% --------- Pre-clinical fUSI Mice MK801 dose dependant data pre-processing analysis -----------------

%% Clear workspaces

clear; close all; clearvars; clc;

%% Assign variables to mice pD scan files

Fn\_mse\_1 = 'mk\_mse2\_c0\_1.scan';

Fn\_mse\_5 = 'mk\_mse5\_c1\_5.scan';

mice\_FN\_all = {{Fn\_mse\_1 Fn\_mse\_5}};

baseline\_Ranges\_all = {241:300 241:300};

%% Load DBS data files Data anaysis fitered data

%% Pre-process all patient fUSI data

sTart = 1; % Set filter parameters

lp1 = 0.0001;

lp2 = 0.03;

Fq = 1;

refLen = 30; binwdt = 120; bsLnRng = 113:119;

parameters = {{1 refLen binwdt} {1 lp1 lp2 Fq}}; % Set preprocessing parameters = cell1 - NormCorre cell2 - lowpass filter

miceData\_all = preProcess\_mkDoseData(mice\_FN\_all, parameters, sTart); % give Cells with each patients ...

% miceData\_all = preProcess\_mkDoseData(mice\_FN\_all, parameters, sTart, bsLnRng); % use if normCorre image template is spgive Cells with each patients ...

%% Visualize raw and filteres fUSI SC Data

mse\_contrast\_raw = {[0.0 0.4] [0.0 0.2] };

mse\_contrast\_mc = {[0.0 0.5] [0.0 0.4] };

mice\_contrast\_raw = {mse\_contrast\_raw};

mice\_contrast\_mc = {mse\_contrast\_mc};

%% Show short movie of the raw and filtered data

close all;

Exp\_labels = {{'mse-2' 'mse-5'}};

m\_num = 1;

nextMse\_data = miceData\_all{m\_num};

File\_num = 2; % patient number

mov\_len = 200; % movie length (s)

next\_mse = Exp\_labels{m\_num}{File\_num};

% \*\*\* View raw data

title\_raw = strcat(next\_mse, ' Raw');

raw\_FN = strcat(next\_mse, '\_raw');

raw\_data = nextMse\_data{1}{File\_num};

fUSimageToMovie\_enhanced(raw\_FN, raw\_data, mov\_len, 'hot', title\_raw, 0, mice\_contrast\_raw{m\_num}{File\_num}); % 1=save movie, 0 = don't save

% \*\*\* View filtered data

title\_filt = strcat(next\_mse, ' Filtered');

filt\_FN = strcat(next\_mse, '\_mcFrq');

filtered\_data = nextMse\_data{2}{File\_num};

fUSimageToMovie\_enhanced(filt\_FN, filtered\_data, mov\_len, 'hot', title\_filt, 0, mice\_contrast\_mc{m\_num}{File\_num}); % 1=save movie, 0 = don't save

%% Show 2D power Doppler mean/max figures (Can save file)

close all;

m\_num = 1;

nextMse\_data\_mc = miceData\_all{m\_num}{2};

mice\_contrast\_2D = {[0.0 0.45] [0.0 0.5] };

for pts = 1 : size(nextMse\_data\_mc,2)

mc3D = nextMse\_data\_mc{pts}; %next mc data

dataRng = baseline\_Ranges\_all{m\_num}; %25 : size(mc3D, 3) - 25; % bdata range

mxAll = max(mc3D(:, :, dataRng), [], 3); % 2D max in time dim

mnAll = mean(mc3D, 3); % 2D minimum

stdAll = std(mc3D, 0, 3); % 2D minimum

figure;

colormap(hot);

subplot(131); imagesc(imadjust(mnAll / max(mnAll, [], 'all'), mice\_contrast\_2D{pts})); title(strcat(Exp\_labels{m\_num}{pts}, ' bsline-Mean'));

set(gca,'xtick',[]); set(gca,'xticklabel',[]); set(gca,'ytick',[]); set(gca,'yticklabel',[]);

set(gca,'fontsize', 14);

subplot(132); imagesc(imadjust(mxAll / max(mxAll, [], 'all'), mice\_contrast\_2D{pts})); title(strcat(Exp\_labels{m\_num}{pts}, ' bsline-Max'));

set(gca,'xtick',[]); set(gca,'xticklabel',[]); set(gca,'ytick',[]); set(gca,'yticklabel',[]);

set(gca,'fontsize', 14);

subplot(133); imagesc(imadjust(stdAll / max(stdAll, [], 'all'), mice\_contrast\_2D{pts})); title(strcat(Exp\_labels{m\_num}{pts}, ' bsline-Std'));

set(gca,'xtick',[]); set(gca,'xticklabel',[]); set(gca,'ytick',[]); set(gca,'yticklabel',[]);

set(gca,'fontsize', 14);

set(gcf, 'Position', [50, 50, 1500, 320]);

nextPat = strcat('MxMnStd\_VascularMap', Exp\_labels{m\_num}{pts}, '\_p', num2str(m\_num), '.png');

saveas(gcf, nextPat);

end

%% Plot global raw and filtered time series

close all;

M\_num = 1;

nextMse\_data\_raw = miceData\_all{M\_num}{1};

nextMse\_data\_mc = miceData\_all{M\_num}{2};

next\_times = 1 : size(nextMse\_data\_mc{1}, 3);

bsln\_color = [0 0 0.1];

stim\_color = [1 0 1];

for pts = 1 : size(nextMse\_data\_mc,2)

raw3D = nextMse\_data\_raw{pts};

mc3D = nextMse\_data\_mc{pts};

bslneRng = baseline\_Ranges\_all{pts};

glo\_raw\_pct = getPercentChange\_1D(squeeze(mean(mean(raw3D))), bslneRng);

glo\_mc\_pct = getPercentChange\_1D(squeeze(mean(mean(mc3D))), bslneRng);

figure;

set(gca,'fontsize',15);

title(strcat('Global Mean - ', Exp\_labels{M\_num}{pts}));

%

% patch([baseline\_Ranges\_all{pts}(1) baseline\_Ranges\_all{pts}(end) baseline\_Ranges\_all{pts}(end) baseline\_Ranges\_all{pts}(1)]/60, ...

% [-100 -100 100 100], bsln\_color, 'FaceAlpha', 0.05, 'EdgeColor', 'none');

%

% patch([Stim\_ranges{1}(end) Stim\_ranges{6}(end) Stim\_ranges{6}(end) Stim\_ranges{1}(end)]/60, ...

% [-100 -100 100 100], stim\_color, 'FaceAlpha', 0.05, 'EdgeColor', 'none');

hold on;

plot(next\_times/60, glo\_raw\_pct, '-b');

plot(next\_times/60, glo\_mc\_pct, '-r');

grid;

xline((baseline\_Ranges\_all{pts}(1))/60,'--k',{'baseline'}, 'fontsize',15);

xline(baseline\_Ranges\_all{pts}(end)/60,'--r',{'MK-injection'}, 'fontsize',15);

ylabel('% pD signal change');

xlabel('t (min)');

yline(0,'-k',{''});

ylim([-20 20]);

xlim([3, 60]);

legend( 'meanGlobal-Raw','meanGlobal-Filtered', 'Location','best');

hold off;

set(gcf, 'Position', [50, 50, 900, 550]);

end

%% Auxiliary functions

function dataCells\_filtered = preProcess\_mkDoseData(dataCell, params, strt, isTempRng)

dataCells\_filtered = {};

for i = strt : size(dataCell, 2)

miceData = {};

mc\_3D\_files = {}; raw\_3D\_files = {};

nextPatient = dataCell{i};

for a = 1 : size(nextPatient, 2)

nextFN = nextPatient{a};

nextData = h5read(nextFN, '/Data');

next\_times = h5read(nextFN, '/acqMetaData/time');

nextData\_raw\_perm = permute(squeeze(nextData), [2 1 3]);

if ~exist('isTemp','var')

nextData\_mc = fUSi\_animalData\_preProcessing(nextData, params);

nextScanFN = strcat(nextFN(1 : end - 5), '\_HF\_mc.scan');

h5write(nextScanFN, '/Data', double(nextData\_mc));

else

bslnRng = isTempRng;

baselineTemplate = mean(nextData\_raw\_perm(:, :, bslnRng), 3);

nextData\_mc = fUSi\_animalData\_preProcessing(nextData, params, baselineTemplate);

nextScanFN = strcat(nextFN(1 : end - 5), '\_HF\_mc.scan');

h5write(nextScanFN, '/Data', double(nextData\_mc));

end

nextData\_mc\_perm = permute(squeeze(nextData\_mc), [2 1 3]);

raw\_3D\_files{a} = double(nextData\_raw\_perm);

mc\_3D\_files{a} = double(nextData\_mc\_perm);

end

miceData{1} = raw\_3D\_files;

miceData{2} = mc\_3D\_files;

dataCells\_filtered{i} = miceData;

end

end

function pct = getPercentChange\_1D(dat, brng)

bsln = mean(dat(brng));

ypct = 100\*((dat - bsln) / bsln);

pct = ypct;

end