## Contents

## Function to carry out the preliminary pre-processing steps of EEG-VR data.

Date: May 2018 Programmed by: D. Bolger

```
%****************************************************************************
```

## DEFINE PATHS FOR SUBJECT LEVEL FILES, CHANNEL LABEL FILE AND FILE CONTAINING ELECTRODE COORDINATES.

These paths need to be changed by a new user.

```
% Path to base directory containing the subject-level data.
DIR_main = fullfile('Users','bolger','Documents','work','Projects','Project-EEG-VR','ExpData',filesep);

% Path to file with channel labels.
chandir = fullfile('Users','bolger','Documents','work','Projects','Project-EEG-VR','ExpData','Chaninfo.mat');

% Path to file containg electrode coordinates for 64-electrode 10-20 system.
chlocpath = fullfile('Users','bolger','Documents','MATLAB','eeglab13_6_5b','plugins','dipfit2.3',...
                  'standard_BESA','standard-10-5-cap385.elp');

% Path to xls file containing word-frequency and uniqueness point data.
xls_wordfreq = fullfile('Users','bolger','Documents','work','Projects','Project-EEG-VR','EEG-VR_verb_frequencies.xlsx');

% Path to the wav files (auditory stimuli).
wav_path = fullfile('Users','bolger','Documents','work','Projects','Project-EEG-VR','VR-Embodiment wav files','soundfiles-fr-final',filesep);

% Path to the parameters *.mat file.
params_path = fullfile('Users','bolger','Documents','work','Projects','Project-EEG-VR','ExpData','EEG-VR_parameters.txt');
```

## OPEN DIALOGUE BOX TO SPECIFY SUBJECT NUMBER AND LIST TO PROCESS.

```
prompt1={'Subject Number/s:' 'List/s:'};
dlg_title = 'Specify subject and list:';
num_lignes = [10;10];
prompt1_ans = inputdlg(prompt1,dlg_title,num_lignes);

subs=cellstr(prompt1_ans{1,1});
sujindx=cellfun(@str2double,subs);
listcurr = cellstr(prompt1_ans{2,1});
```

## OPEN EEGLAB SESSION

Opens an EEGLAB session and presents main EEGLAB GUI.

```
addpath(genpath(''));
[ALLEEG, EEG, CURRENTSET, ALLCOM] = eeglab;
[ALLEEG, EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);


for counter = 1:length(sujindx)
```

## GENERATE THE CURRENT SUBJECT-LEVEL FILE-PATH

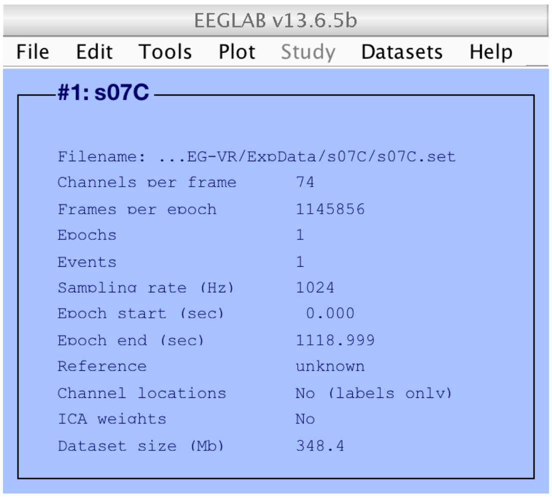
Tries to generate a file path string to matlab the real file-path for current subject.

```
    if sujindx < 10
        subjnom = strcat('s0',num2str(sujindx(counter)),listcurr{counter,1});
    else
        subjnom = strcat('s',num2str(sujindx(counter)),listcurr{counter,1});
    end

    Dirbase = fullfile(DIR_main,subjnom,filesep);     % Define the current subject-level filepath.
```

## LOAD IN THE *.bdf FILE OF THE CURRENT SUBJECT

Loads in the *.bdf file, ensuring that 74 channels are included so that the ERGO1 and ERGO2 data are loaded.

```
    allfiles= dir(Dirbase);
    fileIndex = find(~[allfiles.isdir]);
    filenum=dir(strcat(Dirbase,'*.bdf'));                         %find all the *.bdf files in the current folder
    filenom={filenum.name};


    fullDir = strcat(Dirbase,filenom{1,1});
    fnom = subjnom;

    % The following three lines is added to resolve a bug occurring when
    % opening the *.bdf file.
    x = fileparts( which('sopen') );
    rmpath(x);
    addpath(x,'-begin');

    % Opening up *.bdf file and saving as a *.set file.
    EEG = pop_biosig(fullDir, 'channels',[1:74], 'ref', [] ,'refoptions',{'keepref' 'off'} );
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'setname',char(fnom),'gui','off'); % Create a new dataset for the current raw data file
    [ALLEEG, EEG, CURRENTSET] = eeg_store(ALLEEG, EEG, CURRENTSET);
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(fnom),'filepath',Dirbase);  % Saves a copy of the current resampled dataset to the current directory
    eeglab redraw
```



#1: s07C

```
Filename: ...EG-VR/ExpData/s07C/s07C.set
Channels per frame      74
Frames per epoch        1145856
Epochs                  1
Events                  1
Sampling rate (Hz)      1024
Epoch start (sec)        0.000
Epoch end (sec)         1118.999
Reference               unknown
Channel locations       No (labels only)
ICA weights             No
Dataset size (Mb)       348.4
```

## CALL OF FUNCTION TO EXTRACT TRIAL DATA OUTPUT BY THE STIMULATION SYSTEM.

The trial level files output by unity should be in a "StimData/Trials/" folder saved in the current subject's folder. A stimdata.txt file will be saved in the subject-level directory. However, if this stimdata.txt file already exists for the subject, this step is skipped.

```
    stimdir = fullfile(Dirbase,'StimData',filesep);
    xtest = dir(stimdir);
    filesnom = {xtest.name};

    stim_test = strcat(subjnom,'_stimdata.txt');
    if sum(ismember(filesnom,stim_test))==0
        disp('-------Create stimdata file--------');
        stimfile = EEGVR_extract_trial_data(subjnom,Dirbase); % Call of function to assemble trial-level Unity output files.
    else
        disp('-------Stimdata file already present--------');
    end
```

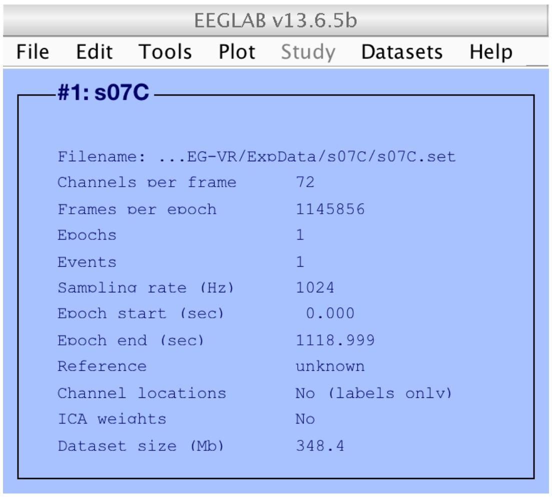
```
-------Create stimdata file--------
```

## EXTRACT THE ERGO DATA (AUDITORY STIMULI AND PHOTODIODE SIGNALS)

the photodiode and auditory stimuli channels are extracted from the EEG and saved to a separate field of the EEG structure, "EEG.dataERGO". It, therefore, erases the channels 73 and 74 and saves the changes. A figure of the auditory stimuli and the photodiode signals over is presented.

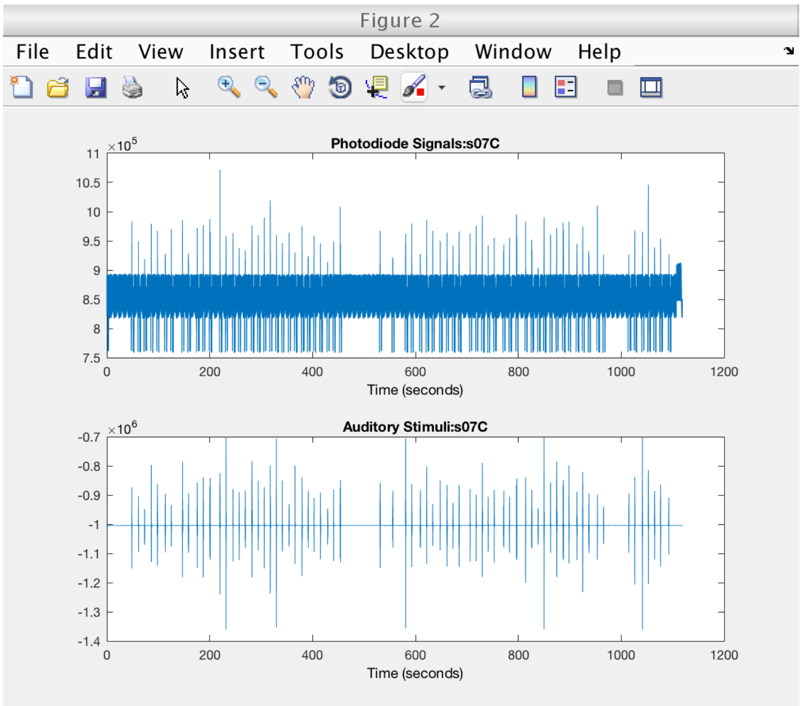
```
    EEG.dataERGO = EEG.data(73:74,:);

    EEG = pop_select( EEG,'nochannel',{'Erg1' 'Erg2'});
    EEG = eeg_checkset( EEG );

    [ALLEEG, EEG, CURRENTSET] = eeg_store(ALLEEG, EEG, CURRENTSET);
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(fnom),'filepath',Dirbase);  % Saves a copy of the current resampled dataset to the current directory
    eeglab redraw

    f1 = figure;
    subplot(2,1,1); plot(EEG.times./1000,EEG.dataERGO(1,:))
    title(strcat('Photodiode Signals:',subjnom));
    xlabel('Time (seconds)');
    subplot(2,1,2); plot(EEG.times./1000,EEG.dataERGO(2,:));
    title(strcat('Auditory Stimuli:',subjnom));
    xlabel('Time (seconds)');
```

```
Removing 2 channel(s)...
Saving dataset...
```



#1: s07C

```
Filename: ...EG-VR/ExpData/s07C/s07C.set
Channels per frame       72
Frames per epoch         1145856
Epochs                   1
Events                   1
Sampling rate (Hz)       1024
Epoch start (sec)         0.000
Epoch end (sec)          1118.999
Reference                unknown
Channel locations        No (labels only)
ICA weights              No
Dataset size (Mb)        348.4
```

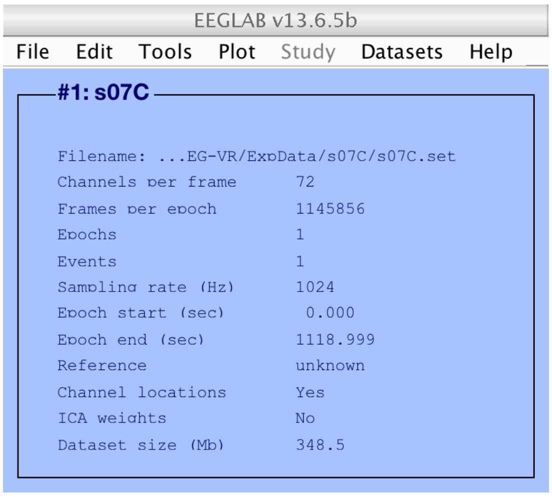


## ADD CHANNEL INFORMATION TO THE CURRENT DATASET.

Channel coordinates and labels are added to the current dataset and the dataset is saved to the current subject-level directory. The Chaninfo.mat file is loaded as it contains the electrode labels. From EEGLAB plugins, the file, "standard-10-5-cap385.elp" is loaded as this contains the correct coordinates for the 10-20 system used here.

```
    chaninfo = load(chandir,'Chaninfo');
    chans = chaninfo.Chaninfo;

    for cnt = 1:length(chans)
        EEG.chanlocs(cnt).labels = chans{1,cnt};
    end

    [ALLEEG, EEG, CURRENTSET] = eeg_store(ALLEEG, EEG, CURRENTSET);
    EEG = eeg_checkset( EEG );

    % !!!! For a new user, need to change this file path.

    EEG=pop_chanedit(EEG, 'lookup',chlocpath);                 % Load channel path information
    [ALLEEG, EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(fnom),'filepath',Dirbase);
    eeglab redraw
```

```
readlocs(): 'BESA' format assumed from file extension
BESA header detected, skipping three lines...
Readlocs: BESA spherical coords. converted, now deleting BESA fields
          to avoid confusion (these fields can be exported, though)
Channel lookup: no location for EXG1,EXG2,EXG3,EXG4,EXG5,EXG6,EXG7,EXG8
Send us standard location for your channels at eeglab@sccn.ucsd.edu
Saving dataset...
```



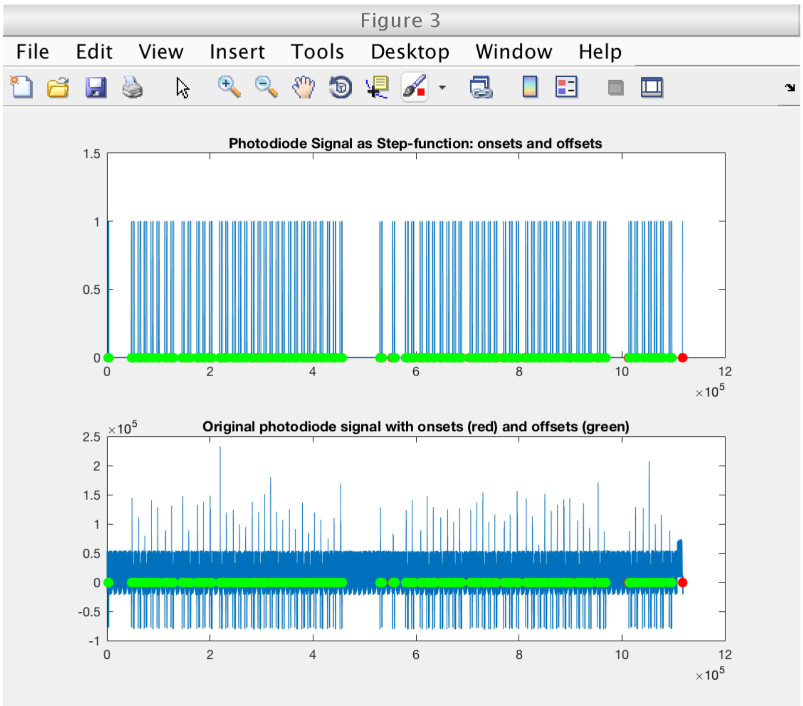
#1: s07C

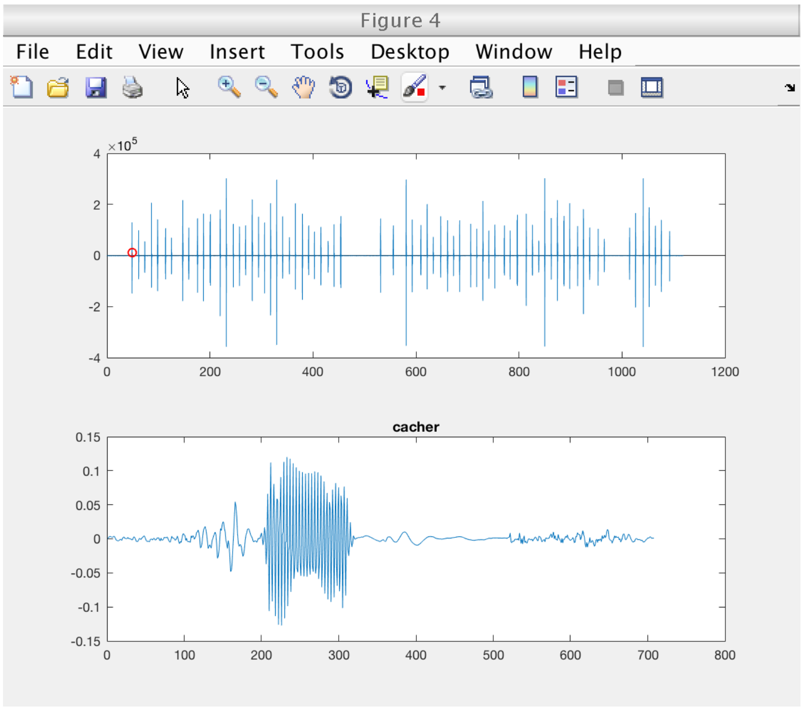
```
Filename: ...EG-VR/ExpData/s07C/s07C.set
Channels per frame      72
Frames per epoch        1145856
Epochs                  1
Events                  1
Sampling rate (Hz)      1024
Epoch start (sec)        0.000
Epoch end (sec)         1118.999
Reference               unknown
Channel locations       Yes
ICA weights             No
Dataset size (Mb)       348.5
```

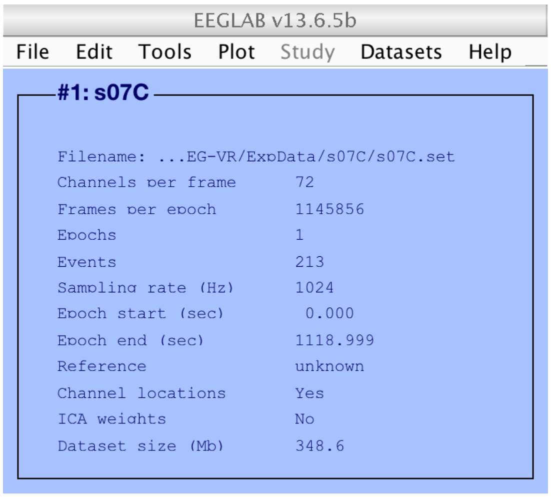
## NEED TO ADD EVENT INFORMATION BASED ON DATA RECORDED ONTO ERGO CHANNELS (AUDITORY STIMUI AND PHOTODIODE SIGNAL).

Trig_onsets_detect() function finds photodiode onset times. EEGVR_detect_audonsets() function finds the auditory stimuli onset times.

```
    trialtrigs = Trig_onsets_detect(EEG.dataERGO,EEG.times,EEG.srate,Dirbase);  % Call function to identify photodiode trigger onsets.
    audonsets = EEGVR_detect_audonsets(EEG.dataERGO(2,:),EEG.times,EEG.srate,Dirbase,subjnom,trialtrigs,wav_path);  % Call function to identify audit
ory stimuli onsets

    % Merging the trialtrigs and audonsets variables.
    T = trialtrigs(~cellfun('isempty',trialtrigs));
    Trialtrigs = reshape(T,[length(T)/size(trialtrigs,2),size(trialtrigs,2)]);
    ttrigs = cell(size(Trialtrigs,1),size(Trialtrigs,2));

    for cnt = 1:length(Trialtrigs)

        ttrigs{cnt,1} = Trialtrigs{cnt,1};
        ttrigs{cnt,2} = Trialtrigs{cnt,2};
        ttrigs{cnt,3} = Trialtrigs{cnt,3};
        ttrigs{cnt,4} = Trialtrigs{cnt,4};
    end

    % Sort latency data for auditory stimulus and photodiode signal onsets.
    lat_all = cat(1,audonsets{:,4},ttrigs{:,4});
    time_all = cat(1,audonsets{:,1},ttrigs{:,2});
    type_all = cat(1,{audonsets{:,2}}',{ttrigs{:,1}}');
    [latencies_all,indx] = sort(lat_all,'ascend');
    lat_all = lat_all(indx);
    time_all = time_all(indx);
    type_all = {type_all{indx}}';

% Integrate the onset times into the event field of the EEG structure.
%Create the EEG.event field from the "audonsets" and "Trigtrials"
% variables.
    EEG.event = [];
    EEG.urevent = [];

    for cnt1 = 1:length(type_all)
        EEG.event(cnt1).type = type_all{cnt1,1};
        EEG.event(cnt1).latency = lat_all(cnt1,1);
        EEG.event(cnt1).urevent = cnt1;
        EEG.urevent(cnt1).type = type_all{cnt1,1};
        EEG.urevent(cnt1).latency = lat_all(cnt1,1);
    end

    % Save the dataset with onset information added to the current
    % subject-level directory.
    [ALLEEG, EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(fnom),'filepath',Dirbase);
    eeglab redraw
```







#1: s07C

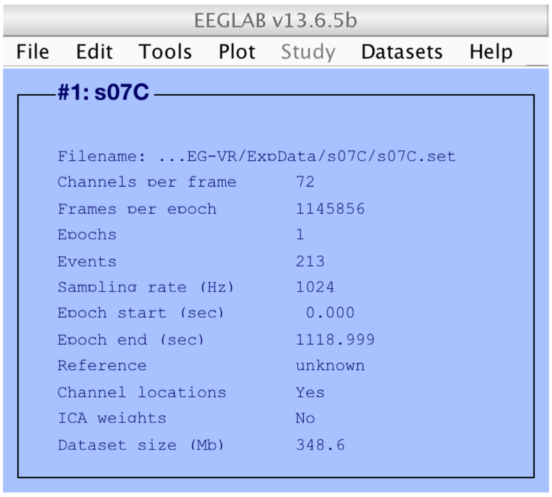
```
Filename: ...EG-VR/ExpData/s07C/s07C.set
Channels per frame        72
Frames per epoch          1145856
Epochs                    1
Events                    213
Sampling rate (Hz)        1024
Epoch start (sec)          0.000
Epoch end (sec)           1118.999
Reference                 unknown
Channel locations         Yes
ICA weights               No
Dataset size (Mb)         348.6
```

## ASSIGN TRIAL-TYPES (GO OR NOGO) AND INFORMATION REGARDING "GOOD" AND "BAD" TRIALS.

This facilitates epoching only go or no-go trials separately and the automatic rejection of bad trials after epoching. Note that the word "bad" is added to auditory stimuli of incorrect trials.

```
[EEG,currconds] = EEGVR_trialtype_assign(EEG,subjnom,Dirbase);

if ~exist('fnom')       % Sometimes the dataset does not have a setname field defined.
    fnom = EEG.setname;
end

% Save the dataset with this trial information added to the current
% subject-level directory.
[ALLEEG, EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
EEG = eeg_checkset( EEG );
EEG = pop_saveset( EEG, 'filename',char(fnom),'filepath',Dirbase);
eeglab redraw
```

```
Saving dataset...
```



#1: s07C

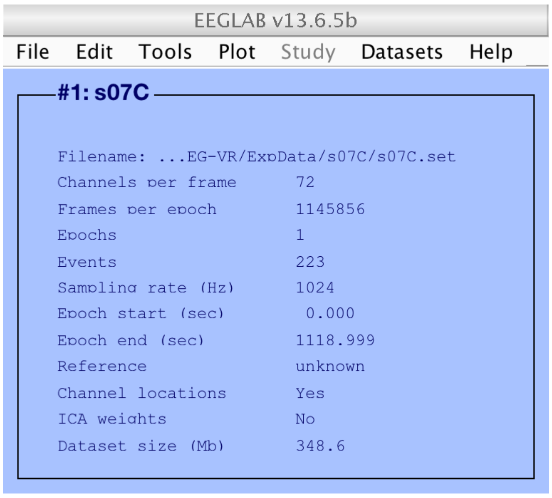
```
Filename: ...EG-VR/ExpData/s07C/s07C.set
Channels per frame      72
Frames per epoch        1145856
Epochs                  1
Events                  213
Sampling rate (Hz)      1024
Epoch start (sec)        0.000
Epoch end (sec)         1118.999
Reference               unknown
Channel locations       Yes
ICA weights             No
Dataset size (Mb)       348.6
```

## INCLUDE ROUTINE HERE TO IDENTIFY THE WORDS "FAIRE PIVOTER" AND "FAIRE TOMBER" AND ADJUST THEIR T0

It incorporates new triggers into the events field following the triggers "tomber", "fairepivoter". It changes the "tomber-go" to "faire-go" and 350ms a new trigger called "tomber" is added. It changes the "fairepivoter" to "faire" followed 350ms later by "fairepivoter". In this way, the non-epoched data has a trigger for both the onset of "faire" and the onsets of "pivoter" and "tomber", just in case we decide to analyse the data with T0 at the onset of "faire"...could always happen!!

```
words_oi = {'tomber' 'fairepivoter'};
words_all = {EEG.event.type};
windx = cell(length(words_oi),1);
newint = 0.35;

% Isolate the indices of these two words.
for wcnt = 1:length(words_oi)
    x = cell2mat(cellfun(@isempty,strfind(words_all,words_oi{1,wcnt}),'UniformOutput',false));
    windx{wcnt} = find([x==0]);
end

latadd = ceil(EEG.srate * newint);  %Number of samples to add
lenadd = length(windx{1,1})+length(windx{2,1});
lentotal = length(words_all)+lenadd;
events_type = cell(lentotal,1);
events_lat = cell(lentotal,1);
events_indx = cell(lentotal,1);

tcnt2 = 0;
for tcnt = 1:length(words_all)
    tcnt
    tcnt2 = tcnt2+1;

    if ismember(tcnt,windx{1,1})  %If current event corresponds to "faire tomber"
        if strcmp(EEG.event(tcnt).type(end-2:end),'bad')
            ender = 'bad';
        else
            ender = EEG.event(tcnt-1).type;
        end

        events_indx{tcnt2} = tcnt2;
        events_type{tcnt2,1} = strcat('faire-',ender);
        events_lat{tcnt2} = EEG.event(tcnt).latency;

        tcnt2 = tcnt2 + 1;
        events_indx{tcnt2} = tcnt2;
        events_type{tcnt2,1} = strcat('tomber-',ender);
        events_lat{tcnt2} = EEG.event(tcnt).latency + latadd;  % Add on the number of samples corresponding to 350ms.

    elseif ismember(tcnt,windx{2,1}) %If current event corresponds to "faire pivoter"
        if strcmp(EEG.event(tcnt).type(end-2:end),'bad')
            ender = 'bad';
        else
            ender = EEG.event(tcnt-1).type;
        end

        events_indx{tcnt2} = tcnt2;
        events_type{tcnt2,1} = strcat('faire-',ender);
        events_lat{tcnt2} = EEG.event(tcnt).latency;

        tcnt2 = tcnt2 + 1;
        events_indx{tcnt2} = tcnt2;
        events_type{tcnt2,1} = strcat('fairepivoter-',ender);
        events_lat{tcnt2} = EEG.event(tcnt).latency + latadd;  % Add on the number of samples corresponding to 350ms.

    else
        events_indx{tcnt2} = tcnt2;
        events_type{tcnt2,1} = EEG.event(tcnt).type;
        events_lat{tcnt2} = EEG.event(tcnt).latency;
    end

end

EEG.event = struct('type',events_type,'latency',events_lat,'urevent',events_indx);

[ALLEEG, EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
EEG = eeg_checkset( EEG );
EEG = pop_saveset( EEG, 'filename',char(fnom),'filepath',Dirbase);
```

```
eeglab redraw
```



#1: s07C

```
Filename: ...EG-VR/ExpData/s07C/s07C.set
Channels per frame      72
Frames per epoch        1145856
Epochs                  1
Events                  223
Sampling rate (Hz)      1024
Epoch start (sec)        0.000
Epoch end (sec)         1118.999
Reference               unknown
Channel locations       Yes
ICA weights             No
Dataset size (Mb)       348.6
```

## PREPARE INFORMATION TEXT-FILE FOR THE CURRENT SUBJECT.

```
    fname=strcat(fnom,'-info.txt');
    fdir=strcat(Dirbase,fname);
    fid=fopen(fdir,'w');
    fprintf(fid,['----------',fnom,'----------\n\n']);

    % Adds information regarding the total number of trials and the total number of "go","nogo" and
    % "bad" trials to this text file.
    nogo_num = length(find(strcmp(currconds{1,5},'NOGO')));     % The number of "nogo" trials.
    go_num = length(currconds{1,5}) - nogo_num;                 % The number of "go" trials.
    bad_num = length(find(strcmp(currconds{1,7},'badtrial')));  % The number of bad trials.
    verbs = currconds{1,2};
    badwords = verbs(strcmp(currconds{1,7},'badtrial'));        % The verbs corresponding to the bad trials.
    badtrial_indx = find(strcmp(currconds{1,7},'badtrial'));

    fprintf(fid,'The total number of trials is: %d\n',length(currconds{1,5}));
    fprintf(fid,'The total number of GO trials is: %d\n',go_num);
    fprintf(fid,'The total number of NOGO trials is: %d\n',nogo_num);
    fprintf(fid,'The number of incorrect trials: %d\n',bad_num);
    fprintf(fid,'The verbs corresponding to incorrect trials:');
    for i = 1:bad_num
        fprintf(fid,'%s\t',badwords{i,1});
    end
    fprintf(fid,'\nTrial indices corresponding to incorrect trials:');
    for i1 = 1:bad_num
        fprintf(fid,'%d\t',badtrial_indx(i1));
    end
    fprintf(fid,'\n---------------------------------------\n');
```

## OPEN THE PARAMETERS FILE WITH PRE-PROCESSING PARAMETERS.

Load the pre-processing parameters defined in the parameters *.txt file into the current workspace.

```
    % !!! Need to change this file path for a new user.
    fid2=fopen(params_path);      % il faut changer le chemin
    mydata=textscan(fid2,'%s %s');

    for i = 1:length(mydata{1,1})                       % generate a parameters structure from the parameters text file
        Params.(genvarname(mydata{1,1}{i}))=mydata{1,2}(i);
    end

    f_low = str2double(Params.fc_low);
    f_hi= str2double(Params.fc_hi);
    SR = str2double(Params.srate);
    SR_orig = EEG.srate;
```

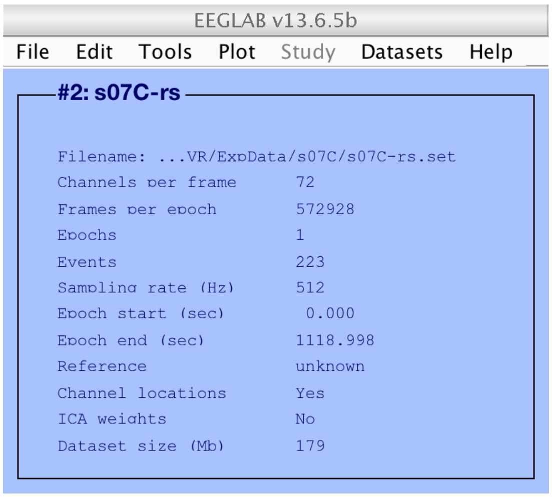
## RESAMPLE THE CONTINUOUS DATA USING THE PARAMETER DEFINED IN THE PARAMETERS TEXT FILE.

Resamples using the EEG resampling function. If the user has the Matlab signal processing toolbox, it uses the Matlab resample() function. Write information regarding the resampling to the subject-level text file.

```
    fprintf(fid,'\nDownsampled from %fHz to %fHz\n',SR_orig,SR);
    display('***********************************Resampling to 512Hz*******************************************')
    fnom_rs = strcat(fnom,'-rs');

    EEG = pop_resample(EEG, SR);   %resample the data at sampling rate defined, sr.
    EEG =eeg_checkset(EEG);
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'setname',char(fnom_rs),'gui','off'); % current set = xx;
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(fnom_rs),'filepath',Dirbase);  % Saves a copy of the current resampled dataset to the current directory
    eeglab redraw
```

```
***********************************Resampling to 512Hz*******************************************
```

```
resampling data 512.0000 Hz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
resampling event latencies...
eeg_checkset note: upper time limit (xmax) adjusted so (xmax-xmin)*srate+1 = number of frames
resampling finished
Creating a new ALLEEG dataset 2
Saving dataset...
```



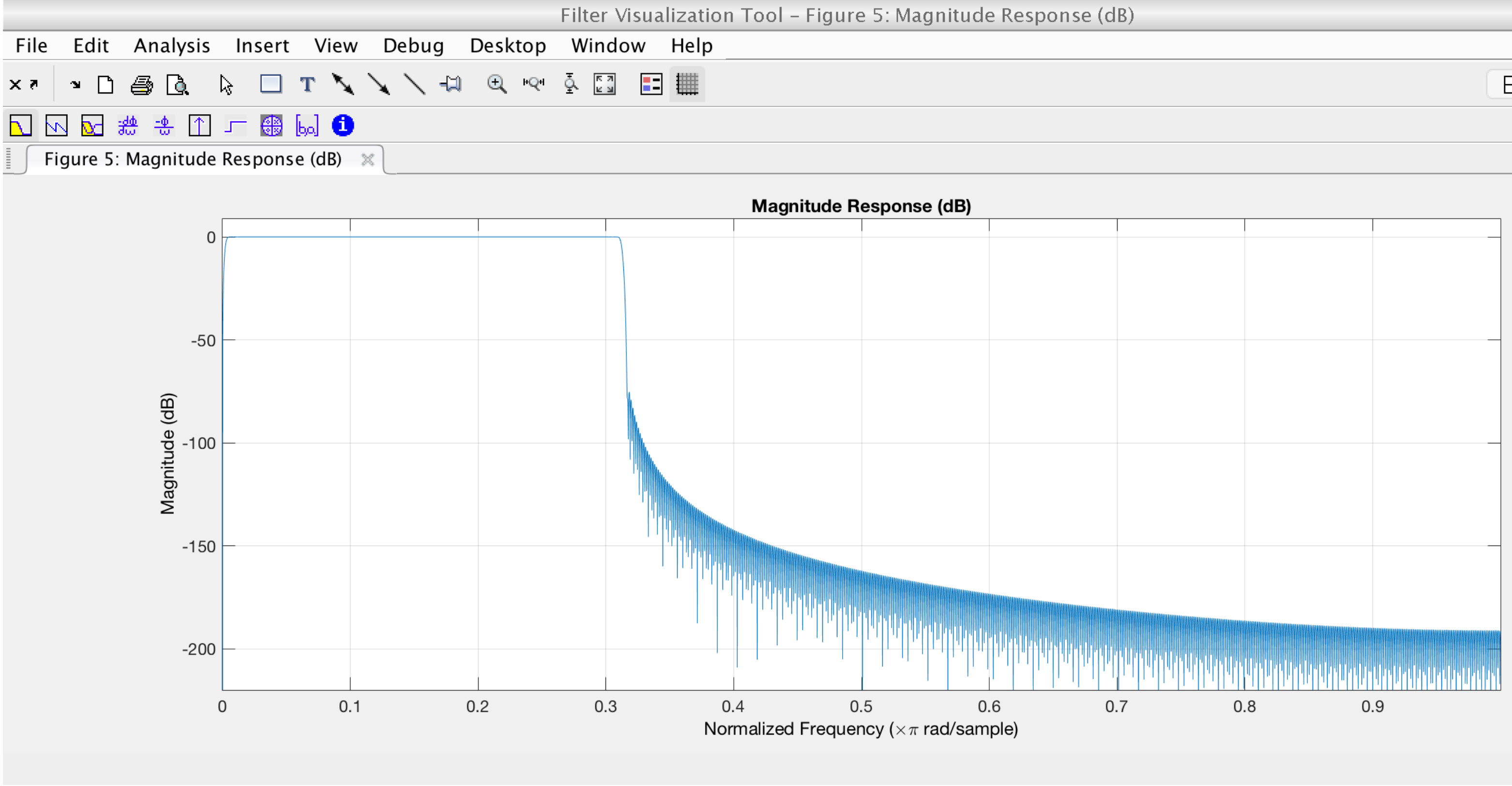
```
Filename: ...VR/ExpData/s07C/s07C-rs.set
Channels per frame       72
Frames per epoch         572928
Epochs                   1
Events                   223
Sampling rate (Hz)       512
Epoch start (sec)         0.000
Epoch end (sec)          1118.998
Reference                unknown
Channel locations        Yes
ICA weights              No
Dataset size (Mb)        179
```

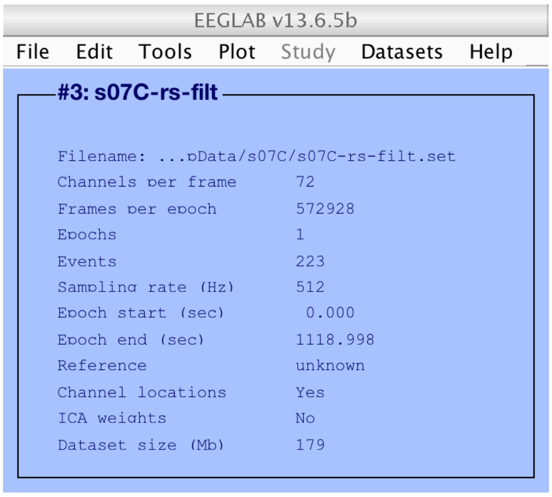
## APPLY BAND-PASS FILTER BETWEEN THE LOWER AND UPPER LIMITS SPECIFIED IN PARAMETERS FILE.

It applies a FIR windowed sinc filter using a blackman filter. The filter frequency response is plotted. The details of the filtering are written to subject information txt file.

```
    display('*********************************Bandpass filtering using a FIR windowed sinc filter***********************************')
    [M, dev]=pop_firwsord('blackman',SR, 2);
    [EEG,com,b]=pop_firws(EEG,'fcutoff',[f_low f_hi],'forder',M,'ftype','bandpass','wtype','blackman');
    fvtool(b);                                     % Visualise the filter characteristics
    fnom_filt=strcat(fnom_rs,'-filt');
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'setname',char(fnom_filt),'gui','off');   %save the resampled data as a newdata se
t.
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(fnom_filt),'filepath',Dirbase);
    eeglab redraw

    fprintf(fid,'Band-pass filtered %f2 - %dHz with %d-order fir windowed sinc filter (blackman).\n',f_low,f_hi,M);
```

```
*********************************Bandpass filtering using a FIR windowed sinc filter***********************************
pop_firws() - filtering the data
firfilt(): |====================| 100%, ETE 00:00
Creating a new ALLEEG dataset 3
Saving dataset...
```


Filter Visualization Tool - Figure 5: Magnitude Response (dB)
File
Edit
Analysis
Insert
View
Debug
Desktop
Window
Help
Figure 5: Magnitude Response (dB)
Magnitude Response (dB)
0
-50
-100
-150
-200
Magnitude (dB)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)



#3: s07C-rs-filt

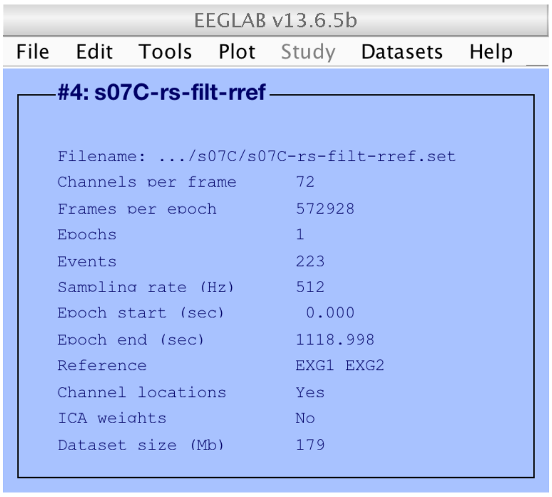
```
Filename: ...pData/s07C/s07C-rs-filt.set
Channels per frame       72
Frames per epoch         572928
Epochs                   1
Events                   223
Sampling rate (Hz)       512
Epoch start (sec)         0.000
Epoch end (sec)          1118.998
Reference                unknown
Channel locations        Yes
ICA weights              No
Dataset size (Mb)        179
```

## RE-REFERENCE THE DATA TO THE ELECTRODES SPECIFIED IN THE PARAMETERS FILE.

The channels used for referencing are generally EXG1 and EXG2, channels 65 and 66, respectively. The details of the re-referencing are written to the information text file.

```
    R=num2str(Params.references{1,1});
    if length(R)==2
        refs=str2double(R);
    elseif length(R)==4
        refs =[str2double(R(1:2)) str2double(R(3:4))];
    end

    display('**********************Rereference to Defined Channel:  does zero potential exist?****************************')
    EEG =pop_reref(EEG, refs, 'method','standard','keepref','on');
    fnom_ref = strcat(fnom_filt,'-rref');
    EEG = eeg_checkset( EEG );
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'setname',char(fnom_ref),'gui','off');   %save the resampled data as a newdata set
.
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(fnom_ref),'filepath',Dirbase);
    EEG = eeg_checkset( EEG );
    eeglab redraw

    here = CURRENTSET;   % Mark the current set.

    fprintf(fid,'Rereferenced using channels %s and %s.\n',EEG.chanlocs(refs(1)).labels,EEG.chanlocs(refs(2)).labels);
    fprintf(fid,'--------------------------------------------------------------------------------\n');
```

```
**********************Rereference to Defined Channel:  does zero potential exist?****************************
Re-referencing data
Creating a new ALLEEG dataset 4
Saving dataset...
```



#4: s07C-rs-filt-rref

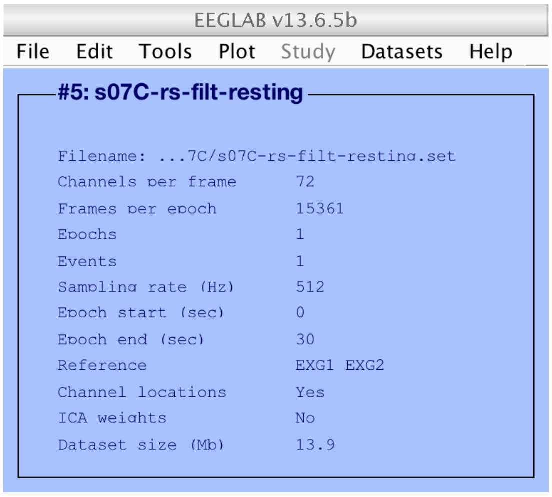
```
Filename: .../s07C/s07C-rs-filt-rref.set
Channels per frame       72
Frames per epoch         572928
Epochs                   1
Events                   223
Sampling rate (Hz)       512
Epoch start (sec)         0.000
Epoch end (sec)          1118.998
Reference                EXG1 EXG2
Channel locations        Yes
ICA weights              No
Dataset size (Mb)        179
```

## EXTRACT THE FIRST 30 SECONDS OF REST-STATE EEG FROM THE RESAMPLED, FILTERED AND RE-REFERENCED CONTINUOUS DATA.

The user needs to specify the upper and lower time limits for the resting state data to avoid include data with movement artifacts etc. The resting-state data is saved to a separate *.set file in the current subject-level folder.

```
    disp('----------------Extracting Resting State Interval-----------------------------');
    timex_lims = [10 40];  % Time limits in seconds.
    EEG = pop_select( EEG,'time',timex_lims );
    fnom_rest = strcat(fnom_filt,'-resting');
    EEG.setname = fnom_rest;
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'setname',char(fnom_rest),'gui','off');   %save the resampled data as a newdata se
t.
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(fnom_rest),'filepath',Dirbase);
    EEG = eeg_checkset( EEG );
    eeglab redraw;
```

```
----------------Extracting Resting State Interval----------------------------
eeg_insertbound(): 2 boundary (break) events added.
eeg_insertbound(): event latencies recomputed and 223 events removed.
eeg_checkset note: upper time limit (xmax) adjusted so (xmax-xmin)*srate+1 = number of frames
Creating a new ALLEEG dataset 5
Saving dataset...
```



#5: s07C-rs-filt-resting

```
Filename: ...7C/s07C-rs-filt-resting.set
Channels per frame      72
Frames per epoch        15361
Epochs                  1
Events                  1
Sampling rate (Hz)      512
Epoch start (sec)       0
Epoch end (sec)         30
Reference               EXG1 EXG2
Channel locations       Yes
ICA weights             No
Dataset size (Mb)       13.9
```

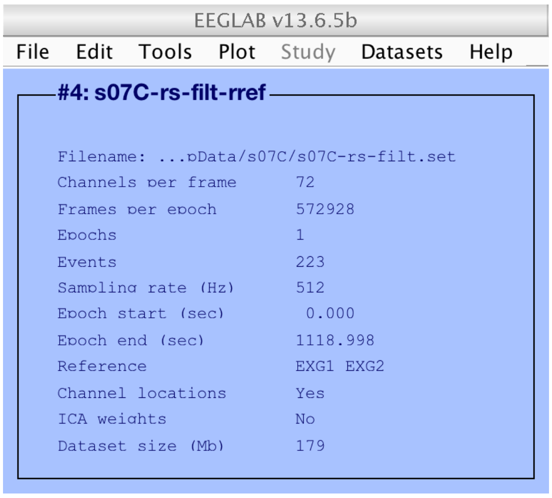
## DETECT POSSIBLE BAD ELECTRODES AUTOMATICALLY VIA EEGLAB KURTOSIS MEASURE.

Those electrodes with a kurtosis value >5 (z-score) are marked as bad. Bad electrodes electrodes detected with the measure are written to the subject information text file.

```
    % Retrieve the dataset before the resting state.
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'retrieve',here,'study',0);
    EEG = eeg_checkset( EEG );
    eeglab redraw

    chans = {EEG.chanlocs.labels};
    [EEG, eindx, measure,~] = pop_rejchan(EEG, 'elec',[1:64] ,'threshold',5,'norm','on','measure','kurt');
    EEG.reject.rejkurtE = eindx;                          %indices of suggested electrodes to reject according to kurtosis.

    if ~isempty(eindx)
        for cntr=1:length(eindx)
            if cntr==1
                fprintf(fid,'Bad electrodes according to kurtosis (threshold z-score 5):  %s  ',chans{eindx(cntr)});
            elseif cntr>1 && cntr<length(eindx)
                fprintf(fid,' %s  ',chans{eindx(cntr)});
            elseif cntr==length(eindx)
                fprintf(fid,' %s \n ',chans{eindx(cntr)});
            end

        end
    else
        fprintf(fid,'No bad electrodes marked according to kurtosis (threshold z-score 5)\n');
    end
```

```
Computing kurtosis for channels...
3 electrodes labeled for rejection
#       Elec.   Measure
1       Fp1     -0.40
2       AF7     -0.68
3       AF3     -0.84
4       F1      -0.89
5       F3      -0.87
6       F5      -0.80
7       F7      -0.62
8       FT7     1.83
9       FC5     56.88   *Bad*
10      FC3     -0.26
11      FC1     -0.67
12      C1      0.87
13      C3      0.36
14      C5      -0.04
15      T7      3.07
16      TP7     3.02
17      CP5     0.97
18      CP3     0.47
19      CP1     1.15
20      P1      7.16    *Bad*
21      P3      4.33
22      P5      1.38
23      P7      2.07
24      P9      4.36
25      PO7     1.89
26      PO3     6.31    *Bad*
27      O1      0.79
28      Iz      2.93
```

```
29      Oz      0.70
30      POz     -0.46
31      Pz      -0.22
32      CPz     0.16
33      Fpz     -0.50
34      Fp2     -0.33
35      AF8     -0.33
36      AF4     -0.98
37      AFz     -0.94
38      Fz      -1.08
39      F2      -1.12
40      F4      -1.13
41      F6      -1.13
42      F8      -1.21
43      FT8     -1.29
44      FC6     -0.55
45      FC4     -0.76
46      FC2     -0.67
47      FCz     -0.90
48      Cz      -0.10
49      C2      -0.32
50      C4      -0.48
51      C6      -0.79
52      T8      -0.66
53      TP8     1.25
54      CP6     -0.10
55      CP4     -0.06
56      CP2     -0.32
57      P2      -4.93
58      P4      -0.27
59      P6      -0.05
60      P8      0.48
61      P10     -1.11
62      PO8     -0.52
63      PO4     -0.42
64      O2      -0.21
```



#4: s07C-rs-filt-rref

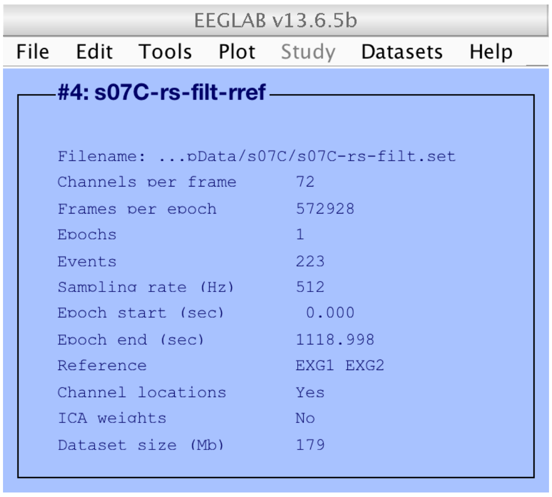
```
Filename: ...pData/s07C/s07C-rs-filt.set
Channels per frame      72
Frames per epoch        572928
Epochs                  1
Events                  223
Sampling rate (Hz)      512
Epoch start (sec)        0.000
Epoch end (sec)         1118.998
Reference               EXG1 EXG2
Channel locations       Yes
ICA weights             No
Dataset size (Mb)       179
```

## RUN THE PREP PIPELINE FUNCTION TO AUTOMATICALLY DETECT NOISY CHANNELS, findNoisyChannels()

Before running this function will need to take out the EXG channels that do not have X Y Z coordinates. This dataset is only used for the noisy channel detection script. This PREP function applies 4 different measures: 1. Robust standard deviation (unusually high or low amplitude) 2. Signal-to-Noise Ratio (SNR) (Christian Kothes, clean_channels() function). 3. Global correlation criteria (Nima Bigdely-Shamlo). 4. RANSAC correlation (but may not always be performed).

```
    EEG = pop_select( EEG,'nochannel',{'EXG1' 'EXG2' 'EXG3' 'EXG4' 'EXG5' 'EXG6' 'EXG7' 'EXG8'});
    EEG.setname = strcat(EEG.setname,'-remove-EXG');
    EEG = eeg_checkset( EEG );

    noisyOut = findNoisyChannels(EEG);   % Call of PREP pipeline function.

    badchan_indx = noisyOut.noisyChannels.all;
    badchans = {chans{[badchan_indx]}};

    fprintf(fid,'Bad channels according to PREP pipeline noisy channel detector:\n');
    for i2 = 1:length(badchans)
        fprintf(fid,'%s\t',badchans{1,i2});
    end

    % Retrieve the original correct dataset before suppression of EXG channels.
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'retrieve',here,'study',0);
    EEG = eeg_checkset( EEG );
    eeglab redraw
```

```
Removing 8 channel(s)...
-----Method 1: Unusually high or low amplitude (using robust std)---------------
-----Method 2: Compute the SNR (based on Christian Kothes clean_channels)
-----Method 3: Global correlation criteria (from Nima Bigdely-Shamlo)------------
-----Method 4: Ransac correlation (may not be performed)------------
```



#4: s07C-rs-filt-rref

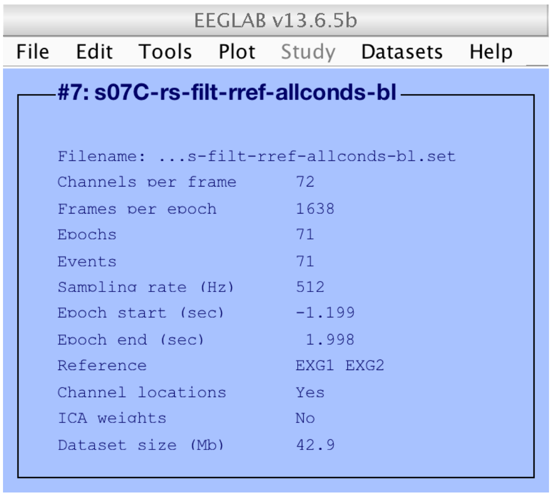
```
Filename: ...pData/s07C/s07C-rs-filt.set
Channels per frame       72
Frames per epoch         572928
Epochs                   1
Events                   223
Sampling rate (Hz)       512
Epoch start (sec)         0.000
Epoch end (sec)          1118.998
Reference                EXG1 EXG2
Channel locations        Yes
ICA weights              No
Dataset size (Mb)        179
```

## EPOCH THE CONTINUOUS DATA BUT INCLUDING ALL CONDITIONS.

Include all conditions (go and nogo) and all words. Epoching is carried out with the verb onset as the T0.

```
    dirsave = EEG.filepath;
    Enom = strcat(EEG.setname,'-allconds');
    Ecurr = {EEG.event.type};
    toexcl = {'right' 'left' 'go' 'nogo' 'faire-go' 'faire-nogo' 'faire-bad'};
    condindx = ~ismember(Ecurr,toexcl);

    EEG = pop_epoch( EEG, Ecurr(condindx), [str2double(Params.wind_low{1,1}) str2double(Params.wind_hi{1,1})], 'newname', char(Enom), 'epochinfo', 'y
es');
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'setname',char(Enom),'gui','off');
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(Enom),'filepath',dirsave);
    EEG = eeg_checkset( EEG );
    eeglab redraw;

    % CARRY OUT BASELINE CORRECTION
    disp('--------------------Baseline correction-----------------------------------');
    Enom_bl = strcat(Enom,'-bl');
    EEG = pop_rmbase( EEG, [Params.wind_low{1,1} 0]);
    [ALLEEG, EEG, CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,'setname',char(Enom_bl),'gui','off');
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(Enom_bl),'filepath',dirsave);
    EEG = eeg_checkset( EEG );
    eeglab redraw

    % Take out any triggers called 'faire';
    x1 = cell2mat(cellfun(@isempty,strfind({EEG.event.type},'faire-'),'UniformOutput',false));
    delindx = find(x1==0);
    EEG = pop_editeventvals(EEG,'delete',delindx);
    EEG = eeg_checkset( EEG );

    [ALLEEG, EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
    EEG = eeg_checkset( EEG );
    EEG = pop_saveset( EEG, 'filename',char(Enom_bl),'filepath',dirsave);
    eeglab redraw
```

```
pop_epoch():71 epochs selected
Epoching...
pop_epoch():71 epochs generated
pop_epoch(): checking epochs for data discontinuity
Creating a new ALLEEG dataset 6
Saving dataset...

--------------------Baseline correction-----------------------------------
pop_rmbase(): Removing baseline...
Creating a new ALLEEG dataset 7
Saving dataset...
```



#7: s07C-rs-filt-rref-allconds-bl

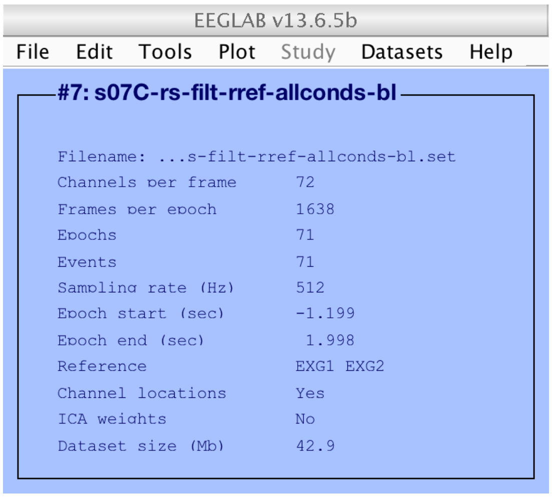
```
Filename: ...s-filt-rref-allconds-bl.set
Channels per frame      72
Frames per epoch        1638
Epochs                  71
Events                  71
Sampling rate (Hz)      512
Epoch start (sec)       -1.199
Epoch end (sec)          1.998
Reference               EXG1 EXG2
Channel locations       Yes
ICA weights             No
Dataset size (Mb)       42.9
```

## CALL OF FUNCTION TO ADD COVARIATE DATA (WORD FREQUENCY AND UP-PHON) TO THE EVENTS.

It is best if the covariate data is added to the epoched data, before cleaning.

```
[EEG] = EEGVR_addcovariates(EEG,Ecurr(condindx),xls_wordfreq);

EEG = pop_saveset( EEG, 'filename',char(Enom_bl),'filepath',dirsave);
EEG = eeg_checkset( EEG );
eeglab redraw;
```

```
Saving dataset...
```



#7: s07C-rs-filt-rref-allconds-bl

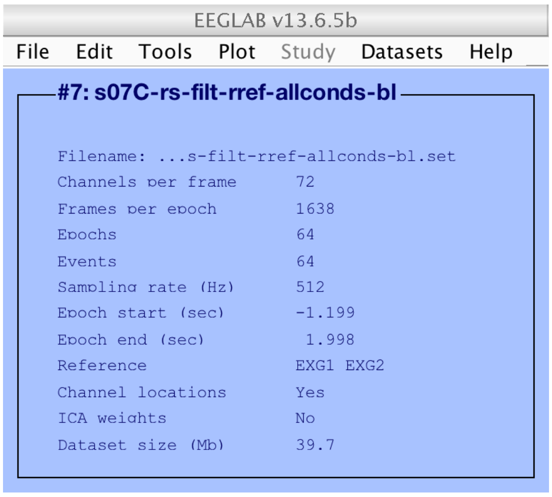
```
Filename: ...s-filt-rref-allconds-bl.set
Channels per frame      72
Frames per epoch        1638
Epochs                  71
Events                  71
Sampling rate (Hz)      512
Epoch start (sec)       -1.199
Epoch end (sec)          1.998
Reference               EXG1 EXG2
Channel locations       Yes
ICA weights             No
Dataset size (Mb)       42.9
```

## REMOVE INCORRECT TRIALS AUTOMATICALLY.

The variable : batrial_indx(). These trials are then removed automatically.

```
disp(horzcat('-----------Remove the ',num2str(length(badtrial_indx)),' incorrect trials----------------------'));
EEG.badtrialindx = badtrial_indx;
EEG = pop_select(EEG, 'notrial', badtrial_indx);
EEG = pop_saveset( EEG, 'filename',char(EEG.setname),'filepath',dirsave);
EEG = eeg_checkset( EEG );
eeglab redraw;
```

```
-----------Remove the 7 incorrect trials----------------------
Removing 7 trial(s)...
Pop_select: removing 7 unreferenced events
```



#7: s07C-rs-filt-rref-allconds-bl

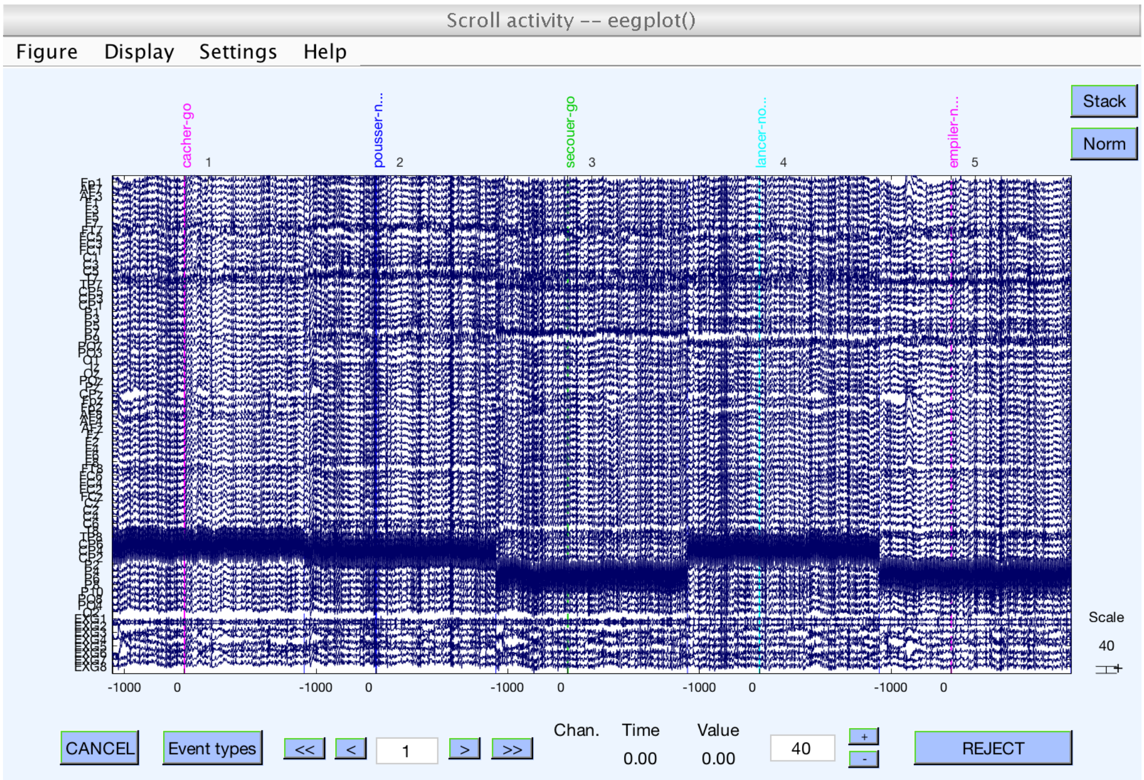
```
Filename: ...s-filt-rref-allconds-bl.set
Channels per frame       72
Frames per epoch         1638
Epochs                   64
Events                   64
Sampling rate (Hz)       512
Epoch start (sec)        -1.199
Epoch end (sec)           1.998
Reference                EXG1 EXG2
Channel locations        Yes
ICA weights              No
Dataset size (Mb)        39.7
```

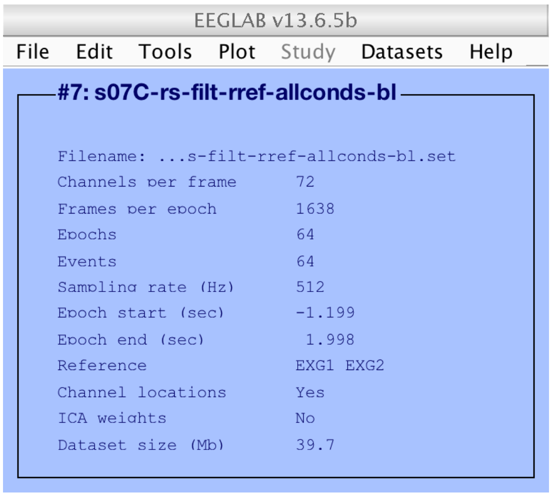
## AUTOMATIC BAD TRIAL DETECTION.

Applies a threshold of 75mV and a maximum of 10% rejection.

```
    fprintf(fid,'\n*******************Condition %s:\n %d trials before cleaning\n','Go-NoGo',size(EEG.data,3));
    [EEG,remepochs] = pop_autorej(EEG, 'nogui','on','threshold',75,'eegplot','on','maxrej',10);
    numrej = find(EEG.reject.rejauto);
    EEG.trialrejauto = remepochs;

    EEG = pop_saveset( EEG, 'filename',char(EEG.setname),'filepath',dirsave);
    EEG = eeg_checkset( EEG );
    eeglab redraw;

    fprintf(fid,'Number of trials marked for rejection (>75mV): %d\n', length(numrej));
    fprintf(fid,'Indx of trials marked for rejection (>75mV):\t');
    for cnt = 1:length(remepochs)
        fprintf(fid,'%d\t',remepochs(cnt))
    end
    fprintf(fid,'\n');
    fclose(fid);
```

```
Running auto-rejection protocol...
72 channel selected
64/64 trials marked for rejection
Computing joint probability for channels...
Computing all-channel probability...
3/64 trials marked for rejection
3 trials marked for rejection
3/64 trials rejected
Removing 3 trial(s)...
Pop_select: removing 3 unreferenced events
Computing joint probability for channels...
Computing all-channel probability...
2/61 trials marked for rejection
2 trials marked for rejection
2/61 trials rejected
Removing 2 trial(s)...
Pop_select: removing 2 unreferenced events
Computing joint probability for channels...
Computing all-channel probability...
0/59 trials marked for rejection
0 trials marked for rejection
0/59 trials rejected
Final kurotsis reject...
Computing kurtosis for channels...
Computing all-channel kurtosis...
3/59 trials marked for rejection
3 trials marked for rejection
Saving dataset...
```





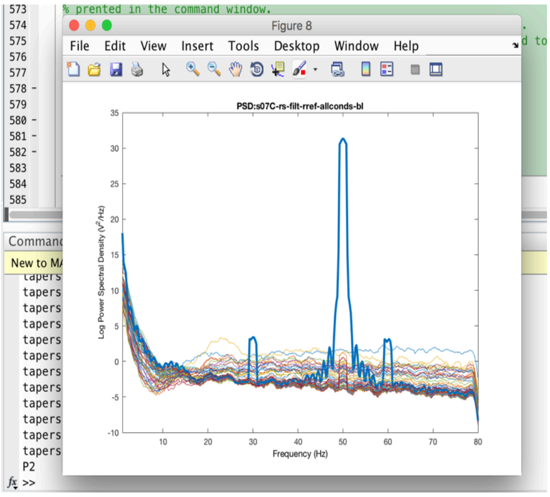
#7: s07C-rs-filt-rref-allconds-bl

```
Filename: ...s-filt-rref-allconds-bl.set
Channels per frame       72
Frames per epoch         1638
Epochs                   64
Events                   64
Sampling rate (Hz)       512
Epoch start (sec)        -1.199
Epoch end (sec)           1.998
Reference                EXG1 EXG2
Channel locations        Yes
ICA weights              No
Dataset size (Mb)        39.7
```

## CALCULATE THE SPECTRUM OF EACH ELECTRODE USING MULTI-TAPER (dpss).

Plot the mean spectrum of each electrode over all trials. The figure is interactive: if you select an electrode, its label will be prented in the command window. Looking at the spectra of the electrodes may detect noisy electrodes. The user can specify the upper and lower limits of the frequency band to analyses (here 1Hz to 120Hz). A copy of the *.fig file generated by the function is saved to the current subject's directory; this file is interactive -clicking on the spectrum of a single electrode on the figure presents the label of the corresponding electrode.'

```
CREx_SpectCalc_multitap(EEG,chans, [1 80]);

specnom = fullfile(EEG.filepath,strcat(EEG.setname(1:3),'-spect'));
saveas(gcf, specnom, 'fig')
close(gcf);
```


% prented in the command window.
Figure 8
File Edit View Insert Tools Desktop Window Help
PSD:s07C-rs-filt-rref-allconds-bl
Log Power Spectral Density (V²/Hz)
Frequency (Hz)
35
30
25
20
15
10
5
0
-5
-10
10
20
30
40
50
60
70
80
Command
New to M
tapers
P2
fx >>

## RUN FUNCTION TO LOCATE BAD EPOCHS AND CHANNELS VISUALLY.

```
    EpochChan_dlg(EEG);
```

```
Not removing any ica components for the moment!
Epoched data

Calculating robust standard deviation...
************************Calculating correlation of low frequency portion of EEG*******************************


Epoched data! Could take too long to calculate all the statistics for all channels and epochs.
************************************Calculation of kurtosis**********************************************
No PSD calculation of segmented data.


Creating a new ALLEEG dataset 1
```

*Published with MATLAB® R2016a*