED DATA – and that you remove the baseline – as this will really mess up your trials marked for rejections.
d. If you are looking at different epochs e.g. pre-action activity vs stimulus locked ERP then need to create different epochs for each.

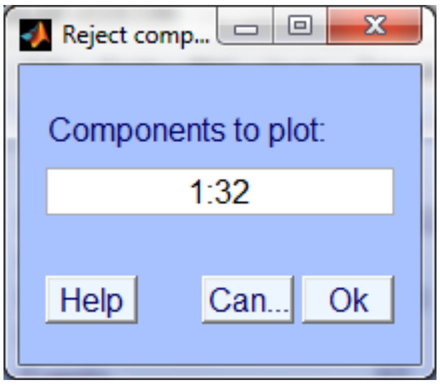
11- Run ICA for blink correction

a. In order to remove blink artefacts an ICA is run on the <u>epoched</u> data
b. Tools>Run ICA
    i. This will take a while – it can be from an hour to a few hours depending on the size of your epoched data

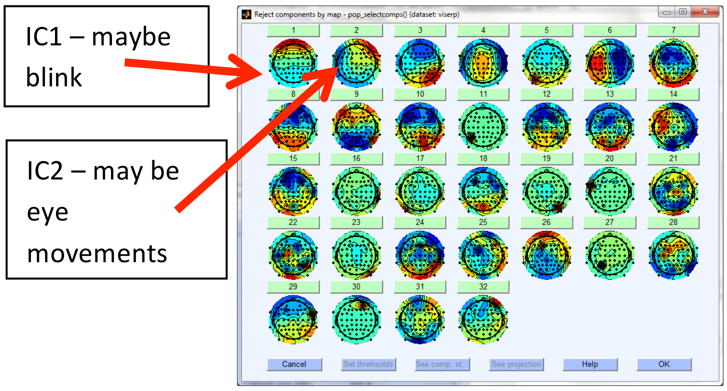

Run ICA decomposition -- pop_runica()

ICA algorithm to use (click to select) runica
Commandline options (See help messages) 'extended', 1
Channel type(s) or channel indices

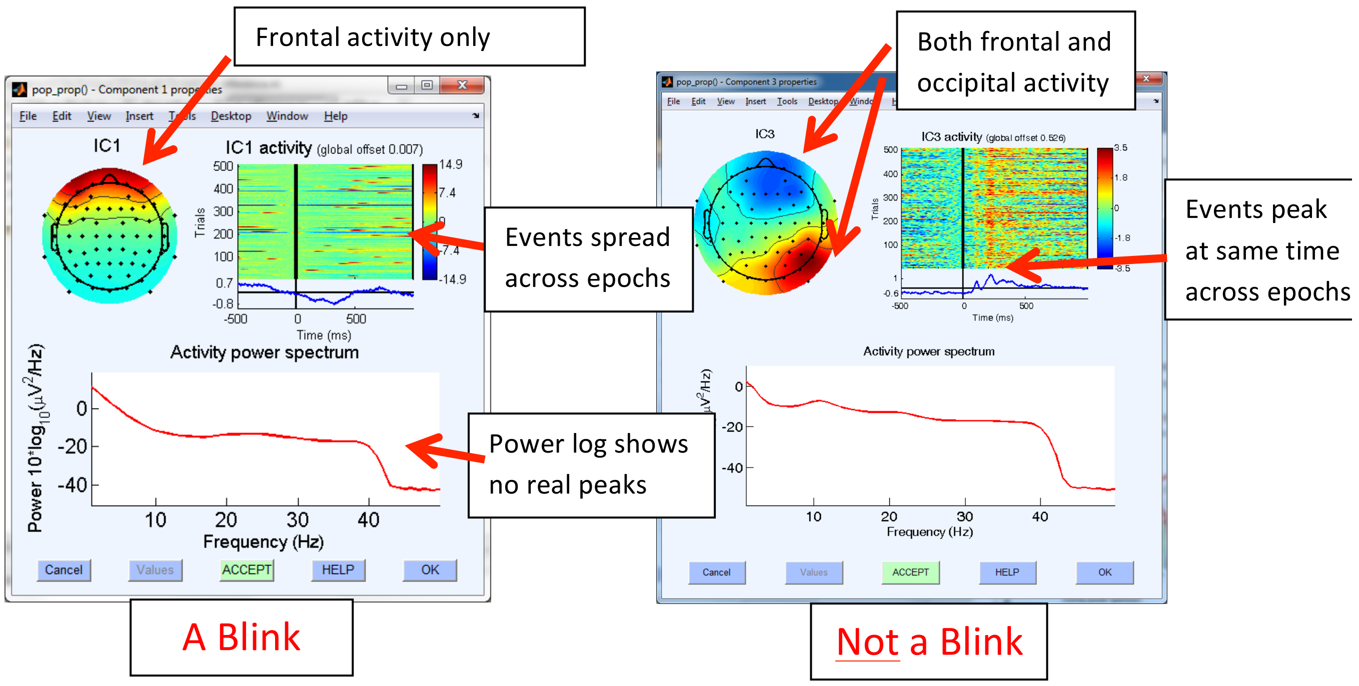
c. SAVE ("ss_expt_name_epoch_name_ica"). Make sure you save this file – as you don't want to redo the ica because you forgot to save it.
d. To look at the ICAs: Tools>Reject Data using ICA>reject components by map – should pop up a window that asks how many components to plot


Reject comp...

Components to plot: 1:32

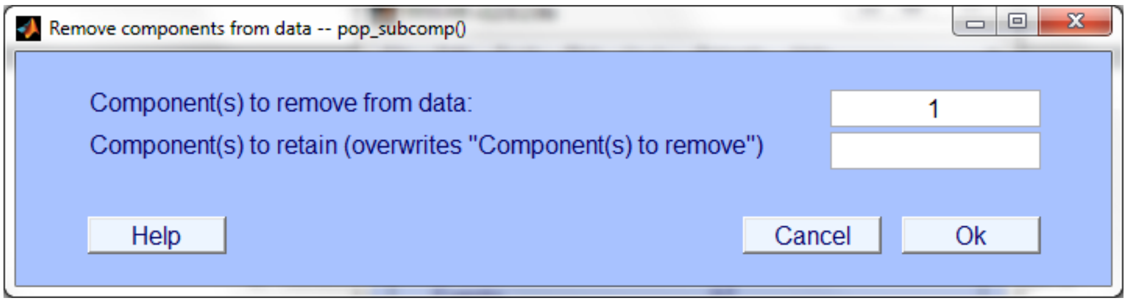


e. Select component with blinks in them – check the power spectrum (and eeglab wiki for what to look at)


Frontal activity only
Events spread across epochs
Power log shows no real peaks
A Blink
Both frontal and occipital activity
Events peak at same time across epochs
Not a Blink
pop_prop() - Component 1 properties
IC1
IC1 activity (global offset 0.007)
Activity power spectrum
Power 10*log10(μV²/Hz)
Frequency (Hz)
Time (ms)
Trials
Cancel
Values
ACCEPT
HELP
OK
pop_prop() - Component 3 properties
IC3
IC3 activity (global offset 0.526)

f. Tools>Remove components (this is to see if the data is correcting the blink)

i. Enter component to remove from data: blink ICA no. (e.g 1)



Remove components from data -- pop_subcomp()

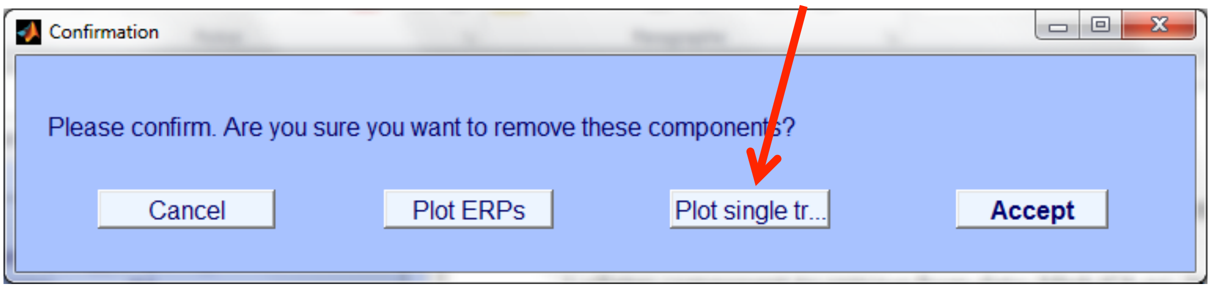
Component(s) to remove from data: 1

Component(s) to retain (overwrites "Component(s) to remove")

Help Cancel Ok

ii. Leave "component to overwrite" blank

iii. When "confirmation" dialog box opens click on "plot single trials"

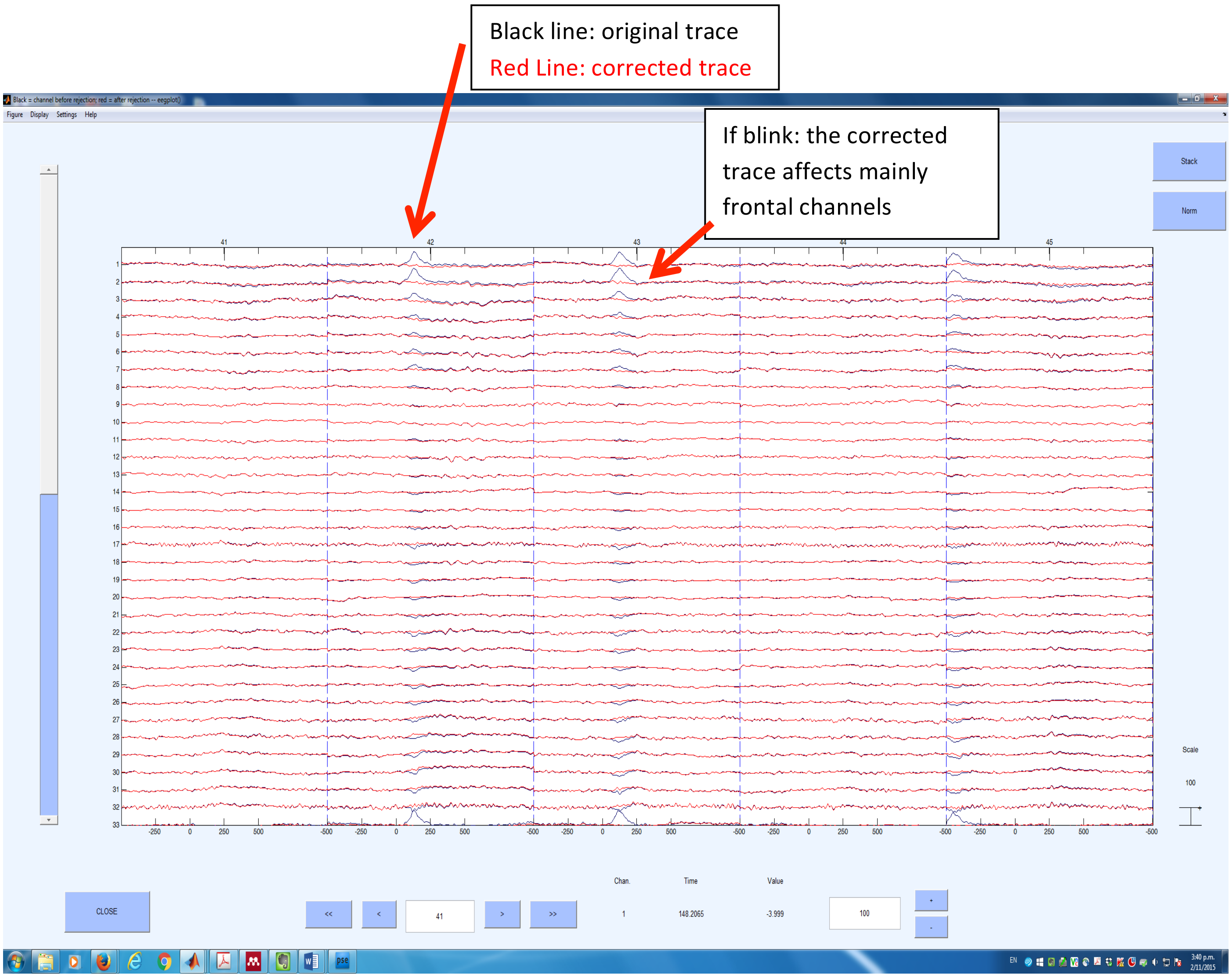


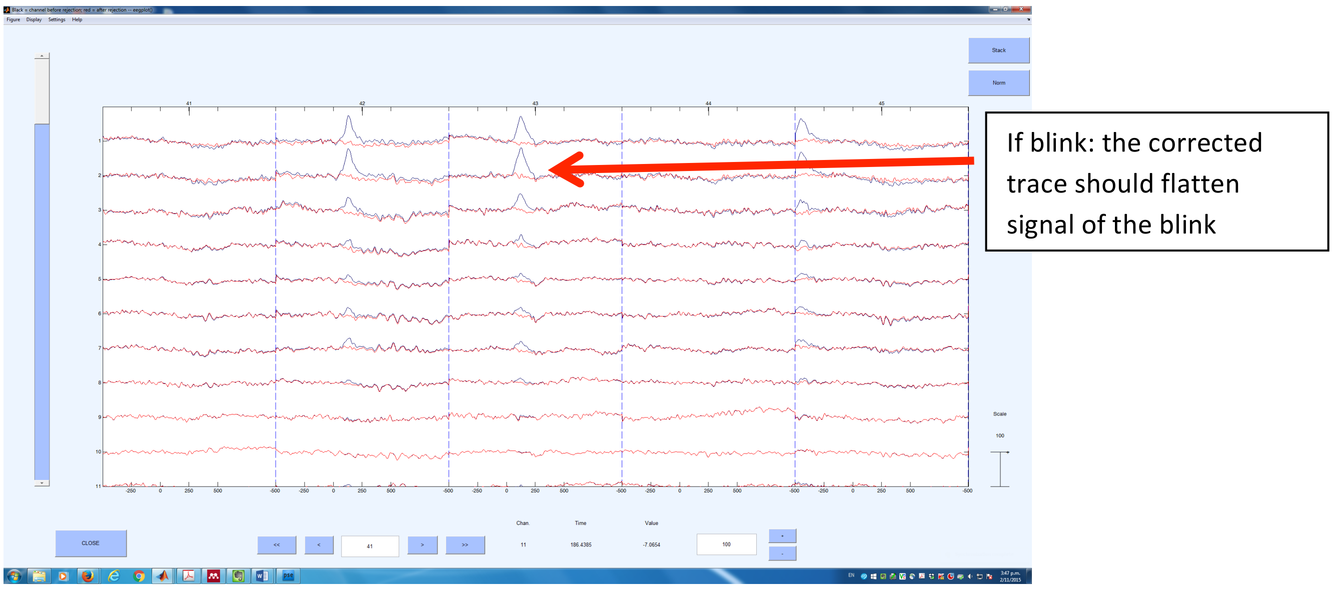
Confirmation

Please confirm. Are you sure you want to remove these components?

Cancel Plot ERPs Plot single tr... Accept

iv. The red line in the plots is your "corrected" raw trace and the black line is your previous epochs – for blink correction there should be a big difference in blinks between red and black lines.


Black line: original trace
Red Line: corrected trace
If blink: the corrected trace affects mainly frontal channels
Black = channel before rejection; red = after rejection -- eegplot()
Figure Display Settings Help
Stack
Norm
Scale
100
CLOSE
Chan.
Time
Value
1
148.2065
-3.999
41


If blink: the corrected trace should flatten signal of the blink
Stack
Norm
CLOSE
Chan.
Time
Value
11
186.4385
-7.0654
100

v. If you want to accept for rejection then click ACCEPT if not then click CANCEL

vi. SAVE the new file as ("ss_expt_name_epoch_name_bc")

g. Save the component numbers that have been removed in excel or a notepad document.

12- Reject Data Epochs

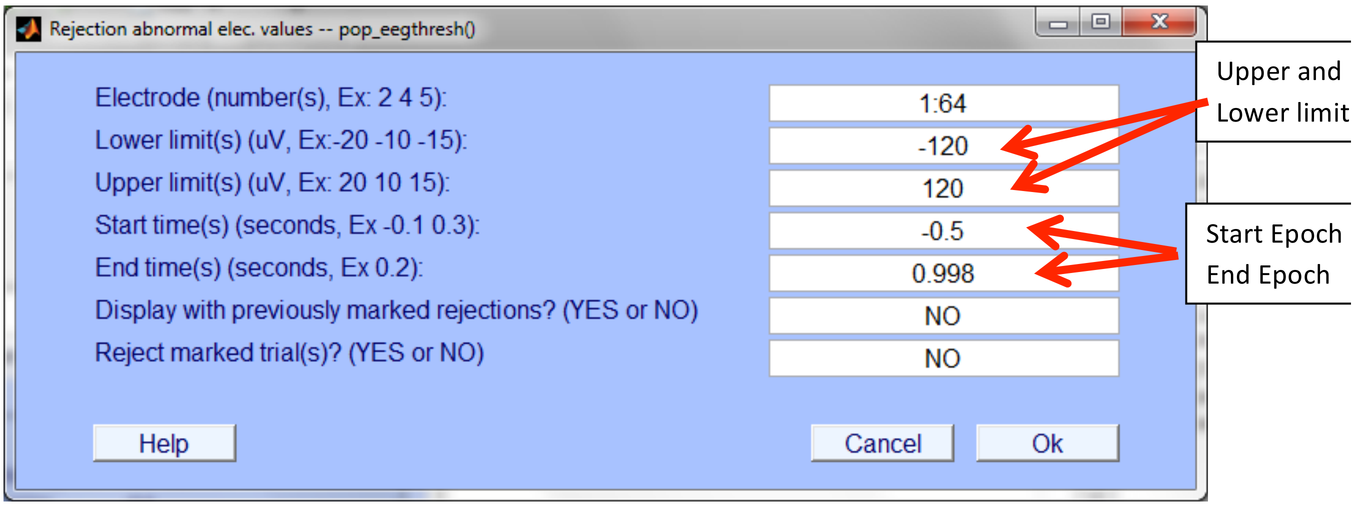
a. To select data epochs for averaging and analysis – need to exclude epochs based on a specific criteria (such as extreme values or low signal drift or sudden gradient changes). Only epochs that meet this criteria will be included in analysis. This is an example for extreme values.

b. Tools>Reject data epochs>reject extreme values

    i. Upper/lower limits (+120/-120) (or 100 depending on how strict you are).

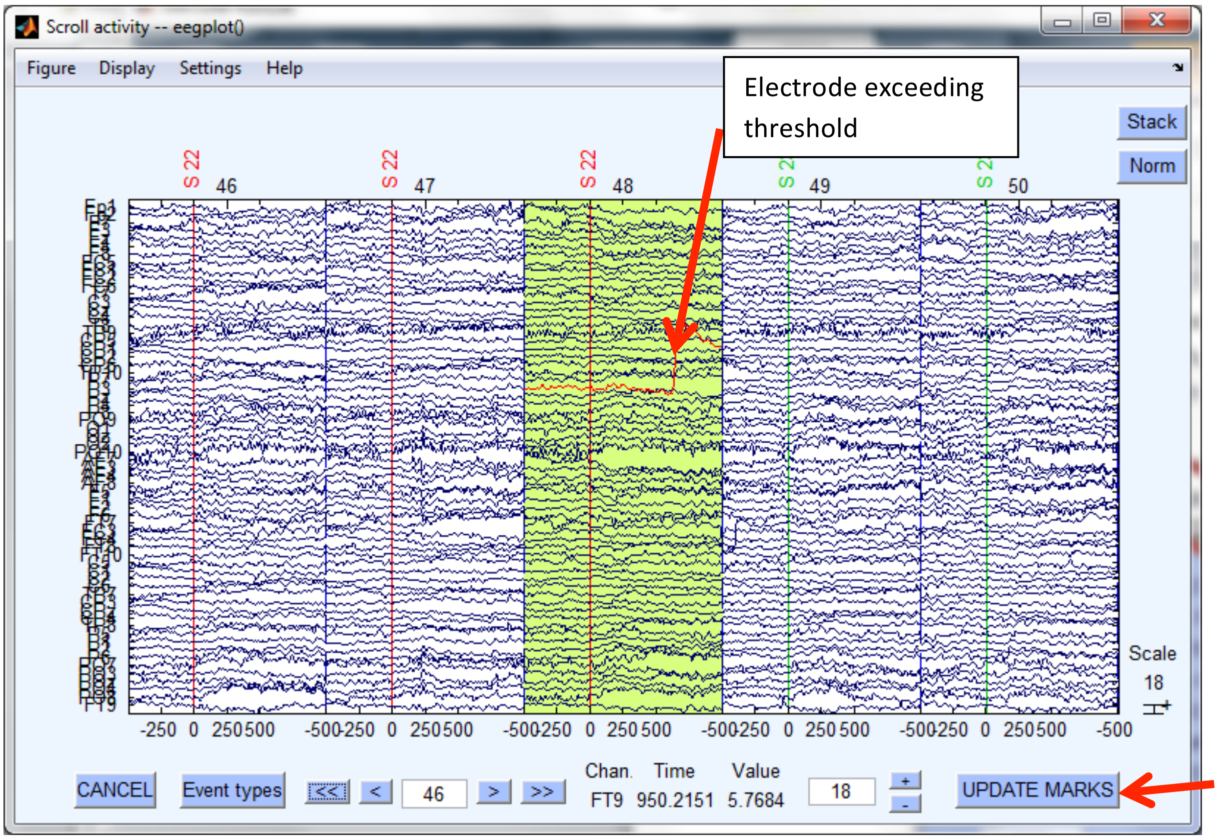
    ii. Start time/End time: specify the window you want artefact rejection to take place. Generally use the start of the epoch to the end of the epoch, but sometimes you may want a large epoch but only want artefact rejection to happen over a smaller window.

    iii. Display with previously marked rejections and Reject marked trials: keep at default "No"


Rejection abnormal elec. values -- pop_eegthresh()

Electrode (number(s), Ex: 2 4 5):	1:64
Lower limit(s) (uV, Ex:-20 -10 -15):	-120
Upper limit(s) (uV, Ex: 20 10 15):	120
Start time(s) (seconds, Ex -0.1 0.3):	-0.5
End time(s) (seconds, Ex 0.2):	0.998
Display with previously marked rejections? (YES or NO)	NO
Reject marked trial(s)? (YES or NO)	NO

c. Popup data plot: This will then highlight the epochs that have extreme values in them



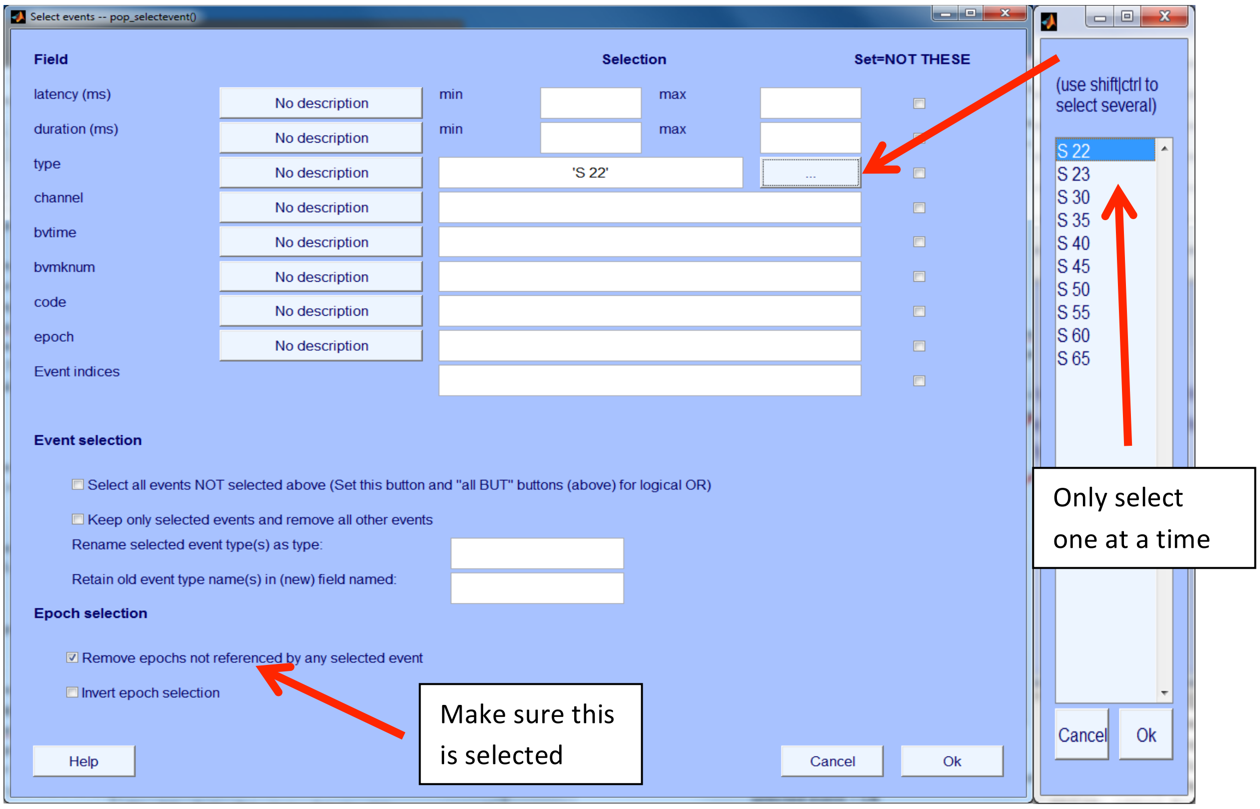
i. Similar to blink correction – a trace in an electrode that is red means it has exceeded your criteria
ii. You can visually inspect and then decide whether to accept them or not.
iii. If you think the epoch is ok – then click on it and it should not be highlighted anymore – then click **update marks**.
iv. If the same electrode is contributing to a lot of rejected epochs then you may want to look into interpolating that channel (you will need to go back to the Interpolation step)
v. Can determine if a channel is contributing to rejection threshold and decide to interpolate (see 9. Interpolation).
    1. Epochs that are going to be rejected are in workspace as **EEG.reject.rejthreshE** (channel x epoch matrix).
    2. If there is a "1" in a cell it means it is tagged for rejection
    3. Sort through each channel (row) to determine how many epochs are marked for rejection
    4. Generally if there is at least 20-25% of epochs rejected – then the channel needs interpolating
    5. EEG lab allows interpolation of a channel at this point too (refer to 8 on how to interpolate a channel).

6. If interpolating at this point then must apply the average reference after interpolation and not at the preprocessing step

d. Tools>Reject data epochs > reject marked epochs (remember once you have marked epochs to reject doesn't mean that they have been rejected) Make sure you do this step.
e. SAVE ("ss_expt_name_epoch_name_rej")
f. Take note of the no. of trials rejected – for erps – ideally need at *least* 80 trials in each condition (or over 75 to 80% of all trials accepted)– anything less than participant should be rejected from further analysis for failing to reach threshold

13- Selecting Data Epochs (Conditions)

a. For each ica corrected/artefact rejected epoch {ss_expt_name_epoch_name_rej } need to create separate conditions for each subject
b. Edit>Select Epochs/Events
c. In the dialog box for select the event click on type "…" and select the condition you want to use (you will create one of these condition files for each condition)



Select events -- pop_selectevent()

Field		Selection		Set=NOT THESE
latency (ms)	No description	min	max	☐
duration (ms)	No description	min	max	☐
type	No description	'S 22'	...	☐
channel	No description			☐
bvtime	No description			☐
bvmknum	No description			☐
code	No description			☐
epoch	No description			☐
Event indices				☐

Event selection

☐ Select all events NOT selected above (Set this button and "all BUT" buttons (above) for logical OR)

☐ Keep only selected events and remove all other events

Rename selected event type(s) as type:

Retain old event type name(s) in (new) field named:

Epoch selection

☑ Remove epochs not referenced by any selected event

☐ Invert epoch selection

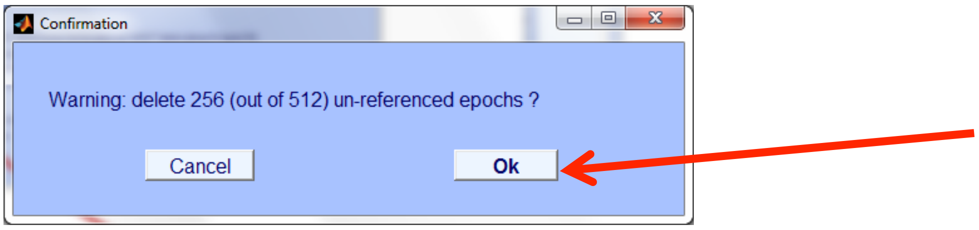
Make sure this is selected

(use shift|ctrl to select several)

S 22
S 23
S 30
S 35
S 40
S 45
S 50
S 55
S 60
S 65

Only select one at a time

d. Need to select (i.e. click) remove epochs not referenced by any selected event – OK
e. Click OK for warning dialogue box.



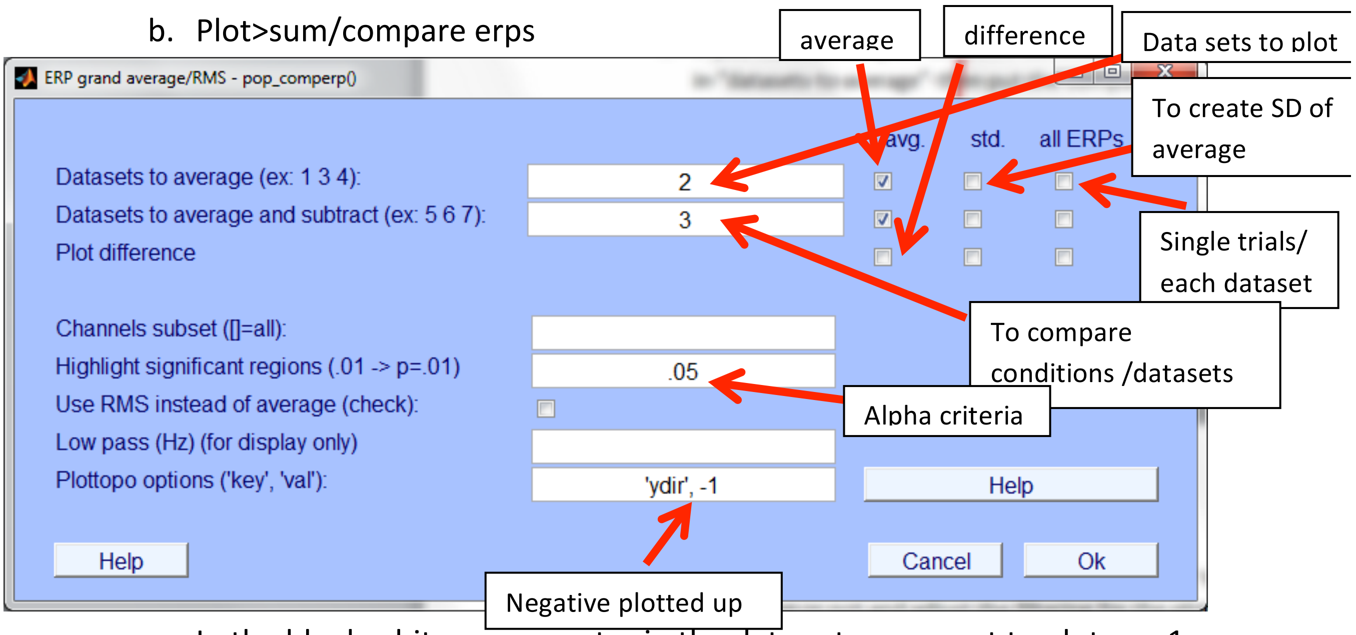
Confirmation

Warning: delete 256 (out of 512) un-referenced epochs ?

f. Name it something different– and save all of these files to a folder within your epoched data folder. Again change the name for each file. SAVE (“ss_expt_name_epoch_name_condition_name“).
g. Important: Note down the no. of epochs that are now included in that event (as this may be important for subject rejection etc.)
h. Redo this for each condition/event you want to look at - make sure you go back to the set file containing all events and not the condition files.

14- Compare Events for One Subject using Plots

a. Load condition datasets (you will be using the dataset number in the eeglab dialog box that has been assigned to these files)
b. Plot>sum/compare erps



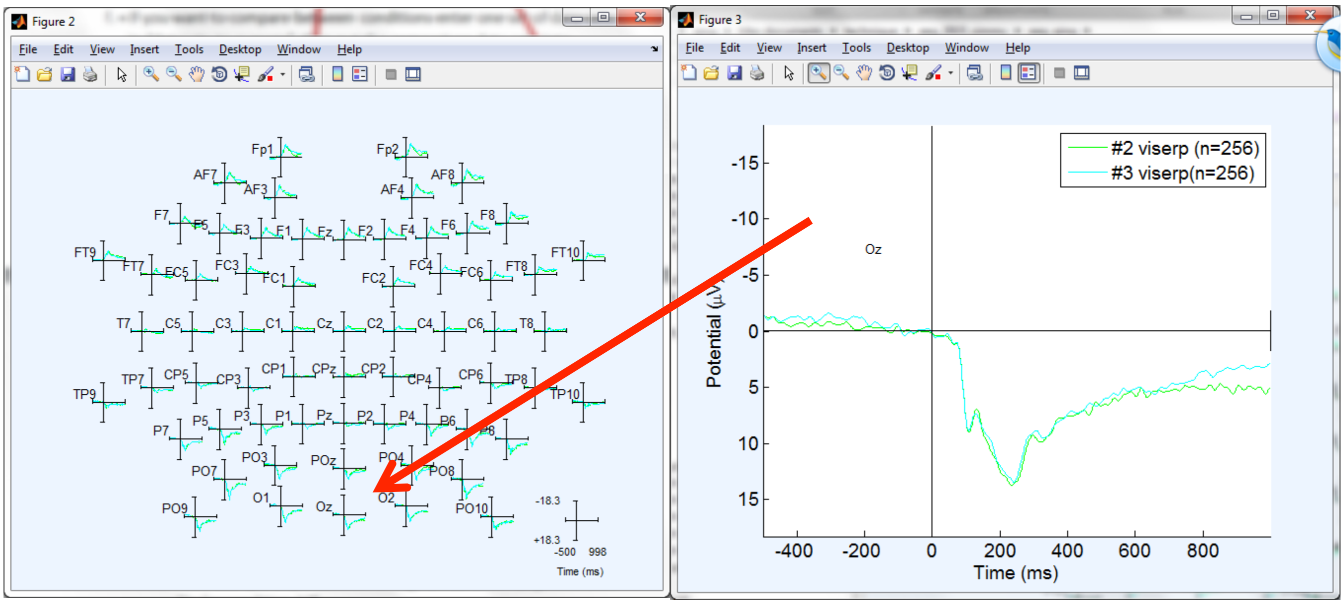
ERP grand average/RMS - pop_comperp()

Datasets to average (ex: 1 3 4): 2
Datasets to average and subtract (ex: 5 6 7): 3
Plot difference
Channels subset ([]=all):
Highlight significant regions (.01 -> p=.01): .05
Use RMS instead of average (check):
Low pass (Hz) (for display only)
Plottopo options ('key', 'val'): 'ydir', -1

c. In the blank white spaces enter in the data sets you want to plot e.g. 1 2 4
d. If want to average together then click “avg”
e. If don’t want the average across the entered conditions then click “all erps”
f. If you want to compare between conditions enter one set of data sets in “datasets to average” then put the comparison data sets into

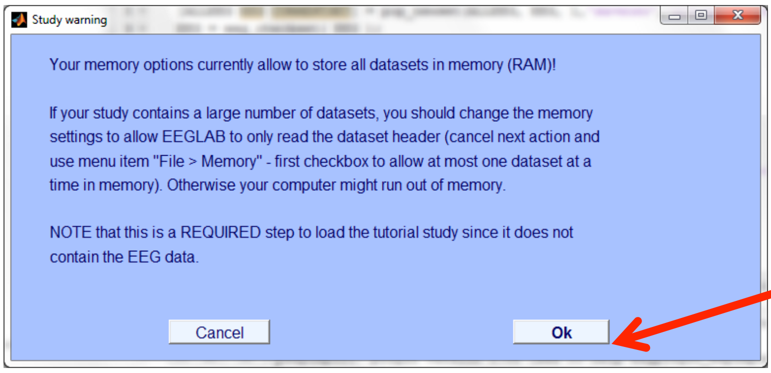
"datasets to average and subtract" - make sure avg is clicked for both rows. If you want to also plot the difference click avg in the plot difference line

g. If you want to highlight which regions are significant then enter .05 into highlight significant regions.
h. Other aspects of this dialog box are about whether you want negative plotted up or not and adjust the filtering for the plots.
i. Output: Plots of your conditions across the scalp



15- Creating Grand Averages – STUDY function

a. To create grand averages you can use the STUDY function in eeglab
    i. You can check the wiki to find out how to use this
b. It is good to use for plotting more than one conditions but maybe not so good to run statistics in
c. To create a study: File>Create Study>Simple ERP Study
    i. Will get a warning about needing a lot of memory



Study warning

Your memory options currently allow to store all datasets in memory (RAM)!

If your study contains a large number of datasets, you should change the memory settings to allow EEGLAB to only read the dataset header (cancel next action and use menu item "File > Memory" - first checkbox to allow at most one dataset at a time in memory). Otherwise your computer might run out of memory.

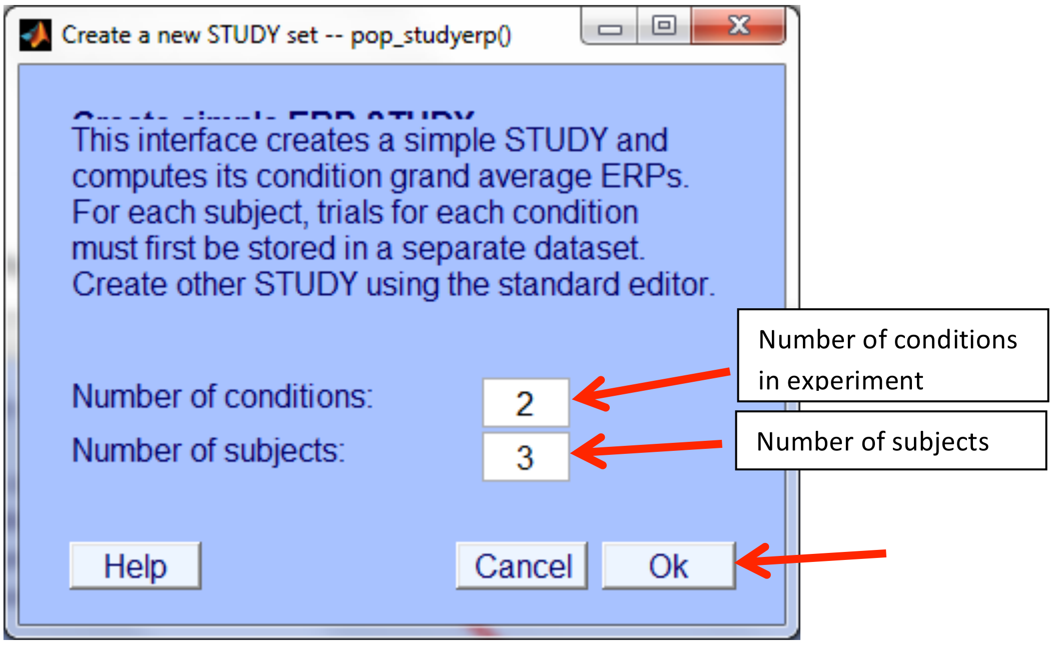
NOTE that this is a REQUIRED step to load the tutorial study since it does not contain the EEG data.

Note: need to have enough memory to open many data files

    ii. OK
d. Window: Create simple ERP study

i. Number of conditions: number of conditions in experiment e.g. 2

ii. Number of subjects: you intend to analyse in the study function

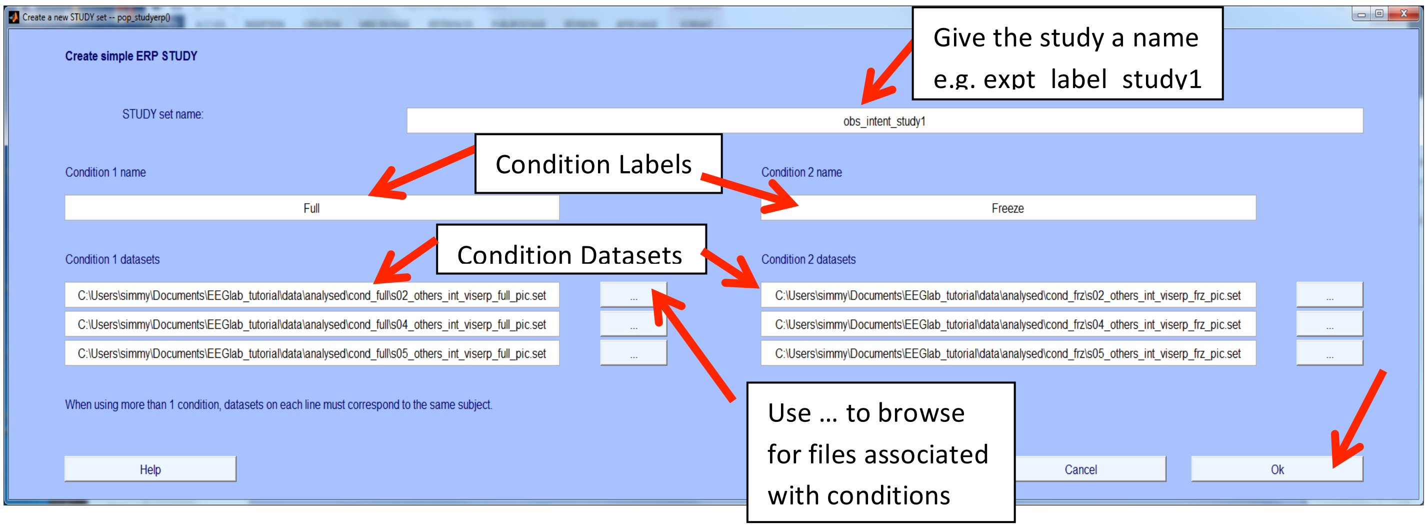

Create a new STUDY set -- pop_studyerp()
Number of conditions in experiment
Number of subjects
Help
Cancel
Ok

This interface creates a simple STUDY and computes its condition grand average ERPs. For each subject, trials for each condition must first be stored in a separate dataset. Create other STUDY using the standard editor.

Number of conditions: 2

Number of subjects: 3

e. Create a new Study window:


Create a new STUDY set -- pop_studyerp()
Give the study a name e.g. expt_label_study1
Condition Labels
Condition Datasets
Use … to browse for files associated with conditions
...
Help
Cancel
Ok

Create simple ERP STUDY

STUDY set name: obs_intent_study1

Condition 1 name: Full

Condition 2 name: Freeze

Condition 1 datasets:

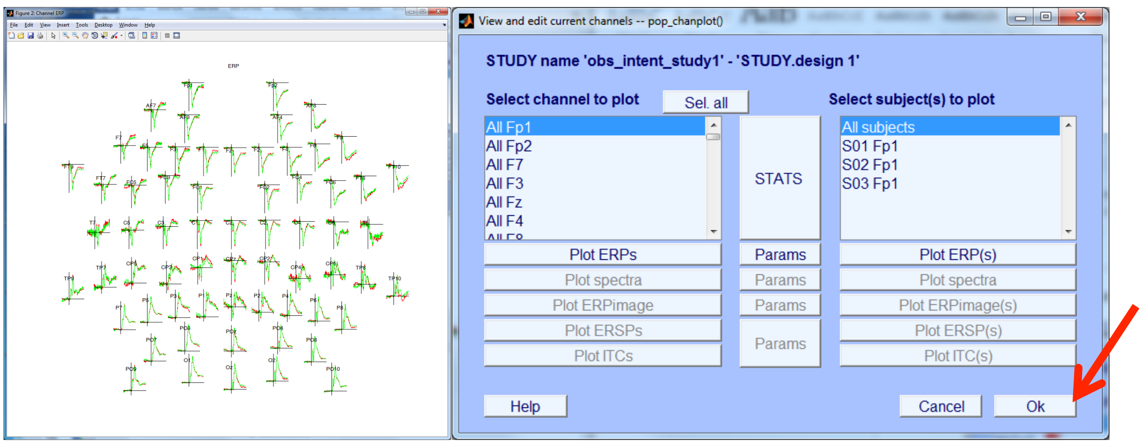
C:\Users\simmy\Documents\EEGlab_tutorial\data\analysed\cond_full\s02_others_int_viserp_full_pic.set
C:\Users\simmy\Documents\EEGlab_tutorial\data\analysed\cond_full\s04_others_int_viserp_full_pic.set
C:\Users\simmy\Documents\EEGlab_tutorial\data\analysed\cond_full\s05_others_int_viserp_full_pic.set

Condition 2 datasets:

C:\Users\simmy\Documents\EEGlab_tutorial\data\analysed\cond_frz\s02_others_int_viserp_frz_pic.set
C:\Users\simmy\Documents\EEGlab_tutorial\data\analysed\cond_frz\s04_others_int_viserp_frz_pic.set
C:\Users\simmy\Documents\EEGlab_tutorial\data\analysed\cond_frz\s05_others_int_viserp_frz_pic.set

When using more than 1 condition, datasets on each line must correspond to the same subject.

f. Will get a pop up of a grand average plot and a window of viewing and editing channels – press OK and the study will turn up in the eeglab workspace



View and edit current channels -- pop_chanplot()

STUDY name 'obs_intent_study1' - 'STUDY.design 1'

Select channel to plot | Sel. all

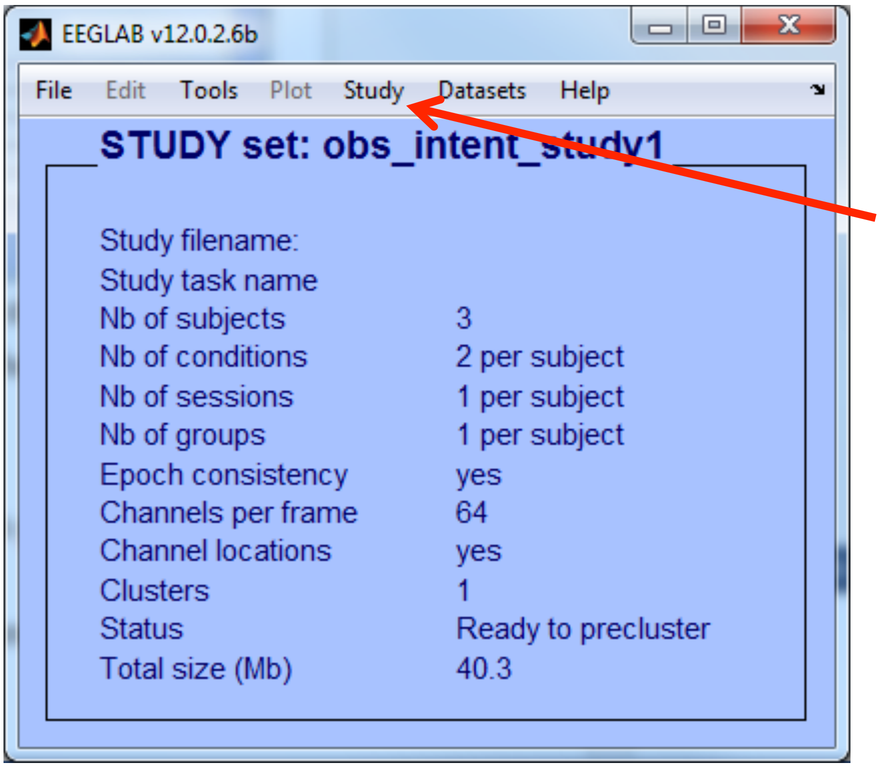
All Fp1
All Fp2
All F7
All F3
All Fz
All F4

STATS

Select subject(s) to plot

All subjects
S01 Fp1
S02 Fp1
S03 Fp1

g. IMPORTANT: SAVE the study. File>save current study as…
h. When reopen study file the options are under the Study Tab in the toolbar

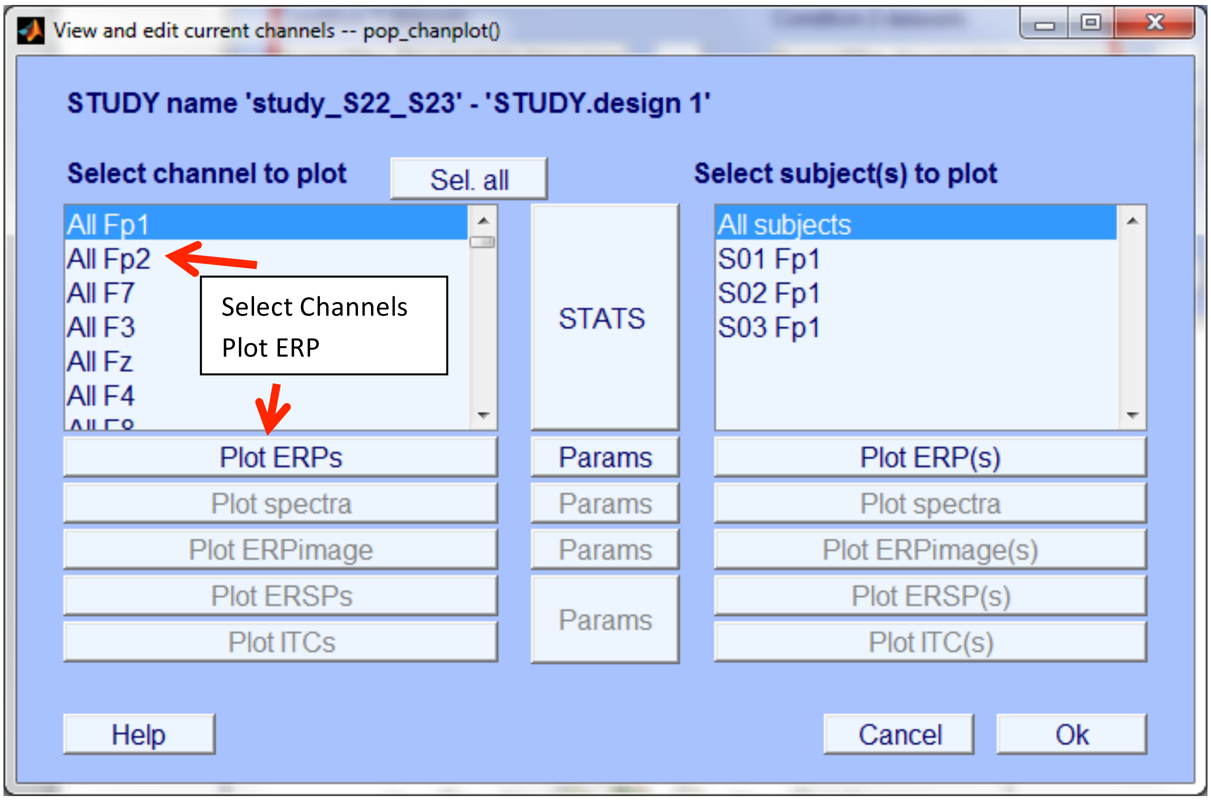


STUDY set: obs_intent_study1

Study filename:	
Study task name	
Nb of subjects	3
Nb of conditions	2 per subject
Nb of sessions	1 per subject
Nb of groups	1 per subject
Epoch consistency	yes
Channels per frame	64
Channel locations	yes
Clusters	1
Status	Ready to precluster
Total size (Mb)	40.3

i. Study>edit>edit study design
   i. Can edit the study – add in more subjects etc

j. Study>Plot Channel Measures

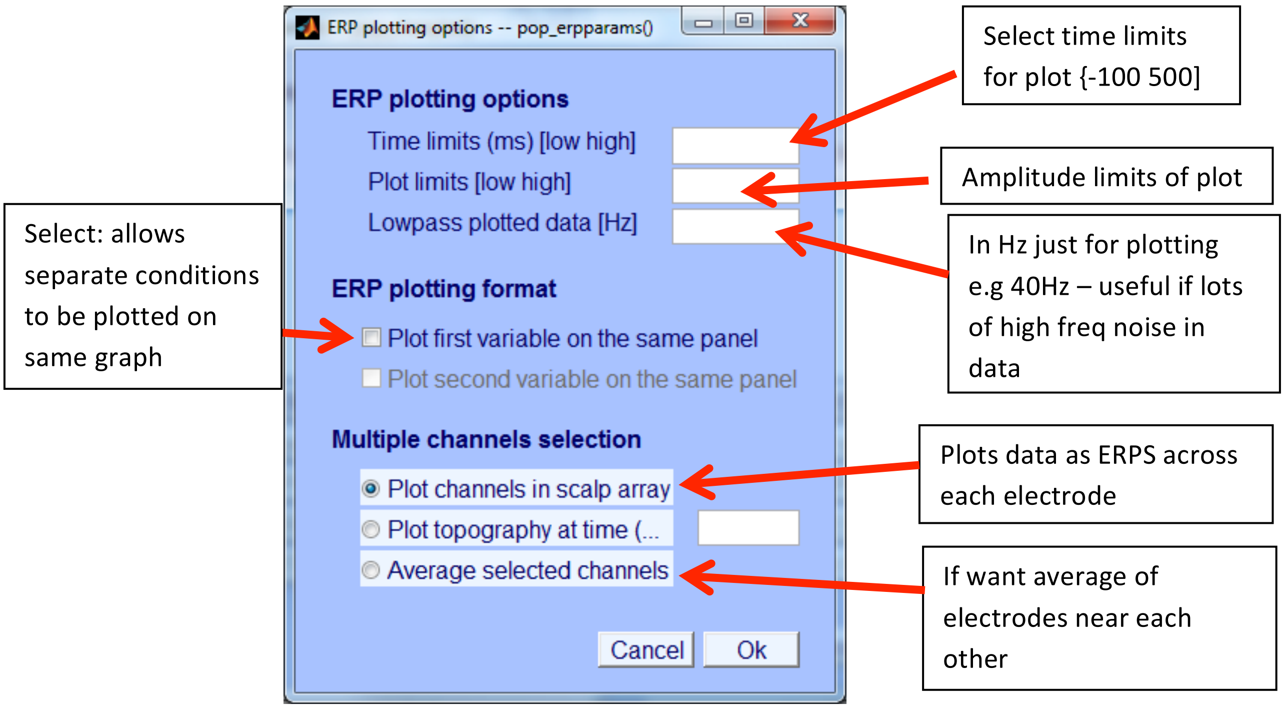


i. Provides a window of view and edit channels

ii. Can select several electrodes to plot

   1. Select channels
   2. Plot ERPS

iii. Parameters


ERP plotting options -- pop_erpparams()
ERP plotting options
Time limits (ms) [low high]
Plot limits [low high]
Lowpass plotted data [Hz]
ERP plotting format
Plot first variable on the same panel
Plot second variable on the same panel
Multiple channels selection
Plot channels in scalp array
Plot topography at time (...
Average selected channels
Cancel
Ok

Select time limits for plot {-100 500]

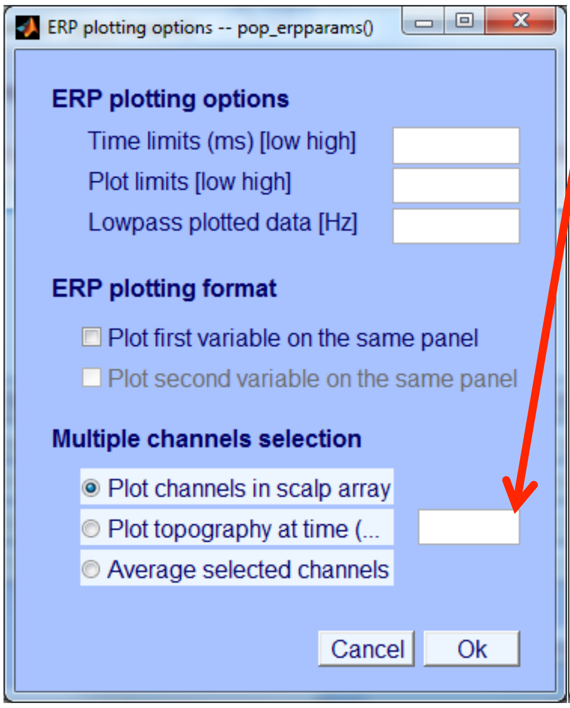
Amplitude limits of plot

In Hz just for plotting e.g 40Hz – useful if lots of high freq noise in data

Select: allows separate conditions to be plotted on same graph

Plots data as ERPS across each electrode

If want average of electrodes near each other



ERP plotting options -- pop_erpparams()

ERP plotting options

Time limits (ms) [low high]

Plot limits [low high]

Lowpass plotted data [Hz]

ERP plotting format

☐ Plot first variable on the same panel

☐ Plot second variable on the same panel

Multiple channels selection

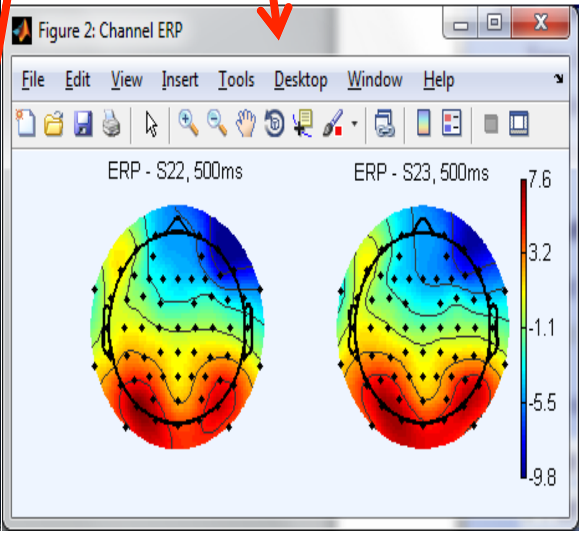
● Plot channels in scalp array

○ Plot topography at time (...

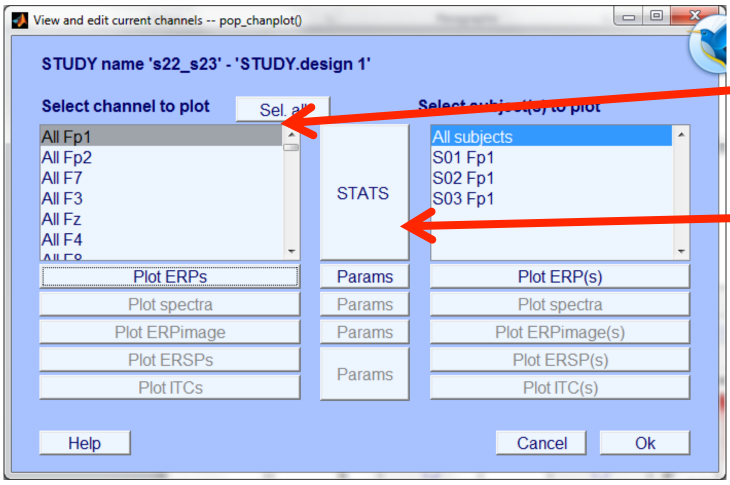
○ Average selected channels

Cancel | Ok

For Topoplot: Enter time point in ms and select "All Channels"



iv. Stats



View and edit current channels -- pop_chanplot()

STUDY name 's22_s23' - 'STUDY.design 1'

Select channel to plot Sel. all

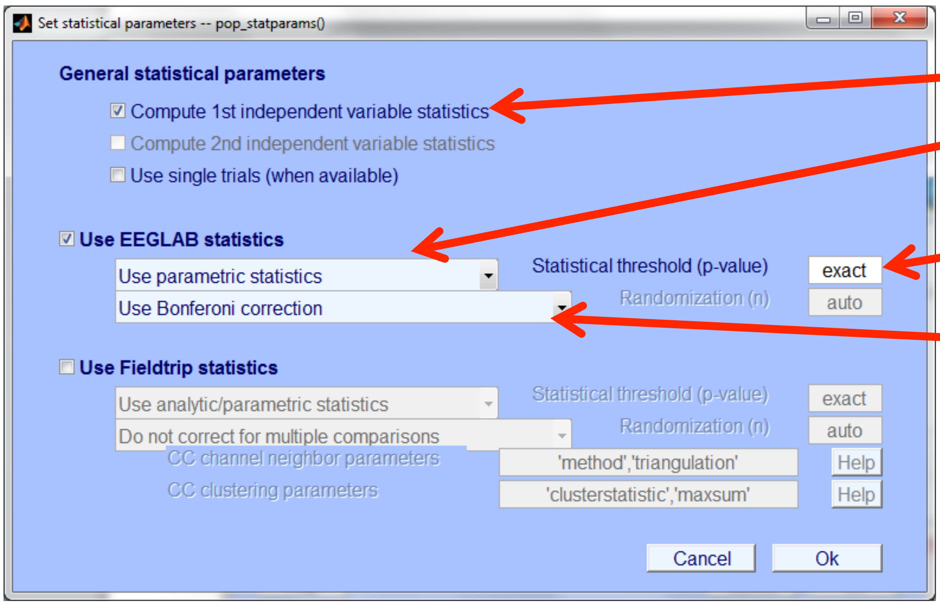
Select subject(s) to plot

Channels		Subjects
All Fp1 All Fp2 All F7 All F3 All Fz All F4 All F8	STATS	All subjects S01 Fp1 S02 Fp1 S03 Fp1
Plot ERPs	Params	Plot ERP(s)
Plot spectra	Params	Plot spectra
Plot ERPimage	Params	Plot ERPimage(s)
Plot ERSPs	Params	Plot ERSP(s)
Plot ITCs		Plot ITC(s)

Help | Cancel | Ok

Select Channels to run Statistics

Click Statistics



Set statistical parameters -- pop_statparams()

General statistical parameters

☑ Compute 1st independent variable statistics

☐ Compute 2nd independent variable statistics

☐ Use single trials (when available)

☑ **Use EEGLAB statistics**

Use parametric statistics | Statistical threshold (p-value): exact

Use Bonferoni correction | Randomization (n): auto

☐ **Use Fieldtrip statistics**

Use analytic/parametric statistics | Statistical threshold (p-value): exact

Do not correct for multiple comparisons | Randomization (n): auto

CC channel neighbor parameters: 'method','triangulation' Help

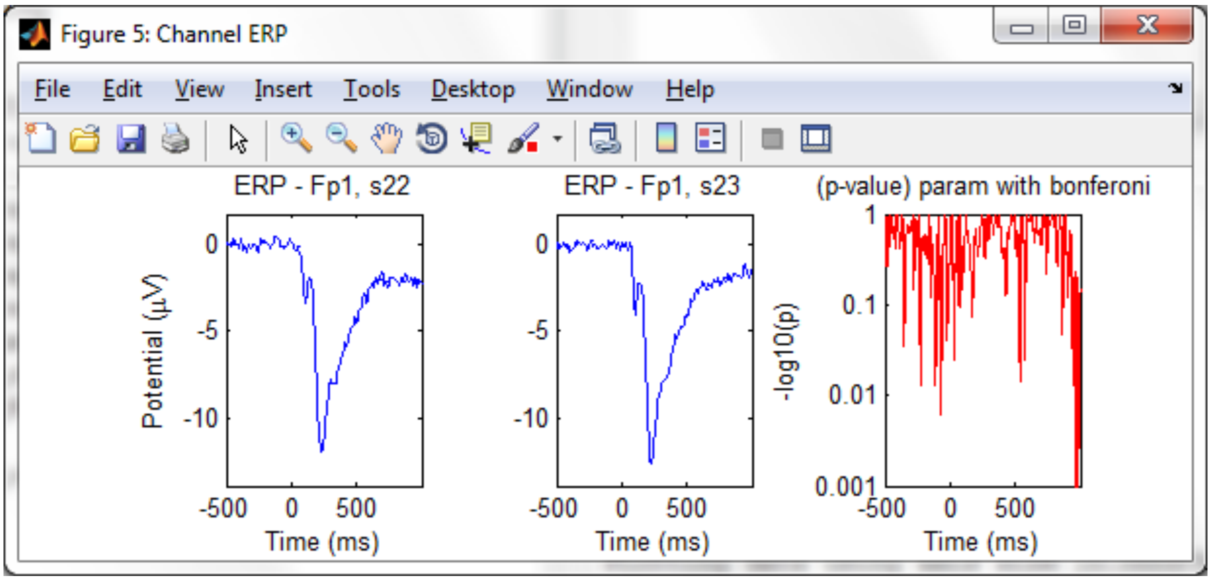
CC clustering parameters: 'clusterstatistic','maxsum' Help

Cancel | Ok

Options are: Parametric, Permutation, Bootstrap

Can keep exact or put in a threshold e.g. .05

To correct for multiple comparisons



16-Creating Grand Averages – using MATLAB workspace

a. Take each participants data for a given condition (e.g. load .set file for condition 1)
b. Data is in EEG.data within workspace (this is a big matrix of numbers (3D array): channels x time points x no. of epochs)
c. You need to create a 2D array of this data to get each subject's data average for that condition (channels x time points) (i.e. average across the number of trials e.g. ss_cond_avg = mean(data, 3))
d. Use each ppts average for that condition and save it into a cell matrix - that way you can access each participants averaged data easily – very handy for doing later statistics.
e. To create a grand average:
    i. Need to concatenate the data into one matrix (use cat function in matlab) (i.e. each condition has each subjects matrix consisting of channels x time points)
    ii. Then average together on last dimension (ndims+1) – this averages across subjects
    iii. This should give you a 64 (channels) x time points matrix of data points.
    iv. Do this for each of your conditions and save into one structure
    v. Select which electrodes you want to plot. Use the numbers assigned in the channel location file (e.g. Cz is 14)
    vi. Select the row (based on the electrode you want) and put that data row into a grand-avg matrix you will use to plot

vii. Do this for each condition, but attenuate each row underneath the previous (otherwise it will overwrite the previous row)

viii. You should end up with a matrix where each row corresponds to all ppts average data for each condition for one electrode.

ix. Use Plot function in matlab to plot the grandavg for these conditions at one electrode.

x. Save everything as a .mat file that can be loaded into matlab later.

17-Statistics

a. PCA?

b. Visually inspect your grand averaged ERPs and identify your components (Auditory ERPs: (p80), N1, P2, P300. Visual ERPs: P1, N170, P3) in the specific electrodes (Peaks: Auditory N1: FCz, Auditory P2: Cz, Auditory P3: CPz/Pz; Visual P1: Iz, Visual N170: Oz (for laterality effects use PO8 and PO7))

c. Select your time windows for analysis. For each component need a start time and end time – we will then create an average amplitude (µV), for each condition, across each subject that will be used in a statistical parametric test.

    i. Ideally the data for stats tests should be presented:

        1. Rows: each subjects data
        2. Columns: each condition (or each component_condition)
        3. Within each cell: the mean average (µV) across the time window for that condition for that subject

d. To calculate mean averages for each subject:

    i. Take cell arrays created in 15b(iv)

    ii. Take each subject's data for your given time window (EEG.data(channel no, [start_timewindow:end_timewindow])

        1. This should create a matrix of all electrodes data for data points within this time window
        2. Averaging across the time windows – gives you the mean amplitude at this time window for individual electrodes

    iii. Take the mean time window data from your selected electrode and append it to a matrix with all the other subjects mean voltage for that condition (i.e. each subject on a different row) Then concatenate the other conditions to form one data matrix of subjects x conditions (e.g. 16 subjects with 4 conditions should end up with a 16x4 matrix for each condition)

   iv. Steps i to iv may differ if you want to average across electrodes – take the mean of the electrode sets first – then calculate the average within the time window.

v. The final matrix can then be saved as a .mat file which can be reloaded into matlab to conduct stats tests or copied into SPSS or excel to create figures or run more tests.

18-Time Frequency Analysis

a. EEGLab allows you to make time frequency calculations on your data.
b. Create large epochs (containing double/triple the time of the lowest frequency you are interested in – so if you want to look at 4 Hz = a cycle every 250ms seconds (1/4 = .25), then you need each epoch to be at least triple that on each side (e.g epoch size -500 to 500ms) – this is due to the tapering function(?) which means data analysed is half that around the epoch size)
c. Calculate ersp –with outputs
d. Ersp is a matrix of freq x times data – containing complex no.s
e. These complex no.s need to be calulcated (some calculations are done)
f. Average
g. Oscillation data tends to need to be log transformed and to do a power analysis is a little more complicated then I can go into here.

19-Convert EEGLab to Fieldtrip

If you want to look into oscillations properly I would suggest doing preprocessing and epoching in EEGLab and then transferring everything into fieldtrip. Although please be aware that fieldtrip uses MATLAB command functions and does not have a GUI. So if you are not familiar with MATLAB at all – it may take a bit to understand what is going on. Also note: that there is a specific format that fieldtrip uses – so instead of having to start the preprocessing/epoching all again, there are a few quicker ways of getting the right info from your eeglab file and using it in your fieldtrip file. In particular fieldtrip needs information about trials, epochs and filenames – unfortunately the function used (eeglab2fieldtrip), doesn't necessarily extract all this info, so if you want to go straight into frequency processing or ERP stats analysis then here is what you need to do:

a. Download the fieldtrip toolbox (follow the install instructions on the fieldtrip wiki)
b. Open up your eeglab file (.set) – this is for the .set files that have already been split from other conditions.
c. Insert the following commands:
    i. data = eeglab2fieldtrip(EEG, ‘preprocessing’); % this is going to convert the eeglab data file to a fieldtrip structure file – but also indicate that it is similar to a “preprocessing” file from fieldtrip
    ii. Get info: data.hdr = ft_readheader(filename); data.events = ft_read_event(filename),
    iii. To get data.trl and data.epochs (important for fieldtrip further analysis):
        1. data.epochs = (find(strcmp (‘trial’, {data.events.type}))); %finds the no. of epochs in events.type.
        2. Then cycle through the epochs in data.epochs to get timing information for each trial
            a. Use data.events(trial_no).sample for trial onset
            b. Use data.events(trial_no).duration + trial onset for trial offset
            c. Use data.events(trial_no).sample for trial onset
            d. Use data.events(trial_no).offset for offset
            e. create a matrix where on each row you have the time onset of trial (trlbegin), time offset of trial (trlend), and the difference between the two (offset). This matrix should be inserted as data.trl
    iv. Take the data structure that you have created (this is now in fieldtrip format) and save it as a .mat file (fieldtrip saves everything as .mat files) and you should be ready to go.