## Plugin code

|  |
| --- |
| imageTitle=getTitle();  dir = getDirectory("image");  makeRectangle(20,20,472,472);  run("Crop");  run("Split Channels");  selectWindow(imageTitle+" (red)");  run("Duplicate...", "duplicate");  run("8-bit");  run("Invert", "stack");  setAutoThreshold("Huang");  setOption("BlackBackground", false);  run("Convert to Mask", "method=Huang background=Light calculate");  run("Object Counter3D", "threshold=128 slice=48 min=2 max=25165824 particles dot=3 font=12");  total\_points = nResults;  IJ.renameResults("Particles");  selectWindow(imageTitle+" (green)");  run("Close");  selectWindow(imageTitle+" (blue)");  waitForUser("Select the right slice");  run("Duplicate...", " ");  run("Gaussian Blur...", "sigma=5");  rename("for\_ws");  run("Duplicate...", " ");  waitForUser("Draw points in each cell");  waitForUser("Draw points in all parts of background");  run("Invert");  setAutoThreshold("Default dark");  run("Threshold...");  setThreshold(0, 6);  setOption("BlackBackground", false);  run("Convert to Mask");  run("Add Slice");  run("Object Counter3D", "threshold=128 slice=1 min=2 max=524288 particles dot=3 font=12");  run("Previous Slice [<]");  run("Previous Slice [<]");  run("Next Slice [>]");  run("Delete Slice");  rename("Particles\_for\_ws");  run("Marker-controlled Watershed", "input=for\_ws marker=Particles\_for\_ws mask=None binary calculate use");  rename("ws\_segmented");  selectWindow("Results");  IJ.renameResults("Points\_ws");  selectWindow("Points\_ws");  IJ.renameResults("Results");  total\_number\_of\_zones=nResults;  print("total\_number\_of\_zones "+total\_number\_of\_zones)  is\_cell\_array=newArray(2\*nResults);  test\_cell\_array=newArray(nResults);  for(i=0;i<total\_number\_of\_zones;i++){  is\_cell\_array[2\*i]=i+1;  if(getResult("Volume",i)<100){  is\_cell\_array[2\*i+1]=1;  test\_cell\_array[i]=1;  }  else{  is\_cell\_array[2\*i+1]=0;  test\_cell\_array[i]=0; }  }  selectWindow("Results");  IJ.renameResults("Points\_ws");  selectWindow("ws\_segmented");  for(i=0;i<total\_number\_of\_zones;i++){  setResult("Zone Number",i,is\_cell\_array[2\*i]);  setResult("Is Cell",i,is\_cell\_array[2\*i+1]);  updateResults();  }  IJ.renameResults("Is\_Cell");  cell\_zone=newArray(total\_number\_of\_zones);  for(i=0;i<total\_number\_of\_zones;i++){  cell\_zone[i]=0;  }  selectWindow("Particles");  IJ.renameResults("Results");  window=40;  for(i=0;i<total\_points; i++){  selectWindow("ws\_segmented");  volume= getResult("Volume", i)\*0.2\*0.2\*0.5;  setResult("Volume\_um3",i,volume);  radius=pow((getResult("Volume\_um3", i)\*3)/(4\*PI),(1/3));  setResult("Radius",i,radius);  v=getPixel(round(getResult("Centre X", i)),round(getResult("Centre Y", i)));  setResult("Cell number", i, v);  if(v!=0){  if(test\_cell\_array[v-1]!=0){  v\_actualised=1+cell\_zone[v-1];  cell\_zone[v-1]=v\_actualised;  }  else{  x=round(getResult("Centre X", i));  y=round(getResult("Centre Y", i));  minDist=1000;  goodLabel=0;  watershedLabel=0;  for(u=-window;u<=window;u++){  for(z=-window;z<=window;z++){  if((x+u>=0)&&(x+u<getWidth())&&(y+z>=0)&&(y+z<getHeight())){  watershedLabel=getPixel(x+u,y+z);  if((watershedLabel!=v)&&(watershedLabel!=0)){  dist=sqrt(u\*u+z\*z);  if(dist<minDist){  minDist=dist;  goodLabel=watershedLabel;  }  }  }  }  }  if(goodLabel!=0){  v\_actualised=1+cell\_zone[goodLabel-1];  cell\_zone[goodLabel-1]=v\_actualised;  }  }  }  }  selectWindow("Results");  IJ.renameResults("Particles");  selectWindow("ws\_segmented");  setResult("Total Number",0,0);  for(i=0;i<lengthOf(cell\_zone);i++){  setResult("Total Number",i, cell\_zone[i]);  }  IJ.renameResults("Results on cells");  selectWindow("Results on cells");  IJ.renameResults("Results");  selectWindow("ws\_segmented");  for(i=0;i<getHeight();i++){  for(j=0;j<getWidth();j++){  v=0;  v\_actualised=0;  v=getPixel(i,j);  if(v!=0){  v\_actualised=1+getResult("Total Area",v-1);  setResult("Total Area", v-1, v\_actualised);  }  }  }  for(i=0;i<lengthOf(test\_cell\_array);i++){  setResult("Is\_Cell",i,test\_cell\_array[i]);  }  selectWindow("Results");  IJ.renameResults("Results on cells");  waitForUser("");  selectWindow("Particles");  IJ.renameResults("Results");  name\_first\_results="list\_lipids\_"+imageTitle;  dotIndex = indexOf(name\_first\_results, ".");  name\_first\_results=substring(name\_first\_results, 0, dotIndex);  name\_first\_results = name\_first\_results + ".csv";  saveAs("Measurements", dir+name\_first\_results);  run("Close");  selectWindow("Results on cells");  IJ.renameResults("Results");  name="number\_of\_lipids\_by\_cell\_"+imageTitle;  dotIndex = indexOf(name, ".");  name=substring(name, 0, dotIndex);  name = name + ".csv";  saveAs("Measurements", dir+name);  run("Close");  selectWindow(imageTitle+" (red)");  run("Close");  selectWindow(imageTitle+" (red)-1");  run("Close");  selectWindow("for\_ws");  run("Close");  selectWindow("for\_ws-1");  run("Close");  selectWindow("Particles\_for\_ws");  run("Close");  selectWindow("Points\_ws");  run("Close");  selectWindow("Log");  run("Close");  selectWindow("Threshold");  run("Close");  selectWindow(imageTitle+" (blue)");  run("Close");  selectWindow("Is\_Cell");  run("Close"); |