//Coloc\_Salk

run("Clear Results");

roiManager("reset");

dir1 = getDirectory("Choose Source Directory for Channel 1 ");

dir2 = getDirectory("Choose Source Directory for Channel 2 ");

dir3 = getDirectory("Choose Destination Directory for segmented Channel 1");

dir4 = getDirectory("Choose Destination Directory for segmented Channel 2");

list1 = getFileList(dir1);

list2 = getFileList(dir2);

/// création tableau de résultats

title1 = "Resultat:";

title2 = "["+title1+"]";

f=title2;

run("New... ", "name="+title2+" type=Table");

print(f,"\\Headings:PhotoA\tPhotoB\tPearson\tNb d'objets total A\tNb d'objets total B\tNb objets coloc A vs B\tNb objets coloc B vs A\tSigma\tMethod\tTreshold");

/// boite de dialogue pour trouver les parametres

Dialog.create("Select Method");

Dialog.addNumber("Min Particle Size (pixels^2)",2, 1,3,"pixels");

Dialog.addNumber("Max Particle Size (pixels^2)",99, 1,3,"pixels");

Dialog.addNumber("Sigma",3, 1,3, "Radius");

Dialog.addChoice("Method", newArray("DoG", "Laplacian"));

Dialog.addChoice("Filter", newArray("Triangle", "Otsu", "Huang"));

Dialog.addCheckbox("Crop needed?", true);

Dialog.show();

minSize = Dialog.getNumber();

maxSize = Dialog.getNumber();

sigma = Dialog.getNumber();

Method = Dialog.getChoice();;

Filtertype = Dialog.getChoice();

Krop=Dialog.getCheckbox();

//setBatchMode(true);

for (i=0; i<list1.length; i++) {

roiManager("reset");

showProgress(i+1, list1.length);

if (Krop==true)

{

setBatchMode(false);

open(dir1+list1[i]);

name1= getTitle();

rename("channel1");

open(dir2+list2[i]);

name2= getTitle();

rename("channel2");

selectWindow("channel1");

setTool("polygon");

waitForUser("dessiner contour zone");

run("ROI Manager...");

roiManager("Add");

run("Crop");

selectWindow("channel2");

roiManager("Select",0);

run("Crop");

setBatchMode(true);

}

else

{

open(dir1+list1[i]);

name1= getTitle();

rename("channel1");

open(dir2+list2[i]);

name2= getTitle();

rename("channel2");

}

JACoP\_PearsonToRT("channel1", "channel2");

pearson= getResult("Pearson",0);

SegPointPoint("channel1", "channel2");

close("channel1");

close("channel2");

list3 = getFileList(dir3);

list4 = getFileList(dir4);

open(dir3+list3[i]);

name1= getTitle();

rename("Thr Channel1");

open(dir4+list4[i]);

name2= getTitle();

rename("Thr Channel2");

run("JACoP ");

waitForUser("Lauch JAcOP obj based");

//run("JACoP ", "imga=[Thr Channel1] imgb=[Thr Channel2] thra=2 thrb=2 objdist="+minSize+"-"+maxSize+"-158.57142857142858-396.4285714285715-true-false-false");

logContent=call("ij.IJ.getLog");

subA=substring(logContent, lastIndexOf(logContent, "Image A:")+8, lastIndexOf(logContent, "Image B:")-1);

subB=substring(logContent, lastIndexOf(logContent, "Image B:")+8, lastIndexOf(logContent, "\n"));

AvsB=substring(subA, 0, lastIndexOf(subA, "centre"));

BvsA=substring(subB, 0, lastIndexOf(subB, "centre"));

ToTA=substring(subA, lastIndexOf(subA, "of ")+3);

ToTB=substring(subB, lastIndexOf(subB, "of ")+3);

print(f, name1+"\t"+ name2 +"\t"+ pearson +"\t"+ ToTA +"\t"+ ToTB +"\t"+ AvsB +"\t"+ BvsA +"\t"+sigma+"\t"+Method+"\t"+Filtertype);

waitForUser("Close JAcOP");

run("Clear Results");

run("Close All");

}

setBatchMode(false);

selectWindow("Resultat:");

function JACoP\_PearsonToRT(channel1, channel2){

run("JACoP ", "imga="+channel1+" imgb="+channel2+" pearson");

logContent=call("ij.IJ.getLog");

pearson=substring(logContent, lastIndexOf(logContent, "r=")+2, lastIndexOf(logContent, "\n"));

setResult("channel1", nResults, channel1);

setResult("channel2", nResults-1, channel2);

setResult("Pearson", nResults-1, pearson);

}

function SegPointPoint(channel1, channel2){

run("Set Measurements...", "area display redirect=None decimal=5");

run("Set Scale...", "distance=0 known=0 pixel=1 unit=pixel");

// si methode Difference of gaussian

if (Method=="Dog")

{

//Difference of gaussian

selectWindow("channel1");

run("8-bit");

run("Subtract Background...", "rolling=5");

run("Duplicate...", "title=DoG-3");

run("Duplicate...", "title=DoG-1");

selectWindow("DoG-3");

run("Gaussian Blur...", "sigma="+sigma+"");

selectWindow("DoG-1");

run("Gaussian Blur...", "sigma=1");

rename("DoG-1");

imageCalculator("Subtract create", "DoG-1","DoG-3");

rename("DoG Channel1");

close("DoG-1");

close("DoG-3");

///application choix du filtre

if (Filtertype=="Otsu")

setAutoThreshold("Otsu dark");

if (Filtertype=="Triangle")

setAutoThreshold("Triangle dark");

if (Filtertype=="Huang")

setAutoThreshold("Huang dark");

run("Convert to Mask");

saveAs("Tiff", dir3+list1[i]);

run("Close");

selectWindow("channel2");

run("8-bit");

run("Subtract Background...", "rolling=5");

run("Duplicate...", "title=DoG-3");

run("Duplicate...", "title=DoG-1");

selectWindow("DoG-3");

run("Gaussian Blur...", "sigma="+sigma+"");

selectWindow("DoG-1");

run("Gaussian Blur...", "sigma=1");

rename("DoG-1");

imageCalculator("Subtract create", "DoG-1","DoG-3");

rename("DoG Channel2");

close("DoG-1");

close("DoG-3");

///application choix du filtre

if (Filtertype=="Otsu")

setAutoThreshold("Otsu dark");

if (Filtertype=="Triangle")

setAutoThreshold("Triangle dark");

if (Filtertype=="Huang")

setAutoThreshold("Huang dark");

run("Convert to Mask");

saveAs("Tiff", dir4+list2[i]);

run("Close");

}

if (Method=="Laplacian")

{

//Laplacian

selectWindow("Channel1");

run("8-bit");

run("Subtract Background...", "rolling=5");

run("FeatureJ Laplacian", "compute smoothing="+sigma+"");

///application choix du filtre

if (Filtertype=="Otsu")

setAutoThreshold("Otsu");

if (Filtertype=="Triangle")

setAutoThreshold("Triangle");

if (Filtertype=="Huang")

setAutoThreshold("Huang");

run("Convert to Mask");

saveAs("Tiff", dir3+list1[i]);

run("Close");

selectWindow("Channel2");

run("8-bit");

run("Subtract Background...", "rolling=5");

run("FeatureJ Laplacian", "compute smoothing="+sigma+"");

///application choix du filtre

if (Filtertype=="Otsu")

setAutoThreshold("Otsu");

if (Filtertype=="Triangle")

setAutoThreshold("Triangle");

if (Filtertype=="Huang")

setAutoThreshold("Huang");

run("Convert to Mask");

saveAs("Tiff", dir4+list2[i]);

run("Close");

}

}