Phenotiki – True phenotyping-in-a-box solution
User's manual

Massimo Minervini[1], *Mario Valerio Giuffrida*[1], *Sotirios A. Tsaftaris*[2]
[1]*IMT Institute for Advanced Studies Lucca, Italy*
[2]*University of Edinburgh, UK*
Manual Version 0.1

June 6, 2016

Abstract

Contents

1 Installation **4**
1.1 Run source code 4
1.2 Install Phenotiki from binaries 5

2 Getting started with Phenotiki **6**
2.1 Pot Tray Analysis 6
2.2 Leaf Annotation 10
2.3 Leaf Counting 12
2.4 Data Extraction 14

3 Troubleshooting **16**
3.1 General 16
3.2 Pot Tray Analysis 17
3.3 Leaf Annotation 17
3.4 Leaf Counting 18

Disclaimer

Software of the *Phenotiki* project is provided 'AS IS' without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of fitness for a purpose, or the warranty of non-infringement. Without limiting the foregoing, the authors make no warranty that:

i. the software will meet your requirements

ii. the software will be uninterrupted, timely, secure or error-free

iii. the results that may be obtained from the use of the software will be effective, accurate or reliable

iv. the quality of the software will meet your expectations

v. any errors in the software obtained from the authors will be corrected.

Software and its documentation made available on the Phenotiki website:

i. could include technical or other mistakes, inaccuracies or typographical errors. The authors may make changes to the software or documentation made available on its website.

ii. may be out of date, and the authors make no commitment to update such materials.

The authors assumes no responsibility for errors or omissions in the software or documentation available from its website.

In no event shall the authors be liable to you or any third parties for any special, punitive, incidental, indirect or consequential damages of any kind, or any damages whatsoever, including, without limitation, those resulting from loss of use, data or profits, whether or not the authors have been advised of the possibility of such damages, and on any theory of liability, arising out of or in connection with the use of this software.

The use of the software obtained by the authors is done at your own discretion and risk and with agreement that you will be solely responsible for any damage to your computer system or loss of data that results from such activities. No advice or information, whether oral or written, obtained by you from the authors or from the Phenotiki website shall create any warranty for the software.

Preamble

This manual explains the usage of the *Phenotiki* software. It will be assumed that you have acquired images using the Phenotiki acquisition device (further details on `www.phenotiki.com`). Nonetheless, this software allows you to analyse time-lapse images of top-view rosette plants obtained with a different acquisition pipeline. Our software is written in MATLAB and we will assume as reference version R2015b.

We are making every effort to fully document our software. Extensive usage instructions are provided in this manual, whereas contextual help can be obtained within the software by clicking on this icon:

1 Installation

Our software is released as stand-alone application pre-packaged for several platforms (Windows, Linux, Mac OS X) or as source code (refer to `www.phenotiki.com` for further details).

1.1 Run source code

Once you have unpacked the archive file containing the source code of Phenotiki, from MATLAB:

1. navigate to the folder containing the unpacked source code (e. g., ~/Desktop/Phenotiki);
2. run the file **PhenotikiMain.m**.

! → At this point, you might be asked to install the following dependencies: **vl_feat** [1] and **libsvm** [2][1].

Install vl_feat: Go to `www.vlfeat.org`, and then go to the download page. Download the *binary package* and save it on your computer.

From your MATLAB, navigate inside the vl_feat folder, go to the *toolbox* folder, and run the file **vl_setup.m**. To check if the installation was successful, navigate to a different folder and type in the command window **vl_demo**.

! → Installing vl_feat will add entries to your MATLAB search paths. Therefore, we recommend to save the new pathdef.m, in order to avoid to repeat the installation of vl_feat every time you run the Phenotiki software. You can do it by typing the **pathtool** command in your command window and pressing the "*Save*" button.[2]

Install libsvm: Go to `https://www.csie.ntu.edu.tw/~cjlin/libsvm/` and follow the instructions to download the libsvm source code.

- If you have Windows (any supported version by MATLAB) 64-bit:
  1. type **pathtool**;
  2. Press "*Add Folder...*" button;
  3. Navigate inside the libsvm folder and select the **windows** sub-folder;
  4. Press "*Save*" button.
- For other systems (Windows 32-bit, Linux 32/64-bit, Mac OS X)[3]:
  1. From MATLAB, navigate to the libsvm folder, then go to *matlab* sub-folder;
  2. type **make**;
  3. If no errors occurred, type **pathtool**;
  4. Press "*Add Folder...*" button;
  5. Navigate to the libsvm folder and select **windows** sub-folder;

[1] Suggested versions: vl_feat 0.9.20 and libsvm 3.20

[2] On certain systems (e. g., Linux, Mac OS X), it might be required to have administrator (i. e. superuser) privileges to overwrite the main pathdef.m file.

[3] Learn more about supported compilers of MATLAB R2015b: `http://www.mathworks.com/support/sysreq/files/SystemRequirements-Release2015b_SupportedCompilers.pdf`

6. Press "*Save*" button.

1.2 Install Phenotiki from binaries

Stand-alone distributions of the Phenotiki software are available for the following operating systems: Windows, Linux, and Mac OS X. The stand-alone version of our software requires the **MATLAB Compiler Runtime**[4] (MCR) installed on your system.

In case the MCR is not already on your system, the Phenotiki installer process will download and install it. For this step, you are required to have an Internet connection during the installation.

! → Superuser permissions may be required to complete this step.

1. Launch the installation executable file
   - **On Windows:** Phenotiki.exe
   - **On Linux:** available soon
   - **On Mac OS X:** Phenotiki.app
2. in the installation program press "*Next >*";
3. choose the installation directory (leave it as is if you are not sure) and press "*Next >*";
4. choose the installation directory for the MCR (leave it as is if you are not sure) and press "*Next >*";
5. accept the License Agreement and press "*Next >*";
6. a summary window will appear: if everything is correct, press "*Install >*";
7. when the installation is complete, press "*Finish*".

You will find Phenotiki inside the folder specified at step 2.

[4] Further information at `http://www.mathworks.com/products/compiler/mcr/index.html`

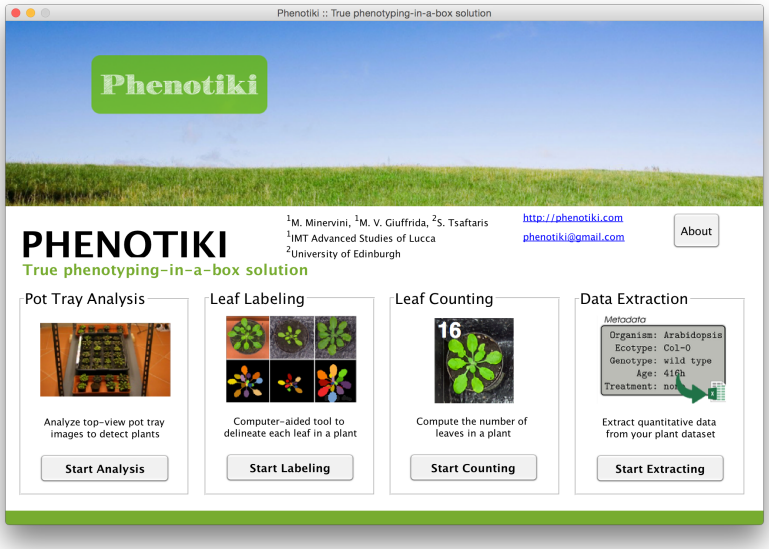

Figure 1: Main window of the Phenotiki software.

2 Getting started with Phenotiki

Once you have obtained and installed a copy of the Phenotiki software, you are ready to launch it. A start-up window will appear, as displayed in Figure 1. From here, you will have access to all available (currently four) analysis modules:

- **Pot Tray Analysis:** this module will detect each plant from the top-view image individually, separating them from the background (e. g., soil, pot, moss);
- **Leaf Labeling:** Phenotiki incorporates a leaf annotation tool [3] (available also as a separate software tool at `www.phenotiki.com`). It allows the user to delineate each individual leaf of a rosette plant;
- **Leaf Counting:** incorporates our machine learning algorithm for counting the number of leaves of rosette plants;
- **Data Extraction:** this module extracts quantitative plant trait descriptors based on the vision tasks performed by the other modules. Extracted data can be studied with specific software (e. g., Matlab or R) for further statistical analysis.

2.1 Pot Tray Analysis

This module assumes as input a sequence of images showing a top view on a population of rosette plants. Execution of this module will produce the following results:

1. detection and segmentation (i. e. delineation) of each individual plant (cf. Figure 2);
2. extraction of the phenotyping traits listed in Table 1.

We recommend the following workflow: *image loading*, *plant centres positioning*, and *parameter settings*. Eventually, pressing on the button "*Start Analysis*" will launch execution.

Table 1: Plant traits extracted by the Pot Tray Analysis module.

Trait	Unit	Description
Projected Leaf Area (PLA)	cm^2	Plant area calculated on the number of visible pixels
Diameter	cm	Longest distance between two points in the boundary
Perimeter	cm	Length of the boundary pixels
Compactness (or solidity)	–	Ranges in [0,1] indicating the solidity of the plant (e.g., absence of holes in the plant surface)
Stockiness (or form factor)	–	Ranges in [0,1] indicating the circularity of the plant
Relative Growth Rate (RGR)	$\%h^{-1}$	Measure of the amount of growth between to consecutive time instants
Color	°	Average value of the Hue (H) component in the HSV color space

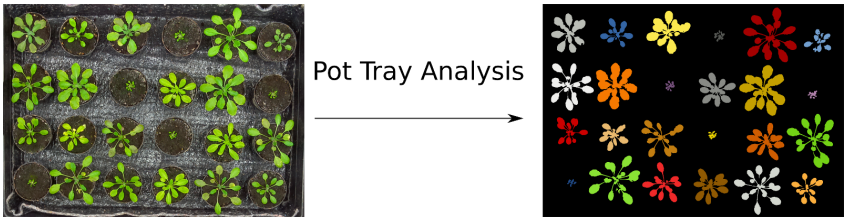

Figure 2: Expected output of the Pot Tray Analysis module.

Load image from Phenotiki Device: The Phenotiki software integrates seamlessly with the device and can easily import images acquired using the Phenotiki device.

1. Press on the button "*Load Phenotiki Acquired Image*";
2. navigate to the folder containing the time-lapse images acquired with the Phenotiki device;
3. after selecting the folder press on "*Open*".

! → Make sure that image files conform to the following filename format: IMG_YYYY-MM-DD_HH-NN.png (cf. Table 2 for further details), which describes the acquisition date and time of that picture. This information is used within the software to automatically sort the time-lapse images according to their chronological order, as well as to extract time-dependent phenotyping traits (further details in Section 2.4).

Import other images: To import time-lapse images acquired with a device different than Phenotiki and that do not conform to the filename format described previously, you need to build an index file first. This operation has to be done externally to Phenotiki software.

For example, follow these steps:

1. open a spreadsheet software (e. g., Microsoft Excel);
2. fill the first column (A) with the filenames of your images;
3. fill the second column (B) with the timestamps referring to the time

Table 2: List of abbreviation used for the image filename format.

Abbreviation	Meaning	Example
YYYY	Year (4 digits)	2016
MM	Month (2 digits)	01
DD	Day (2 digits)	14
HH	Hour (2 digits, 24hr format)	19
NN	Minutes (2 digits)	00

your images were taken (date format YYYY-MM-DD HH:NN, please refer Table 2 for further details);

4. enter as many rows as the images in the dataset (expected result is shown in Figure 3) and finally save your file in Comma-Separated Value format (CSV) *inside* the same folder where your images are located.

Then, in the Phenotiki software:

1. o to the *Pot Tray Analysis* tool;
2. press the "Load Dataset" button;
3. a new window for file selection will appear. In the bottom part, select as file type "*Comma-separated Values files (.csv)*";
4. navigate to the folder containing your images and the created CSV file;
5. choose the CSV file;
6. Press "*Open*".

Placing plant centers: The segmentation algorithm implemented on Phenotiki relies on the prior knowledge of the position of plants [4]. In order to mark the position of each plant:

1. Hold on the keyboard either:
   - **On Mac OS X:** ⌘
   - **On other systems:** ctrl
2. as you hold the aforementioned key, move the mouse cursor on the first plant;
3. press the left-button of your mouse to mark the plant;

	A	B	C
1	IMG_2013-09-28_08-00.png	2013-09-28 08:00	
2	IMG_2013-09-28_19-40.png	2013-09-28 19:40	
3	IMG_2013-09-29_08-00.png	2013-09-29 08:00	
4	IMG_2013-09-29_19-40.png	2013-09-29 19:40	
5	IMG_2013-09-30_08-00.png	2013-09-30 08:00	
6			
7			
8			
9			
10			
11			
12			
13			

Figure 3: Example of CVS for non-phenotiki dataset

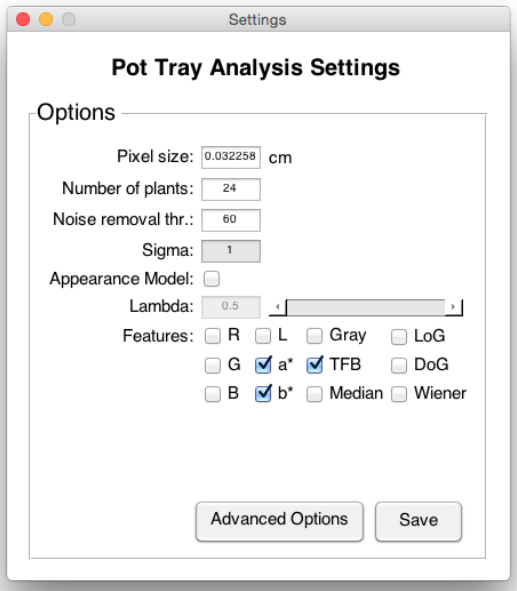

Figure 4: Window showing the set of parameter of the segmentation tool

4. repeat these steps for all plants;
5. when you have finished, release the key you have been holding.

! → For best results, place a plant's mark approximately on its centre and make sure that the mark *IS* in the plant region.

Parameter settings: The module exposes several parameters of the plant segmentation pipeline, that can be tuned in case default values do not produce acceptable results. Different acquisition conditions might require to tune the parameters to best suit the acqired data. Figure 4 shows the interface to set the parameters, while the following table reports a brief explanation of what they stand for. Detailed explanation of the plant segmentation approach and all parameters can be found in [4].

! → The *Number of Plants* is set automatically when centers are placed correctly.

! → Setting these parameters may require some expertise. Please go to Section 3.2 to learn more how to face frequently occurring issues.

Get the results: Press on the button "*Start Analysis*" to launch module execution. Once plant segmentation is done, results will be displayed as in Figure 2. On the right-hand side of the window, the list of plants with their location (in pixel) is shown. From here you can:

- **assign genotype/group name:** by pressing in the corresponding white cell, you can add/modify the name of the group of that plant. This operation is automatically propagated to all the images in the stack, assigning the same group to the same subject. However, you may assign individual group name across time.
- **read data**: by pressing the right button of your mouse on the line of the table of the plant you are interested in, a contextual menu will appear. From here, you select *Properties* to get access to all the

Table 3: Parameters used in the plant segmentation tool.

Parameter	Description	Default value
Pixel size	Size of pixel in centimeters	Read below
Noise threshold	Remove objects having area smaller than a threshold (pixels)	60-80
Sigma	Larger Sigma produces more regular boundaries, but thin petioles could be missed	1
Appearance Model	Use a plant appearance model to adapt to appearance changes	false
Lambda	Influence of the appearance model (if used) compared to image features (value in [0,1])	0.5
Features	List of image features used during the segmentation	TFB, a*, b*

details of that particular subject. At this point, leaf counting is not provided. This will be discussed in Section 2.3.

! → Save your progress! So far your data are not stored on your hard drive. By pressing the button "*Save*", you can save your data. We recommend to do it at each step of your analysis to avoid any data loss. The database of your data is a *.mat* file. To learn how to extract plant traits, go to Section 2.4.

Modality selection: On the right-hand side of the interface, you can find a pop-up menu allowing to switch between different visualization modalities. For this module, you can visualise:

1. *raw image*: the original RGB images;
2. *detected plants:* colour-coded segmentation masks (cf. Figure 2);
3. *contour:* RGB images with contour overlaid on the plants. This modality is helpful to assess visually the goodness of the segmentation;
4. *FG mask:* similar to #2, but all plant masks are white.

Load your data: If you wish to load data from a previous analysis, press the "*Load Dataset*" button and navigate to the folder containing the database file with *.mat* extension.

2.2 Leaf Annotation

This module is largely based (and improves upon) our stand-alone Leaf Annotation tool, for which a detailed manual is available on the Phenotiki website (`http://phenotiki.com/download.html`). The leaf segmentation engine is based on the random walker algorithm [5] and the overall approach is described in [3].

The user annotate the image by drawing scribbles on the leaves (at least a scribble per leaf, for all the leaves in the plant). The suggested workflow for this module is the following: *load dataset*, *leaf annotation*, β *selection*, *segmentation*.

Load Dataset: On the left-hand side of the interface, press on the "*Load Dataset*" button. Once the dialog window is open, navigate to the folder containing the

.mat you prepared beforehand (e. g., from the Pot Tray Analysis). The interface allows you to browse through tray and plant images.

Leaf annotation: Once you have selected the plant to annotate, you will need to draw a scribble for each leaf. The software will use these scribbles to delineate each individual leaf and generate a leaf segmentation of the plant. Scribbles can be of several types:

- *Dot:* a single pixel is marked to belong to a certain leaf;
- *Line:* all the pixels laying on a line segment are marked to belong to a leaf;
- *Freehand:* a freehand line is marked to belong to a specific leaf.

! → If you feel more comfortable with using other third-party image manipulation software (e. g., Photoshop, GIMP), you can also use them to create a binary image for an annotation. You can import such external scribbles by pressing the "*Load from file...*" button.

As you add scribbles, on the right-hand side a list of annotations will be populated. They are organised hierarchically:

i *label:* a label is a container of scribbles identifying a single leaf;

ii *scribble:* one or more scribbles are contained in a label, and each will mark pixels that belong to the same leaf.

In many cases, a single dot scribble is enough to obtain a suitable leaf segmentation. However, mature plants with heavily overlapping leaves may require more detailed annotations to produce an accurate leaf segmentation.

From the list of labels and leaves, you can access a contextual menu (using the right-button of your mouse) allowing you to:

- delete labels;
- delete scribbles;
- change label to a scribble;
- copy labels applied to the same plant at the previous time instant.

! → Each label is assigned a numerical id. When you delete a label, the numerical id of the others will not change. For example, if you have *Label 1*, *Label 2*, and *Label 3*, and you delete *Label 2*, than *Label 3* will not change. This is to maintain label consistency across time.

β selection: This is the only parameter required for this module. It controls the sensitivity of the segmentation algorithm to variations in colour intensity between neighboring image pixels. Higher values of β correspond to lower sensitivity to sudden changes in color intensity. Example values are: 15, 30, 80. Experiment with different values to find the one that produces best results on your dataset.

Leaf segmentation: Once labels have been placed, press the "*Segment*" button in the lower part of the interface. The time to required by the software to generate the segmentation is usually in the order of few seconds and depends on image resolution. When it is done, the software will show you the outcome.

Refine result: If you are not satisfied with the outcome, you may add further scribbles to obtain refined results. For example, you can change a dot annotation

into a line, or even replace it with a more detailed freehand scribble. In some cases, it could be useful to add multiple scribbles to the same leaf label. To add multiple annotations to the same label, go to the list of labels on the right-hand side of the interface and press on the desired label (e. g., *Label 1*). At this point, choose one of the tools (e. g., line) and annotate as before. This new scribble will appear with the same colour of the other scribbles within the same label. If you want add more labels, press on '<*new label*>' from the same list.

! → Note that the number of labels can be used by the *Leaf Counting* module, which will be discussed in the next section. Therefore, if you are more interested in the number of leaves than in obtaining a leaf segmentation, you can use this module to produce "training data" for the learning-based counting algorithm. In this case, using the dot annotation tool is an efficient method to rapidly annotate the number of leaves for several plants.

2.3 Leaf Counting

The Leaf Counting module is able to count the number of leaves in a plant image. It is based on a *machine learning* algorithm, so it can adapt to different types of plants and varying imaging conditions. It can do so by relying on a sufficient amount of labelled data. Therefore, the user has to provide leaf count for several plants, such that the software can learn a model that will be applied to the rest of the dataset.

The suggested workflow for this module is the following: *data loading*, *model training*, *testing*.

Interface: In Figure 5 the Leaf Counting module interface is shown. It has seven panels and they will be referenced next in the manual. A brief explanation is given as follows:

A. **Plant list:** here you can load data that you previously processed with the other modules. The interface lists all plants across all images and provides a summary showing: total number of plants (i. e. number of subjects multiplied by number of time-lapse pictures), training set size (i. e. number of labelled data), and testing set size (i. e. number of unlabelled data).

B. **Learning setup:** here you set which plants will belong to the training and testing set, respectively. Although only labelled plants will be included in the training set, you have the possibility to filter them by group (e. g., to test performance on a specific group of plants). You can also assign a name to a model, such that you can train several models (e. g., corresponding to different training sets or parameters) according to specific needs.

C. **Model parameters:** the set of parameters for the learning algorithm can be tuned from this panel.

D. **Results:** every time a model is learnt, a new entry on the results list will appear. From there, you may check the performance of the learning algorithm, as well as repeat the same experiment with different parameters. When a leaf count is estimated, these values are not automatically applied to the dataset. In order to apply leaf counting estimates to the dataset press on the *Assign Count* button.

E. **Cross-validation:** in order to fast check whether your model is well-trained, you may run a cross-validation. It will split the training

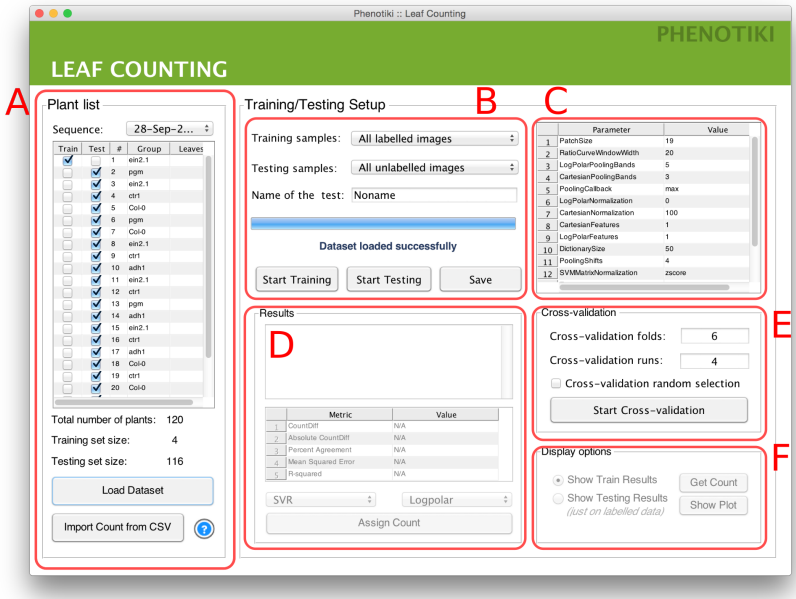

Figure 5: Leaf counting module. Panels are explained thoroughly in text.

set in K folds, out of which $K-1$ are used for training, and the remaining one is left for testing. Since data are randomly split into folds, it is recommended to repeat the cross-validation a certain number of times. Each learned model will appear separately in the results list.

F. **Display:** training and test counts are displayed. Moreover, performance plot for training error may be shown.

Load data: Press on the "*Load Dataset*" button to load data you previously processed with the other modules. The number of count used for the training set can be provided using the following methods:

- **Leaf Annotation Tool:** using the leaf annotation tool, you will also provide the number of leaves in all the annotated plants. This information will be used by the Leaf Counting Tool as training set.
- **Manual prompt:** the list of subject in the Plant List panel (cf. Figure 5) has 5 columns. The last one contains the number of leaves that the software already found in the database (either by previous counts, or leaf annotation tool). By double-clicking on a cell you can input a value.
- **External file:** you may load the number of leaves using an external CSV file. It is a similar import procedure as discussed in Section 2.1. In a spreadsheet program (e. g., Excel), create a column for each subject in the image and record the known leaf count for each time instant. In Figure 6 it is shown graphically how to format the CSV file. Please remember to save the file in *.csv* format.

Model training: Once a certain amount of training data is available (i. e. plants for which the number of leaves is known), you can train the regression model. Firstly, select the training samples (default is all labelled plants, but for example you may train only on specific genotypes), then press on **Start Training**.

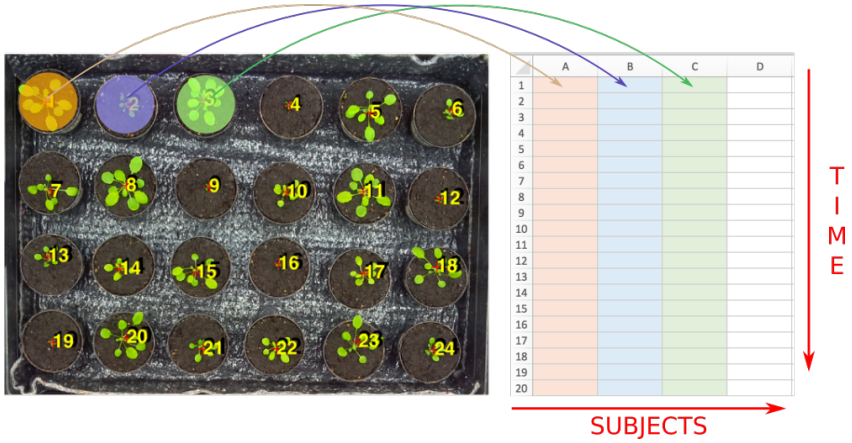

Figure 6: Loading known leaf counting from a CSV file. Using a spreadsheet program (e. g., Excel), you may create your counting file to provide labelled data to the leaf counting module. Each column corresponds to a plant subject, whereas the rows correspond to time vertically you record the number of leaves across time. Plants and columns have to match with respect to Plant ID added in the Pot Tray Analysis module (bold yellow number). Unknown entries have to be marked with **zeroes** '0'. Remember to save the final file in *.csv* format.

Once this operation is completed results will be displayed (Figure 5D). Please refer to [6] for further explanation about the performance measures.

! → The time required for this operation depends on the size of the training set. It may require several minutes to be completed.

! → You may train several models according to different training samples or different parameters. We advice you to keep track of your changes by providing a unique name to each test. Tests with same name will be **overwritten**.

Testing: To count leaves from plants with unknown counting, select a trained model from panel *D*, select the testing set form panel *B*, and finally press **Start Testing** button. Once this process is done, you may plot results from the controls in panel *F* of Figure 5.

! → At this stage, the leaf counting is **not yet applied** to the subjects, because you may want to do several tests before applying estimated counts. Once you are satisfied with the count, press **Assign Count** button in panel *D*.

Parameters: Our leaf counting model may require to the user to set some parameters. A full list of parameters with description is provided in Table 4. Interested readers may refer to [6] for further details. To test the performance of a set of parameters, we suggest to run K-fold cross validation from panel *E* in Figure 5.

2.4 Data Extraction

This module allows you to extract plant traits from the image data analysed using all other modules. You may extract the following data:

- **Plots:** you can plot phenotyping traits across time and save the plot;

Table 4: Description of the parameters of the leaf counting algorithm.

Parameter	Explanation	Suggested value
Patch size	Size of image patches from images that are extracted	1/4 of the average-sized leaf
Ratio Curve Window Width	Size of the sliding window in the log-polar domain to detect patch candidates	As wide as an average-sized leaf
LogPolar Pooling Bands	Number of areas where features are pooled together in the log-polar space	5
Cartesian Pooling Bands	Number of areas where features are pooled together in the Cartesian space	3
Pooling Callback	Function used for the pooling	max (max pooling)
LogPolar Normalization	Rescaling value in order to normalize the height of log-polar images	0 (dynamic scaling)
Cartesian Normalization	Rescaling value in order to normalize the size of Cartesian images	100 (100x100 pixels)
Cartesian Features	Boolean value: 1 means that Cartesian features are used, 0 otherwise	1
LogPolar Features	Boolean value: 1 means that log-polar features are used, 0 otherwise	1
Dictionary Size	Number of clusters for the k-means, which builds the patches dictionary	50-80
Pooling Shifts	Dataset augmentation via pooling shifts	4
SVM Matrix Normalization	Normalization function for the SVM data	zscore

- **Images:** you can save images from the dataset (e.g., segmentation masks, annotated leaves);
- **Raw data:** data extracted from plants are formatted into a CSV file for subsequent analysis in external tools.

2.4.1 Data plot

Once you load your dataset, you can plot data of the phenotyping traits extracted from your plant images. The list of data available for plotting is shown in Table 1. An example of the interface is depicted in Figure 7.

When you plot the data for a specific traits, you may set the span of time. You also have three different modalities of plotting: (i) all, (ii) specific group, and (iii) specific subject. By selecting *all*, the software will plot the mean value as a line of the trait for each group (please go to Section 2.1 to learn how to associate plants to groups); the line will appear in the middle of a wider band, indicating the lowest and higher bound for that trait. Instead, by selecting *specific group*, the software will plot all the subjects individually that belong to a specific group/genotype. You may change the group from the drop-down menu *specify*. By selecting *specific subject*, the software will plot the selected trait of the specified subject. Plots can

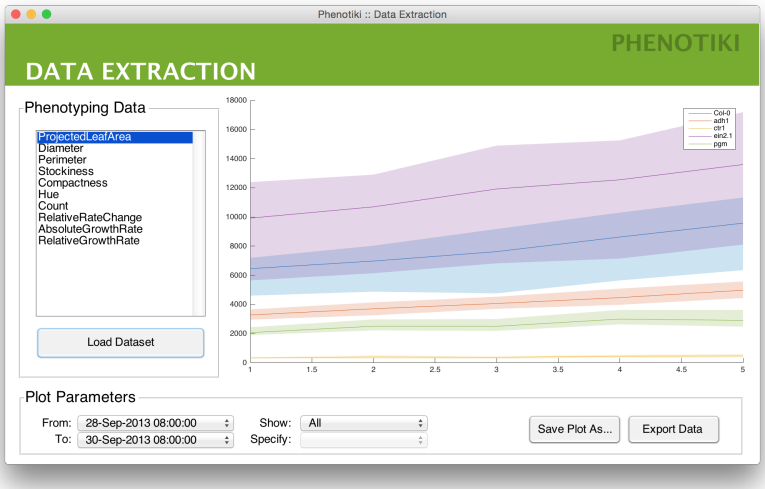

Figure 7: Data extraction tool interface.

be exported by pressing **Save Plot As**. They will be saved as images in PNG format.

2.4.2 Data and images extraction

By pressing on *Extract Data*, you launch the export wizard to extract data or images.

Extracting data: To extract data, select the item *Data* and then press **Next**, then select which traits you want to extract. You can choose multiple items (press **Select all** if you want all of them) and then press **Next**. At this point, the software will ask you to choose the folder where data will be saved.

! → Each trait will generate a different CSV file.

Extract Images : to extract images instead, choose the item *Images* and then press **Next**. You can extract either *Individual Plants* or *Entire Trays*. Once you have done your choice, press **Next** and select the folder where you want to save the images.

3 Troubleshooting

3.1 General

How can I move the data from another workstation? If you wish to share data with colleagues or move the data to another workstation, you simply need to copy the plant's image files and the database file with *.mat* extension that was generated by the modules. When you load your data from the new workstation, the software will ask you to provide the new path where images are located.

! → This procedure is applicable also in case you move the folder of the images somewhere else on the same computer.

3.2 Pot Tray Analysis

Cannot find suitable parameters. What can I do? Finding good parameters for the plant segmentation may not be immediate for challenging datasets. If you have fair results and you want to improve them, then we suggest to select a few images to trial different configurations. In the unluckily case that segmentation results are not accurate enough, we suggest to run a *grid search*. With this method, the software will automatically test a range of parameters and it will find the best ones according to some supervision.

First, segment manually a few of plant images using an external image editing software (e. g., GIMP, Photoshop). Create a black-and-white (also known as binary) image with just 0's and 1's, where 0 (black) means background and 1 (white) means plant. When you are done with this, open the Phenotiki Analysis Software and go to Pot Tray Analysis tool and load your dataset. Then, from the list of images, select one plant image for which you have manually created the foreground/background segmentation and select it by with the right-button of your mouse. A contextual menu will appear and press *Set ground-truth*. When the file selection window appears, select the corresponding mask for this image and repeat this operation for all the plant images for which you have obtained the segmentation mask.

Then press on the **Setting** button in the bottom part of the Pot Tray Analysis Tool and press on **Advanced Options** (cf. Figure 4). The full list of parameters will appear and you may choose the ones you want to include in the grid search, by ticking the check-boxes in the list. When you choose a parameter, you may define a range of values that the software will test against the ground-truth you have provided before. Ranges can be set in the following way.

- **numeric values:** MATLAB syntax is accepted. Some examples are:
  - $[0, 10, 20, 30, 40, 50]$ you specify manually the values to test;
  - $0 : 10 : 50$ you specify all the values between 0 and 50 with a step of 10 values;
- **text values:** cell array MATLAB syntax is accepted. Some examples are:
  - *{Feature1, Feature2}*: the software will use *Feature1* and *Feature2* separately;
  - *{ {Feature1,Feature2}, {Feature1,Feature3} }*: the software will use *Feature1* and *Feature2* at the same time, and then *Feature1* and *Feature3* at the next test;
- **Boolean values:** for Boolean values you do not need to specify range of values, as they can be either 0 or 1.

! → The more images have the ground-truth, the better it is in terms of accuracy, even though the grid search will take longer.

3.3 Leaf Annotation

A leaf is split in half by another leaf. What should I do? In this case, we suggest to add (at least) two annotations for the same leaf. Once you annotate one part of a leaf and a new label is created, select the newly created label from the list to keep adding annotations. You will have two

or more annotations with the same label. To annotate a new leaf, select <*new label*> from the list of label.

I would like to keep consistent labels across time. If you want to assign the same label to the same leaf across time, we recommend to start the annotation from the first time instant (or the last). Then, move to the next time instant and from the list of labels click with the right-button of your mouse and select *copy from the previous plant* (or *copy from the next plant*). In this way, annotations are copied with consistent numeric identifiers. Then, edit the copied labels to adjust to growth and movement of the plant and add new labels if necessary.

Part of a leaf is assigned to another leaf. Use more refined annotation (such as lines or freehand scribbles) to better delineate the image region of each leaf.

3.4 Leaf Counting

Coming soon.

Copyright

References

[1] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008, http://www.vlfeat.org/.

[2] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*,

vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[3] M. Minervini, M. V. Giuffrida, and S. A. Tsaftaris, "An interactive tool for semi-automated leaf annotation," in *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP) Workshop*. Swansea, UK: BMVA Press, Sep. 2015, pp. 6.1–6.13.

[4] M. Minervini, M. M. Abdelsamea, and S. A. Tsaftaris, "Image-based plant phenotyping with incremental learning and active contours," *Ecological Informatics*, vol. 23, pp. 35–48, Sep. 2014.

[5] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[6] M. V. Giuffrida, M. Minervini, and S. A. Tsaftaris, "Learning to count leaves in rosette plants," in *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP) Workshop*. Swansea, UK: BMVA Press, Sep. 2015, pp. 1.1–1.13.