**2.4. Image quantification of powdery mildew disease - MACRO**

### Macro: "Crop Ruler and Label - New Version"

1. Rotates Image (180 Degrees): the image is rotated 90 degrees to the right twice to achieve a full 180-degree rotation.
2. Crops Out the Ruler and Edges: the macro sets the selection tool to a rectangle and defines a cropping area (makeRectangle(78, 102, 2064, 3484)) to remove any ruler or edges that may interfere with the analysis. The cropped image is then rotated back 180 degrees (90 degrees twice) to restore its original orientation.
3. Identifies and Labels the Reference Area: the macro uses the magic wand tool in legacy mode (tolerance = 20) to select a label at coordinates (630, 390).A rectangle is fitted around the selection and enlarged by 50 pixels to ensure the label is fully included.
4. Sets the Label to Blue: the background color is set to blue (RGB: 0, 150, 255). The selected label area is cleared and recolored to blue to ensure consistency in future processing.
5. Deselects All Selections: the macro ensures no selection is active before proceeding with further analysis.

*macro "crop ruler and label - new version" {*

*//rotate 180 degrees*

*run("Rotate 90 Degrees Right");*

*run("Rotate 90 Degrees Right");*

*//crop out ruler and edges*

*setTool("rectangle");*

*makeRectangle(78, 102, 2064, 3484);*

*run("Crop");*

*//rotate 180 degrees*

*run("Rotate 90 Degrees Right");*

*run("Rotate 90 Degrees Right");*

*////set label to blue*

*//use the magic wand tool in legacy mode with a high tolerance (20) to*

*//select the coordinates (630, 390), to select the label*

*doWand(630, 390, 20.0, "Legacy");*

*//fit a rectangle to the selection and enlarge it, to ensure the edges of*

*//the label are included in the selection*

*run("Fit Rectangle");*

*run("Enlarge...", "enlarge=50 pixel");*

*//set the default colour to blue and cut the selection to make it blue*

*setBackgroundColor(0, 150, 255);*

*run("Clear", "slice");*

*//deselect*

*run("Select None");*

### Macro: "Area - Blue Background"

6. Duplicates the Image: a copy of the image is created to perform color-based segmentation without altering the original.

7. Splits the Image into Hue, Saturation, and Brightness (HSB) Components: the RGB image is converted into an HSB stack and separated into individual images for Hue (0), Saturation (1), and Brightness (2).

8. Thresholding for Powdery Mildew Detection: custom threshold values are set: Hue: 135-141 (to isolate relevant colors); Saturation: 0-197 and; Brightness: 0-255. These thresholds are applied, and regions outside them are masked.

9. Combining Thresholded Layers: logical AND operations are used to merge the thresholded hue, saturation, and brightness images, refining the selection.

10. Final Image Processing for Color Thresholding: the resulting image is converted into a binary mask, where detected areas are white and everything else is black. A selection is created from the thresholded image and transferred back to the original image.

11. Measurement and Analysis: the background is set to black, and everything outside the selected area is cleared. The scale is set to pixels, ensuring that measurements are conducted correctly. The area of powdery mildew coverage is measured and added to the results table. All selections are cleared to finalize this step.

*macro "Area - Blue Background" {*

*//duplicate image*

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

*// Color Thresholder 2.1.0/1.53c*

*// On duplicated iamge*

*min=newArray(3);*

*max=newArray(3);*

*filter=newArray(3);*

*a=getTitle();*

*run("HSB Stack");*

*run("Convert Stack to Images");*

*selectWindow("Hue");*

*rename("0");*

*selectWindow("Saturation");*

*rename("1");*

*selectWindow("Brightness");*

*rename("2");*

*//hue values*

*min[0]=135;*

*max[0]=141;*

*filter[0]="stop";*

*//sat values*

*min[1]=0;*

*max[1]=197;*

*filter[1]="pass";*

*//brightness values*

*min[2]=0;*

*max[2]=255;*

*filter[2]="pass";*

*for (i=0;i<3;i++){*

*selectWindow(""+i);*

*setThreshold(min[i], max[i]);*

*run("Convert to Mask");*

*if (filter[i]=="stop") run("Invert");*

*}*

*imageCalculator("AND create", "0","1");*

*imageCalculator("AND create", "Result of 0","2");*

*for (i=0;i<3;i++){*

*selectWindow(""+i);*

*close();*

*}*

*selectWindow("Result of 0");*

*close();*

*selectWindow("Result of Result of 0");*

*rename(a);*

*// Colour Thresholding-------------*

*//make binary*

*setOption("BlackBackground", false);*

*run("Convert to Mask");*

*//create selection on colour thresholded image then close it*

*run("Create Selection");*

*close();*

*//restore selection on original image*

*run("Restore Selection");*

*//set background colour to black*

*setBackgroundColor(0, 0, 0);*

*//clear everything outside the selection*

*run("Clear Outside");*

*//set scale so the measurement is done in pixels*

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

*//set measurements so the area is measured*

*run("Set Measurements...", "area mean min display redirect=None decimal=3");*

*//measure and therefore add the senescence area to the results table*

*run("Measure");*

*//deselect everything*

*run("Select None");*

*}*

### Macro: "PM Area"

12. Converts Image to Grayscale: the image is converted to a 32-bit grayscale format to standardize pixel intensity values.

13. Applies Automatic Thresholding: the default thresholding method is applied to segment powdery mildew from the background.

14. Binary Mask Conversion and Inversion: the image is converted into a binary mask, where detected areas appear white and the rest is black. The mask is inverted to ensure proper detection of powdery mildew regions.

15. Measurement of Powdery Mildew Coverage: the scale is set to pixels for accurate area calculation. The macro defines measurement parameters (area, mean, min values) and records the area of powdery mildew in the results table.

*macro "PM area" {*

*//make greyscale*

*run("32-bit");*

*//threshold*

*setAutoThreshold("Default");*

*//make binary*

*run("Convert to Mask");*

*//invert*

*run("Invert");*

*//make binary*

*run("Convert to Mask");*

*//measure*

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

*run("Set Measurements...", "area mean min limit display redirect=None decimal=3");*

*run("Measure");*

### Macro: "Rotate\_180"

16. Final Image Rotation: the image is rotated 180 degrees (90 degrees right twice) to return it to its original position.

*macro "rotate\_180" {*

*//rotate 180 degrees*

*run("Rotate 90 Degrees Right");*

*run("Rotate 90 Degrees Right");*