%% set file path: manually select folder. Load data.

fp = uigetdir;

cd(fp)

list=dir('\*\_DBscan.mat');

list={list.name};

th = 0.5035; % global threshold of the classification model. Set based on 1247 classical and 1042 alternative CCSs that were manually selected, ensuring that maximally 10% of the classical CCSs are misclassified

flg = 1;

%% analyse files on by one

celldata = table('Size', [size(list, 2), 24], 'VariableTypes',{'string','single','single','single','single','single','single','single','single','single','single', 'single', 'single','single','single','single','single','single','single','single','single','single','single','string'}, ...

'VariableNames', {'filename','nrFCL\_nrclusters','nrCP\_nrclusters','nrFCL\_cell','nrCP\_cell','nrFCL\_nrCP','areaFCL\_clathrinarea','areaCP\_clathrinarea','areaFCL\_cell','areaCP\_cell','areaFCL\_areaCP', 'nrptsFCL\_totalnrpts','nrptsCP\_totalnrpts','nrptsFCL\_cell','nrptsCP\_cell','nrptsFCL\_nrptsCP','totalnrcluster','totalnrcluster\_cellarea','cellarea','clathrinarea','clathrinarea\_cellarea','totalnrpts','totalnrpts\_cellarea','category'});

for f=1:size(list, 2)

fn = list{1,f};

file = [fp filesep fn];

[prop\_cl, cellarea] = get\_prop\_clusters(file,1, th); % runs separate MATLAB\_CODE\_get\_prop\_clusters, which calculates cluster properties, classifies each cluster as pit or lattice, and plots the result in an image

if flg ==1

save([file(1:strfind([fp fn], '.mat')) '\_th\_' num2str(th) '\_individual\_clusters\_classified.mat'], 'prop\_cl', 'cellarea');

end

ind\_CP = strcmp(prop\_cl.type(:), 'pit');

ind\_FCL = strcmp(prop\_cl.type(:), 'lattice');

celldata.filename(f) = fn;

% per cell: number of clusters

celldata.nrFCL\_nrclusters(f) = sum(ind\_FCL)./size(prop\_cl, 1);

celldata.nrCP\_nrclusters(f) = sum(ind\_CP)./size(prop\_cl, 1);

celldata.nrFCL\_cell(f) = sum(ind\_FCL)./cellarea;

celldata.nrCP\_cell(f) = sum(ind\_CP)./cellarea;

celldata.nrFCL\_nrCP(f) = sum(ind\_FCL)./sum(ind\_CP);

% per cell: area of clusters

celldata.areaFCL\_clathrinarea(f) = sum(prop\_cl.area(ind\_FCL))./sum(prop\_cl.area(:));

celldata.areaCP\_clathrinarea(f) = sum(prop\_cl.area(ind\_CP))./sum(prop\_cl.area(:));

celldata.areaFCL\_cell(f) = sum(prop\_cl.area(ind\_FCL))./cellarea;

celldata.areaCP\_cell(f) = sum(prop\_cl.area(ind\_CP))./cellarea;

celldata.areaFCL\_areaCP(f) = sum(prop\_cl.area(ind\_FCL))./sum(prop\_cl.area(ind\_CP));

% per cell: nrpts of clusters

celldata.nrptsFCL\_totalnrpts(f) = sum(prop\_cl.nrpts(ind\_FCL))./sum(prop\_cl.nrpts(:));

celldata.nrptsCP\_totalnrpts(f) = sum(prop\_cl.nrpts(ind\_CP))./sum(prop\_cl.nrpts(:));

celldata.nrptsFCL\_cell(f) = sum(prop\_cl.nrpts(ind\_FCL))./cellarea;

celldata.nrptsCP\_cell(f) = sum(prop\_cl.nrpts(ind\_CP))./cellarea;

celldata.nrptsFCL\_nrptsCP(f) = sum(prop\_cl.nrpts(ind\_FCL))./sum(prop\_cl.nrpts(ind\_CP));

% per cell: additional data

celldata.totalnrcluster(f) = size(prop\_cl, 1);

celldata.totalnrcluster\_cellarea(f) = size(prop\_cl, 1)./cellarea;

celldata.cellarea(f) = cellarea;

celldata.clathrinarea(f) = sum(prop\_cl.area(:));

celldata.clathrinarea\_cellarea(f) = sum(prop\_cl.area(:))./cellarea;

celldata.totalnrpts(f) = sum(prop\_cl.nrpts(:));

celldata.totalnrpts\_cellarea(f) = sum(prop\_cl.nrpts(:))./cellarea;

end

save('all\_files\_cell\_classification.mat','celldata')