Documentation for: icaeyeblinkmetrics()
Version 2.0

This EEGLAB toolbox is designed for automated/semi-automated selection of ICA components associated with eye-blink artifact using time-domain measures. The toolbox is based on the premises that 1) an ICA component associated with eye blinks should be more related to the recorded eye blink activity than other ICA components, and 2) removal of the ICA component associated with eye blinks should reduce the eye blink artifact present within the EEG following back projection.

Other than the EEG input, the only required input for the function is specification of the channel that exhibits the artifact (in most cases the VEOG electrode). This can either be stored within the EEG.data matrix or within EEG.skipchannels. It will then identify eye-blinks within the channel to be used for computation of the metrics listed below.

The toolbox does not change the data in any way, it only provides an output stored in 'EEG.icaquant' providing:

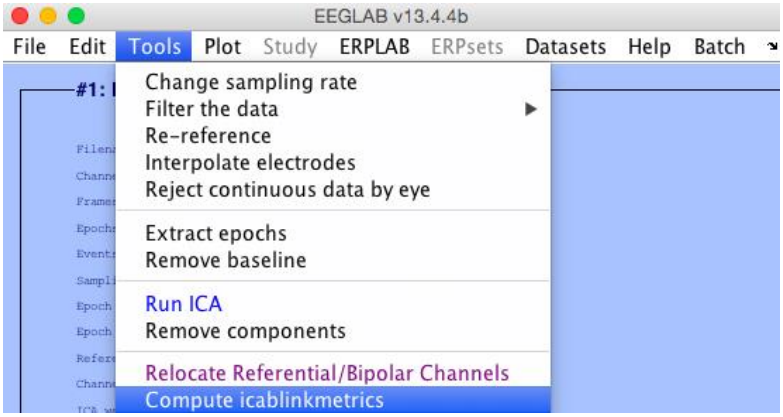
1. Metrics:
    a. The correlation between the measured artifact in the artifact channel and each ICA component. (i.e. how similar the ICA component is to the eye blink)
    b. The adjusted normalized convolution of the ICA component activity with the measured artifact in the artifact channel. (i.e., how well does the ICA component overlap with the eye blink)
    c. The percent reduction in the artifact present in the EEG for each ICA component if it was removed. (i.e., how much is the eye blink artifact reduced in the EEG when the component is removed)
2. Identified Components: the ICA components that exhibit statistically significant values for all three metrics. Alpha defaults to $p \leq 0.001$.
3. Artifact Latencies: the latencies that were used in the computation of the metrics.

Example Implementation Using the Tools menu of EEGLAB

Step 1: Open a continuous EEG dataset that has had ICA weights computed (i.e., EEG.icaweights should not be empty).

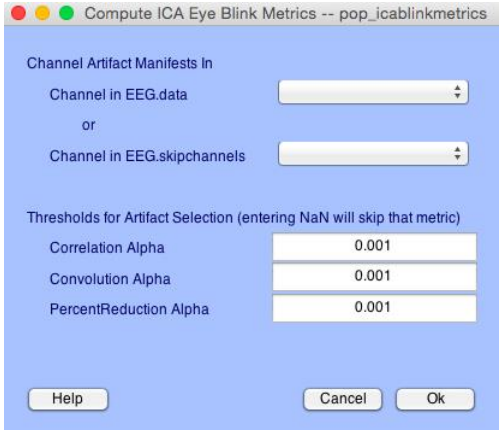
Step 2: Using the EEGLAB Graphical User Interface, click on the Tools menu.

Then select 'Compute icablinkmetrics'



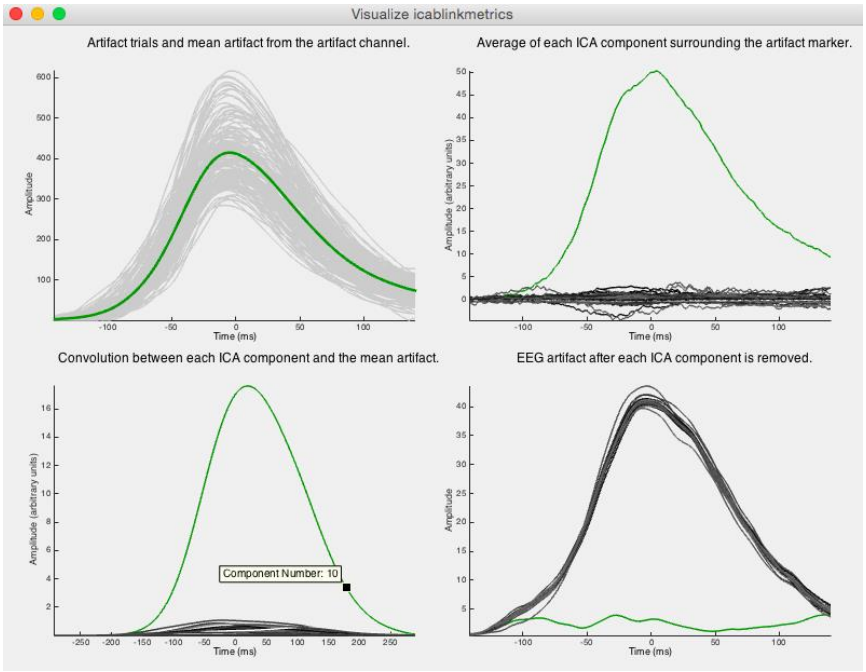
Step 3: A separate Graphical User Interface will pop up asking you to select the channel where the eye blink artifact manifests. This is typically your VEOG channel, but any channel where the eye blink artifact clearly presents should work (i.e., FP1). If the channel is included in the EEG.data, select from that option box. If the data has been relocated to EEG.skipchannels, there is a separate option box to select from.

Step 4: Specify the Alpha criterion necessary for selection of a component. The default is $p \leq 0.001$ which is applied globally, but the criterion can be applied differentially for each metric. Only those components which exhibit statistically significant values below the criterion will be identified.



Step 5: After the function has identified eye blink artifacts in the specified channel and then computed the metrics, it will display time-series plots showing the mean and individual trials for the eye blinks (upper left); the mean of each ICA component in the period surrounding the eye blinks (upper right); the convolution between each mean ICA component and the mean artifact (lower left); and the mean EEG data in the period surrounding the eye blinks when each component is removed (lower right). Clicking on a line will provide a tooltip displaying the component number.

Step 6: The Command Window will also display the identified component (plotted in the figure in green) and the location where the blink metrics were stored in the EEG structure.



Example Implementation Code Using Command Window

```
clear; clc; % Clear memory and the command window
eeglab; % Start eeglab: Requires Matlab Signal Processing Toolbox & Statistics Toolbox


filelocation = '/Studies/'; % Establish path to where the SET file is located


%% Prepare Data for ICA computation
% Read in a continuous EEG dataset with any bad channels/segments removed
EEG = pop_loadset('filename','CleanRawEEG.set','filepath',filelocation); EEG = eeg_checkset(EEG);

% Remove additional data points 2 seconds outside of triggers
winPadding = 2; winStart = (EEG.event(1).latency-(EEG.srate*((1000/EEG.srate)*winPadding)));
winStop = (EEG.event(end).latency+(EEG.srate*((1000/EEG.srate)*winPadding)));
if (winStart < 1); winStart = 1; end; if (winStop > EEG.pnts); winStop = EEG.pnts; end;
EEG = pop_select( EEG, 'point', [winStart, winStop]); EEG = eeg_checkset(EEG);

% HighPass Filter the data to remove slow drift
EEG = pop_basicfilter(EEG,1:size(EEG.chanlocs,2),'Filter','highpass',...
  'Design','butter','Cutoff',0.05,'Order',2,'Boundary',87);

% Relocate referential/bipolar channels so they can be recovered later if necessary
EEG = movechannels(EEG,'Location','skipchannels','Direction','Remove','Channels',{'M1','M2','HEOG'});

% Compute ICA ('icatype','binica' is faster if available; block size changed to be more similar to binica)
EEG = pop_runica(EEG,'icatype','runica',...
    'options',{'extended',1,'block',floor(sqrt(EEG.pnts/3)),'anneal',0.98}); EEG = eeg_checkset(EEG);

% Save Post ICA file
pop_saveset(EEG,'filename','PostICA.set','filepath',filelocation,'savemode','onefile');


%% icablinkmetrics() Process
% Read in data
EEG = pop_loadset('filename','PostICA.set','filepath', filelocation); EEG = eeg_checkset(EEG);

% Compute Blink Metrics (default alpha level is p <= 0.001)
EEG.icaquant = icablinkmetrics(EEG,...
    'ArtifactChannel',EEG.data(find(strcmp({EEG.chanlocs.labels},'VEOG')),:),...
    'Alpha',0.001,'VisualizeData','True');


%% Display reminders for where the data are stored and sort metrics
disp('ICA Metrics are located in:  EEG.icaquant.metrics')
disp('Selected ICA component(s) are located in:  EEG.icaquant.identifiedcomponents')


% Remove Artifact ICA component(s)
EEG = pop_subcomp(EEG,EEG.icaquant.identifiedcomponents,0);
```