Theory and Practical Uses of SuRVoS2

Michele C. Darrow, Imanol Luengo, Avery Pennington, Olly King, Elaine M. L. Ho, Matt Spink, Win Tun, Mark Basham

Image Processing

Computer Vision

Machine Learning

Computational Techniques

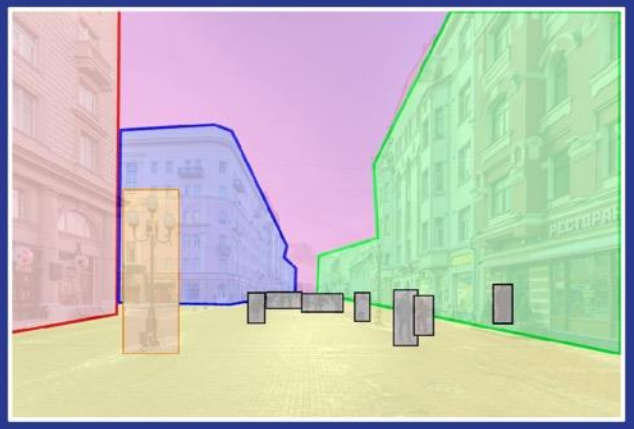
Computer Vision

Digital Image Understanding

Processing

Analysis

*



* Original Image: https://en.wikipedia.org/wiki/Arbat_Street

Image Processing

Computer Vision

Machine Learning

Computational Techniques

Image Processing

Image Manipulation and Enhancing

Noise Reduction

Feature Extraction

*

* Original Image: https://en.wikipedia.org/wiki/Arbat_Street

Image Processing

Computer Vision

Machine Learning

Computational Techniques

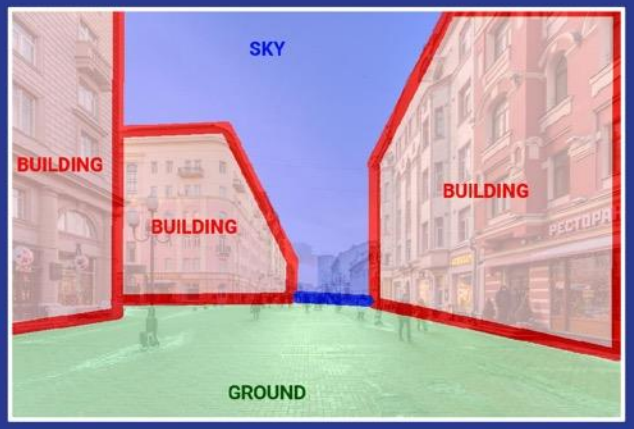
Machine Learning

Computational Learning

Classification

Regression

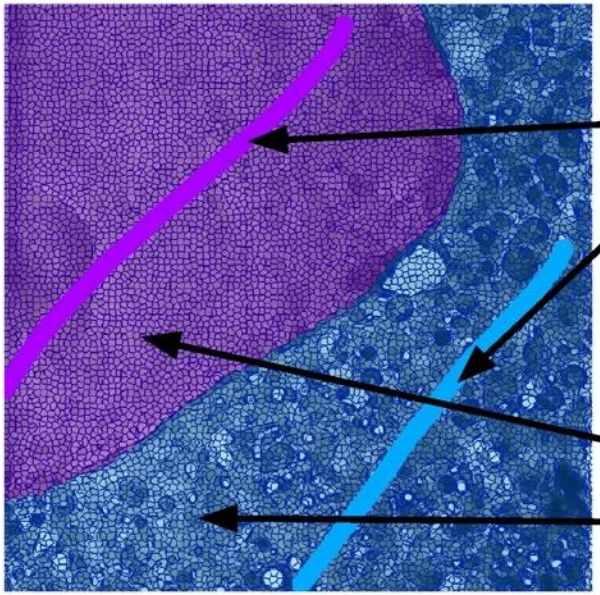
*



* Original Image: https://en.wikipedia.org/wiki/Arbat_Street

Problem

- Different imaging modalities / cell type
- Organelles have different shape / appearance
- **No previous training data is available**



Annotations

Nucleus

Cytoplasm

Predictions

Nucleus

Cytoplasm

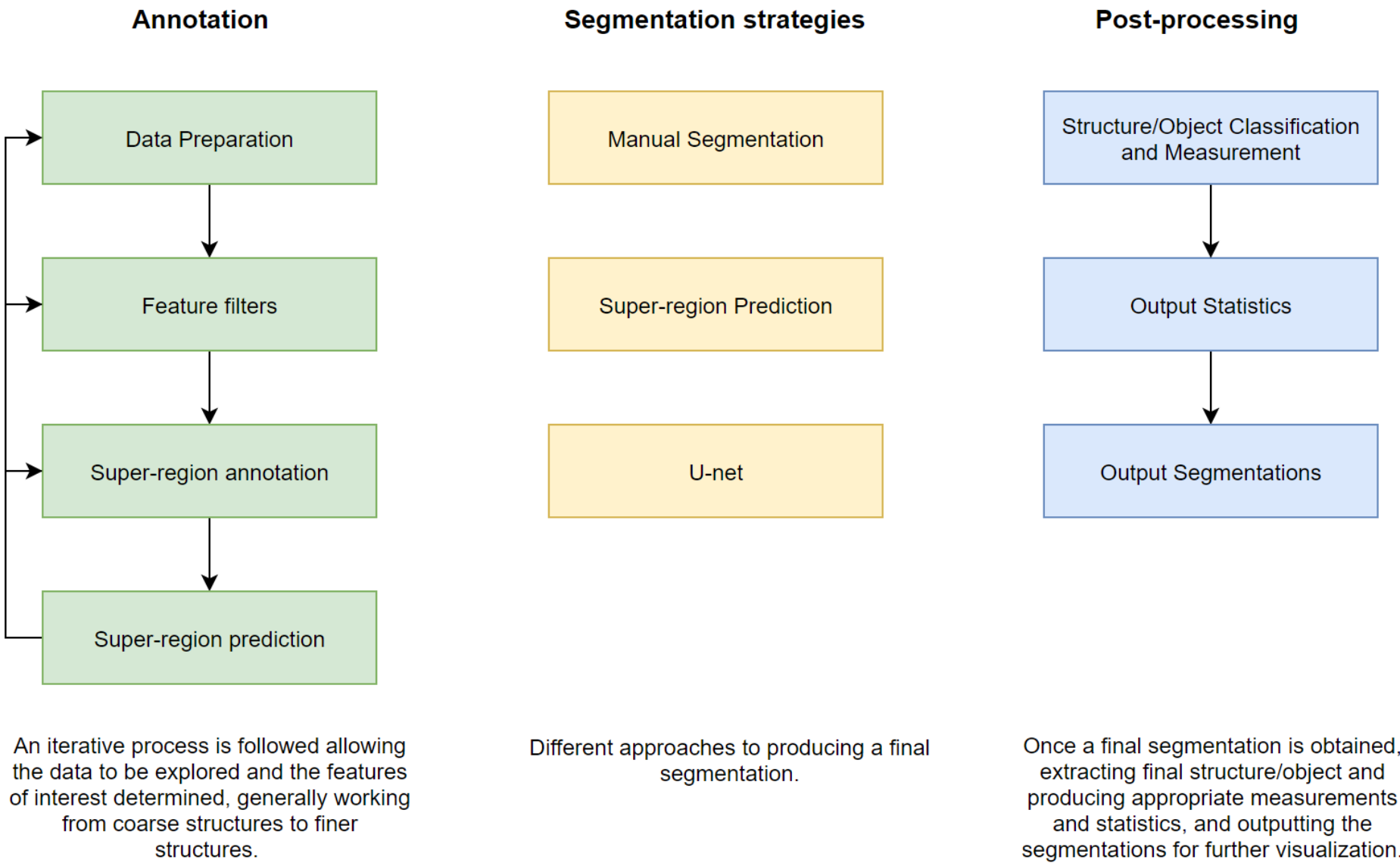
SuRVoS

- Assist the user to annotate data.
- Learn to segment with user annotations.

Assisted Interaction

Human Knowledge

Human Users



An iterative process is followed allowing the data to be explored and the features of interest determined, generally working from coarse structures to finer structures.

Different approaches to producing a final segmentation.

Once a final segmentation is obtained, extracting final structure/object and producing appropriate measurements and statistics, and outputting the segmentations for further visualization.

Multi-axis Encoder-Decoder CNN

- CNN-based segmentation models, particularly the U-net, are a state-of-the-art segmentation model when trained with sufficient, high-quality annotation (and when predicting on a suitable volume).

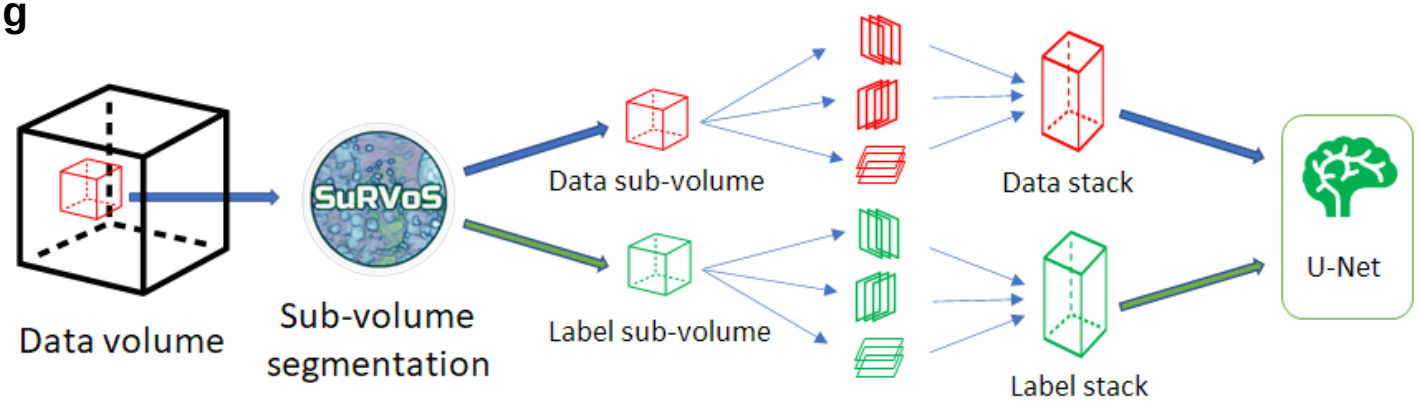
- The U-net is provided as a plugin in the Pipelines panel.

- The U-net can be trained on a volume e.g. a 128x128x128 cube of precisely annotated data.

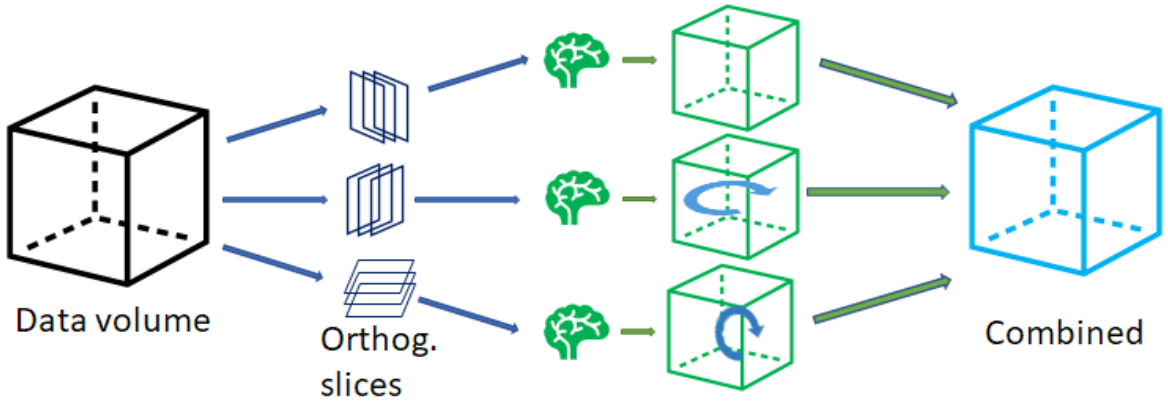
- By using SuRVoS supervoxel annotation and the super-region segmentation pipeline, it is possible to annotate a detailed volume of biological image data and export it as training data for a U-net model.

U-Net: Training and Predicting in Multiple Dimensions

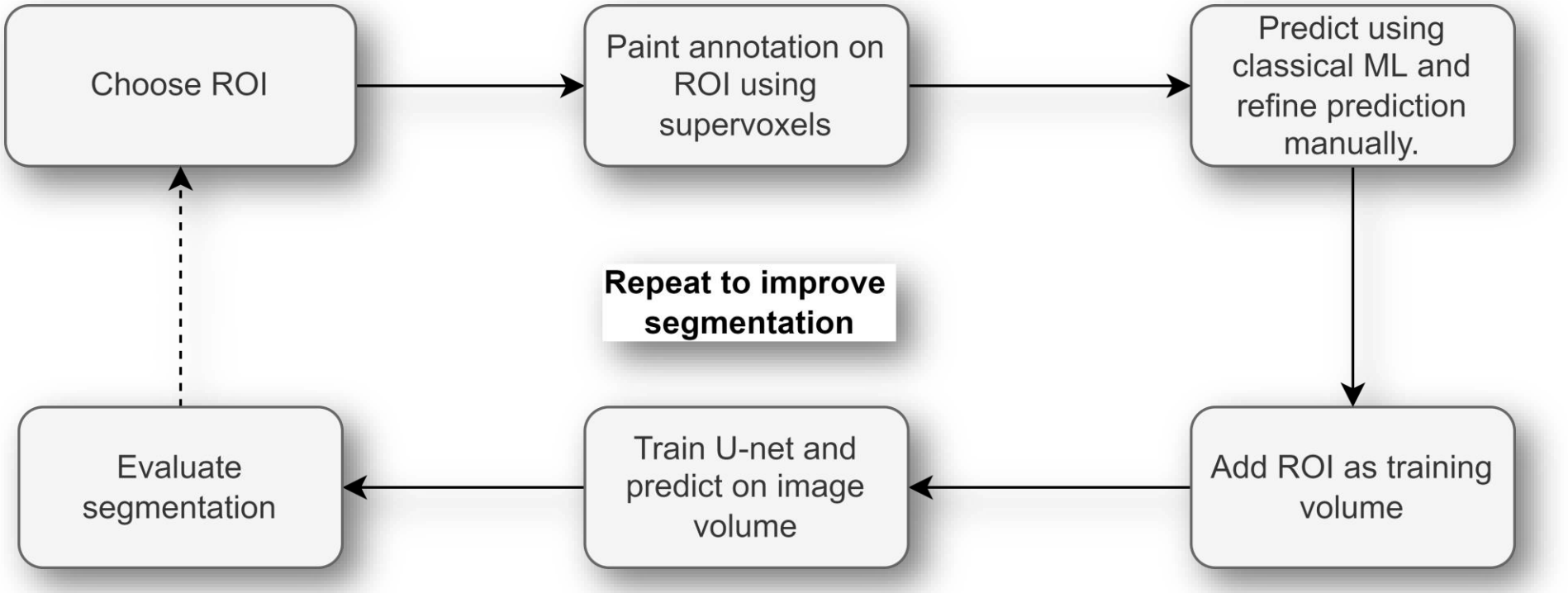
Training



Prediction

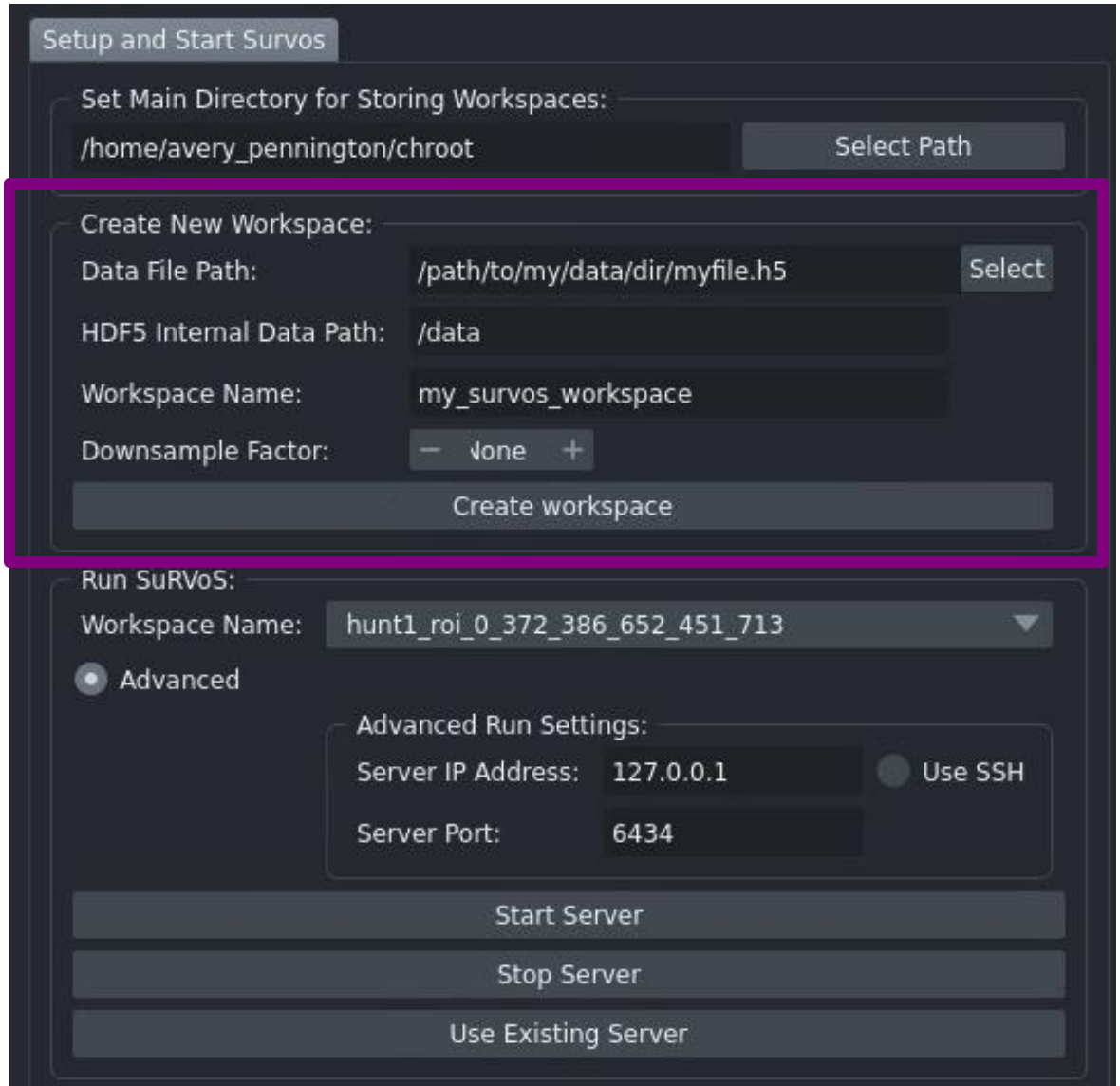


Segmentation Workflow



Training Manual

Running SuRVoS

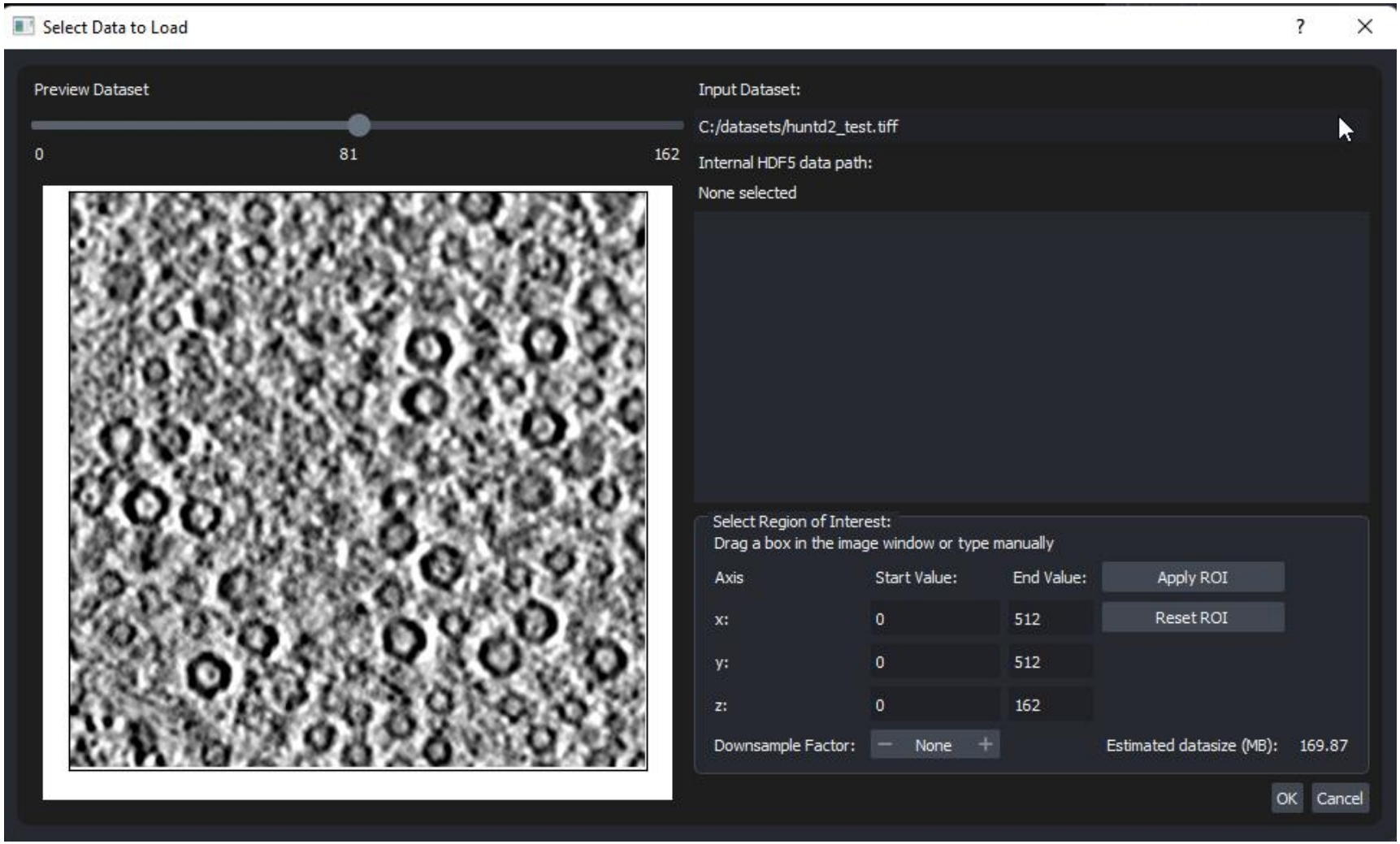


1. For a new image, under Data File Path click Select.

2. (Next slide) Use the ROI selection panel to select an ROI

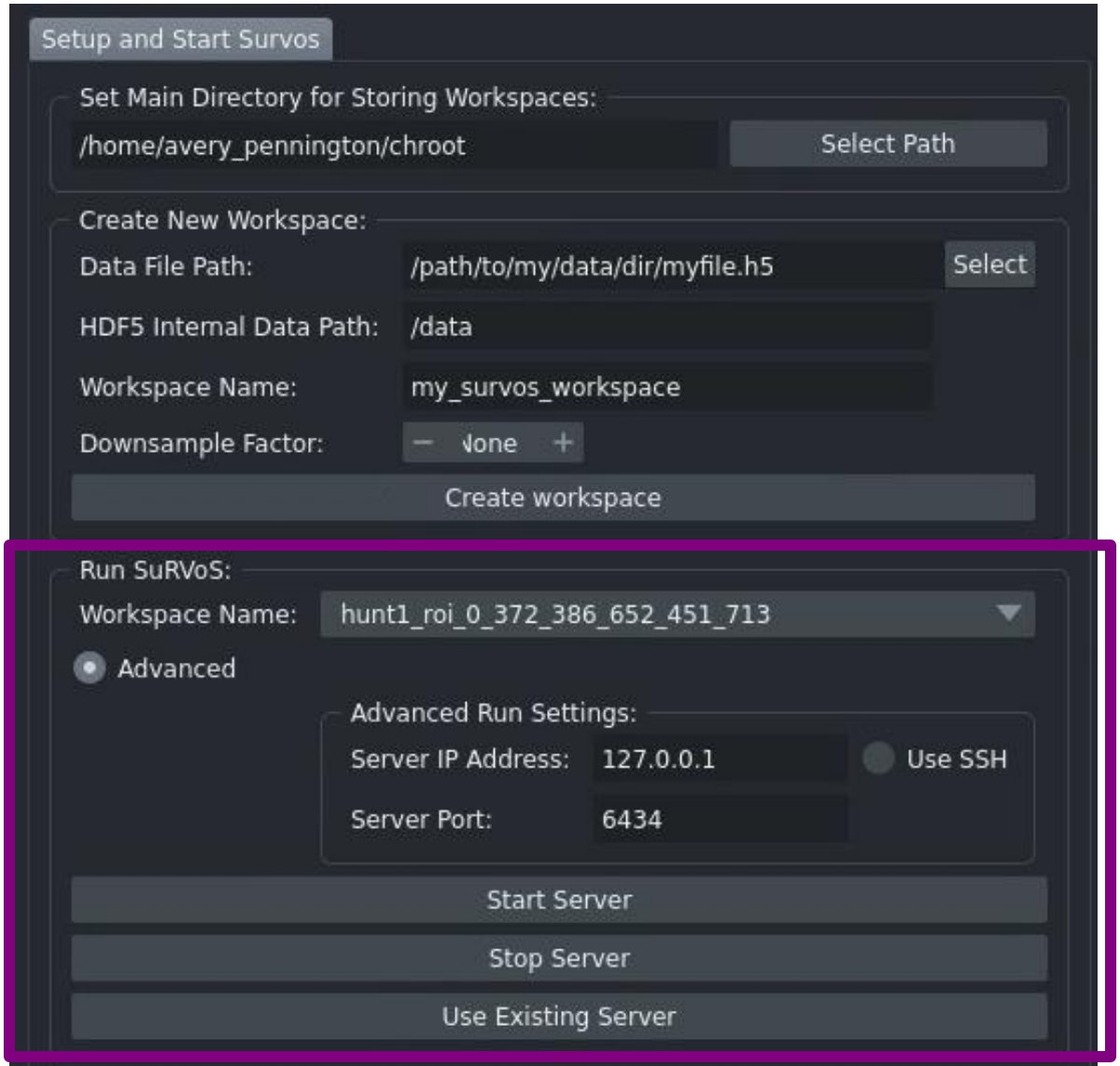
3. Give the workspace a name

4. Click 'Create Workspace'



This window is used to select the Region of Interest and crop your image to that Region.

Running SuRVoS



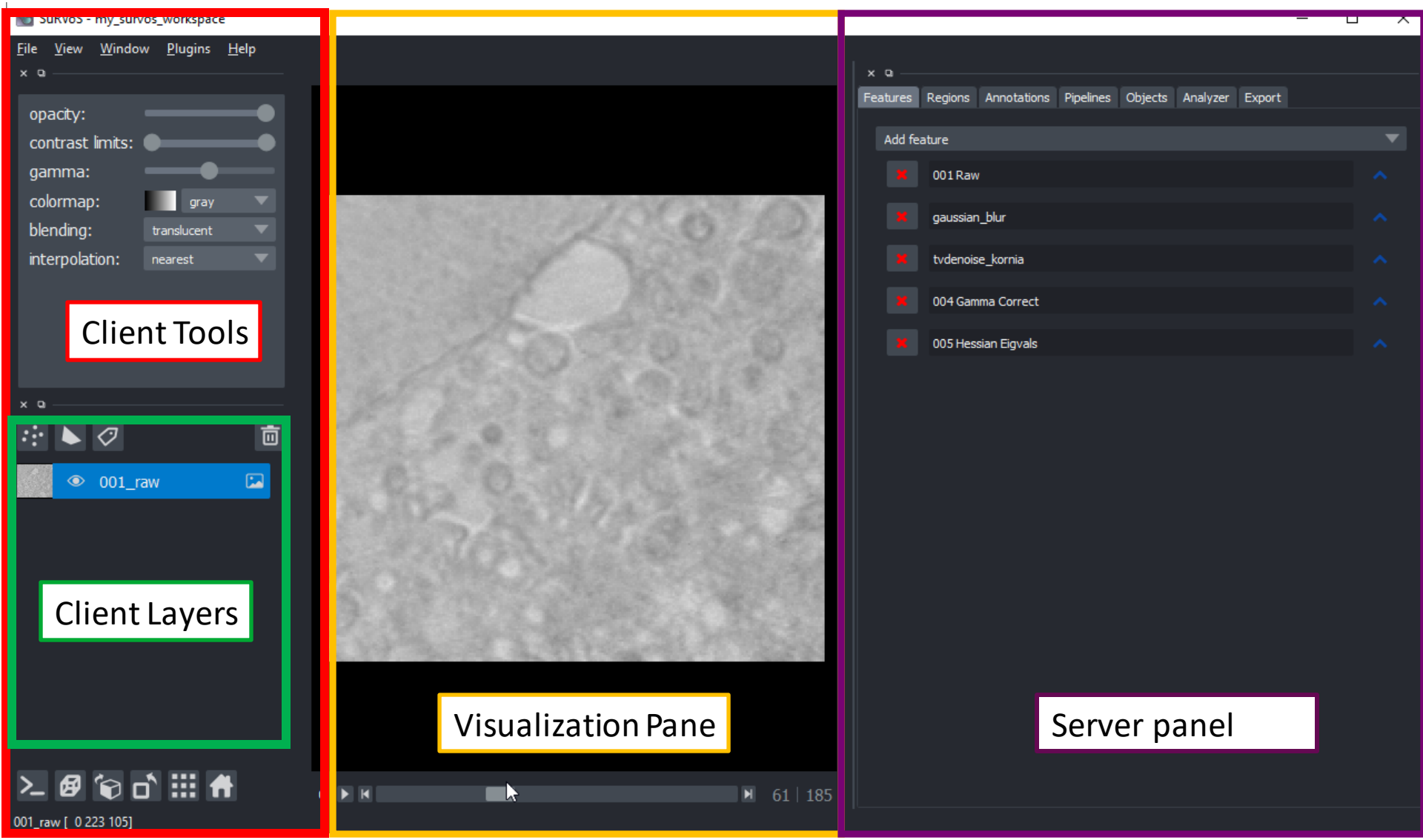
1. Select workspace name from dropdown (will be filled in automatically if workspace has just been created).

2. Click advanced. Check that SSH settings are off (in most cases).

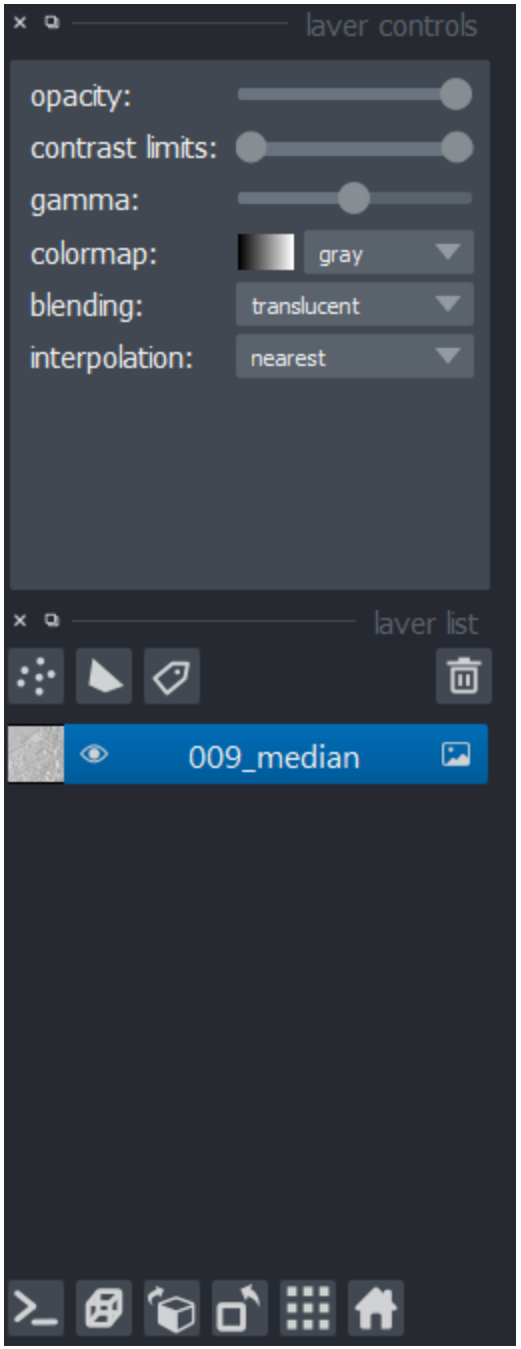
3. Click ‘Start Server’

If an existing program is on a given port then there will be an error. Change the port number and try again.

Orientation to SuRVoS



Layer controls and Layers in GUI



The left side of SuRVoS is the Napari volumetric data viewer. More information can be found at napari.org/docs.

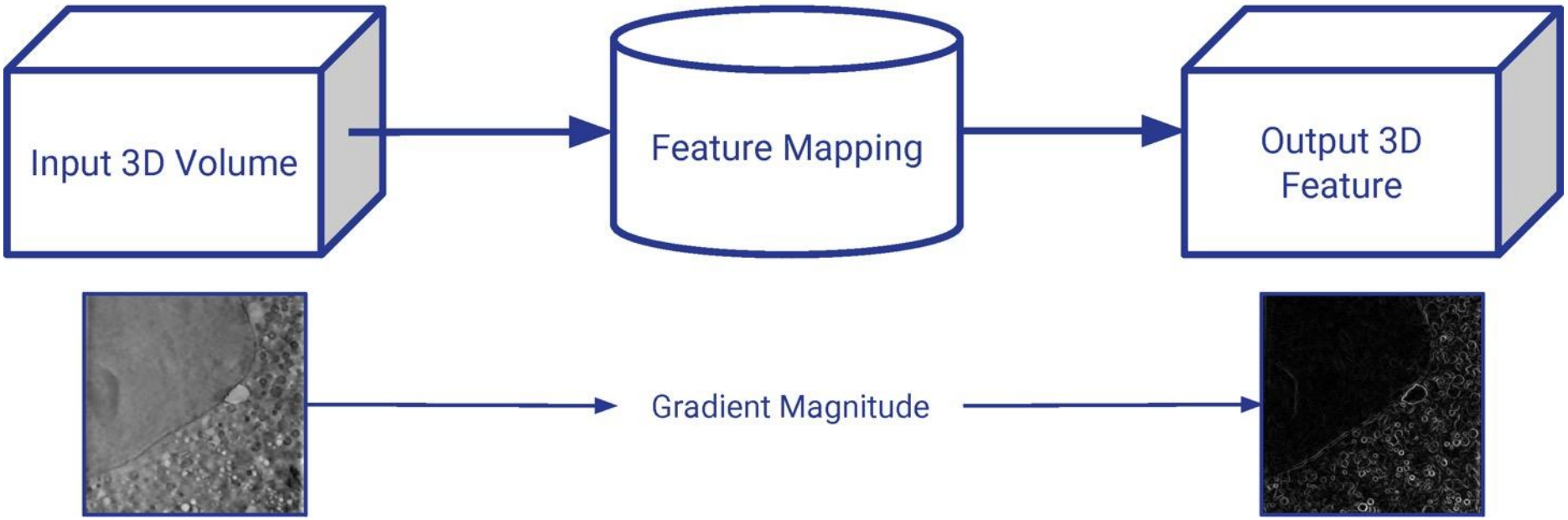
SuRVoS supports 3 Napari layer types
1. Images (with a mountain and sun picture icon) correspond to SuRVoS Features.
2. Labels (With a label icon) correspond to SuRVoS Annotations
3. Points (Points icon) are supported in SuRVoS Objects

Each layer type has a specific set of controls. Left are the controls for images, including opacity, contrast limits, gamma and colormap settings.

The bottom row of icons supports 3d viewing of The image volume (the cube icon)

Data Preprocessing

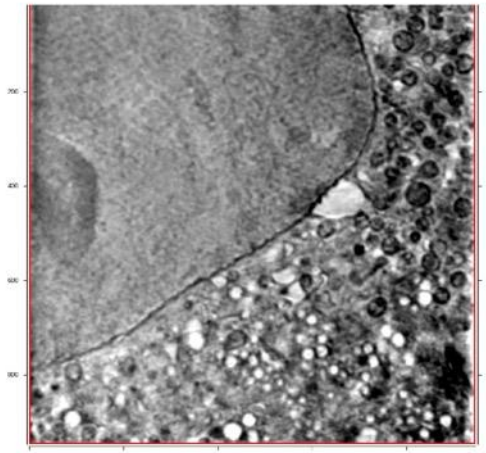
- Every preprocessing method outputs a *feature channel*
- *Feature channels* are obtained by modifying each pixel according to a function applied to their neighbourhood.
- *Feature channels* are volumes of the same size as the input volume
- *Feature channels* can be visualized inside **SuRVoS**



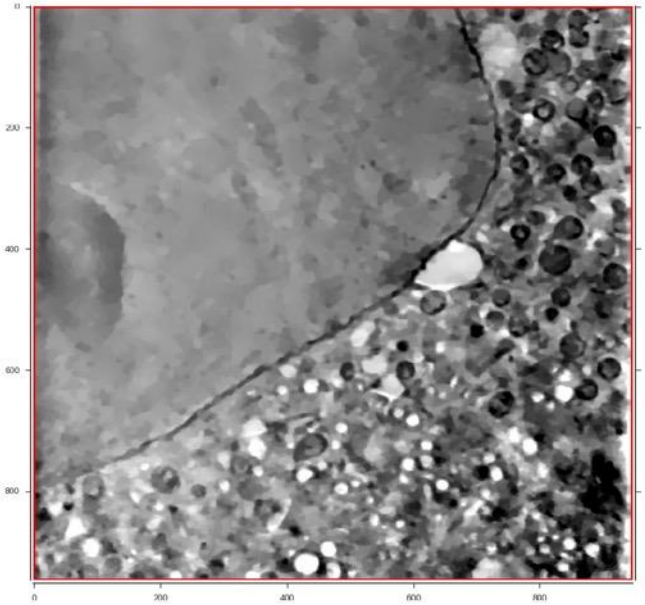
Data Preprocessing

Denoising

Original Image



Gaussian Smooth

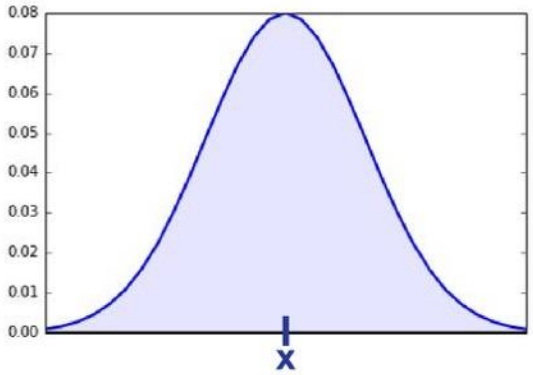


Total Variation

- Over-smooth
- Preserve Strong Edges
- Easier to identify objects

Data Preprocessing

Gaussian filters



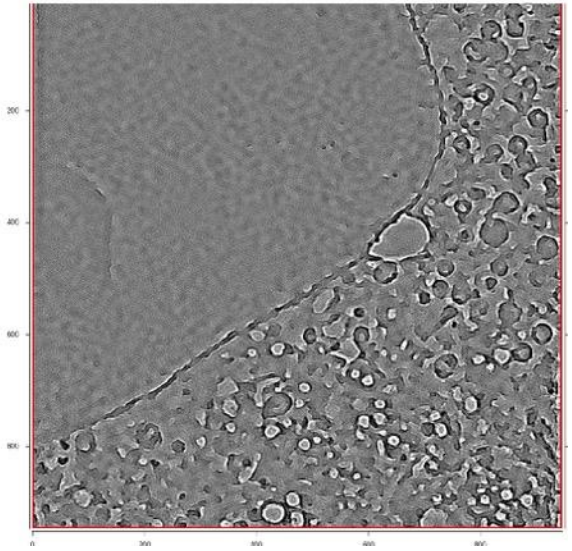
1D Gaussian:

Pixels near the center have more importance.

- A Gaussian neighbourhood of size N x N x N centered on every pixel
- Better data fidelity.

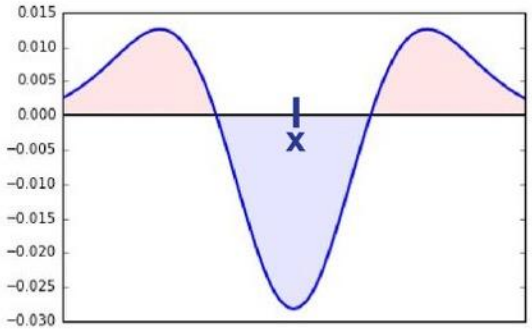
Mean Subtraction

Original Image



Data Preprocessing

Edge filters



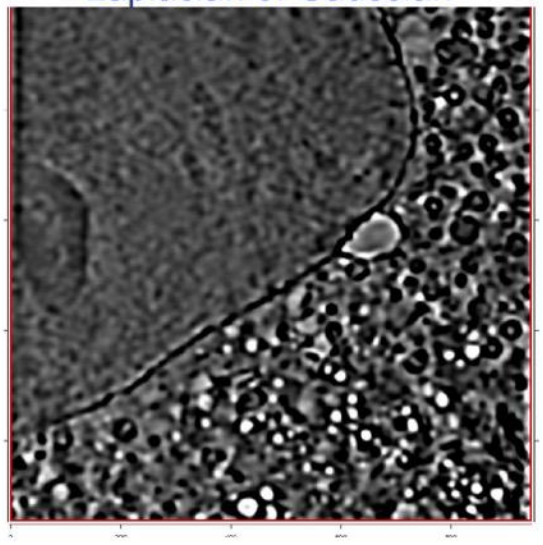
1D Laplacian of Gaussian:

Intensity near the center is substracted to the surroundings: (*x = red - blue*).

- A Laplacian neighbourhood of size N x N x N centered
- Identify objects brighter or darker than their surroundings.

Laplacian of Gaussian

Original Image



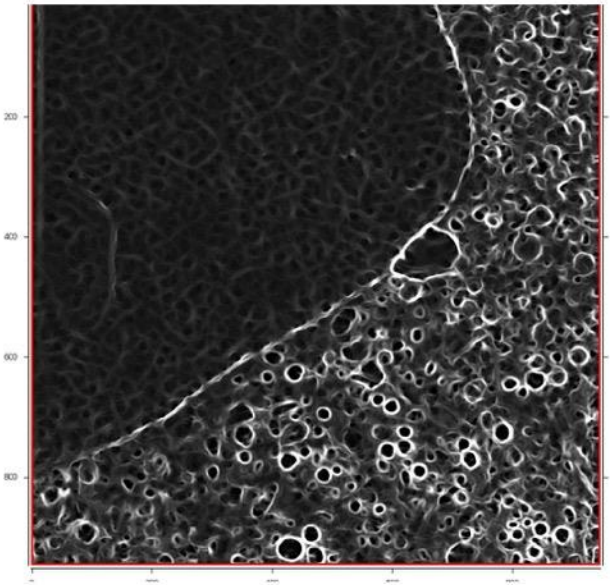
Data Preprocessing

Blob filters

Projects the data to analyze its main axis of variance

- Hessian Eigenvalues: texture
- Structure Tensor Eigenvalues: structure

Original Image



Largest Eigenvalue of the Hessian Matrix

Available Feature Filters

- Raw
  - Threshold
  - Invert
- Denoising
  - Gaussian Filter
  - Total Variation Filter
  - Median
- Gaussian Features
  - Gaussian Normalization
  - Gaussian Centre
  - Spatial Gradient
- Blob Detection
  - Difference of Gaussian
  - Laplacian of Gaussian
  - Determinant of Structure Tensor
- Texture and Structure
  - Hessian Eigenvalues

Available Filter and Feature Algorithms

- **Raw**

  Binary selection of data

- **Denoising**

  Removes noise

- **Gaussian Features**

  Uses info from neighboring voxels based on a Gaussian neighborhood

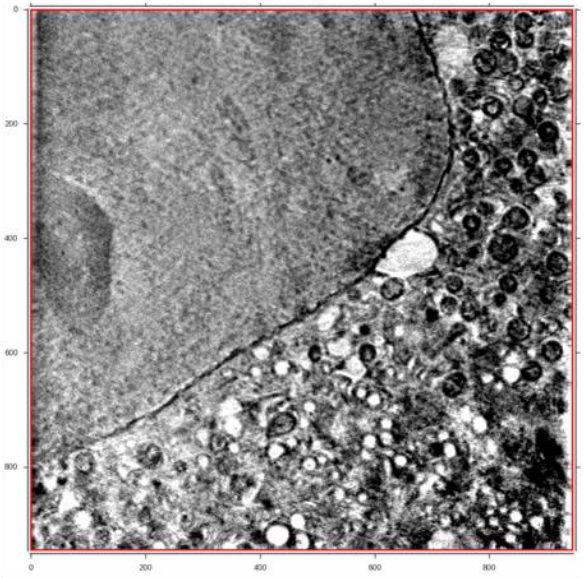
- **Blob Detection**

  Uses info from neighboring voxels based on a Laplacian neighborhood

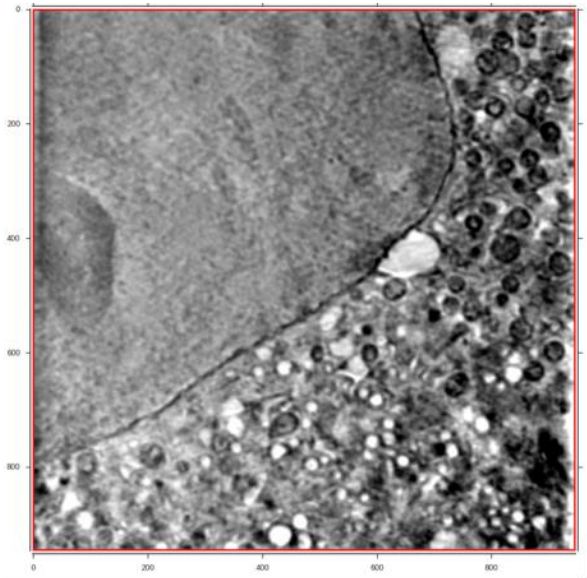
- **Texture and Structure**

  Highlights textural differences

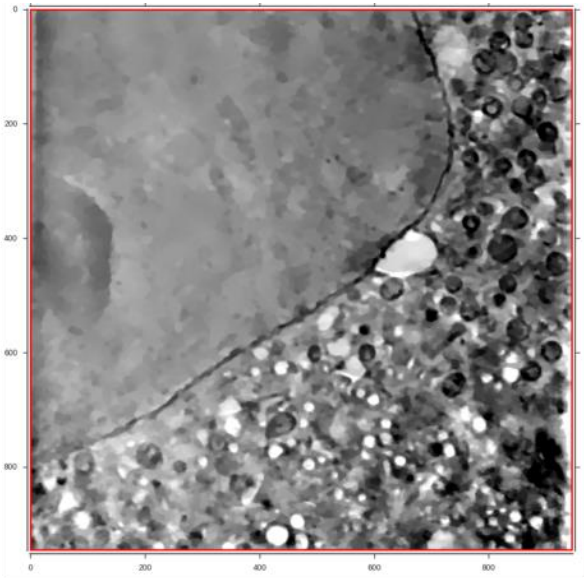
Available Filter Algorithms



(a) Raw SIRT reconstruction



(b) Gaussian Smooth



(c) Total Variation

Recommend

Sigma: 2.0 (range 1-5)

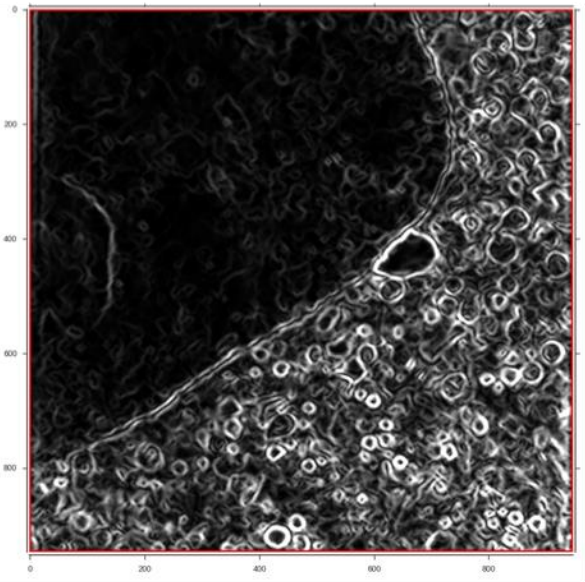
Recommend

Lambda: 10 (range 1-15)
(lower more denoising)

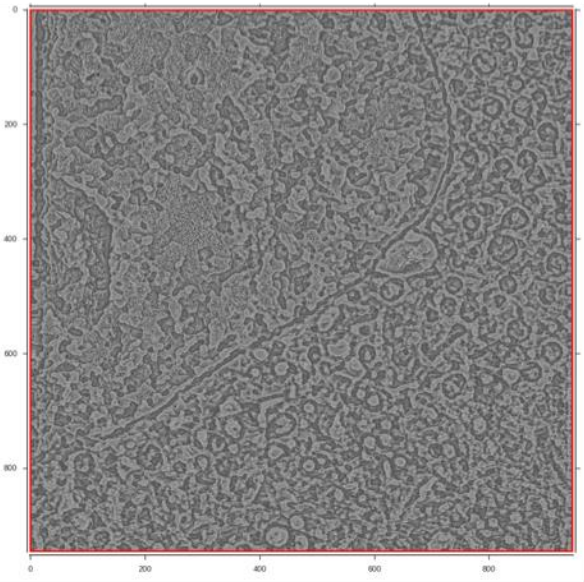
Maxiter: 100 (range 50-500)
(more iterations more denoising)

Note: Recommend to use Gaussian filter for supervoxel and megavoxel calculations, total variation for filter and feature calculations.

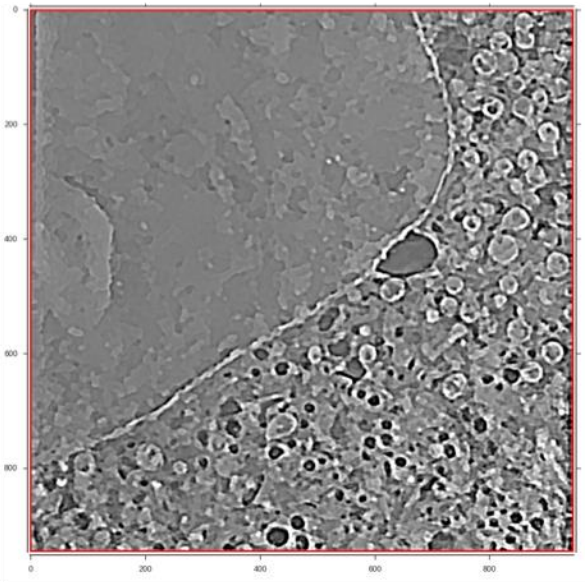
Selected Feature Filters



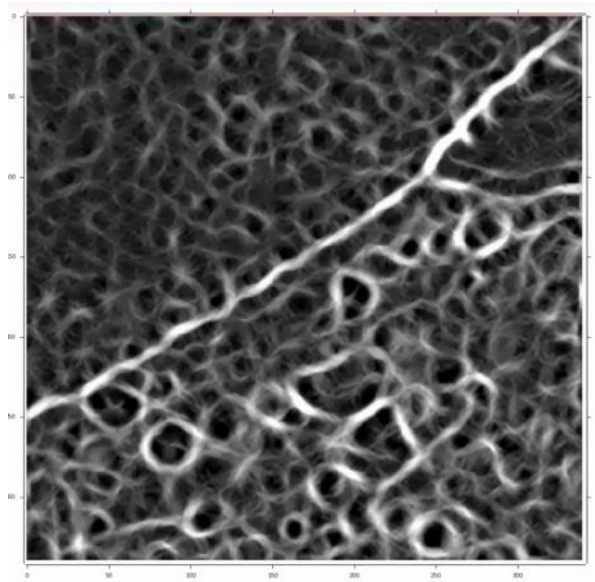
(d) Gradient Magnitude



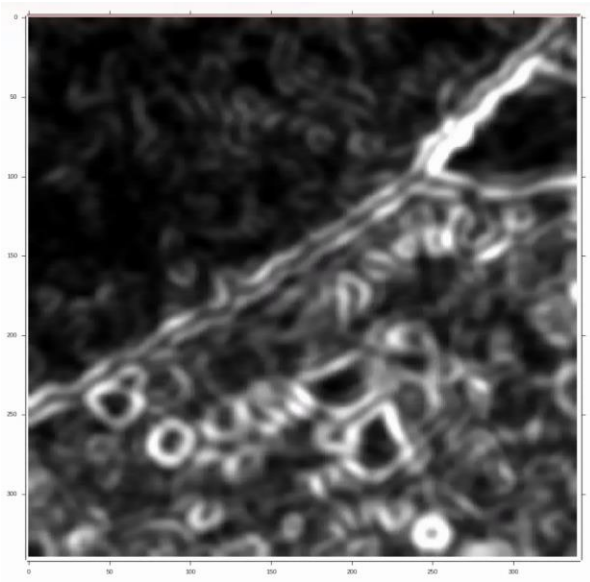
(e) Gaussian Local Normalization



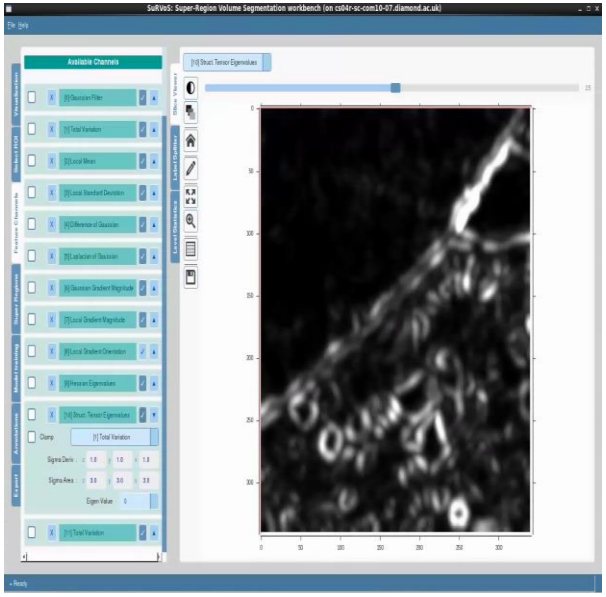
(f) Laplacian of Gaussian



(g) Hessian Eigenvalues



(h) Local Gradient Magnitude



(i) Structure Tensor Eigenvalues

The Feature Panel

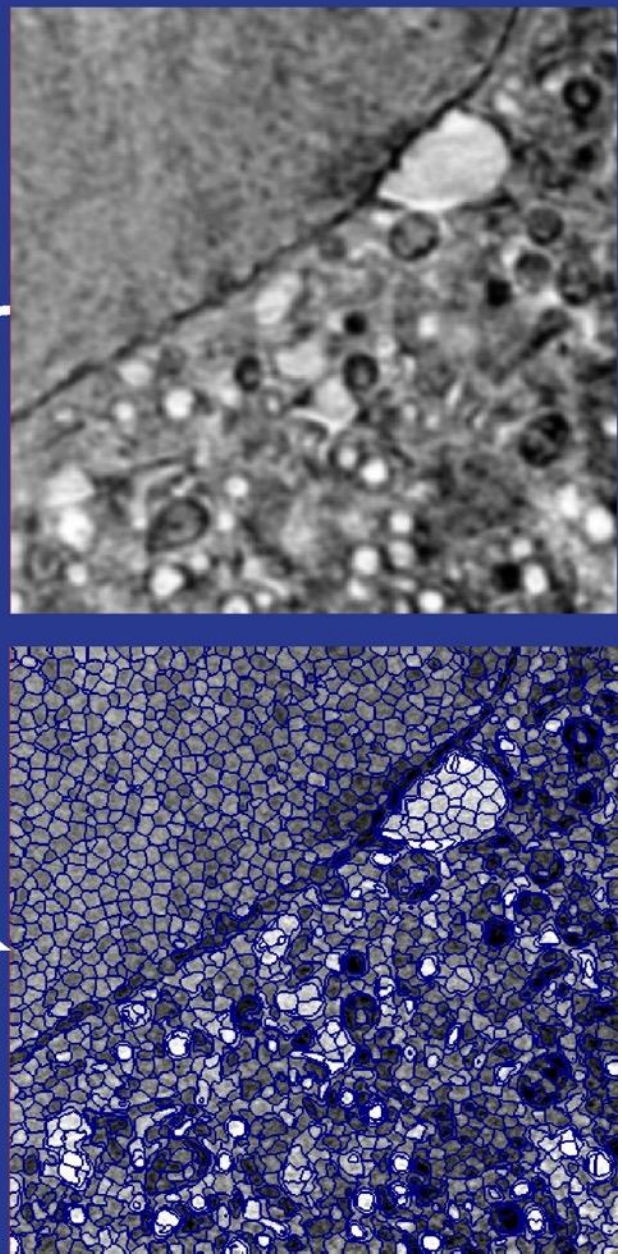
Select feature in dropdown
Click on name, or arrow to see details
Change input and algorithm values if needed
Click compute to run

Note: For filters that use them, coordinate order is Z, Y, X

Data Representation

Represent data in coherent regions

- Voxels
- SuperVoxels
- MegaVoxels



Data Representation

Num superpixels: 200

Original Image → Assign to each pixel the mean color of all the pixels that belong to that superpixel → **Reconstructed Image**

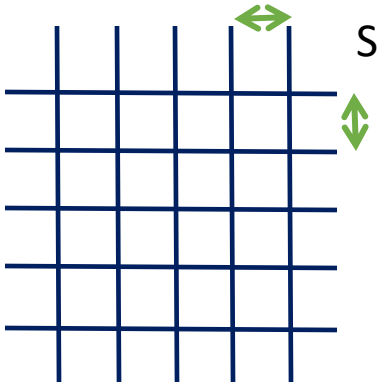
Superpixels extracted

Only 200 superpixels

Super Region Parameters: Supervoxels

SuperVoxels:

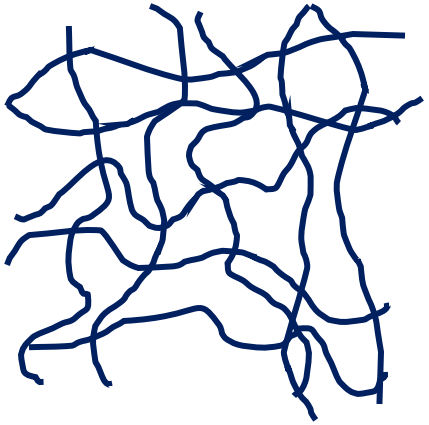
- SP shape
- Compactness



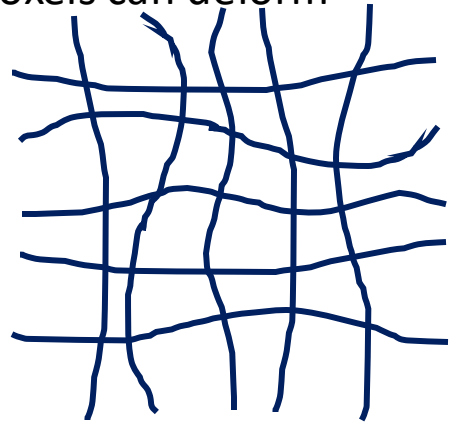
SP Shape: i.e. # of voxels to include in supervoxel grid

Note: order is Z, Y, X

Compactness: how much supervoxels can deform

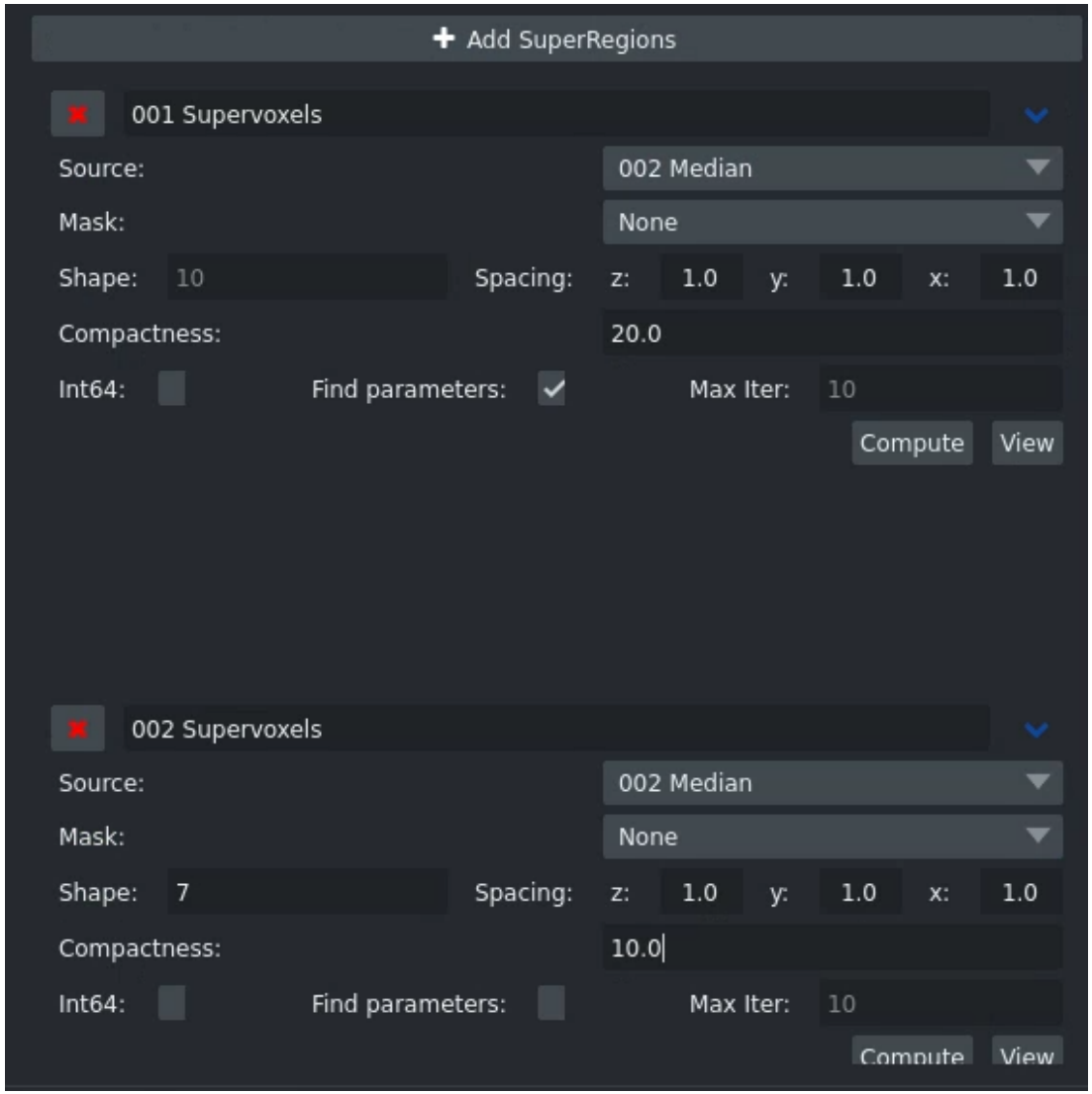


Low compactness #



High compactness #

Super Region Parameters: Recommendations



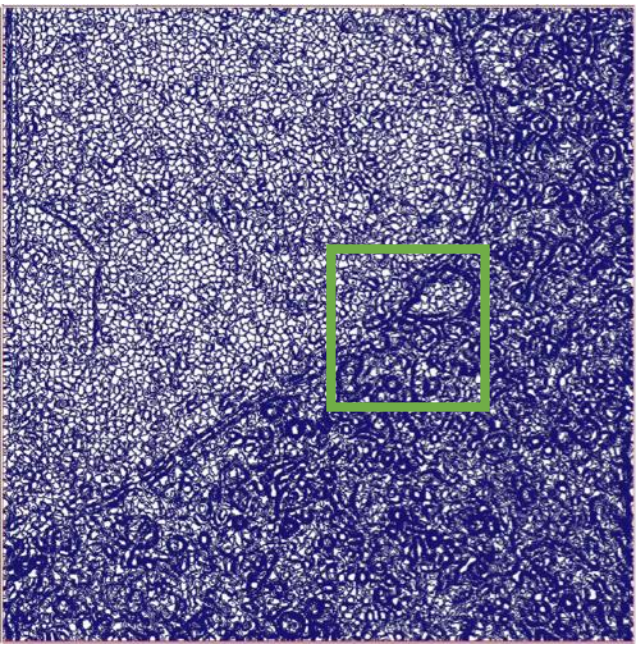
SuperVoxels:

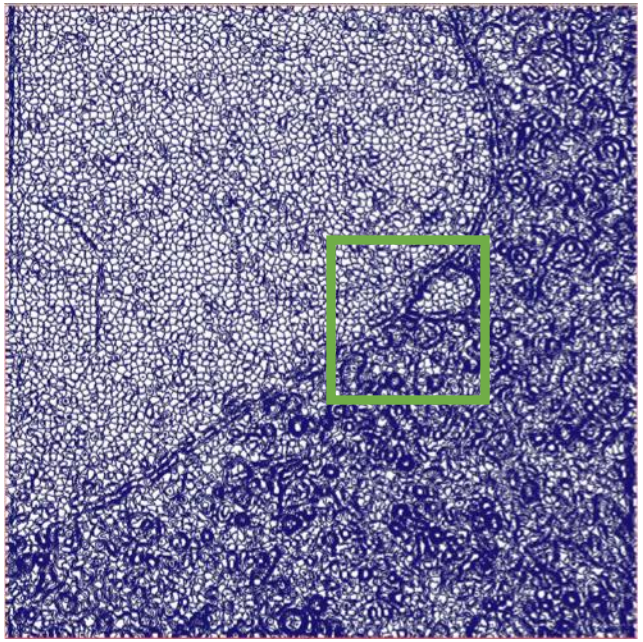
- Source: Data channel
- SP shape: Default 10
- Spacing: 1x1x1
- Compactness: 20 (range 5-100)
- Order is Z, Y, X
- A Compactness value of around 10-20 often follows obvious contours in the image closely while ignoring some detail and reducing noise.

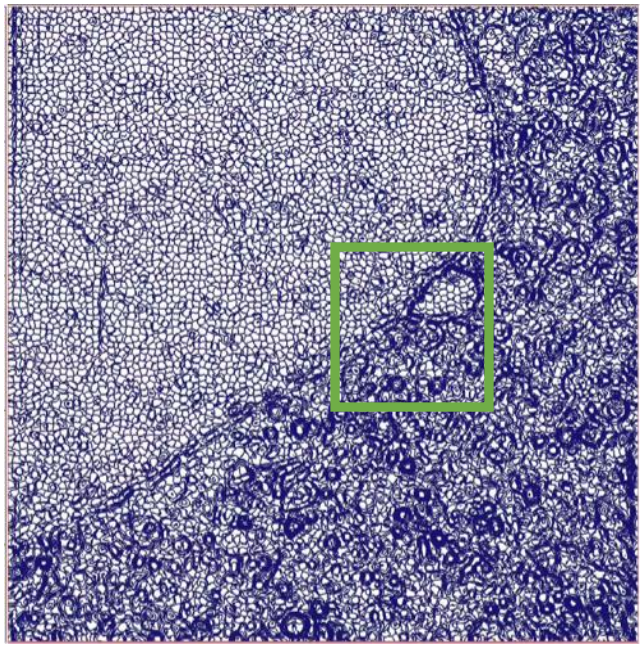
10x10x10 / 1x1x1 / **10**

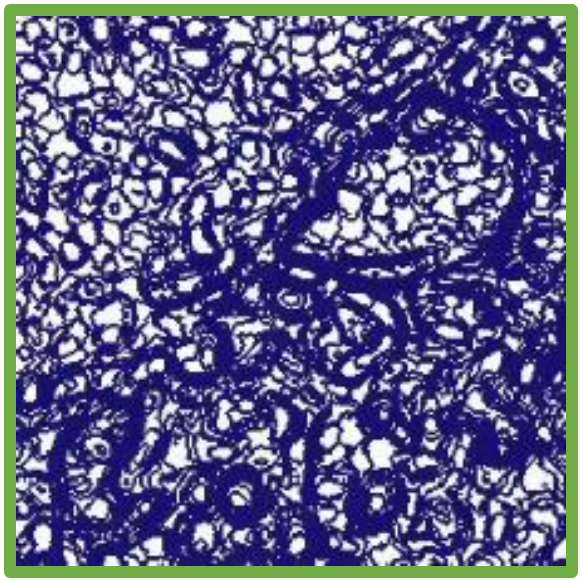
Default:
10x10x10 / 1x1x1 / 20

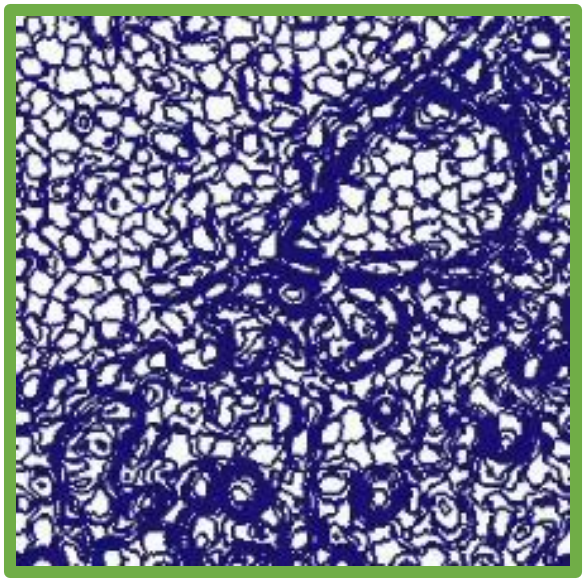
10x10x10 / 1x1x1 / **30**

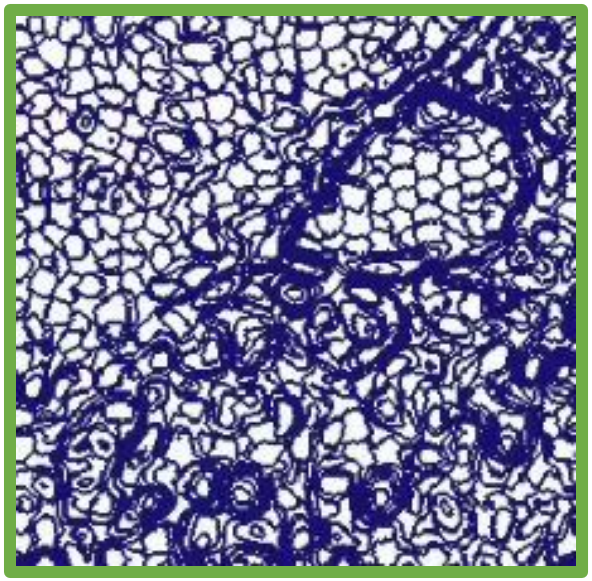








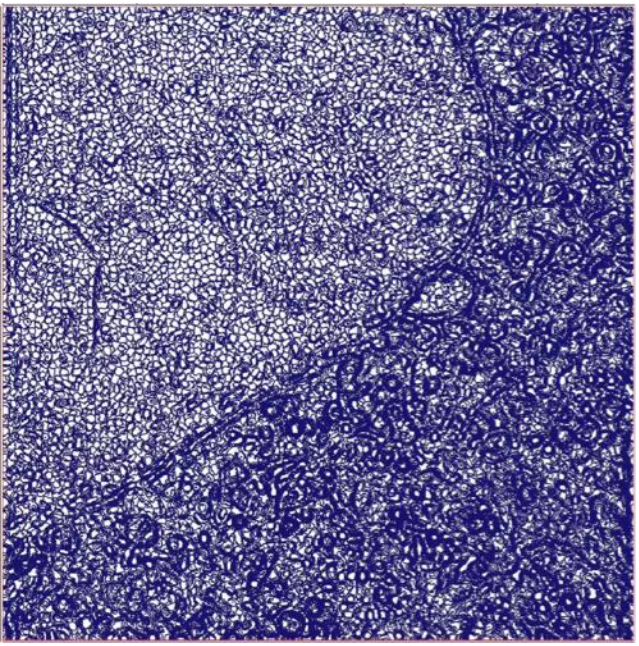
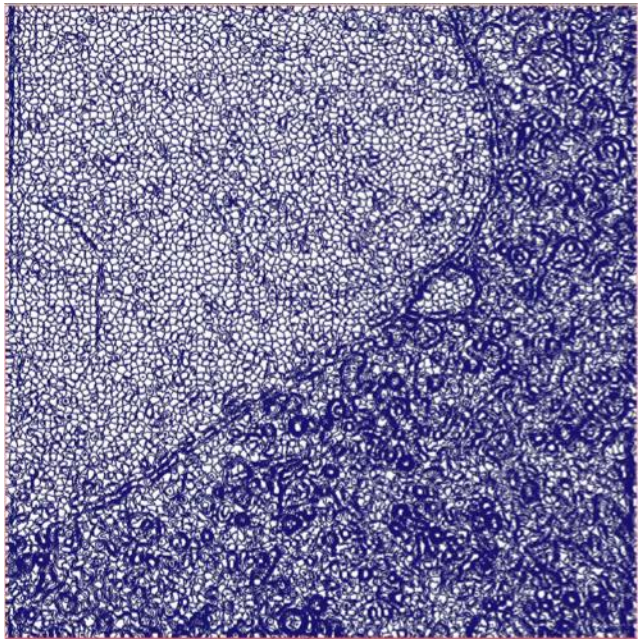
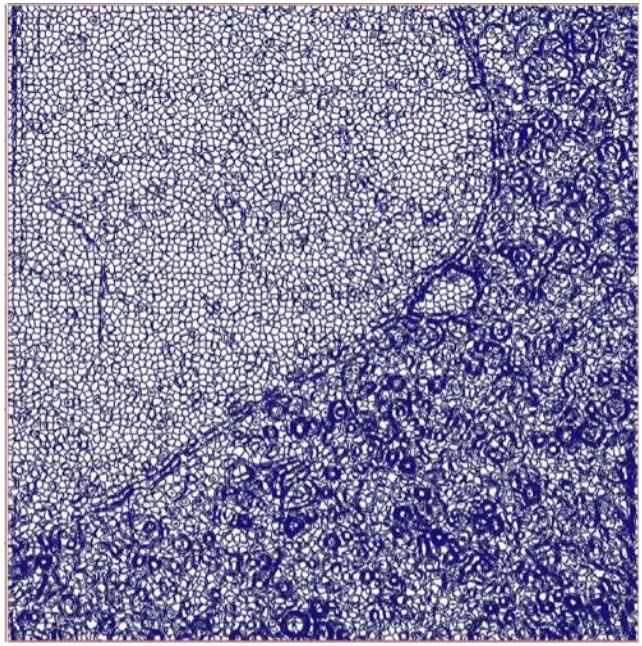
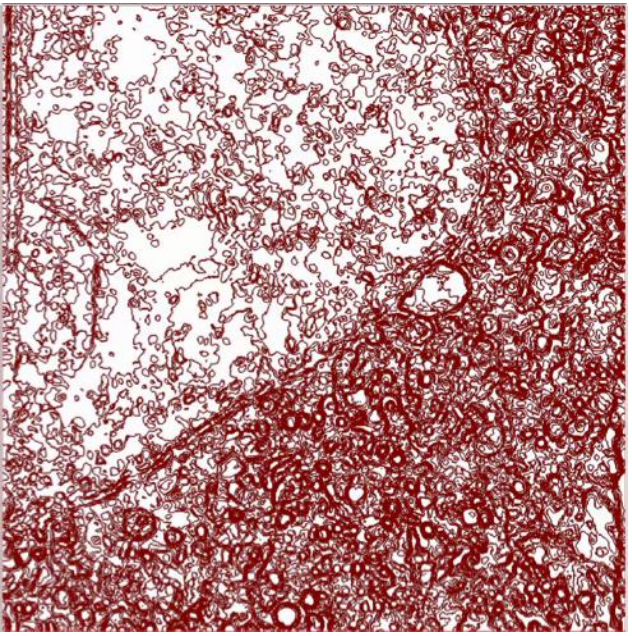
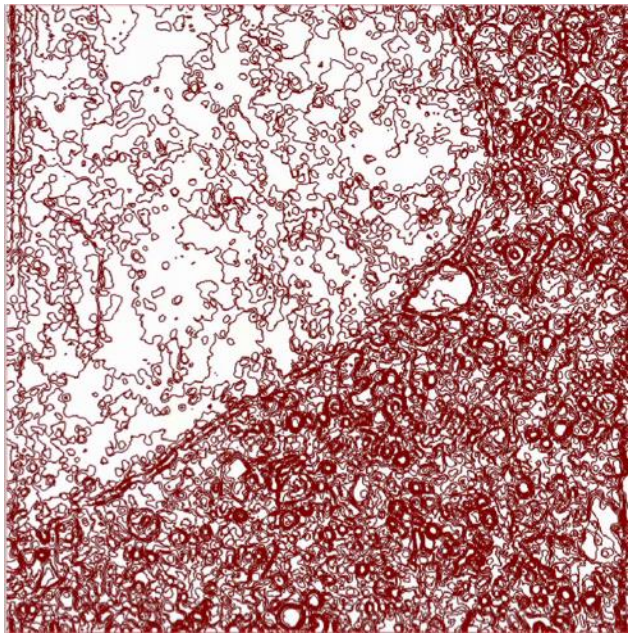
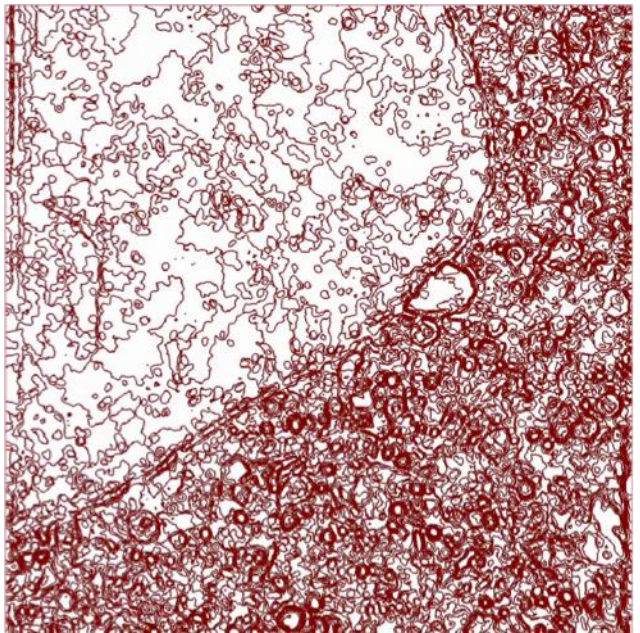




10x10x10 / 1x1x1 / **10**

Default:
10x10x10 / 1x1x1 / 20

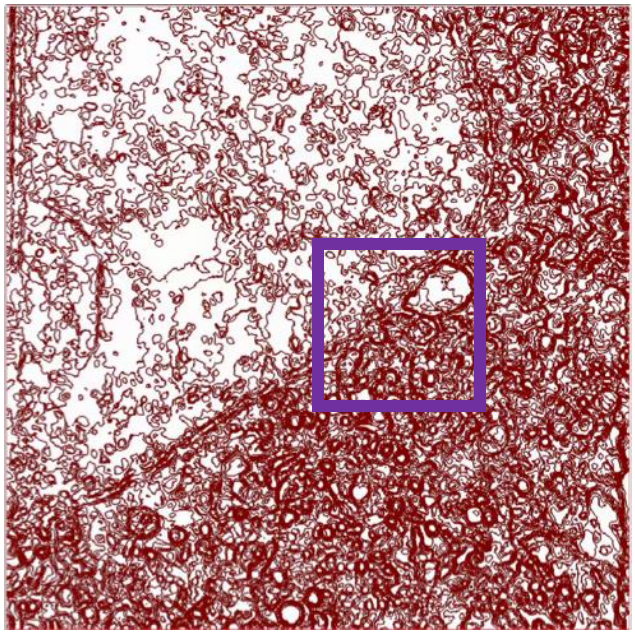
10x10x10 / 1x1x1 / **30**

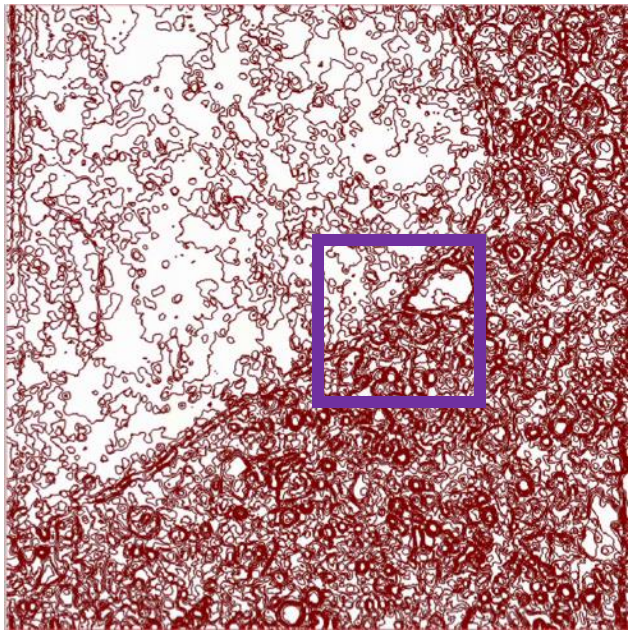







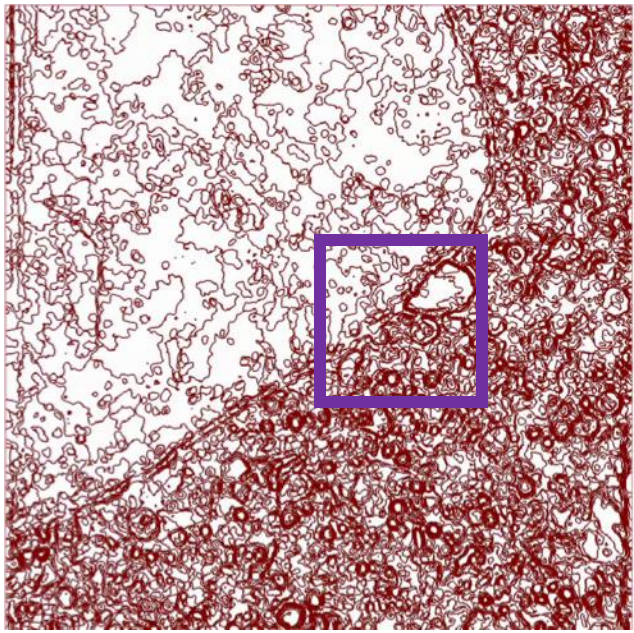
10x10x10 / 1x1x1 / **10**

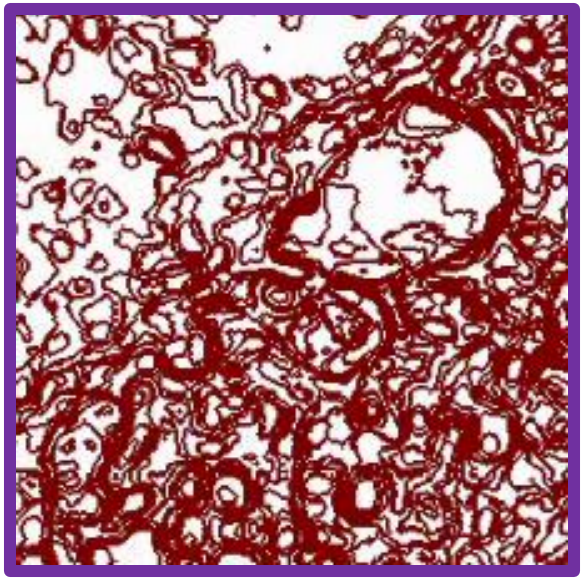
Default:
10x10x10 / 1x1x1 / 20

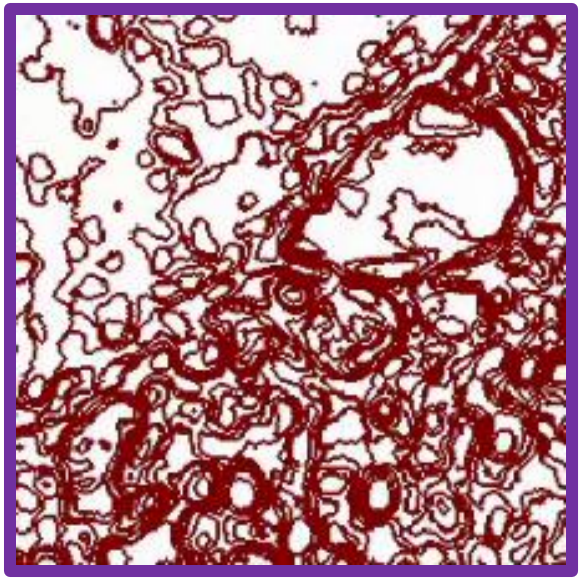
10x10x10 / 1x1x1 / **30**

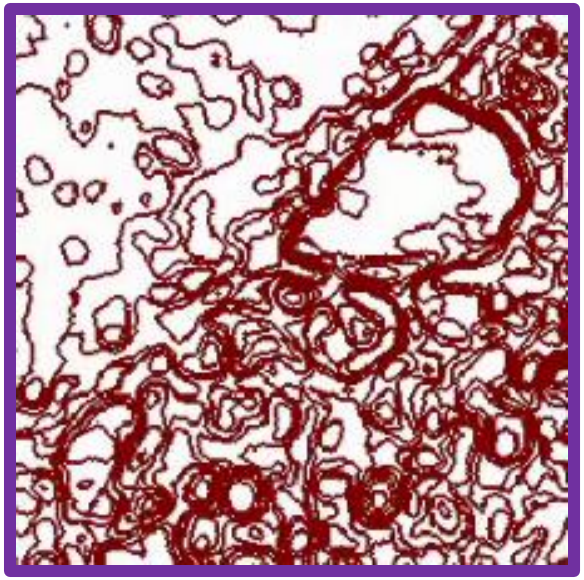










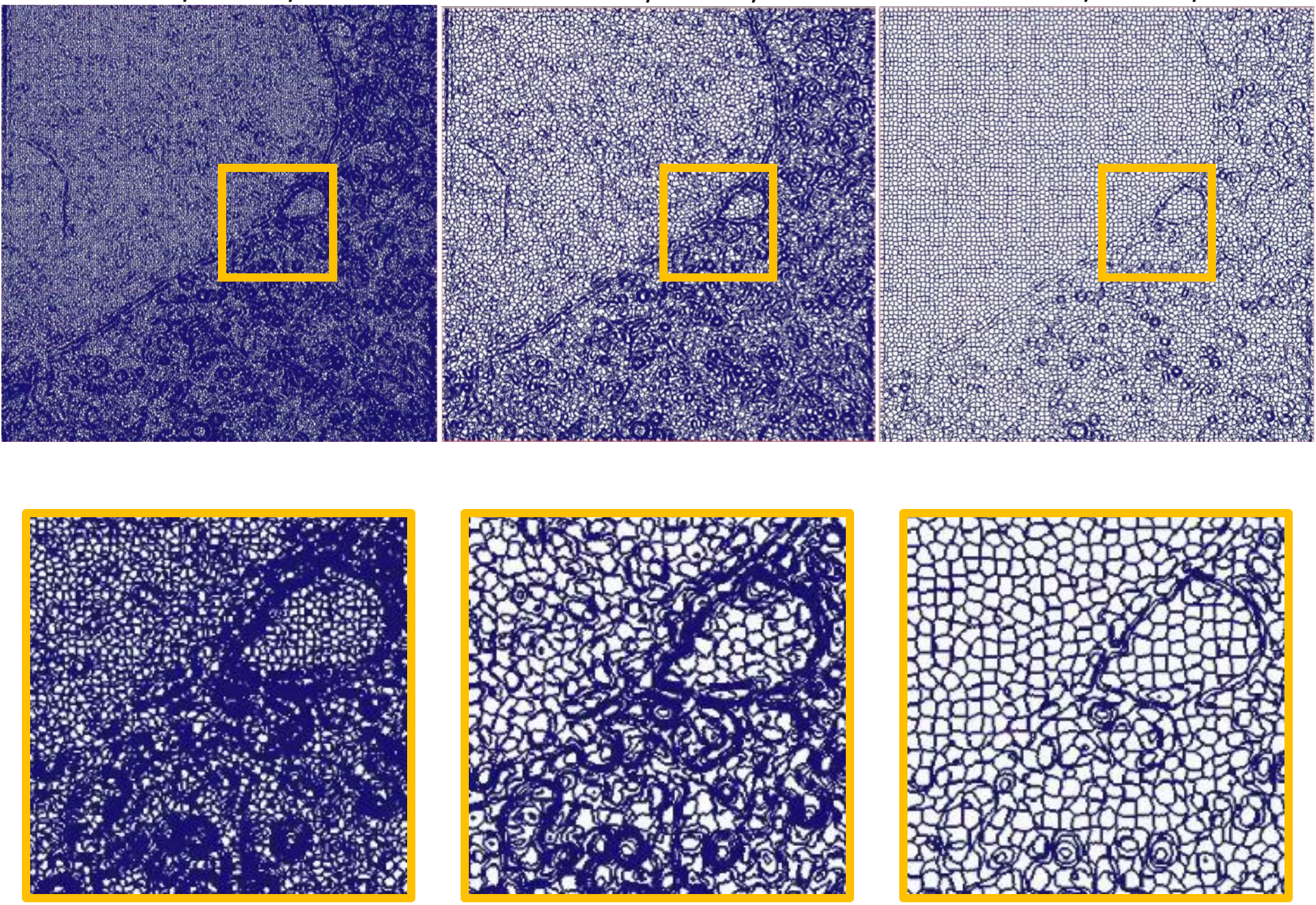


Default:

5x5x5 / 1x1x1 / 20

10x10x10 / 1x1x1 / 20

10x10x10 / **3x3x3** / 20

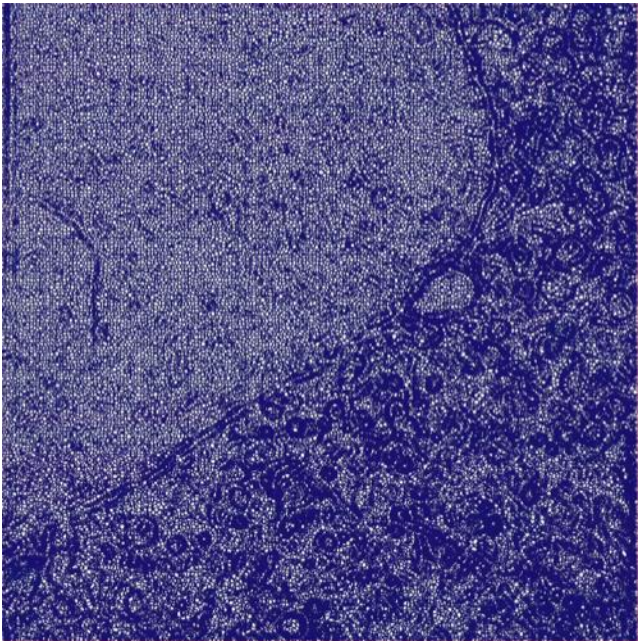


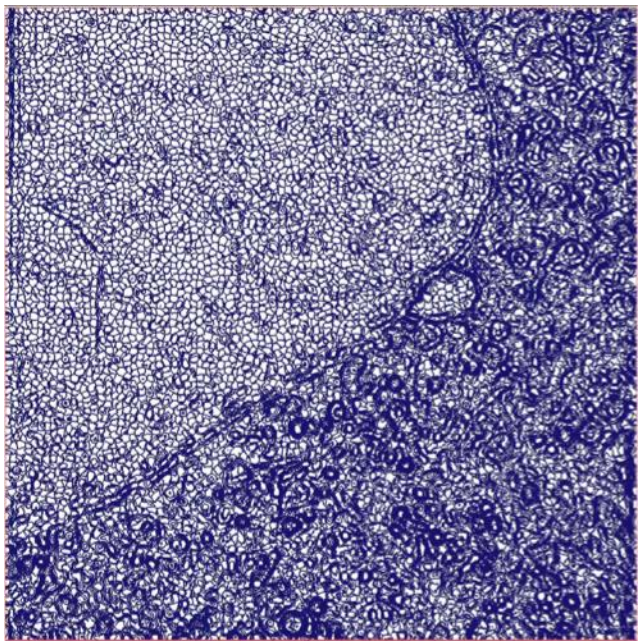
Default:

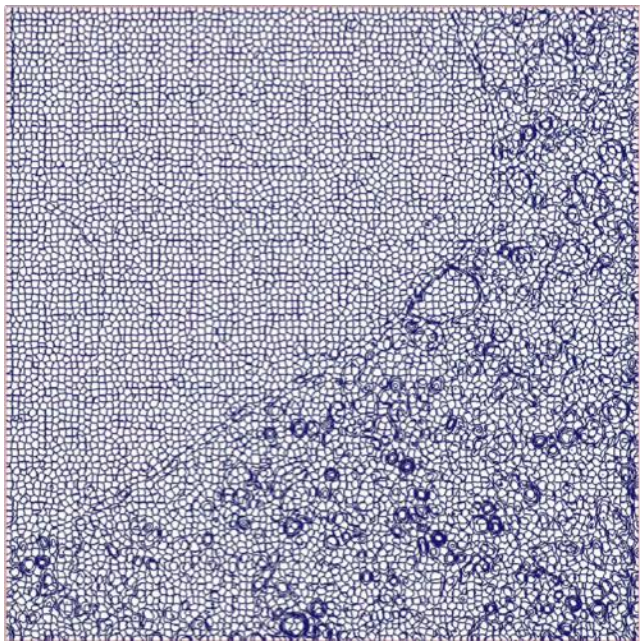
5x5x5 / 1x1x1 / 20

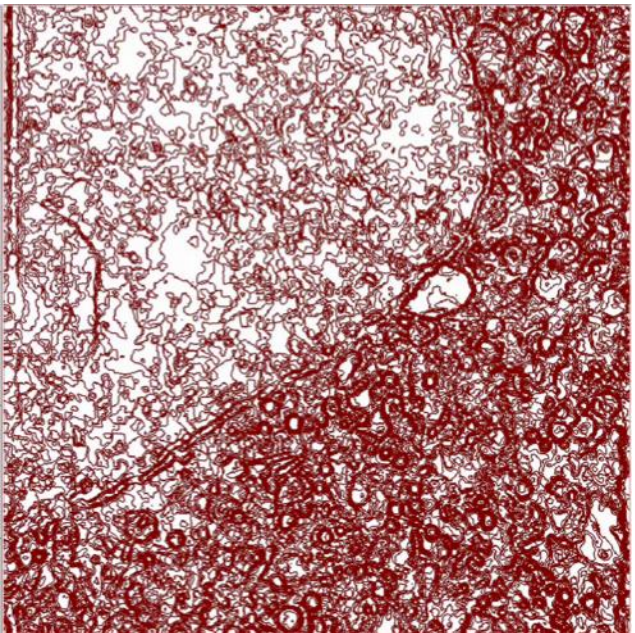
10x10x10 / 1x1x1 / 20

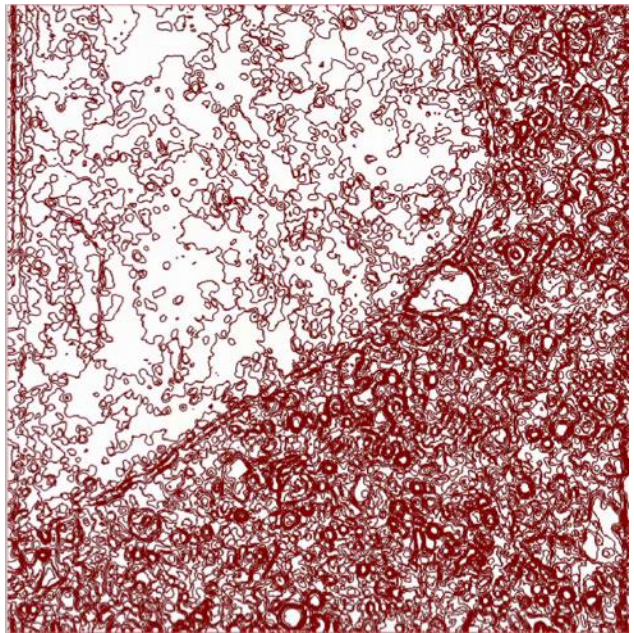
10x10x10 / **3x3x3** / 20

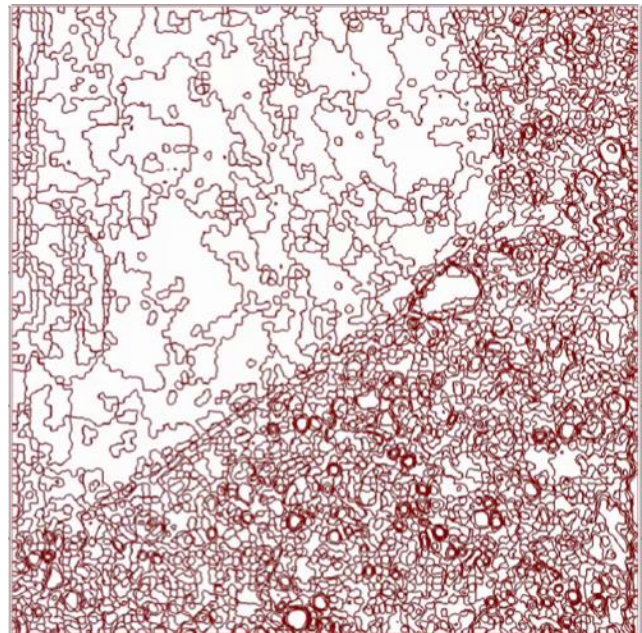








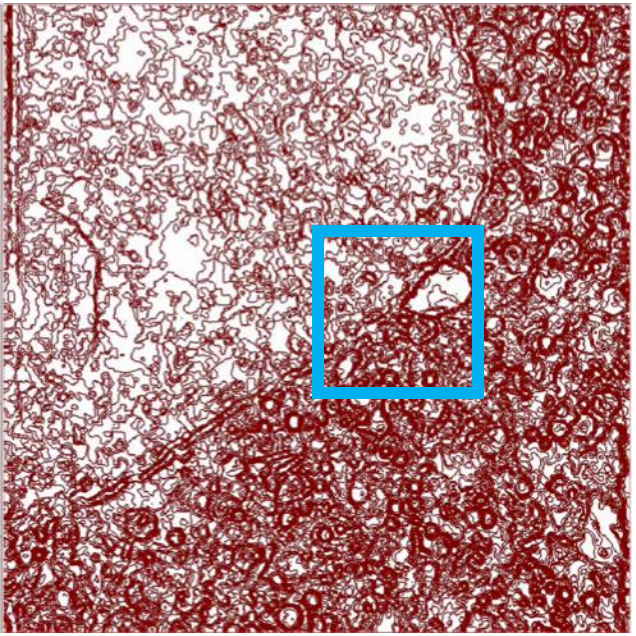


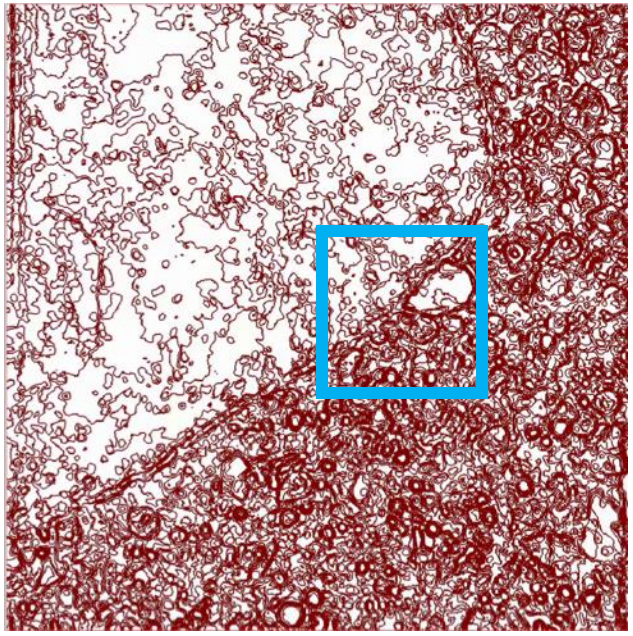


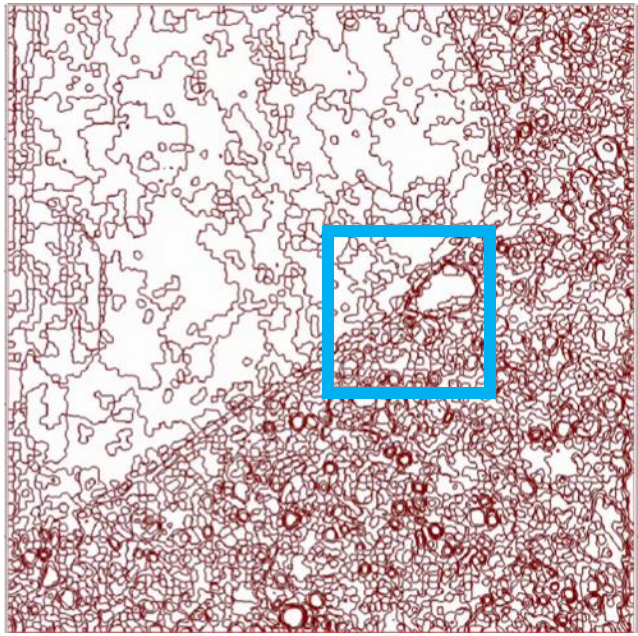
5x5x5 / 1x1x1 / 20

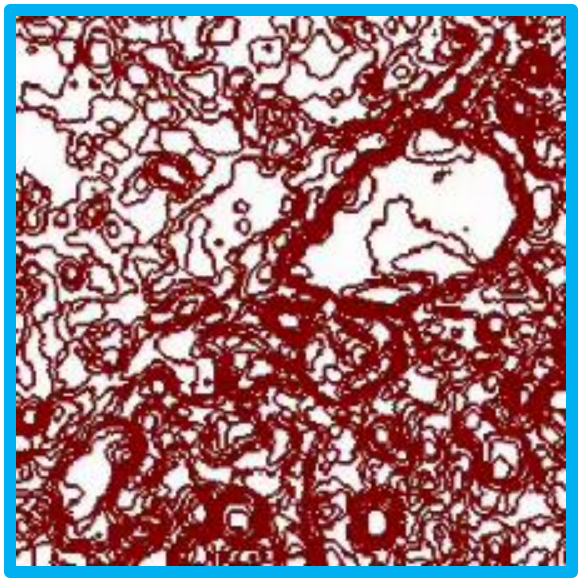
Default:
10x10x10 / 1x1x1 / 20

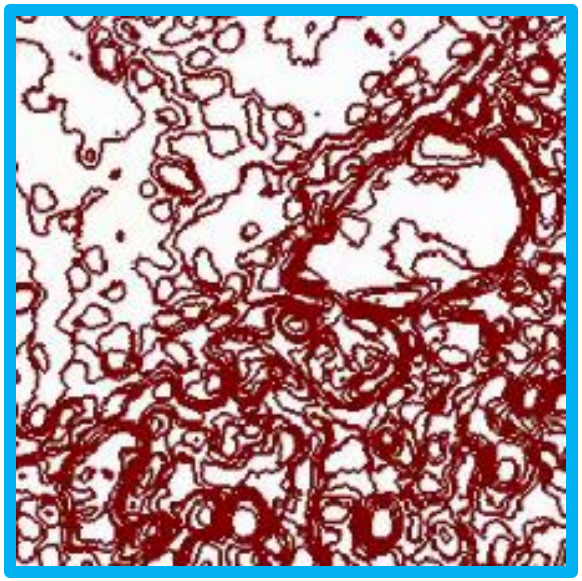
10x10x10 / **3x3x3** / 20

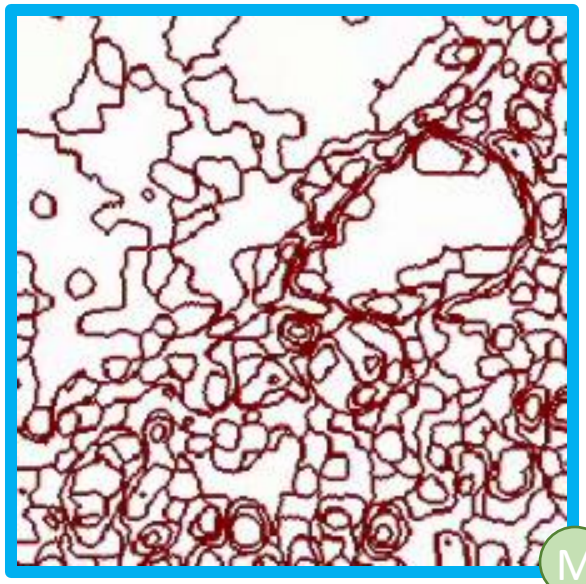




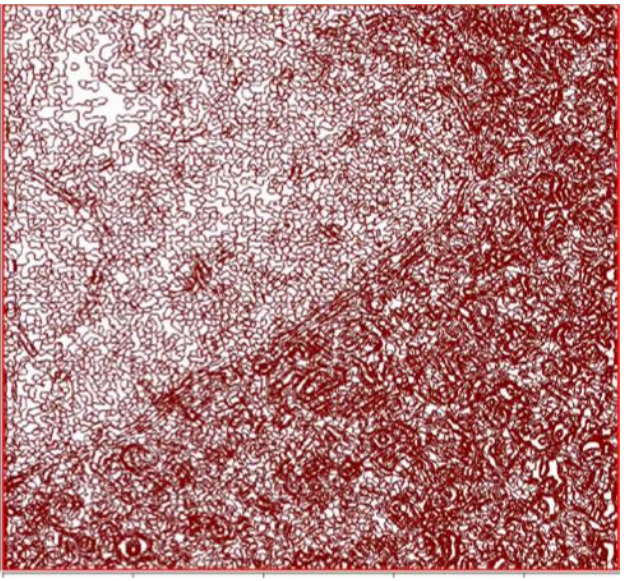




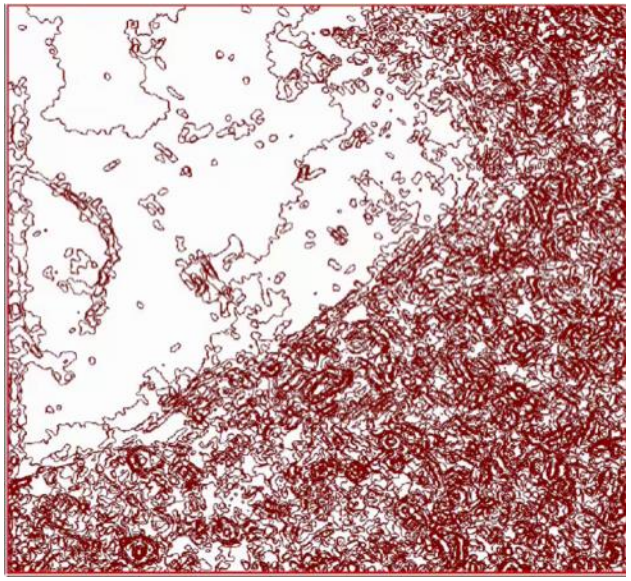




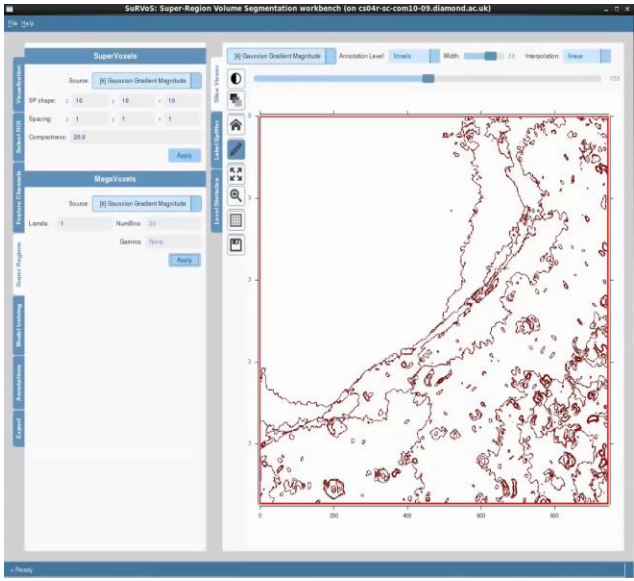
λ=0.01/NumBins=20



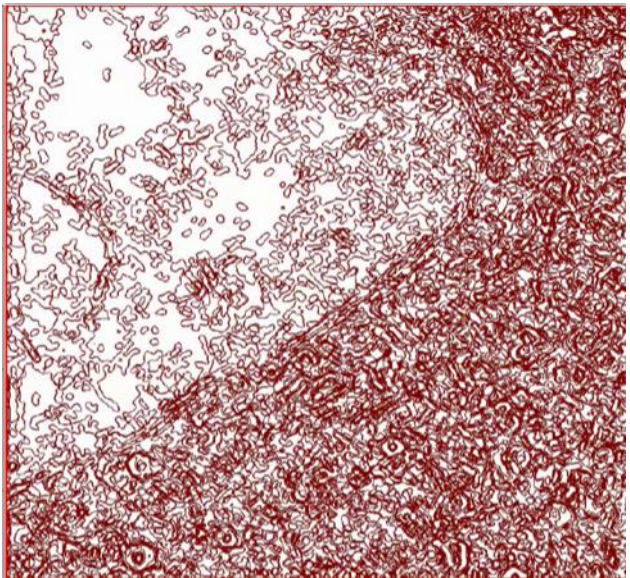
Default:
λ=0.1/NumBins=20



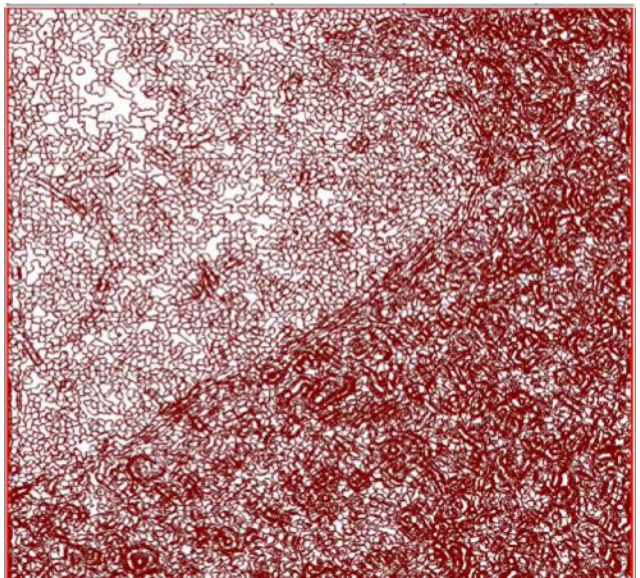
λ=1/NumBins=20



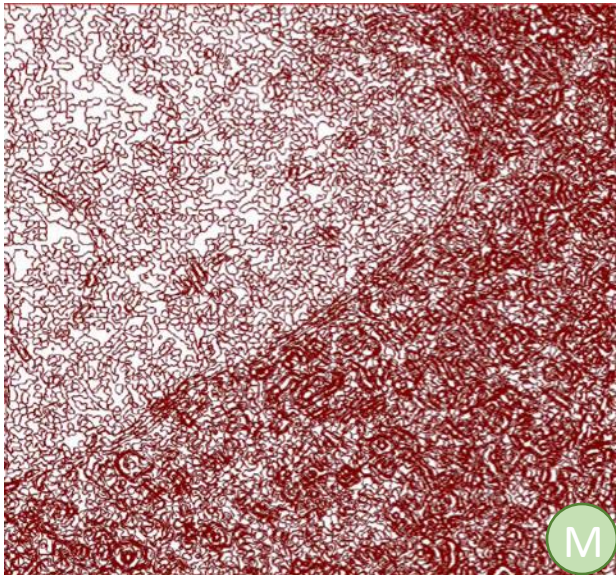
λ=0.1/**NumBins=10**



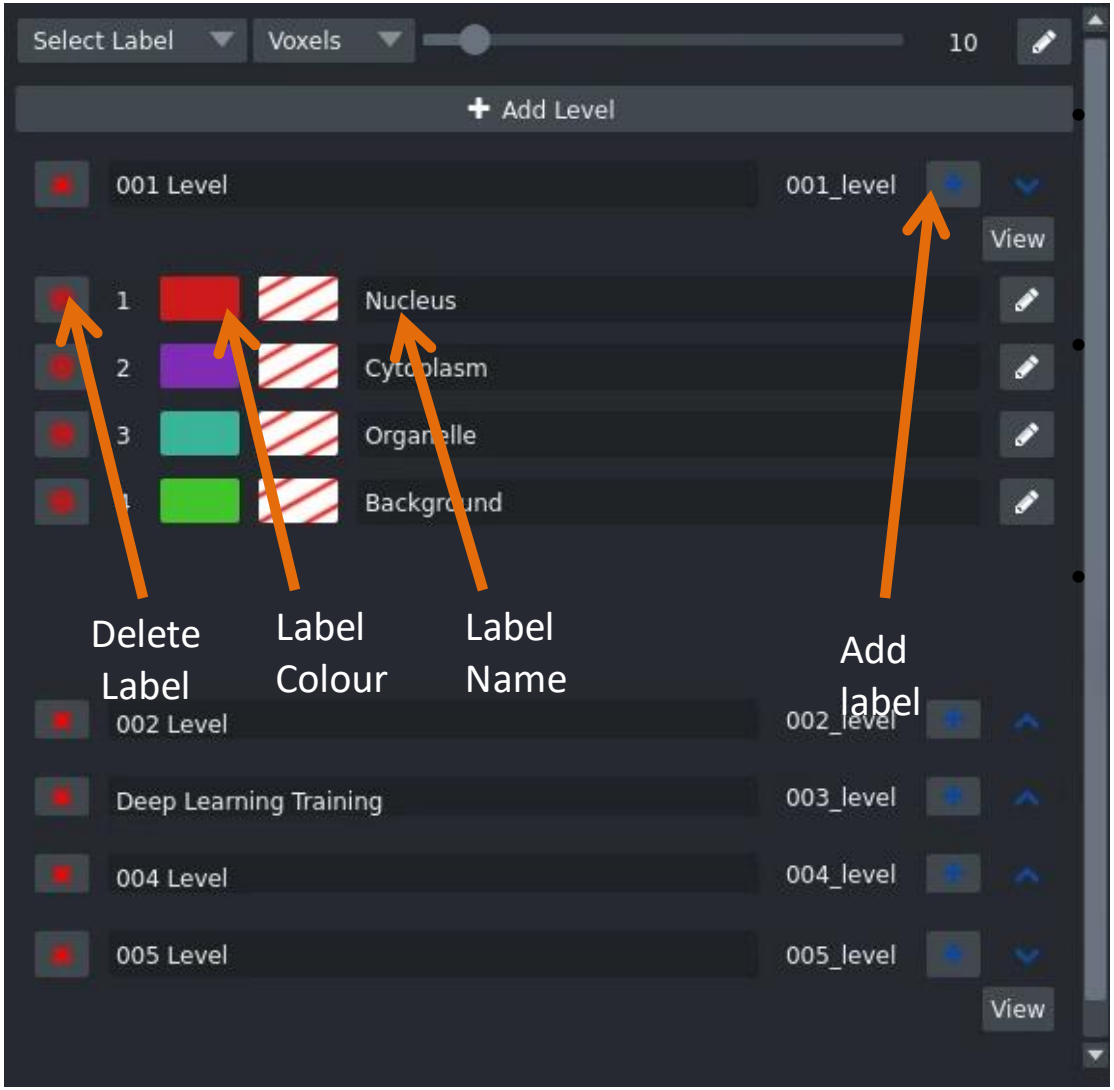
λ=0.1/**NumBins=30**



Defaults, **gamma=auto**



The Annotation Tab

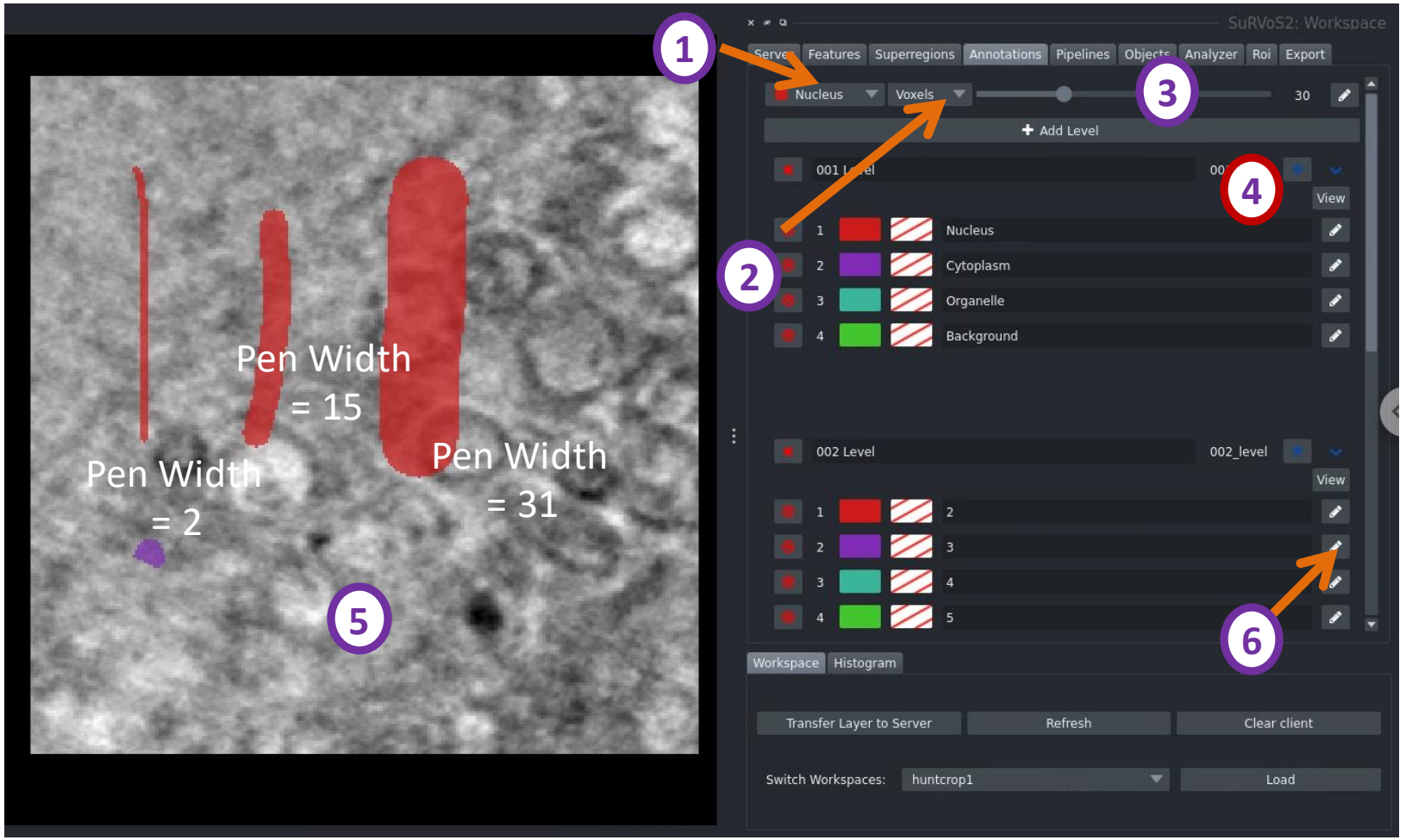


To add a Level click on Add Level. Levels are given a default name but can be renamed.

To add Labels to a Level, click Add Label. Multiple labels can be added to any Level.

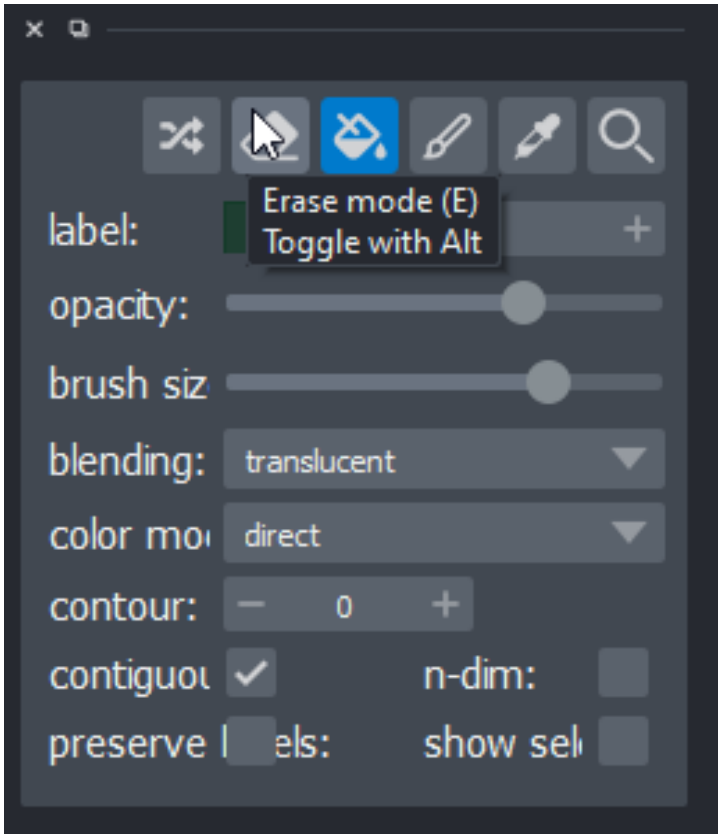
Labels can be given names, the colour used to represent them in annotations can be changed. Or if they are no longer needed, labels can be deleted.

Painting Voxels



1) In the annotation tab 'Select label' dropdown select the label you wish to annotate with.
2) Select annotation level (**voxel** or **supervoxel**)
3) Choose an appropriate brush width for the feature that you are annotating.
4) Press set to confirm.
5) Paint on the view.
6) When painting you can click on the pen icon by a label to change to that label.

Erasing Annotations



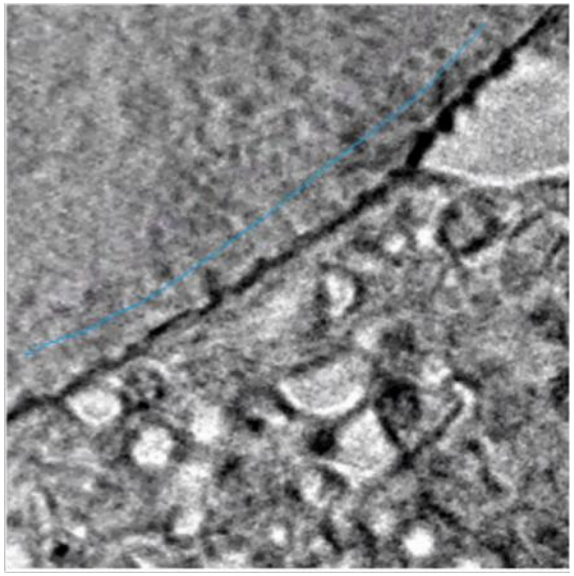
1) In the layer controls select the eraser tool.

2) Change the brush size.

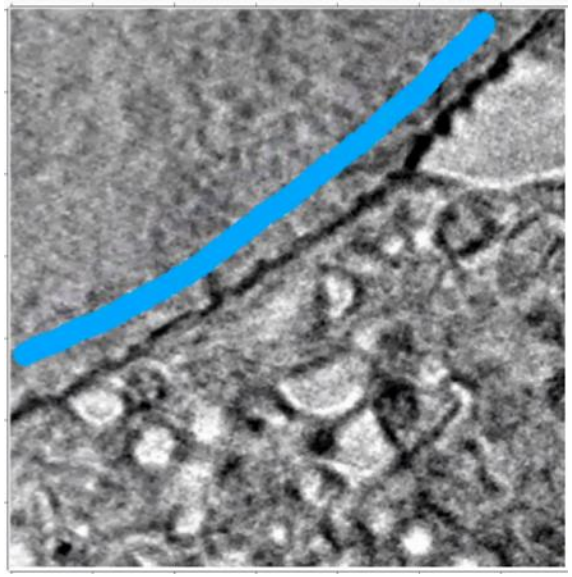
3) To begin annotating again, select the appropriate label and parameters in the annotation panel.

4) The fill tool doesn't work on SuRVoS annotation images, but the eye-dropper tool does work.

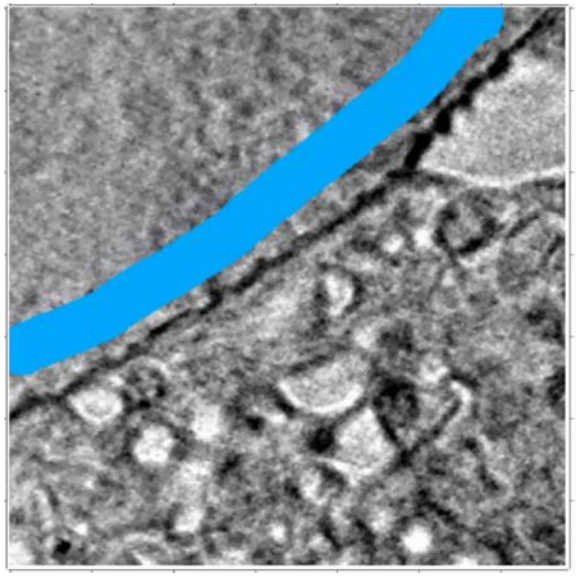
Annotation Using Voxels



Pen Width = 1



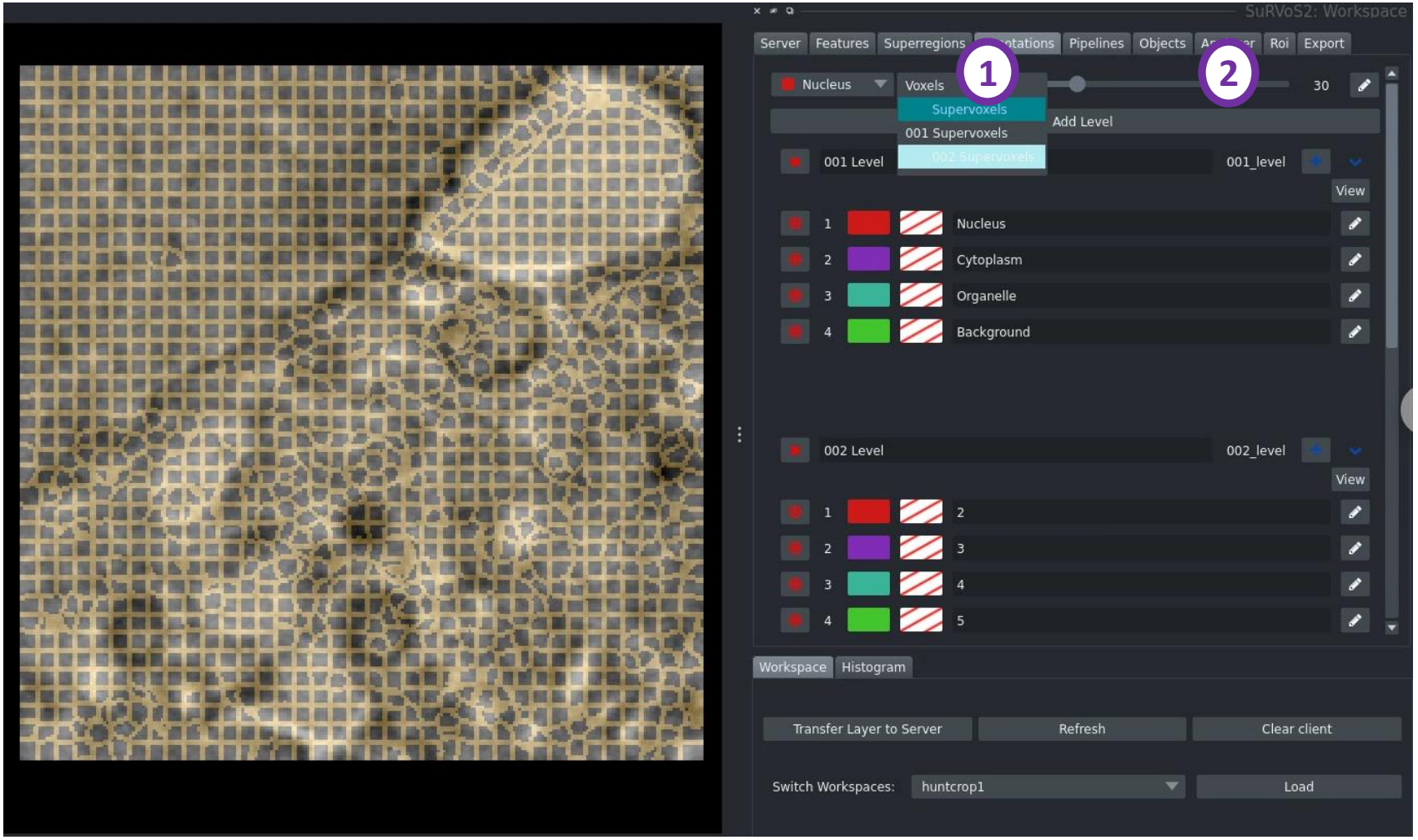
Pen Width = 15



Pen Width = 31

- Annotation using voxels can be done with different pen widths.
- Whilst annotating in voxels annotations do not penetrate the volume in Z

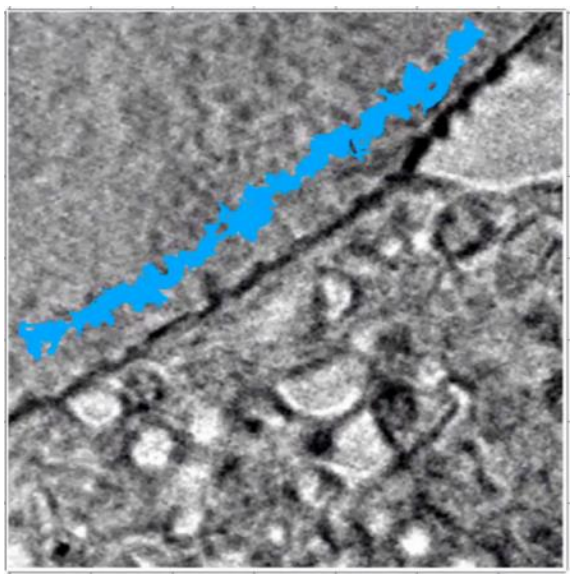
Annotation Using Supervoxels



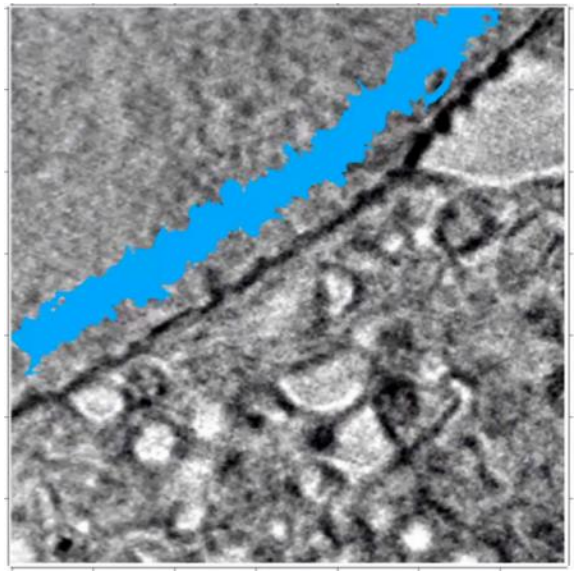
1) Select annotation level (voxel, **supervoxel**, or megavoxel)

2) Choose a width appropriate for the feature that you are annotating and the supervoxel size and draw using the left mouse button.

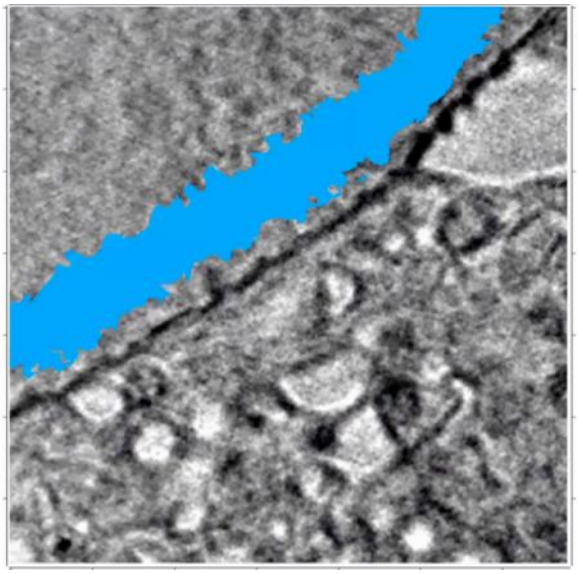
Annotation Using Supervoxels



Pen Width = 1



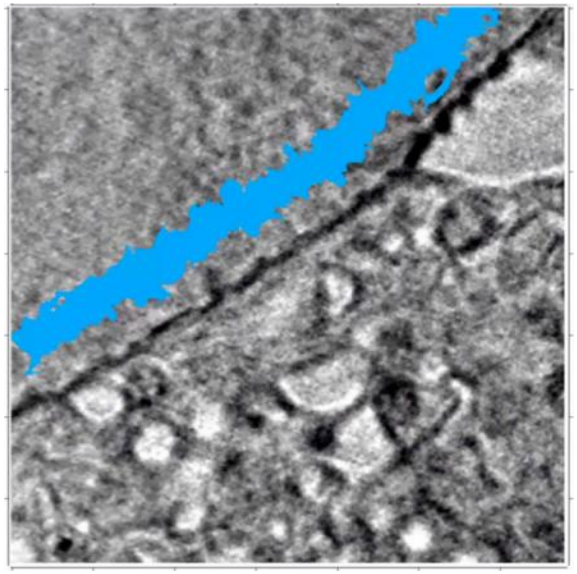
Pen Width = 15



Pen Width = 31

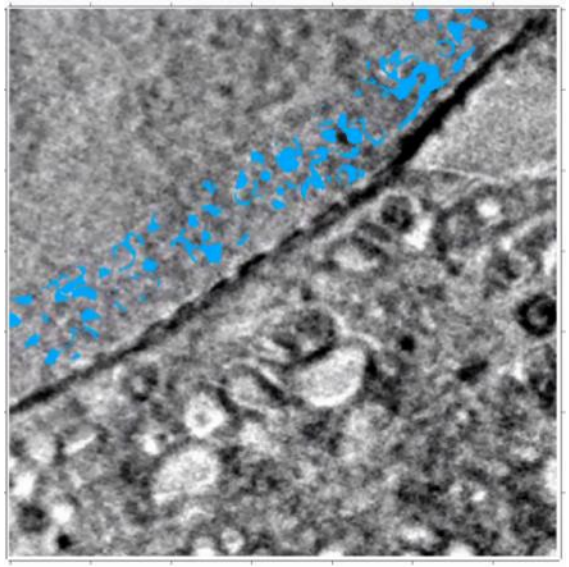
- The entire supervoxel is selected if any paint touches them.
- Whilst annotating in supervoxels annotations penetrate the volume in Z.

Annotation Using Supervoxels



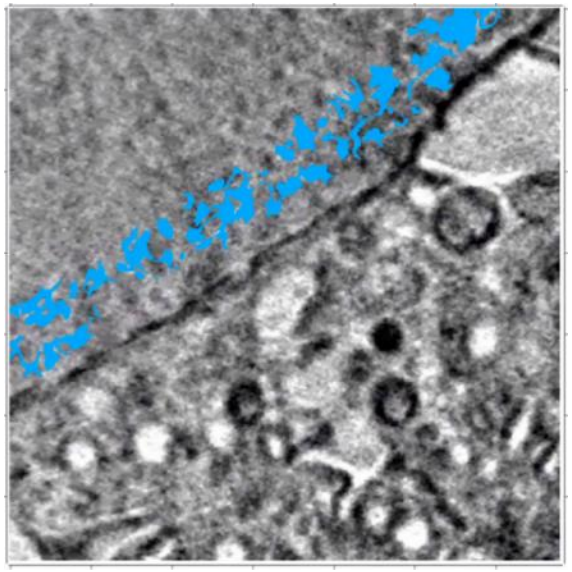
Pen Width = 15
Center Slice

10 slices below

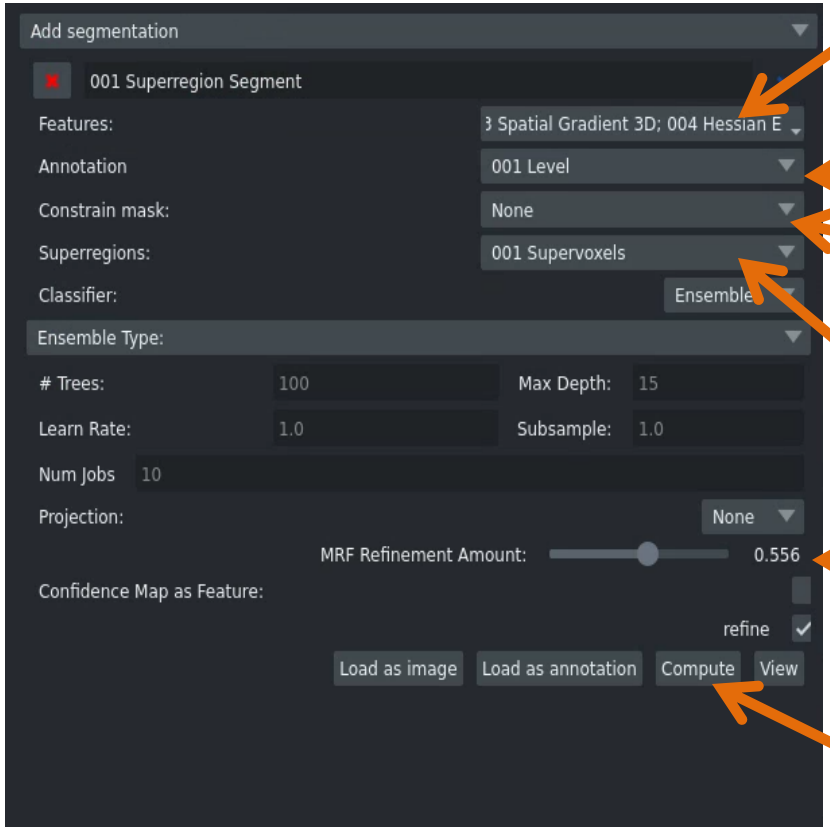


Annotating using supervoxels penetrates multiple Z-slices

10 slices above



Super-region Segmentation Pipeline



Select features for training and prediction

- Generally want to choose as many sources as possible
- Generally do not choose raw data
- For large areas try: Total Variation, Hessian Eigenvalues
- For small areas try: the above, plus Gaussian Feature Filters

Choose which level to train with

Optional (can constrain model training region based o annotation levels)

Choose the supervoxel region image to use.

Improves the prediction of the resultant model (The refine checkbox turns off the refinement, The slider value sets the amount of refinement.

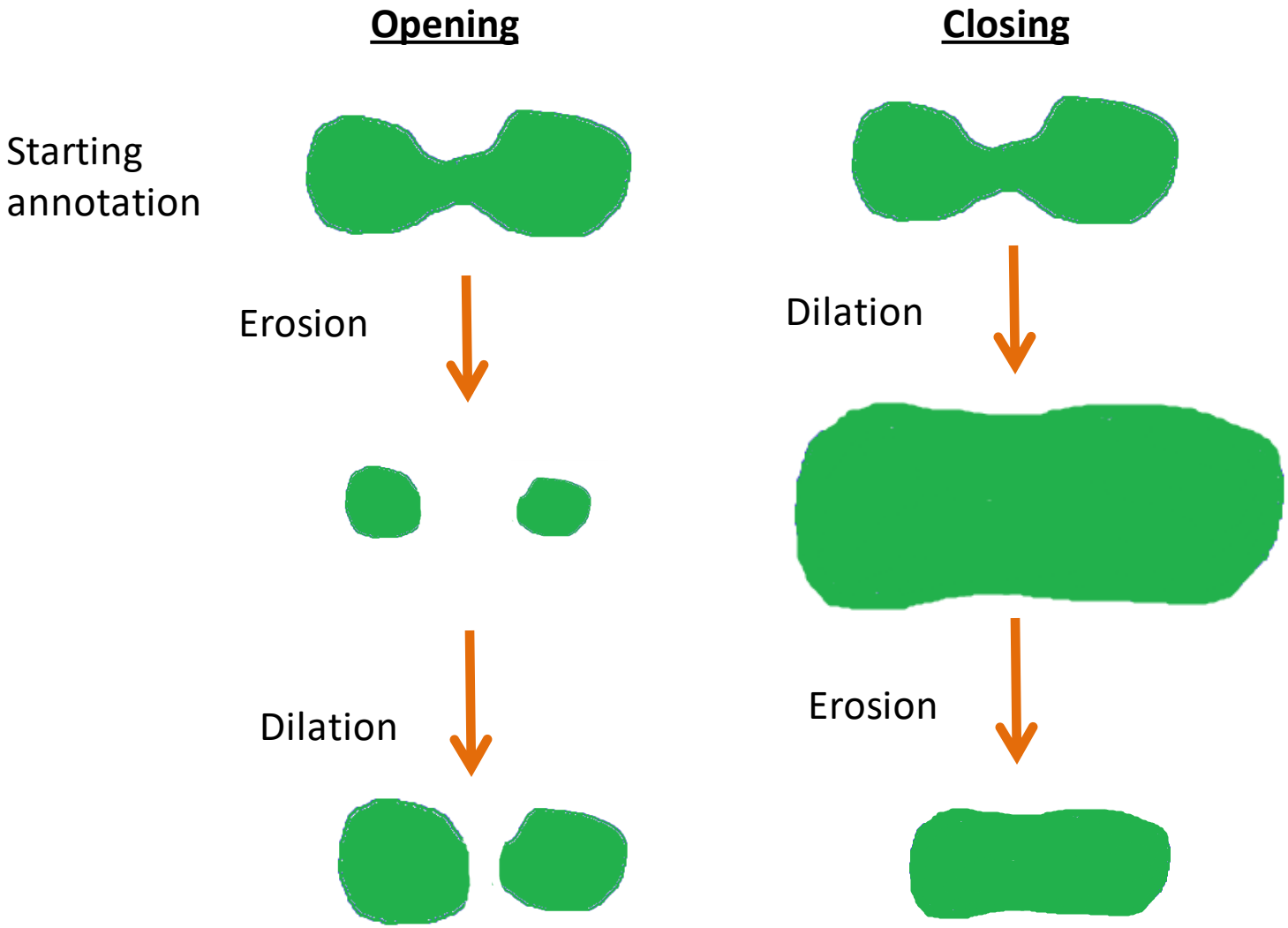
Train and predict

Morphology Filters (in Features tab)

Morphology:

- **Dilation -** This adds pixels to the inner and outer boundary of the annotation.
- **Erosion -** This strips away layers of pixels from the inner and outer boundary of the annotation. Small objects can be deleted easily with erosion.
- **Opening** - Erosion followed by dilation.
- **Closing** - Dilation followed by erosion.

Segmenting Organelles: Refinement



SuRVoS Workspace

HDF5: on-disk storage (.h5/.hdf5 extension)

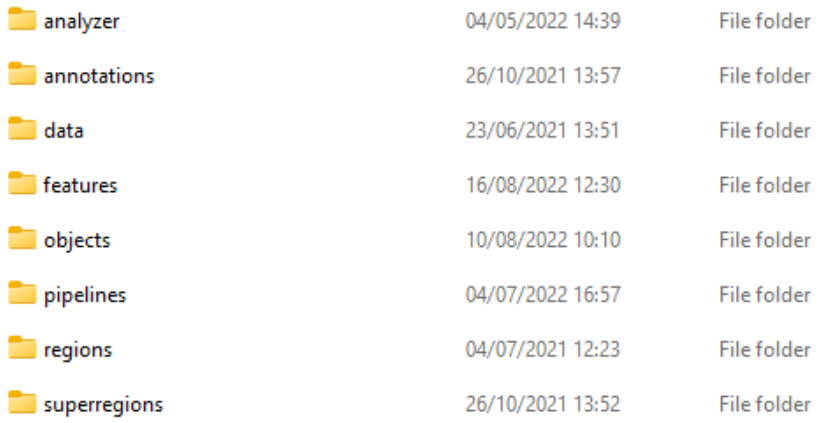
- Read data to memory on-the-fly
- Only load required data

Pros:

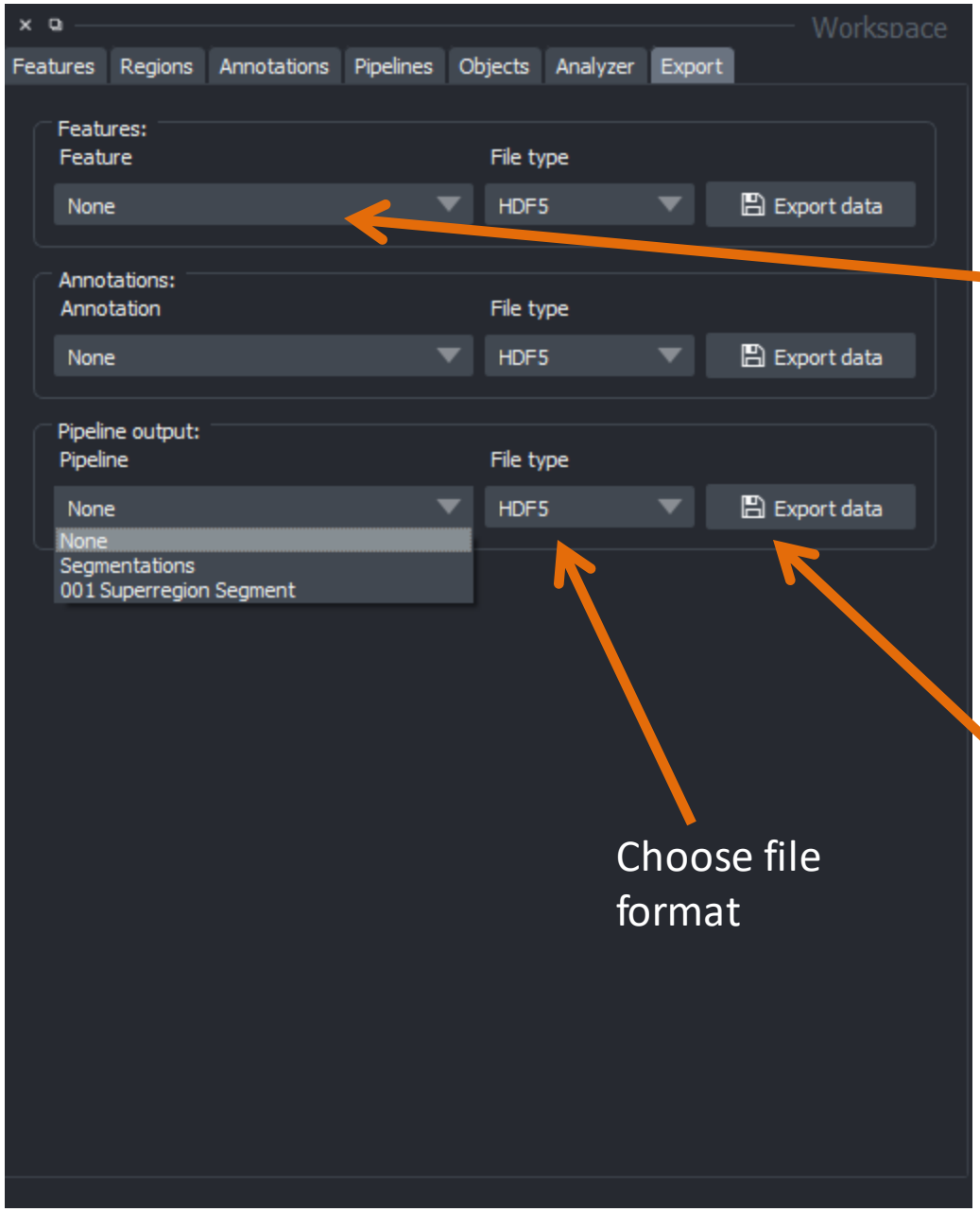
- Work with very large data (larger than RAM)
- Work on Region of Interests efficiently
- Safe. Robust.

Cons:

- Performance loss on loading data to memory and saving to disk.



Exporting Data

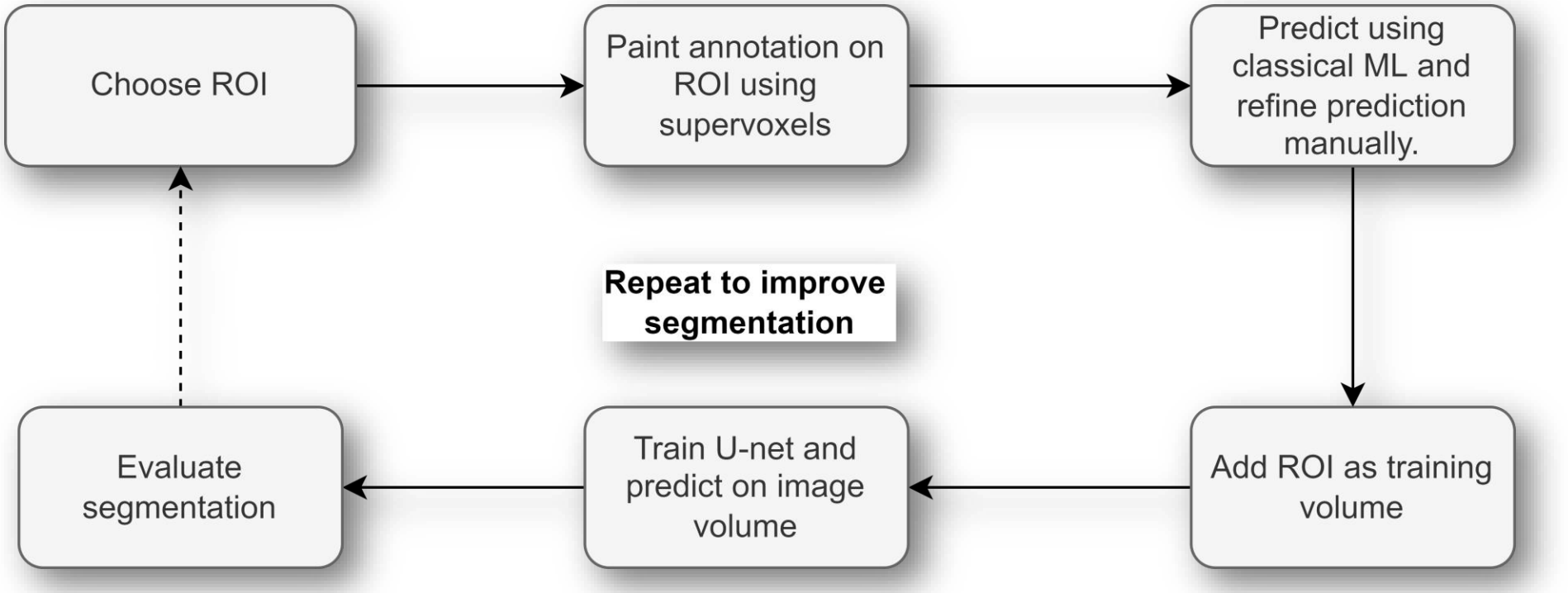


Choose data to export

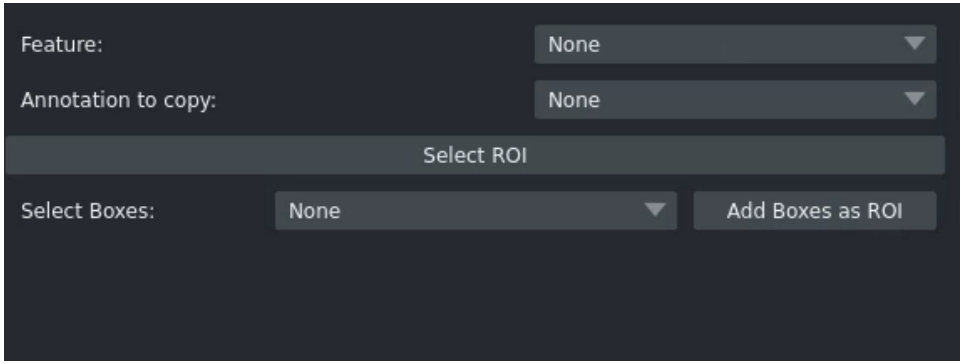
Choose file format

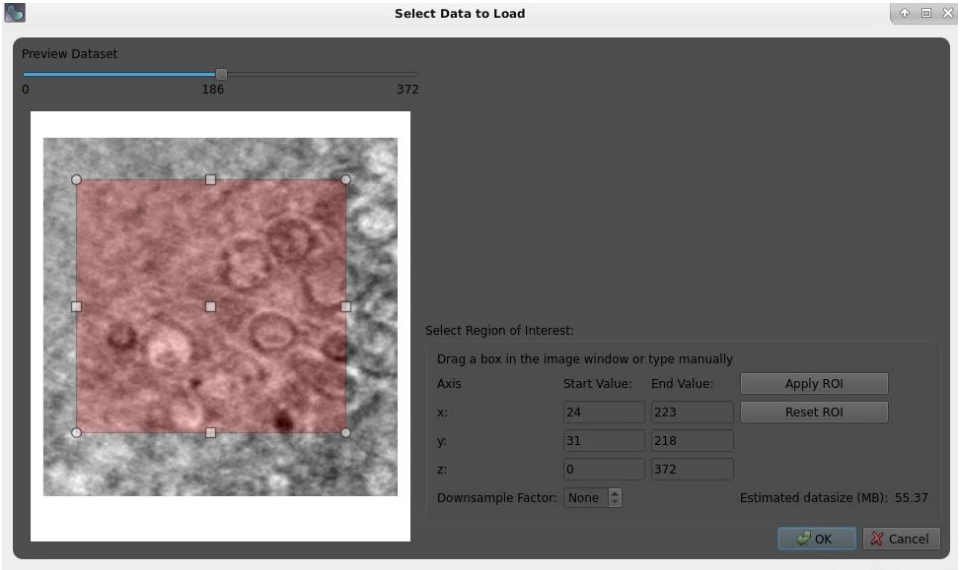
When ready, click export and a file dialog pops up to choose the location to export to.

Segmentation Workflow



ROI Panel





Choose the feature to crop the ROI from.

Click Select ROI

Use the Dialog box to draw a ROI on the image to select the X and Y coordinates of the ROI. Enter in the Z-coordinates manually.

Click Apply ROI

Click OK

Click Refresh

Find your ROI on the workspace dropdown box.

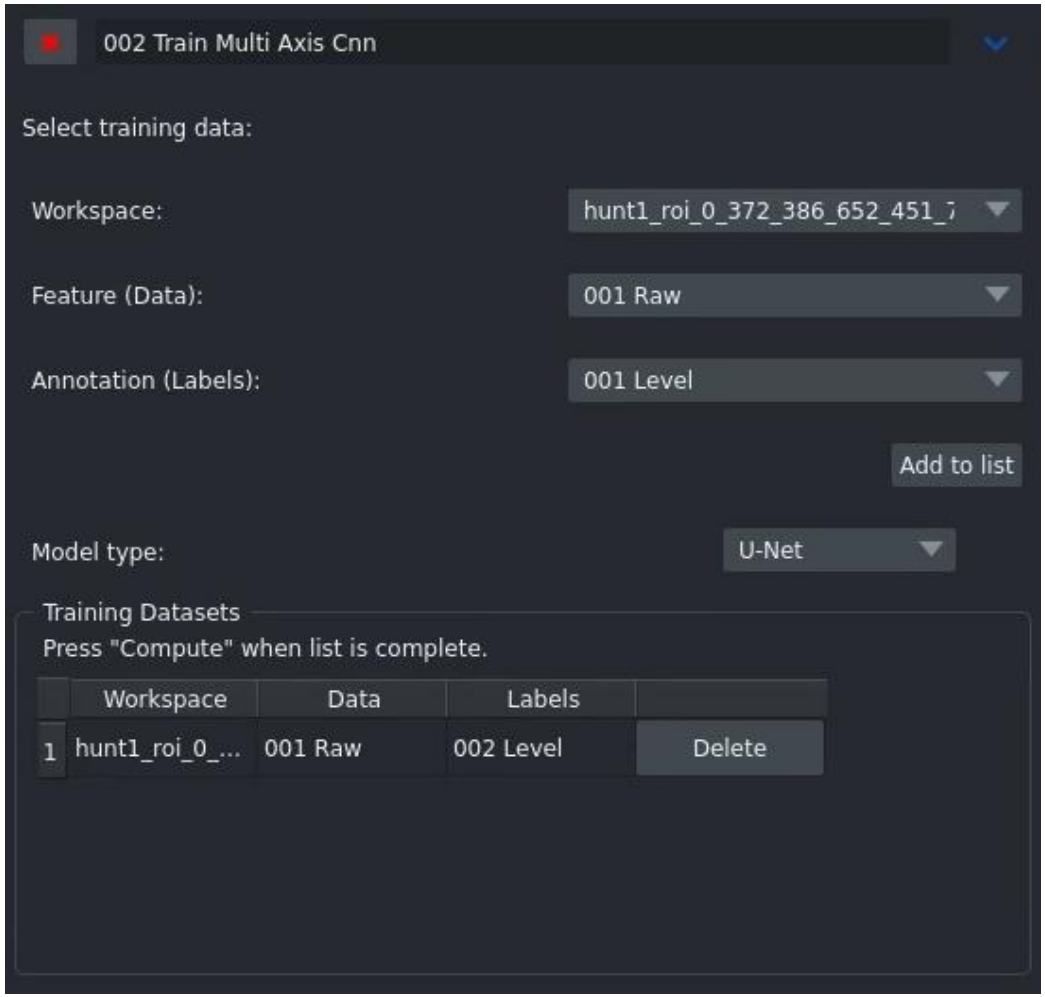
Segmentation Workflow

The labels for the deep learning module need to cover the entire image. At the moment the best way to achieve this is to paint some annotation using supervoxels and then use the Superregion Segment plugin to predict over the remaining voxels.

The superregion segment plugin uses image data from multiple features as well as the provided annotation and the supervoxels to train a classical machine learning (Ensemble or SVM) method and predict over the entire volume. The MRF Refinement process adds some computation time (usually not much, and this process is generally < 1 minute) but smooths the labels.

Once the segmentation is predicted, it can be corrected using 'Load As Annotation' which makes the prediction an annotation layer. The annotation tools can be used to manually correct it before it is used as training data for the CNN.

U-net: Training



1. Choose the workspace to get the training data from (may be an ROI workspace).

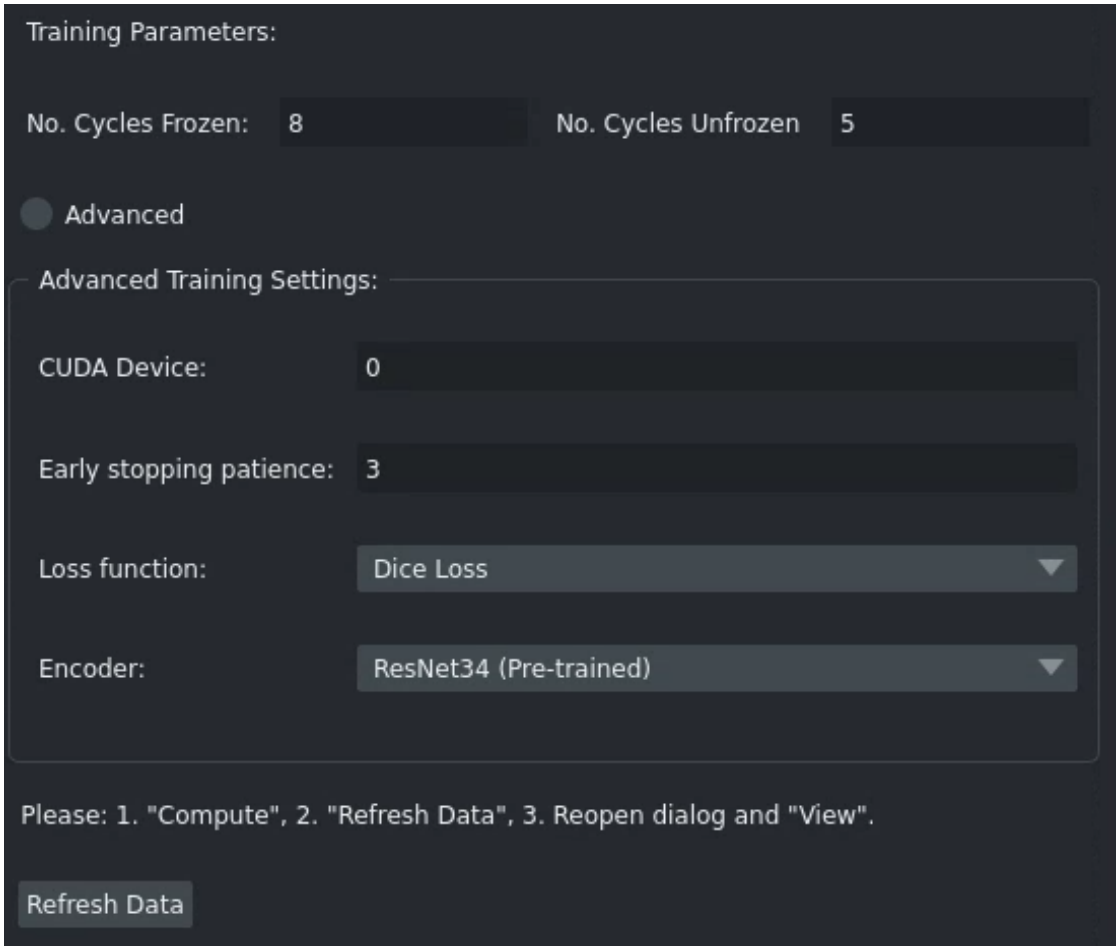
2. Choose input feature to use as the image data to train on.

3. Choose annotation level to use for training U-net.

4. Add to list. The training dataset appears in the table below. It can be removed with the Delete button.

5. Select the Model type, such as U-net, U-net++, FPN or DeepLab variants.

Training parameters



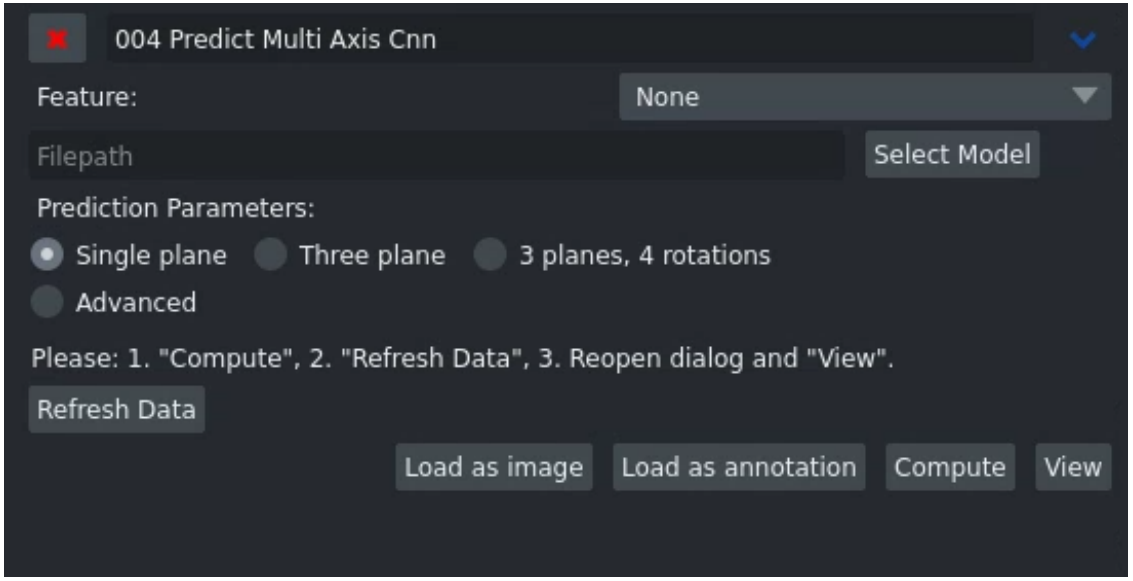
3. Set the Number of cycles, both frozen and unfrozen. The defaults here are often effective.

4. Advanced settings include setting the CUDA Device, a mechanism to stop early if training is not improving significantly, selection of the Loss function to use and selection of the encoder to use for the U-net-type architecture.

5. Compute. This prints a lot of activity to the terminal.

6. View the results or load them as an image or as annotation (e.g. for editing the output).

U-net: Prediction on original volume



1. Predict on a new image volume by selecting the annotation level and feature as in the training step.

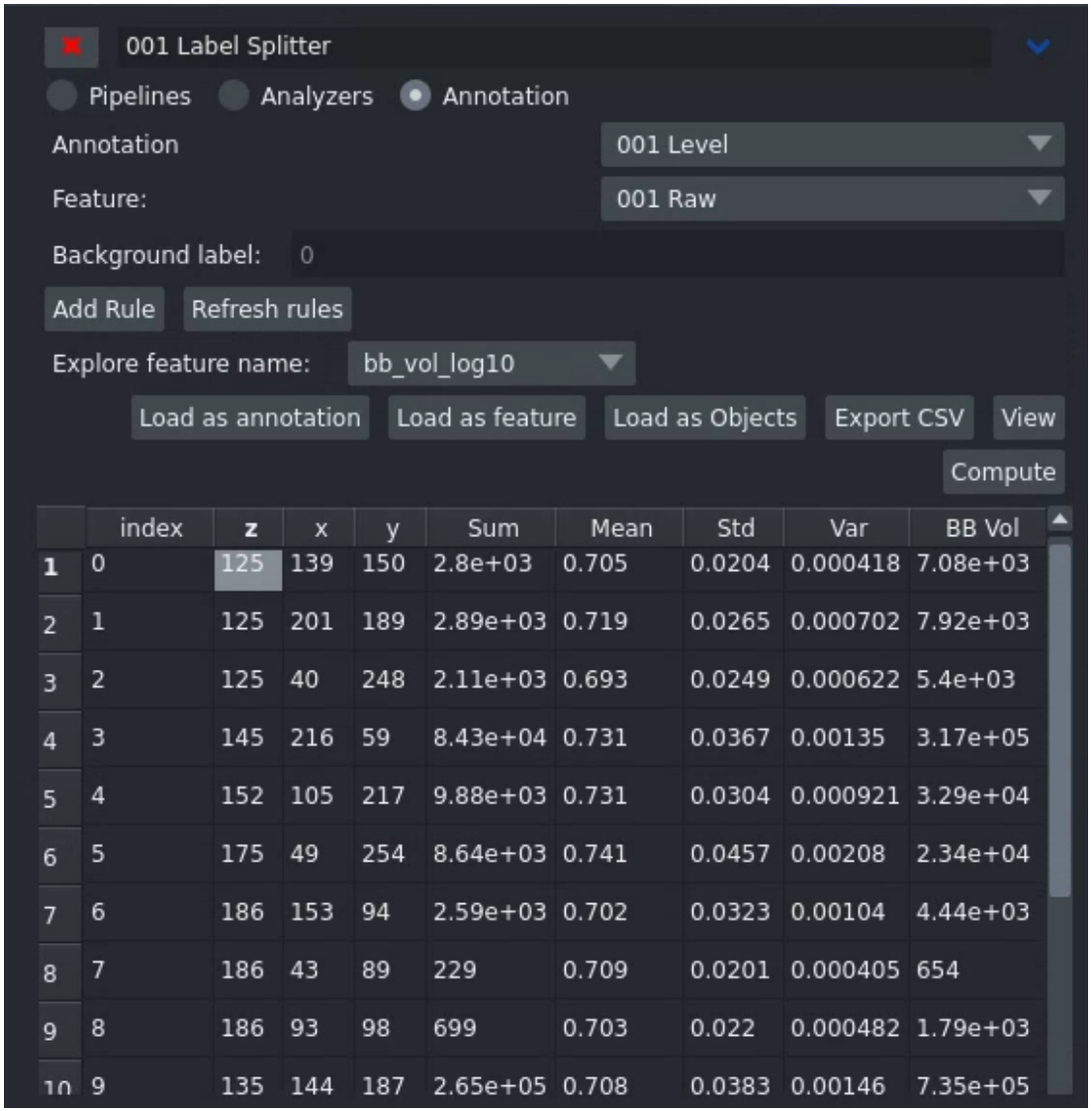
2. Select Model, which opens up a file dialog where you have to locate the saved trained U-net model from a training plugin.

3. Select Single-Plane or Three-Plane, where Three Plane predicts in 3 directions and combines the output.

4. Compute.

5. View or Load in as image or as annotation.

Label Analyzer/Splitter



	index	z	x	y	Sum	Mean	Std	Var	BB Vol
1	0	125	139	150	2.8e+03	0.705	0.0204	0.000418	7.08e+03
2	1	125	201	189	2.89e+03	0.719	0.0265	0.000702	7.92e+03
3	2	125	40	248	2.11e+03	0.693	0.0249	0.000622	5.4e+03
4	3	145	216	59	8.43e+04	0.731	0.0367	0.00135	3.17e+05
5	4	152	105	217	9.88e+03	0.731	0.0304	0.000921	3.29e+04
6	5	175	49	254	8.64e+03	0.741	0.0457	0.00208	2.34e+04
7	6	186	153	94	2.59e+03	0.702	0.0323	0.00104	4.44e+03
8	7	186	43	89	229	0.709	0.0201	0.000405	654
9	8	186	93	98	699	0.703	0.022	0.000482	1.79e+03
10	9	135	144	187	2.65e+05	0.708	0.0383	0.00146	7.35e+05

1. Select the type of input (e.g. Pipelines)
2. Select the layer to use.

3. Select a feature to use as an image for the calculation of image features (probably the raw data).

4. Choose a feature name to explore.

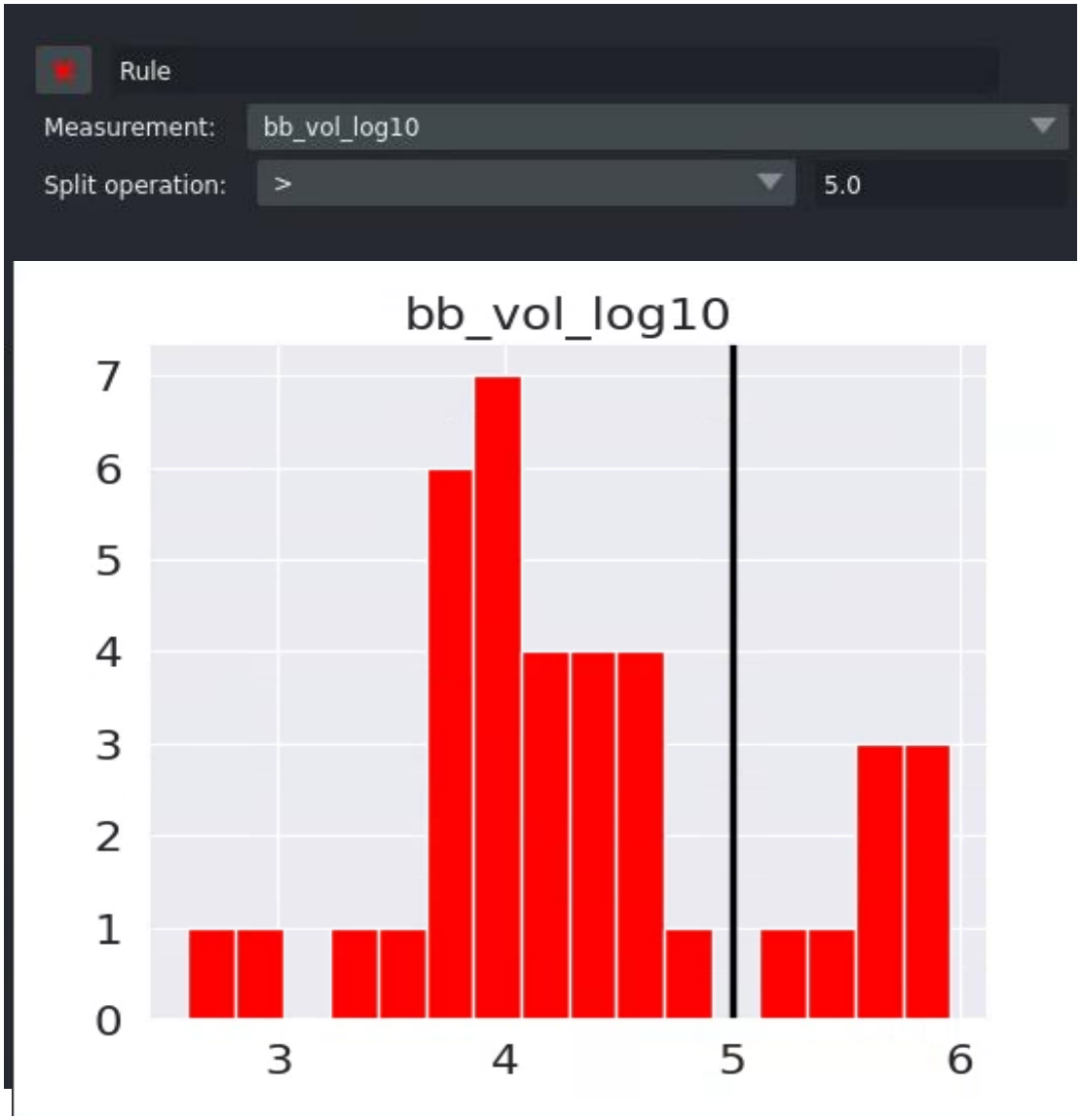
5. Click Compute.

View the table and click on individual entries.

View the plot.

Select a different feature and re-Compute.

Label Splitter



The Label Splitter allows you to add rules to split the components by properties such as bounding volume size.

Acknowledgements

Nottingham University
Andrew P. French
Tony Pridmore

Diamond Light Source
Imanol Luengo
Michele Darrow
Mark Basham
Avery Pennington
Olly King
Win Min Tun
Liz Duke
Matt Spink
Alan Ashton
Kyle Dent

Rosalind Franklin Institute
Elaine M. Ho
Luis Perdigao

Also thanks to
Nvidia Corporation for the donation of a Tesla K40.