% Cell Align Pipeline plots Event-Aligned Timestamps for One Session

% Derived from code: time stamp cell session and compile single session

clear; close all; clc;

cell\_1\_name = '2023\_07\_13\_0001\_1';

cell\_2\_name = '2023\_07\_13\_0001\_2';

cell\_names\_eq = eq(cell\_1\_name, cell\_2\_name);

cell\_session = cell\_1\_name(cell\_names\_eq);

cells\_co\_occur = 0.1; % in seconds, co-ocurrence defined across full session

cells\_corr\_thresh = 0.15; % minimum R correlation for inclusion

cells\_std\_thresh = 1; % for thresholding currents from noise

% Load .mat for EPSC Data, Timestamp, and Sample Frequency

[fnEPSC, drDECMAT, ~] = uigetfile('\*.xlsx',' Pick the Excel Data file'); % also defines root folder

[pathfile,align\_namefile,extfile] = fileparts([drDECMAT fnEPSC]);

disp('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')

disp(['Loading Cell Pair Data ' cell\_session(1:end-1)])

% load Cell Channel Data

cell\_1\_channel = readtable([drDECMAT fnEPSC],'Sheet',cell\_1\_name,'VariableNamingRule','preserve'); % Cell 1 channel

cell\_2\_channel = readtable([drDECMAT fnEPSC],'Sheet',cell\_2\_name,'VariableNamingRule','preserve'); % Cell 2 channel

disp('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')

disp('Start and Initialize Parameters for Analysis')

events\_samp\_start = 50; % 50 samples = -5 ms, total 100 Hz

events\_samp\_end = 50; % 50 samples = +5 ms, total 100 Hz

cell\_1\_std\_sigma = cells\_std\_thresh; % sigma threshold to accept epsc (from z-score)

cell\_2\_std\_sigma = cells\_std\_thresh; % sigma threshold to accept epsc (from z-score)

debug\_figures = false;

plot\_figures = true;

save\_figures = false;

write\_tables = false;

disp('Parameters Initialized')

% %% Load Excel EPSC Data - Load Raw .xlsx for Cell Peaks

cell\_1\_indx = cell\_1\_channel.('Event Num.'); % peak event index

cell\_1\_time = cell\_1\_channel.('Event Time (s)'); % peak event time in s

cell\_1\_base = cell\_1\_channel.('Baseline (pA)'); % moving window in pA

cell\_1\_peak = cell\_1\_channel.('Peak (pA)'); % peak current from 0 in pA

cell\_1\_amp = cell\_1\_channel.('Amplitude (pA)'); % peak current from baseline in pA

cell\_1\_rise = cell\_1\_channel.('Rise Time (ms)'); % 10% to 90% in ms

cell\_1\_halfwidth = cell\_1\_channel.('Half-Width (ms)'); % rise to decay in ms

cell\_1\_decay = cell\_1\_channel.('Decay % (ms)'); % 90% to 10% in ms

cell\_1\_AUC = cell\_1\_channel.('AUC (pA ms)'); % area under rise to decay in pA\*ms

cell\_1\_AUCtime = cell\_1\_channel.('AUC Time (ms)'); % in ms

cell\_2\_indx = cell\_2\_channel.('Event Num.'); % peak event index

cell\_2\_time = cell\_2\_channel.('Event Time (s)'); % peak event time in s

cell\_2\_base = cell\_2\_channel.('Baseline (pA)'); % moving window in pA

cell\_2\_peak = cell\_2\_channel.('Peak (pA)'); % peak current from 0 in pA

cell\_2\_amp = cell\_2\_channel.('Amplitude (pA)'); % peak current from baseline in pA

cell\_2\_rise = cell\_2\_channel.('Rise Time (ms)'); % 10% to 90% in ms

cell\_2\_halfwidth = cell\_2\_channel.('Half-Width (ms)'); % rise to decay in ms

cell\_2\_decay = cell\_2\_channel.('Decay % (ms)'); % 90% to 10% in ms

cell\_2\_AUC = cell\_2\_channel.('AUC (pA ms)'); % area under rise to decay in pA\*ms

cell\_2\_AUCtime = cell\_2\_channel.('AUC Time (ms)'); % in ms

base\_line = [cell\_1\_base;cell\_2\_base];

dataend\_1 = ceil(cell\_1\_time(end));

dataend\_2 = ceil(cell\_2\_time(end));

dataend\_end = max(dataend\_1,dataend\_2);

epsc\_samp\_freq = 10000;

epsc\_time\_line = 0:1/epsc\_samp\_freq:dataend\_end;

epsc\_sess\_dur = epsc\_time\_line(end);

cell\_1\_event\_count = numel(cell\_1\_time);

cell\_2\_event\_count = numel(cell\_2\_time);

disp(['MATLAB Data for ' align\_namefile ' Loaded'])

disp('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')

% %% EPSC Events - Identify Cell Activity

win\_samp\_width = events\_samp\_start + events\_samp\_end;

delta\_t = 1/epsc\_samp\_freq;

epsc\_time\_ms = 1000\*(delta\_t:delta\_t:(dataend\_end - 2\*delta\_t)); % change s to ms

% for peak current thresholding in pA

base\_mu\_mean = mean(base\_line); % current baseline mean

base\_sigma\_dev = std(base\_line); % current baseline sigma

% set peak current above amplitude threshold in pA

sel\_thresh = abs(cells\_std\_thresh.\*base\_sigma\_dev) + abs(base\_mu\_mean);

cell\_1\_index = find(abs(cell\_1\_peak) >= sel\_thresh);

cell\_1\_event\_amps = cell\_1\_peak(cell\_1\_index);

cell\_1\_event\_times = cell\_1\_time(cell\_1\_index);

cell\_2\_index = find(abs(cell\_2\_peak) >= sel\_thresh);

cell\_2\_event\_amps = cell\_2\_peak(cell\_2\_index);

cell\_2\_event\_times = cell\_2\_time(cell\_2\_index);

disp(['EPSC Session Duration: ' num2str(epsc\_sess\_dur) ' s']);

disp('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')

disp('>>>>>>>>>>>>>>>>>>>>>>>>>')

disp(['Cell 1 total: ' num2str(numel(cell\_1\_time)) ' events']);

disp(['Cell 1 mean epsc: ' num2str(mean(cell\_1\_peak)) ' pA']);

disp(['Cell 1 rate: ' num2str(round(numel(cell\_1\_time)/((epsc\_sess\_dur - cell\_1\_time(1))),4)) ' Hz']);

disp('>>>>>>>>>>>>>>>>>>>>>>>>>')

disp(['Cell 2 total: ' num2str(numel(cell\_2\_time)) ' events']);

disp(['Cell 2 mean epsc: ' num2str(mean(cell\_2\_peak)) ' pA']);

disp(['Cell 2 rate: ' num2str(round(numel(cell\_2\_time)/((epsc\_sess\_dur - cell\_2\_time(1))),4)) ' Hz']);

disp('>>>>>>>>>>>>>>>>>>>>>>>>>')

if debug\_figures==true

pause

if save\_figures==false

close all

end

end

% %% EPSC Events - Correlate Alignments

% Find Events by Timestamp

delay\_to\_cell\_1 = [];

delay\_to\_cell\_2 = [];

event\_cell\_1\_ts = [];

event\_cell\_2\_ts = [];

for ievent1 = 1:length(cell\_1\_time)

temp\_event\_1 = cell\_1\_time(ievent1);

for ievent2 = 1:length(cell\_2\_time)

temp\_event\_2 = cell\_2\_time(ievent2);

temp\_delay = temp\_event\_2 - temp\_event\_1;

if -cells\_co\_occur<=temp\_delay && temp\_delay<=cells\_co\_occur

delay\_to\_cell\_1(ievent1,ievent2) = temp\_delay;

event\_cell\_1\_ts(ievent1,ievent2) = temp\_event\_1;

else

delay\_to\_cell\_1(ievent1,ievent2) = NaN;

event\_cell\_1\_ts(ievent1,ievent2) = NaN;

end

end

end

for ievent2 = 1:length(cell\_2\_time)

temp\_event\_2 = cell\_2\_time(ievent2);

temp\_peak\_2 = cell\_2\_peak(ievent2);

for ievent1 = 1:length(cell\_1\_time)

temp\_event\_1 = cell\_1\_time(ievent1);

temp\_delay = temp\_event\_1 - temp\_event\_2;

if -cells\_co\_occur<=temp\_delay && temp\_delay<=cells\_co\_occur

delay\_to\_cell\_2(ievent2,ievent1) = temp\_delay;

event\_cell\_2\_ts(ievent2,ievent1) = temp\_event\_2;

else

delay\_to\_cell\_2(ievent2,ievent1) = NaN;

event\_cell\_2\_ts(ievent2,ievent1) = NaN;

end

end

end

cell\_delay\_trim\_1 = delay\_to\_cell\_1(~isnan(delay\_to\_cell\_1));

cell\_delay\_trim\_2 = delay\_to\_cell\_2(~isnan(delay\_to\_cell\_2));

cell\_to\_1\_ts\_trim = reshape(cell\_delay\_trim\_1,1,numel(cell\_delay\_trim\_1));

cell\_to\_2\_ts\_trim = reshape(cell\_delay\_trim\_2,1,numel(cell\_delay\_trim\_2));

mean\_delay\_1\_2 = mean(cell\_to\_1\_ts\_trim,'omitnan');

mean\_delay\_2\_1 = mean(cell\_to\_2\_ts\_trim,'omitnan');

sum\_overlaps\_1\_2 = sum((numel(cell\_delay\_trim\_1)+numel(cell\_delay\_trim\_2))/2);

% Define Cross Correlation

cell\_1\_ts\_trim = event\_cell\_1\_ts(~isnan(event\_cell\_1\_ts));

cell\_2\_ts\_trim = event\_cell\_2\_ts(~isnan(event\_cell\_2\_ts));

[corr\_ts\_wide\_1\_2,lags\_ts\_wide\_1\_2] = funct\_cowan\_corr(cell\_2\_ts\_trim,cell\_1\_ts\_trim,cells\_co\_occur,10,'none');

[corr\_ts\_zoom\_1\_2,lags\_ts\_zoom\_1\_2] = funct\_cowan\_corr(cell\_2\_ts\_trim,cell\_1\_ts\_trim,cells\_co\_occur/10,10,'none');

[corr\_ts\_narr\_1\_2,lags\_ts\_narr\_1\_2] = funct\_cowan\_corr(cell\_2\_ts\_trim,cell\_1\_ts\_trim,cells\_co\_occur/100,10,'none');

lim\_corr\_wide = [0 ceil(10\*max(corr\_ts\_wide\_1\_2)+cells\_corr\_thresh)]/10;

lim\_corr\_zoom = [0 ceil(10\*max(corr\_ts\_zoom\_1\_2)+cells\_corr\_thresh)]/10;

lim\_corr\_narr = [0 ceil(10\*max(corr\_ts\_narr\_1\_2)+cells\_corr\_thresh)]/10;

[wide\_peaks, wide\_index] = findpeaks(corr\_ts\_wide\_1\_2);

max\_corr\_wide = max(corr\_ts\_wide\_1\_2);

max\_lag\_wide = max(lags\_ts\_wide\_1\_2(wide\_index));

max\_peak\_wide = max(corr\_ts\_wide\_1\_2(wide\_index));

[zoom\_peaks, zoom\_index] = findpeaks(corr\_ts\_zoom\_1\_2);

max\_corr\_zoom = max(corr\_ts\_zoom\_1\_2);

max\_lag\_zoom = max(lags\_ts\_zoom\_1\_2(zoom\_index));

max\_peak\_zoom = max(corr\_ts\_zoom\_1\_2(zoom\_index));

[narr\_peaks, narr\_index] = findpeaks(corr\_ts\_narr\_1\_2);

max\_corr\_narr = max(corr\_ts\_narr\_1\_2);

max\_lag\_narr = max(lags\_ts\_narr\_1\_2(narr\_index));

max\_peak\_narr = max(corr\_ts\_narr\_1\_2(narr\_index));

index\_thresh\_wide = corr\_ts\_wide\_1\_2>cells\_corr\_thresh;

max\_thresh\_wide = max(corr\_ts\_wide\_1\_2(index\_thresh\_wide));

index\_thresh\_zoom = corr\_ts\_zoom\_1\_2>cells\_corr\_thresh;

max\_thresh\_zoom = max(corr\_ts\_zoom\_1\_2(index\_thresh\_zoom));

index\_thresh\_narr = corr\_ts\_narr\_1\_2>cells\_corr\_thresh;

max\_thresh\_narr = max(corr\_ts\_narr\_1\_2(index\_thresh\_narr));

mean\_corr\_wide = mean(corr\_ts\_wide\_1\_2,'omitnan');

mean\_corr\_zoom = mean(corr\_ts\_zoom\_1\_2,'omitnan');

mean\_corr\_narr = mean(corr\_ts\_narr\_1\_2,'omitnan');

mean\_lag\_wide = mean(lags\_ts\_wide\_1\_2(index\_thresh\_wide),'omitnan');

mean\_lag\_zoom = mean(lags\_ts\_zoom\_1\_2(index\_thresh\_zoom),'omitnan');

mean\_lag\_narr = mean(lags\_ts\_narr\_1\_2(index\_thresh\_narr),'omitnan');

mean\_thresh\_wide = mean(corr\_ts\_wide\_1\_2(index\_thresh\_wide),'omitnan');

mean\_thresh\_zoom = mean(corr\_ts\_zoom\_1\_2(index\_thresh\_zoom),'omitnan');

mean\_thresh\_narr = mean(corr\_ts\_narr\_1\_2(index\_thresh\_narr),'omitnan');

Cell\_1\_2\_Corrs\_fig = figure('units', 'normalized', 'outerposition', [0 0.05 1 0.5]);

sgtitle([cell\_session(1:end-1) ' Cross Correlations, ' num2str(cells\_corr\_thresh) ' R Threshold'],'Interpreter','none')

subplot(1,3,1)

bar(lags\_ts\_wide\_1\_2,corr\_ts\_wide\_1\_2)

hold all

scatter(lags\_ts\_wide\_1\_2(index\_thresh\_wide),corr\_ts\_wide\_1\_2(index\_thresh\_wide),10,'k','filled')

xlabel([num2str(10\*cells\_co\_occur/1) ' s Window'])

xline(0,'--k','LineWidth',1)

xline(mean\_lag\_wide,'k','LineWidth',2)

ylabel('Cross Correlation (R\_c\_o\_r\_r)')

yline(cells\_corr\_thresh,'-.k','LineWidth',1)

ylim(lim\_corr\_wide)

title('Wide Cooccurrence')

subplot(1,3,2)

bar(lags\_ts\_zoom\_1\_2,corr\_ts\_zoom\_1\_2)

hold all

scatter(lags\_ts\_zoom\_1\_2(index\_thresh\_zoom),corr\_ts\_zoom\_1\_2(index\_thresh\_zoom),10,'k','filled')

xlabel([num2str(10\*cells\_co\_occur/10) ' s Window'])

xline(0,'--k','LineWidth',1)

xline(mean\_lag\_zoom,'k','LineWidth',2)

ylabel('Cross Correlation (R\_c\_o\_r\_r)')

yline(cells\_corr\_thresh,'-.k','LineWidth',1)

ylim(lim\_corr\_zoom)

title('Zoomed Cooccurrence')

subplot(1,3,3)

bar(lags\_ts\_narr\_1\_2,corr\_ts\_narr\_1\_2)

hold all

scatter(lags\_ts\_narr\_1\_2(index\_thresh\_narr),corr\_ts\_narr\_1\_2(index\_thresh\_narr),10,'k','filled')

xlabel([num2str(10\*cells\_co\_occur/100) ' s Window'])

xline(0,'--k','LineWidth',1)

xline(mean\_lag\_narr,'k','LineWidth',2)

ylabel('Cross Correlation (R\_c\_o\_r\_r)')

yline(cells\_corr\_thresh,'-.k','LineWidth',1)

ylim(lim\_corr\_narr)

title('Narrow Cooccurrence')

if debug\_figures==true

pause

if save\_figures==false

close all

end

end