function [prop\_cl cellarea] = get\_prop\_clusters(file, flg, th)

%% load in parameters for graphical representation

cm=1/25:1/25:1;cm=cm';

pe=[0 0 0];

pe=cat(1, pe, [cm cm/2 zeros(25, 1)]);

pe=cat(1, pe, [ones(25, 1) 0.5+cm/2 zeros(25, 1)]);

pe=cat(1, pe, [ones(25, 1) ones(25, 1) cm]);

colormap(pe)

%% load data. Only analyze clusters with ≥ 50 localizations

load(file, 'prop\_50', 'DB', 'selpts')

ind\_DB50 = ismember(DB, prop\_50.ID);

pts\_DB = selpts(ind\_DB50,:);

%% calculate area of the cell

j = boundary(selpts(:,1), selpts(:,2),0.8);

cellarea = polyarea(selpts(j,1), selpts(j,2));

%% create empty prop\_cl table

prop\_cl = table('Size', [max(DB), 10], 'VariableTypes',{'single','single','single','single','single','single', 'single', 'single', 'single', 'string'}, 'VariableNames', {'ID', 'nrpts','center', 'area', 'perimeter', 'density', 'ecc', 'distance', 'classification\_model', 'type'});

%% prepare to plot data

if flg

sx = round(512\*50\*0.107);

sy = sx;

BW = zeros(sy, sx);

BWcp=BW; BWfcl=BW;

end

%% For every cluster with ≥ 50 localizations, calculate cluster properties and classify as a pit or a lattice.

for i = 1:size(prop\_50,1)

prop\_cl.ID(i) = prop\_50.ID(i);

pts\_cl = selpts(find(DB == prop\_50.ID(i)), :);

bb = boundary(pts\_cl(:,1), pts\_cl(:,2), 0.6); % find edges cluster

% calculate cluster properties

prop\_cl.area(i) = polyarea(pts\_cl(bb,1), pts\_cl(bb,2));

prop\_cl.nrpts(i) = length(pts\_cl);

prop\_cl.density(i) = length(pts\_cl)/prop\_cl.area(i);

[A, c] = MinVolEllipse (pts\_cl', .01);

prop\_cl.center(i, 1:2) = c';

[~, Q, ~] = svd(A); r1 = 1/sqrt(Q(1,1)); r2 = 1/sqrt(Q(2,2));

prop\_cl.ecc(i) = axes2ecc(max(r2,r1), min(r2,r1));

prop\_cl.perimeter(i) = perimeter(polyshape(pts\_cl(bb,1), pts\_cl(bb,2)));

ind = ismember(pts\_DB, pts\_cl, 'rows');

DD = pdist2(pts\_cl(bb,:),pts\_DB(~ind,:));

prop\_cl.distance(i) = min(DD(:));

% apply cluster classification model on every cluster, descide if cluster is a classical CCS (pit) or alternative CCS (lattice)

prop\_cl.classification\_model(i)=((prop\_50.nrpts(i)./1000).\*prop\_cl.area(i).\*prop\_cl.perimeter(i).\*prop\_cl.ecc(i))./(min(DD(:)));

if prop\_cl.classification\_model(i) < th; prop\_cl.type(i) = 'pit'; else; prop\_cl.type(i) = 'lattice'; end

% for plotting the figure

if flg

ROI = poly2mask(50\*pts\_cl(bb,1), 50\*pts\_cl(bb,2), sx, sy);

BW = BW + ROI.\*prop\_cl.classification\_model(i);

end

end

prop\_cl(prop\_cl.ID(:,1)==0,:)=[];

disp('done')

disp('BW image calculated!')

%% plot figure with 2 pannels. Left side = classified image (white = pits, pink = FCLs). Right side = super-resolution image for comparison.

if flg

% calculate classified image

BWcp(BW(:)<th & BW(:)>0)=1; BWfcl(BW(:)>=th)=1;

BW\_RGD=zeros(size(BW,1), size(BW,2), 3);

BW\_RGD(:,:,1)=BWcp+BWfcl; BW\_RGD(:,:,2)=BWcp+0.0736\*BWfcl; BW\_RGD(:,:,3)=BWcp+0.6471\*BWfcl;

h=figure; set(gcf, 'position', [300 300 900 450], 'color', 'w');

subplot(1,2,1);image(BW\_RGD); axis tight equal off; title('white = pits; pink = FLCs');

% calculate super-resolution image

mx = round((512/0.05)\*.107);

cod = floor(selpts(:,1:2)/0.05)+1;

cod(cod(:,1)<1, :)=[];

cod(cod(:,2)<1, :)=[];

hst = accumarray(cod(:, [2, 1]), 1, [mx, mx]);

hst=single(hst);

hsts=imgaussfilt(hst,1);

subplot(1,2,2);imagesc(hsts, [0 10]); colormap(pe)

axis equal tight

drawnow

pause (0.01)

outfile = [file(1:strfind(file, '.mat')-1) '\_cluster\_classification.png'];

print(outfile, '-dpng')

close(h)

end