Supporting information for " Garryanalyzer: a Morphometric Workflow and Open-Source ImageJ Plugin for Quantitative Morphological Analysis of PNW Quercus Leaves"

Appendix S7. Installation and use of Garryanalyzer.

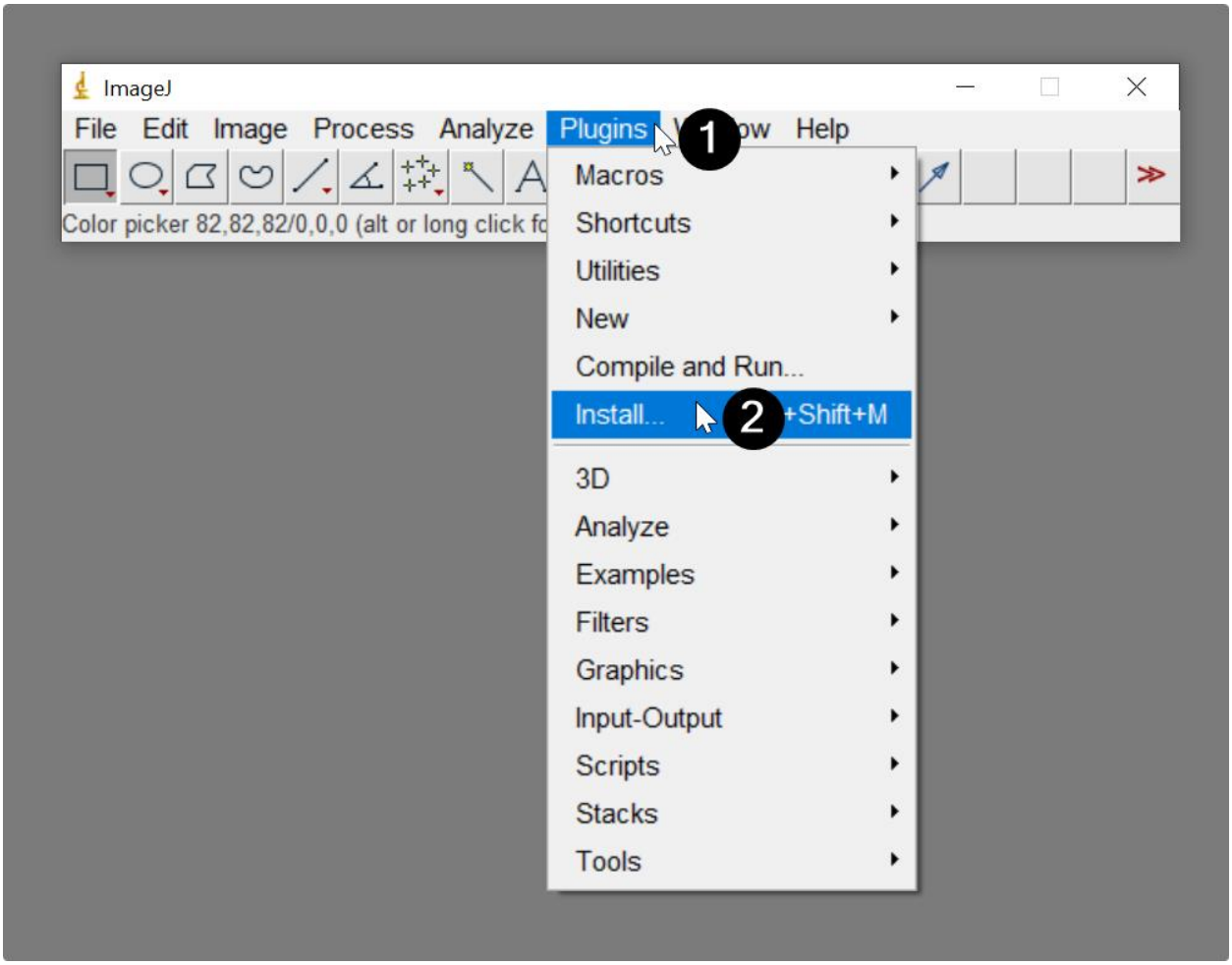
Downloads and updated version of Garryanalyzer can be found hosted on a Github page (https://github.com/zxie8561/Garryanalyzer). Garryanalyzer requires ImageJ version v1.47 or newer, which can be found from the ImageJ website (https://imagej.net/ij/download.html).

The following is a guide for installation, instructions for software usage, interpretation of results, and troubleshooting for using Garryanalyzer.

Contents

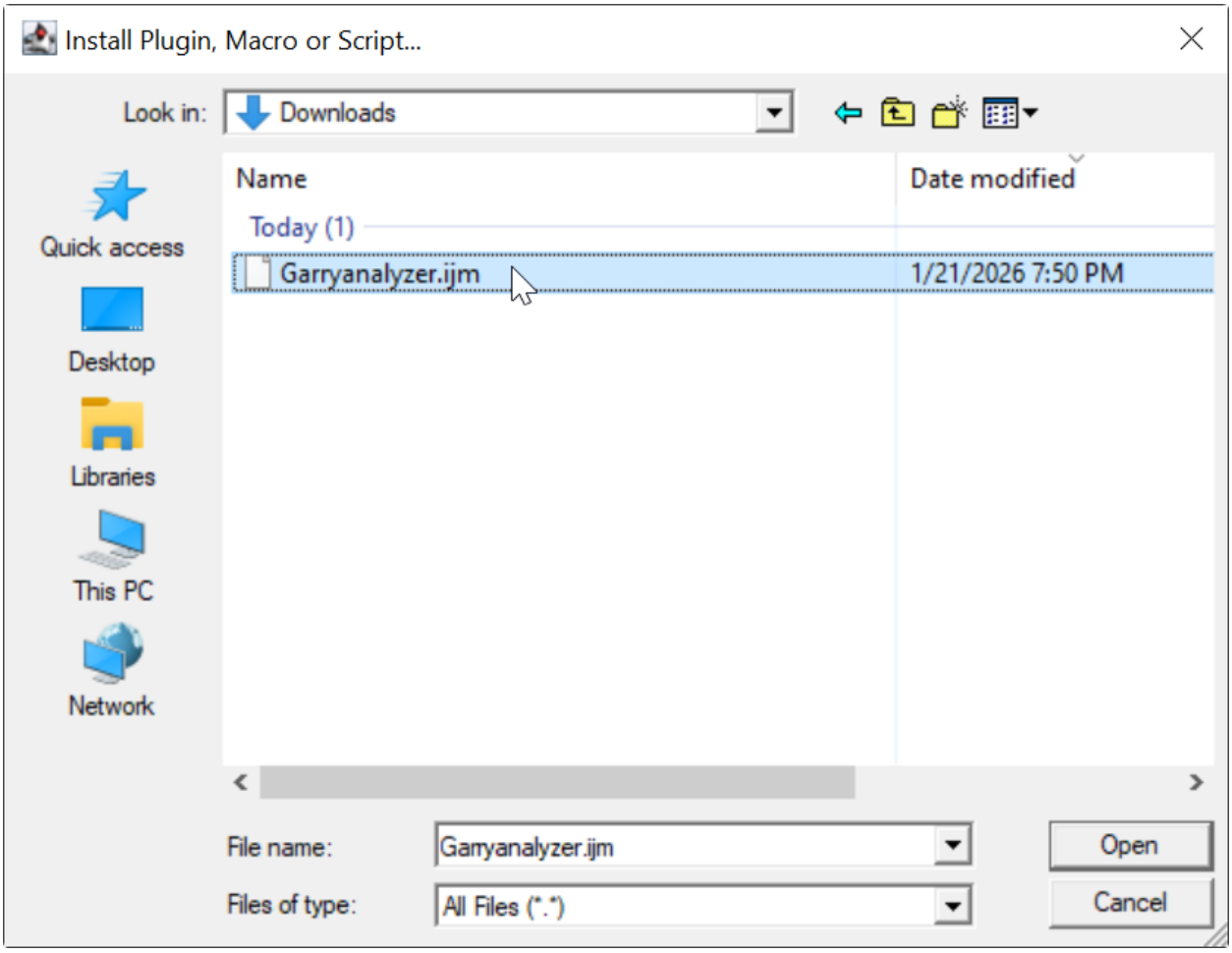
Installation Guide

Garryanalyzer can be installed for ImageJ through the Plugins menu.

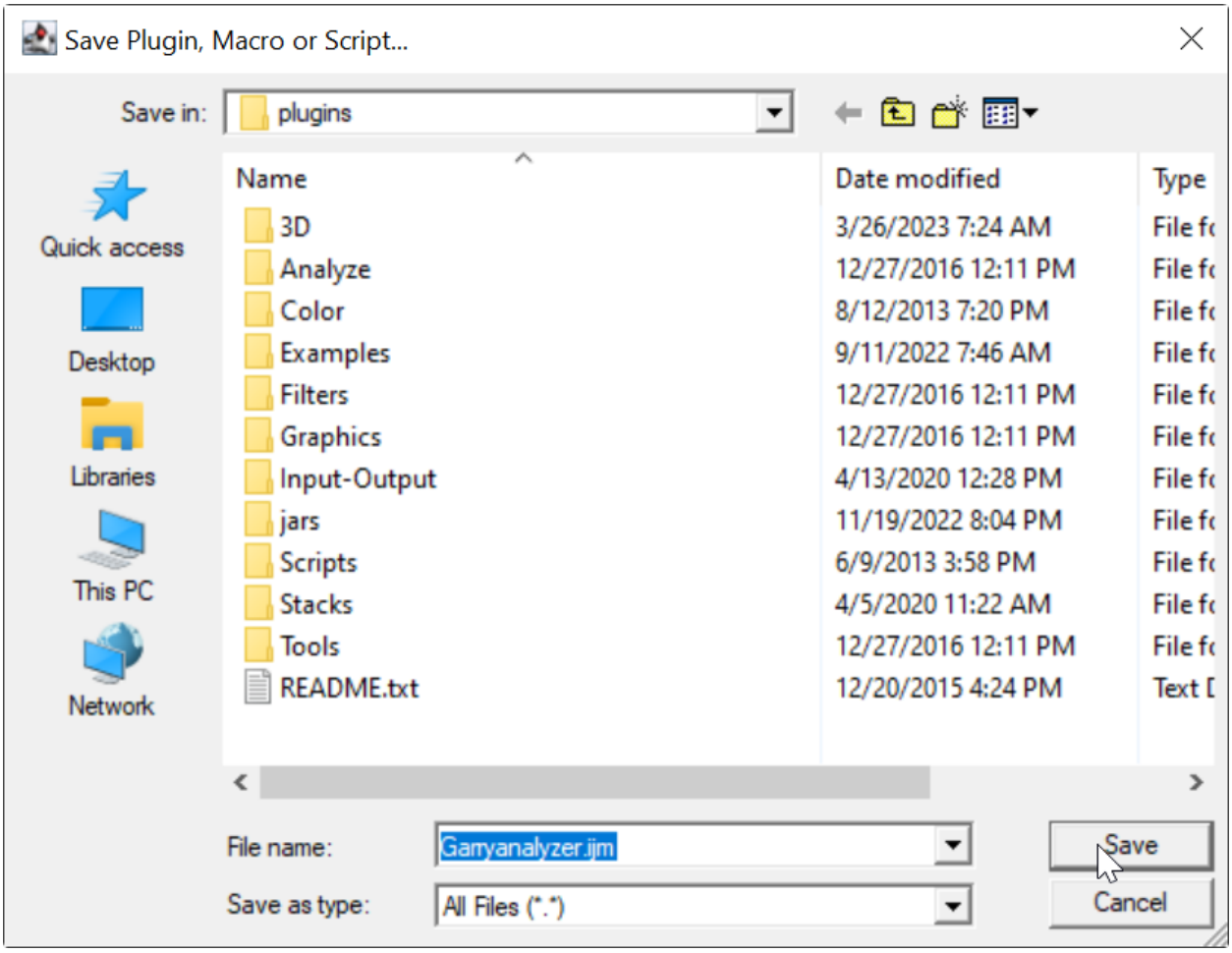


Installation for plugin files is performed under ImageJ "Plugins > Install".

Navigate to the location where Garryanalyzer was downloaded, select the file, then click Open.

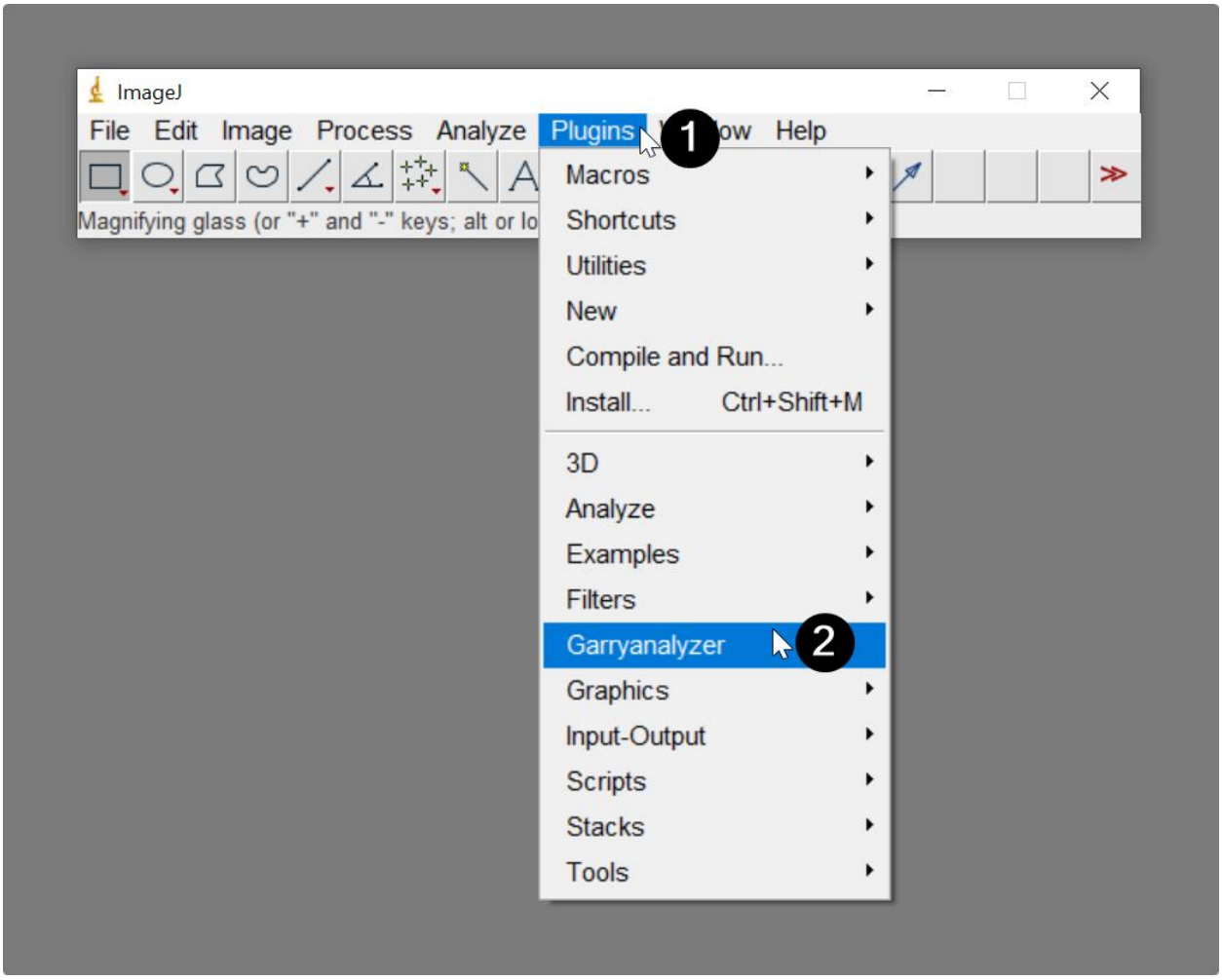


Maintain the default target folder and file name, and save the file to the plugins folder.



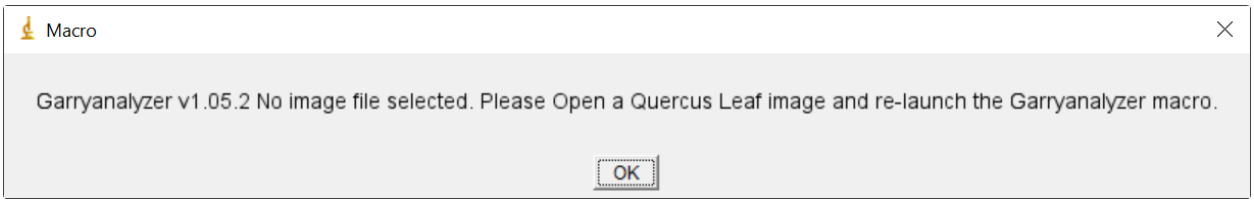
Usage of Garryanalyzer

Once installed, Garryanalyzer will appear in alphabetical order in the Plugins menu in the second section of the dropdown. The presence here verifies that Garryanalyzer is installed.



Verification that Garryanalyzer is installed, found under ImageJ "Plugins > Garryanalyzer".

At this stage, if an image file is not open and active, Garryanalyzer will give an error message.



Suitable leaf input images

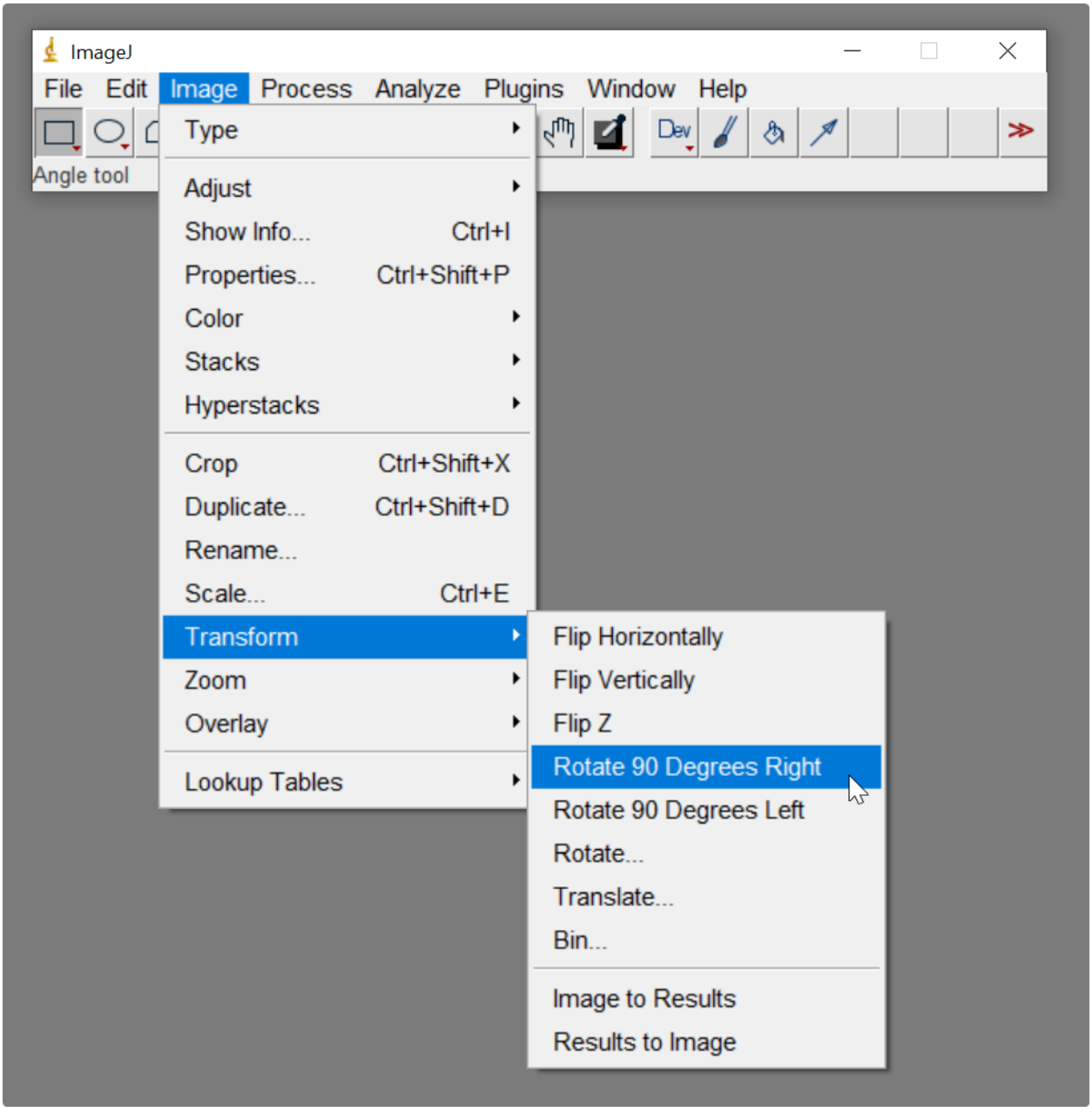
Example of a suitable leaf image, a horizontal leaf with the petiole on the left or right.

Suitable image formats include TIF, PNG, JPG, BMP.
Dimensions tested varied from approximately 1500 x 1200 to 2100 x 1400 pixels, 300 DPI.

Lower image resolution, and in particular lower object contrast, may cause issues further outlined in the **Troubleshooting** section.

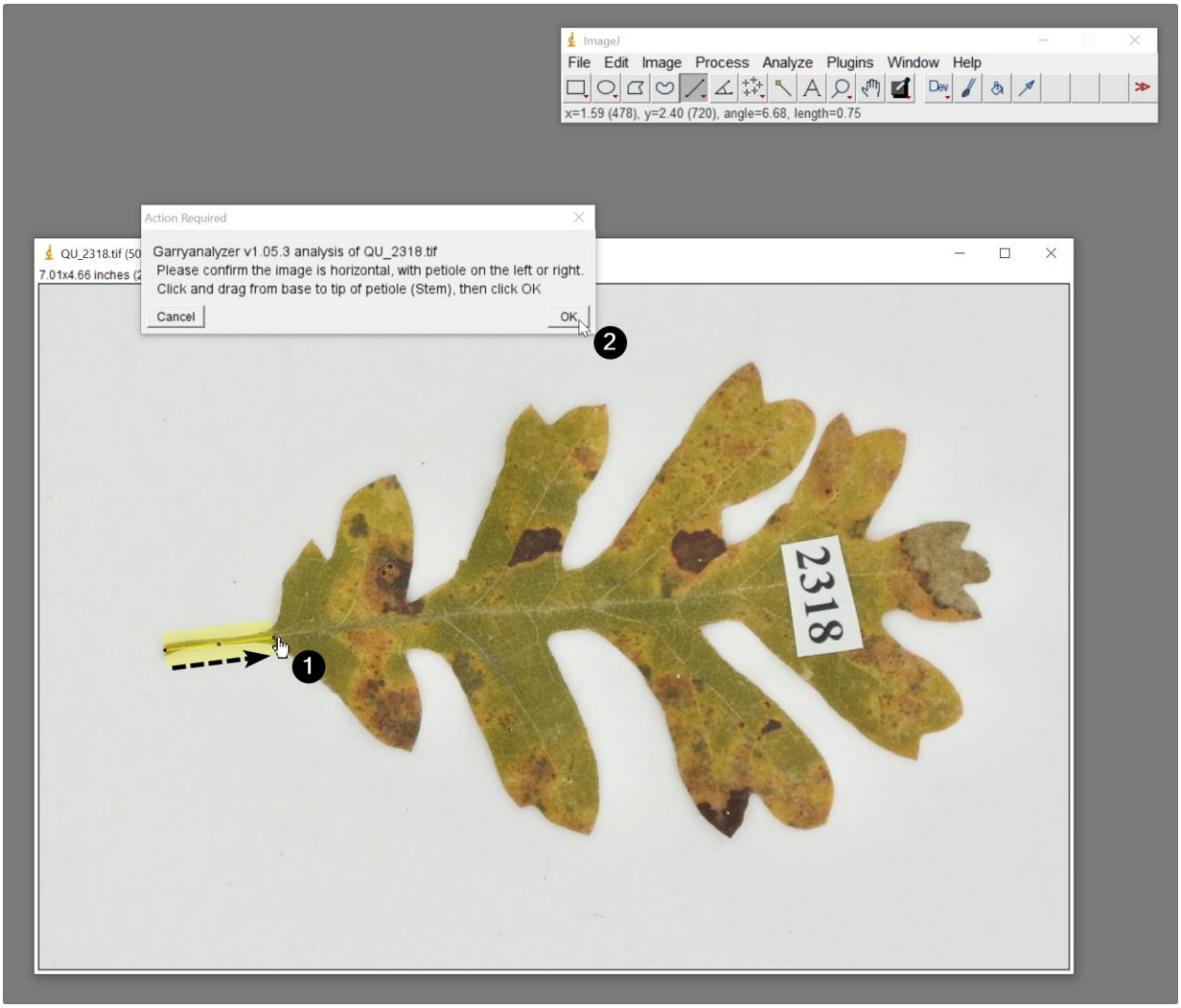
Examples of an unsuitable leaf image, a leaf with the petiole on the bottom or top orientation.

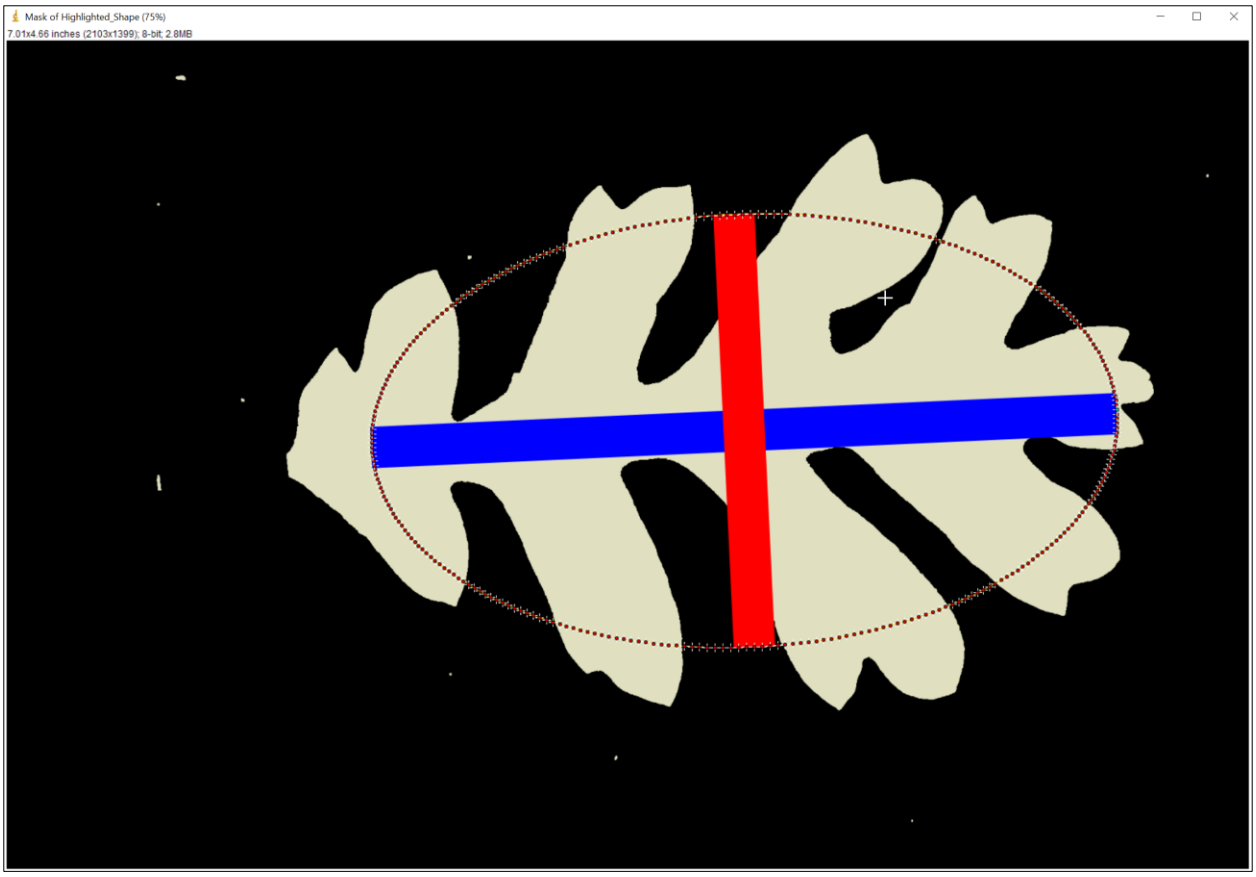
If images are in an unsuitable orientation, it is recommended to either process them beforehand to be correctly rotated, or utilize ImageJ's native "Image > Transform > Rotate 90 Degrees Right or Left" tool respectively as needed.



Another example of an unsuitable leaf image includes one with highly contrasting objects, such as a ruler or border, taking up a significant portion of the frame. Images like this may still work, but may cause undesirable issues. In order to avoid mismeasurement, we recommend cropping images to remove borders, digitally processing out artifacts, such as other leaves or debris, or retaking images if possible.

Once an image is loaded in the correct orientation, and Garryanalyzer is run through “Plugins > Garryanalyzer”, a prompt will appear to select the petiole. This is the only instance of required input, and the user is asked to click and drag from the base to tip of the petiole, then click OK.

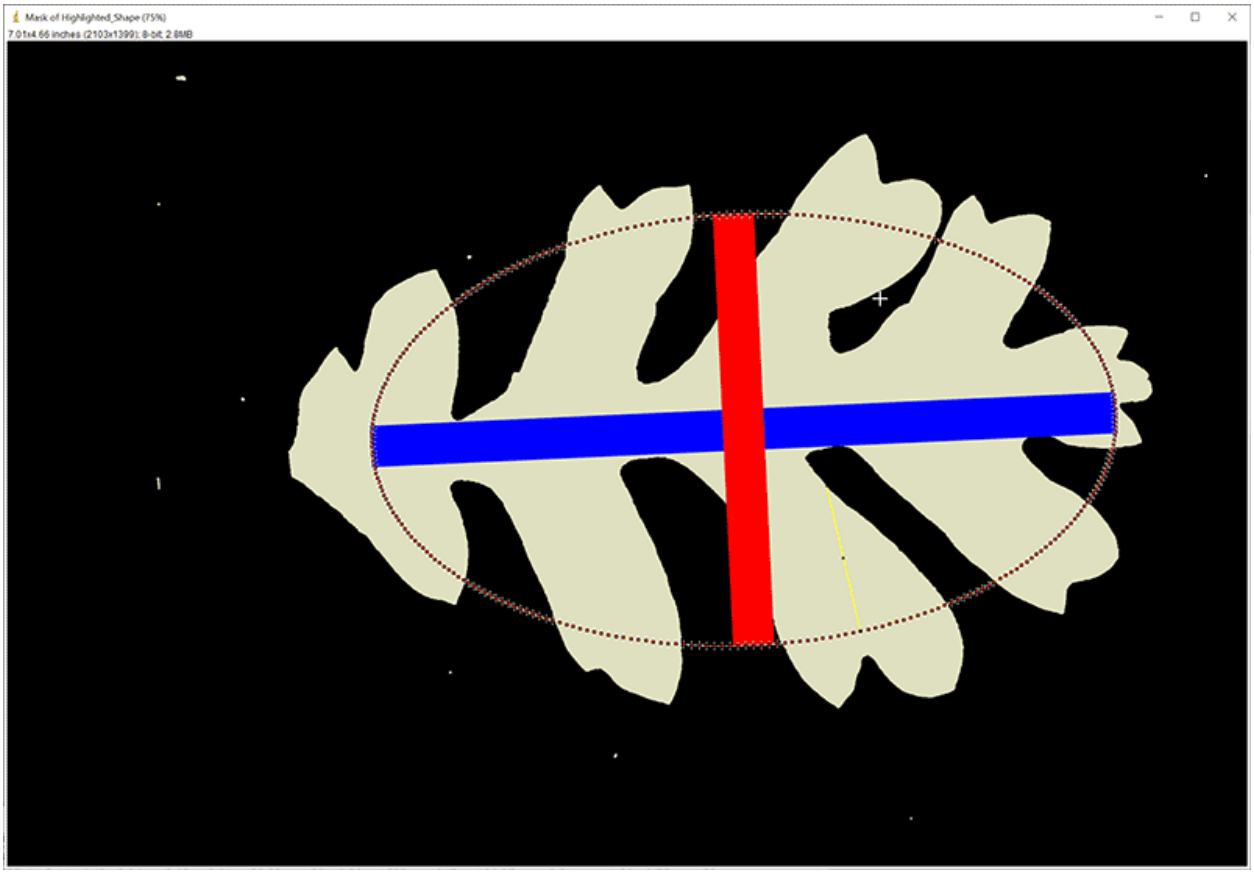




This will prompt a “Mask of Highlighted_Shape” image to temporarily be created, which is an overlay on the isolated blade section of the leaf, to come up during the following automated processing sequence. This is meant as a guide for the user to observe the progress of the tool, and can optionally be saved by uncommenting a line to do so in the Garryanalyzer script file under part I.

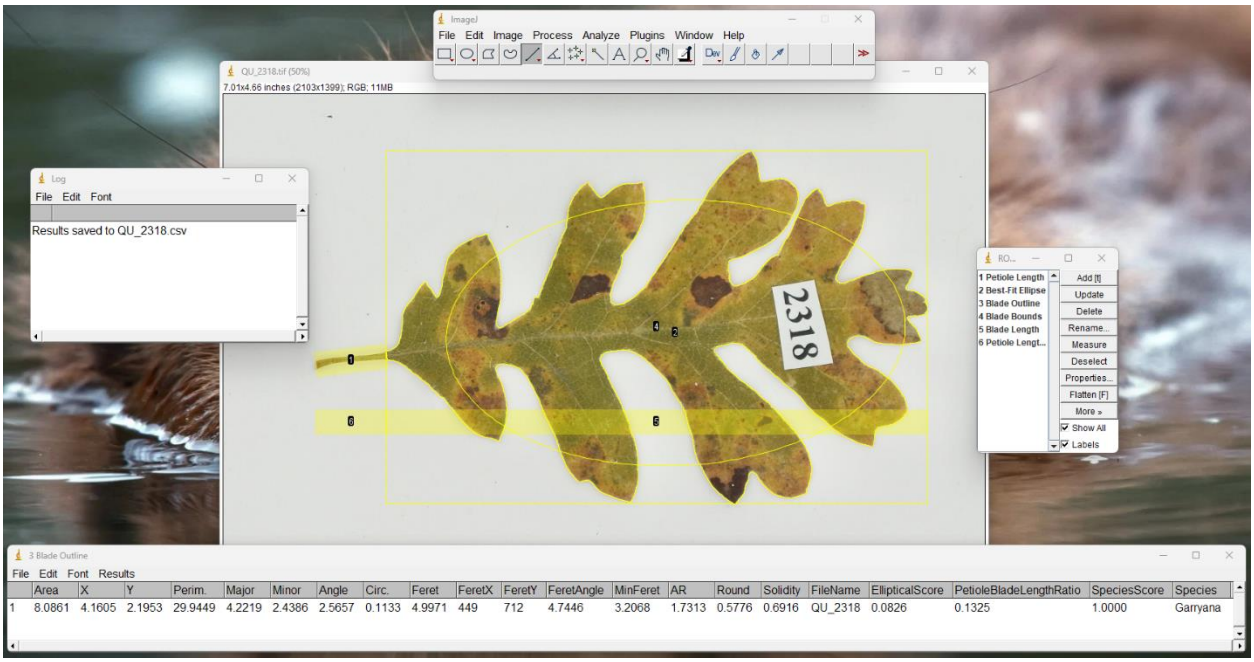
Ellipse Score Generation

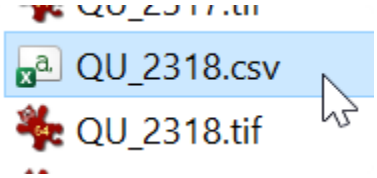
Visual representation of Ellipse Score being calculated; the best-fit ellipse and blade leaf shape are both converted to 300-point splines, and a loop for each spline of the blade shape attempts to find the closest point along the set of best-fit ellipse spline nodes, which is what is represented as the line being drawn around the leaf.



For an animated visual representation of analysis used to derive the elliptical scores for different leaf images, users may uncomment the lines in the code by searching “Visual representation of Analysis”, however, this extends processing time significantly.

After successful processing, the ROI (Regions of Interest) Manager will be populated with 6 entries, and a Results window detailing various base ImageJ variables as well as additional variables added by Garryanalyzer of the processed image (Filename, EllipticalScore, PetioleBladeLengthRatio, SpeciesScore) which are also exported to a corresponding CSV file in the same folder as the processed image.





Example: QU_2318.tif was processed through Garryanalyzer, and a resultant QU_2318.csv is created in the same directory.

Example CSV output and Interpretation of Results

Contents of QU_2318.csv:

Area	X	Y	Perim.	Major	Minor	Angle	Circ.
8.0861	4.1605	2.1953	29.9449	4.2219	2.4386	2.5657	0.1133

Feret	FeretX	FeretY	FeretAngle	MinFeret	AR	Round	Solidity
4.9971	449	712	4.7446	3.2068	1.7313	0.5776	0.6916

FileName	EllipticalScore	PetioleBladeLengthRatio	SpeciesScore	Species
QU_2318	0.0826	0.1325	1	Garryana

The species score is used to predict the species of the leaf image and is calculated from elliptical score and petiole length / blade length ratio. The species score is calculated in part O-a of the script and can be commented out when measuring taxa other than *Q. garryana* or *Q. robur*.

Included in the base ImageJ measurements are some absolute or position-based measurements, such as area or X / Y. Please note that these measurements would not be usable unless a calibration step was performed. Garryanalyzer does not have automated calibration, although it is possible with base ImageJ function Analyze > Calibrate.

A few of the base ImageJ parameters may be of interest to users studying the general shape homogeneity or irregularity in other taxa. Circularity (“Circ.”) measures the proportion between the blade area and perimeter. Solidity measures the proportion between the area of the blade and its convex area. Roundness (“Round”) measures the proportion between the blade area and the area of a circle drawn from the length of the blade. Another ratio-based particle measurement, Aspect Ratio (AR), measures the ratio between the length and width of the blade. Detailed explanation of these parameters can be found in ImageJ’s documentation (https://imagej.net/ij/docs/menus/analyze.html).

Troubleshooting

Resolution: It can be the case that images that are too low resolution or with low contrast do not register the intended shapes and morphometry for ImageJ or Garryanalyzer to interpret.
In a test for resolution accuracy, the large .TIF file was reduced to a lower resolution, to illustrate the potential effect on Garryanalyzer's variables. It's suspected that 500 x 300 pixels are around the lower end for acceptable resolution of a leaf image, where the details of a leaf, such as closely spaced lobing, may otherwise start to blur together, causing incorrect morphometric output. If leaf images are too low resolution, this can cause issues as ImageJ and Garryanalyzer require a certain level of detail to be present for measurements to proceed as intended.

FileName	EllipticalScore	PetioleBladeLengthRatio	Resolution (pixels)
QU_2318	0.0826	0.1325	2103 X 1399
QU_2318	0.0798	0.1420	500 x 333

Contrast: Lower contrast can often be a more significant issue, for instance when a clearly contrasting black or white background is not provided for the leaf to register against, which may result in an incorrect automated highlighting of the leaf blade by the software's contrast detection, an example of this would be a leaf in poor lighting that falsely causes only part of the leaf to be highlighted. In order to address mismeasurement caused by contrast issues, images may have to be manually adjusted for higher contrast between the leaf and the background, or retaken with more contrasting lighting and a clearer background.

Failure to run: If Garryanalyzer is unable to run on a system, the user is advised to update ImageJ and reinstall the latest available version of Garryanalyzer. If updates are not available or do not resolve the issue, users are invited to reach out to Garryanalyzer's developers directly through the software's host page on Github.

Failure to install: If automatic installation does not work, such as in certain versions of FIJI, an ImageJ derivative, manually adding the Garryanalyzer file to the Plugins folder followed by a restart of the program may correct this issue.