% 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;

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

epsc\_samp\_freq = 10000;

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

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

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

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

disp('Initiating Cell Pair Data Sessions for Analysis')

a\_list = [1,2,3,5,6,9,11,13];

b\_list = [4,7,8,10,12,14,15];

alpha\_list = {};

alpha\_list(a\_list,1) = {'ENNN'};

alpha\_list(b\_list,1) = {'ENEN'};

corr\_names = strings;

corr\_session = [];

corr\_stat\_path = pwd;

corr\_stat\_file = 'EPSC\_Correlation';

corr\_sheets = sheetnames([corr\_stat\_file '.xlsx']);

types\_list = [];

types\_list(1,a\_list) = 1;

types\_list(1,b\_list) = 2;

%% Concatenate All Sessions

corr\_Z\_100\_cat = [];

corr\_Z\_50\_cat = [];

corr\_Z\_10\_cat = [];

lag\_Z\_100\_cat = [];

lag\_Z\_50\_cat = [];

lag\_Z\_10\_cat = [];

cell\_sess\_time = [];

cell\_1\_amps\_cat = [];

cell\_2\_amps\_cat = [];

cell\_1\_counts\_cat = [];

cell\_2\_counts\_cat = [];

cell\_1\_rates\_cat = [];

cell\_2\_rates\_cat = [];

cell\_ts\_group\_cat = [];

% load Cell Session Data

for colabel\_idx = 1:length(corr\_sheets)

curr\_sess = corr\_sheets(colabel\_idx);

curr\_stat = readtable([corr\_stat\_path '\' corr\_stat\_file],'Sheet',curr\_sess,'VariableNamingRule','preserve');

temp\_numbers = cellfun(@isnumeric,table2cell(curr\_stat));

temp\_fields = fieldnames(curr\_stat);

temp\_structs = {(temp\_fields{(temp\_numbers==0)})};

for temp\_idx = 1:length(temp\_structs)

curr\_stat.(temp\_structs{temp\_idx}) = NaN;

end

corr\_names(colabel\_idx,:) = curr\_sess; % Session

corr\_session(colabel\_idx,:) = table2array(curr\_stat); % Session

cell\_1\_name = [char(corr\_names(colabel\_idx,:)) '\_1'];

cell\_2\_name = [char(corr\_names(colabel\_idx,:)) '\_2'];

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

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

% 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

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\_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

% 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);

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));

cell\_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\_Z\_100\_1\_2,lags\_ts\_Z\_100\_1\_2] = funct\_cowan\_corr(cell\_2\_ts\_trim,cell\_1\_ts\_trim,cells\_co\_occur/10,10,'none');

[corr\_ts\_Z\_50\_1\_2,lags\_ts\_Z\_50\_1\_2] = funct\_cowan\_corr(cell\_2\_ts\_trim,cell\_1\_ts\_trim,cells\_co\_occur/20,10,'none');

[corr\_ts\_Z\_10\_1\_2,lags\_ts\_Z\_10\_1\_2] = funct\_cowan\_corr(cell\_2\_ts\_trim,cell\_1\_ts\_trim,cells\_co\_occur/100,10,'none');

corr\_Z\_100\_cat = cat(1,corr\_Z\_100\_cat,corr\_ts\_Z\_100\_1\_2);

corr\_Z\_50\_cat = cat(1,corr\_Z\_50\_cat,corr\_ts\_Z\_50\_1\_2);

corr\_Z\_10\_cat = cat(1,corr\_Z\_10\_cat,corr\_ts\_Z\_10\_1\_2);

lag\_Z\_100\_cat = cat(1,lag\_Z\_100\_cat,lags\_ts\_Z\_100\_1\_2);

lag\_Z\_50\_cat = cat(1,lag\_Z\_50\_cat,lags\_ts\_Z\_50\_1\_2);

lag\_Z\_10\_cat = cat(1,lag\_Z\_10\_cat,lags\_ts\_Z\_10\_1\_2);

cell\_1\_event\_peaks = mean(cell\_1\_peak,'omitnan');

cell\_2\_event\_peaks = mean(cell\_2\_peak,'omitnan');

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

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

temp\_sess\_time = round(max([cell\_1\_time; cell\_2\_time]));

cell\_sess\_time(colabel\_idx) = temp\_sess\_time;

cell\_1\_event\_rate = cell\_1\_event\_count/temp\_sess\_time;

cell\_2\_event\_rate = cell\_2\_event\_count/temp\_sess\_time;

cell\_1\_counts\_cat = cat(1,cell\_1\_counts\_cat,cell\_1\_event\_count);

cell\_2\_counts\_cat = cat(1,cell\_2\_counts\_cat,cell\_2\_event\_count);

cell\_1\_amps\_cat = cat(1,cell\_1\_amps\_cat,cell\_1\_event\_peaks);

cell\_2\_amps\_cat = cat(1,cell\_2\_amps\_cat,cell\_2\_event\_peaks);

cell\_1\_rates\_cat = cat(1,cell\_1\_rates\_cat,cell\_1\_event\_rate);

cell\_2\_rates\_cat = cat(1,cell\_2\_rates\_cat,cell\_2\_event\_rate);

if any(colabel\_idx == a\_list)

cell\_ts\_group\_cat = cat(1,cell\_ts\_group\_cat,repmat('a',1,numel(cell\_1\_ts\_trim))');

elseif any(colabel\_idx == b\_list)

cell\_ts\_group\_cat = cat(1,cell\_ts\_group\_cat,repmat('b',1,numel(cell\_2\_ts\_trim))');

end

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

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

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

disp(['Cell 1 mean epsc: ' num2str(round(cell\_1\_event\_peaks),3) ' pA']);

disp(['Cell 1 rate: ' num2str(round(cell\_1\_event\_rate,4)) ' Hz']);

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

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

disp(['Cell 2 mean epsc: ' num2str(round(cell\_2\_event\_peaks),3) ' pA']);

disp(['Cell 2 rate: ' num2str(round(cell\_2\_event\_rate,4)) ' Hz']);

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

end

%% Correlation Distributions Across Lag Window

disp('Statistics on Correlation Distributions Across Lag Windows')

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

lags\_A\_100 = mean(lag\_Z\_100\_cat,1);

lags\_A\_50 = mean(lag\_Z\_50\_cat,1);

lags\_A\_10 = mean(lag\_Z\_10\_cat,1);

corr\_A\_100\_a = corr\_Z\_100\_cat(a\_list,:)';

corr\_A\_50\_a = corr\_Z\_50\_cat(a\_list,:)';

corr\_A\_10\_a = corr\_Z\_10\_cat(a\_list,:)';

corr\_A\_100\_b = corr\_Z\_100\_cat(b\_list,:)';

corr\_A\_50\_b = corr\_Z\_50\_cat(b\_list,:)';

corr\_A\_10\_b = corr\_Z\_10\_cat(b\_list,:)';

max\_A\_100\_a = max(corr\_A\_100\_a,[],'omitnan');

max\_A\_100\_b = max(corr\_A\_100\_b,[],'omitnan');

max\_A\_50\_a = max(corr\_A\_50\_a,[],'omitnan');

max\_A\_50\_b = max(corr\_A\_50\_b,[],'omitnan');

max\_A\_10\_a = max(corr\_A\_10\_a,[],'omitnan');

max\_A\_10\_b = max(corr\_A\_10\_b,[],'omitnan');

mean\_A\_100\_a = mean(corr\_A\_100\_a,2,'omitnan');

mean\_A\_100\_b = mean(corr\_A\_100\_b,2,'omitnan');

mean\_A\_50\_a = mean(corr\_A\_50\_a,2,'omitnan');

mean\_A\_50\_b = mean(corr\_A\_50\_b,2,'omitnan');

mean\_A\_10\_a = mean(corr\_A\_10\_a,2,'omitnan');

mean\_A\_10\_b = mean(corr\_A\_10\_b,2,'omitnan');

sigma\_A\_100\_a = std(corr\_A\_100\_a,0,2,'omitnan');

sigma\_A\_100\_b = std(corr\_A\_100\_b,0,2,'omitnan');

sigma\_A\_50\_a = std(corr\_A\_50\_a,0,2,'omitnan');

sigma\_A\_50\_b = std(corr\_A\_50\_b,0,2,'omitnan');

sigma\_A\_10\_a = std(corr\_A\_10\_a,0,2,'omitnan');

sigma\_A\_10\_b = std(corr\_A\_10\_b,0,2,'omitnan');

A\_100\_a\_std\_sqrt = sqrt(size(mean\_A\_100\_a,1));

A\_50\_a\_std\_sqrt = sqrt(size(mean\_A\_50\_a,1));

A\_10\_a\_std\_sqrt = sqrt(size(mean\_A\_10\_a,1));

A\_100\_a\_std\_errs = sigma\_A\_100\_a/A\_100\_a\_std\_sqrt;

A\_50\_a\_std\_errs = sigma\_A\_50\_a/A\_50\_a\_std\_sqrt;

A\_10\_a\_std\_errs = sigma\_A\_10\_a/A\_10\_a\_std\_sqrt;

A\_100\_b\_std\_sqrt = sqrt(size(mean\_A\_100\_b,1));

A\_50\_b\_std\_sqrt = sqrt(size(mean\_A\_50\_b,1));

A\_10\_b\_std\_sqrt = sqrt(size(mean\_A\_10\_b,1));

A\_100\_b\_std\_errs = sigma\_A\_100\_b/A\_100\_b\_std\_sqrt;

A\_50\_b\_std\_errs = sigma\_A\_50\_b/A\_50\_b\_std\_sqrt;

A\_10\_b\_std\_errs = sigma\_A\_10\_b/A\_10\_b\_std\_sqrt;

corr\_Z\_100\_a\_center = corr\_Z\_100\_cat(a\_list,6:16);

corr\_Z\_50\_a\_center = corr\_Z\_50\_cat(a\_list,6:16);

corr\_Z\_10\_a\_center = corr\_Z\_10\_cat(a\_list,6:16);

corr\_Z\_100\_b\_center = corr\_Z\_100\_cat(b\_list,6:16);

corr\_Z\_50\_b\_center = corr\_Z\_50\_cat(b\_list,6:16);

corr\_Z\_10\_b\_center = corr\_Z\_10\_cat(b\_list,6:16);

% disp('Is there higher maximum correlation (perioccurrence) for colabeled versus uncolabeled sessions?')

figure

hold all

shadedErrorBar(1000\*lags\_A\_100,mean\_A\_100\_a,A\_100\_a\_std\_errs,'lineProps','k');

shadedErrorBar(1000\*lags\_A\_100,mean\_A\_100\_b,A\_100\_b\_std\_errs,'lineProps','r');

xlim([1000\*lags\_A\_100(1) 1000\*lags\_A\_100(end)])

title('100 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

figure

hold all

shadedErrorBar(1000\*lags\_A\_50,mean\_A\_50\_a,A\_50\_a\_std\_errs,'lineProps','k');

shadedErrorBar(1000\*lags\_A\_50,mean\_A\_50\_b,A\_50\_b\_std\_errs,'lineProps','g');

xlim([1000\*lags\_A\_50(1) 1000\*lags\_A\_50(end)])

title('50 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

figure

hold all

shadedErrorBar(1000\*lags\_A\_10,mean\_A\_10\_a,A\_10\_a\_std\_errs,'lineProps','k');

shadedErrorBar(1000\*lags\_A\_10,mean\_A\_10\_b,A\_10\_b\_std\_errs,'lineProps','b');

xlim([1000\*lags\_A\_10(1) 1000\*lags\_A\_10(end)])

title('10 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

disp('There is a trend in cooccurrence for colabeled versus uncolabeled sessions.')

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

disp('End Correlation Distributions Across Lag Window')

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

%% Correlation Statistics Across Sessions

disp('Statistics on Normality of Correlations, Distributions, Histograms')

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

disp('Are colabeled versus uncolabeled recording session durations different?')

cells\_group = [repmat('C1',[15 1]); repmat('C2',[15 1])];

types\_group = repmat(alpha\_list,[2 1]);

[p\_AN\_sess\_time,~,stats\_AN\_sess\_time] = anova1(cell\_sess\_time,alpha\_list,'off');

figure

multcompare(stats\_AN\_sess\_time);

title('Durations')

disp('No difference in duration for colabeled versus uncolabeled sessions.')

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

disp('Are colabeled versus uncolabeled recording sessions intrinsically different?')

% figure('units', 'normalized', 'outerposition', [0 0 1 1]);

[p\_amps\_colabels,~,stats\_amps\_colabels] = anovan([cell\_1\_amps\_cat;cell\_2\_amps\_cat],{cells\_group,types\_group},'model','interaction','varnames',{'Amps','Session'},'display','off');

figure

multcompare(stats\_amps\_colabels,'Dimension',[2 2]);

title('Amplitudes')

[p\_counts\_colabels,~,stats\_counts\_colabels] = anovan([cell\_1\_counts\_cat;cell\_2\_counts\_cat],{cells\_group,types\_group},'model','interaction','varnames',{'Counts','Session'},'display','off');

figure

multcompare(stats\_counts\_colabels,'Dimension',[2 2]);

title('Counts')

[p\_rates\_colabels,~,stats\_rates\_colabels] = anovan([cell\_1\_rates\_cat;cell\_2\_rates\_cat],{cells\_group,types\_group},'model','interaction','varnames',{'Rates','Session'},'display','off');

figure

multcompare(stats\_rates\_colabels,'Dimension',[2 2]);

title('Rates')

disp('No difference in EPSC, Count, or Rate for colabeled versus uncolabeled sessions.')

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

%% Signal Detection Correlation Distributions

% Define binary response variables.

bin\_res = 0.001;

bin\_values = -bin\_res:bin\_res:0.50;

% Prediction Accuracy for True = Colabeled, at Choice Lag

disp('Prediction Accuracy for True = Colabeled Sessions at Choice Lag');

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

lag\_interest = 11; % Choose range between 1 to 21 lags

% Define true false histograms.

hist\_100\_a\_corrs\_cat = histcounts(corr\_A\_100\_a(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

hist\_50\_a\_corrs\_cat = histcounts(corr\_A\_50\_a(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

hist\_10\_a\_corrs\_cat = histcounts(corr\_A\_10\_a(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

hist\_100\_b\_corrs\_cat = histcounts(corr\_A\_100\_b(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

hist\_50\_b\_corrs\_cat = histcounts(corr\_A\_50\_b(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

hist\_10\_b\_corrs\_cat = histcounts(corr\_A\_10\_b(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

% Define true false histograms.

Trues\_cumsum\_100 = cumsum(flip(hist\_100\_b\_corrs\_cat));

Falses\_cumsum\_100 = cumsum(flip(hist\_100\_a\_corrs\_cat));

AUC\_ROC\_T\_F\_100 = trapz(Falses\_cumsum\_100,Trues\_cumsum\_100);

Choice\_T\_F\_100 = sqrt(2).\*norminv(AUC\_ROC\_T\_F\_100);

Trues\_cumsum\_50 = cumsum(flip(hist\_50\_b\_corrs\_cat));

Falses\_cumsum\_50 = cumsum(flip(hist\_50\_a\_corrs\_cat));

AUC\_ROC\_T\_F\_50 = trapz(Falses\_cumsum\_50,Trues\_cumsum\_50);

Choice\_T\_F\_50 = sqrt(2).\*norminv(AUC\_ROC\_T\_F\_50);

Trues\_cumsum\_10 = cumsum(flip(hist\_10\_b\_corrs\_cat));

Falses\_cumsum\_10 = cumsum(flip(hist\_10\_a\_corrs\_cat));

AUC\_ROC\_T\_F\_10 = trapz(Falses\_cumsum\_10,Trues\_cumsum\_10);

Choice\_T\_F\_10 = sqrt(2).\*norminv(AUC\_ROC\_T\_F\_10);

disp(['Performance AUC at Bin ' num2str(lag\_interest) ' = ' num2str(AUC\_ROC\_T\_F\_100\*100) ' %']);

disp(['Performance d Prime at Bin ' num2str(lag\_interest) ' = ' num2str(Choice\_T\_F\_100) ' in SDT']);

disp(['Performance AUC at Bin ' num2str(lag\_interest) ' = ' num2str(AUC\_ROC\_T\_F\_50\*100) ' %']);

disp(['Performance d Prime at Bin ' num2str(lag\_interest) ' = ' num2str(Choice\_T\_F\_50) ' in SDT']);

disp(['Performance AUC at Bin ' num2str(lag\_interest) ' = ' num2str(AUC\_ROC\_T\_F\_10\*100) ' %']);

disp(['Performance d Prime at Bin ' num2str(lag\_interest) ' = ' num2str(Choice\_T\_F\_10) ' in SDT']);

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

figure

hold all

ax\_plot(1) = plot(Falses\_cumsum\_100,Trues\_cumsum\_100,'r','LineWidth',2);

ax\_plot(2) = plot(0:bin\_res:1,0:bin\_res:1,'--k','LineWidth',1);

lgd\_plot = legend('box','off');

lgd\_plot.String{1} = ['AUC = ' num2str(AUC\_ROC\_T\_F\_100)];

lgd\_plot.String{2} = ['d prime = ' num2str(Choice\_T\_F\_100)];

xlabel('False positive rate')

ylabel('True positive rate')

title(['ROC 100 ms at Lag = ' num2str(1000\*lags\_A\_100(lag\_interest)) ' ms'])

figure

hold all

ax\_plot(1) = plot(Falses\_cumsum\_50,Trues\_cumsum\_50,'g','LineWidth',2);

ax\_plot(2) = plot(0:bin\_res:1,0:bin\_res:1,'--k','LineWidth',1);

lgd\_plot = legend('box','off');

lgd\_plot.String{1} = ['AUC = ' num2str(AUC\_ROC\_T\_F\_50)];

lgd\_plot.String{2} = ['d prime = ' num2str(Choice\_T\_F\_50)];

xlabel('False positive rate')

ylabel('True positive rate')

title(['ROC 50 ms at Lag = ' num2str(1000\*lags\_A\_50(lag\_interest)) ' ms'])

%% Load Random Jitter Session Iterations

load('EPSC\_Corr\_Rand\_Iter\_0')

%% Correlation Aligned Distributions Across Lag Window

lags\_A\_100 = mean(lag\_Z\_100\_cat,1);

lags\_A\_50 = mean(lag\_Z\_50\_cat,1);

lags\_A\_10 = mean(lag\_Z\_10\_cat,1);

corr\_A\_100\_a = corr\_Z\_100\_cat(a\_list,:)';

corr\_A\_50\_a = corr\_Z\_50\_cat(a\_list,:)';

corr\_A\_10\_a = corr\_Z\_10\_cat(a\_list,:)';

corr\_A\_100\_b = corr\_Z\_100\_cat(b\_list,:)';

corr\_A\_50\_b = corr\_Z\_50\_cat(b\_list,:)';

corr\_A\_10\_b = corr\_Z\_10\_cat(b\_list,:)';

mean\_A\_100\_a = mean(corr\_A\_100\_a,2,'omitnan');

mean\_A\_50\_a = mean(corr\_A\_50\_a,2,'omitnan');

mean\_A\_10\_a = mean(corr\_A\_10\_a,2,'omitnan');

sigma\_A\_100\_a = std(corr\_A\_100\_a,0,2,'omitnan');

sigma\_A\_50\_a = std(corr\_A\_50\_a,0,2,'omitnan');

sigma\_A\_10\_a = std(corr\_A\_10\_a,0,2,'omitnan');

A\_100\_a\_std\_sqrt = sqrt(size(mean\_A\_100\_a,1));

A\_50\_a\_std\_sqrt = sqrt(size(mean\_A\_50\_a,1));

A\_10\_a\_std\_sqrt = sqrt(size(mean\_A\_10\_a,1));

A\_100\_a\_std\_errs = sigma\_A\_100\_a/A\_100\_a\_std\_sqrt;

A\_50\_a\_std\_errs = sigma\_A\_50\_a/A\_50\_a\_std\_sqrt;

A\_10\_a\_std\_errs = sigma\_A\_10\_a/A\_10\_a\_std\_sqrt;

mean\_A\_100\_b = mean(corr\_A\_100\_b,2,'omitnan');

mean\_A\_50\_b = mean(corr\_A\_50\_b,2,'omitnan');

mean\_A\_10\_b = mean(corr\_A\_10\_b,2,'omitnan');

sigma\_A\_100\_b = std(corr\_A\_100\_b,0,2,'omitnan');

sigma\_A\_50\_b = std(corr\_A\_50\_b,0,2,'omitnan');

sigma\_A\_10\_b = std(corr\_A\_10\_b,0,2,'omitnan');

A\_100\_b\_std\_sqrt = sqrt(size(mean\_A\_100\_b,1));

A\_50\_b\_std\_sqrt = sqrt(size(mean\_A\_50\_b,1));

A\_10\_b\_std\_sqrt = sqrt(size(mean\_A\_10\_b,1));

A\_100\_b\_std\_errs = sigma\_A\_100\_b/A\_100\_b\_std\_sqrt;

A\_50\_b\_std\_errs = sigma\_A\_50\_b/A\_50\_b\_std\_sqrt;

A\_10\_b\_std\_errs = sigma\_A\_10\_b/A\_10\_b\_std\_sqrt;

center\_mean\_A\_100\_b = mean\_A\_100\_b(11)

center\_mean\_A\_100\_a = mean\_A\_100\_a(11)

center\_mean\_A\_50\_b = mean\_A\_50\_b(11)

center\_mean\_A\_50\_a = mean\_A\_50\_a(11)

center\_mean\_A\_10\_b = mean\_A\_10\_b(11)

center\_mean\_A\_10\_a = mean\_A\_10\_a(11)

center\_std\_dev\_A\_100\_b = sigma\_A\_100\_b(11)

center\_std\_dev\_A\_100\_a = sigma\_A\_100\_a(11)

center\_std\_dev\_A\_50\_b = sigma\_A\_50\_b(11)

center\_std\_dev\_A\_50\_a = sigma\_A\_50\_a(11)

center\_std\_dev\_A\_10\_b = sigma\_A\_10\_b(11)

center\_std\_dev\_A\_10\_a = sigma\_A\_10\_a(11)

Cell\_Sess\_Corrs\_fig = figure('units', 'normalized', 'outerposition', [0 0.02 1 0.96]);

sgtitle(['Perioccurrence by Correlation Lags Across ' num2str(size(types\_list(1,:),2)) ' Sessions'])

subplot(3,3,1)

hold all

shadedErrorBar(1000\*lags\_A\_100,mean\_A\_100\_b,A\_100\_b\_std\_errs,'lineProps','r');

xlim([1000\*lags\_A\_100(1) 1000\*lags\_A\_100(end)])

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.1,'-.k')

title('100 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

subplot(3,3,2)

hold all

shadedErrorBar(1000\*lags\_A\_50,mean\_A\_50\_b,A\_50\_b\_std\_errs,'lineProps','g');

xlim([1000\*lags\_A\_50(1) 1000\*lags\_A\_50(end)])

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.1,'-.k')

title('50 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

subplot(3,3,3)

hold all

shadedErrorBar(1000\*lags\_A\_10,mean\_A\_10\_b,A\_10\_b\_std\_errs,'lineProps','b');

xlim([1000\*lags\_A\_10(1) 1000\*lags\_A\_10(end)])

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.1,'-.k')

title('10 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

subplot(3,3,4)

hold all

shadedErrorBar(1000\*lags\_A\_100,mean\_A\_100\_a,A\_100\_a\_std\_errs,'lineProps','k');

xlim([1000\*lags\_A\_100(1) 1000\*lags\_A\_100(end)])

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.1,'-.k')

title('100 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

subplot(3,3,5)

hold all

shadedErrorBar(1000\*lags\_A\_50,mean\_A\_50\_a,A\_50\_a\_std\_errs,'lineProps','k');

xlim([1000\*lags\_A\_50(1) 1000\*lags\_A\_50(end)])

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.1,'-.k')

title('50 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

subplot(3,3,6)

hold all

shadedErrorBar(1000\*lags\_A\_10,mean\_A\_10\_a,A\_10\_a\_std\_errs,'lineProps','k');

xlim([1000\*lags\_A\_10(1) 1000\*lags\_A\_10(end)])

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.1,'-.k')

title('10 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.25])

%% Correlation Jittered Distributions Across Lag Window

lags\_J\_100 = mean(lags\_Z\_100\_iter,1,'omitnan');

lags\_J\_50 = mean(lags\_Z\_50\_iter,1,'omitnan');

lags\_J\_10 = mean(lags\_Z\_10\_iter,1,'omitnan');

mean\_J\_100\_a = mean(mean\_Z\_100\_a\_iter,1,'omitnan')

mean\_J\_50\_a = mean(mean\_Z\_50\_a\_iter,1,'omitnan')

mean\_J\_10\_a = mean(mean\_Z\_10\_a\_iter,1,'omitnan')

J\_100\_a\_std\_devs = std(mean\_Z\_100\_a\_iter,0,1,'omitnan')

J\_50\_a\_std\_devs = std(mean\_Z\_50\_a\_iter,0,1,'omitnan')

J\_10\_a\_std\_devs = std(mean\_Z\_10\_a\_iter,0,1,'omitnan')

J\_100\_a\_std\_sqrt = sqrt(size(mean\_Z\_100\_a\_iter,1));

J\_50\_a\_std\_sqrt = sqrt(size(mean\_Z\_50\_a\_iter,1));

J\_10\_a\_std\_sqrt = sqrt(size(mean\_Z\_10\_a\_iter,1));

J\_100\_a\_std\_errs = J\_100\_a\_std\_devs/J\_100\_a\_std\_sqrt;

J\_50\_a\_std\_errs = J\_50\_a\_std\_devs/J\_50\_a\_std\_sqrt;

J\_10\_a\_std\_errs = J\_10\_a\_std\_devs/J\_10\_a\_std\_sqrt;

mean\_J\_100\_b = mean(mean\_Z\_100\_b\_iter,1,'omitnan')

mean\_J\_50\_b = mean(mean\_Z\_50\_b\_iter,1,'omitnan')

mean\_J\_10\_b = mean(mean\_Z\_10\_b\_iter,1,'omitnan')

J\_100\_b\_std\_devs = std(mean\_Z\_100\_b\_iter,0,1,'omitnan')

J\_50\_b\_std\_devs = std(mean\_Z\_50\_b\_iter,0,1,'omitnan')

J\_10\_b\_std\_devs = std(mean\_Z\_10\_b\_iter,0,1,'omitnan')

J\_100\_b\_std\_sqrt = sqrt(size(mean\_Z\_100\_b\_iter,1));

J\_50\_b\_std\_sqrt = sqrt(size(mean\_Z\_50\_b\_iter,1));

J\_10\_b\_std\_sqrt = sqrt(size(mean\_Z\_10\_b\_iter,1));

J\_100\_b\_std\_errs = J\_100\_b\_std\_devs/J\_100\_b\_std\_sqrt;

J\_50\_b\_std\_errs = J\_50\_b\_std\_devs/J\_50\_b\_std\_sqrt;

J\_10\_b\_std\_errs = J\_10\_b\_std\_devs/J\_10\_b\_std\_sqrt;

center\_mean\_J\_100\_b = mean\_J\_100\_b(11)

center\_mean\_J\_100\_a = mean\_J\_100\_a(11)

center\_mean\_J\_50\_b = mean\_J\_50\_b(11)

center\_mean\_J\_50\_a = mean\_J\_50\_a(11)

center\_mean\_J\_10\_b = mean\_J\_10\_b(11)

center\_mean\_J\_10\_a = mean\_J\_10\_a(11)

center\_std\_dev\_J\_100\_b = J\_100\_b\_std\_devs(11)

center\_std\_dev\_J\_100\_a = J\_100\_a\_std\_devs(11)

center\_std\_dev\_J\_50\_b = J\_50\_b\_std\_devs(11)

center\_std\_dev\_J\_50\_a = J\_50\_a\_std\_devs(11)

center\_std\_dev\_J\_10\_b = J\_10\_b\_std\_devs(11)

center\_std\_dev\_J\_10\_a = J\_10\_a\_std\_devs(11)

[~,p\_100\_a\_rand\_sess,~,~] = ttest2(mean\_Z\_100\_a\_iter(:,11), corr\_Z\_100\_cat(a\_list,11),'Vartype','unequal');

[~,p\_100\_b\_rand\_sess,~,~] = ttest2(mean\_Z\_100\_b\_iter(:,11), corr\_Z\_100\_cat(b\_list,11),'Vartype','unequal');

[~,p\_50\_a\_rand\_sess,~,~] = ttest2(mean\_Z\_50\_a\_iter(:,11), corr\_Z\_50\_cat(a\_list,11),'Vartype','unequal');

[~,p\_50\_b\_rand\_sess,~,~] = ttest2(mean\_Z\_50\_b\_iter(:,11), corr\_Z\_50\_cat(b\_list,11),'Vartype','unequal');

[~,p\_10\_a\_rand\_sess,~,~] = ttest2(mean\_Z\_10\_a\_iter(:,11), corr\_Z\_10\_cat(a\_list,11),'Vartype','unequal');

[~,p\_10\_b\_rand\_sess,~,~] = ttest2(mean\_Z\_10\_b\_iter(:,11), corr\_Z\_10\_cat(b\_list,11),'Vartype','unequal');

Rand\_Mean\_Corrs\_fig = figure('units', 'normalized', 'outerposition', [0 0.02 1 0.96]);

sgtitle(['Mean Correlations, Jittered Random ' num2str(num\_iterations) ' Iterations'])

subplot(3,3,1)

hold all

shadedErrorBar(1000\*lags\_J\_100,mean\_J\_100\_b,J\_100\_b\_std\_devs,'lineProps','r');

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.06,'-.k')

title('100 ms Lag Jittered');xlabel('Lag (ms)');xlim([-100 100]);ylabel('Correlation');ylim([0 0.25])

subplot(3,3,2)

hold all

shadedErrorBar(1000\*lags\_J\_50,mean\_J\_50\_b,J\_50\_b\_std\_devs,'lineProps','g');

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.03,'-.k')

title('50 ms Lag Jittered');xlabel('Lag (ms)');xlim([-50 50]);ylabel('Correlation');ylim([0 0.25])

subplot(3,3,3)

hold all

shadedErrorBar(1000\*lags\_J\_10,mean\_J\_10\_b,J\_10\_b\_std\_devs,'lineProps','b');

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.01,'-.k')

title('10 ms Lag Jittered');xlabel('Lag (ms)');xlim([-10 10]);ylabel('Correlation');ylim([0 0.25])

subplot(3,3,4)

hold all

shadedErrorBar(1000\*lags\_J\_100,mean\_J\_100\_a,J\_100\_a\_std\_devs,'lineProps','k');

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.06,'-.k')

title('100 ms Lag Jittered');xlabel('Lag (ms)');xlim([-100 100]);ylabel('Correlation');ylim([0 0.25])

subplot(3,3,5)

hold all

shadedErrorBar(1000\*lags\_J\_50,mean\_J\_50\_a,J\_50\_a\_std\_devs,'lineProps','k');

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.03,'-.k')

title('50 ms Lag Jittered');xlabel('Lag (ms)');xlim([-50 50]);ylabel('Correlation');ylim([0 0.25])

subplot(3,3,6)

hold all

shadedErrorBar(1000\*lags\_J\_10,mean\_J\_10\_a,J\_10\_a\_std\_devs,'lineProps','k');

xline(0,'-k','LineWidth',10,'Alpha',0.1);yline(0,'-.k');yline(0.01,'-.k')

title('10 ms Lag Jittered');xlabel('Lag (ms)');xlim([-10 10]);ylabel('Correlation');ylim([0 0.25])

%% Correlation Aligned Jitter Strength - Plots

mean\_Jitter\_100 = mean([mean\_Z\_100\_a\_iter;mean\_Z\_100\_b\_iter],1,'omitnan');

mean\_Jitter\_50 = mean([mean\_Z\_50\_a\_iter;mean\_Z\_50\_b\_iter],1,'omitnan');

J\_100\_std\_devs = std([mean\_Z\_100\_a\_iter;mean\_Z\_100\_b\_iter],0,1,'omitnan');

J\_50\_std\_devs = std([mean\_Z\_50\_a\_iter;mean\_Z\_50\_b\_iter],0,1,'omitnan');

J\_100\_std\_sqrt = sqrt(size([mean\_Z\_100\_a\_iter;mean\_Z\_100\_b\_iter],1));

J\_50\_std\_sqrt = sqrt(size([mean\_Z\_50\_a\_iter;mean\_Z\_50\_b\_iter],1));

J\_100\_std\_errs = J\_100\_std\_devs/J\_100\_std\_sqrt;

J\_50\_std\_errs = J\_50\_std\_devs/J\_50\_std\_sqrt;

color\_jitts = [25 25 25]./255;

color\_100\_b = [60 120 240]./255;

color\_100\_a = [20 60 100]./255;

color\_50\_b = [120 60 240]./255;

color\_50\_a = [80 40 80]./255;

max\_a\_matrix = 2\*ones([length(max\_A\_100\_a) 1]);

max\_b\_matrix = 1\*ones([length(max\_A\_100\_b) 1]);

% Correlations\_Jitter = figure('units', 'normalized', 'outerposition', [0 0.16 1 0.67]);

% sgtitle('Perioccurrence by Correlation Lags Across Sessions')

% subplot(1,2,1)

figure

hold all

H\_100\_Aligns\_B = shadedErrorBar(1000\*lags\_J\_100,mean\_A\_100\_b,A\_100\_b\_std\_errs);

H\_100\_Aligns\_B.mainLine.Color = color\_100\_b; H\_100\_Aligns\_B.mainLine.LineWidth = 1;

H\_100\_Aligns\_B.patch.FaceColor = color\_100\_b; H\_100\_Aligns\_B.patch.FaceAlpha = 0.5;

H\_100\_Aligns\_B.edge(1).Color = color\_100\_b; H\_100\_Aligns\_B.edge(2).Color = color\_100\_b;

H\_100\_Aligns\_B.edge(1).LineWidth = 1; H\_100\_Aligns\_B.edge(2).LineWidth = 1;

H\_100\_Aligns\_A = shadedErrorBar(1000\*lags\_J\_100,mean\_A\_100\_a,A\_100\_a\_std\_errs);

H\_100\_Aligns\_A.mainLine.Color = color\_100\_a; H\_100\_Aligns\_A.mainLine.LineWidth = 1;

H\_100\_Aligns\_A.patch.FaceColor = color\_100\_a; H\_100\_Aligns\_A.patch.FaceAlpha = 0.5;

H\_100\_Aligns\_A.edge(1).Color = color\_100\_a; H\_100\_Aligns\_A.edge(2).Color = color\_100\_a;

H\_100\_Aligns\_A.edge(1).LineWidth = 1; H\_100\_Aligns\_A.edge(2).LineWidth = 1;

H\_100\_Jitters = shadedErrorBar(1000\*lags\_J\_100,mean\_Jitter\_100,J\_100\_std\_errs);

H\_100\_Jitters.mainLine.Color = color\_jitts; H\_100\_Jitters.mainLine.LineWidth = 1;

H\_100\_Jitters.patch.FaceColor = color\_jitts; H\_100\_Jitters.patch.FaceAlpha = 0.2;

H\_100\_Jitters.edge(1).Color = color\_jitts; H\_100\_Jitters.edge(2).Color = color\_jitts;

H\_100\_Jitters.edge(1).LineWidth = 1; H\_100\_Jitters.edge(2).LineWidth = 1;

xlim([1000\*lags\_J\_100(1) 1000\*lags\_J\_100(end)]);xline(0,'-k','LineWidth',10,'Alpha',0.1);xline(0,'k','LineWidth',2);yline(0.1,'-.k','LineWidth',1)

title('10 ms across 200 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.15])

legend({'Colabeled' 'Uncolabeled' 'Jittered'},'Box','off','Location','northwest')

% create smaller axes in top right, and plot on it

h = axes('Parent',gcf,'Position',[0.35 0.75 0.1 0.1]);

hold all

box on

scatter(max\_a\_matrix,max\_A\_100\_a,10,'filled','MarkerFaceColor',color\_100\_a,'MarkerEdgeColor',color\_100\_a,'XJitter','randn','XJitterWidth',0.4)

scatter(max\_b\_matrix,max\_A\_100\_b,10,'filled','MarkerFaceColor',color\_100\_b,'MarkerEdgeColor',color\_100\_b,'XJitter','randn','XJitterWidth',0.4)

set(gca,'XLim',[0 3],'YLim',[0 0.3])

yline(0.15,'--k','LineWidth',1)

% subplot(1,2,2)

figure

hold all

H\_50\_Aligns\_B = shadedErrorBar(1000\*lags\_J\_50,mean\_A\_50\_b,A\_50\_b\_std\_errs);

H\_50\_Aligns\_B.mainLine.Color = color\_50\_b; H\_50\_Aligns\_B.mainLine.LineWidth = 1;

H\_50\_Aligns\_B.patch.FaceColor = color\_50\_b; H\_50\_Aligns\_B.patch.FaceAlpha = 0.5;

H\_50\_Aligns\_B.edge(1).Color = color\_50\_b; H\_50\_Aligns\_B.edge(2).Color = color\_50\_b;

H\_50\_Aligns\_B.edge(1).LineWidth = 1; H\_50\_Aligns\_B.edge(2).LineWidth = 1;

H\_50\_Aligns\_A = shadedErrorBar(1000\*lags\_J\_50,mean\_A\_50\_a,A\_50\_a\_std\_errs);

H\_50\_Aligns\_A.mainLine.Color = color\_50\_a; H\_50\_Aligns\_A.mainLine.LineWidth = 1;

H\_50\_Aligns\_A.patch.FaceColor = color\_50\_a; H\_50\_Aligns\_A.patch.FaceAlpha = 0.5;

H\_50\_Aligns\_A.edge(1).Color = color\_50\_a; H\_50\_Aligns\_A.edge(2).Color = color\_50\_a;

H\_50\_Aligns\_A.edge(1).LineWidth = 1; H\_50\_Aligns\_A.edge(2).LineWidth = 1;

H\_50\_Jitters = shadedErrorBar(1000\*lags\_J\_50,mean\_Jitter\_50,J\_50\_std\_errs);

H\_50\_Jitters.mainLine.Color = color\_jitts; H\_50\_Jitters.mainLine.LineWidth = 1;

H\_50\_Jitters.patch.FaceColor = color\_jitts; H\_50\_Jitters.patch.FaceAlpha = 0.2;

H\_50\_Jitters.edge(1).Color = color\_jitts; H\_50\_Jitters.edge(2).Color = color\_jitts;

H\_50\_Jitters.edge(1).LineWidth = 1; H\_50\_Jitters.edge(2).LineWidth = 1;

xlim([1000\*lags\_J\_50(1) 1000\*lags\_J\_50(end)]);xline(0,'-k','LineWidth',10,'Alpha',0.1);xline(0,'k','LineWidth',2);yline(0.1,'-.k','LineWidth',1)

title('5 ms across 100 ms Lag Window');xlabel('Lag (ms)');ylabel('Correlation');ylim([0 0.15])

legend({'Colabeled' 'Uncolabeled' 'Jittered'},'Box','off','Location','northwest')

% create smaller axes in top right, and plot on it

h = axes('Parent',gcf,'Position',[0.78 0.75 0.1 0.1]);

hold all

box on

scatter(max\_a\_matrix,max\_A\_50\_a,10,'filled','MarkerFaceColor',color\_50\_a,'MarkerEdgeColor',color\_50\_a,'XJitter','randn','XJitterWidth',0.4)

scatter(max\_b\_matrix,max\_A\_50\_b,10,'filled','MarkerFaceColor',color\_50\_b,'MarkerEdgeColor',color\_50\_b,'XJitter','randn','XJitterWidth',0.4)

set(gca,'XLim',[0 3],'YLim',[0 0.3])

yline(0.10,'--k','LineWidth',1)

%% Correlation Distribution Signal Detection - Center Bin

% Define binary response variables.

bin\_res = 0.001;

bin\_values = -bin\_res:bin\_res:0.50;

% Prediction Accuracy for True = Colabeled, at Choice Lag

disp('Prediction Accuracy for True = Colabeled Sessions at Choice Lag');

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

color\_jitts = [25 25 25]./255;

color\_100\_b = [60 120 240]./255;

color\_100\_a = [20 60 100]./255;

color\_50\_b = [120 60 240]./255;

color\_50\_a = [80 40 80]./255;

lag\_interest = 11; % Choose range between 1 to 21 lags

% Define true false histograms.

hist\_100\_a\_corrs\_cat = histcounts(corr\_A\_100\_a(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

hist\_50\_a\_corrs\_cat = histcounts(corr\_A\_50\_a(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

hist\_100\_b\_corrs\_cat = histcounts(corr\_A\_100\_b(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

hist\_50\_b\_corrs\_cat = histcounts(corr\_A\_50\_b(lag\_interest,:),'BinEdges',bin\_values,'Normalization','probability');

% Define true false histograms.

Trues\_cumsum\_100 = cumsum(flip(hist\_100\_b\_corrs\_cat));

Falses\_cumsum\_100 = cumsum(flip(hist\_100\_a\_corrs\_cat));

AUC\_ROC\_T\_F\_100 = trapz(Falses\_cumsum\_100,Trues\_cumsum\_100);

Choice\_T\_F\_100 = sqrt(2).\*norminv(AUC\_ROC\_T\_F\_100);

Trues\_cumsum\_50 = cumsum(flip(hist\_50\_b\_corrs\_cat));

Falses\_cumsum\_50 = cumsum(flip(hist\_50\_a\_corrs\_cat));

AUC\_ROC\_T\_F\_50 = trapz(Falses\_cumsum\_50,Trues\_cumsum\_50);

Choice\_T\_F\_50 = sqrt(2).\*norminv(AUC\_ROC\_T\_F\_50);

disp(['Performance AUC at Choice Lag = ' num2str(AUC\_ROC\_T\_F\_100\*100) ' %']);

disp(['Performance d Prime at Choice Lag = ' num2str(Choice\_T\_F\_100) ' in SDT']);

disp(['Performance AUC at Choice Lag = ' num2str(AUC\_ROC\_T\_F\_50\*100) ' %']);

disp(['Performance d Prime at Choice Lag = ' num2str(Choice\_T\_F\_50) ' in SDT']);

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

% Plot the ROC curves.

% Corrs\_Prime\_Choice\_fig = figure('units', 'normalized', 'outerposition', [0 0.16 1 0.67]);

% subplot(1,2,1)

figure

hold all

ax\_plot(1) = plot(Falses\_cumsum\_100,Trues\_cumsum\_100,'Color',color\_100\_b,'LineWidth',3);

ax\_plot(2) = plot(0:bin\_res:1,0:bin\_res:1,'--k','LineWidth',2);

legend({['AUC = ' num2str(round(AUC\_ROC\_T\_F\_100,2))],'Chance = 0.50'},'box','off','Location','northwest');

xlabel('False positive rate')

ylabel('True positive rate')

title(['Receiver Operator Characteristic 100 ms at Lag = ' num2str(1000\*lags\_A\_100(lag\_interest)) ' ms'])

% subplot(1,2,2)

figure

hold all

ax\_plot(1) = plot(Falses\_cumsum\_50,Trues\_cumsum\_50,'Color',color\_50\_b,'LineWidth',3);

ax\_plot(2) = plot(0:bin\_res:1,0:bin\_res:1,'--k','LineWidth',2);

legend({['AUC = ' num2str(round(AUC\_ROC\_T\_F\_50,2))],'Chance = 0.50'},'box','off','Location','northwest');

xlabel('False positive rate')

ylabel('True positive rate')

title(['Receiver Operator Characteristic 50 ms at Lag = ' num2str(1000\*lags\_A\_50(lag\_interest)) ' ms'])