Phenotiki – LeafAnnotationTool

*Massimo Minervini**
Pattern Recognition and Image Analysis (PRIAn)
IMT Institute for Advanced Studies Lucca, Italy
Version 1.0

March 18, 2016

Abstract

This document describes an annotation tool that can be used to semi-automatically segment leaves in images of plants. This tool is designed to exist as a stand-alone application and user interface (developed in MATLAB), relying on an interactive graph-based segmentation algorithm to propagate expert provided priors (in the form of pixels) to the rest of the image, using the random walk formulation to find a good per-leaf segmentation. An experimental validation of the tool and further details on the underlying computer vision approach can be found in [**?**]. The tool can be used to either create data for training larger machine-learning based applications or as part of a phenotyping analysis by integrating also phenotype extraction software.

Contents
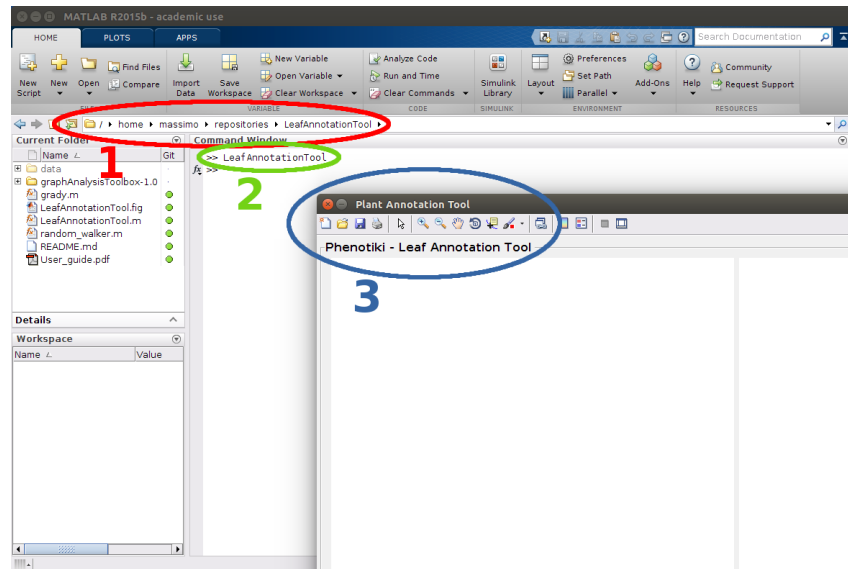
*massimo.minervini@imtlucca.it

Figure 1: Steps to launch the LeafAnnotationTool: (1) browse for the folder containing the code; (2) type `LeafAnnotationTool` and hit enter; (3) the application will open.

1 System requirements and installation

LeafAnnotationTool is currently distributed as MATLAB source code, therefore it requires a MATLAB environment to be used. The tool can run on any operating system supported by MATLAB (e. g., Linux, Windows, Mac).

First, download the source code. Two options are available:

- clone the LeafAnnotationTool repository from GitHub (`https://github.com/phenotiki/LeafAnnotationTool`);

- download the software package from `phenotiki.com` and extract the ZIP archive into a local folder.

To launch the application, please follow these steps, which are also illustrated in Figure 1. After starting Matlab:

1. browse for the folder containing the LeafAnnotationTool code;

2. in the command window (i.e. after `>>`), type `LeafAnnotationTool` and then press the enter button on the keyboard (alternatively, right click with the mouse on 'LeafAnnotationTool.m' and select 'Run'.);

3. the interface of the application should open in a new window.

! → Notice that the application should not be started from 'LeafAnnotationTool.fig'.

The segmentation approach relies on the random walks algorithm by Leo Grady in 2006 [**?**]. We used Grady's implementation in MATLAB of the Random Walker and his Graph Analysis Toolbox, which are included in the download package of LeafAnnotationTool and are also freely available on Grady's website: `http://cns.bu.edu/~lgrady/software.html`.
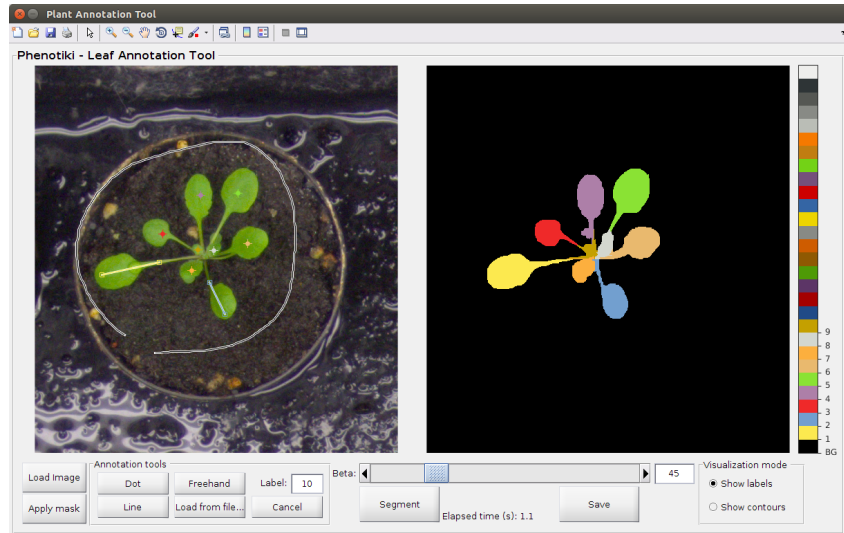
Figure 2: Graphical user interface of LeafAnnotationTool. In this example, we first draw a freehand curve around the plant to indicate background, then we add as many dots as the number of leaves. The underlying random walks segmentation algorithm is able to propagate those seeds and provide an accurate leaf delineation.

2 Accepted image formats

LeafAnnotationTool accepts in input colour images in all image formats supported by MATLAB, such as PNG, TIFF, JPEG, or Bitmap.

It is possible to use a foreground/background segmentation mask (i.e. a delineation of the whole plant against the background). The mask should be a binary image, where 0 denotes background and 1 (or 255) denotes foreground. The image containing the mask can be encoded in any suitable format, such as indexed PNG.

LeafAnnotationTool also allows to import leaf annotations from file, in the form of a binary image where leaf center coordinates are denoted by 1 and zero otherwise, or a binary image containing as many 'disconnected' scribbles as pixel drawings with value 1 and zero otherwise.

3 Using the tool

LeafAnnotationTool provides a user-friendly graphical user interface (GUI) to support interactively experts during the leaf annotation process. Figure 2 shows an example of leaf segmentation obtained using our tool. The typical workflow of the annotation process is described in the following paragraphs.

Load images: First, we load an RGB image of a plant, which is internally converted to the CIE *L*a*b** color space. The software offers the possibility to eliminate background by applying a plant segmentation mask (e.g., obtained by manual delineation or via automated segmentation approaches), that can be loaded from file (see Section 2) using the *Apply mask* button.

Expert annotation: Next, expert annotation is performed using the available *Annotation tools*, in order to mark leaf locations in the image. We permit four types of pixel annotations:

Dot :
> Allows to mark a single pixel as seed;

Line :
> Allows to annotate a set of pixels laying on a line segment;

Freehand :
> Allows to draw free-form curves, useful to annotate highly overlapping leaves; and

Load from file :
> Allows to load annotations from file (see Section 2).

Zooming and panning facilitate the annotation of complicated regions. The last annotation can be deleted by pressing the `Ctrl+Z` key combination. When a new image is loaded, the previous annotations and segmentation results will be cleaned up.

More often than not, a simple dot is sufficient to obtain satisfactory segmentation, however depending on how much the leaves are cluttered and overlapping, in some cases line or freehand curve annotations may be necessary. In general, more detailed annotations lead to a more accurate segmentation.

A different label is assigned to each annotation: 'BG' to the background and integer numbers from 1 to K to the leaves, where K is the total number of leaves to segment. The label counter in the interface informs on which label will be used for the next annotation.

! → If a foreground/background mask has been applied with *Apply mask*, then the label counter is automatically set to 1, meaning that it is already possible to annotate the first leaf. Otherwise, the first label is 'BG' and the first annotation must necessarily be for the background (e. g., a freehand curve around the plant, as shown in Figure 2).

The leaves can be annotated in any order, but if multiple images of the same plant are being processed (e.g., time-lapse sequence of a growing plant), it is recommended to annotate the leaves in the same order, so the leaves will consistently have the same label from one image to another.

Leaf segmentation: A slider allows to set the only free parameter β to control the sensitivity of the segmentation algorithm to variations in intensity between neighboring pixels. Further details are available in [**?**].

When the annotations are in place, pressing the *Segment* button will run the segmentation algorithm and the leaf segmentation result will be produced within few seconds (depending on available computational power, image size, and amount of labeled/unlabeled pixels) and displayed in the right-hand side of the interface (cf. Figure 2).

Results: The GUI offers two visualization modalities, showing either the leaf labeling (cf. Figure 2) or the segmentation outline overlaid on the original RGB image. It is possible to switch between the two visualization modes.

In case of not satisfactory results, the *Cancel* button cleans the image from all the scribbles, and the user can repeat the annotation procedure from scratch. Otherwise, iterative segmentation and refinement is also possible to some extent: it is possible to change the value of β or move (but not edit) the existing annotations, and then press again the *Segment* button to update the results. Currently, it is not possible to edit the contour of the segmentation, but future versions of LeafAnnotationTool may offer this feature.

The leaf segmentation mask can be eventually exported as an indexed PNG image file with the *Save* button. In the output image index '0' denotes background and all subsequent indexes denote leaf labels. A color palette is embedded in the PNG file to map indexes to displayed color, maximizing contrast and improving legibility.