

Multicellular Analysis Processing Platform for Experimental Research

User Manual and Guide

This document serves as a how-to guide for the [MATLAB®](#)-powered tool developed for the manuscript entitled: “MAPPER: An open-source, high-dimensional image analysis pipeline unmasks differential regulation of *Drosophila* wing features.” For more information, please see the associated [publication](#) and corresponding [GitHub](#) page.

Nilay Kumar¹⁺, Francisco Huizar¹⁺, Trent Robinett¹, Keity J. Farfán-Pira², Dharsan Soundarrajhan¹, Maria Unger¹, Pavel Brodskiy¹, Marcos Nahmad², Jeremiah J. Zartman^{1*}

¹Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, IN 46556, USA. ²Department of Physiology, Biophysics, and Neurosciences, Center for Research and Advanced Studies of the National Polytechnical Institute (Cinvestav), Mexico City, Mexico 07360. *Correspondence: jzartman@nd.edu, (+1) 574-631-0455. ¹These authors contributed equally to this work.



UNIVERSITY OF
NOTRE DAME

CHEMICAL
AND BIOMOLECULAR
ENGINEERING

Table of Contents

Preface	5
Why was MAPPER created?	5
Who can use MAPPER?	5
How can I contribute to future iterations of MAPPER?	5
Troubleshooting inquiries	5
Step One: Install MATLAB and the necessary toolboxes	6
Downloading MATLAB®.....	6
Installing MATLAB®	6
Selecting and installing necessary toolboxes	7
(Optional) If MATLAB® version 2019b or later is already installed	7
Step Two: Download the latest version of MAPPER	8
Step Three: Place your data into MAPPER	9
Understanding MAPPER's folder structure.....	9
batch_output_files.....	9
demo_batchProcessing.....	9
demo_hingeRemoval_Uncropped.....	9
demo_individualProcessing	9
dependencies	9
GUI_Logos.....	9
individual_output_files.....	10
user_Data	10
Placing your data properly into MAPPER	10
Step Four: Wing hinge removal of raw images	11
Step Five: Image processing with U-Net or ILASTIK	14
Downloading and installing Python for U-Net classifier.....	14
Training of U-Net for segmentation of the wing images	15
Loading a pre-trained model for making predictions on new image data	16
Downloading and installing ILASTIK pixel classifier.....	16
Selecting a classification module.....	16
ILASTIK module 1 (One channel - Grayscale)	16
ILASTIK module 2 (One channel - Grayscale)	17
ILASTIK module 3 (Three channels - RGB)	17
ILASTIK module 4 (One channel - Grayscale)	17

ILASTIK module 5 (One channel - Grayscale)	17
ILASTIK module 6 (Three channels - RGB)	18
Processing images in ILASTIK	18
Step Six: Running MAPPER	24
Understanding the input parameters	24
Size of wing filter.....	24
Size of intervein filter.....	24
Threshold distance trichome	24
Number of harmonics.....	25
Conversion factor.....	25
Output spreadsheet save name	25
Individual wing processing	25
Loaded raw image	25
Labeled output image	25
Biological axes display.....	26
User operation	26
Biological axes	26
Individual intervein output	26
Global statistics.....	26
Biological axes length	26
Enable unique output label checkbox.....	26
Running an example individual wing	26
Batch processing.....	31
Step Seven: Accessing and understanding your output data.....	32
Location of output files from batch processing.....	32
Understanding the wing feature measurements output.....	33
Column A: Wing name	33
Column B: Genotype.....	33
Columns C—L: Intervein areas 1—10.....	33
Columns M—V: Intervein trichome counts 1—10.....	34
Column W: Total wing area.....	34
Column X: Total trichome density	34
Column Y: AP axis	34

Column Z: PD axis	34
Column AA: d(L3-L4)	35
Understanding the EFD coefficients output	35
(Optional) Train your own ILASTIK module	36
General advice for ILASTIK.....	36
Obtaining the correct folder structure	36
Starting a new ILASTIK project	36
Setting up the training data and features.....	37
Training the pixel classifier	39
Understanding ILASTIK training tools	41
Navigation Cursor (1).....	41
Brush Cursor (2)	41
Eraser Cursor (3)	41
Size (4)	41
Live Update (5)	41
Current View (6).....	41
Probability checkbox (7).....	41
Segmentation (8)	41
Selected label (9)	41

Preface

Why was MAPPER created?

The Multicellular Analysis Processing Platform for Experimental Research (MAPPER) tool was created to enable high-dimensional data analysis of wing phenotypes using *Drosophila* as a model organism. The unique capability of MAPPER to compartmentalize the *Drosophila* wing and provide diverse data for each compartment such as area, trichome counts, and landmark region measurements can allow researchers that utilize *Drosophila* to discover subtle phenotypes that are tedious to obtain by hand-measurements alone.

Who can use MAPPER?

MAPPER was created with versatility in mind for all researchers in the *Drosophila* community. Even those with little background or experience in MATLAB or coding can implement MAPPER with the streamlined Graphical User Interface (GUI) and easy-to-access data output. Thus, the *Drosophila* community as a whole that comprises of biologists, geneticists, engineers, and bioinformaticians to rapidly obtain high-dimensional fingerprints of their *Drosophila* wing images. Modifiable parameter inputs on the MAPPER GUI enable processing of images from different imaging microscopes to further lessen restrictions on implementation of MAPPER for different *Drosophila* research groups. MAPPER's distinct image processing capabilities additionally enable any image orientation of the wing to be processed.

The latest version of MAPPER is the initial release V1.0.0 that is usable for Windows/PC, MacOS, and Linux/Ubuntu users of MATLAB. The tool was built using MATLAB 2019b. This requires users to have at least MATLAB version 2019b installed or later to run the tool.

How can I contribute to future iterations of MAPPER?

Because MAPPER utilizes deep-learning and machine-learning for image segmentation of the *Drosophila* wings, we are in constant need of new images or ILASTIK pixel-classification modules (see step four below) that can help us train our models for future iterations of MAPPER. You can view the types of wings we have already processed on our [GitHub page](#) and their corresponding ILASTIK modules. If you have a new ILASTIK module trained on a unique set of data, we would be more than happy to feature it on our GitHub page for others to have open-access to! We will highlight your institution, department, and research group alongside your ILASTIK module to ensure proper credit is given.

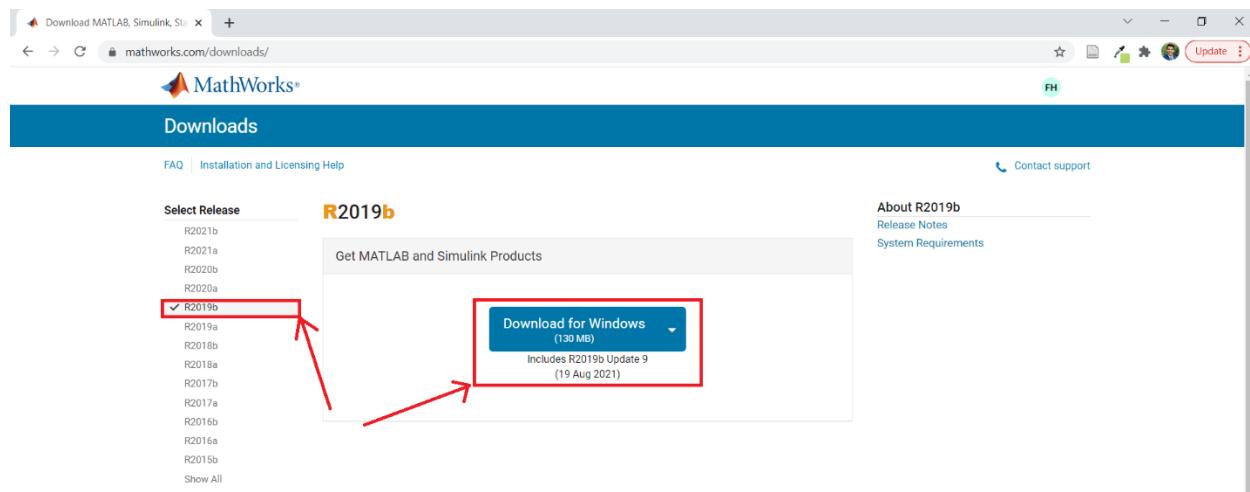
Troubleshooting inquiries

If you encounter any issues while trying to implement MAPPER, we are open to helping you troubleshoot the issue and implement fixes to ensure MAPPER's usability. All troubleshooting inquiries should be filled out using [this Google Form](#). We will do our best to actively monitor incoming inquiries, however, if it takes longer than one week for us to respond, please reach out to the lab PI (Dr. Jeremiah Zartman) at izartman@nd.edu. To ensure we can best address your issue, please explain in detail the issue you encountered, what steps you take that reproduce the error, or any associated MATLAB error messages that appear.

Step One: Install MATLAB and the necessary toolboxes

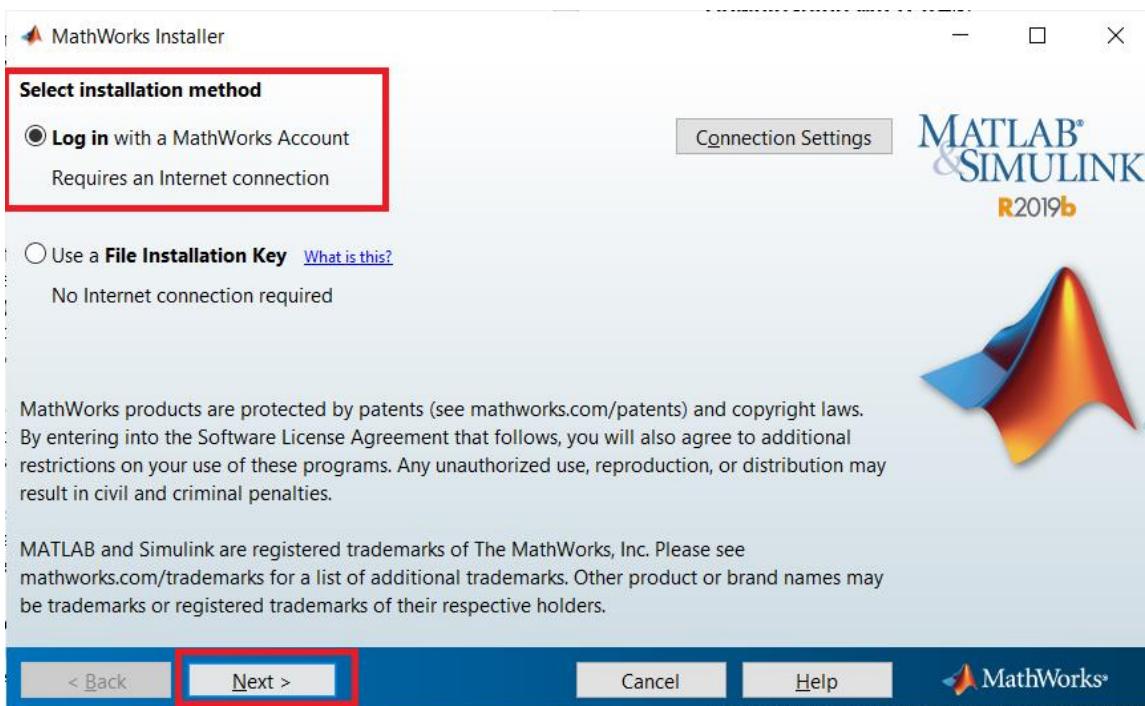
Downloading MATLAB®

In order to run MAPPER, you will first need to download and install MATLAB® from the [official MathWorks website](#). To ensure proper functionality of MAPPER, we recommend installing MATLAB® version 2019b due to MAPPER being created with version 2019b. Later releases of MAPPER are acceptable but may encounter version-specific errors. Download the version of MATLAB® suitable for your operating system (Windows, MacOS, or Linux) and run the installer.



Installing MATLAB®

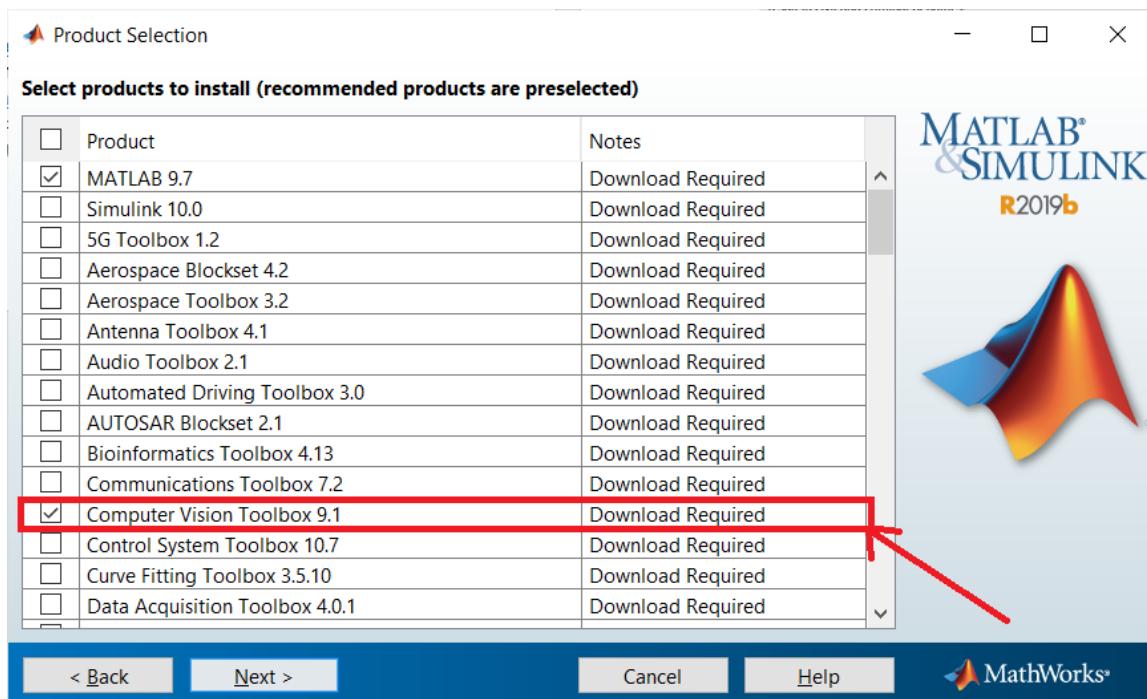
Log in to your MathWorks® account upon the installer's prompt and follow installation instructions.



Selecting and installing necessary toolboxes

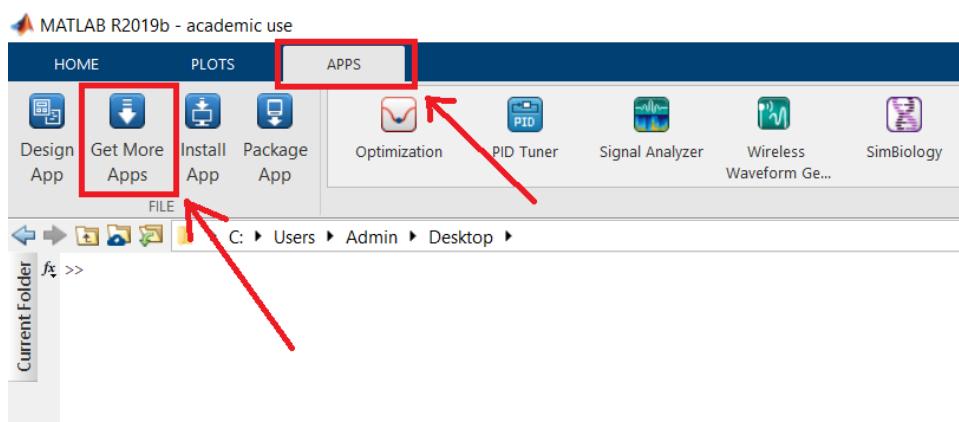
In order to run MAPPER, you will need to have installed the following three MATLAB® toolboxes:

1) [Computer Vision Toolbox](#), 2) [Image Processing Toolbox](#), and 3) [Statistics and Machine Learning Toolbox](#). During the installation process of MATLAB®, you will be prompted for installation of additional product selection. To install the toolboxes, please ensure you fill out the checkbox next to the required toolbox name for all three toolboxes and continue through the installation instructions.

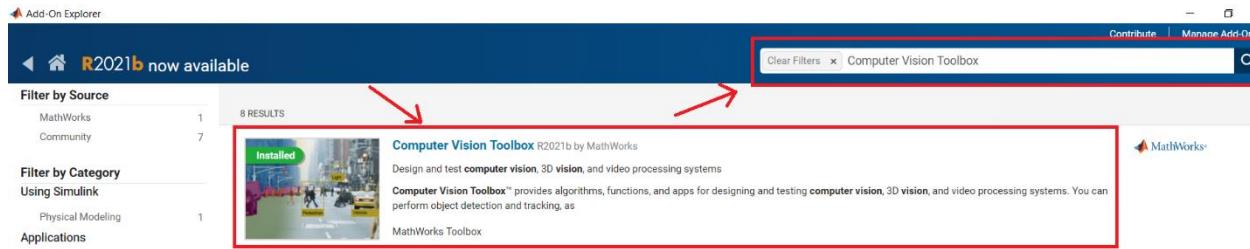


(Optional) If MATLAB® version 2019b or later is already installed

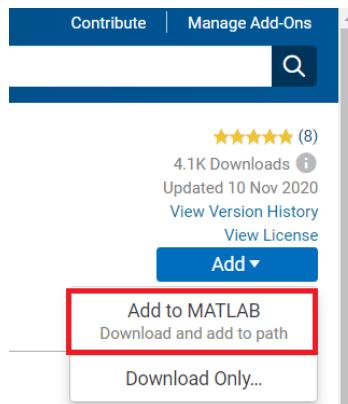
If MATLAB® is already installed, you can easily install the required toolboxes using the GUI and toolbars available in MATLAB®. Open the MATLAB® software application and navigate to the “APPS” section of the toolbar and select “Get More App” in the upper left-hand corner.



In the new “Add-On Explorer” enter the name of the required toolboxes into the upper right-hand search box and navigate to the toolbox.



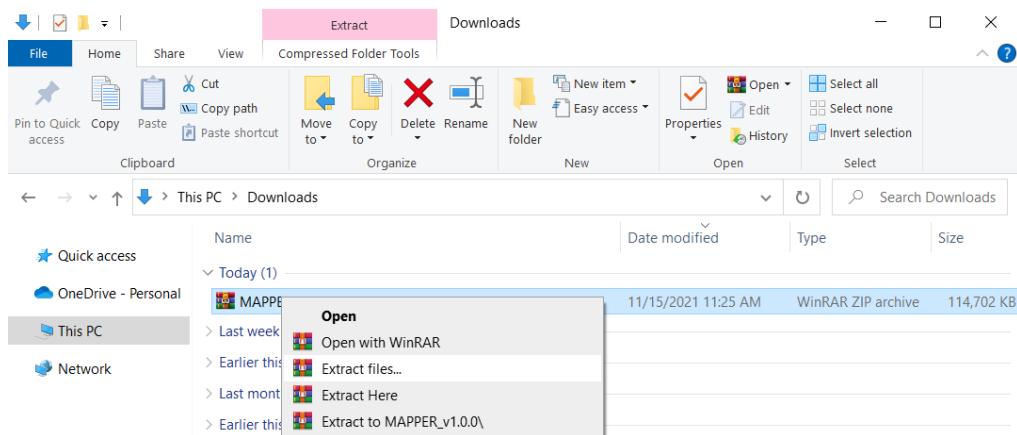
After clicking on the correct toolbox, in the upper right-hand corner select “Add” > “Add to MATLAB, download and add to path.” Follow and complete the installation instructions that proceed.



Complete this for all three toolboxes, then close and restart MATLAB.

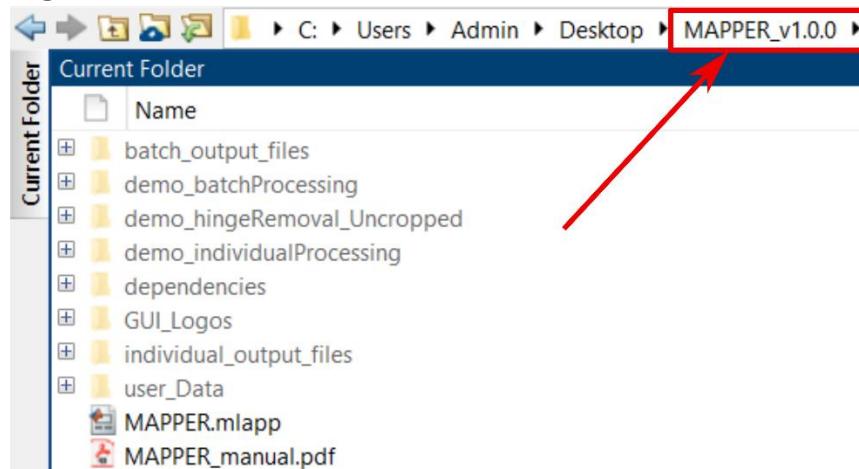
Step Two: Download the latest version of MAPPER

You can download the latest version of MAPPER [here](#). Once you have downloaded the .ZIP folder, extract the folder to an easily accessible location.



Step Three: Place your data into MAPPER

Understanding MAPPER's folder structure



MAPPER's folder and sub-folder structure was designed to contain only the application (MAPPER.mlapp) and user manual (MAPPER_manual.pdf) in the outermost folder. **DO NOT** modify contents of the sub-folders (other than the "user_Data" sub-folder) to ensure proper functionality of MAPPER. Each sub-folder contains the following:

batch_output_files

This sub-folder contains MAPPER's output after processing and extracting your data. It will contain ".csv" files and ".mat" files of your data for you to analyze at your discretion. Step seven of this manual will go more into detail of the contents of the files.

demo_batchProcessing

This sub-folder contains images and files that can be used for demonstration purposes of MAPPER's batch processing capabilities. Step six of this manual will go into more detail of how to execute batch processing.

demo_hingeRemoval_Uncropped

This sub-folder contains images that can be used for demonstration purposes of the wing hinge removal step of MAPPER. Step four of this manual will go more into detail of how to implement wing hinge removal of your raw data.

demo_individualProcessing

This sub-folder contains images and files that can be used for demonstration purposes of MAPPER's individual image processing capabilities. Step six of this manual will go into more detail of how to execute individual image processing.

dependencies

This sub-folder contains all MATLAB® functions necessary to run MAPPER. Modifying any of the contents of this sub-folder has an increased likelihood of breaking the functionality of MAPPER. **DO NOT** modify the functions within this sub-folder unless you are familiar with MATLAB® and the inner workings of MAPPER.

GUI_Logos

This sub-folder contains images that are displayed on the MAPPER GUI.

individual_output_files

This sub-folder contains a labeled segmentation of a wing processed by individual processing. The sub-folder additionally contains “.mat” files of the labeled image, the raw image, and the segmentation of the image. These files serve for troubleshooting purposes of individual wings.

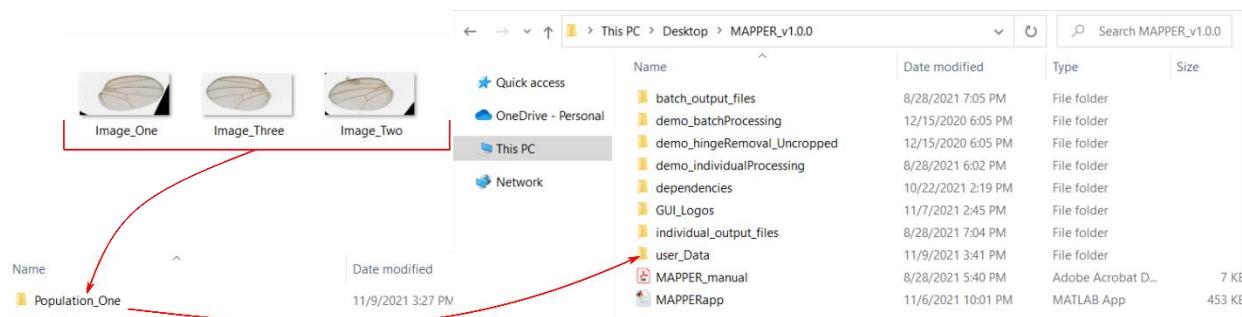
user_Data

This sub-folder is empty upon the first download of MAPPER and will be where you place your raw images of *Drosophila* wings.

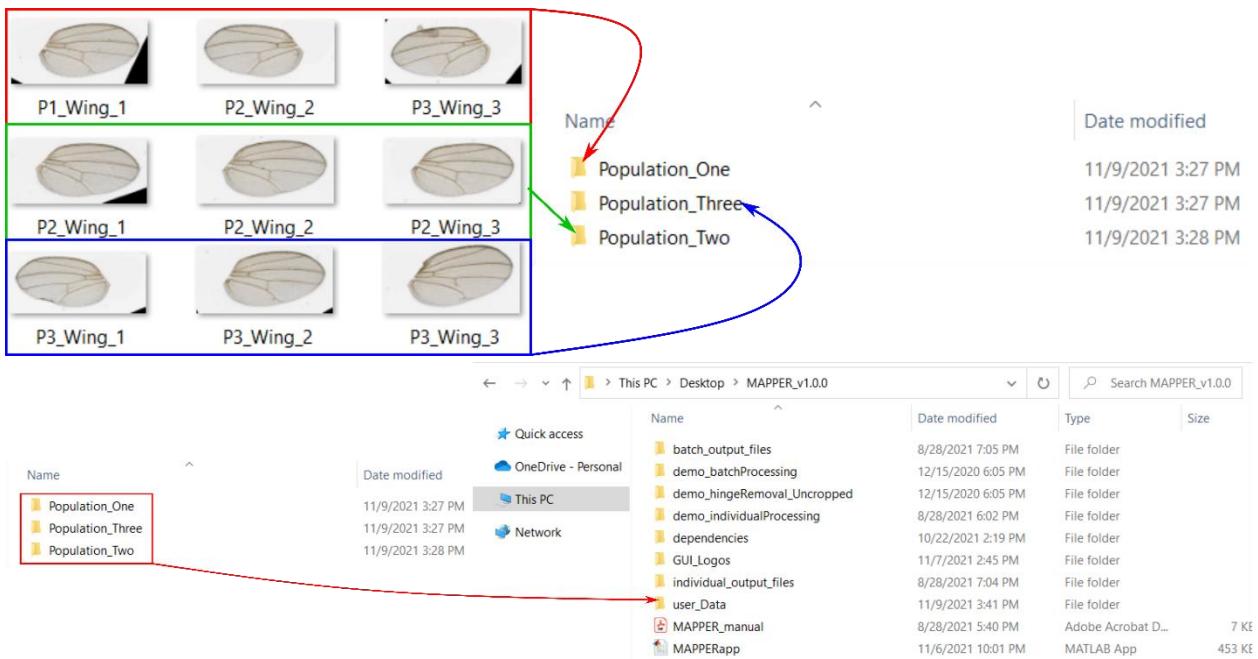
All sub-folders of the main MAPPER folder will be transparent upon starting MATLAB® because they are not yet accessible to the MATLAB® path. This is automatically taken care of upon running the MAPPER application.

Placing your data properly into MAPPER

Raw wing images (.TIF) format can be copied and pasted into the “user_Data” folder for individual wing processing (single images that need to be analyzed). However, for batch processing (analyzing a collection of images), there must be a specific folder structure for MAPPER to be able to process the images correctly. For a single set of images from one population of collected images, you can have one folder containing your raw images. This folder can be moved to the “user_Data” folder or copied/pasted to the “user_Data” folder.

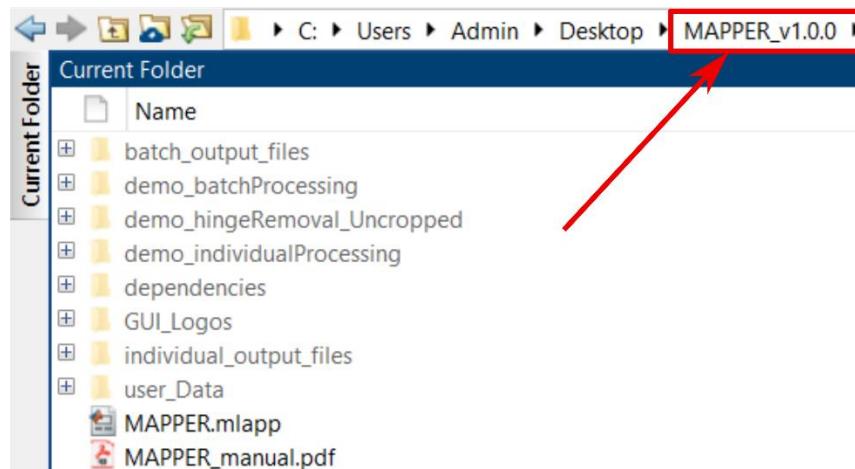


For multiple sets of images (e.g., images from multiple populations of flies), label each population with a different name for you to be able to identify them. Additionally, it may be beneficial for you to label the raw images in a manner that allows them to be connected back to the correct population the images come from. Each folder should contain the raw .TIF images for the population needing to be processed. All folders should then be moved or copied/pasted to the “user_Data” folder.

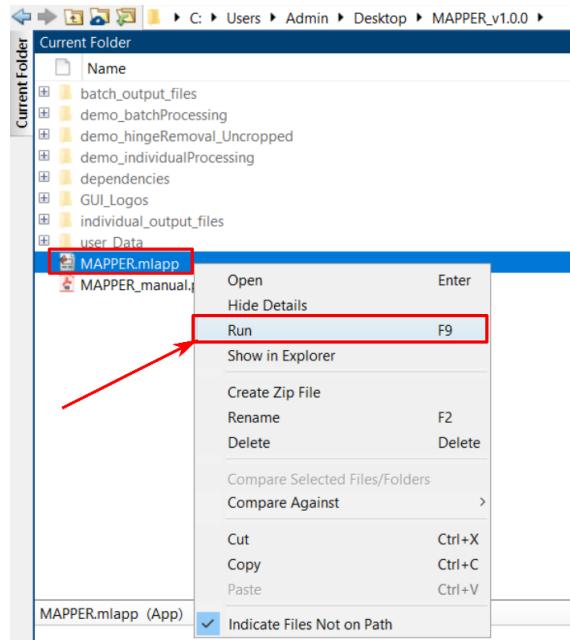


Step Four: Wing hinge removal of raw images

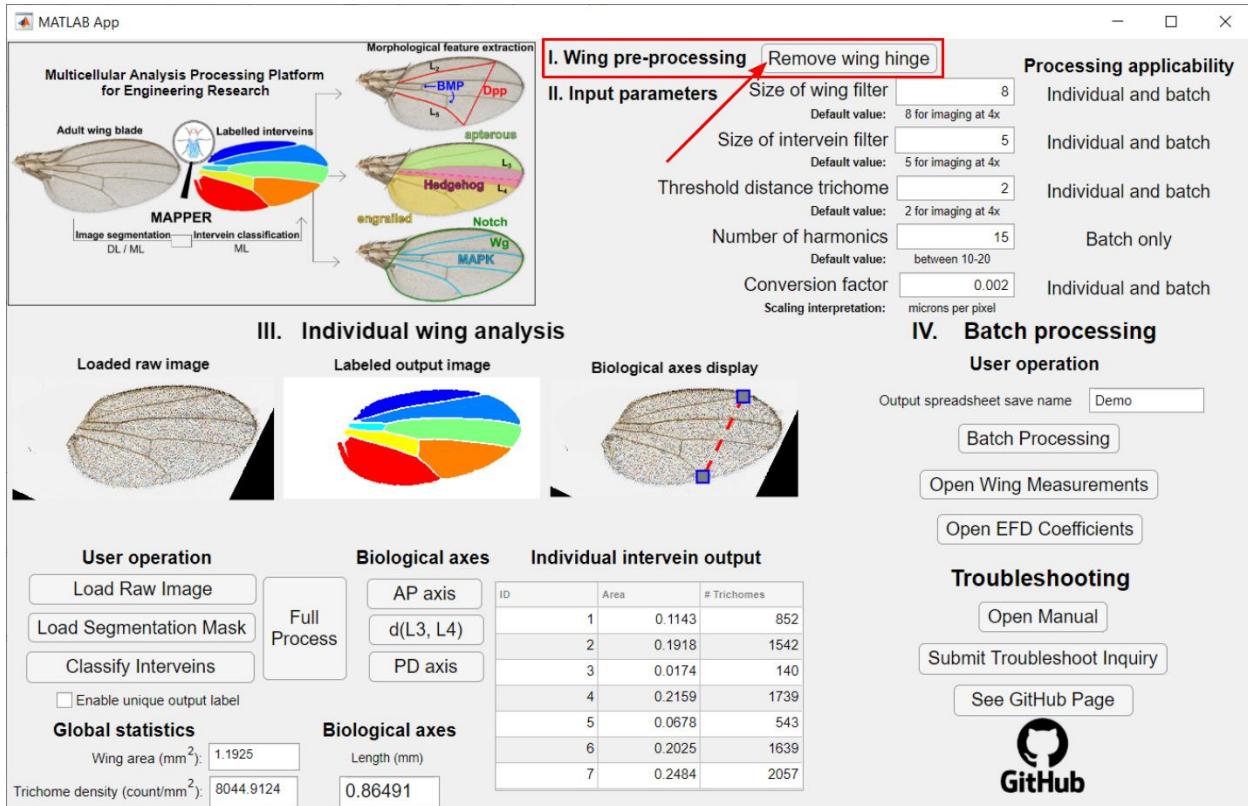
Start the MATLAB® software and navigate to the location where you have placed the main MAPPER folder.



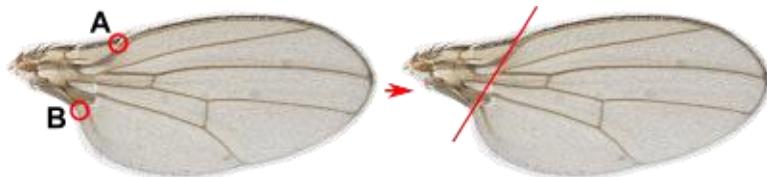
Right-click the “MAPPER.mlapp” and select “Run” to start the application and have the GUI available.



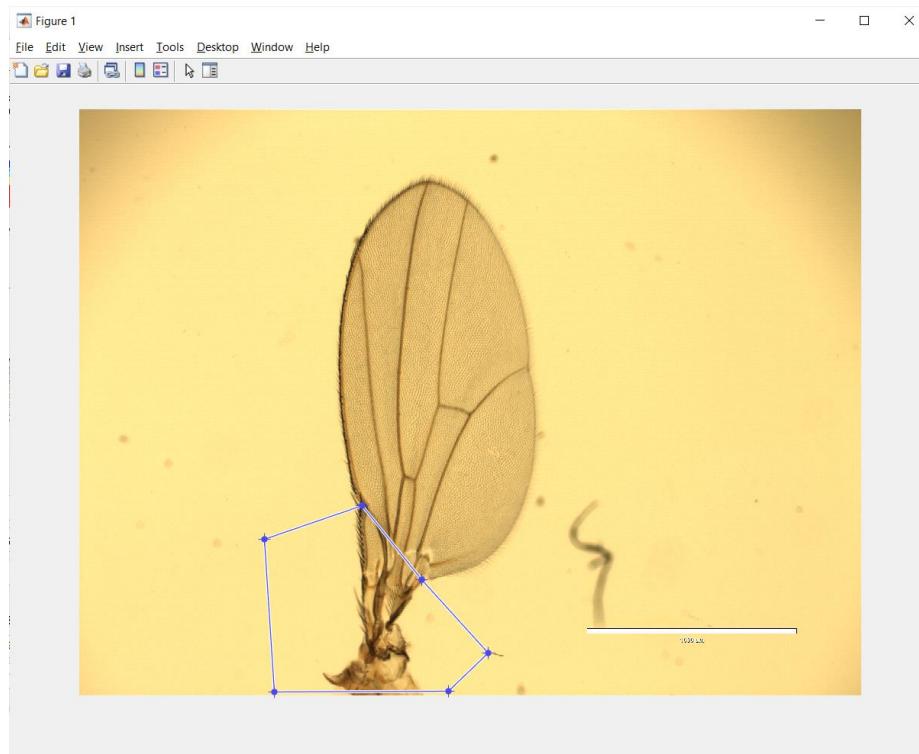
In section **I. Wing pre-processing** of the GUI in the upper-right corner, click the “Remove wing hinge” button. This will bring up a prompt for you to navigate to the folder containing raw images you would like to process. Alternatively, you can select the “demo_hingeRemoval_Uncropped” subfolder in order to practice wing hinge removal of demonstration wings provided.



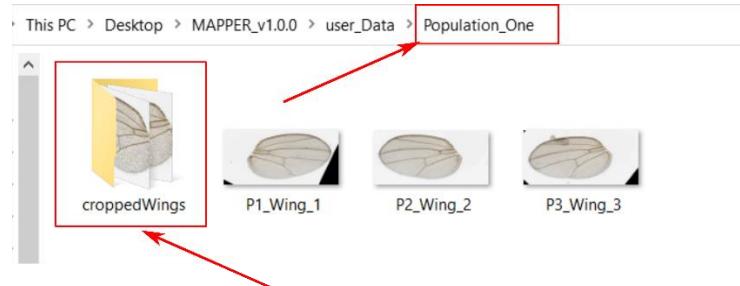
After clicking the “Remove wing hinge” button and selecting a folder with data, a MATLAB® Figure with the first image in your folder will appear. You will need to click points that crop out the wing hinge region of the adult *Drosophila* wing. For our images, we have identified a suitable cutout region for the wing hinge by drawing an extended line through the subcostal break (point A in the below image) and through the costal break (point B in the below image).



To do this in MAPPER, each point you click on the Figure will generate a blue point on the figure. You will need to generate a polygon surrounding the wing hinge region using subsequent clicks. We have found the easiest method to do this is to have the first two points generated by clicking be the subcostal and costal breaks. After finishing creating a polygon around the wing hinge, double click the starting point to proceed with processing of the next image. Remove the wing hinge from each image in the folder using the same procedure.



After removing the wing hinge from all of your images, a new folder containing the cropped images named “croppedWings” will appear in the folder you initially selected to process.



If you are processing a single folder, you can proceed to the next step. If you are processing multiple folders/populations, you will need to perform the wing hinge removal step for each of the folders containing the raw wing images you would like to process.

Step Five: Image processing with U-Net or ILASTIK

In order to obtain wing segmentation masks, you can either use our pre-trained [U-Net model](#), or the ILASTIK pixel classifier. Details on how to use either are detailed below. U-Net should be the first approach used if the wings are images at a high enough resolution (4X magnification or greater) and the wings do not contain imaging artefacts (tissue, tears, mounting media etc.). If issues arise when using the segmentation masks generated, proceed to try use ILASTIK for wing segmentation. **NOTE:** You only need to use one of the methods, either U-Net or ILASTIK to obtain a segmentation mask to be processed in MATLAB.

Downloading and installing Python for U-Net classifier

In order to properly segment the input *Drosophila* wings with the U-Net model, you will need to [download](#) and [install Python](#). In order to run U-Net, you must have the following Python [libraries installed](#):

- [numpy](#)
- [scipy](#)
- [itertools](#)
- [matplotlib](#)
- [seaborn](#)
- [PyTorch](#)

Once Python is installed, you can download our pre-trained, public U-Net model folder [here](#). The full folder contains:

- Image training data in the “imagedata” subfolder
- The python code for training the U-net in an iPython notebook labeled “Deep learning-Unet-CrossEntropy-Four_classes”
- The python code for using the U-net in an iPython notebook labeled “Model Usage”
- A presentation on utilizing U-Net for *Drosophila* wing segmentation in a Microsoft Powerpoint labeled “wingSegmentation”

Training of U-Net for segmentation of the wing images

The training data for U-Net is stored inside the “imagedata” subfolder. The training data comprises of a raw wing image along with corresponding ground truthsm indicating different subcomponents desired to be segmented out during the process. The current model has been trained for identification of four different components namely the intervein, veins, wing marginal hairs and the background from which these components have to be separated. Each of these components has been labelled as number 1-4 denoting their classes. Currently the model has been trained only for segmenting these subsections of wings from image data generated using a medical grade slide scanner whose details are available in the main and SI text of the paper.

For new data that is not being properly segmented by the pre-trained model, the users can create ground truth labels manually either using a package like MATLAB’s image labeler or using a semi-automated approach of generating training data like ILASTIK. It should be noted that the ground truth labels have to be generated for the wing images where the hinge has been already removed using the MAPPER app. The users can select a certain number of wings (~50) from the data for which the segmentation mask has been generated. They can then train ILASTIK to classify these subsections within the wing images. The generated segmentation masks at the desired levels of accuracy can then be used to update the U-Net. Details of using ILASTIK for segmentation of wing images can be found in the subsequent sections. The model can then be trained using the [Jupyter](#) notebook titled “Deep_learning-Unet-CrossEntropy-Four_classes.ipynb”. The model should be saved with a name and the “.pt” extension for further use. The aforementioned notebook has been split into the following components:

- Imports – This section imports all the libraries needed for training the U-Net
- Loading data – The section loads in the raw images and the corresponding ground truth images for the purpose of training the U-Net model.
- Look at the training data: all have the same shape – Visualizes the reshaped input data and the corresponding ground truth labels. The images used for training are of different sizes and hence need to be reshaped to one size so as to be compatible as input for the U-Net. All the images are reshaped into size 128x128 pixels.
- Network architecture – Defines the architecture of U-Net.
- Training phase – Training of weights for the corresponding U-Net architecture is carried out using PyTorch.
- Model evaluation (accuracy) on test data – The section uses seaborn to visualize the confusion matrix for the classification process.
- Saving model – This section saves the trained model as “.pt” files. These files can then later be used for classification of any new wing image data.
- Test examples – Visualizes the predictions made by the trained model on the images generated using the slide scanner.
- Image from web (outside our lab) – Visualizes the predictions made by the trained model on a random data obtained from the web.

Details about the model and training can be found in the main text and the presentation tiled “wingSegmentation.pptx”

Loading a pre-trained model for making predictions on new image data

The models generated during the training process are saved in the “model” subfolder. The notebook “Model Usage.ipynb” loads these pretrained models and makes prediction on any new image data. The notebook has been subdivided into the following sections whose details are provided below:

- Imports - This section imports all the libraries needed for using the pretrained model for image segmentation
- Network architecture – Defines the architecture of the U-Net model. This is necessary as the pretrained model contains the weights that needs to be linked to the model architecture for making predictions.
- Loading model – Loads the pretrained U-Net model. As an example, in our notebook, we use the “model_all_labels_train_1000.pt” model file for making predictions with our pre-trained model on 1000 wing images.
- Image from web (outside our lab) – The section applies the trained model into the desired image and saves it as a “.png” file for further use in the MATLAB-based MAPPER application. The desired image is loaded at “In [21], Line 2” and the segmentation mask is saved at “In [23], Line 12” of the section.

Downloading and installing ILASTIK pixel classifier

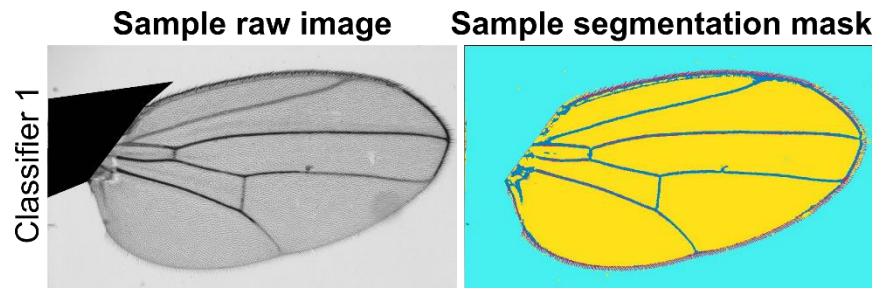
In order to properly segment the input *Drosophila* wing images using a pixel classifier, you will need to utilize the [Pixel Classification workflow using ILASTIK](#). You must first [download](#) and [install](#) ILASTIK for your operating system.

Selecting a classification module

We have pre-trained pixel classification modules utilizing ILASTIK for several wing images that have already been processed. As we train new modules and new modules are provided to us from the *Drosophila* community, we will iteratively add to our collection of pre-trained modules available for you to use on our [GitHub](#) page. In this manual, we provide links to download six pre-trained ILASTIK modules. Below each module link, you will find a representative image of the *Drosophila* wings that were used to train the module. You should download and use the ILASTIK module that has the closest resemblance in lighting, background, brightness, contrast, and saturation to the images you would like to process. If none of the available ILASTIK modules closely resemble the images you would like to process, there are detailed instructions in a later step on how to train your own ILASTIK module. **NOTE:** The number of channels of your images must match the number of channels in the training data for the ILASTIK module you choose (*i.e.*, RGB channel images must have an ILASTIK module trained on RGB images).

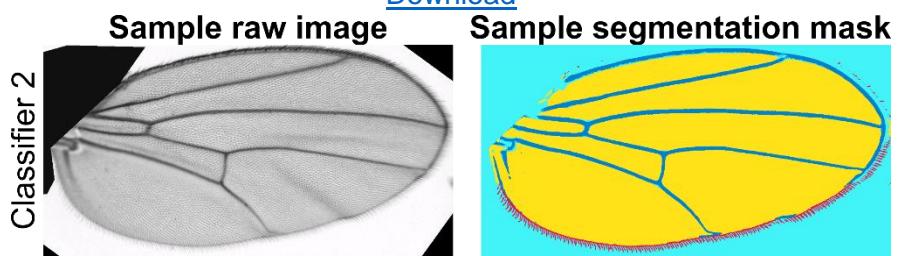
ILASTIK module 1 (One channel - Grayscale)

[Download](#)



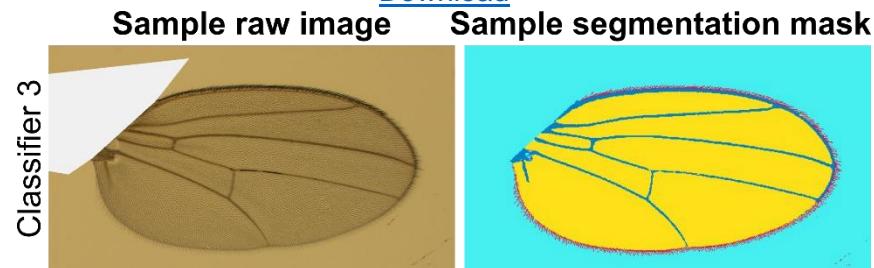
ILASTIK module 2 (One channel - Grayscale)

[Download](#)



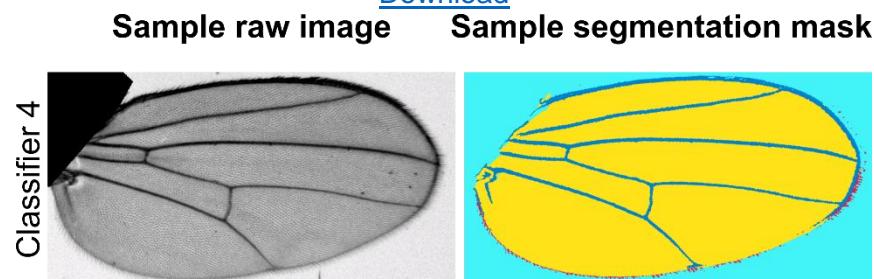
ILASTIK module 3 (Three channels - RGB)

[Download](#)



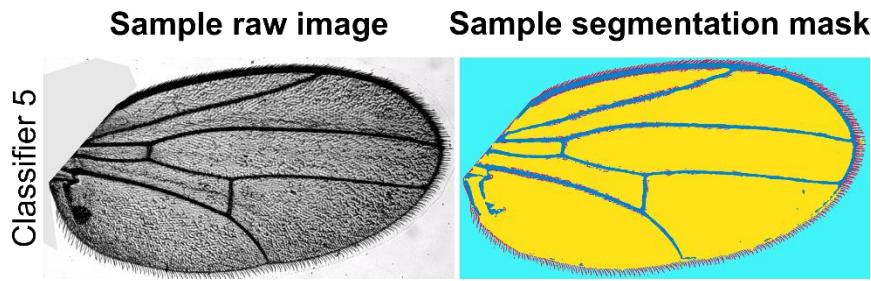
ILASTIK module 4 (One channel - Grayscale)

[Download](#)



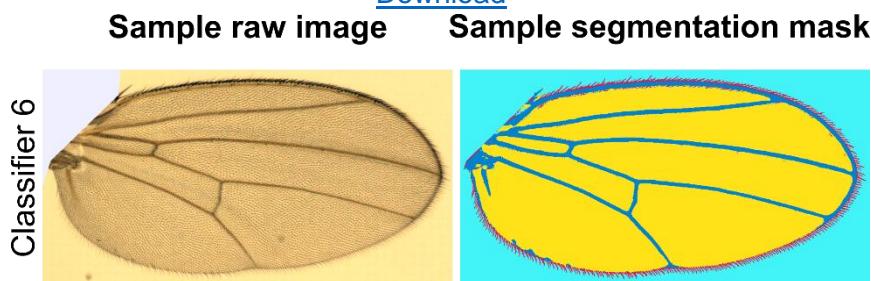
ILASTIK module 5 (One channel - Grayscale)

[Download](#)



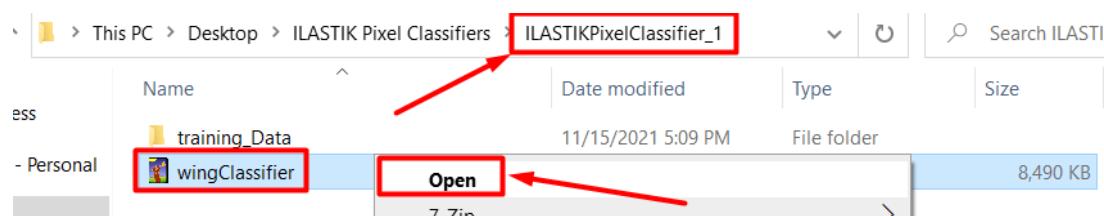
ILASTIK module 6 (Three channels - RGB)

[Download](#)

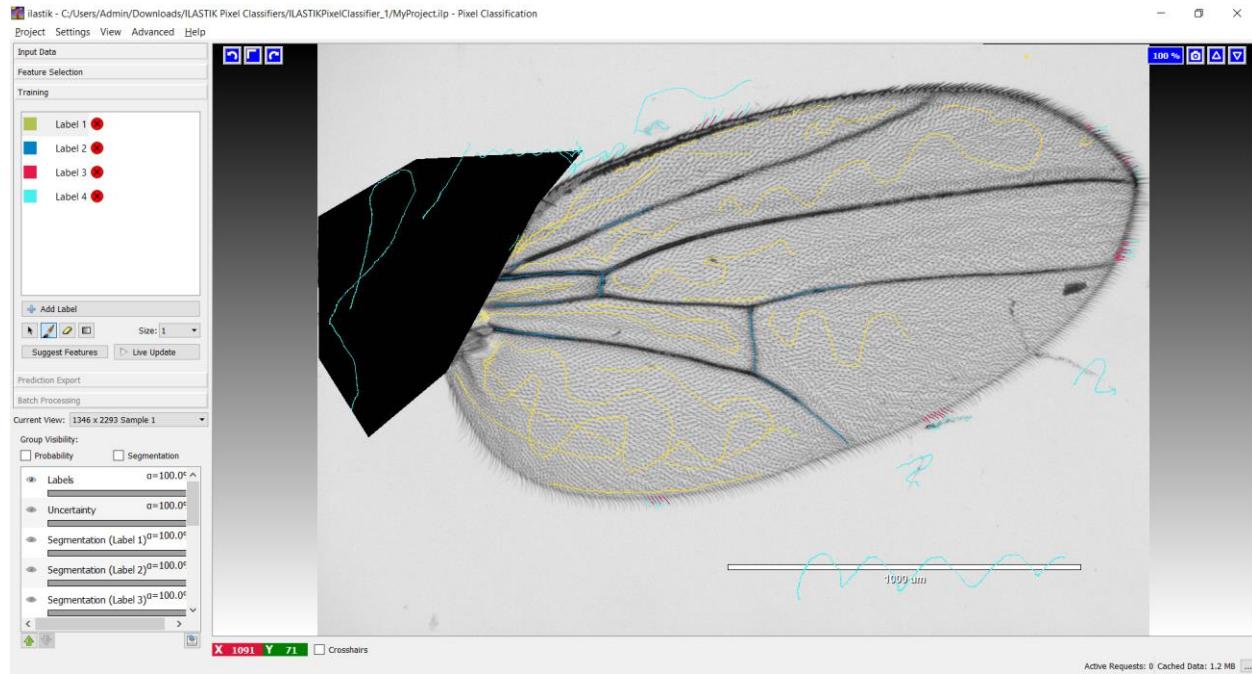


Processing images in ILASTIK

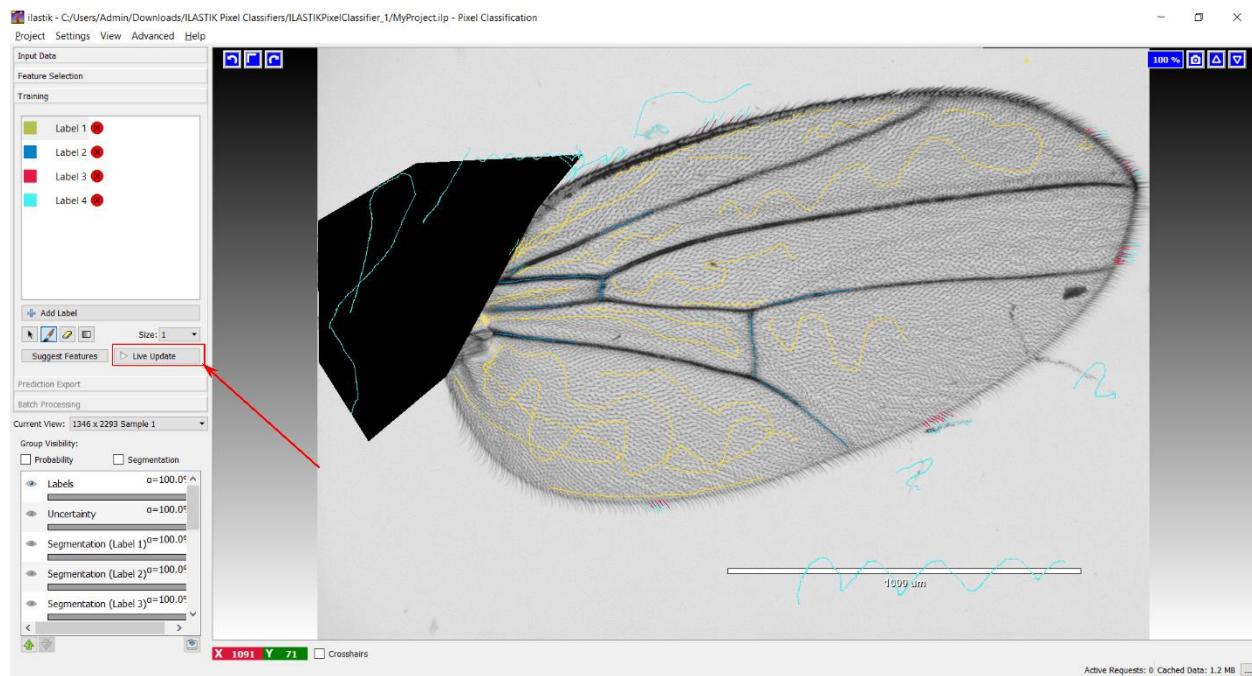
After selecting and downloading your module of choice, extract the .ZIP folder to an easily accessible location. The folder will contain 1) *Drosophila* wing images (with the wing hinge removed) that were used as training data to build the ILASTIK classifier in a “training_Data” folder and 2) a “wingClassifier.ilp” file that serves as the ILASTIK pixel classifier that has been pre-trained. Right-click the ILASTIK project file and select “Open” to start up ILASTIK. **NOTE:** ILASTIK start-up time may take several minutes.



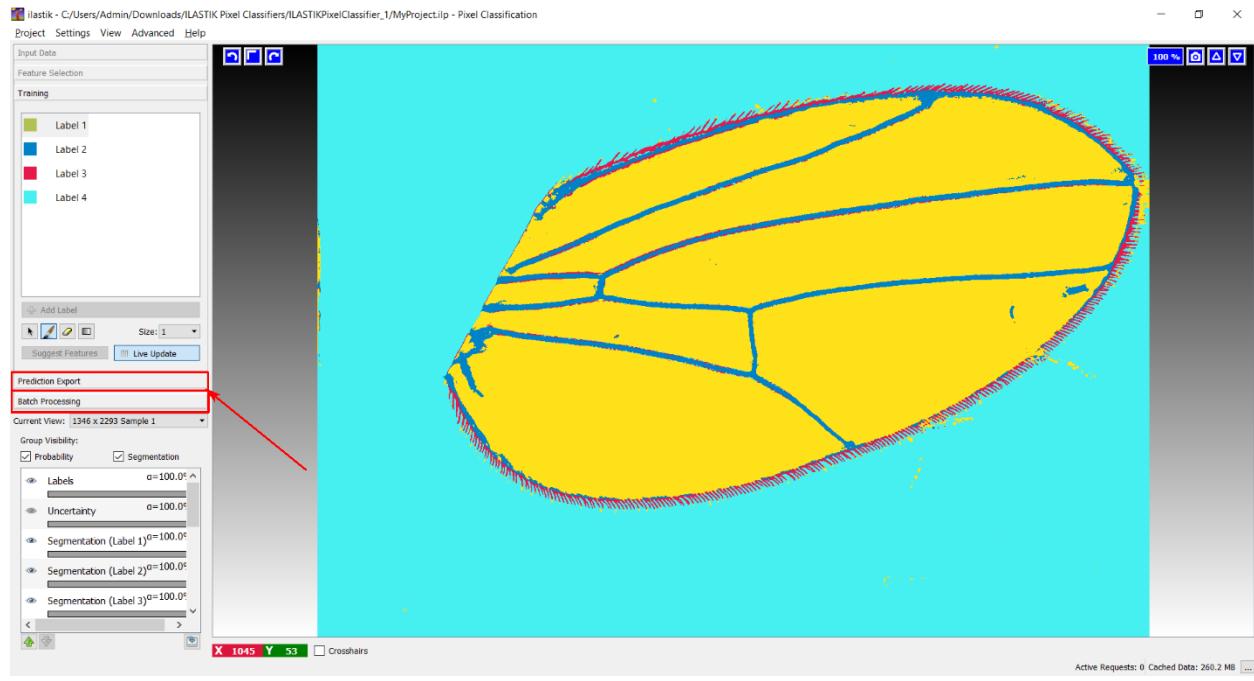
You will know ILASTIK start-up has completed when you can see a sample wing image marked with four different colors corresponding to different components of the image. The meanings of these colors are expanded upon in a later section for training your own ILASTIK module.



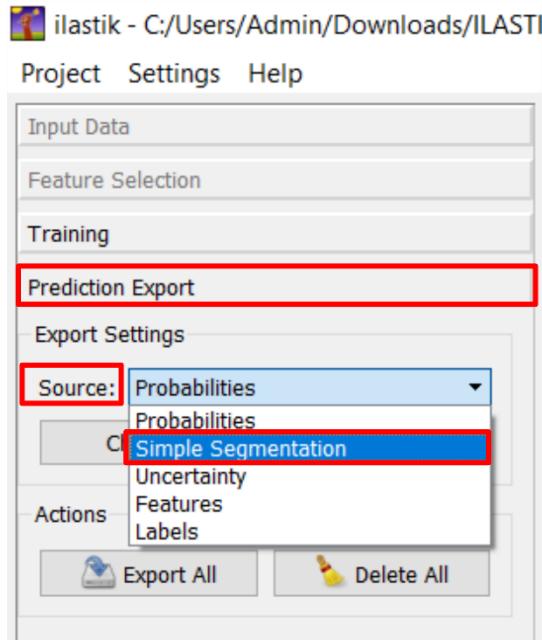
Proceed to click the “Live Update” button to view a segmented wing using the pre-trained classifier.



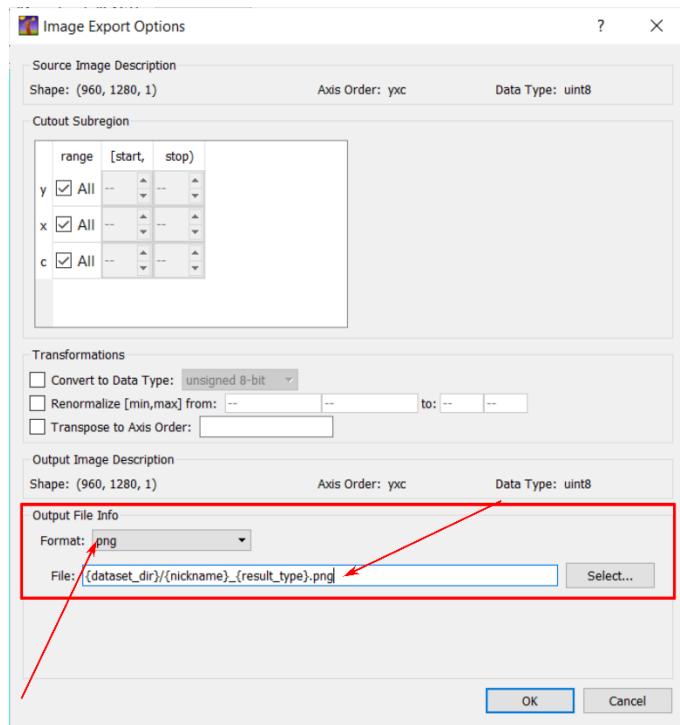
A progression bar can be found in the bottom-left corner of the page. Once the progression bar has reached 100%, ILASTIK will update the view of the image to be a color-coordinated segmentation mask of the pre-trained wing image. The “Batch Processing” and “Prediction Export” toolbars will also be available to click.



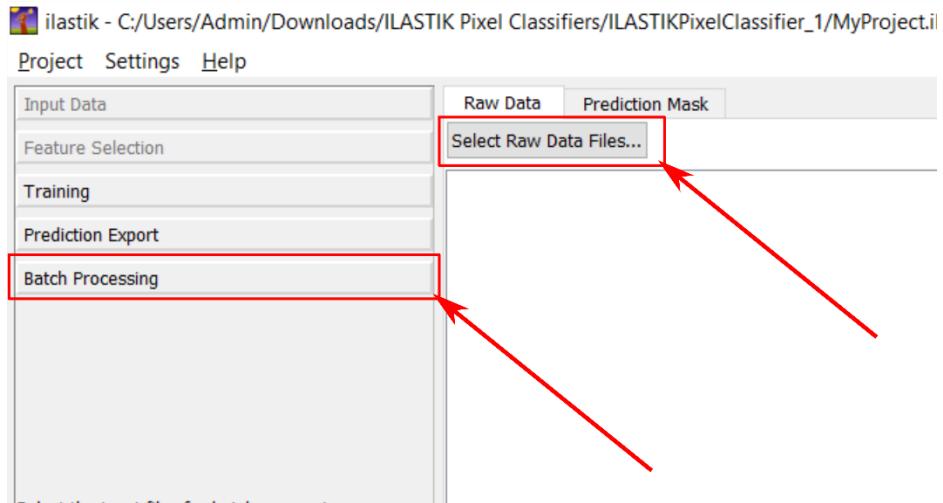
Proceed to click the “Prediction Export” toolbar to bring up the export settings of the output files from ILASTIK. In the “Source” dropdown menu, select “Simple Segmentation.”



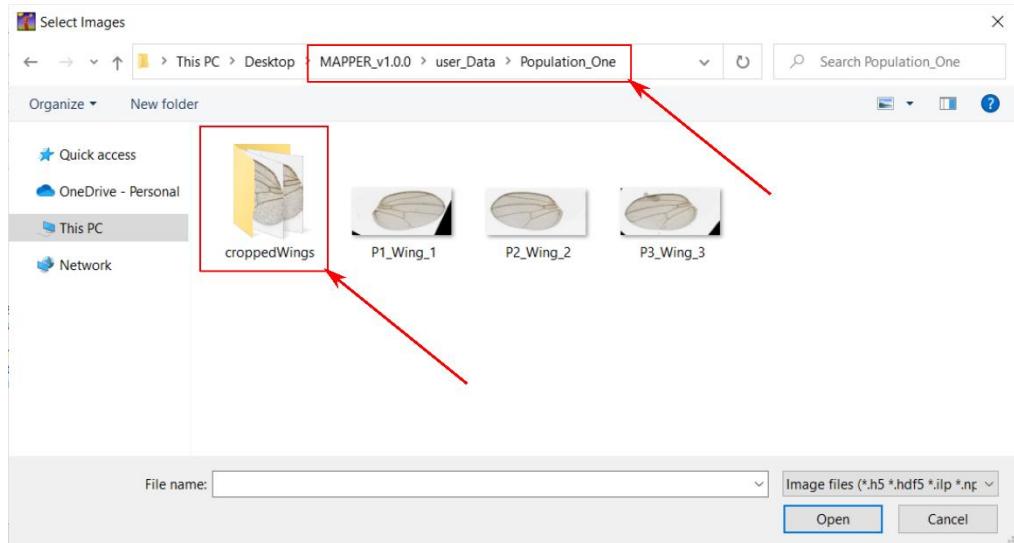
Below the dropdown box, click the “Choose Export Image Settings...” button. In the “Image Export Options” window, ensure that the “Output File Info” is formatted to be a “.png” file, with the file name as “{dataset_dir}/{nickname}_{result_type}.png”. Once this is done, close the export options by clicking “OK”. This will ensure that your output files are named and formatted correctly to be able to be processed by MAPPER.



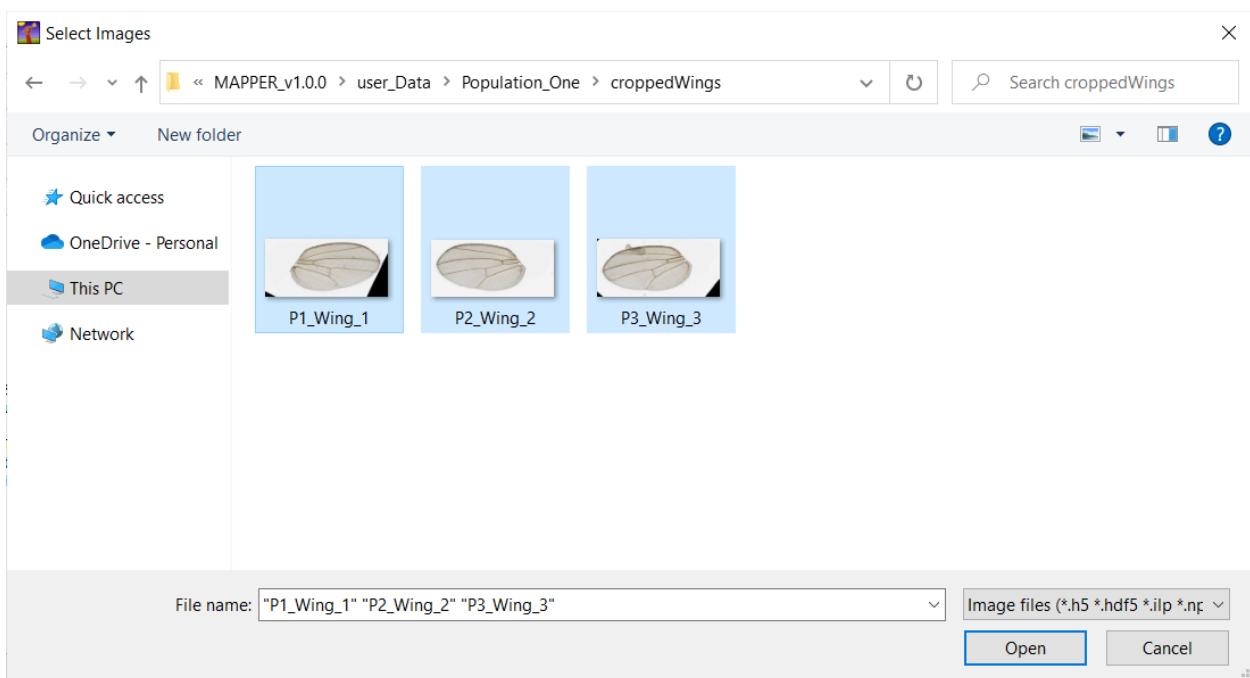
Next, you will need to click the “Batch Processing” toolbar on the left side of the available toolbars. Upon clicking the toolbar, you will need to click the “Select Raw Data Files...” button to select the images you would like to process with ILASTIK.



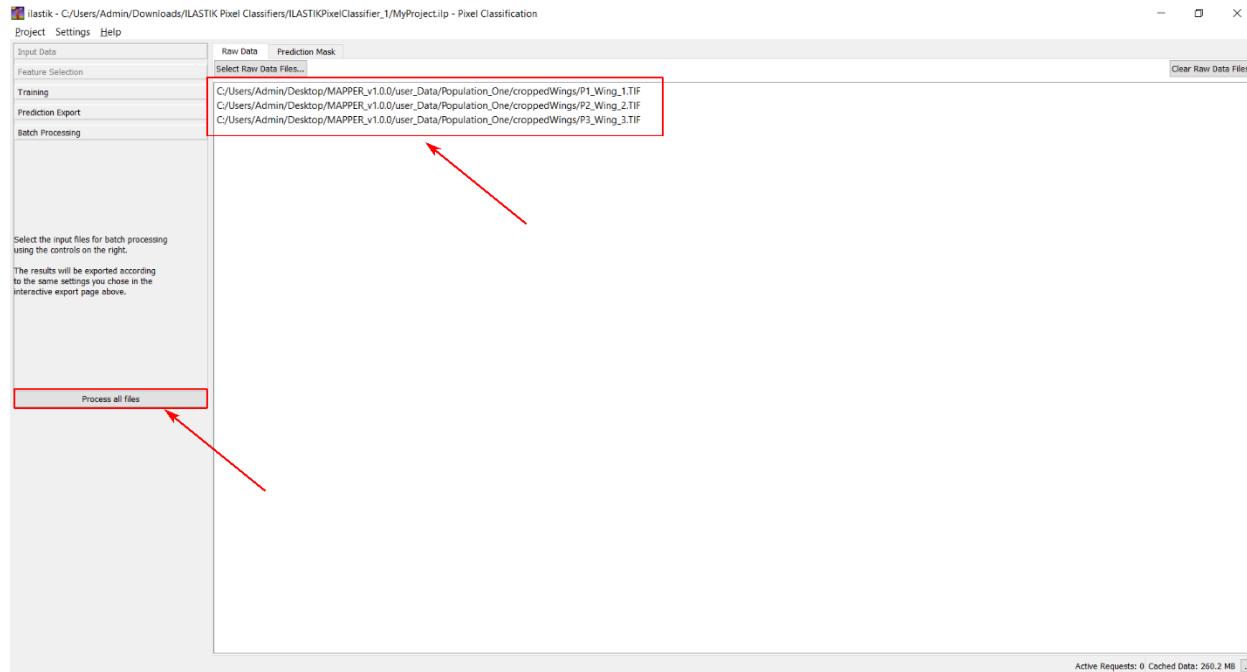
Upon clicking “Select Raw Data Files...”, navigate to the location of your “user_Data” folder containing the folders with your wing hinge removed data (e.g., “user_Data/Population_One/”) in the pop-up explorer box. Then, select the “croppedWings” folder that contains the images you processed with wing hinge removal step.



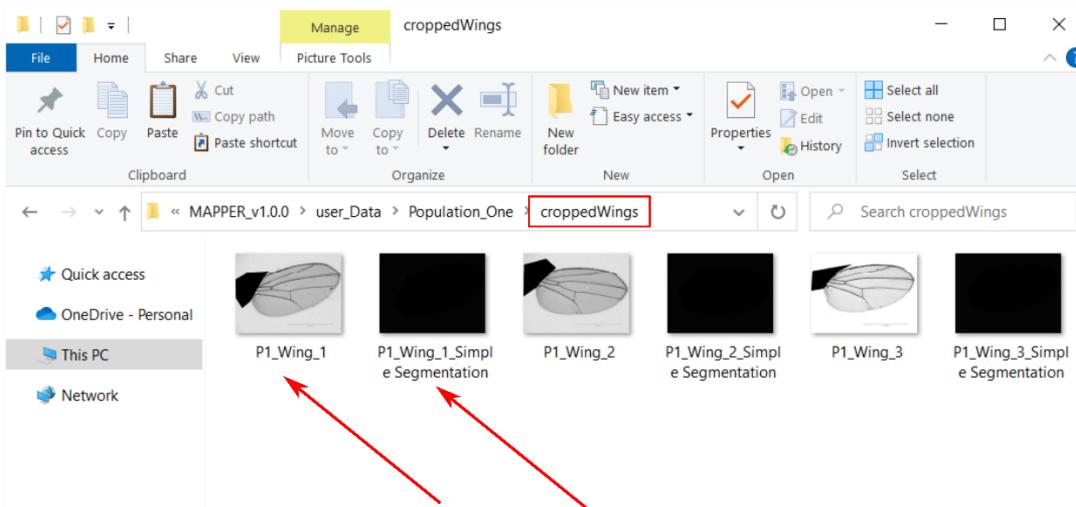
Within that folder, highlight and select all of the cropped wings and click "Open."



The file location and name of the images will then appear in the ILASTIK raw data files section. Once you confirm you have selected the images you would like to process, click the "Process all files" button.



After ILASTIK has processed the images, you should be able to observe new “.png” files in the “croppedWings” folder you selected your images from. These files will have the naming convention “**YOUR_IMAGE_Simple Segmentation.png**” where “**YOUR_IMAGE**” indicates the original name, and “**_Simple Segmentation**” indicates the type of output from ILASTIK. If the naming scheme of the output files does not contain “**_Simple Segmentation**” after your original image name, MAPPER **will not** be able to process your images. If this is the case, please review the “Prediction Export” settings detailed earlier.



If you are processing wings from multiple populations, repeat the process of clicking “Select Raw Data Files...”, navigating to the “croppedWings” folder of the next population, and selecting the images with the wing hinge removed for each of the populations. The end result will be such that

each of the “croppedWings” folder in your population folders will have 1) the cropped wing images and 2) the ILASTIK simple segmentation outputs.

Step Six: Running MAPPER

Understanding the input parameters

In section **II. Input parameters** of the GUI in the upper-right corner, you can modify parameters of MAPPER’s image processing pipeline to be custom-fit to your images and output preferences (*i.e.*, images from different magnifications, length-scales, and desired filtering thresholds). Each of the parameters has a tooltip available that provides a brief explanation of the parameter functionality that can be accessed by hovering the mouse over the parameter name. More specifically, the parameters have the following functionality:

Size of wing filter

This parameter value is passed into the “wingMorphFilter.m” dependency file, which is then utilized in the [“strel” function](#) in MATLAB®. The code uses this function to create a 2-D disk-shaped structuring element with a center pixel of the structuring element that has a specified radius (*i.e.*, the size of the wing filter) value. The radius value specifies the distance from the center pixel to the end of the structuring element. The returned structuring element consists of all pixels whose centers are no greater than the radius value away from the origin. This structuring element is then utilized for morphological erosion and dilation operations that aid in proper filtering of edge cases (e.g, where an intervein region and intervein transition) to label intervein regions correctly. The value of the size of the wing filter must be a nonnegative integer and has a default value of 8 for raw images taken at 4X magnification. A higher value should be used for images taken at greater magnification as there are more pixels contained in the raw data file.

Size of intervein filter

This parameter value is passed into the “interveinMorphFilter.m” dependency file, which is then utilized in the [“strel” function](#) in MATLAB®. The code uses this function to create a 2-D disk-shaped structuring element with a center pixel of the structuring element that has a specified radius (*i.e.*, the size of the intervein filter) value. The radius value specifies the distance from the center pixel to the end of the structuring element. The returned structuring element consists of all pixels whose centers are no greater than the radius value away from the origin. This structuring element is then utilized for morphological erosion and dilation operations that aid in proper filtering of edge cases (e.g, where an intervein region and intervein transition) to label veins correctly. The value of the size of the wing filter must be a nonnegative integer and has a default value of 5 for raw images taken at 4X magnification. A higher value should be used for images taken at greater magnification as there are more pixels contained in the raw data file.

Threshold distance trichome

This parameter value is passed into the “statsAreaTrichome.m” dependency file and the “trichome.m” dependency file. A distance (in pixels) between two potential trichomes is calculated using the distance formula and compared to the input parameter threshold value. If the calculated distance is less than or equal to the input threshold, a trichome is removed. In other words, the parameter value represents the minimum tolerable distance between two trichomes in order for both trichomes to be validated and counted. This is to ensure that trichomes are not double

counted. This parameter must be a nonnegative integer and has a default value of 2 for raw images taken at 4X magnification. A higher value should be used for images taken at greater magnification as there are more pixels contained in the raw data file.

Number of harmonics

This parameter value is used in calculation of the Elliptic Fourier Descriptors (EFDs) to approximate the shape of a closed contour (i.e., the different labeled compartments of the wing). The Fourier series is described by the following equation:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^k [a_n \cos(nx) + b_n \sin(nx)]$$

for $n = 1$ to k harmonics. The higher the value of the parameter for number of harmonics (i.e., value of k), the better the approximation of the shape will be to the closed contour. We have found that the default value of 15 for this parameter closely captures the contour shapes of the wing compartments without excessive computational demand. A higher or lower value can be used at the user's discretion. More information about the number of harmonics can be found in the manuscript.

Conversion factor

This parameter value is the length scale conversion factor of microns per pixel of your raw image. This can be found manually by measuring the length (in pixels) of an overlaid scale bar of known length in microns on your raw images. You must then divide the micron length of the scale bar by the length of the measured pixels.

Output spreadsheet save name

Changing the text input for this parameter allows you to name the batch processing output ".csv" and ".mat" files. The output files are located in the "batch_output_files" subfolder of the main MAPPER folder. The naming convention of the files will be "TEXT_EFD_Output" for the Elliptic Fourier Descriptor files and "TEXT_Wing_Measurements_Output" for the automated wing measurement data where "TEXT" is the input parameter you specify.

Individual wing processing

In section **III. Individual wing analysis** of the GUI in the left-hand side, you will see output images and buttons associated with processing a single image in MAPPER. This is to ensure quality control of your input images if batch processing results in errors. Each of the sections detail the following:

Loaded raw image

Displayed in this figure is the image of the *Drosophila* wing selected that has had the wing hinge removed.

Labeled output image

Displayed in this figure is the color-coded labeled intervein segmentation of your selected wing that has been process by ILASTIK and MAPPER.

Biological axes display

Displayed in this figure is the biological axes measurement of your choice overlaid in a red-dashed line on the cropped wing image.

User operation

These buttons enable the user to use MAPPER to process a single image. More information on how to use these buttons is provided in a proceeding section.

Biological axes

These buttons enable the user to choose which of the biological axis measurement is displayed and overlaid on the “Biological axes display” figure. Choices range from the anterior-posterior (AP) axis, the proximal-distal (PD) axis, or the distance between the 3rd and 4th longitudinal veins (d(L3,L4)).

Individual intervein output

This table contains the output wing area measurements (mm^2) and total trichome counts for each intervein region identified by MAPPER. Intervein regions are numbered by their ID values.

Global statistics

This section displays the raw numeric value of the total wing area (mm^2) and total trichome density (count per mm^2) for the entire wing.

Biological axes length

This section displays the raw numeric value of the distance (mm) of the chosen biological axis.

Enable unique output label checkbox

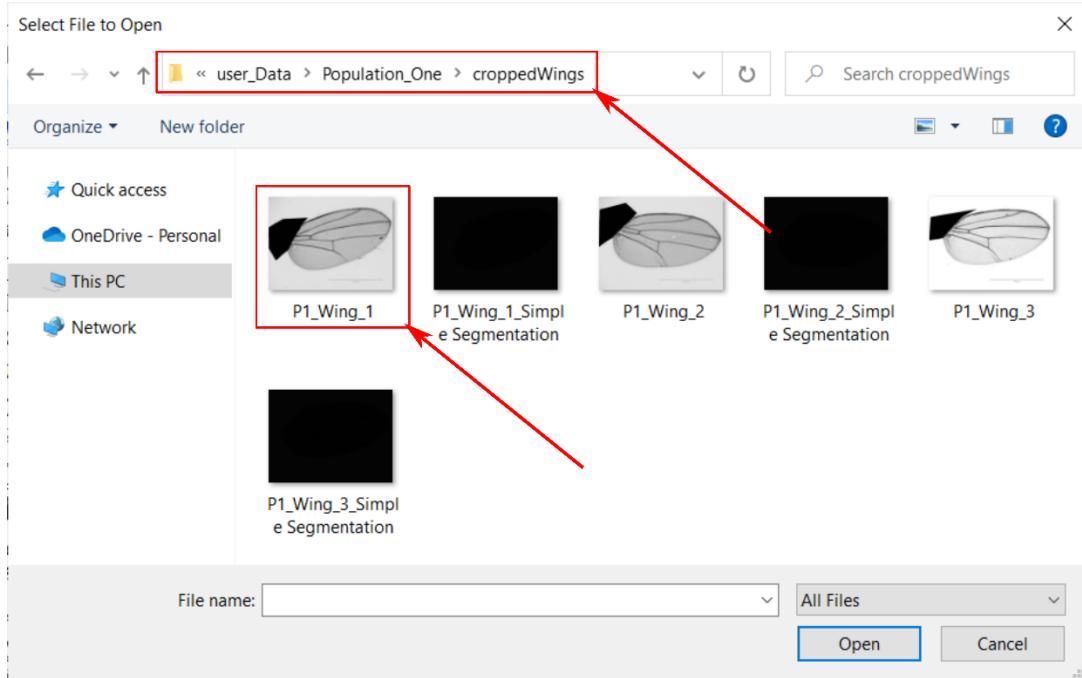
This checkbox when filled, will save a unique color-coded segmented wing of your chosen image that was processed. The saved image can be found as a “.png” file in the “individual_output_files” subfolder with the same save name as the original selected image. When this checkbox is unfilled, the color-coded label is saved as “imageOutput.png” instead of having a unique name. This ensures that the folder is not filled with too many files upon handling individual processing and will only save unique files upon request.

Running an example individual wing

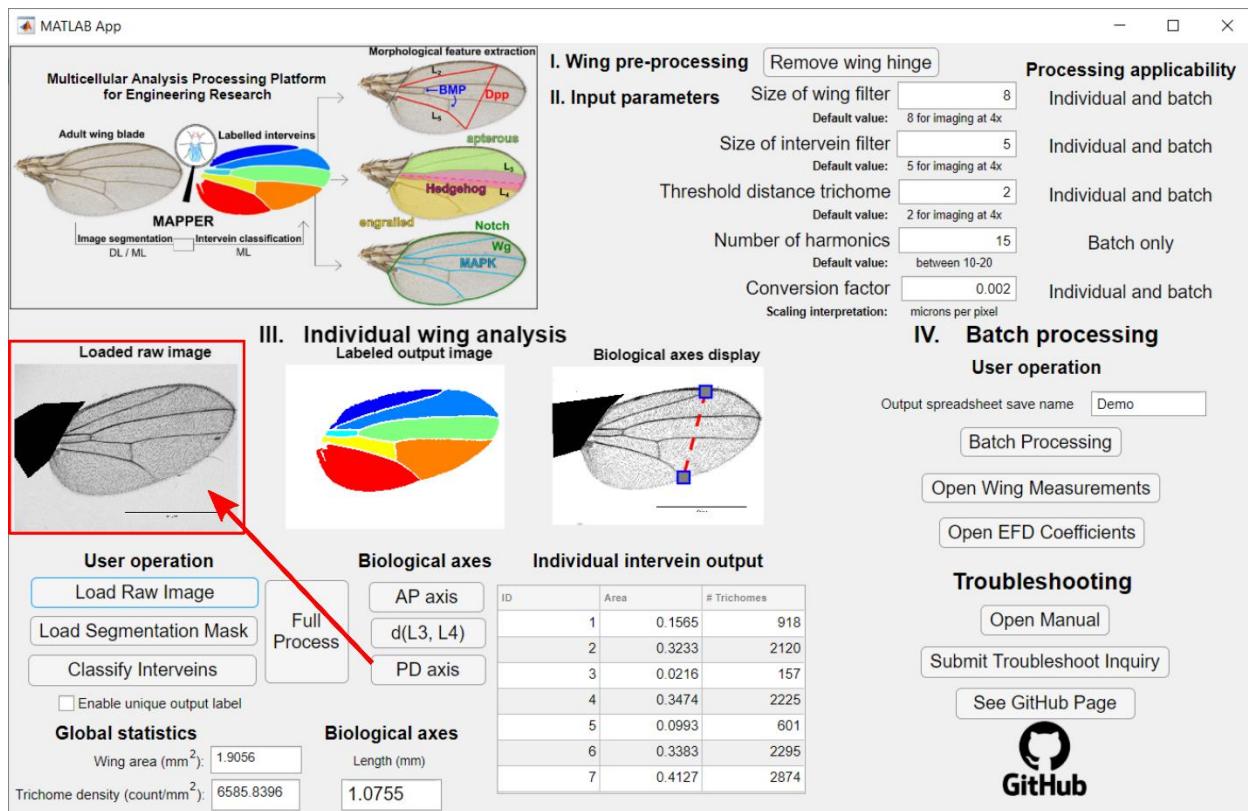
In order to process an individual wing in MAPPER, you will need to use the “User operation” section buttons of the GUI. Individual processing can be handled sequentially for troubleshooting purposes, or all at once.

Sequential processing

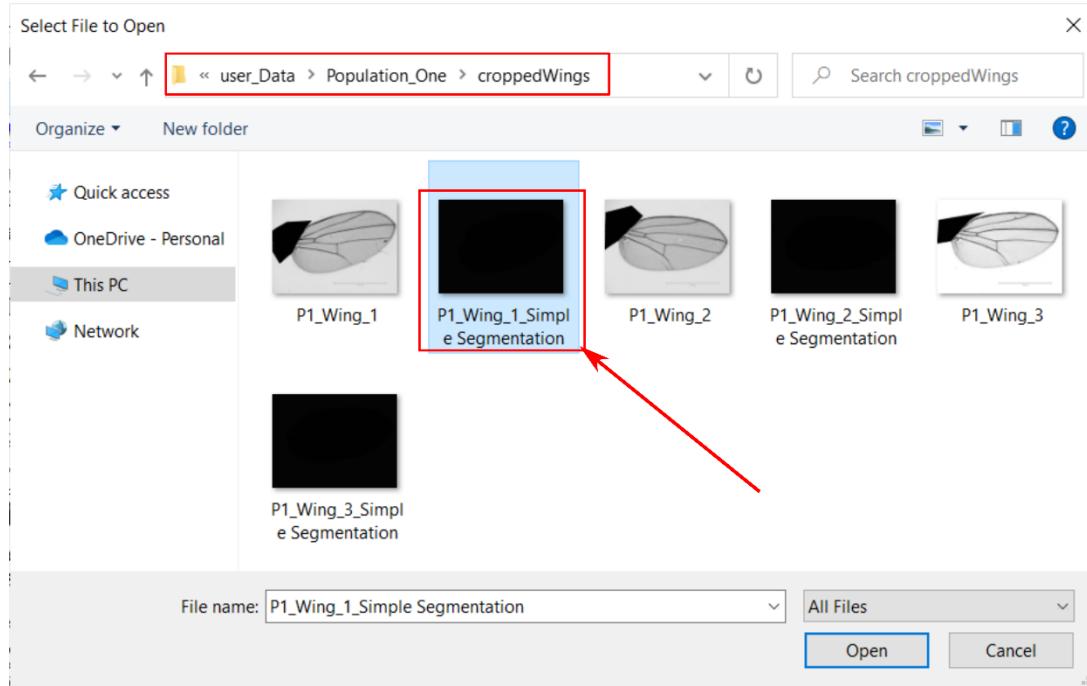
First, you will need to click the “Load Raw Image” button. This will bring up a prompt for you to navigate to the wing hinge removed image you would like to process. For example, if your first batch of data is within a folder named “Population_One” in the “user_Data” folder, you will navigate to “user_Data/Population_One/croppedWings” and select the cropped wing image you would like to process.



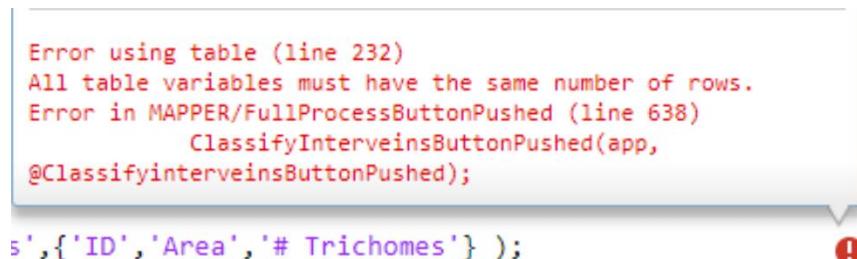
After completing this step, you will see your selected wing in the “Loaded raw image” figure of section III. Individual wing analysis.



Next, you will click the “Load Segmentation Mask” button. This will bring up a prompt for you to navigate to the ILASTIK output file labeled “_Simple Segmentation.png” for the wing you selected. For example, you will navigate to the same subfolder “user_Data/Population_One/croppedWings” and select “IMAGE_NAME_Simple Segmentation.png”.



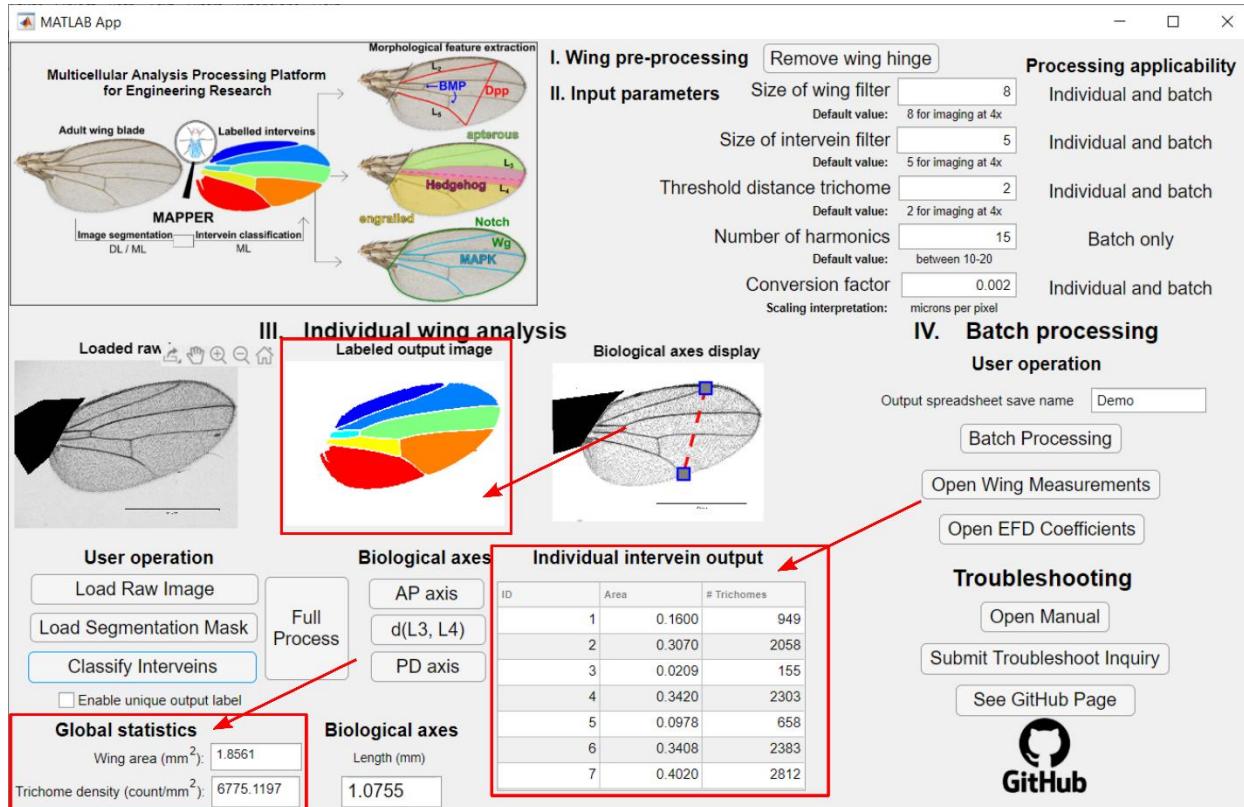
If there is an issue or error warning upon trying to complete this step, there is likely an error in how your image was processing in ILASTIK. This could be due to your images being incompatible with the training data of your selected ILASTIK module (*i.e.*, you need a different ILASTIK module), or the export settings of ILASTIK were not specified as instructed. A common error for this step results from the output simple segmentation file from ILASTIK not having been segmented correctly. This results in the following error in MAPPER:



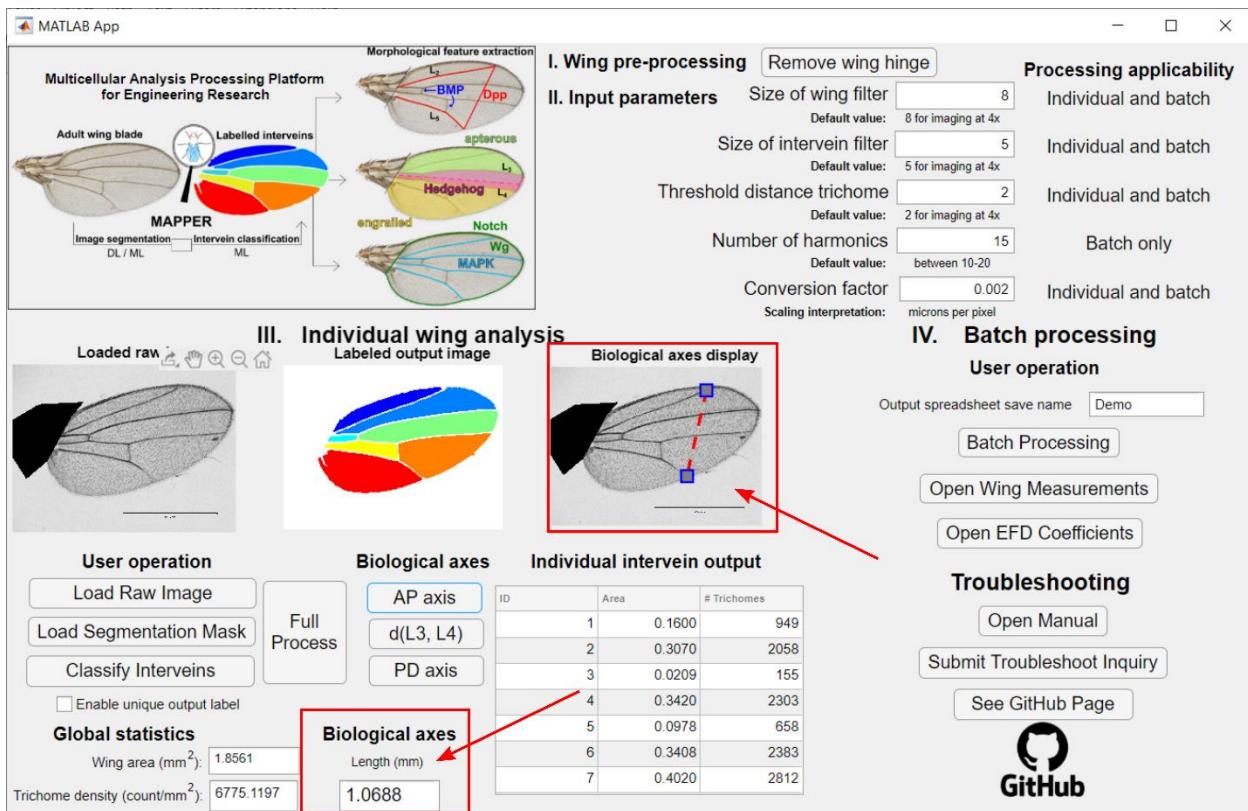
To fix this issue, you will need to try a different module, or train your own using the instructions in a subsequent section of this manual.

Next, you will click the “Classify Interveins” button. This will update the “Labeled output image” figure to be a color-coded segmentation of your input wing image. Additionally, the output numeric

values of the “Individual intervein output” and “Global statistics” sections will be updated and display the measurements for your selected wing. Wing area measurements are in units of mm² and trichome density is in units of trichome count per mm². If there are issues or error warning upon completion of this step, the image you are trying to process was not compatible with the training data of your ILASTIK module. An example of a potential error is no color-coded labeled wing appearing in the figure, or a partial wing being shown. If this is the case, you should try to use another ILASTIK module to process your wings.

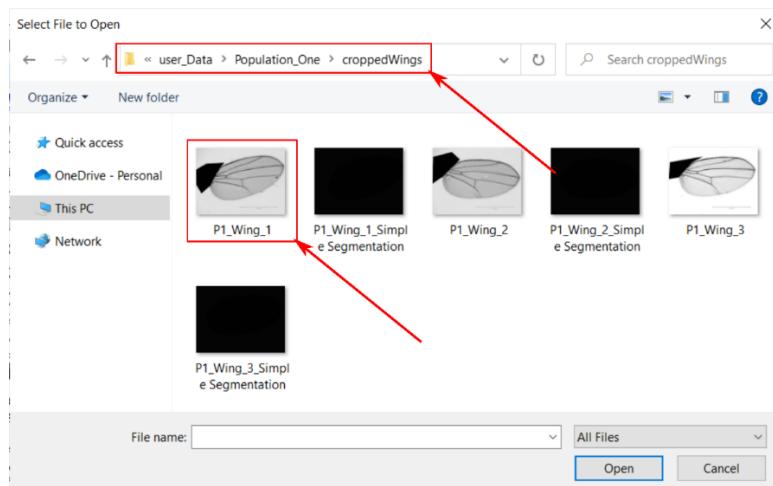


After this step, you can click the “Biological axes” buttons to have MAPPER process and calculate the biological axis measurement of your choosing. Choices range from the anterior-posterior (AP) axis, the proximal-distal (PD) axis, or the distance between the 3rd and 4th longitudinal veins (d(L3,L4)). After clicking one of the three buttons, the corresponding measurement will be displayed in the “Biological axes” section by Global statistics, and a visual representation of the measurement will be displayed in the “Biological axes display” figure.



Single step processing of an individual image

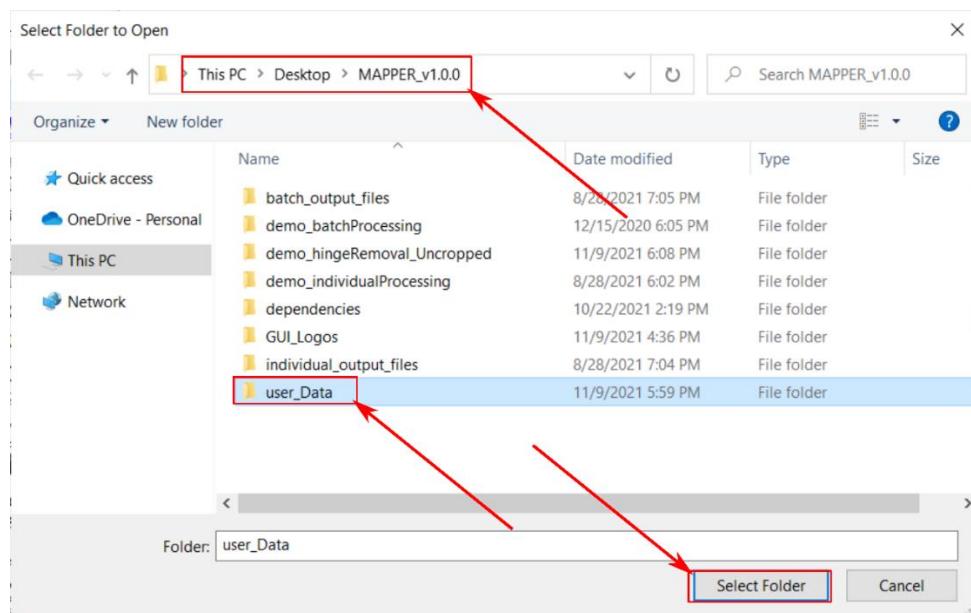
The sequential processing step detailed previously should be used to troubleshoot if errors occur using the “Full Process” button. By clicking the “Full Process” button, MAPPER will bring up a prompt for you to navigate to the wing hinge removed image you would like to process. For example, if your first batch of data is within a folder named “Population_One” in the “user_Data” folder, you will navigate to “user_Data/Population_One/croppedWings” and select the cropped wing image you would like to process. With the ILASTIK export settings completed as specified in previous steps, MAPPER will automatically locate and identify the correct segmentation mask, classify the interveins for you, and automatically output the AP axis length measurement.



Batch processing

In order to process all images within your “user_Data” folder and obtain spreadsheets with the output measurements, you will need to utilize section **IV. Batch processing** of the GUI. First, you will need to modify the input text of the “Output spreadsheet save name” parameter to be what you would like your spreadsheets to be named. The default parameter input name is “Demo.” **NOTE:** If you batch process with the same input names across runs, the previous files will be overwritten; please ensure you input unique names for each run to properly save your output files.

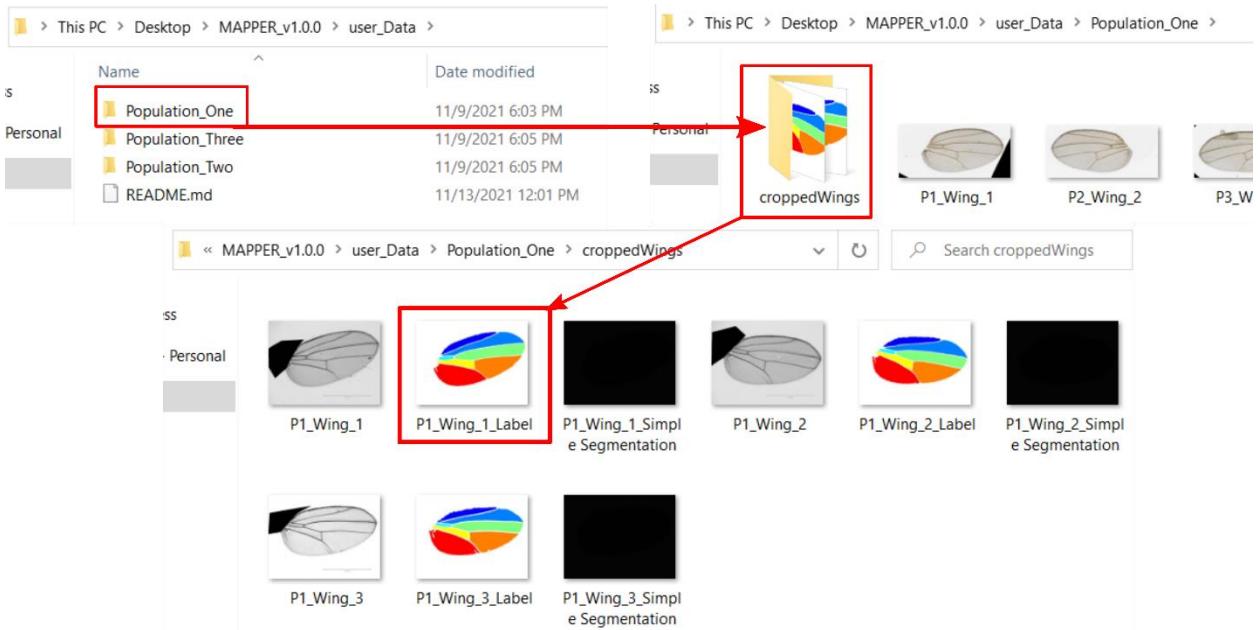
You will then proceed to click the “Batch Processing” button. This will bring up a prompt for you to navigate to your “user_Data” subfolder within the MAPPER main folder. You will need to click the “user_Data” folder, then click “Select Folder.”



After this, MAPPER will begin to process each of the images within the “croppedWings” folders of all the “Population” folders. You will know when batch processing is complete when the MATLAB® console outputs the “Batch processing complete” message.

```
Processing image file C:\Users\Admin\Desktop\MAPPER_v1.0.0\user_Data\Population_Two\croppedWings\P2_Wing_2.tif
B =
1x1 cell array
{2649x2 double}
Processing image file C:\Users\Admin\Desktop\MAPPER_v1.0.0\user_Data\Population_Two\croppedWings\P2_Wing_3.tif
%%%%%%%%%%%%%
Batch processing complete!
fx >> |
```

Once batch processing is complete, within each population folder and the corresponding “croppedWings” folder, you will find output segmented wing labels of all of your batch processed wings.



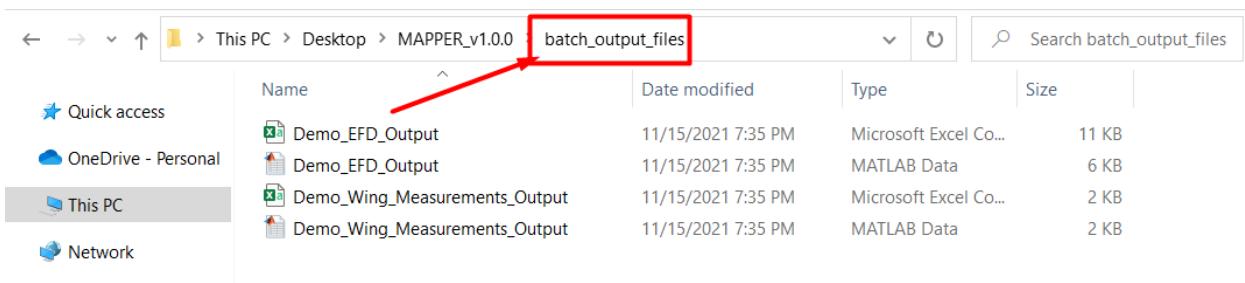
If you encounter any errors during batch processing, you can identify the last remaining image that was attempted to be processed in the MATLAB® console. Once you have identified the name of the image, you can troubleshoot by individually processing the wing with the error.

```
B =
 1x1 cell array
    /2649x2 double
Processing image file C:\Users\Admin\Desktop\MAPPER_v1.0.0\user_Data\Population_Two\croppedWings\P2_Wing_3.tif
=====
Batch processing complete!
fx >>
```

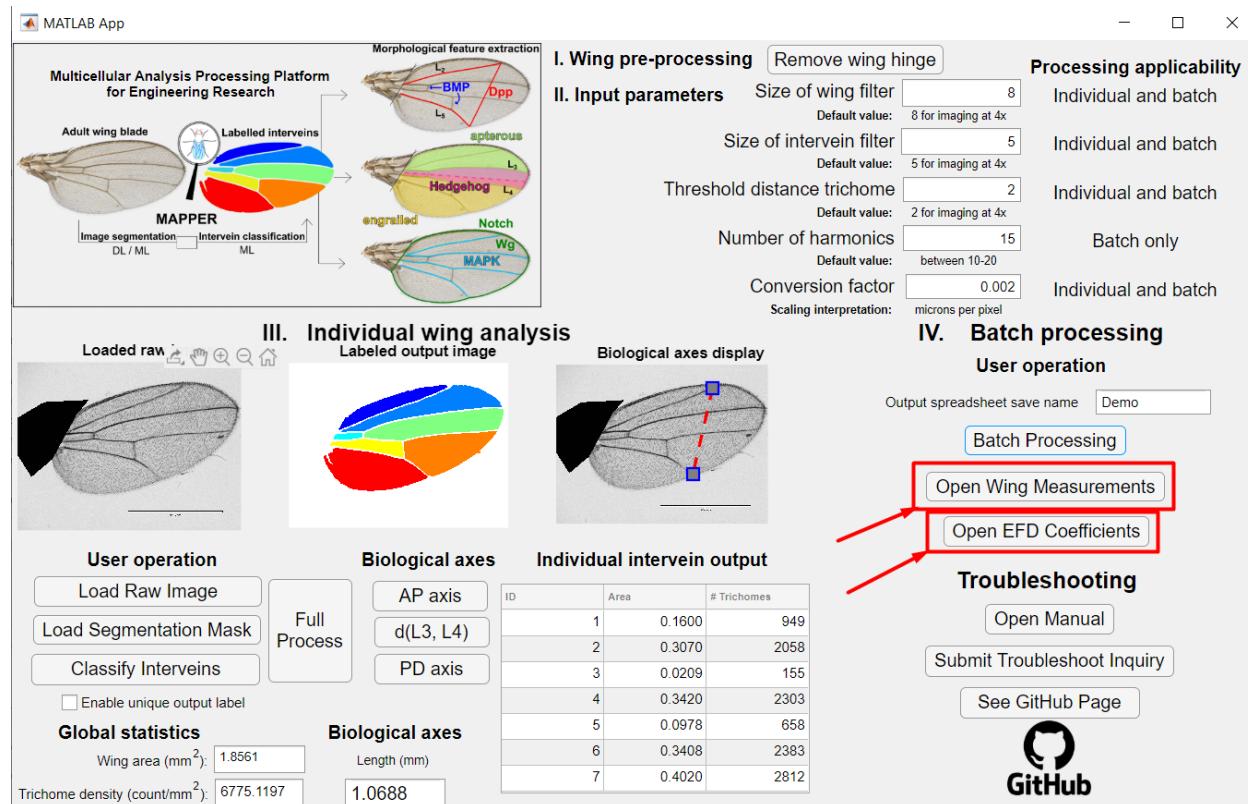
Step Seven: Accessing and understanding your output data

Location of output files from batch processing

Once batch processing is complete, you can access the output data (wing feature measurements or EFD coefficients) as either “.csv” files or “.mat” files. The files are located in the “batch_output_files” subfolder in the main MAPPER folder with the save name you specified in the previous step.



The wing measurements are within the files appended with “_Wing_Measurements_Output” and the EFD coefficients are within the files appended with “_EFD_Output.” Alternatively, you can click the “Open Wing Measurements” or “Open EFD Coefficients” buttons on the MAPPER GUI to prompt Microsoft Excel to open the corresponding spreadsheet.



Understanding the wing feature measurements output

When opening the “.csv” file for the wing measurements, you will have a spreadsheet with columns A – AA filled with the following contents:

Column A: Wing name

This is the name of the file that was processed.

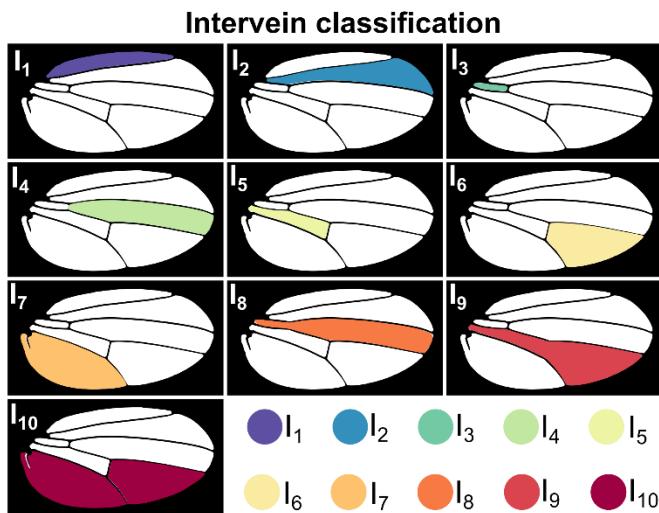
Column B: Genotype

This is an identifier for separating which of the populations the file belongs to. Because each user will have unique populations that are batch processed, the identifiers are numeric values (*i.e.*, the first folder processed will have a genotype of 1, the second folder processed will have a genotype of 2, *etc.*) The order of processing of the folders by MAPPER is by alphabetic order.

Columns C–L: Intervein areas 1–10

These columns contain the intervein area measurements (mm^2) for each individual intervein compartment. A standard wing with no vein defects will only have measurements corresponding to intervein regions 1–7 with 8–10 being empty. If vein defects are present in the wing, or the

wing was improperly segmented by ILASTIK, intervein regions 8—10 may have measured values. See the below image for identification of intervein regions and their corresponding compartmentalization on a wing. Having a measured value for intervein region 8 is indicative of the processed image having a missing or incomplete anterior cross-vein (ACV). Having a measured value for intervein region 9 is indicative of the processed image having a missing or incomplete posterior cross-vein (PCV). Having a measured value for intervein region 10 is indicative of the processed image having a missing or incomplete 5th longitudinal vein. **NOTE:** If your image does indeed have a complete 5th longitudinal vein and you are obtaining measurements for the 10th intervein region, your ILASTIK pixel classifier is not correctly identifying the intervein extending to the edge of the periphery and will need additional training. Information on training an ILASTIK module is found in the subsequent section.



Columns M—V: Intervein trichome counts 1—10

These columns contain the intervein trichome counts for each individual intervein compartment. Similar to the previous measurements, a standard wing with no vein defects will only have measurements corresponding to intervein regions 1—7 with 8—10 being empty. If vein defects are present in the wing, or the wing was improperly segmented by ILASTIK, intervein regions 8—10 may have measured values.

Column W: Total wing area

This column contains the total wing area of the processed wings in mm².

Column X: Total trichome density

This column contains the total wing trichome density of the processed wings in trichome count per mm².

Column Y: AP axis

This column contains the length measurement of the AP axis in mm.

Column Z: PD axis

This column contains the length measurement of the PD axis in mm.

Column AA: d(L3-L4)

This column contains the length measurement of the distance between the 3rd and 4th longitudinal veins.

Understanding the EFD coefficients output

When opening the “.csv” file for the EFD coefficients, you will have a spreadsheet with a variable number of columns depending on the chosen parameter input value for the “Number of harmonics” parameter in section II. Input parameters of the GUI. The first column (Wing name, A) of the spreadsheet will contain the name of the file that was processed. The second column (Genotype, B) of the spreadsheet is an identifier for separating which of the populations the file belongs to. Because each user will have unique populations that are batch processed, the identifiers are numeric values (*i.e.*, the first folder processed will have a genotype of 1, the second folder processed will have a genotype of 2, etc.) The order of processing of the folders by MAPPER is by alphabetic order

Each subsequent column will contain the EFD coefficients for the image processed in multiples of four depending on your parameter input for the number of harmonics. Having the default 15 harmonics will result in 60 total EFD coefficients. Because the contour being described is 2-D, two separate Fourier series are used for the coefficients:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^k [a_n \cos(nx) + b_n \sin(nx)]$$
$$f(y) = \frac{1}{2}c_0 + \sum_{n=1}^k [c_n \cos(ny) + d_n \sin(ny)]$$

EFDs are calculated with a_0 and c_0 being zero values. Thus, coefficient 1 in Column C of the spreadsheet corresponds to coefficient a_1 , coefficient 2 in Column D of the spreadsheet corresponds to coefficient b_1 , coefficient 3 in Column E of the spreadsheet corresponds to coefficient c_1 , and coefficient 4 in Column F of the spreadsheet corresponds to coefficient d_1 . This pattern is repeated after this such that coefficient 5 is coefficient a_2 etc. Each coefficient provides information about the shape of the wing and individual intervein regions and can be used to reconstruct a contour of the wing if desired.

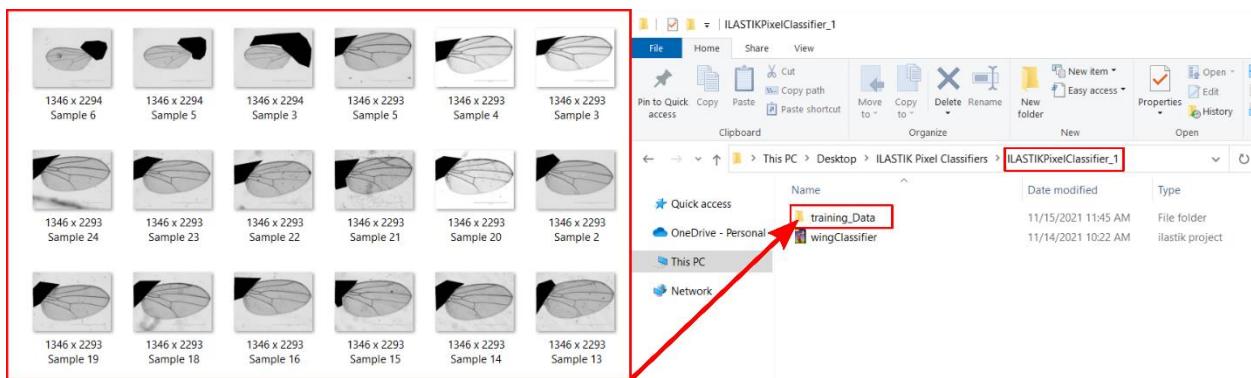
(Optional) Train your own ILASTIK module

General advice for ILASTIK

In order to train your own ILASTIK module, you will need training data of the images you would like to process. The more training data you can provide, the more refined the resulting segmentation will be. However, do keep in mind diminishing returns on training too many images that will unnecessarily increase computational time required to run ILASTIK. We recommend at least five to ten training images for a given microscope imaging setup. Additionally, to prevent having to train multiple ILASTIK classifiers, we recommend taking images at the same lighting, magnification, gain, and exposure settings on your microscope to run MAPPER more easily in future data processing. The most important aspect of the training data is to thoroughly train near the edge cases (*i.e.*, where the intervein region transitions into a vein region or into the non-wing background). Failing to properly train edge cases will result in incomplete segmentation. To refine the edge cases when labeling (detailed in the subsequent sections), you can zoom in and out of the viewable image by holding “Ctrl” and using the mouse to scroll up (zoom in) or scroll down (zoom out). We also recommend being cautious when using training data that contains large obscurities in the wing or underneath (*e.g.*, a bubble from the mounting media underneath the wing, a torn wing, a folded over wing, or dissected tissue stuck under the wing). These obscurities will provide unnecessary confusion in the training data as these are likely anomalies from all the images you would like to process. Small obscurities are sorted out in MAPPER; however, large obscurities make it difficult for MAPPER and ILASTIK to segment wings.

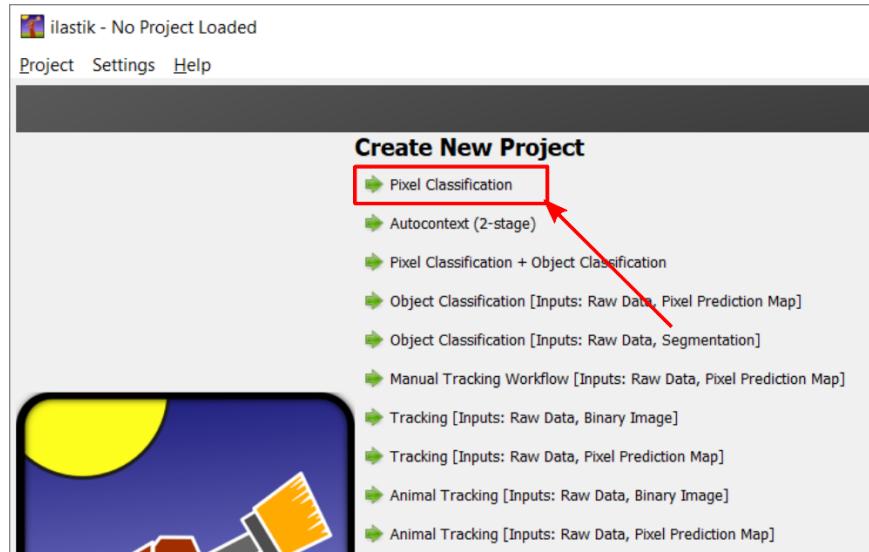
Obtaining the correct folder structure

There is no absolute folder structure necessary to train an ILASTIK classifier, however, for standardized use of the MAPPER tool for the *Drosophila* community, we recommend having one main folder (*e.g.*, “ILASTIKPixelClassifier”) and one subfolder (*e.g.*, “training_Data”). In the training data subfolder, you are to place the .TIF images of sample training data of your *Drosophila* wing images with the wing hinge removed.

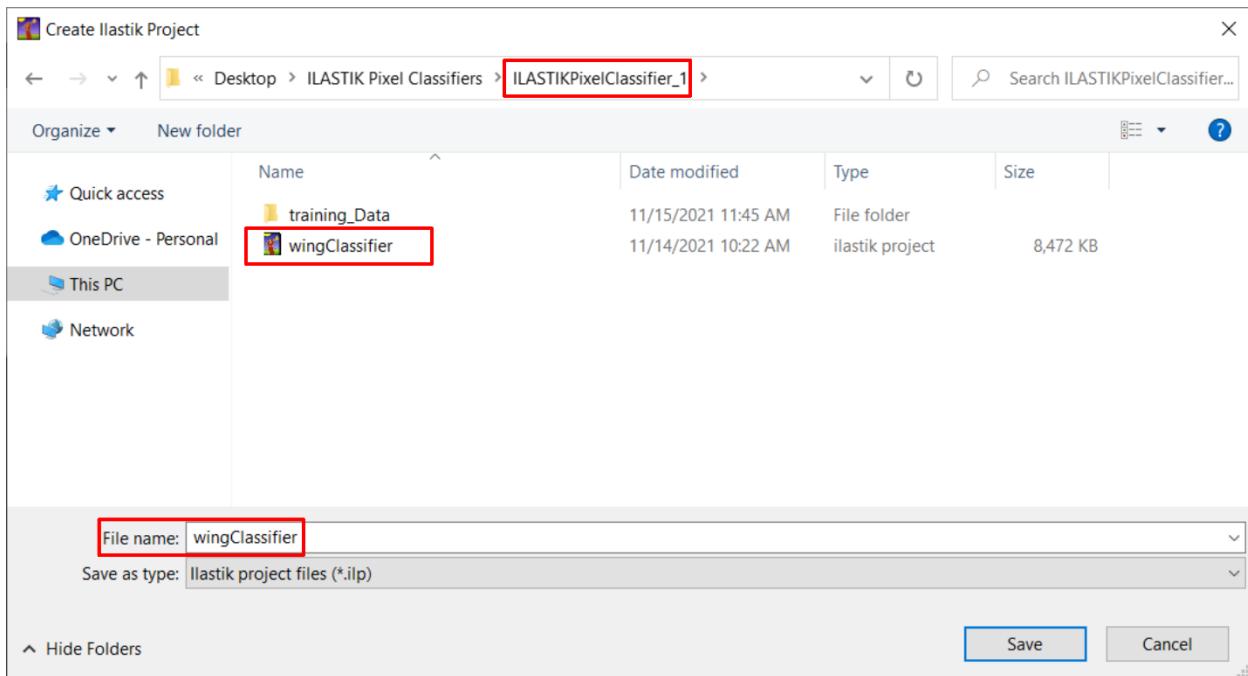


Starting a new ILASTIK project

Open the ILASTIK software and wait for the initial screen to open. Once ILASTIK is opened, click “Create New Project” > “Pixel Classification”.

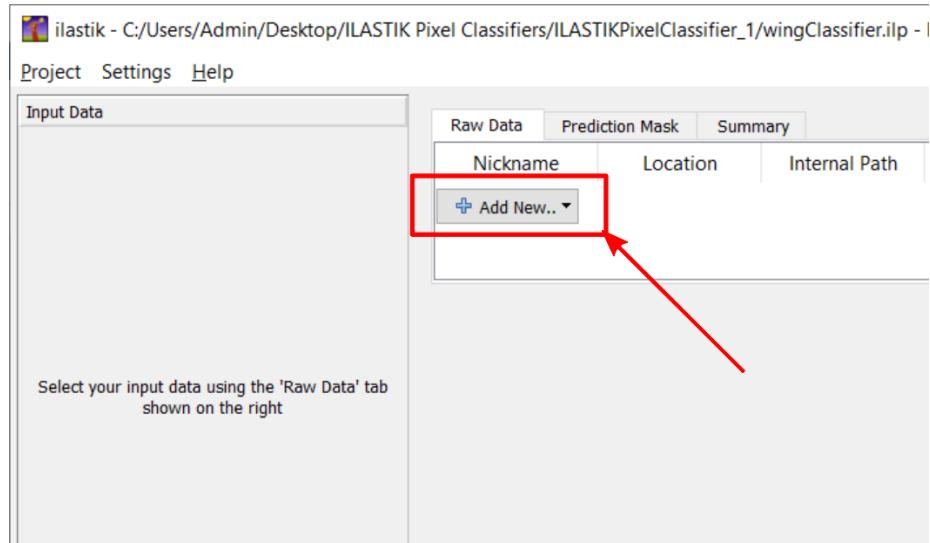


This will prompt you to name the new project and specify the save location. Navigate to the location of the main folder and save the new project as “wingClassifier.ilp”.

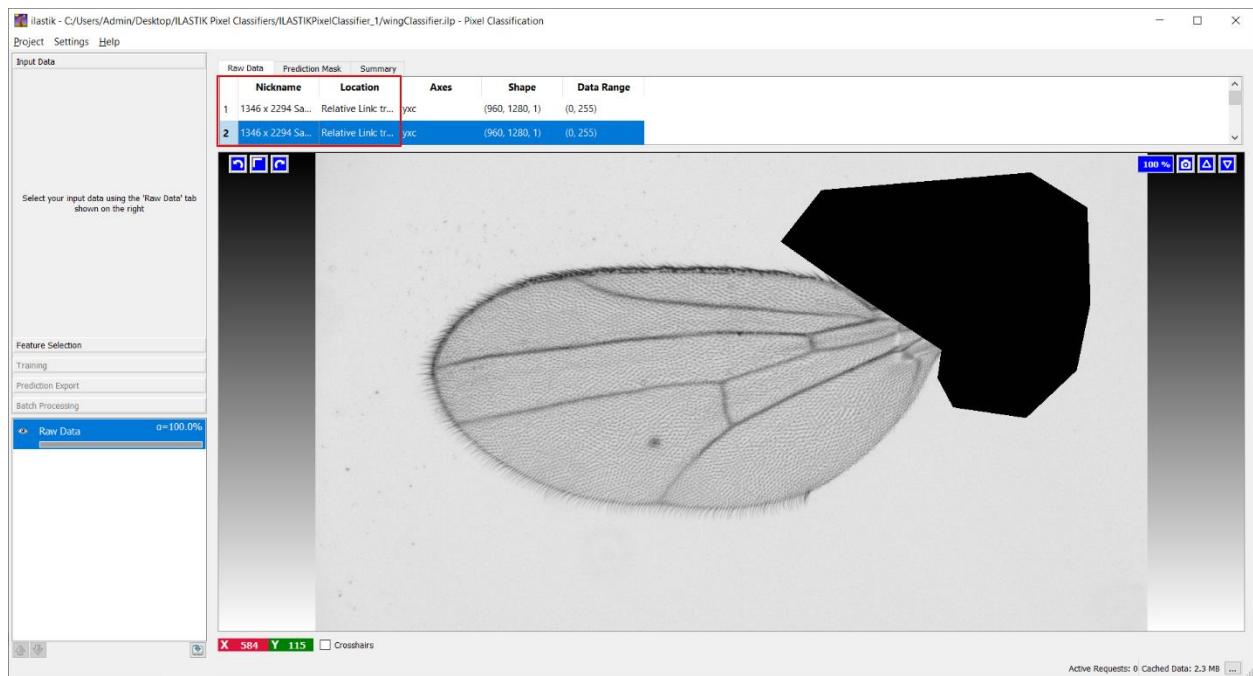


Setting up the training data and features

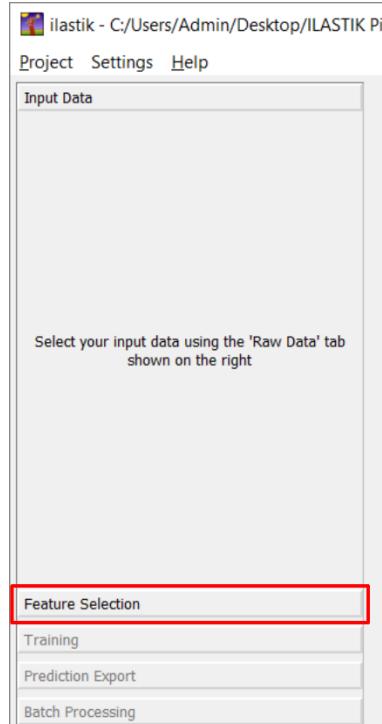
Once the project has opened, you will need to click the “Add New..” > “Add separate Image(s)…” button to select your training data. You will navigate to the location of your training data and select all of the training images.



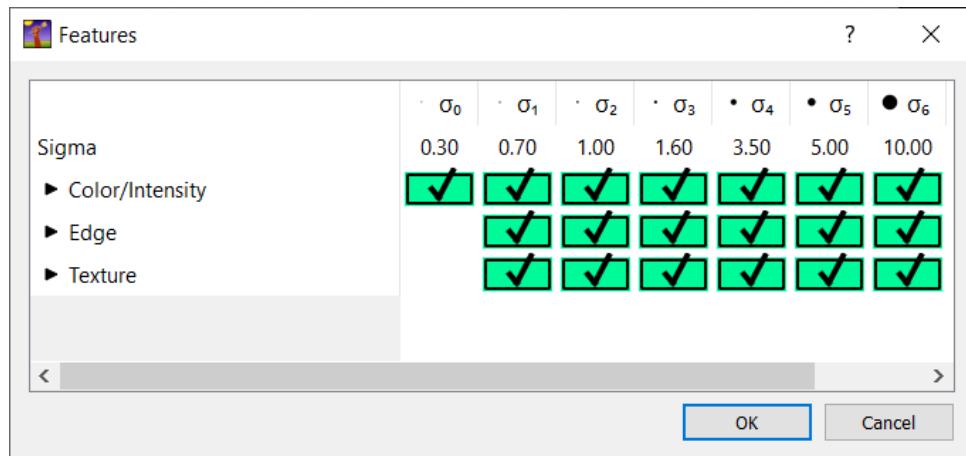
Once this is done, you will see one of your training images shown in the display as well as the “Nickname” and “Location” of the training images you specified.



You will then need to click the “Feature Selection” tab on the left-hand side of the application.

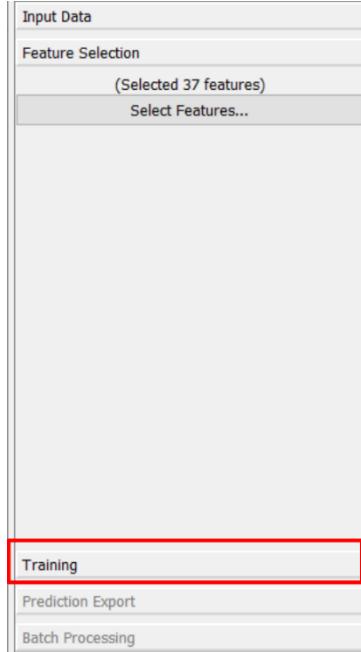


Proceed to click “Select Features...” and the available trainable features will appear. Ensure to click all checkboxes for all features and select “OK.”

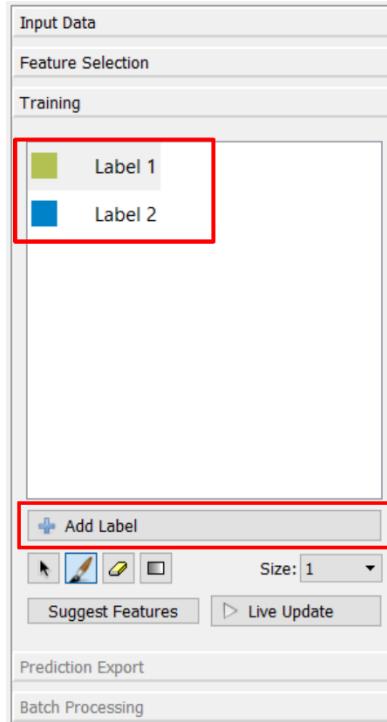


Training the pixel classifier

Click the “Training” tab on the left-hand side of the application.



There will be two default labels that appear in the “Training” tab. “Label 1” will be in yellow, and “Label 2” will be in blue. You will need to add two additional labels by clicking the “+ Add Label” button twice to have “Label 3” in red and “Label 4” in light blue. For our trained ILASTIK classifiers, we have opted to have “Label 1” (yellow) be the color for the intervein region, “Label 2” (blue) be the color for the wing veins, “Label 3” (red) be the color for the wing marginal hairs, and “Label 4” (light blue) be the color for non-wing background.



Understanding ILASTIK training tools

Each of the available buttons on the “Training” tab layout have specific functions that aid in training. For location of the buttons on the “Training” tab layout, please refer to the numbers in the image and tool names below. Those functions are detailed as follows:

Navigation Cursor (1)

This tool aids in moving around the currently visible training image. A hotkey to immediately select this cursor is “N.”

Brush Cursor (2)

This tool is used to paint over pixels with the label color you have highlighted. A hotkey to immediately select this cursor is “B.”

Eraser Cursor (3)

This tool is used to erase paint from pixels already colored in with the brush cursor. A hotkey to immediately select this cursor is “E.”

Size (4)

This dropdown selection tool is used to change the size of the brush and eraser cursors to cover more pixels per stroke.

Live Update (5)

This button will classify and segment your currently selected image using the labels you have painted. Clicking this button may take some time to run and can be tracked with a green progress bar in the bottom left-hand corner. We advise keeping this button off when using the brush and eraser cursors (as ILASTIK iteratively updates the live update with every new pixel), and only turning the button on for troubleshooting of segmented outputs.

Current View (6)

This dropdown selection tool will allow you to select the different images of your loaded training data to label all of the available training data. A hotkey to immediately switch to the next available image is the “PgDn” key.

Probability checkbox (7)

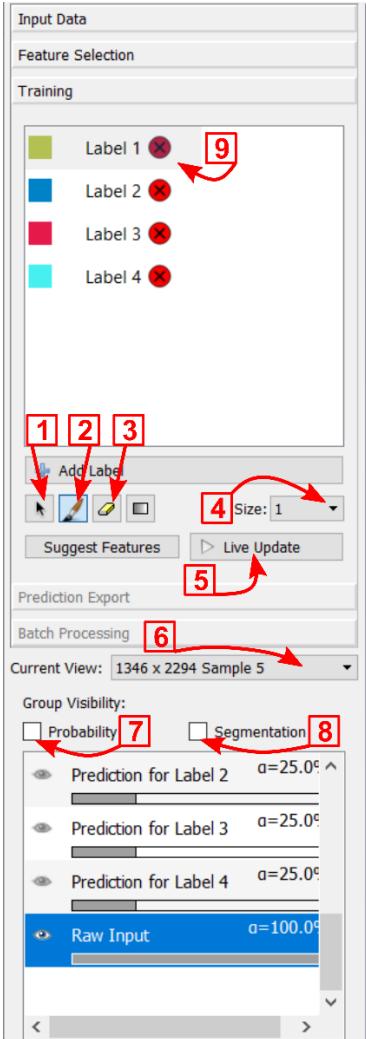
When this checkbox is filled, ILASTIK will color-code the pixels of the currently viewed image with your trained labels based on the likelihood of the pixel belonging to a particular label. This additionally requires the “Live Update” button to be turned on.

Segmentation (8)

When this checkbox is filled, ILASTIK will color-code a segmentation mask (*i.e.*, compartmentalized image) of the raw image you have currently viewed. This additionally requires the “Live Update” button to be turned on.

Selected label (9)

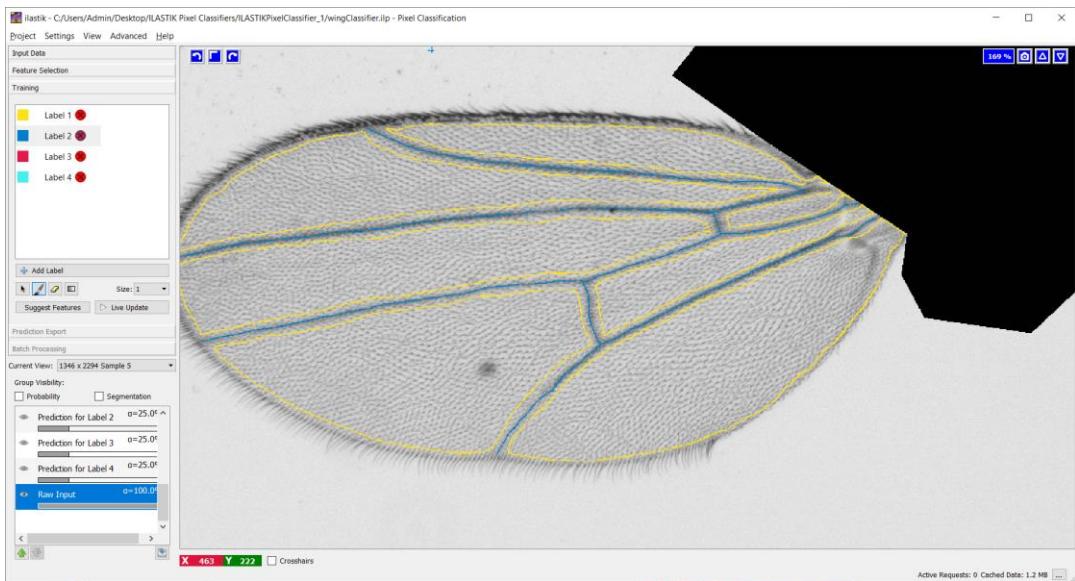
This tool allows you to change which label you would like to associate with the brush cursor. The label color that will be painted is the one highlighted in a light gray box.



To train the pixel classifier, first you must select which label you would like to paint on the viewed training image. To start with, you can select “Label 1” which corresponds to the intervein region. Once you select “Label 1” by clicking it, you can use the brush tool to paint over the intervein regions of your first image. We recommend a brush size of 1 to prevent diminishing returns on the training data at the cost of increased computational time required for ILASTIK to process images. For at least one wing, we recommend outlining entire borders of the intervein regions as seen below:



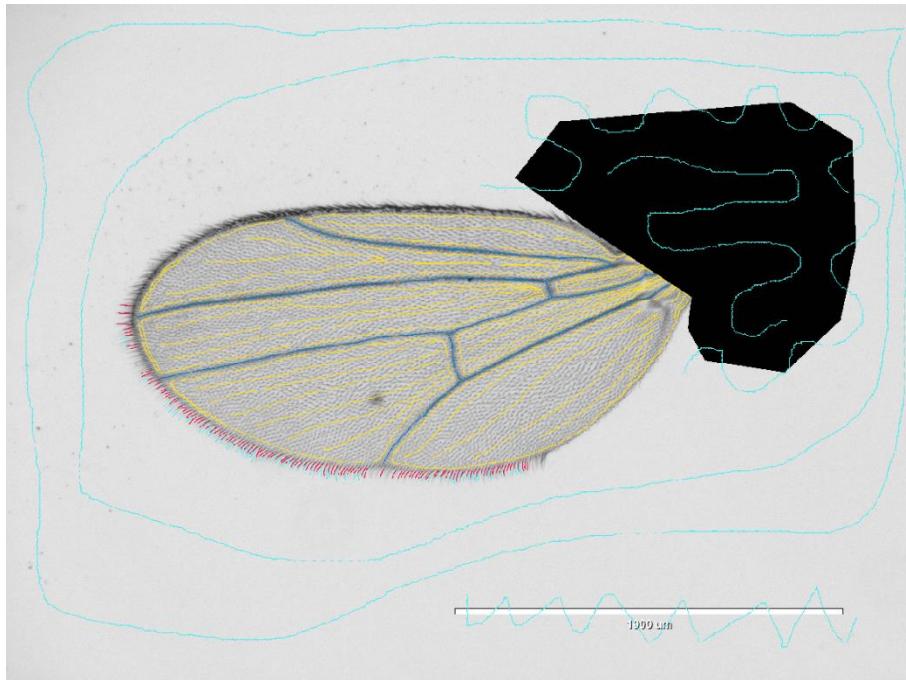
Next, select “Label 2” and begin using the brush tool to paint over the veins of the first image. Again, we recommend using a brush size of 1 and for at least one image to have outlined all of the veins.



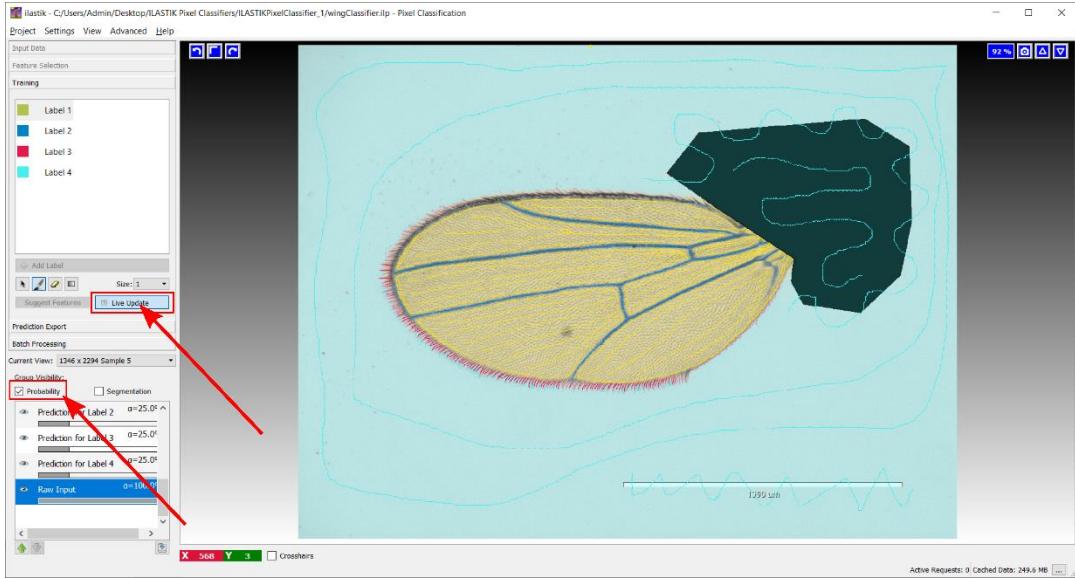
Next, select “Label 3” and begin using the brush tool to paint over the wing marginal hairs. We recommend zooming in to a specific region of the wing by holding “Ctrl” and scrolling up with the mouse in order to paint the wing marginal hairs. After you have painted the wing marginal hairs, while still zoomed in, select “Label 4” then begin using the brush tool to paint in between the wing marginal hairs to train non-wing background pixels.



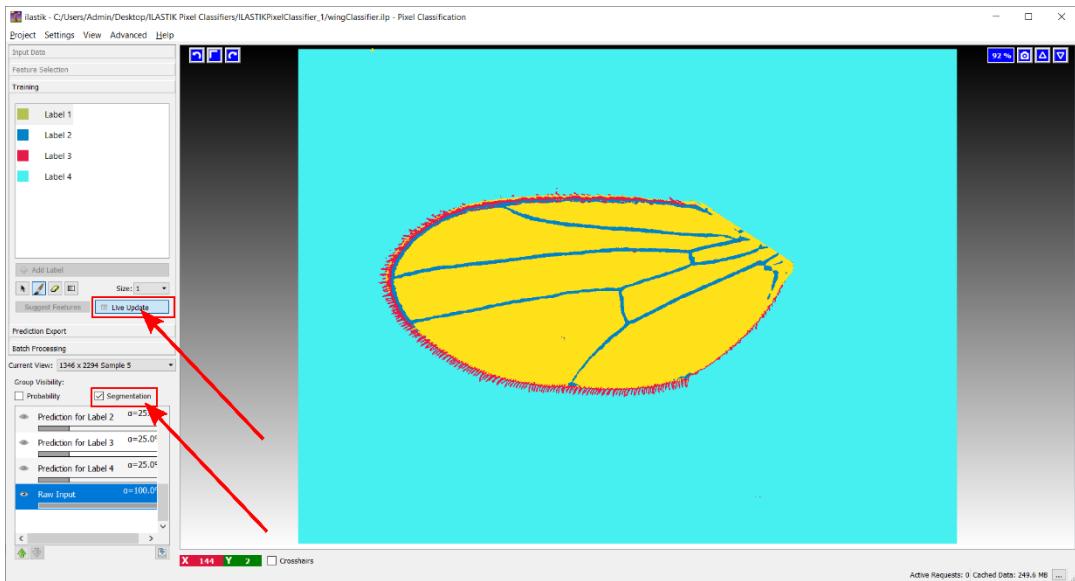
Once this is done, you can zoom out and while still having “Label 4” selected, paint over non-wing background. This will include any scale bars, cropped out wing hinge region, and the slide background. Finish the training of the image by selecting “Label 1” and painting over the middle of the intervein regions. A fully labeled sample image is seen below:



After completing this training, click the “Live Update” button for ILASTIK to process a probability mask which displays a semi-transparent segmentation mask of the wing image with the labels color-coded over the predicted labels based on the training data you painted.

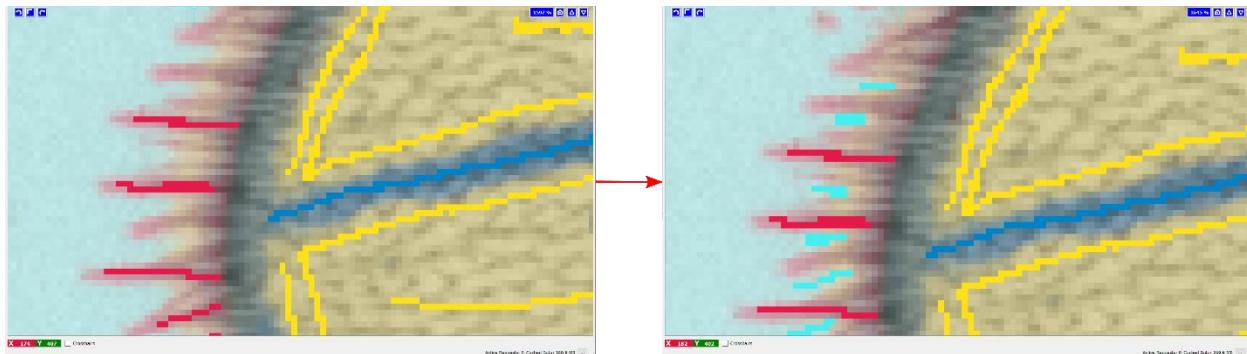


Additionally, you can select the “Segmentation” checkbox to view a non-transparent segmentation mask of the wing image.

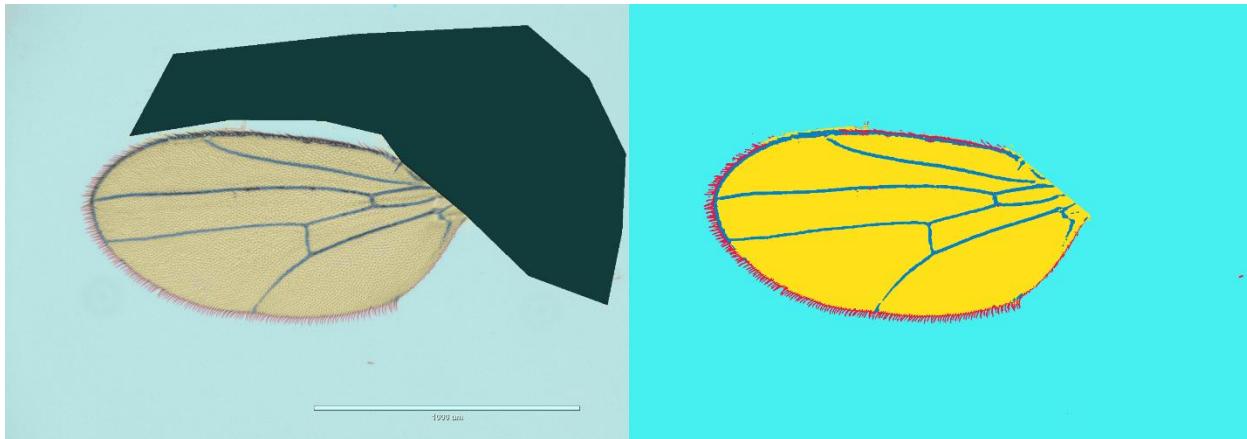


You should be able to see a fairly well segmented wing after training the image, however, certain pixels may be incorrectly labeled by ILASTIK. To correct this, turn off the “Live Update” button, select the “Probability” checkbox, unselect the “Segmentation” checkbox, and zoom in to where there are incorrectly labeled pixels. For the incorrectly labeled pixels, select the correct label in the “Training” tab layout, and use the brush tool to paint over the incorrectly labeled pixels with the correct label color. This will provide more training data to ILASTIK to prevent mislabeled pixels. In the example below, we have zoomed in to where the space between the marginal hairs is incorrectly labeled as the intervein region. This is denoted by the yellow color in between the marginal hairs; however, the yellow color label should correspond to the intervein region alone.

To fix this, we have selected “Label 4” (light blue) and painted in between the marginal hairs where the incorrectly labeled pixels are.



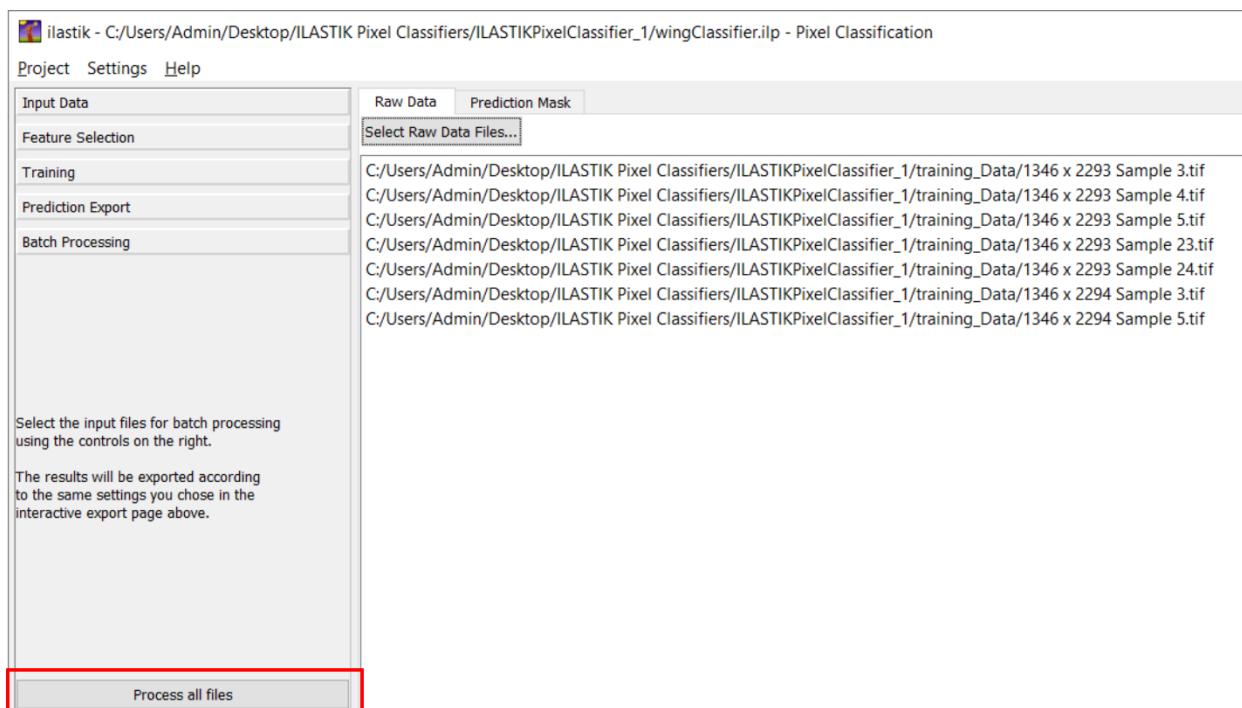
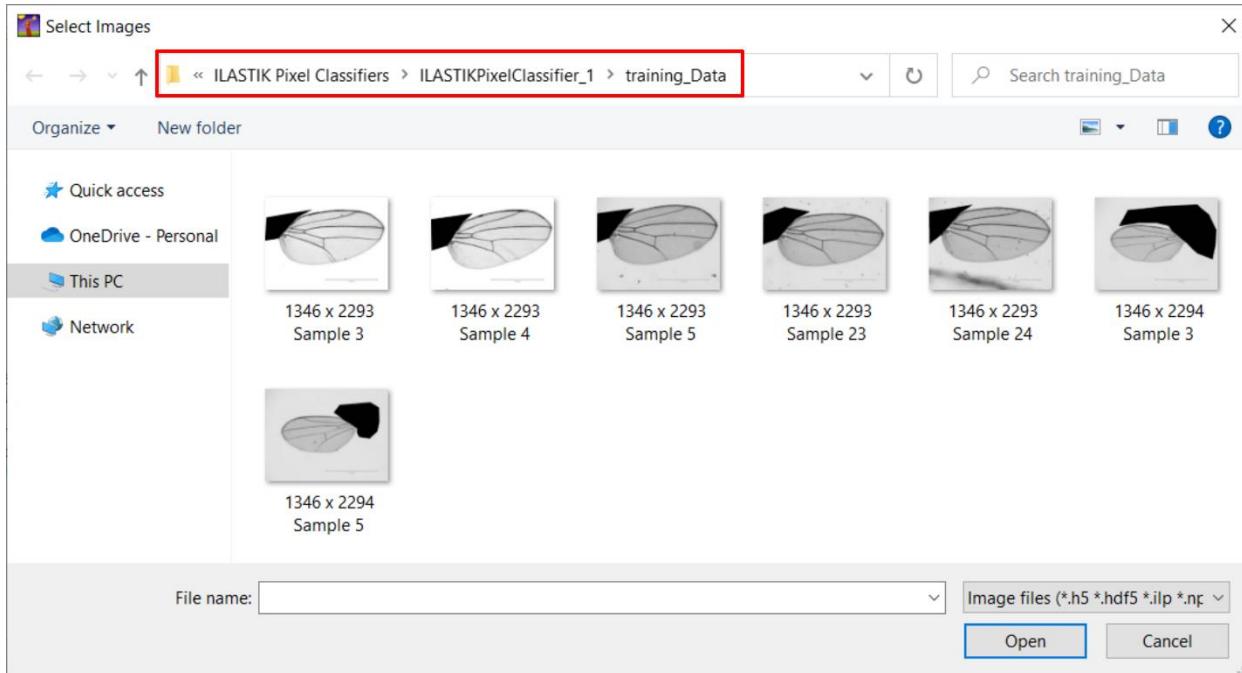
You should repeat this in a similar manner for any regions of the wing where there are mislabeled pixels. After this, you can proceed to train the pixel classifier on the next available image of your training data set. With the next image in view, you can repeat the steps of enabling “Live Update”, viewing the “Probability” and “Segmentation” masks, turning off “Live Update,” and painting over incorrectly labeled pixels. For example, without providing any training on the second image of the images, the below pictures are the probability and segmentation masks.



Using only training labels from the first image, the second image is fairly well segmented, but requires manual labelling of a few mislabeled pixels. Repeat the process of zooming in to the mislabeled pixels and painting over them in the correct label color. Once this is done, you can repeat the entire process of: 1) Moving on to the next training image, 2) clicking “Live Update”, 3) viewing the “Probability” mask, 4) turning off “Live Update”, and 5) correctly labeling the mislabeled pixels until all images are sufficiently segmented. Once this is done, save the project.

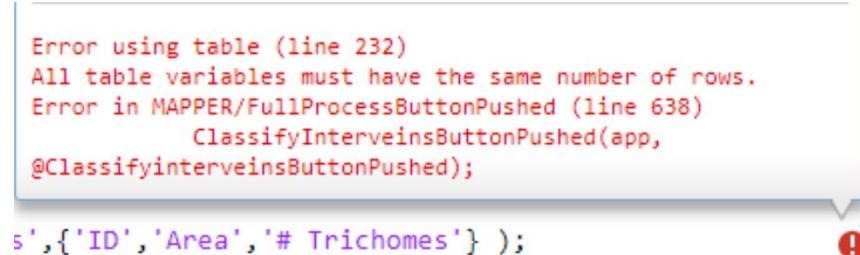
The ILASTIK “.ilp” file can now be used to process images and output segmentation masks for MAPPER. To ensure quality control, you can apply batch processing within ILASTIK to the raw images you have just trained on. To do this, follow the Step Five of this manual with the exception

of using the newly trained classifier on the training data files instead of files within the main MAPPER folder.



After processing the simple segmentations of the training data, you can use the individual processing (Step Six of this manual) to observe if your training data has been correctly processed by MAPPER. If this is the case, this new ILASTIK module is ready to use on any future data.

However, if there are errors when MAPPER attempts to process the segmentations of your training data, you may need to refine the training and pixel classification of your ILASTIK module. A common error when testing a new ILASTIK module is that the new module does not have enough training data of the pixels and their corresponding labels. This results in the following error in MAPPER:



This error message comes from the output simple segmentation file from ILASTIK not having been segmented correctly. To fix this issue, you will need to supply more training data to the ILASTIK module. With no issues in training, your newly segmented wing should appear as detailed in Step Six of the manual.

