**//ESM II Script macro “Measure masks”**

// =================================================

// ======Macro to automatically measure the size and shape =======

// =============of multiple images showing masks============

//Developed by W.A. Out & J.F. Pertusa Grau as part of the EU Marie Curie Intra-European Fellowship "Phytores" (273610).

//Please refer to the original paper when using this macro.

//This macro was run using FIJI version 1.48c on a Mac Os X version 10.6.8.

//Based on two plugins:

// Particles\_8 by Gabriel Landini, G.Landini@bham.ac.uk

// Copyright (c) G. Landini

/\*\*

\* Bob Dougherty 6/27/2004

\* Adapt Wayne Rasband's Custom Particle Analyzer to the case of Measure Roi

\* Version 1 2/26/2007 Update with using the new version of the Custom Particle Analyzer by

\* Greg Joss and Wayne Rasband

\*/

/\* License:

Copyright (c) 2004, 2005, OptiNav, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions

are met:

Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

Neither the name of OptiNav, Inc. nor the names of its contributors

may be used to endorse or promote products derived from this software

without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*/

// Loop to run the macro repetitively, see also the last line.

var shouldKeepRunning = true;

while (shouldKeepRunning)

{

//To open a series of images that are located in a single folder and contain "mask" in the name.

//The images are converted into a stack.

run("Image Sequence...", "dir starting=1 increment=1 scale=100 file=Mask or=[] sort");

//To convert the stack into a montage (single slice) showing all the masks.

run("Make Montage...", "scale=0.25 first=1 last=150 increment=1 border=0 font=12");

//To binarise the new image to prepare for collecting measurements (a thresholded 8-bit image is required).

//To obtain measurements of all masks instead of the montage itself.

setThreshold(129, 255);

setOption("BlackBackground", false);

run("Convert to Mask");

//To run the plugin Measure Roi Curve to measure the masks.

//A new window with masks appears to visualise the action.

run("Measure Roi Curve", "size=0-Infinity circularity=0.00-1.00 show=Masks display");

//To take the measurements of two obtained variables from the plugin Measure Roi Curve, Curve\_Length Curve\_Width, and remember the values in two arrays.

n=nResults;

x=newArray(nResults);

y=newArray(nResults);

for (i=0; i<=n-1; i++) {

x[i]=getResult("Curve\_Length", i);

y[i]=getResult("Curve\_Width", i);

}

//To close the window with results.

run("Close");

selectWindow("Results");

run("Close");

//To binarise the image (montage) to prepare for collecting measurements (a thresholded 8-bit image is required).

//To obtain measurements of all masks instead of the montage itself.

setThreshold(0, 129);

setOption("BlackBackground", false);

run("Convert to Mask");

//To run plugin Particles\_8 to measure the masks.

//Be aware of the risk that digits behind the comma change are rounded up when running the macro repetitively.

//This can be solved by adding the line “rt.setPrecision (3);” below the line “public void run(ImageProcessor ip) {” in the script of Particles\_8.

run("Particles8 ", "label morphology show=Particles minimum=0 maximum=9999999 display redirect=None");

//To combine the results of the plugin Measure Roi Curve with the results of the plugin Particles8.

setOption("ShowRowNumbers", false);

updateResults;

for (i=0; i<=n-1; i++) {

setResult("Curve\_Length", i, x[i]);

setResult("Curve\_Width", i, y[i]);

}

setOption("ShowRowNumbers", false);

updateResults;

//To save the results of the plugins Particles\_8 and Measure Roi Curve in a newly created folder "Data" in the directory that also contains the folder "Masks".

dir=File.directory;

mydir = dir+"Data"+File.separator;

File.makeDirectory(mydir);

title = "P8+mroi" + ".xls";

saveAs("Measurements", mydir+title);

//To close window with results

run("Close");

//To close all windows.

while (nImages>0)

{

selectImage(nImages);

close();

}

//To inform the user and to offer the chance to restart or end the process.

shouldKeepRunning = getBoolean (

"The measurements have been written to a data folder at " + mydir + "\n\n" +

"Would you like to restart?");

} //Loop back while the user wants to continue

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END macro to automatically measure masks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*