% control cells threshold 20

clear all; close all; clc;

X=2;

width = 1.5\*4; % Width in inches

height = 1.5\*3; % Height in inches

alw = X\*1; % AxesLineWidth

fsz = X\*8; % Fontsize

tsz = X\*2; % text Fontsize sa

lw = X\*1; % LineWidth

msz = 3\*X;

% Load control Data Files

Tag1 =[];Tag2 =[];Tag3 =[];Date=[];

addpath('/cntrl parse/');

dirname='./cntrl parse/';

files=dir([dirname,'\*.mat']);

for ii=1:1:length(files) %loop over files

AD(ii) = load([dirname,files(ii).name]);

end

kk = 1;

for ii=1:1:length(files) %loop over files

branches = AD(ii).D.branches;% number of branches

AVGbranchesLengthe = AD(ii).D.AvgBranchelength;% Avg branch length

Pixelsize(ii)= (AD(ii).D.Pixelsize(1,1));% pixel size

AreaSize(ii) = AD(ii).D.analysedArea;% Area in micrometers squared

Junctions = AD(ii).D.junctions;

Correct\_angles = AD(ii).D.DirectionEngelcorrect;

Amplitud\_angle = AD(ii).D.DirectionAMP;

for jj = 1:1:size(AD(ii).D.branches,1)

skeletonlength(jj,ii) = Pixelsize(ii).\*(branches(jj,1))\*(AVGbranchesLengthe(jj,1)); % length in micrometers

total\_skeletonlength = sum(skeletonlength);% total skeletal length in um per cell

Nor\_total\_skeletonlength (1,ii) = (total\_skeletonlength(1,ii))./(AreaSize(ii));% um skeleton per cell per um2 cell area

NumJunctions(jj,ii) = (Junctions(jj,1));

total\_NumJunctions = sum(NumJunctions);% junctions per cell

Nor\_total\_NumJunctions (1,ii) = (total\_NumJunctions(1,ii))./(AreaSize(ii));% Junctions per cell per um2 cell area

end

for jj = 1:1:size(AD(ii).D.DirectionEngelcorrect,1)

Allcells\_Correct\_angles(jj,ii) =Correct\_angles(jj,1);

Nor\_Allcells\_Amplitud\_angle(jj,ii) =Amplitud\_angle(jj,1).\*(Nor\_total\_skeletonlength (1,ii));% amplitude at each angle multiply by nor skeletal size(um/um^2)

end

%all raw data

Resultsctl.ttSize = Nor\_total\_skeletonlength; % um skeleton per cell per um2 cell area

Resultsctl.Junctions = Nor\_total\_NumJunctions; % Junctions per cell per um2 cell area

Resultsctl.direction =Allcells\_Correct\_angles;% degree

Resultsctl.directionAmp =Nor\_Allcells\_Amplitud\_angle;% amplitude at each angle multiply by nor skeletal size(um/um^2)

% means all raw data

Resultsctl.ttSizeMean=nanmean(Resultsctl.ttSize,2);

Resultsctl.JunctionsMean = nanmean(Resultsctl.Junctions,2);

Resultsctl.directionAmpMean = nanmean(Resultsctl.directionAmp,2);

%SEM all data

Resultsctl.ttSizeSEM = [std(Resultsctl.ttSize,0,2)./sqrt(size(Resultsctl.ttSize,2))];

Resultsctl.directionAmpSEM = [std(Resultsctl.directionAmp,0,2)./sqrt(size(Resultsctl.directionAmp,2))];

Resultsctl.JunctionsSEM = [std(Resultsctl.Junctions,0,2)./sqrt(size(Resultsctl.Junctions,2))];

save(['directory’,'Resultsctl'],'Resultsctl');

%%

%%

% HFpEF cells threshold 20

clear all; close all; clc;

X=2;

width = 1.5\*4; % Width in inches

height = 1.5\*3; % Height in inches

alw = X\*1; % AxesLineWidth

fsz = X\*8; % Fontsize

tsz = X\*2; % text Fontsize sa

lw = X\*1; % LineWidth

msz = 3\*X;

% Load control Data Files

Tag1 =[];Tag2 =[];Tag3 =[];Date=[];

addpath('/KO parse/');

dirname='./KO parse/';

files=dir([dirname,'\*.mat']);

for ii=1:1:length(files) %loop over files

AD(ii) = load([dirname,files(ii).name]);

end

kk = 1;

for ii=1:1:length(files) %loop over files

branches = AD(ii).D.branches;% number of branches

AVGbranchesLengthe = AD(ii).D.AvgBranchelength;% Avg branch length

Pixelsize(ii)= (AD(ii).D.Pixelsize(1,1));% pixel size

AreaSize(ii) = AD(ii).D.analysedArea;% Area in micrometers squared

Junctions = AD(ii).D.junctions;

Correct\_angles = AD(ii).D.DirectionEngelcorrect;

Amplitud\_angle = AD(ii).D.DirectionAMP;

for jj = 1:1:size(AD(ii).D.branches,1)

skeletonlength(jj,ii) = Pixelsize(ii).\*(branches(jj,1))\*(AVGbranchesLengthe(jj,1)); % length in micrometers

total\_skeletonlength = sum(skeletonlength);% total skeletal length in um per cell

Nor\_total\_skeletonlength (1,ii) = (total\_skeletonlength(1,ii))./(AreaSize(ii));% um sekelton per cell per um2 cell area

NumJunctions(jj,ii) = (Junctions(jj,1));

total\_NumJunctions = sum(NumJunctions);% junctions per cell

Nor\_total\_NumJunctions (1,ii) = (total\_NumJunctions(1,ii))./(AreaSize(ii));% Junctions per cell per um2 cell area

end

for jj = 1:1:size(AD(ii).D.DirectionEngelcorrect,1)

Allcells\_Correct\_angles(jj,ii) =Correct\_angles(jj,1);

Nor\_Allcells\_Amplitud\_angle(jj,ii) =Amplitud\_angle(jj,1).\*(Nor\_total\_skeletonlength (1,ii));% amplitude at each angle multiply by nor skeletal size(um/um^2)

end

=

end

%all raw data

ResultsKO.ttSize = Nor\_total\_skeletonlength; % um skeleton per cell per um2 cell area

ResultsKO.Junctions = Nor\_total\_NumJunctions; % Junctions per cell per um2 cell area

ResultsKO.direction =Allcells\_Correct\_angles;% degree

ResultsKO.directionAmp =Nor\_Allcells\_Amplitud\_angle;% amplitude at each angle multiply by nor skeletal size(um/um^2)

% means all raw data

ResultsKO.ttSizeMean=nanmean(ResultsKO.ttSize,2);

ResultsKO.JunctionsMean = nanmean(ResultsKO.Junctions,2);

ResultsKO.directionAmpMean = nanmean(ResultsKO.directionAmp,2);

%SEM all data

ResultsKO.ttSizeSEM = [std(ResultsKO.ttSize,0,2)./sqrt(size(ResultsKO.ttSize,2))];

ResultsKO.directionAmpSEM = [std(ResultsKO.directionAmp,0,2)./sqrt(size(ResultsKO.directionAmp,2))];

ResultsKO.JunctionsSEM = [std(ResultsKO.Junctions,0,2)./sqrt(size(ResultsKO.Junctions,2))];

save(['directory','ResultsKO'],'ResultsKO');

%%

%Figs ctl Vs. HFpEF

clear all; close all; clc;

addpath('~/Dropbox/Code/Matlab/General/ExportFig');

X=2;

width = 1.5\*4; % Width in inches

height = 1.5\*3; % Height in inches

alw = X\*1; % AxesLineWidth

fsz = X\*8; % Fontsize

tsz = X\*2; % text Fontsize sa

lw = X\*1; % LineWidth

msz = 3\*X;

addpath('all parse');

dirname='./all parse/';

files=dir([dirname,'\*.mat']);

for ii = 1:1:size(files,1)

load([dirname,files(ii).name]);

end

% Figure TT amp at each direction

figure(301)

plot(Resultsctl.direction, Resultsctl.directionAmp, 'o', 'MarkerSize',0.5, 'MarkerEdgeColor' ,[0.6 0.6 0.6],'MarkerFaceColor',[0.6 0.6 0.6])

hold on

plot(ResultsKO.direction, ResultsKO.directionAmp, 'o', 'MarkerSize',0.5, 'MarkerEdgeColor' ,[1 0.8 0.8],'MarkerFaceColor',[1 0.8 0.8])

hold on

errorbar(ResultsKO.direction(:,1), ResultsKO.directionAmpMean,ResultsKO.directionAmpSEM,'ok','MarkerEdgeColor','r','markerfacecolor','r','markersize',1.5, 'CapSize',2, 'LineWidth', 0.1, 'Color','r')

hold on

errorbar(Resultsctl.direction(:,1), Resultsctl.directionAmpMean,Resultsctl.directionAmpSEM,'ok','MarkerEdgeColor','k','markerfacecolor','k','markersize',1.5, 'CapSize',2, 'LineWidth', 0.1, 'Color','k')

hold on

ylabel('Tubular length (\mum / \mum^{2})')

xlabel('Direction (^{ o })')

ylim([0 0.01]);

xlim([-45 135]);

set(gca,'xtick',[-30:30:120])

set(gcf,'Units', 'inches','PaperSize', [width height]);

set(gca,'color','none','fontsize',fsz,'fontname','Helvetica','tickdir','out','yminortick','off','box','off','xminortick','off');

set(gcf,'PaperOrientation','Portrait','PaperUnits','normalized','PaperPosition', [0 0 1 1]);

FigName = ['TT amp at each direction\_control'];

print(gcf, '-dpdf',['./Figures\_TT/',FigName]);

% Figure control TT length

figure(22)

hold on;

bar([1],Resultsctl.ttSizeMean,'k','EdgeColor','k','LineWidth', 1.5);

hold on;

bar([2],ResultsKO.ttSizeMean,'r','EdgeColor','k','LineWidth', 1.5);

hold on;

errorbar([1],Resultsctl.ttSizeMean,Resultsctl.ttSizeSEM,'ok','markerfacecolor','k','markersize',1.5\*msz, 'CapSize',10, 'LineWidth', 1);

hold on;

errorbar([2],ResultsKO.ttSizeMean,ResultsKO.ttSizeSEM,'ok','markerfacecolor','r','markersize',1.5\*msz, 'CapSize',10, 'LineWidth', 1);

hold on;

scatter(ones(size(Resultsctl.ttSize(1,:))).\*(1+(rand(size(Resultsctl.ttSize(1,:)))-0.5)/2),Resultsctl.ttSize(1,:),10, 'MarkerEdgeColor' ,[0.6 0.6 0.6],'MarkerFaceColor',[0.6 0.6 0.6]);

hold on;

scatter(ones(size(ResultsKO.ttSize(1,:))).\*(2+(rand(size(ResultsKO.ttSize(1,:)))-0.5)/2),ResultsKO.ttSize(1,:),10, 'MarkerEdgeColor' ,[1 0.8 0.8],'MarkerFaceColor',[1 0.8 0.8]);

ylabel('total tubular length (\mum / \mum^{2})')

ylim([0 .4])

lh = legend('control','KO ', 'location', 'northoutside','orientation', 'horizontal');

set(gcf,'Units', 'inches','PaperSize', [width height]);

set(gca,'color','none','fontsize',fsz,'fontname','Helvetica','tickdir','out','yminortick','off','box','off','xminortick','off');

set(gcf,'PaperOrientation','Portrait','PaperUnits','normalized','PaperPosition', [0 0 1 1]);

set(lh,'fontsize',6);

FigName = ['skeleton size\_cont'];

print(gcf, '-dpdf',['./Figures\_TT/',FigName]);

% Figure control junctions

figure(23)

hold on;

bar([1], Resultsctl.JunctionsMean,'k','EdgeColor','k','LineWidth', 1.5);

hold on;

bar([2], ResultsKO.JunctionsMean,'r','EdgeColor','r','LineWidth', 1.5);

hold on;

errorbar([1], Resultsctl.JunctionsMean, Resultsctl.JunctionsSEM,'ok','markerfacecolor','k','markersize',1.5\*msz, 'CapSize',10, 'LineWidth', 1);

hold on;

errorbar([2], ResultsKO.JunctionsMean, ResultsKO.JunctionsSEM,'ok','markerfacecolor','r','markersize',1.5\*msz, 'CapSize',10, 'LineWidth', 1);

hold on;

scatter(ones(size(Resultsctl.Junctions(1,:))).\*(1+(rand(size(Resultsctl.Junctions(1,:)))-0.5)/2),Resultsctl.Junctions(1,:),10, 'MarkerEdgeColor' ,[0.6 0.6 0.6],'MarkerFaceColor',[0.6 0.6 0.6]);

hold on;

scatter(ones(size(ResultsKO.Junctions(1,:))).\*(2+(rand(size(ResultsKO.Junctions(1,:)))-0.5)/2),ResultsKO.Junctions(1,:),10, 'MarkerEdgeColor' ,[1 0.8 0.8],'MarkerFaceColor',[1 0.8 0.8]);

ylabel('junctions / \mum^{2}')

ylim([0 1])

lh = legend('control','KO ', 'location', 'northoutside','orientation', 'horizontal');

set(gcf,'Units', 'inches','PaperSize', [width height]);

set(gca,'color','none','fontsize',fsz,'fontname','Helvetica','tickdir','out','yminortick','off','box','off','xminortick','off');

set(gcf,'PaperOrientation','Portrait','PaperUnits','normalized','PaperPosition', [0 0 1 1]);

set(lh,'fontsize',6);

FigName = ['Junctions\_cont'];

print(gcf, '-dpdf',['./Figures\_TT/',FigName]);