**//ESM I Script macro “Draw a mask”**

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

// ======Macro to obtain a mask of an object======

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

/\*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. \*/

//This macro aims to separate an object from its background by obtaining the object's mask.

//The various commands of the macro create new images, thus assuring that the original image remains unchanged.

//By using the macro "Measure Masks" it is possible to subsequently measure the masks of various objects automatically.

//Loop to run the macro repetitively, see also the end of the macro.

var shouldKeepRunning=true, t=true, loop=0;

while (shouldKeepRunning==true)

{

 //To open a random image in a random file, i.e. a photograph showing an object of interest.

 run("Open...");

 ID=getImageID();

 //Here the scale of the photographs can be set, assuming that they all have the same scale.

 //run("Set Scale...", "distance=256 known=20 pixel=1 unit=µm global");

 //Loop to calibrate the photograph and to do so only once when analysing the first photograph during a session of the macro,

 //see also the end of the macro.

 if (loop==0) {

 //To set the line tool to draw a straight line along the scale bar.

 setTool("line");

 waitForUser ("To calibrate: draw a straight line along your scale bar first and press ok afterwards.\n\nEnter the length of the scale bar and the relevant unit of length in the next window.\nThe obtained scale will be applied automatically to other photographs analysed during this session.");

 //To show the window where the calibration data can be entered.

 run("Set Scale...");

 }

 //If the picture consists of a stack of images: to merge this stack into a RGB image with only a single slice.

 //Afterwards, the image is transformed into an 8-bit greyscale image, which is required for the later function "Analyze Particles".

 t=getBoolean ("Is the image an image stack?");

 if (t==true) {

 run("Stack to RGB");

 run("8-bit");

 selectImage(ID);

 close();

 }

 //To continue the process if the picture is not a stack but consists of a single slice only.

 //Also in this case, the image is transformed into an 8-bit image.

 else

 run("8-bit");

 //To zoom in on an object from which the mask will be obtained.

 setTool("zoom");

 waitForUser ("Zoom in: click twice on the object that is to be drawn and press ok afterwards. \nWhen not using a constant zoom factor, the measurements will not be comparable.");

 //To obtain the mask of the object by drawing a closed line

 //that follows the outer boundary of the object,

 // using the freehand selections tool.

 setTool("freehand");

 waitForUser ("Make a drawing of your object of interest, following its outer boundaries. \nPress ok afterwards. \nYou can refine the drawing with Alt and Shift. \nYou can start again by clicking on the photograph.");

 //To collect information about the image, related to the position of the rectangle later.

 List.setMeasurements;

 xf = List.getValue("FeretX")\*(256/80);

 yf = List.getValue("FeretY")\*(256/80);

 //To fill the selection, thus obtaining a white mask of the object.

 run("Fill", "slice");

 //To set the colors

 run("Colors...", "foreground=black background=white selection=blue");

 //To make the image binary, required for the function "Analyze Particles", thus obtaining a white mask on a black background.

 run("Auto Threshold", "method=Default");

 //To show the masks of the object in black on a white background.

 //1 in Size has been chosen randomly as a minimum value to allow the command to run.

 run("Analyze Particles...", "size=1-Infinity circularity=0.00-1.00 show=Masks");

 //To select a small part of the image containing the mask of the object

 //and eliminate the remaining part of the figure.

 //In the macro "Measure masks", this image fragment will later be combined with similar fragments in a single window.

 //To activate the rectangle tool.

 setTool("rectangle");

 //To draw a rectangle with a standard size, ideally around the mask of the object.

 //The size of the rectangle can be adapted according to the user's needs by changing the numbers in the next line.

 makeRectangle(700, 700, 600, 600);

 getSelectionBounds(x, y, w, h);

 makeRectangle(xf, yf, w, h);

 //To drag the selection around the mask of the object.

 waitForUser ("Drag the rectangle around your drawing and press ok. \n\nAvoid changing the size of the rectangle to ensure that the drawing will be measured in a later stage.");

 //To crop: to cut away the major part of the figure that does not contain the mask.

 run("Crop");

 //To save under the name "Mask of ImageID" in a folder "Masks" within the source folder.

 title=getTitle(); //Assign the Window name to a variable

 dir=File.directory;

 mydir = dir+"Masks"+File.separator; // Create a "results" directory inside the later

 File.makeDirectory(mydir);

 saveAs("Tiff", mydir+title);

 //To close all active windows

 close();

 close();

 //To apply the scale that was defined by the user when opening the macro to all photographs

 //that will be analysed during this session of the macro.

 loop=loop+1;

 //To restart or end the process of obtaining masks.

 shouldKeepRunning = getBoolean ("Well done! Would you like to restart?");

} //Loop back while the user wants to continue

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

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END macro to obtain a mask of an object\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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