LOBSTER Annotator manual

LOBSTER Annotator brings a set of tools to assist the annotation of 2D/3D microscopy images. It can process single channel 2D/3D TIFF files, or a folder of 2D TIFF files where each image represents a Z slice of a 3D volume. Folder of independent 2D images can also be processed sequentially in the same session. The annotations are exported as 2D/3D TIFF masks.

The software is optimized for an interactive experience with limited hardware for volumes of up to about 64 Megavoxels. Larger images[1] can be processed on dedicated workstations, or at a reduced scale (e.g. to extract coarse regions from large images).

Requirements

LOBSTER executable is only provided for Windows (7/8/10).

Installation

- Download **LOBSTER annotator** from the GitHub repository
- Unzip the content to an empty folder
- Install Matlab 2021a (version 9.10) Runtime from this page

Usage

Launch **LOBSTER_Annotator.exe**.

[1] The software cannot process 3D TIFF files larger than 4 GB (BigTIFF), but it is possible to process larger 3D images exported as a 2D image series.

Initial Dialog Boxes

Image format options:

3D TIFF file (4 GB max)	Input is a multi-TIFF 3D image
TIFF series (3D image folder)	Input is a Z series of 2D TIFF images from the same volume
2D TIFF images (folder)	Input is a set of 2D TIFF images to be processed independently

Large image detected

For volumes exceeding 64 Megavoxels or 1 Megapixels per slice, image downscaling is recommended by the software. This can be bypassed by forcing downsampling to 1 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 by Z interpolating user contours

(Y) **Cylinder** Model filaments or tubes as plain cylinders

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

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

(K) **Skeleton** Skeletonize annotation mask

(C) **Classify** Classify pixels interactively

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

Utilities

(E) **Erase** Erase annotation mask

(T) **Transfer** Swap annotation mask classes

(F) **Fit** Refine annotation mask (experimental)

(G) **Grab** Extract an object from annotation mask and export it

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

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

Annotations Display

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

(Z) **Projection** Toggle depth projection of annotation mask

(V) **3D Render** 3D render a user defined region of interest

File & Options

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

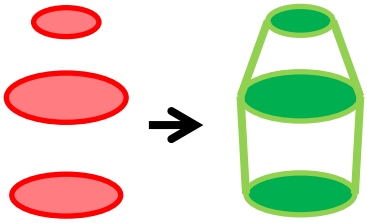
(O) **Option** Set: Z Ratio, Z projection depth, 3D options, annotation mask opacity

(H) **Help** Open manual

(Q) **Quit** Exit

(P) Paint

Sample test images: Mouse3D, CellPilar3D



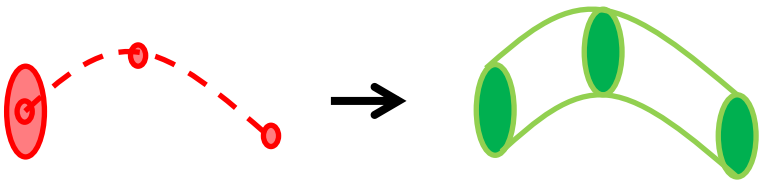
A 3D object is segmented by Z interpolating contours drawn by the user in different Z slices. The contours are drawn in **Class 1** by first pressing '**1**' and then holding the left button (release upon completion). The contours can be erased by drawing with **Class 0** ('**0**'). Press '**i**' to interpolate (contours should be drawn in at least two slices). The result is stored to **Class 2**.

Alternatively, use the brush tools (icons in the right panel) to draw the contours. The bottom icon is a regular brush painting (left button) with the last class selected or erasing (right button). The top icon 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. Make sure to deselect the brush and left click on the image after usage.

Note: Brush strokes are the only operations that cannot be undone by Undo!

(Y) Cylinder

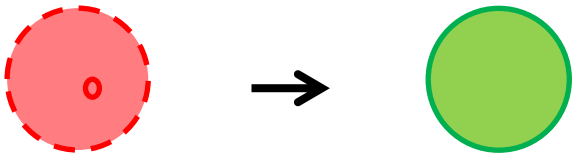
Sample images: BloodVessels3D, Fungi3D



A filament or a tube is traced by sliding a disk of fixed radius along. The first step consists in setting the diameter of the tube by setting 2 opposing points at the surface of the object. For accurate results, these points should be drawn at a position where the tube intersects the Z slice more or less perpendicularly, or where it runs parallel. Nodes are laid along the centreline of the tube (possibly in different Z slices), and the cylinder is drawn by pressing '**Esc**'.

(W) Wand

Sample images: Mouse3D, BloodVessels3D, Fungi3D, Nuclei



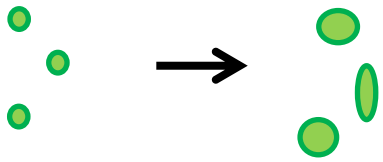
Starting from a user seed, connects neighbouring pixels with similar intensity. The intensity tolerance is set (0 to 1) as the maximum relative intensity decrease from the seed (e.g. 0.5 for a maximum variation of 50%). The tolerance can be adjusted from the Option panel ('**o**'). From this panel, it 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**).

Convex objects from **Class 1** can be split apart automatically by pressing '**s**', or from manually set cut lines (use '**c**'). Objects from **Class 1** that have been erroneously split apart can be manually merged by pressing '**j**' and clicking on two fragments. Finally, an object can be removed by pressing '**r**', or only kept by pressing '**f**' (remove all unconnected objects). 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): minimum area of significant 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**':

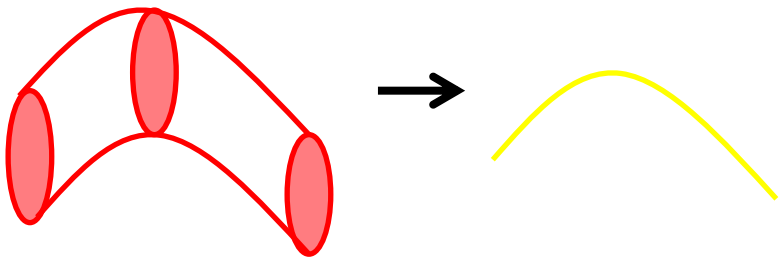
- **Sigma** (0.5 – 8 pixels): Increase for larger objects
- **LocalMaxBox** (3 – 25 pixels): Increase to limit spurious detection for larger objects
- **MinLog** (0.01-1): Increase to decrease detection sensitivity

Objects can be segmented from markers with '**f**' and removed individually with '**r**'.

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

(K) Skeleton

Sample images: BloodVessels3D, Fungi3D



Estimate the centreline of objects in **Class 1** by pressing '**k**'. In case the results are not satisfactory, remove centrelines by pressing '**e**' and adjust parameters with '**o**':

Pre-processing:

- **PreCloseRad** (0-9 pixels): Increase to remove small irregularities
- **2D holes fill area** (0-1000 pixels): Increase to fill 2D holes creating bee-hives artefacts
- **MinVol** (0-1000 pixels): Increase to remove small skeleton fragments

Post-processing:

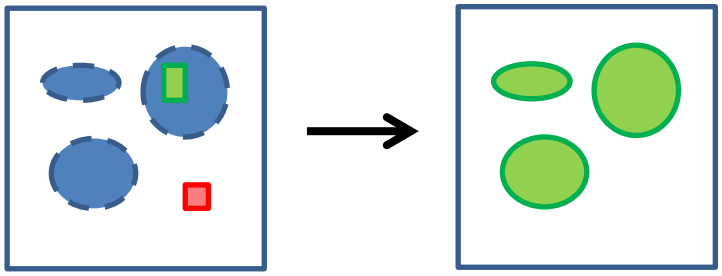
- **Minimum branch length** (0-64 pixels): Iteratively prune small branches
- **Pruning iterations** (0-7): Number of branch pruning iterations
- **Prune 4-way branching points** (0/1): Remove 4-way branch points

Centrelines can also be further edited: remove a branch or a branch point with '**b**' and left click. Manually add a branch with '**t**' (select start and end points inside **Class 1**). To add a branch with a new branch / end point, set **Trace mode** to '**1**' from '**o**'.

Note: Centreline pixels are encoded in the annotation mask with intensity 200, branch points with intensity 220 and end points with intensity 250.

(C) Classify

Sample images: Any



This tool will attempt to classify all the pixels from the image, the annotation mask in hence cleared prior to starting. Classification is performed iteratively by first annotating small regions (training), predicting and refining the result by annotating some more regions if needed. Three classes are available (1-3) and at least two classes should be annotated before training. It is recommended to use **Class 1** for background as it appears transparent.

Regions are annotated by pressing the number of the corresponding class and drawing small rectangular areas. 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 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 changed from '**o**':

Number of trees (5-100): Higher values 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 the prediction level (and set them to background).

Local threshold (0-1): Pixels which intensity is below this fraction of their 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 block locations. This can help differentiating structures with similar intensity but at different locations.

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

Classify the pixels of the images in input 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. For training, make sure that a **/Models** subfolder is present in the image folder and run **LOBSTER_Annotator_UNET_Train.exe**. 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 CPU (compared to few minutes).

3D Render (V)

Note: Microsoft .NET 4 should be installed. If the 3D viewer does not run, install it from this webpage: https://www.microsoft.com/en-us/download/details.aspx?id=17718

Set the region to 3D render by holding the left button to draw a rectangle. The depth of the region is set from '**o**', as well as the Z aspect ratio of the image. Zoom by holding the right mouse button. Annotation mask can be toggled with '**m**'. Low and high intensity can be clipped by pressing '**c**'. Reset view with '**Space**'. Close the window of the viewer to exit.

Note: **Class 1** is not displayed (unless it is the only class in the annotation mask).

Options (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 - ∞ 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): Set the opacity of the annotation mask.