An automatic algorithm for blink-artifact suppression based on iterative template matching: Application to single channel recording of cortical auditory evoked potentials [Supporting material]

Joaquin T. Valderrama[1,2,3,*], Angel de la Torre[4], Bram Van Dun[1,2]

[1] *National Acoustic Laboratories, NSW, Australia*
[2] *HEARing Co-operative Research Centre, Australia*
[3] *Department of Linguistics, Macquarie University, NSW, Australia*
[4] *Department of Signal Processing, Telematics and Communications, University of Granada, Granada, Spain*
[*] *Corresponding author:* `joaquin.valderrama@nal.gov.au`, `joaquin.valderrama@mq.edu.au`

Contents

A ITMS Matlab routine

This section presents a Matlab (The Mathworks, Inc., Natick, MA) routine that implements the Iterative Template Matching and Suppression (ITMS) algorithm. The input parameters of this function are the raw electroencephalogram (EEG) $y(n)$, the predefined blink-artifact template $h_0(n)$, the number of iterations I, the sampling frequency f_s, and a $DEBUG$ flag (0|1) to plot details of the method. The sampling frequency must be defined in Hz, and the EEG and the predefined template as column vectors, i.e.

$$y = \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{pmatrix}, \qquad h_0 = \begin{pmatrix} h_0(0) \\ h_0(1) \\ \vdots \\ h_0(L-1) \end{pmatrix},$$

where N represents the total number of samples of the EEG and L is the number of samples of the blink-artifact waveform. The ITMS function returns (1) the enhanced EEG (in which the blink-artifacts are suppressed) $y_{enhanced}(n)$; (2) the blink-artifact waveform $h(n)$; (3) the blink-events m_k; (4) the blink-event amplitudes A_k; and (5) the blink-artifact model x_{blink}(n), returned as column vectors:

$$y_{enhanced} = \begin{pmatrix} y_{enhanced}(0) \\ y_{enhanced}(1) \\ \vdots \\ y_{enhanced}(N-1) \end{pmatrix}, \qquad h = \begin{pmatrix} h(0) \\ h(1) \\ \vdots \\ h(L-1) \end{pmatrix}, \qquad m_k = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_K \end{pmatrix},$$

$$A_k = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_K \end{pmatrix}, \qquad x_{blink} = \begin{pmatrix} x_{blink}(0) \\ x_{blink}(1) \\ \vdots \\ x_{blink}(N-1) \end{pmatrix},$$

where K is the number of detected blink-events. ITMS also returns the blink-to-EEG ratio as an unitary value in dB (SNR).

The file "ITMS.zip" is available for download from the online supplementary material section of this paper. This compressed file contains the ITMS Matlab function [ITMS.m], the initial blink template used in this study [h0.mat], and a script file that gives an example of executing the ITMS function [ITMS_Check.m] with a representative EEG, stored in [Data.mat]. The EEG $y(n)$ and the predefined template $h_0(n)$ must use the same sampling frequency, which in this study was $f_s = 1$ kHz.

ITMS requires an initial template to operate. ITMS can be implemented either with the provided blink-artifact template or with any other template with a similar waveform to the blink-artifact morphology expected in the test EEG. Appendix C presents a sensitivity study of the ITMS method to the initial template, showing that ITMS is stable, and that the selection of the initial template is not critical.

The Matlab function that implements the ITMS algorithm is presented below. This script was developed with the R2017a version, and it uses functions from the "Signal Processing" and "Statistics and Machine Learning" toolboxes.

```
function [y_enhanced,h,m,A,x_blink,SNR] = ITMS(y,h0,I,fs,DEBUG)
% [ITMS] Iterative Template Matching Suppression algorithm
% This Matlab/Octave function provides an estimation of the blink-artifact
% process (blink template, blink-events positions, blink-events amplitudes)
% from an input EEG, and removes the blink artifact from the EEG.
%
% INPUT:  [y]          EEG (column vector, N samples)
%         [h0]         Initial blink template (column vector, L samples)
%         [I]          Number of iterations (unitary value)
%         [fs]         Sampling frequency in Hz (unitary value in Hz)
%         [DEBUG]      flag for providing or not detailed info [1|0]
% OUTPUT: [y_enhanced] Enhanced EEG (column vector, N samples)
%         [h]          Normalized blink template (column vector, L samples)
%         [m]          Blink-event positions (in samples, column, K events)
%         [A]          Blink-event amplitudes (column, K events)
%         [x_blink]    Blink-artifact model (column vector, N samples)
%         [SNR]        Blink-to-EEG ratio (unitary value in dB)
%
% Please cite: Valderrama JT, de la Torre A, Van Dun B (2017). An
% automatic algorithm for blink-artifact suppression based on iterative
% template matching: Application to single channel recording of cortical
% auditory evoked potentials. Journal of Neural Engineering

% Method parameters
h = h0;                         % Template initialized with h0
L = length(h);                  % Template length in samples
Fade = round(0.2*fs);           % Fade-in & -out number of samples (0.2 sec)
[b_lpf,a_lpf] = butter(4,2*20/fs); % Filter coefficients: 20 Hz LPF for h

% Iterations
for i=1:I
    % Matched filter h(-n) applied to the EEG
    z = filter(flipud(h),1,y);
    % Amplitudes and positions of local maxima in matched filter output
    [Z_LM,idx_LM] = local_maxima_estimation(z);
    % Threshold estimation from distribution of local maxima
    T = threshold_estimation(Z_LM,DEBUG);
    if isempty(T)    % If no threshold is detected, function terminates
        y_enhanced=y; h=zeros(L,1); m=[]; A=[]; x_blink=zeros(size(y));
        SNR=[]; return;
    end
    % Blink-events positions
    m = blink_event_positions(Z_LM,idx_LM,T,L,length(y));
    % Blink-artifact template estimated from EEG and blink positions
    h = template_estimation(y,m,L,Fade,b_lpf,a_lpf,DEBUG);
end
% Estimation of Blink-event amplitudes
A = amplitude_estimation(z,m,L,h);

% Estimation of blink model and suppression of blink artifact from EEG
impulses = zeros(size(y));      % Initialization
impulses(m) = A;                % Blink-events (positions and amplitudes)
x_blink = filter(h,1,impulses); % Convolution with template: blink model
y_enhanced = y-x_blink;         % Blink-artifact supression

% SNR estimate
Norm2 = filter(ones(size(h)),1,y.*y);
z_Norm2 = z.*z./Norm2;
z_Norm_k2 = z_Norm2(m+L-1);
SNR = 10*log10(mean(z_Norm_k2./(1-z_Norm_k2)));
```

```
if DEBUG
    clf(figure(53))
    DELTA = max(x_blink)-min(x_blink);
    plot([y+2*DELTA x_blink+DELTA y_enhanced]);
    legend('raw EEG','blink artifact estimates','enhanced EEG')
    xlabel('Time (Samples)'); ylabel('Amplitude');
    title(sprintf('SNR = %.2f dB\n',SNR))
    fprintf('SNR = %.2f dB\n',SNR);
end
return;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LOCAL FUNCTIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Provides local maxima (amplitudes and positions)
function [X,idx_X] = local_maxima_estimation(x)
idx = (2:(length(x)-1))';                    % Samples index
cond = x(idx)>x(idx+1) & x(idx)>x(idx-1);   % Local maxima condition
X = x(cond);                                 % Amplitudes of local maxima
idx_X = idx(cond);                           % Positions of local maxima
return;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Provides the threshold between the distribution of local maxima
% corresponding to noise (narrow mode around 0) and the distribution
% corresponding to the blink events (wide mode with bigger mean). The
% probability density function is estimated with a Gaussian kernel.
% DEBUG is a flag for providing or not detailed information.
function T = threshold_estimation(Ampl,DEBUG)
BW = 0.15*iqr(Ampl);                       % Bandwidth of the Kernel
domain = linspace(min(Ampl)-5*BW,max(Ampl)+5*BW,1000); % Axis of amplitudes
dA = domain(2)-domain(1);                  % Step in the axis of amplitudes
Kernel = exp(-(-4.0*BW:dA:4.0*BW).^2/(2*BW*BW));  % Gaussian kernel
Kernel = Kernel/sum(Kernel);               % Kernel normalization
Histogram = histc(Ampl,domain);            % Amplitudes histogram
PDF = filtfilt(Kernel,1,Histogram);        % Kernel based prob. density funct
[pdf_max,idx_max]=max(PDF);       %#ok Mode around 0 corresponding to noise
idx = (2:(length(PDF)-1))';       % Search of local minima in pdf
cond = idx>idx_max & PDF(idx)<=PDF(idx-1) & PDF(idx)<PDF(idx+1);
if sum(cond)
    idx_T = min(idx(cond)); % First local minimum in pdf
    T = domain(idx_T);      % T: amplitude corresponding to this minimum
else
    T = [];                 % If no minimum is found, no threshold provided
    idx_T = [];
end
if DEBUG
    clf(figure(51))
    plot(domain,Histogram,'-b',domain,PDF,'.-r',T,PDF(idx_T),'*k')
    grid on;
    title('Threshold estimation')
    ylabel('Number of local maxima'); xlabel('Amplitude')
    legend('Histogram','Kernel-based pdf','Threshold')
end
return;
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Provides the blink event positions (threshold is applied and group delay
% [L-1], associated to matched filter, is compensated).
function m = blink_event_positions(Z_LM,idx_LM,T,L,N)
m = idx_LM(Z_LM>T);                   % Blink-event positions
m = m-(L-1);                          % Group delay compensation
m = m(m>0 & m<N-L);                   % Triggers check:
return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Estimates the blink artifact template from EEG (y) and the blink-event
% positions (m). This function also applies low-pass filtering, fade-in
% fade-out and normalization.
function h = template_estimation(y,m,L,Fade,B_lpf,A_lpf,DEBUG)
K = length(m);                        % Number of blink-events
H = zeros(L,K);                       % Matrix of sweeps
idx=1:L;                              % Index for the impulsive response h
for k=1:K
    idx1 = idx+m(k)-1;                % Index for current sweep
    Sweep = y(idx1);                  % Sweeps selection
    H(idx,k) = Sweep-mean(Sweep);     % Demeaning & storage into H matrix
end
if K>1
    % Template estimation using only amplitudes between percentile 25-75
    H = sort(H,2);            % Amplitudes are sorted for removing outlayers
    i0 = round(K*0.25); i1=round(K*0.75); % 1st/3rd quartiles
    H = H(:,i0:i1);                       % Amplitudes between perc. 25-75
    W = hamming(length(i0:i1));           % Hamming window for averaging
    W = W/sum(W);                         % Window normalization
    h = H*W;                              % Template: average with window
else
    h = H;                                % If K=1, h=H
end
h = filtfilt(B_lpf,A_lpf,h);                % Filtered template
h(1:Fade) = h(1:Fade).*(0:Fade-1)'/Fade;    % Fade-in
h(end:-1:end-Fade+1) = h(end:-1:end-Fade+1).*(0:Fade-1)'/Fade; % Fade-out
h = h/sqrt(sum(h.*h));                      % Normalization
if DEBUG
    fprintf('%d blinks detected\n',K)
    clf(figure(52))
    plot(h,'.-')
    xlabel('Time (samples)'); ylabel('Amplitude'); grid on
    title(sprintf('Blink-artifact template (%d blink events detected)',K));
end
return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Estimates the amplitudes from the local maxima
function A = amplitude_estimation(z,m,L,h)
Z = z(m+L-1);                               % Amplitudes estimates form z
K = length(m);                              % Number of blink-events
Rh = zeros(K,K);                            % B matrix initialization
Corr_h = xcorr(h);                          % Template autocorrelation
Corr_h = Corr_h(L:end);                     % 1-side autocorrelation
for k1=1:K
    for k2=1:K
        dt = round(abs(m(k2)-m(k1)));       % Time difference |m(k2)-m(k1)|
        if(dt<L), Rh(k1,k2) = Corr_h(dt+1); end
    end
end
A = Rh\Z;                                   % Amplitude estimates
```

B Predefined blink-artifact template $\mathbf{h_0}$

The ITMS algorithm requires a generic predefined blink-artifact template $h_0(n)$ $n = 0, \ldots, L-1$ (figure 1), where L is the total number of samples in the EEG, to estimate the blink-events in the first iteration. The estimate of the blink-events is necessary to obtain accurate blink-artifact templates in succeeding iterations ($h_1(n)$ to $h_I(n)$), thus $h_0(n)$ must represent the blink-artifact waveform typically found in analyzed EEGs. In this study, we built the generic $h_0(n)$ template by averaging the blink-artifact waveforms estimated in a set of 14 normal hearing subjects (7 males, 43.64 ± 7.30 yr). The subjects used to obtain the generic $h_0(n)$ template were different from those used in the rest of the study. The steps that we followed to obtain the $\mathbf{h_0}$ template are presented below:

1. From the set of 14 EEGs, we selected one EEG in which the blink-artifacts could be visually identified. The selected EEG was from subject 5. Figure 2.A shows a segment of this EEG.

2. We visually marked the local maxima of 10 representative blink-artifacts, and we selected a time window of 2 seconds around them (0.5 s pre- and 1.5 s post-local maxima). Figure 2.A shows in red asterisks the local maxima visually selected in three blink-events. The corresponding EEG segments are highlighted in purple.

3. The 10 selected EEG segments were averaged after compensating their group delay, estimated by the cross-correlation operation. These EEG segments are presented in figure 2.B. This figure also shows the 'raw average' of the 10 segments, and the 'processed average'. The 'processed average' is the result of demeaning, 20 Hz low-pass filtering (4^{th} order, Butterworth), and amplitude-normalizing the 'raw average' signal. A time window of 1.4 s (0.25 s pre- and 1.15 s post-local maximum) was selected, and a linear fade-in and fade-out of 200 ms was applied.

4. The 'processed average' signal was used as $h_0(n)$ in the ITMS method to estimate the blink-artifact waveform considering all blinks detected in the EEG. Figure 2.C shows the blink-artifact templates estimated in four iterations. This figure shows that (1) the 'processed average' (despite it was built with only 10 blinks) was efficient to obtain an accurate blink-artifact template; and (2) the blink-artifact waveform and the number of detected blinks converge after the first iteration.

5. The blink-artifact template was estimated in the remaining 13 subjects with the ITMS method, using the blink-artifact template obtained in the first subject as $h_0(n)$. Figure 2.D shows an example of the blink-artifact templates estimated in four iterations in a representative subject (subject 13). In this example, the template waveform and the number of detected blink-events converge from the second iteration. Figure 2.E shows the blink-artifact waveforms estimated in the 14 subjects, along with the total number of detected blink-events. This figure shows that the blink-artifact waveforms from all subjects present a similar pattern.

6. The average and amplitude-normalization of the blink-artifact waveforms estimated in the full set of subjects lead to the $h_0(n)$ template used in this study.

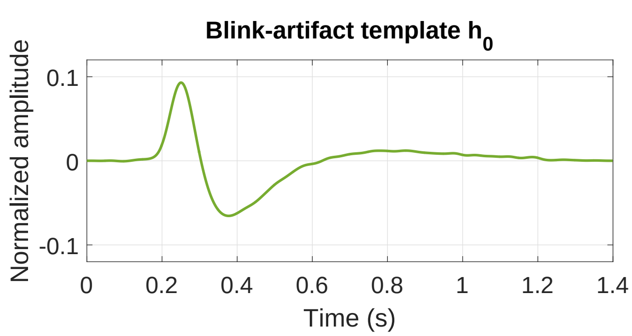


Figure 1: Predefined blink-artifact template $h_0(n)$.

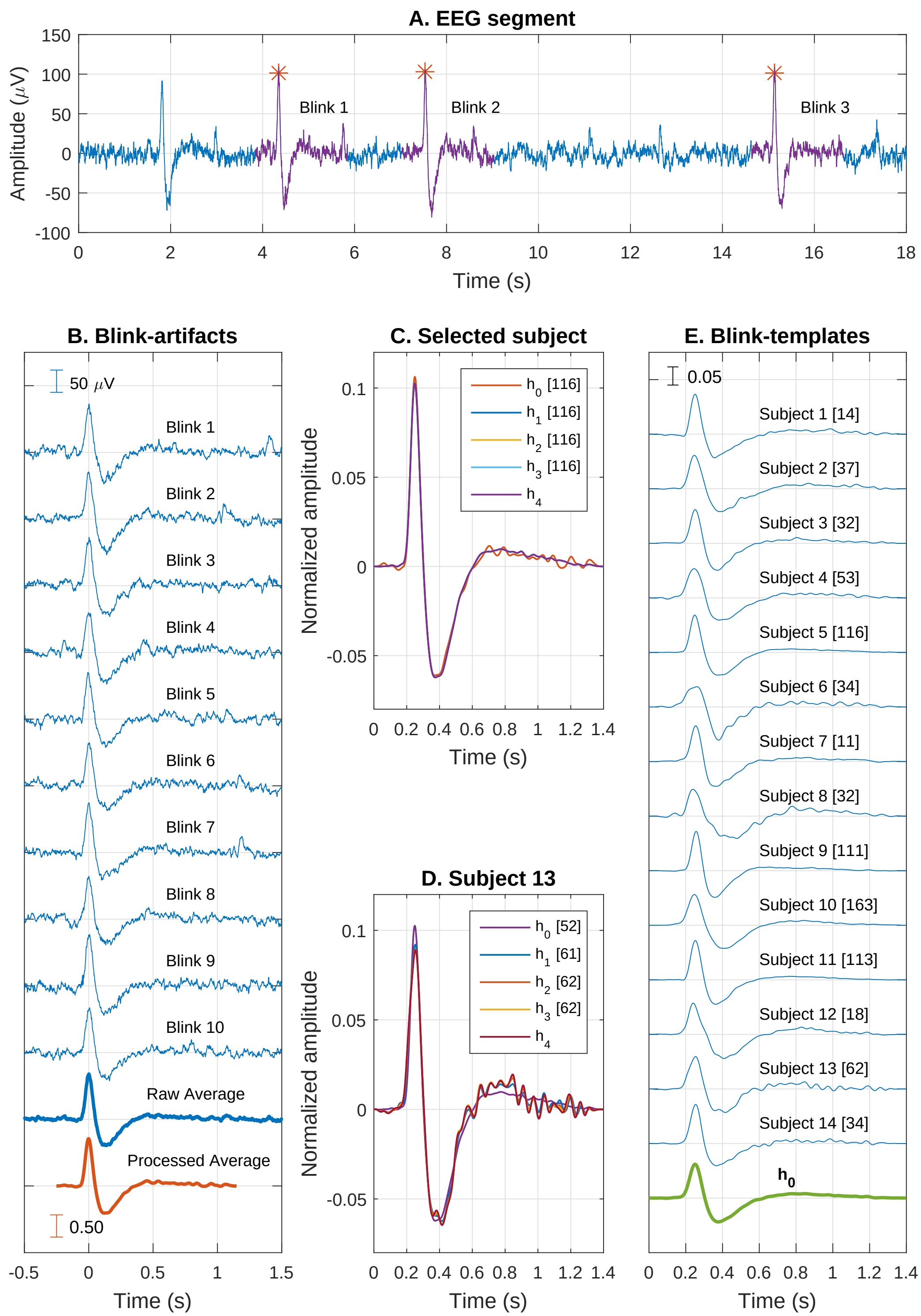


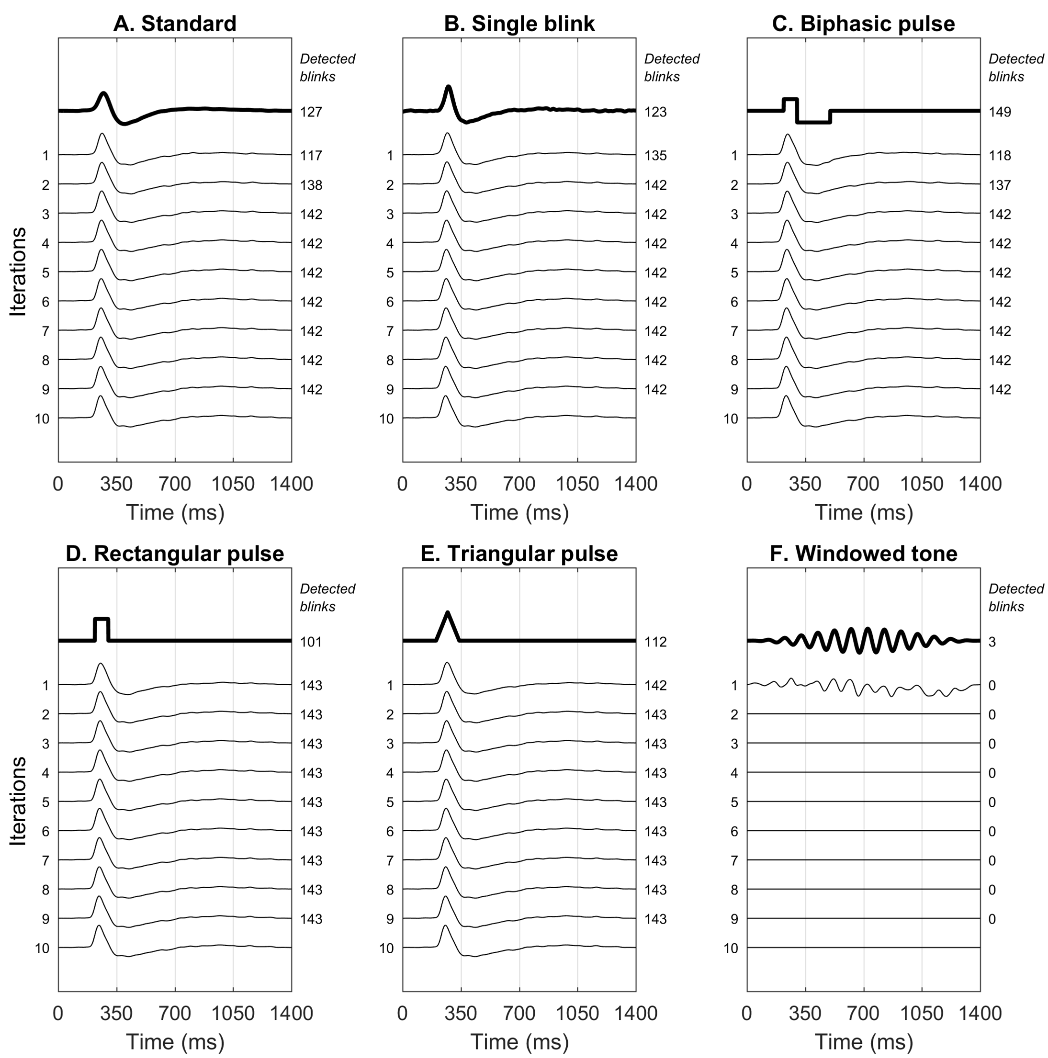
Figure 2: Process to obtain $h_0(n)$. (A) 18 s EEG segment from the selected subject (subject 5). Three visually identified blinks are marked with a red asterisk. (B) 10 blinks visually identified, raw average, and processed average. The blue errorbar represents the amplitude escale for the 10 blinks and the raw average. The orange errorbar is the normalized amplitude escale for the processed average. (C) Blink-artifact templates in four iterations using the processed average as $h_0(n)$. The number of detected blinks are shown in brackets. (D) Example (subject 13) of the blink-artifact template estimates obtained in four iterations using the blink-artifact template obtained in the selected subject (subject 5) as $h_0(n)$. (E) Blink-artifact templates obtained in 14 subjects (blue), and blink-artifact waveform used as predefined template in this study $h_0(n)$ (green). The errorbar shows the normalized amplitude escale.

C ITMS sensitivity to the initial blink-artifact template h_0

The ITMS algorithm requires an initial blink-artifact template (h_0) to operate. Appendix B describes in detail the process that we followed to obtain the initial template used in this study. Here we evaluate the performance of ITMS when initialized with different types of templates and different lengths. The conclusion of this analysis is that the selection of the initial template is not critical, provided that the duration and morphology of the initial template is similar to the blink-artifact waveform typically found in the EEG.

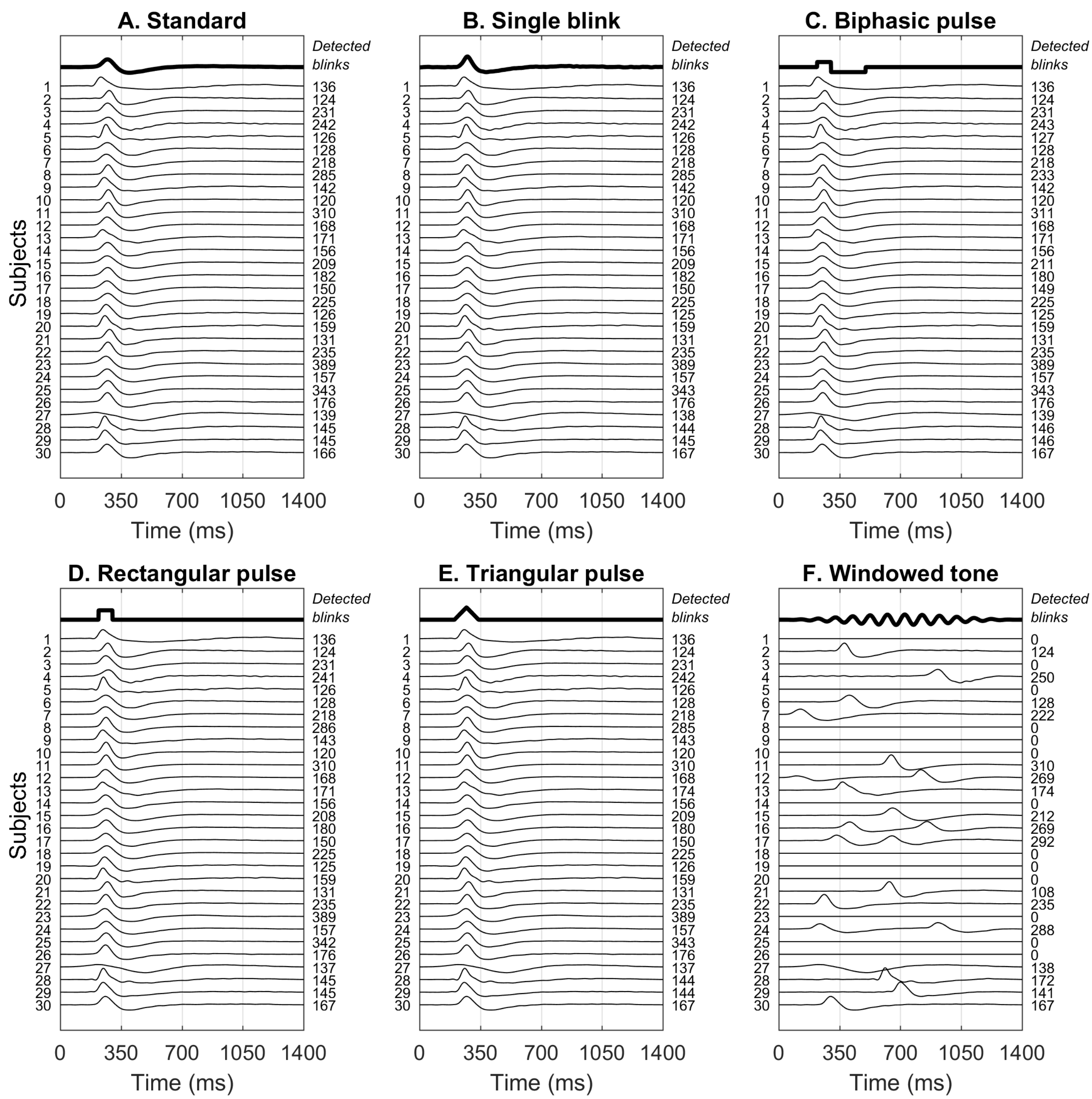
C.1 ITMS initialized with templates of different morphology [Analysis in 1 subject]

This figure shows the blink-artifact waveforms obtained with the ITMS method at the first 10 iterations in a representative subject (subject 9) when the algorithm was initialized with (A) the standard predefined blink-artifact template, described in appendix B; (B) a single blink-artifact obtained directly from an EEG segment in subject 11; (C) a synthesized biphasic rectangular pulse (80 ms in high, 200 ms in low); (D) a monophasic rectangular pulse (80 ms in high); (E) a triangular pulse of 140 ms; and (F) a Hann-windowed 10 Hz tone of 1400 ms. These initial templates are highlighted with black thick lines in the figure. This figure also shows the detected blink-events with the blink-artifact waveforms estimated in each iteration. This figure shows that the convergence of the ITMS method is stable in this subject (cases A to E), since different initial templates lead to a similar result after a few iterations. ITMS does not detect any blink-event in case F.



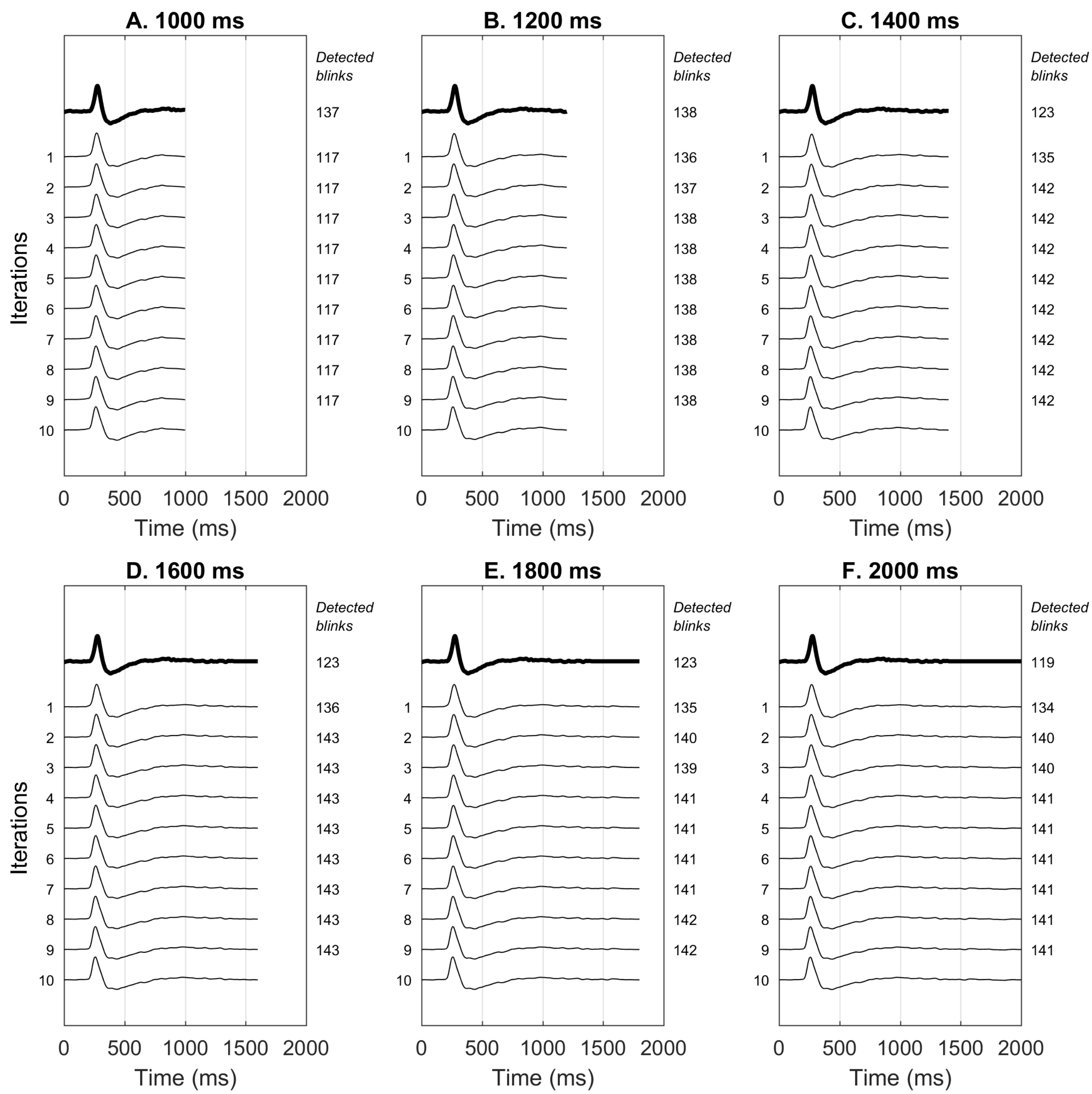
C.2 ITMS initialized with templates of different morphology [Analysis in 30 subjects]

The figure below shows the blink-artifact waveforms obtained in 30 normal hearing subjects (14 males, 44.73 ± 6.88 yr) with the ITMS method (10 iterations), initialized with the templates described in section C.1. This figure also shows the blink-events detected in each subject with each initial template. This analysis shows that the blink-artifact waveforms and the number of detected blink-events are very similar when the ITMS method is initialized with the standard template (case A), with a single blink-event (case B), with a synthesized biphasic rectangular pulse (case C), with a monophasic rectangular pulse (case D), and with a triangular pulse (case E). However, the results are not consistent if the morphology of the initial template does not match the expected blink-artifact waveform (case F). These results point out that the initial blink-artifact template h_0 is not critical, provided that the morphology of the initial template is similar to the blink-artifact waveform typically found in the EEG.



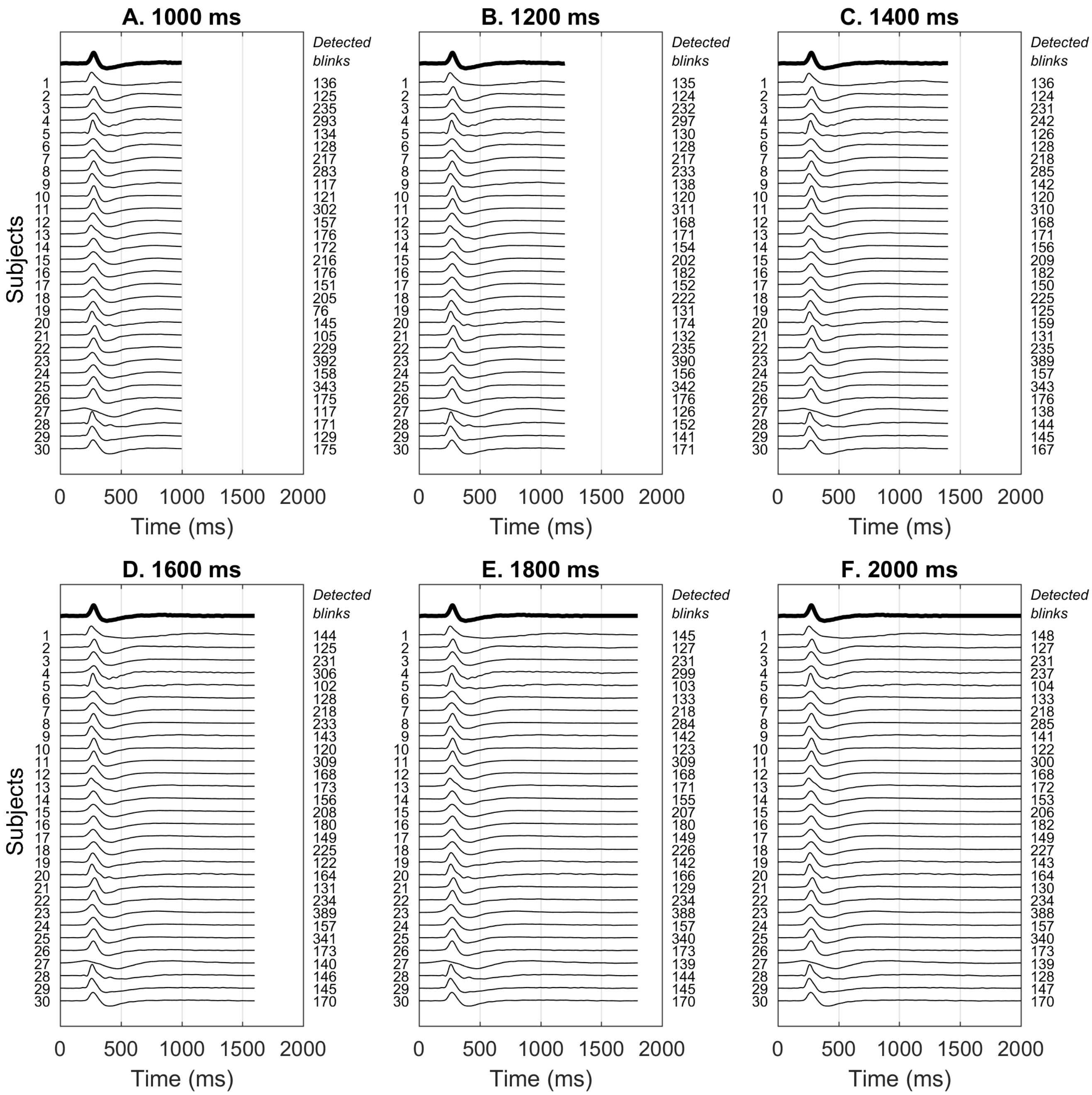
C.3 ITMS initialized with templates of different length [Analysis in 1 subject]

This figure shows the blink-artifact waveforms obtained with the ITMS method at the first 10 iterations in a representative subject (subject 9) when the algorithm was initialized with a template of different lengths. The initial template consisted of a single blink-artifact obtained directly from an EEG segment (subject 11). This figure shows that (a) convergence occurs in the first iterations, and (b) the blink-artifact waveforms and the number of detected blink-events are very similar in the cases B to F. In this subject, a duration of 1000 ms (case A) might be insufficient to completely characterize the blink-artifact waveform.



C.4 ITMS initialized with templates of different length [Analysis in 30 subjects]

This figure shows the blink-artifact waveforms obtained with 30 normal hearing subjects (14 males, 44.73 ± 6.88 yr) with the ITMS method (10 iterations), initialized with a blink-artifact template consisting of a single blink-artifact obtained directly from an EEG segment (subject 11) with different lengths. This analysis shows that the blink-artifact morphology and the number of detected blink-events is very similar in all cases, excepting for case A, in which the number of detected blink-events deviates significantly from the rest of the cases in some subjects. These results point out that ITMS is stable to the duration of the blink-artifact template, and that a duration of 1400 ms is appropriate to characterize the blink-artifact waveform.



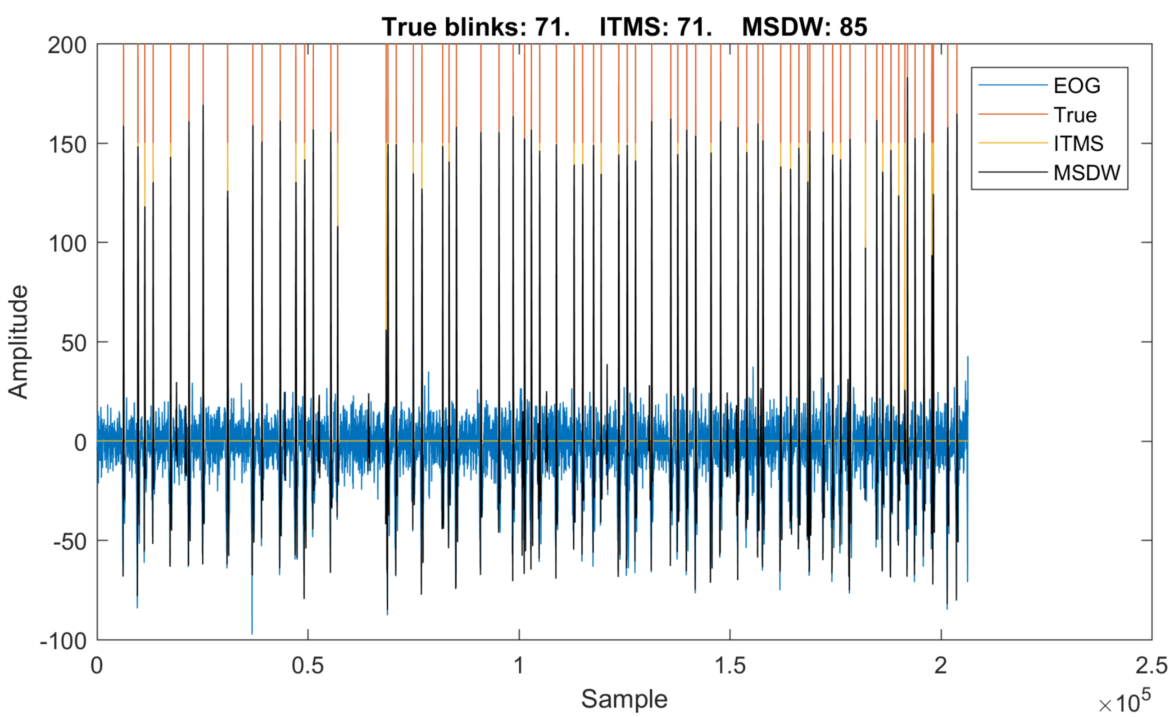
D Blink-artifact detection with a public database

The performance of ITMS and MSDW in detecting blink-events was evaluated with real data obtained from a public database (details in section 2.3 of the manuscript). The file “ROC_RealEEGs.zip” is available for download from the online supplementary material section of this paper. This compressed file contains a number of folders and Matlab scripts that can be used to replicate the results presented in this paper, and to evaluate the performance of the ITMS method in detecting blink-events with any other alternative technique that may arise in the future. Please follow these steps to use this supplementary material:

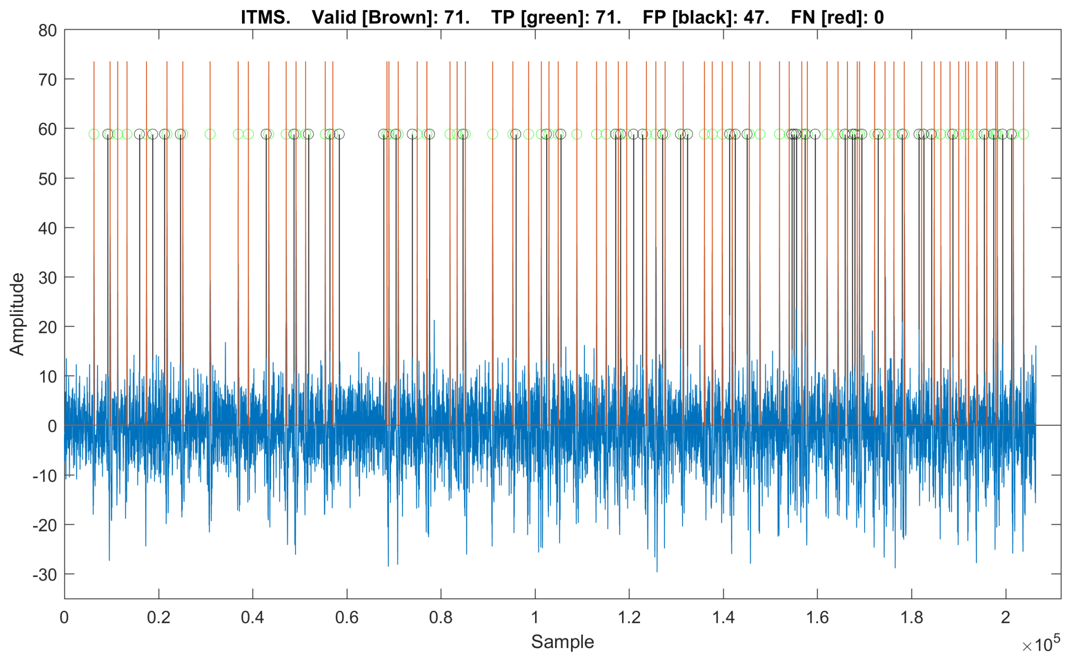
1. Download the file “ROC_RealEEGs.zip” from the online supplementary material section of this paper. Unzip the file and move the “ROC_RealEEGs” folder to the desired directory (to the desktop for example).

2. Download the public database from the URL: `https://sccn.ucsd.edu/~arno/fam2data/publicly_available_EEG_data.html` [4.31 GB]. Copy the files from each subject to the corresponding subject folder in the “OpenDatabase” folder.

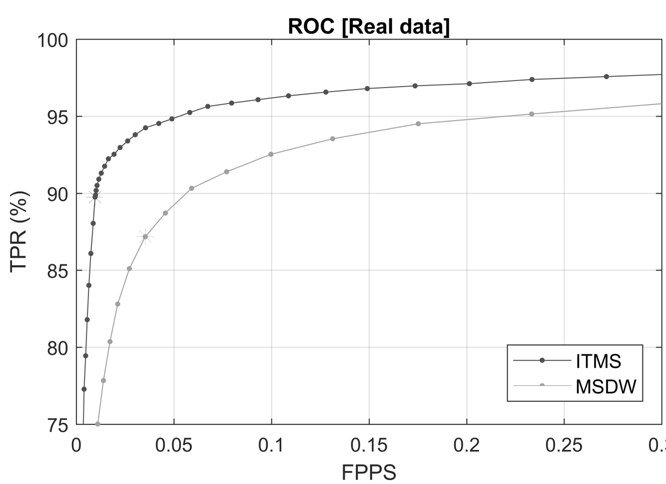
3. Run the script “S1_DetermineTrueBlinks.m”. This script goes over the full database and automatically determines the ‘true’ blink-events from the FP1 EEG channel according to the conditions specified in section 3.4.1 of the paper. In the ‘variables initialization’ section of the script, set the flag ‘DEBUG’ to 1 to plot relevant results and visually follow the procedure across files. When this flag is active, figure 1 plots the EEG (in blue), the blink-events detected by ITMS (in yellow), the blink-events detected by MSDW (in black), and the blink-events categorized as ‘true’ blink-events (in red). The figure below shows an example (subject ‘cba’, day 1, file 1). Additionally, figures 51, 52, and 53 plot details of the ITMS method. If the ‘DEBUG’ flag is on, it is necessary to press a key after the analysis of each file. Once all files have been analysed, the script stores the variable ‘m_Real.mat’ in the folder ‘Results’. This variable contains the ‘true’ blink-events.



4. Run the script “S2_BuildROC.m”. This script goes over the full database and evaluates the performance of the ITMS and MSDW methods in detecting blink-events from the Fz EEG channel, in terms of true positives (TP), false positives (FP), and false negatives (FN) at different thresholds.. Please be aware that **this script takes a long time to finish**. This script also has a ‘DEBUG’ flag that can be set to 1 to visualize relevant results. The figure below shows an example (subject ‘cba’, day 1, file 1) of the performance or ITMS for one specific threshold. This figure shows the raw EEG in blue. The ‘true’ blink-events (determined in the previous step) are shown in brown, in this particular case, this EEG contains 71 ’true’ blink-events. The blink-events detected by ITMS with the analysed threshold that match the ‘true’ blink-events are considered as true positives, and are plotted with green circles. The detected blink-events that do not match the ‘true’ blink-events are false positives, and are represented with black circles. The false negatives (red circles) represent the ‘true’ blink-events not detected by the method. At the end of the analysis, the script stores relevant variables under the file ‘ROC_Real.mat’ in the ‘Results’ folder.



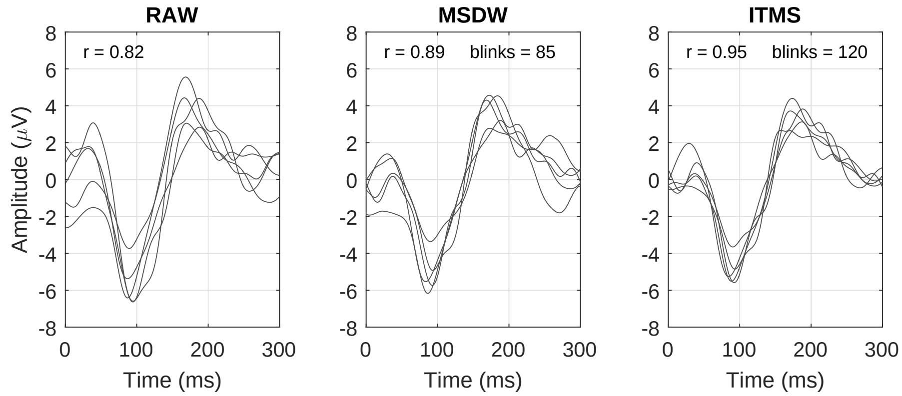
5. Finally, run the script “S3_PlotROC.m” to plot the ROC curve and store a printed figure in the ‘Results’ folder.



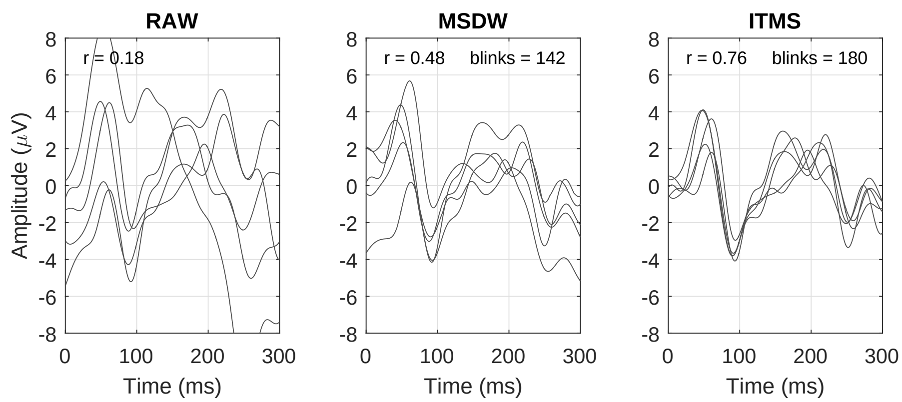
E Mean correlation coefficients in 30 subjects

The following figures show the 5 CAEP signals obtained in 30 subjects in each EEG scenario: unprocessed (RAW), and processed with the MSDW and ITMS methods. Each CAEP signal was obtained by averaging and demeaning 50 sweeps (EEG segments corresponding to the first 300 ms from stimulus onset). The CAEP signals are overlapping to show their degree of reproducibility. The number of detected blinks and the mean of the correlation coefficients (r) estimated between all possible combinations of the 5 CAEP signals taken two at a time (10 statistics) is shown at the top of the chart.

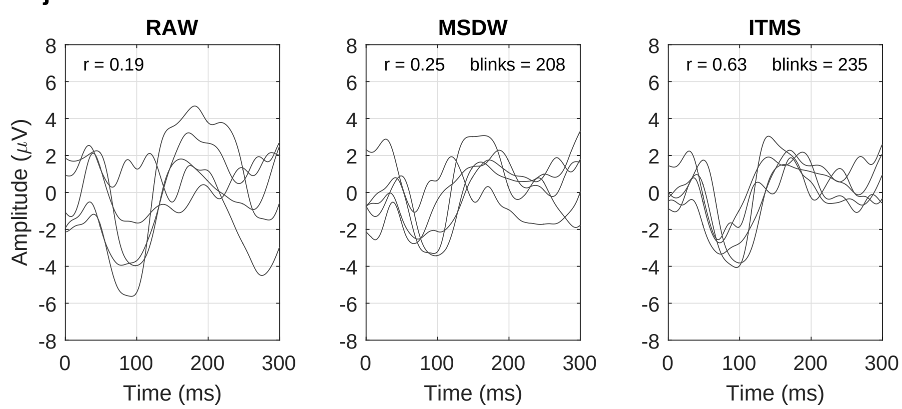
Subject 1



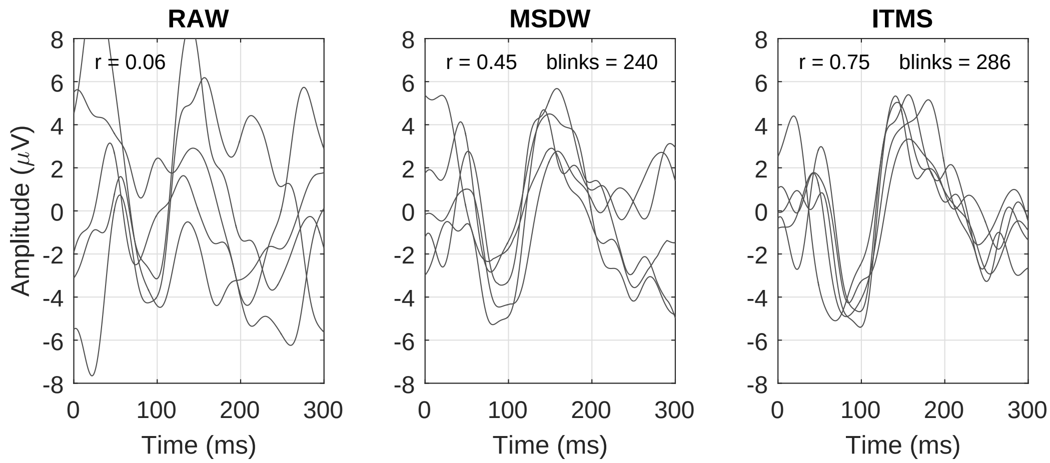
Subject 2



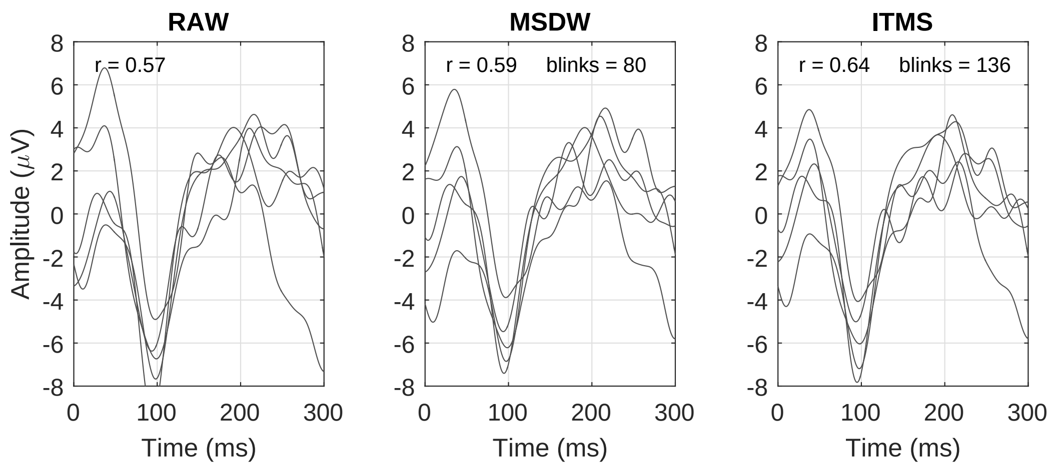
Subject 3



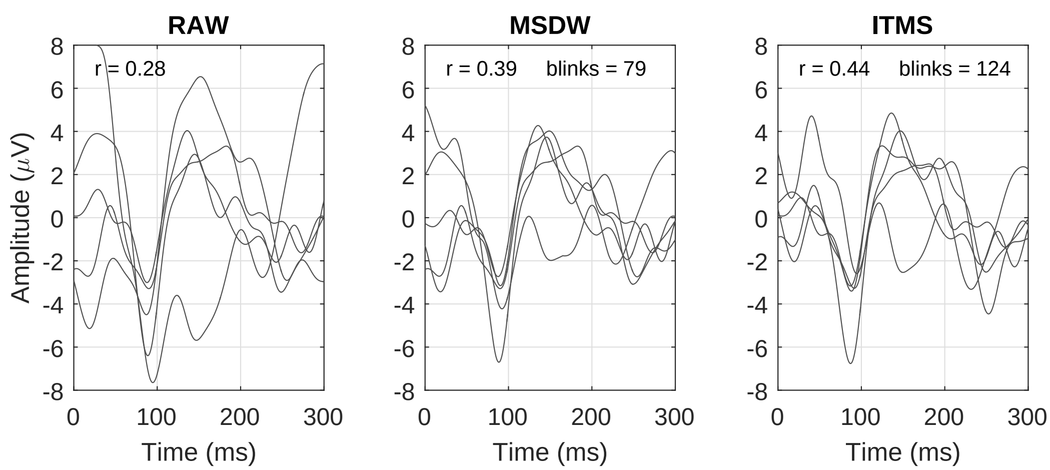
Subject 4



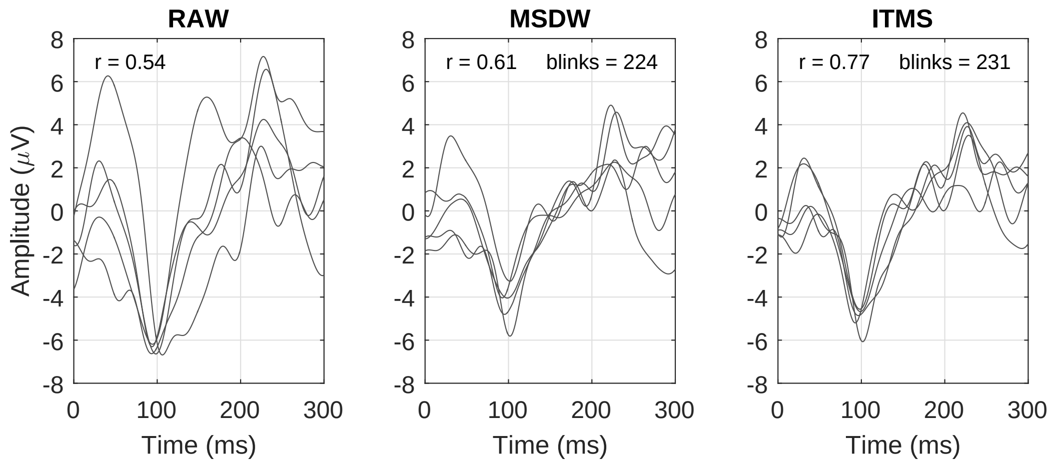
Subject 5



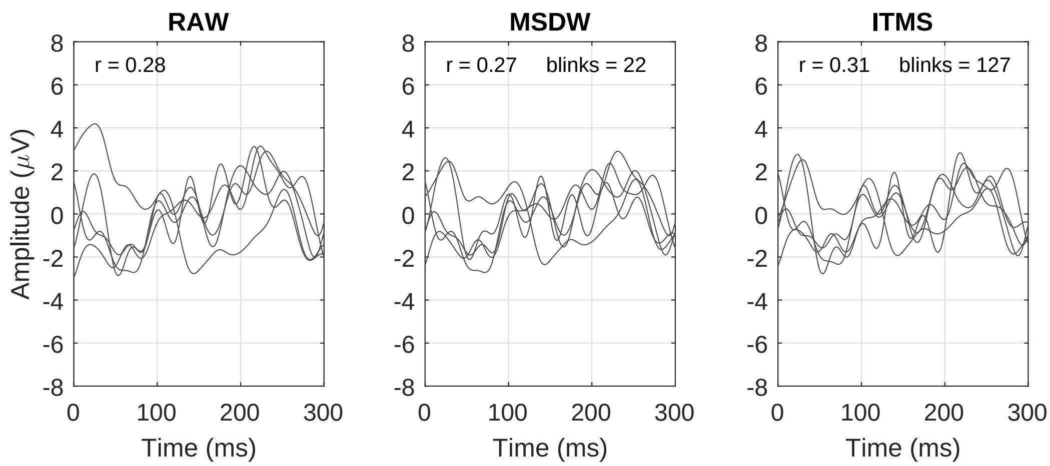
Subject 6



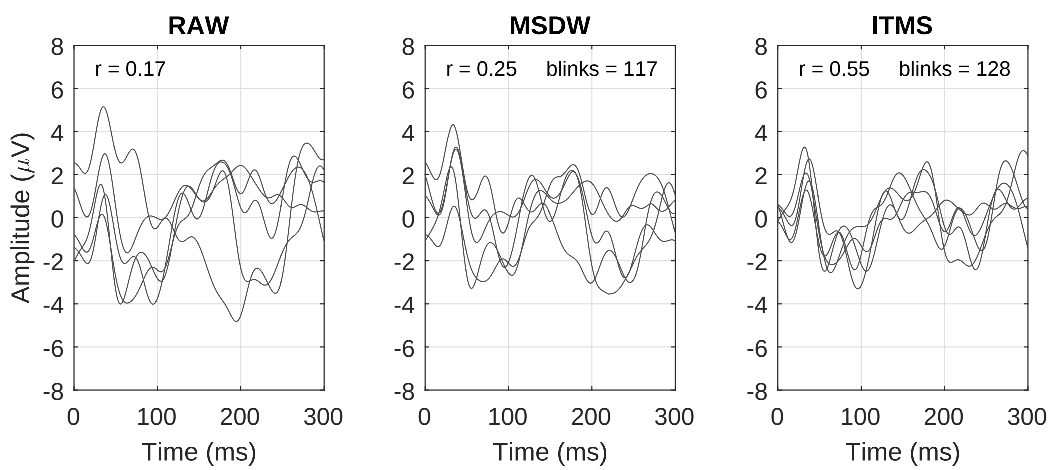
Subject 7



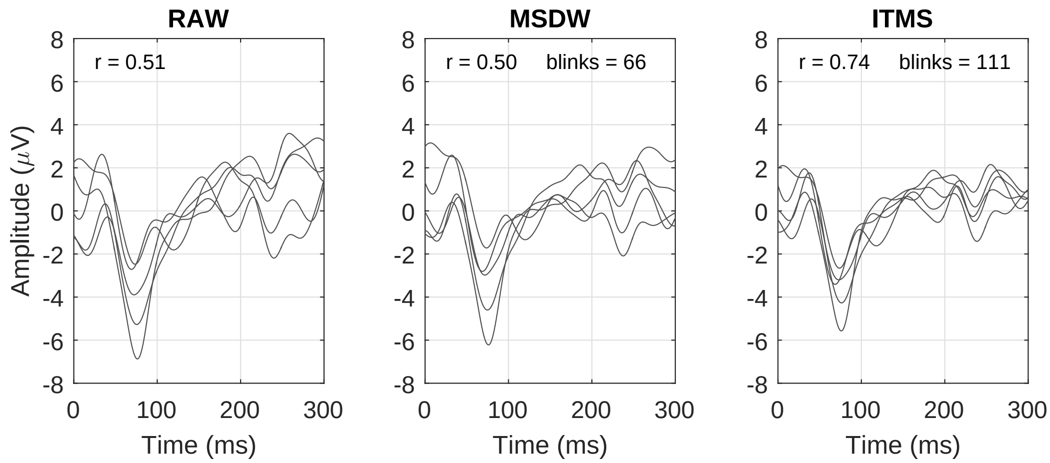
Subject 8



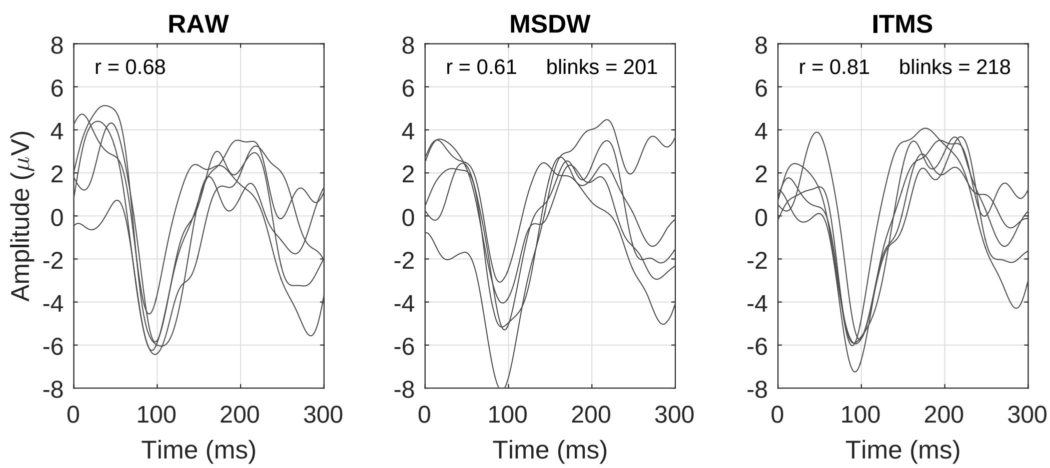
Subject 9



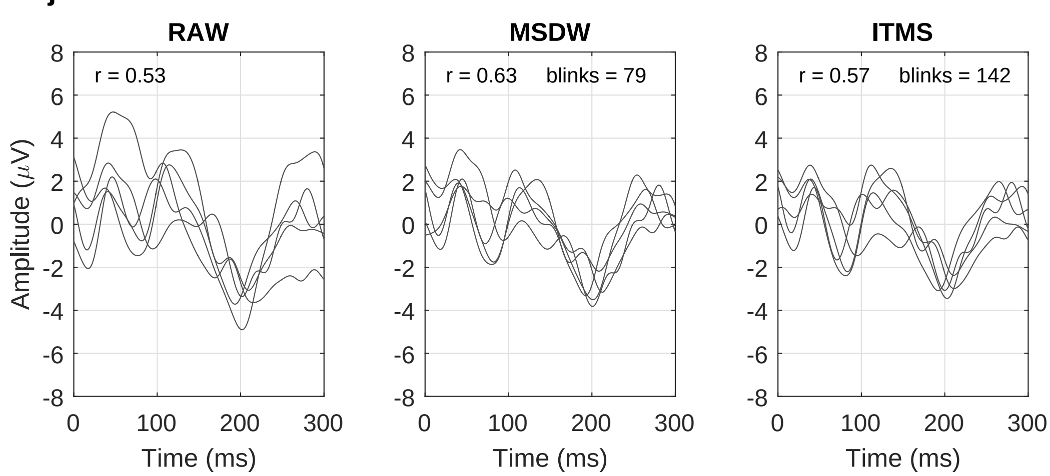
Subject 10



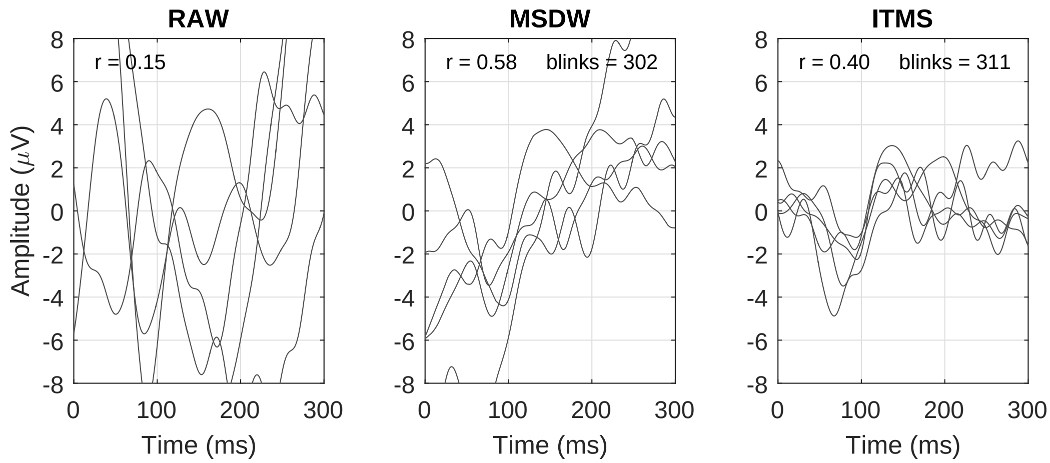
Subject 11



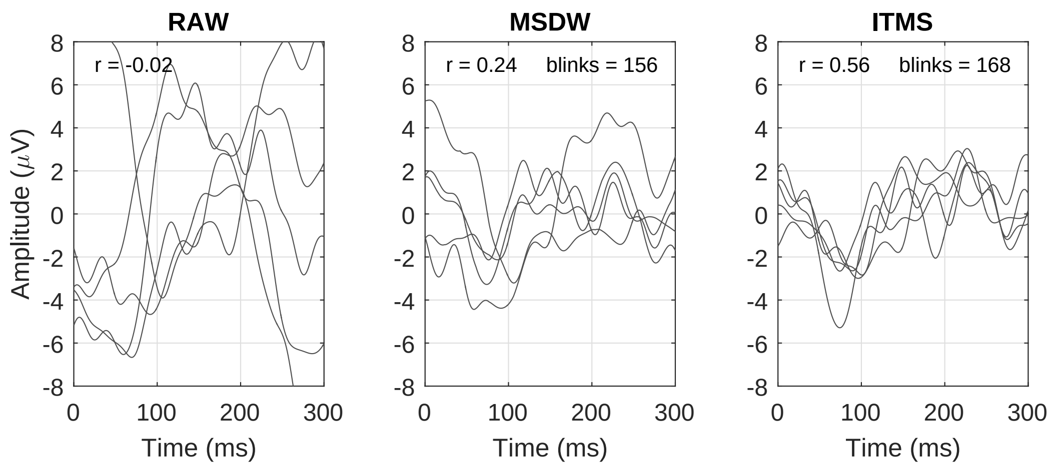
Subject 12



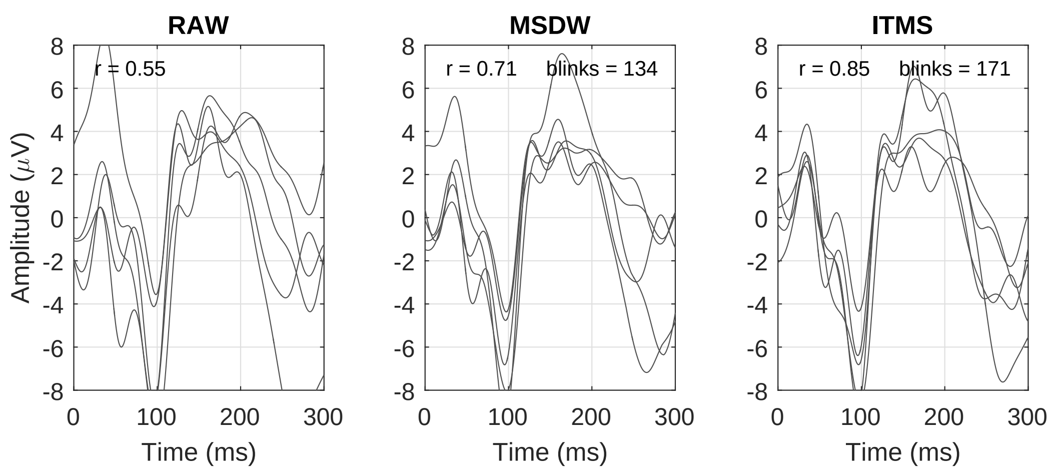
Subject 13



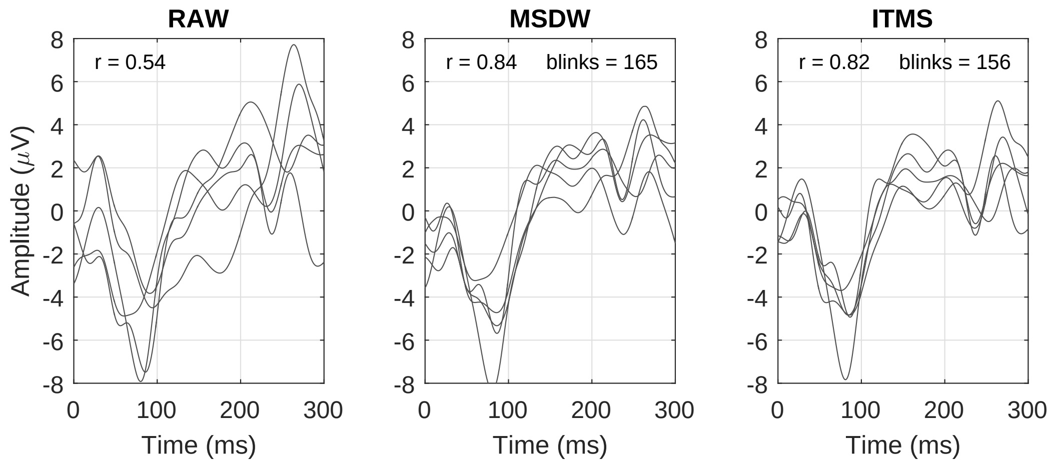
Subject 14



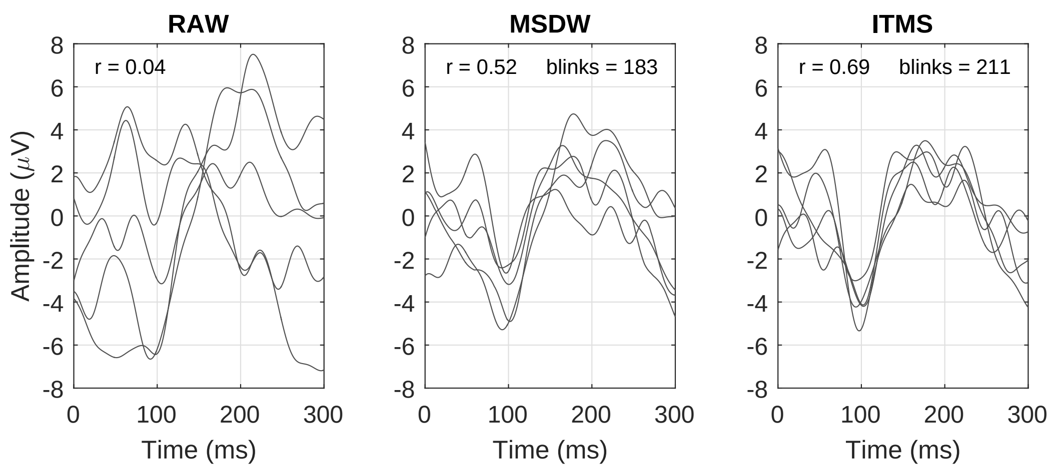
Subject 15



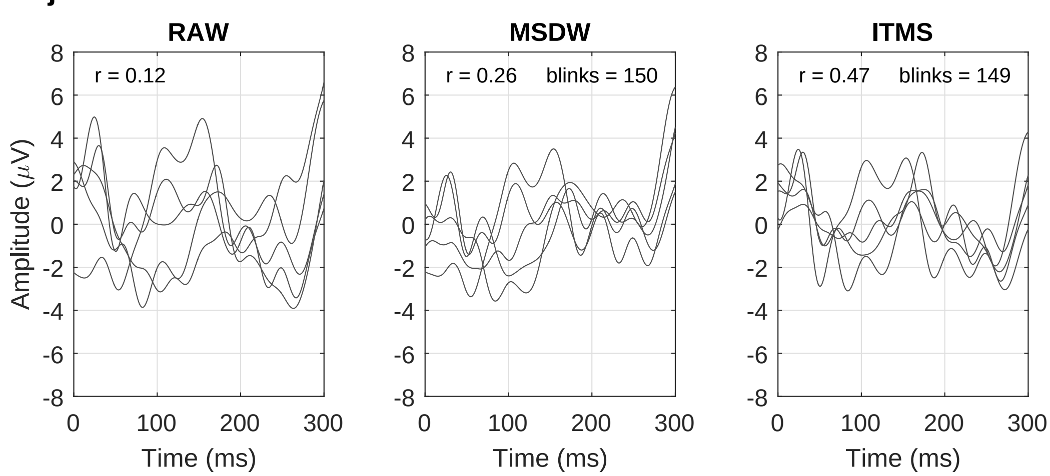
Subject 16



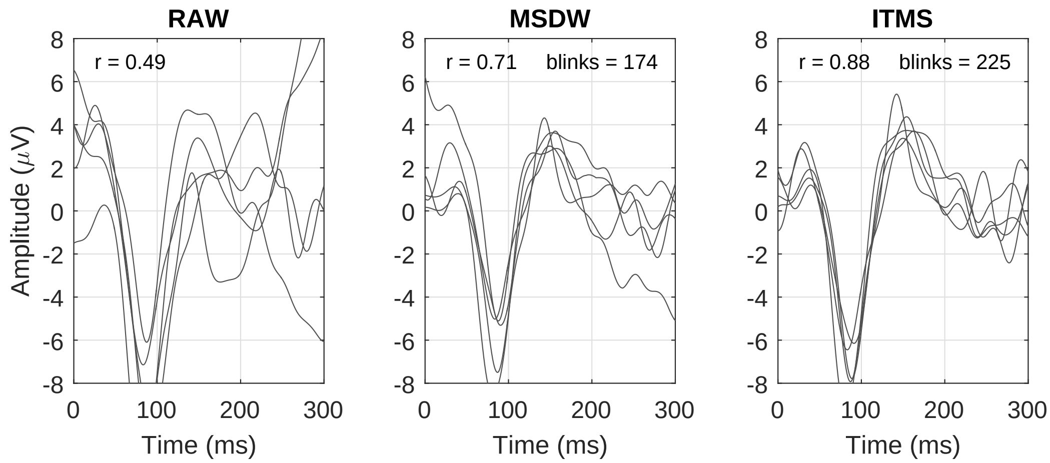
Subject 17



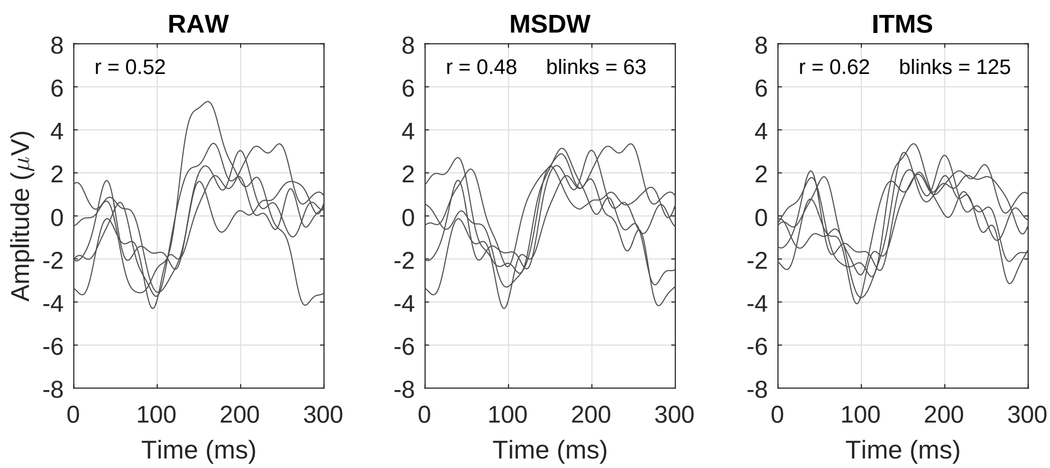
Subject 18



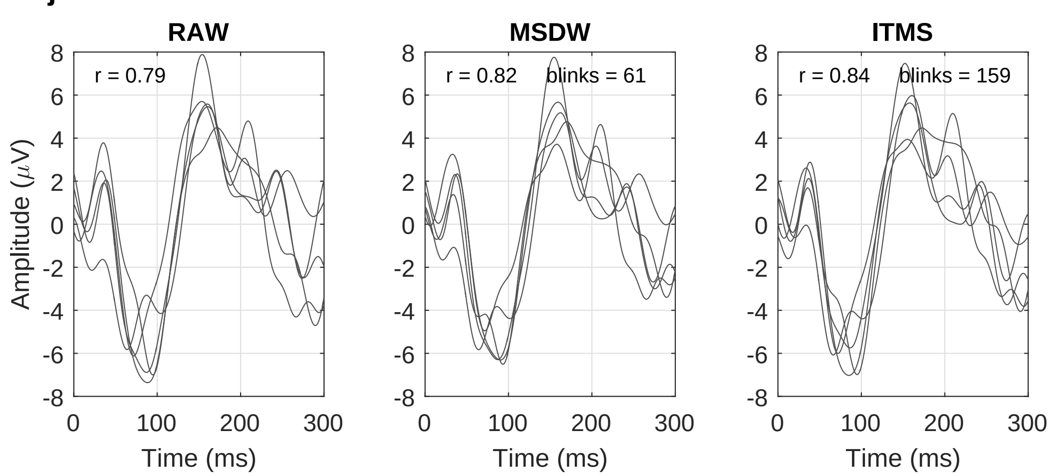
Subject 19



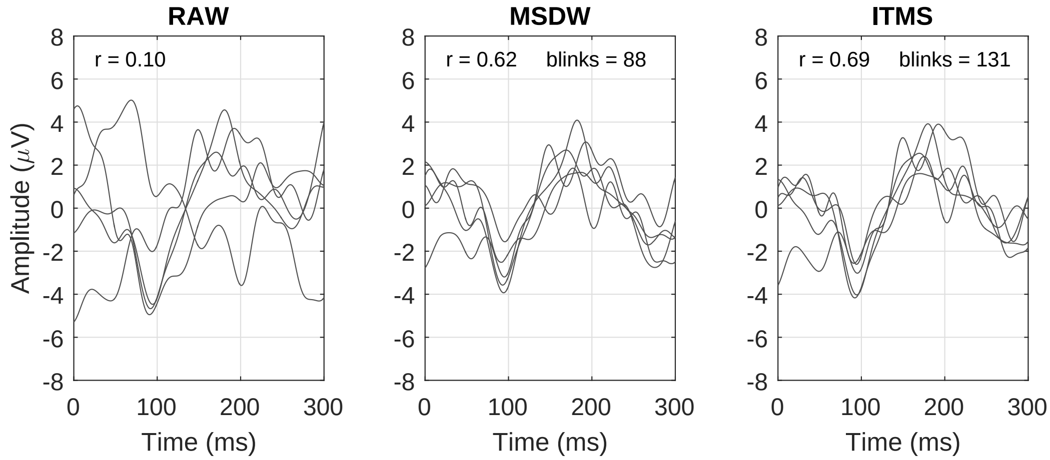
Subject 20



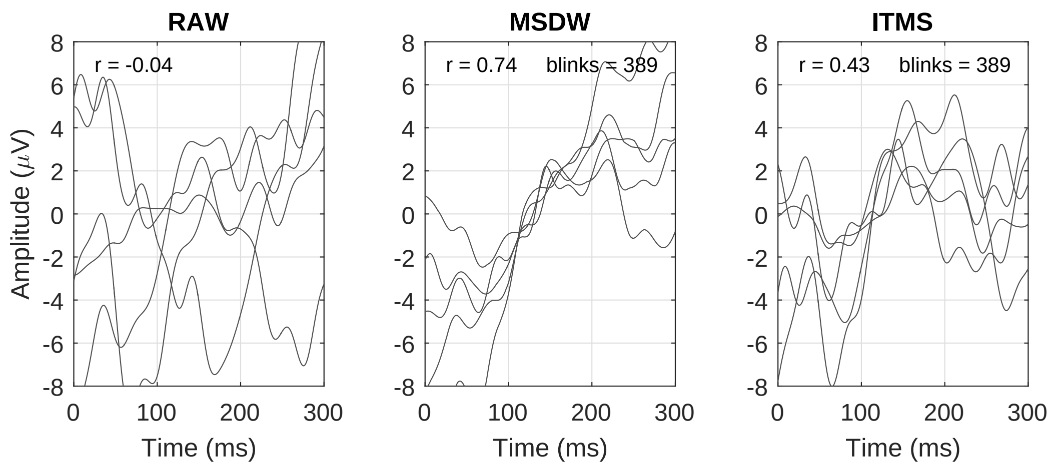
Subject 21



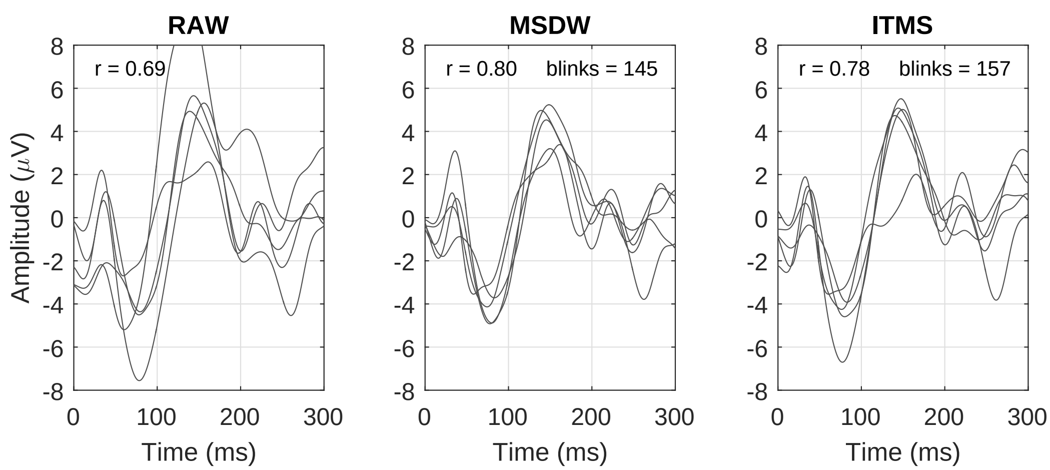
Subject 22



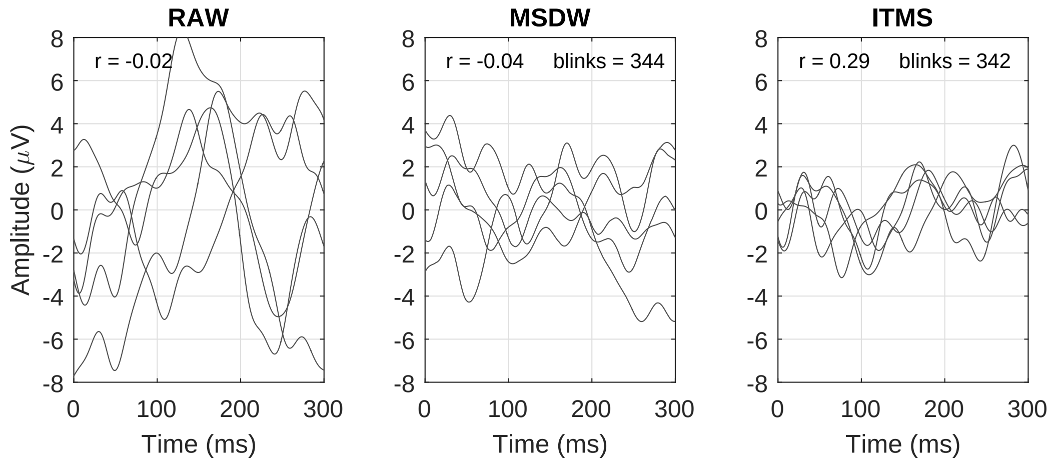
Subject 23



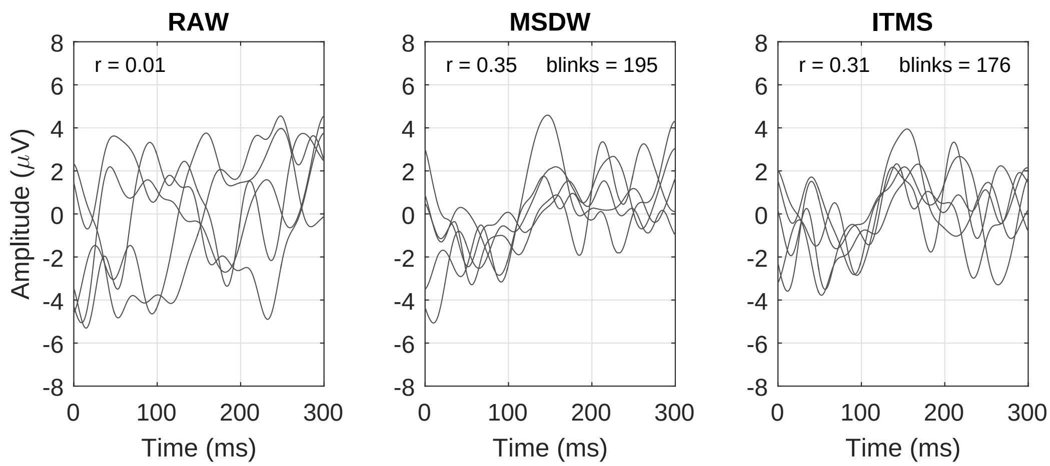
Subject 24



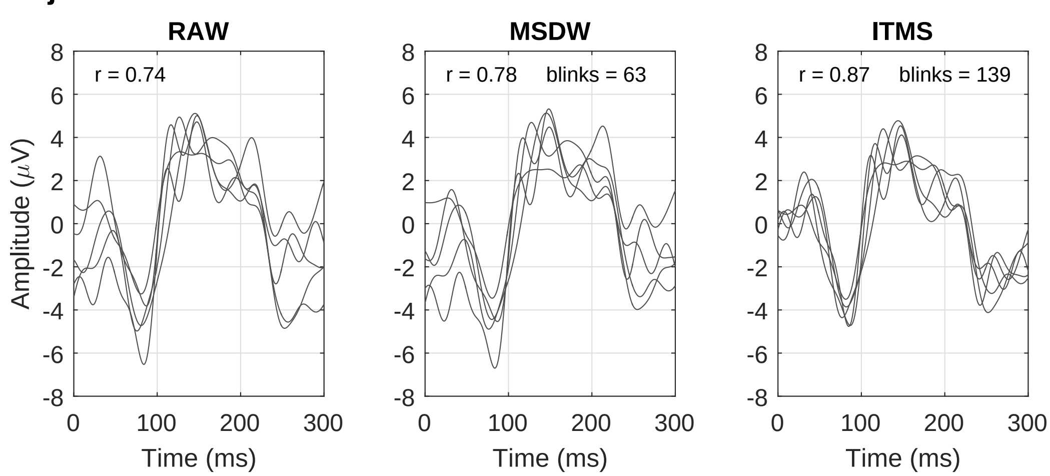
Subject 25



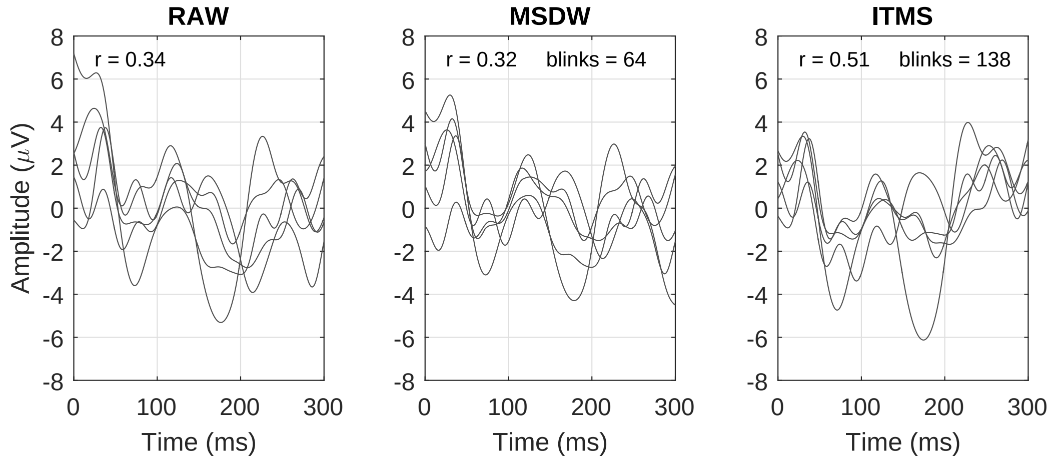
Subject 26



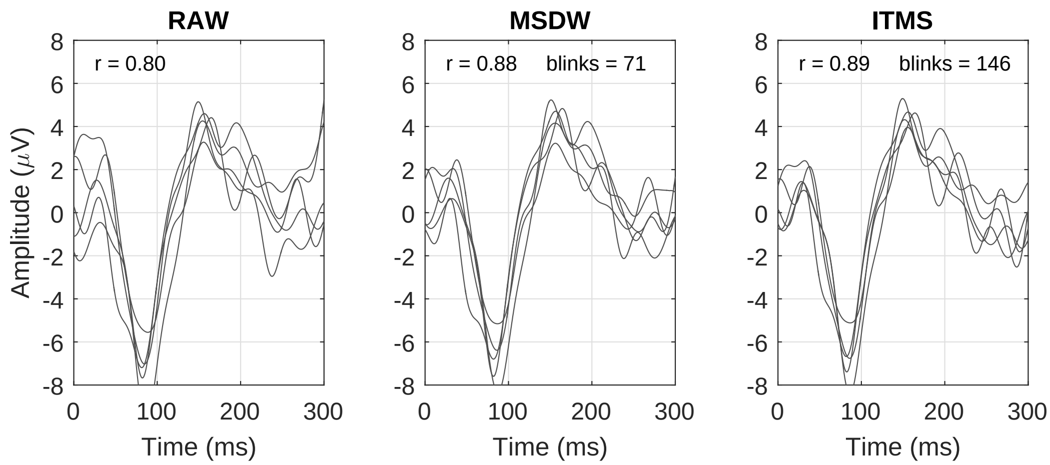
Subject 27



Subject 28



Subject 29



Subject 30