LOBSTER Annotator manual

LOBSTER annotator provides a set of tools to assist the annotation of 2D/3D microscopy images. The results are exported as annotation masks with intensity values encoding the class of each pixel (up to 3 classes, e.g. background / object centre, object periphery).

The software is optimized to provide an interactive experience (even with limited hardware) for images of up to about 64 Megapixels. Even though it is not designed to deal with big data, larger images (up to 4 GB per file or per Z slice when loading TIFF series) can be processed on dedicated workstations with large memory (or by applying some XY downscaling).

Requirements

LOBSTER annotator requires Windows (7/8/10)

Installation

- Download **LOBSTER annotator** from the GitHub repository
- Unzip the content to an empty folder
- If not already installed, install **dotNetFx40_Full_x86_x64.exe**
- Install Matlab 2019b (version 9.7)MCR from this page.

Usage

Launch **LOBSTER_Annotator.exe**.

Configuration steps

Image format options:

Single TIFF file (3D image)	Input is a multi-TIFF 3D image
TIFF series (3D image)	Input is a Z series of 2D TIFF images representing a 3D image
TIFF folder (2D image(s))	Input is a set of 2D TIFF images

Large image detected

For images exceeding 64 Megapixels (or 1 Megapixels per Z slice), XY downscaling is recommended by the software. This recommendation can be bypassed but, depending on your hardware, you might run out of memory or the processing might become significantly slower!

Image viewer control

Intensity adjustments	Hover mouse over image by holding left button
Lookup table	Set from top drop-down menu
Scroll through Z slices	Mouse wheel
Zoom	Hover mouse over image by holding wheel button
Pan	Hover mouse over image by holding right button

Commands

Annotation Tools

(P) **Paint** Segment objects from interpolated 2D contours drawn in different Z slices

(Y) **Cylinder** Segment filaments/tubes by cylinders

(W) **Wand** Segment objects by growing regions of similar intensity from a starting point

(B) **Blob** Detect and segment blob-like objects

(K) **Skeleton** Skeletonize Class 1 in annotation mask

(C) **Classify** Classify pixels interactively with random forest over block features

(U)**net Batch** Classify pixels from pre-trained U-NET network (2D images only)

Utilities & Display

(T) **Transfer** Transfer annotation mask classes to different classes

(E) **Erase** Erase annotation mask

(M) **Mask** Toggle annotation mask visibility

(Z) **Projection** Locally Z-project annotation mask (user defined depth)

(V) **3D Render** 3D render (user defined region and depth)

(F) **Fit** Iteratively refine annotation mask (experimental)

File & Options

(S) **Save** Save annotation mask (same folder as image + **_ann** suffix)

(L) **Load** Load annotation mask (same folder as image + **_ann** suffix)

(G) **Grab** Extract an object and export it to a TIFF file (at original resolution)

(N) **New** Process a new image

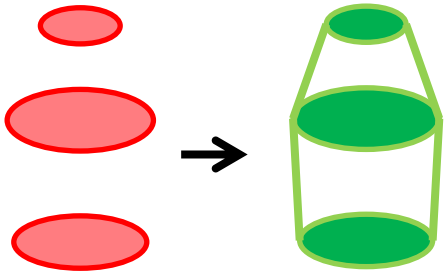
(O) **Option** Set Z Ratio, Z projection & 3D rendering depths, and annotation mask opacity

(H) **Help** Open manual

(Q) **Quit** Exit

(P) Paint

Sample test images: Mouse3D, CellPilar3D



A 3D object is segmented by interpolating closed contours drawn in different Z slices. The contours must be drawn in **Class 1** (press '**1**' and hold the left button, release upon completion). The contours can be corrected by drawing with **Class 0** ('**0**') to erase them, or competed with **Class 1** (add missing part). Once all the contours are drawn, press '**i**' to interpolate (at least 2 contours must be drawn). The result is stored in **Class 2**.

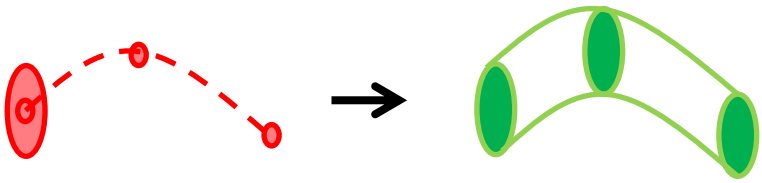
Alternatively, use the two brush tools (right side panel) to draw the contours. The bottom brush is a regular brush painting with the last class selected (left button) or erasing (right button). The top brush is a "Smart brush" performing local thresholding of the bright/dark areas (use '**b**' to switch) inside the brush region.

The radius of the brushes can be adjusted by holding the wheel and moving the mouse. It is not necessary to fill holes inside contours (as long as the contours are closed) as this is automatically performed prior to interpolation. After usage, make sure to deselect the brush and left click on the image to proceed.

Note: Brush strokes cannot be undone with Undo.

(Y) Cylinder

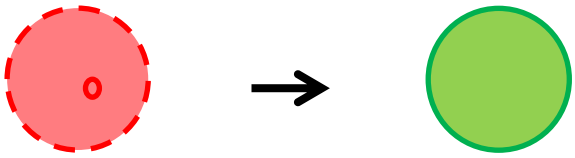
Sample images: BloodVessels3D, Fungi3D



A filament or a tube is traced by laying down points along its centreline and moving a disk along it. The first step is to set the diameter of the tube by clicking 2 opposing points on the surface of the object. For accurate results, these points should be drawn in a Z slice where the tube intersects the XY plane perpendicularly (disk) or parallelly (rectangle). The nodes of the cylinder should be set along its centreline (possibly in different Z slices) and the cylinder is drawn by pressing '**Esc**'.

(W) Wand

Sample images: Mouse3D, BloodVessels3D, Fungi3D, Nuclei



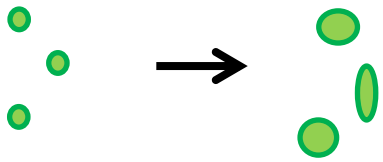
Starting from a user seed, the wand connects neighbouring pixels with similar intensity. The intensity tolerance is set between 0 and 1 as the maximum relative intensity decrease from the seed (e.g. 0.5 for a maximum variation of 50%). The tolerance can be adjusted from the Option panel ('**o**'). From this panel, it is also possible to restrict the growth of the region to a maximum distance (**Wand distance**), adjust the strength of the pre-filter (**Blur**, increase value for noisy image), or prevent bright pixels to be included if they are out of tolerance **(Threshold low/up**).

Objects in **Class 1** can be split apart by pressing '**s**' or by manually drawing a cut line with '**c**'. Fragments of objects from **Class 1** that have been erroneously split apart can be manually merged back by pressing '**j**' and clicking on two fragments. Finally, an object can be removed by pressing '**r**', or kept by pressing '**f**' (all unconnected objects removed). The parameters used for automated object splitting can be adjusted from Options '**o**':

- **Algorithm** (0/1): watershed or region descent (faster but less accurate)
- **Distance map blur radius** (0-15 pixels). Increase radius to split less objects
- **Kept 2D holes minimum area** (0-1000 pixels): the minimum area of kept 2D holes

(B) Seeded blobs

Sample images: CellPilar3D, Nuclei



Automatically mark blob-like bright objects by pressing '**s**'. Markers can also be added manually by pressing '**a**' and removed with '**r**'. The parameters used to mark the objects automatically can be adjusted with '**o**':

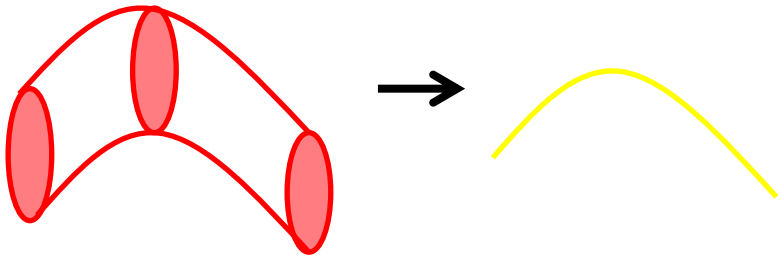
- **Sigma** (0.5 – 8 pixels): Increase for larger objects
- **LocalMaxBox** (3 – 25 pixels): Increase to limit spurious detection for larger objects
- **MinLog** (0.01-1): Increase to decrease detection sensitivity

Seeds can be grown to segment objects by pressing '**f**'. Objects can be removed by pressing '**r**'.

Note: For best results, it is recommended to add one (or several) seed(s) to the background, and then remove the objects connected to these seeds.

(K) Skeleton

Sample images: BloodVessels3D, Fungi3D



Estimate the skeleton (centreline) of objects in **Class 1** by pressing '**k**'. In case the results are not satisfactory, remove all skeleton traces by pressing '**e**' and restart. The parameters used for the pre-processing steps and as post-processing to analyse the resulting skeleton can be edited from Options '**o**'.

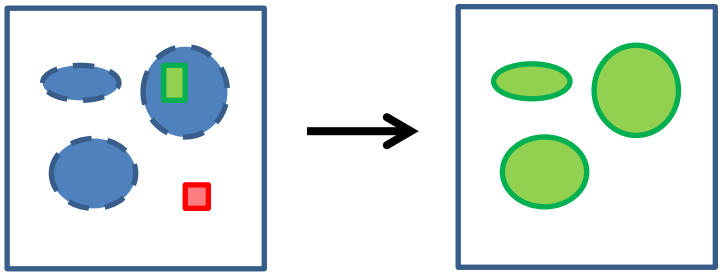
- **PreCloseRad** (0-9 pixels): Increase to remove more small irregularities
- **2D holes fill area** (0-1000 pixels): Increase to fill larger 2D holes
- **MinVol** (0-1000 pixels): Increase to remove larger skeleton fragments
- **Minimum branch length** (0-64 pixels): iteratively prune small branches
- **Pruning iterations** (0-7): number of branch pruning iterations
- **Prune 4-way branching points** (0/1): remove 4-way branch points

Analysed skeletons can be further edited: remove a branch or a branch point by pressing '**b**' and clicking it. Manually add a missing branch with '**t**' (set start and end points inside **Class 1**). To add a branch including branching / end point, set **Trace mode** to '**1**' from Option '**o**'.

Note: Skeleton pixels are encoded in the annotation mask with values 200, branch points with value 220 and end points with value 250.

(C) Classify

Sample images: Any



Starting pixel classification erase the annotation mask since all pixels from the image are classified by this function. Classification is typically performed iteratively by first annotating some small areas (training), classifying (predicting) and refining the result by annotating some more pixels. Three classes are available (1-3) and at least pixels from 2 classes should be annotated before training/predicting.

Pixel regions are annotated by pressing the number of the corresponding class and drawing small rectangular areas. **Class 1** (transparent) should be used as the background class and Class **2** and **3** for objects. If needed, toggle the visibility between classification results and manual annotations by pressing '**a**'. Annotations can be cleared by drawing over with **Class 0**.

After annotating, press '**Space**' to classify the current slice (fast), or '**Enter**' to classify the whole image (slower). This includes a round of training (unless the model was loaded from file) followed by prediction. Trained classifiers are saved by pressing '**s**' and loaded by pressing '**l**'.

The default classifier settings should cover most scenarios but they can be set from **Option** '**o**':

Number of trees (5-50): Higher number can improve classification but slow down operation

Prediction level: Skip blocks which mean intensity is above (positive level) / below (negative level) the absolute value of prediction level and set them to background.

Local threshold (0-1): Pixels which intensity is below this fraction of their associated block mean intensity are set to background. Set to 0 to disable.

Use spatial features (0/1): When set to 1 the classifier is aware of blocks location. This can help classifying touching structures with similar intensity to different classes.

(D) Deep learning batch classify (2D image folder only)

Classify the pixels of all the images from the folder by applying pre-trained U-NET deep learning network. The network model should be stored in the subfolder **/Models** inside the image folder. In case several models are stored in this folder the first one in alphabetic / numeric order is used. The results are stored as annotation masks that can be loaded with '**l**'.

A new network can be trained from manual annotations by preparing annotation masks for the images and saving them as PNG files with the same name as the images and **_ref** suffix.

Make sure that a **/Models** subfolder is present in the image folder and run **LOBSTER_Annotator_UNET_Train.exe** to train the network. The network is exported to the **/Models** subfolder as **Model.mat**. It is highly recommended to use a modern GPU when training a new network as this operation can last several hours on a regular CPU (compared to few minutes).

3D render selected volume (V) – Windows only (Microsoft .NET 4 should be installed)

Set the region to 3D render by holding the left mouse button to draw a rectangle. The depth of the region is set from **Option** '**o**'. Make sure that the Z Ratio of the image is properly set from **Option** '**o**' so that the objects are displayed with the correct aspect ratio. Zoom by holding the right mouse button. Annotation mask can be toggled with '**m**'. Lower and higher intensity can be clipped by pressing '**c**'. Reset view with '**Space**'. Close the window of the viewer to exit. **Class 1** is not displayed (unless it is the only class defined in the mask) as it is considered the background class.

Option (O)

Z Ratio (0.25 – 2.5): The ratio between slice spacing and pixel size. Used for 3D rendering and to draw cylinders with valid aspect ratio

Z projection depth (3-127 slices): The number of slices used for local Z projection ('**z**')

3D rendering depth (8-Inf slices): The number of slices (around current slice) used for 3D rendering ('**v**'). Set to '**Inf**' to render all slices

Mask opacity (0.1 - 1): The opacity of the annotation mask.