//macro to quantify the heartbeat in 10x objective RFP images from imaging machine

//"A001", "A002", "A003","A004", "A005", "A006", "A007","A008", "A009", "A010", "A011","A012", "B001", "B002", "B003", "B004", "B005", "B006","B007", "B008", "B009", "B010", "B011", "B012", "C001", "C002", "C003", "C004", "C005", "C006", "C007", "C008", "C009", "C010", "C011", "C012", "D001", "D002", "D003", "D004", "D005", "D006", "D007", "D008", "D009", "D010", "D011", "D012", "E001", "E002", "E003", "E004", "E005", "E006", "E007", "E008", "E009", "E010", "E011", "E012", "F001","F002","F003", "F004", "F005", "F006", "F007", "F008","F009", "F010", "F011", "F012", "G001", "G002", "G003", "G004", "G005", "G006", "G007", "G008", "G009", "G010", "G011", "G012", "H001", "H002", "H003", "H004", "H005", "H006", "H007", "H008", "H009", "H010", "H011", "H012");

Table.create("heart\_beat"); //number of heartbeat are stored in this table

Table.create("X\_heart\_beat");

wells = newArray ("A001", "A002", "A003","A004", "A005", "A006", "A007","A008", "A009", "A010", "A011","A012", "B001", "B002", "B003", "B004", "B005", "B006","B007", "B008", "B009", "B010", "B011", "B012", "C001", "C002", "C003", "C004", "C005", "C006", "C007", "C008", "C009", "C010", "C011", "C012", "D001", "D002", "D003", "D004", "D005", "D006", "D007", "D008", "D009", "D010", "D011", "D012", "E001", "E002", "E003", "E004", "E005", "E006", "E007", "E008", "E009", "E010", "E011", "E012", "F001","F002","F003", "F004", "F005", "F006", "F007", "F008","F009", "F010", "F011", "F012", "G001", "G002", "G003", "G004", "G005", "G006", "G007", "G008", "G009", "G010", "G011", "G012", "H001", "H002", "H003", "H004", "H005", "H006", "H007", "H008", "H009", "H010", "H011", "H012");

n = lengthOf(wells);

for (i = 0; i <n ; i++){

//setBatchMode("hide");

well = wells[i];

run("Image Sequence...", "open=[name\_of\_the fowlder /] file="+well+" sort");

rename("Image");

run("Duplicate...", "use");

//run("Brightness/Contrast...");

resetMinAndMax();

rename("Image\_T");

run("Image Sequence...", "open=[ name\_of\_the fowlder /template.tif] file=template sort"); //open template image for cropping

rename("template");

//cropping

run("Template Matching Image", "template=template image=Image\_T rotate=[] matching\_method=[Normalised cross-correlation] number\_of\_objects=1 score\_threshold=0.7 maximal\_overlap=0.6 add\_roi");

selectWindow("Image");

roiManager("Select", 0);

run("Duplicate...", "duplicate range=1-300 use");

rename("Image\_2");

//run("Brightness/Contrast...");

resetMinAndMax();

run("Median...", "radius=2 stack");

roiManager("deselect");

roiManager("delete");

//find standard deviation

n=nSlices;

for (k = 1; k < n; k++) {

setSlice(k);

//run("Find Maxima...", "prominence=200 strict exclude output=[Point Selection]");

run("Set Measurements...", "area mean standard modal min centroid center perimeter bounding fit shape feret's integrated median skewness kurtosis area\_fraction stack display redirect=None decimal=3");

run("Measure");

//roiManager("add");

}

newImage("Untitled", "16-bit black", 800, 500, 1);

m=nResults;

//reproduce standard deviation graph

for (j = 0; j < m-1; j++) {

a=getResult("StdDev", j);

b=getResult("StdDev", j+1);

makeLine((j+10)\*2, a\*2, (j+11)\*2, b\*2);

roiManager("add");

}

//find max and min of the Standard deviation to exclude non beating heart

max=Table.getColumn("StdDev", "Results");

indices\_max = Array.findMaxima(max, 1);

imax = indices\_max[0]; // keep the largest

maxima= max[imax];

min=Table.getColumn("StdDev");

indices\_min = Array.findMinima(min, 1);

imin = indices\_min[0];// just keep the largest

minima= min[imin];

mean= (maxima)-(minima);

print(mean);

if (mean<5){

print("[heart\_beat]",well +","+ 0);

}

else {

middle= (minima\*2)+mean;

//print(middle);

//make horizontal line

makeLine(10, middle, 700, middle);

roiManager("add");

//create triangular shapes

roiManager("Fill");

run("Make Binary");

run("Fill Holes");

run("Options...", "iterations=1 count=1 black edm=8-bit do=Erode");

run("Analyze Particles...", "size=0-Infinity show=Masks display clear add");

res=getValue("results.count");

final\_res=res/2;

print("[heart\_beat]",well +","+ final\_res);

res=getValue("results.count");

for (p = 0; p < res; p++) {

Y=getResult("Y", p);

if (Y>middle) {

x=getResult("XM", p);

print("[X\_heart\_beat]",well +","+ x);

}

}

}

close("\*");

run("Clear Results");

roiManager("deselect");

roiManager("delete");

}

//remember to save the tables!!!!