Electronic Supplementary Information for

Savitzky-Golay Processing and Bidimensional Plotting of Current-Time Signals from Stochastic Blocking Electrochemistry to Analyze Mixtures of Rod-Shaped Bacteria

Ashley Tubbs,[a] Junaid U. Ahmed,[b] Jayani Christopher,[a] and Julio C. Alvarez[a]*

[a] *Chemistry Department, Virginia Commonwealth University, Richmond, VA, 23294.*
[b] *Chemistry Department, Khulna University of Engineering and Technology, Bangladesh.*

MATLAB script for performing Savitzky-Golay derivative filter on *i-t* recordings from stochastic blocking electrochemistry.

A Savitsky Golay-filter is applied to the signal of interest by selecting an interval of data points equal to an odd number (11 for our case). The center point of this interval is replaced by a value from polynomial fitting using convolution coefficients listed in the original paper by Savitzky and Golay for a data set of 11. Then, the interval moves one data point over to calculate the next center value and continue the iteration until all values in the signal are replaced by polynomial fits.
In the code below, lines in green explain what the other code lines are doing.

```
%%Initial Inputs

    clear
    clc

    DELIMITED = ',';
    HEADER = 16; %Adjust to last line of heading text

%%Import Data

    [file,path] = uigetfile('*.txt');
    filename = horzcat(path,file);
    data_import = importdata(filename,DELIMITED,HEADER);
    data(:,:) = data_import.data;

%% Data is being pulled from the numbered columns -- adjust as appropriate

    time(:,:) = data(:,1); % Time, s
    I(:,:) = data(:,2) * 10^9; % Current, nA
    dt = 0.05; % adjust for sampling rate

%% Establish window size

    ncount = 'Enter the window size (5, 7, 9,....25) and press "Enter" ';

    ncount = input(ncount);


%% Establish Savitzky-Golay reference table (SG coefficients and
normalization factors)

    SGtable = (-(ncount-1)/2):((ncount-1)/2); % SG coefficients for quadratic
first-derivative filter

    normquad = [10 28 60 110 182 280 408 570 770 1012 1300]; %Normalization
coefficients for quadratic first-derivative filter (only for window sizes 5-
25)

    normco = normquad(:,1+((ncount-5)/2)); %Pulls normalization coefficient
for input window size (ncount)

%% SG Analysis

    datawin = ones(size(time,1)-(ncount+1),ncount); %Create table of indices
for moving window to reference

for k = 1:(size(time,1)-(ncount+1))

    datawin(k,:) = k:k+(ncount-1);

    SGdata (k,:) = (1/(dt*normco))*(SGtable*I(datawin(k,:))); %Savitzky-Golay
first-derivative filtering
```

```
end

    SGtime = time(((ncount-1)/2+1):(end-(((ncount-1)/2)+2)),:); %Cut excess
time from beginning and end of time data to match SGdata length



%% plot i-t data

    Fig1Title = erase(file,'.txt')
    figure(1);
    plot(time,I,'LineWidth',0.1,'DisplayName',[Fig1Title],'Color','r');
    legend('Location','northeast')
    xlabel('time (s)')
    ylabel('I/t (nA/s)')
    set(gca,'TickLength',[0 0]);

    ax1 = gca;

    hold on


%% plot SG data


    figure(2);
    plot(SGtime,SGdata,'LineWidth',0.1,'DisplayName',['MatLab '
num2str(ncount) ' pt SG Filter'],'Color','r');
    legend('Location','northeast')
    xlabel('time (s)')
    ylabel('I/t (nA/s)')
    set(gca,'TickLength',[0 0]);

    ax1 = gca;

    hold on


%% ID peaks based on average y-value and noise

    SGavg = mean(SGdata(3/dt:end,:)); %Find average from t = 3s to end of
expt. Remove 3 s to remove segment of exponential current decay

    SGNoise = std(SGdata(3/dt:end,:)); %Find noise from t = 3s to end of
expt. Remove 3 s to remove segment of exponential current decay

    [pks,locs] =
findpeaks(SGdata,'MinPeakHeight',(SGavg+SGNoise),'MinPeakDistance',(0.75/dt))
; %Identify peaks


%% Remove peaks with area under curve less than 0.01 nA
```

```
for k = 1:size(locs,1)

    PkSizeDown = locs(k) - (0.1/dt);
    PkSizeUp = locs(k) + (0.1/dt);


    if PkSizeDown <= 0
        AUC(k) =
trapz(SGtime(1:(locs(k)+(0.1/dt))),SGdata(1:(locs(k)+(0.1/dt))));

    elseif PkSizeUp >= size(SGtime)
        AUC(k) = trapz(SGtime((locs(k)-(0.1/dt)):end),SGdata((locs(k)-
(0.1/dt)):end));

    else
        AUC(k) = trapz(SGtime((locs(k)-
(0.1/dt)):(locs(k)+(0.1/dt))),SGdata((locs(k)-(0.1/dt)):(locs(k)+(0.1/dt))));

    end


end

    notpeaks = AUC < 0.01;

    locs(notpeaks) = [];

        clear notpeaks

        clear AUC


%% Remove non-peaks in exponential decay curve

for k = 1:size((nonzeros(locs<20/dt)),1) % Limits k to peak locations in
first 20 seconds of i-t curve

    PkSizeDown = locs(k) - (5/dt);

    if PkSizeDown <= 0

        PkChk(k) = 1; % If peak location is in first 5 seconds of i-t curve,
PkChk = 1

    else

        PkChk(k) = SGdata(locs(k))<SGdata(locs(k)-(4/dt)); %If peak current
is less than the current 4 seconds earlier, PkChk = 1

    end

end
```

```
    exist PkChk;

    PK = ans;

if PK == 1
    notpeaks = PkChk>0;
    locs(notpeaks) = []; %any location that equals 1 is not a peak, and thus
deleted
end


%% Plot remaining peaks and noise

    figure(2);
    FirstLocs = plot(SGtime(locs),SGdata(locs),'v','MarkerFaceColor',[0
0.4470 0.7410],'MarkerEdgeColor',[0 0.4470 0.7410]);

    ax2 = axes('Position',[.65 .65 .23 .23]); %Add inset to graph to plot
step corresponding to peak


%% Manually add extra peaks

    ExtraPks = input('Are there extra peaks that need to be added? [Y/N]
','s');

if ExtraPks == 'Y'

   NumEPs = input('How many extra peaks would you like to add? ');
   disp('Scroll to zoom. Click to exit zoom, then select the peak you would
like to add.')

   for k = 1:NumEPs

       zoom xon
       w = waitforbuttonpress; %MatLab will allow you to zoom by hovering
over the graph and scrolling
       [x,y] = ginput(1);
       NewPeak = dsearchn(SGtime, x);

plot(SGtime(NewPeak),SGdata(NewPeak),'v','MarkerFaceColor','#7E2F8E','MarkerE
dgeColor','#7E2F8E');
       locs = sort(cat(1,locs,NewPeak));
       zoom out

   end

end

%% Select Peak Start and End Points
```

```
    disp('Scroll to zoom in and out of the peak. Click to exit zoom, then
click on the start and end points of the peak.')
    disp('If the yellow indicator is highlighting an incorrectly identified
peak, please select an area that encompasses 9 seconds or more.')


for k = 1:size(locs,1)

    if (locs(k) - 5/dt) <= 0
        upperlim = SGtime(locs(k) + 5/dt);
        set(ax1,'XLim',[SGtime(1,1) upperlim]) %Sets the x-axis to zoom in on
a peak that starts within 5 seconds of the beginning of the i-t curve
        temp = plot(ax2,SGtime(1:locs(k)+2.5/dt,:),I(1:locs(k)+2.5/dt,:));
%Plots the corresponding step in the inset

    elseif size(SGtime,1) <= (locs(k)) + 5/dt
        lowerlim = SGtime(locs(k) - 5/dt);
        set(ax1,'XLim',[lowerlim SGtime(end,:)]); %Sets the x-axis to zoom in
on a peak that ends within 5 seconds of the end of the i-t curve
        temp = plot(ax2,SGtime(locs(k)-2.5/dt:end,:),I(locs(k)-
2.5/dt:end,:)); %Plots the corresponding step in the inset

    else
        lowerlim = SGtime(locs(k) - 5/dt);
        upperlim = SGtime(locs(k) + 5/dt);
        set(ax1,'XLim',[lowerlim upperlim]); %Sets the x-axis to zoom in on
peak k to within +/- 5 seconds
        temp = plot(ax2,SGtime(locs(k)-2.5/dt:locs(k)+2.5/dt,:),I(locs(k)-
2.5/dt:locs(k)+2.5/dt,:)); %Plots the corresponding step in the inset

    end

    pmark =
plot(ax1,SGtime(locs(k)),SGdata(locs(k)),'v','MarkerFaceColor','y','MarkerEdg
eColor','y'); %Highlights the mark on the graph yellow

    zoom xon
    w = waitforbuttonpress;
    [x,y] = ginput(2);
    ind1 = dsearchn(SGtime, x(1)); %Find index of first data point
    ind2 = dsearchn(SGtime, x(2)); %Find index of second data point
    deltat(k) = x(2)-x(1); %Calculate delta t for peak k
    AUC(k) = trapz(SGtime(ind1:ind2),SGdata(ind1:ind2)); %Area under the
curve for peak k



    delete(pmark)
    delete(temp)
    cla(ax2,'reset');

    PCheck = ismember(SGtime(locs(k)),SGtime(ind1:ind2,:));

    if PCheck == 0
```

```
        [M,N] = max(SGdata(ind1:ind2,:));
        TempSG = SGtime(ind1:ind2,:);
        M = TempSG(N);
        NewPeak = dsearchn(SGtime,M);

plot(SGtime(NewPeak),SGdata(NewPeak),'v','MarkerFaceColor','#7E2F8E','MarkerE
dgeColor','#7E2F8E');
        locs = cat(1,locs,NewPeak);


    end

    xlim auto
    ylim auto

end


    hold off
    zoom off



    locs = sort(locs);

    notpeaks = deltat > 9.000;
        deltat(notpeaks) = [];
        AUC(notpeaks) = [];

    SGresults = table(AUC',deltat','VariableNames',["delta_i","delta_t"])


%% Reset

    disp('When you are ready to continue, press any key')
        pause;

        close all %Closes all figures

    disp('Please update dt if the sampling interval will change for the next
file analyzed');
```