LeafI – The Leaf Interrogator

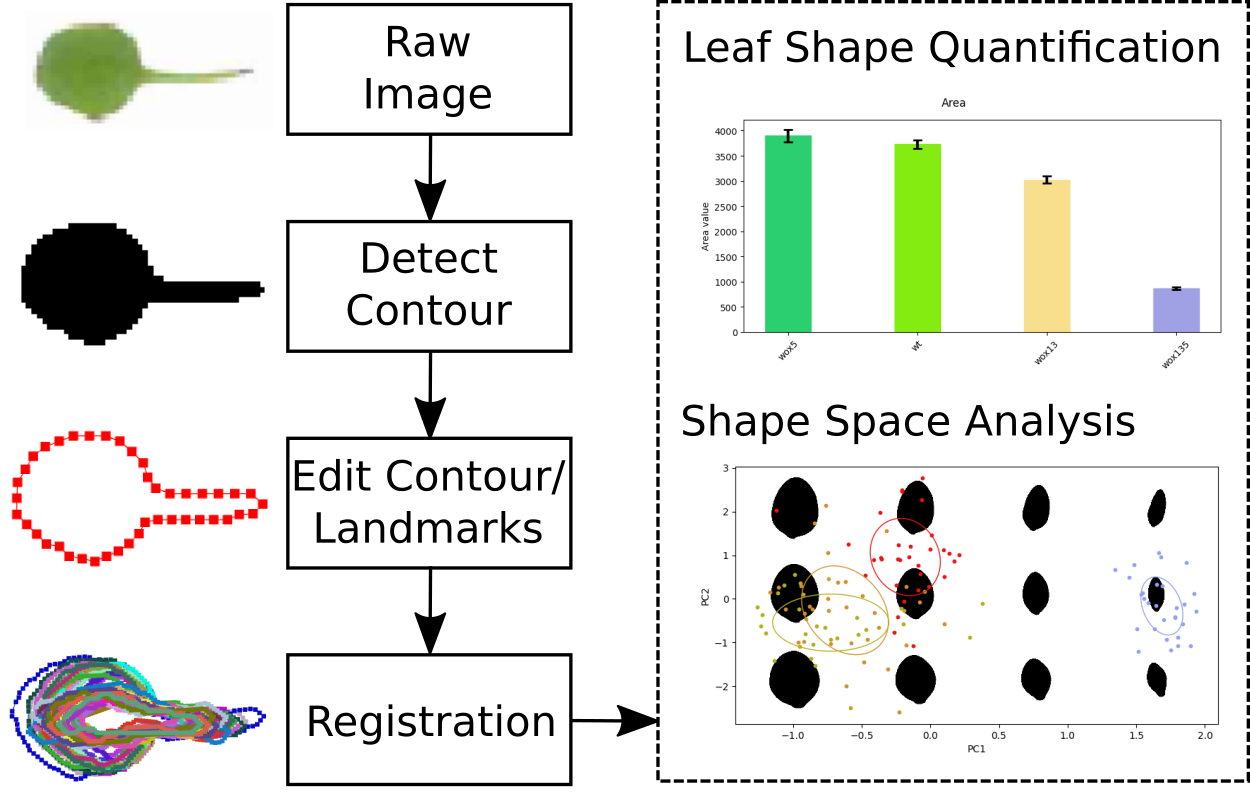
**User Guide**

Authors Leafi: Babak Hashemi, Adam Runions

Authors User Guide: Soeren Strauss, Adam Runions

**Overview**

The Leaf Interrogator (LeafI), is an image analysis software for performing morphometric analysis of leaf shapes (Fig1).



*Figure 1: The LeafI data analysis pipeline. Starting with raster images, leaf contours are detected and extracted. The user can then edit the contour and adjust the number of samples along the contour. The collection of all contours can be registered to align their position and orientation. Processed contours can be analyzed using simple shape measures, or shape-space analysis.*

**Getting Started**

This section contains a step by step guide on how to perform the standard workflow data analysis using LeafI.

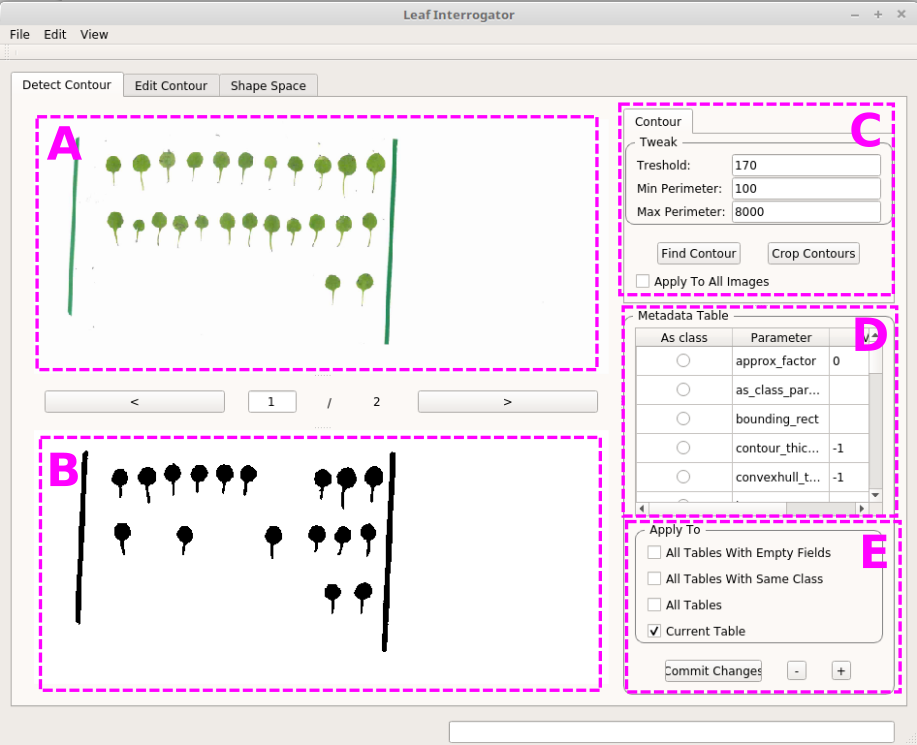
**1. Start LeafI**

* Open a console in the LeafI main folder and activate your LeafI environment ENV

source ENV/bin/activate

(see also the installation section)

* Run: python3 main.py which opens the main window.



*Figure 2: The LeafI main window with the open default tab (“Detect Contour”). (A) raw image viewer, (B) processed image viewer, (C) parameters for contour extraction, (D) metadata table, (E) options to edit the metadata.*

**2. Import & Save Data**

2.1 Open Images

When starting a project from scratch, first open image files (or folders) via the menu “File/Open File” & “File/Open Folder” or drag and drop them onto the main window.

The standard workflow is to start with images containing either single leaves or multiple scanned leaves (see Figure 2A). For multiple leaves, LeafI is able to separate them into individual images.

**Note:** All images should have the same resolution. Images larger than 2500 pixels in width or height will be resized.

**Note:** Make sure that image file names are unique.

2.2 Project Files

LeafI can save and load previously analyzed data as project files. These files are essentially gzip archives of the source images and data generated during the processing and analysis of the leaf data.

When a project file is opened with LeafI, it is unzipped into the /temp folder.

Overview project folder structure

* data\_dir contains the source images used for analysis
* For each input image a subfolder is created containing the following files:
  + the thresholded image used for contour extraction
  + a CSV file with the resampled contour coordinates (X,Y)
  + a CSV file containing the metadata information for the entire image
  + When there are multiple contours in the image, a subdirectory “cropped” is present. For each contour extracted from the original image, it contains:
    - Image indicating the location of the contour in the original image
    - Image containing only the extracted contour
    - CSV files with contour coordinates (X,Y) for the extracted, resampled and the aligned contour
    - CSV meta-data table

**Note:** Resampled and aligned contours are not updated automatically. Before using these contours in a shape-space or outside LeafI make sure to resample and re-align contours.

Analysis results are saved directly into the project folder

* For each Measure process applied to the contours, a plot and CSV file with the raw quantifications is included.
* The average contour from the shape space analysis (CSV, (X,Y) coordinates)

**Note:** Analysis results are updated when you run the measure process in LeafI. Before using analysis results, always rerun the measure or shape-space analysis of interest before using the results contained in the project file. This guarantees that the analysis will be consistent with the current project.

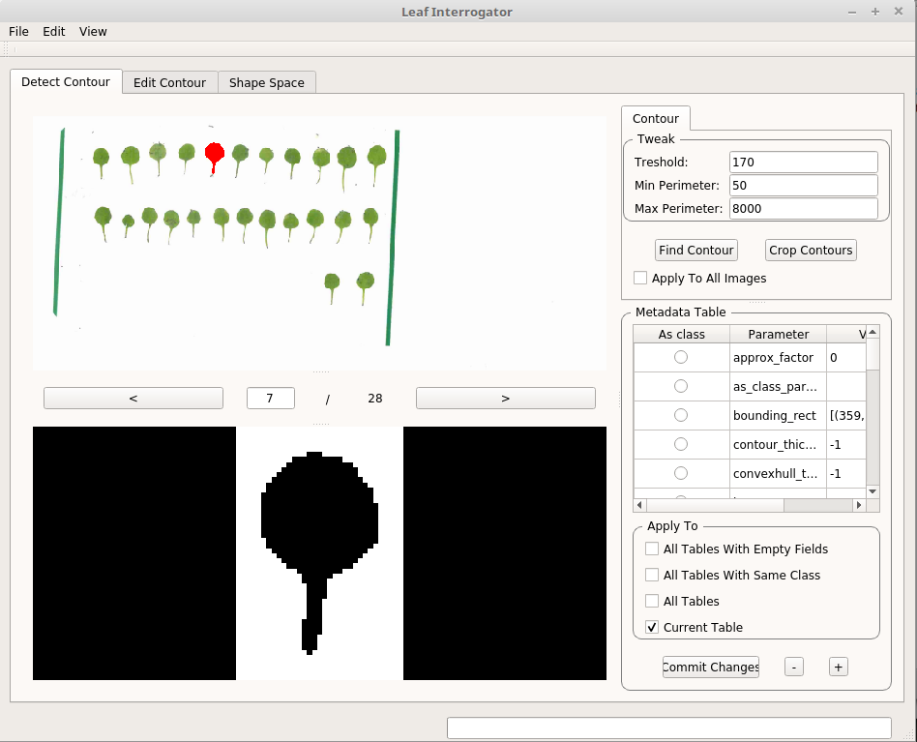
**3. Data Preprocessing and Analysis**

*3.1 The “Detect Contour” tab*

* After loading the images, rotate them if necessary. This can be done using View->Rotate Left/Right.

Next, it is required to detect the contours of individual leaves. For composite images, this is a two-step process.

* First, press the button “Find Contour” to threshold the image and create a binary image. The resulting binary image will be shown in the bottom image viewer (see Figure 2B). Parameters can be adjusted depending on the image quality and leaf sizes (Fig 2C). If the result does not show all the leaves or additional objects that aren’t leaves, the tool can be re-run with different parameters. Please note that additional objects can also be deleted following contour extraction. For images containing a single leaf, this is the only step to perform.
* For composite images, leaves must be cropped from the original image using the “Crop Contours” button. This creates individual images for each leaf, and highlights their position in the original image in red (Figure 3).
* **Note:** “Find Contour” can be applied to all loaded images at once by ticking the “Apply To All Images” checkbox. By contrast, “Crop Contours” must be applied to each composite image individually.
* **Note:** Additional non-leaf objects can be removed after cropping by viewing the object and running “Edit->Delete” from the menu bar.



*Figure 3: View after running “Find Contour” and “Crop Contours” on an image containing multiple leaves. It is possible to inspect single extracted leaves using the buttons “<” and “>”. The currently shown processed leaf is highlighted in the raw image viewer in red.*

The Metadata table

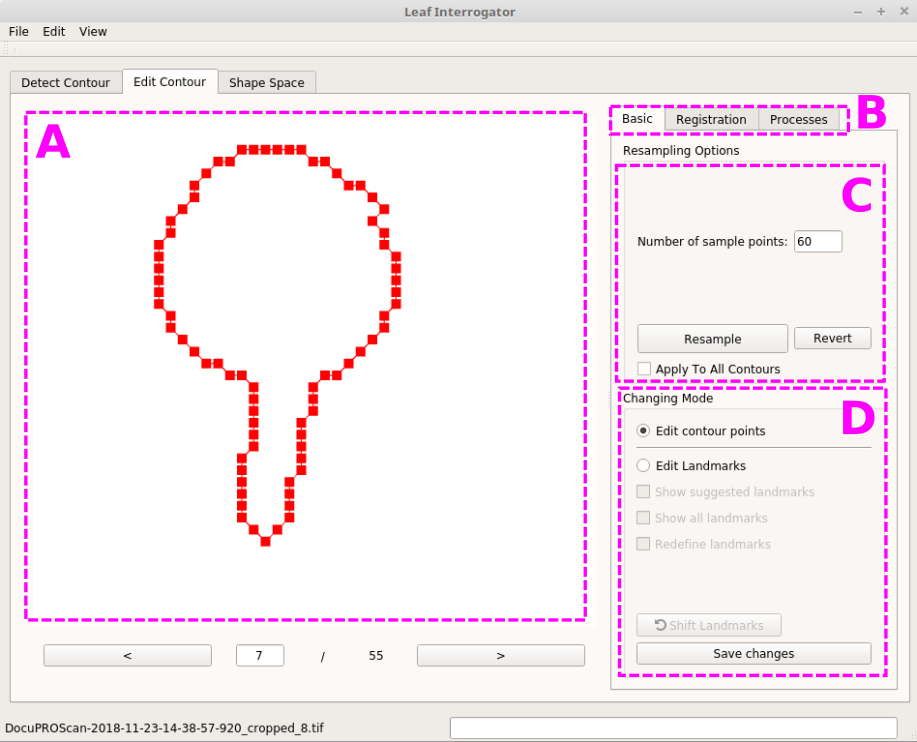
The metadata table is located on the right hand side of the main window below the Find/Crop Contour buttons (Fig 2D). This table stores additional data for every leaf, including custom data and fields introduced by the user.

When analyzing different species or genotypes it is useful to provide a “class” parameter in the metadata table for each leaf.

* To add a class parameter to all contours, select Apply To “All Tables” and press “+”. A new field will now be added at the end of the table.
* Now scroll to the end of the table to the new empty row. Add a name to the “Parameter” column (e.g. “genotype”) and the information you want to save in the “Value” column (e.g. “Wild Type”).
* In this example we want to use the new field as class parameter, so the radio button in the first column needs to be selected as well.
* Press “Commit Changes” to save the changes to the table. This step is always required after changing the metadata table of a leaf. After saving the rows are ordered alphabetically by “Parameter” name.
* Now the tables of all leaves will have the metadata field “genotype” filled with the value “Wild Type”. To assign the correct genotype of all leaves it is necessary to go through them one by one using the “>” or “<” buttons and change their value of the “genotype” field if required. Make sure to select “Current Table” before you press “Commit Changes” after changing the value of a table.
* **Note:** Both columns (“Parameter” and “Value”) need to contain an entry and will not be saved otherwise!
* **Note:**
  + Fields/rows can be added using the “+” button and deleted using the “-” button
  + Double clicking to a table cell allows you to edit them
  + Press the “Commit Changes” button to save your edits (with the appropriate setting in the “Apply To” check box).

*3.2 The “Edit Contour” tab*

After the detection step the leaf contours can be edited, visualized and analyzed.



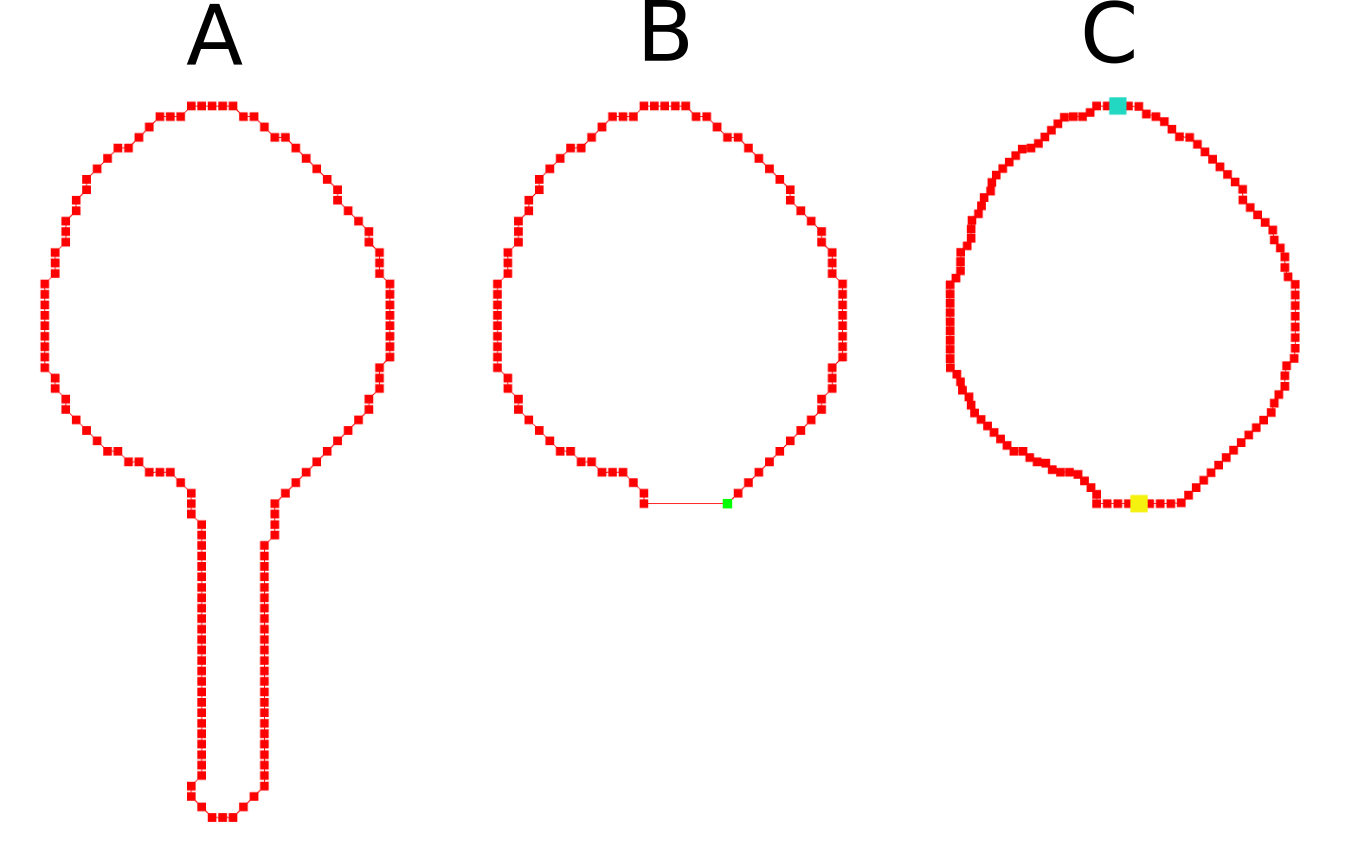
*Figure 4: The “Edit Contour” tab. (A) The main visualization window. (B) the sub-tabs. (C) Re-sampling options. (D) Changing landmarks options.*

3.2.1 The “Basic” subtab

By default the sub-tab “Basic” is active, where contours can be resampled and edited. Here also the landmarks which are used for the shape-space analysis can be edited (Fig 4).

Editing the contour

* The contour can be edited using keyboard and mouse by clicking into the main visualization area (Fig 4A). Mouse over the main visualization area to see the following tool tip with the shortcuts for editing contour points:
  + Select a point: mouse **Left-Click**
  + Delete a point from contour: 1. select a point 2. press **Backspace (Delete** on mac**)**
  + Add a point to a contour: **Hold Control (Command** on mac**) + Left-Click** on desired position
  + Remove part of the contour: **Hold Shift** and **Left-Click** on start and end position. ***Note:*** *The contour direction of removing is* ***Anti-Clockwise.***
  + To add a landmark (edit landmarks must be enabled): **Hold Control (Command** on mac**) + Left-Click** on contour. ***Note:*** *The order (position) of the landmarks are important for registration methods. They are differentiated by different colors.*
* One example of a common contour edit is removing the petiole. The steps on how to do that are shown in Figure 5. This can be achieved with two mouse clicks while holding the control key.
* To save your edits to the contour press “Save Changes”.
* **Note:** Extracting the contour again (i.e. run “Find Contour” in the “Detect Contours” tab), will replace the saved contour.

*Figure 5: Contour editing steps. (A) Initial contour after cropping. (B) Contour after manually removing the petiole. (C) Resampled contour (note the increase of contour points) with manually set landmarks (in cyan and yellow).*

Resampling

This creates equally spaced out border points along the contour and is required for the shape space analysis (see Fig 4C).

* Select” Apply to all contours” and press the “Resample” button.
* The number of sampling points can be adjusted if required. This controls the number of points between successive landmarks on the contour (Fig 5C).

Editing landmarks

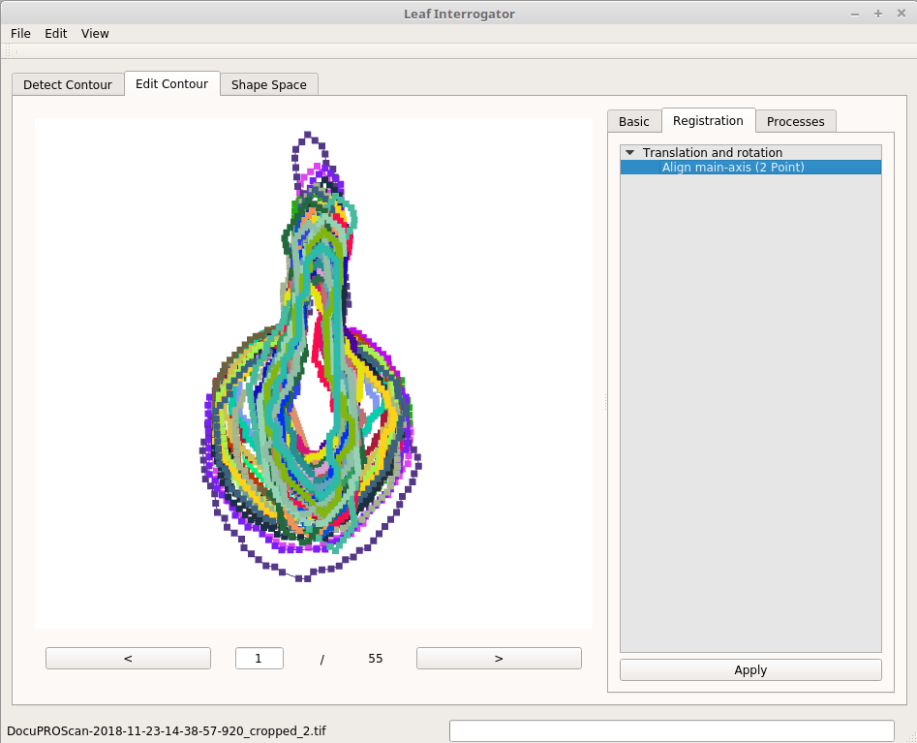
Landmarks are required to align all the leaf contours (see Fig 4D).

* Two default landmarks are set automatically, but can also be set or edited manually using the options below the “Edit landmarks” option. “Edit landmarks” must be enabled to change the position and number of landmarks.
* Default landmarks are placed approximately at the tip and base of the leaf.
* To ignore the default landmarks and set them manually check the “Redefine Landmarks” checkbox and select with Ctrl + Left Mouse the appropriate contour points

**Note:** You should visually verify the position of landmarks prior to performing shape space analysis.

**Note:** When editing the contour or landmarks make sure to press the button “Save changes” following every operation! (We observed a bug on some systems where it is required to press “Save changes” twice.)

**Note:** Any contour/landmark edits can be reverted using the “Revert” button. This will restore the contour to the last saved state.



*Figure 6: The “Edit Contour” tab with the “Basic” subtab. All registered contours are visualized.*

3.2.2 The “Registration” subtab

This sub-tab enables the leaf registration which is required to increase the accuracy for all further shape space analyses. Registration removes variance related to differences in the position and orientation of leaf contours. There is currently only one method to compute the registration: Select “Align main-axis (2 point)” in the “Translation and Rotation” folder and press “Apply”. You should now see the registered contours as in Figure 6.

3.2.3 The “Processes” subtab

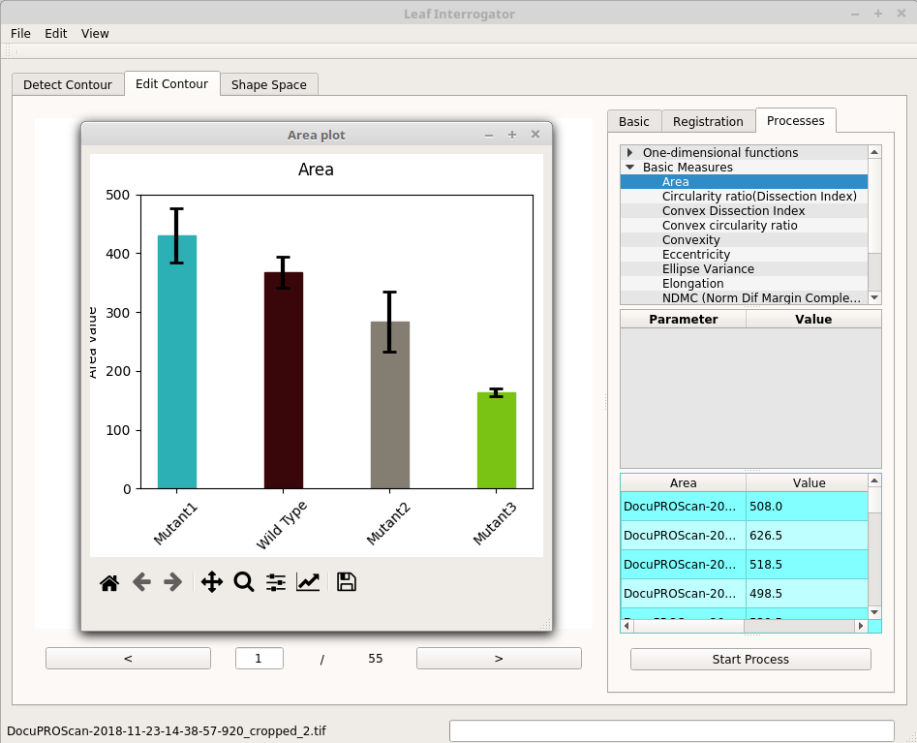
This sub-tab contains processes which quantify leaf contour based properties.

Currently implemented are the following processes:

* Area: the area enclosed by the contour, in pixels2 based on the dimensions of the original image (which is backed up in the data\_dir of the project)
* Elongation: 1 – ratio of sides of a fitted rectangle, 0 for a square, larger values with a maximum of 1 for increasingly elongated shapes
* Perimeter: the perimeter of the contour in pixels

To run a process, select it from the process tree and press the “Start Process” button. Once the process has finished, the results are listed in the table at the bottom right widget where the values are listed for each leaf. Basic processes also output a separate plot window.

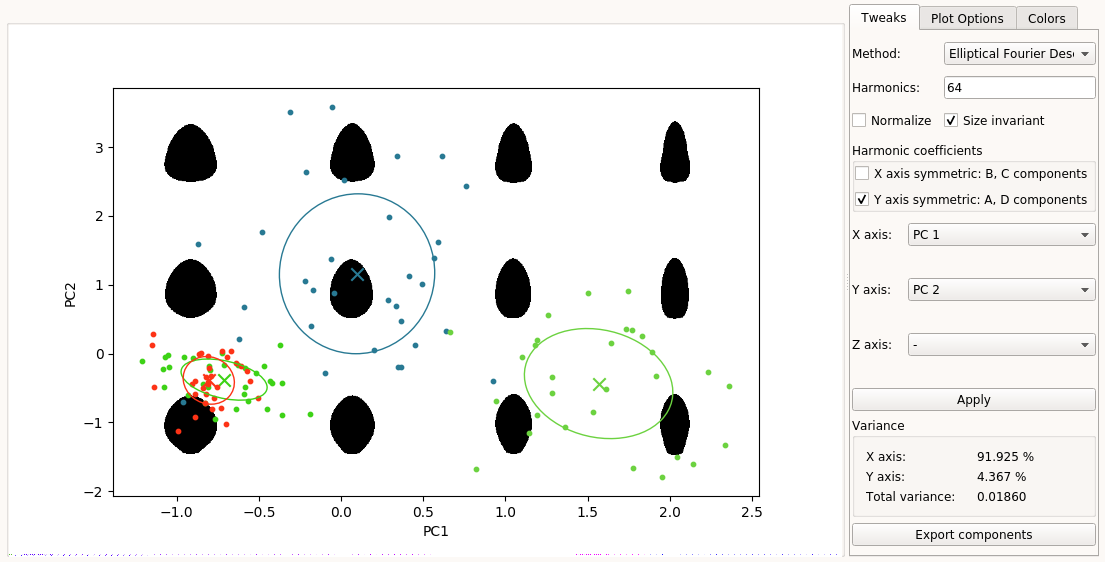
The data generated by processes is also stored within the project file/folder as a CSV file named in the format “ProcessName\_results.csv”.



*Figure 7: After running a process, a separate plot window opens with a bar graph showing the mean and standard error for each class. The computed values of the process can be found in the bottom right of the main window (table with cyan background) and are also saved within the project folder.*

*3.3 The “Shape Space” tab*

Here an advanced morphometrics analysis using Elliptical Fourier Descriptors (EFD) can be performed.

*Figure 8: The “Shape Space” tab with the “Shape space” sub-tab shown. An elliptical fourier analysis was performed, creating a plot of the selected components (X-axis: first and Y-Axis: second principal component). The percentage of variance explained by the selected components can be found in the bottom right corner of the window.*

3.3.1 The “Shape space” subtab

By default this sub-tab is open (Fig 8) which allows to set the options for the morphometrics analysis and the data plot to be generated. Upon pressing the “Apply” button the analysis is computed and a scatter plot using the selected options of the fields “X/Y axis” is created with the reconstructed images created by sampling the eigenshapes shown in the background.

Ellipitcal Fourier Analysis

Converts contours into Elliptical Fourier Descriptors (EFD), before generating the shape space.

Parameters:

* Harmonics: Controls the number of EFD coefficients used in the analysis.
* Normalize: Normalizes contours to eliminate scaling and rotation. Only needed if registration has not been applied to the contours.
* Size invariant: Rescales the EFD coefficients to make the resulting shape space scale-invariant.
* Harmonic coefficients: Allows the user to isolate symmetric components of the EFD coefficients. This can be used to isolate the bi-laterally symmetric component of leaf shape variation.

Data Export

* Plots and numerical data can be exported using the “Export components” button. This will let the user choose a filename to save the plot and CSV files containing the numerical results of the shape space analysis.The files exported include:
  + - Shape space plot (as a svg, pdf or png)
    - Mean EFD coefficients
    - Orthogonal components (principal component vectors)
    - Prepared data (EFD values for each contour)
    - Transformed data (Prepared data in the coordinate system provided by the principal components)

3.3.2 The “Plot” subtab

This tab allows the plot to be further customized, e.g. to add standard deviation and standard error ellipses, or change some basic aesthetics. Furthermore, clicking on a data point in the plot area causes the image corresponding to the data point to be shown in the top right panel of the plot options tab.

3.3.3 The “Colors” subtab

This tab allows the colors used in the plots for different class groups to be modified. Clicking on a color opens a color selection window.

**4. Installation**

Step 1, download:

* Go to the directory where you want LeafI to be installed
* Clone from repository (you need an account and access first):

git clone https://gitlab.mpcdf.mpg.de/g-adamrunions/leafinterrogator\_zhang\_et\_al.git

Step 2, create the python environment:

* First, install python virtual environment package, and then upgrade pip:

apt-get install python-virtualenv

apt-get install python3-venv

* Create a virtual environment for LeafI (local directory):

python3 -m venv leafiEnv

Step 3, enable and configure environment:

* Activate the environment, and upgrade pip:

source leafiEnv/bin/activate

pip3 install --upgrade pip

* Installed required packages using (you can find minimal\_requirements.txt in the main directory for LeafI) :

python3 -m pip install -r minimal\_requirements.txt

If downloading times-out, rerun the preceding command

* If there are missing packages, these can be installed individually using:

python3 –m pip install <package\_name>==<version-number>

Note that “==<version-number>” can be omitted for all packages except opencv-python

* The complete set of packages to install are:

freetype-py  
 imutils  
 mahotas  
 matplotlib  
 networkx  
 numpy  
 pandas  
 Pillow  
 PyQt5  
 PyOpenGL  
 pyrr  
 QtAwesome  
 scikit-learn  
 scipy  
 opencv-python==3.4.9.31

Step 4, run LeafI:

* make sure your environment is activated (source leafiEnv/bin/activate)
* Run: python3 main.py