
# ShuttleTracker

## Tutorial and overview

featuring version 1.3.0 (September 2019)

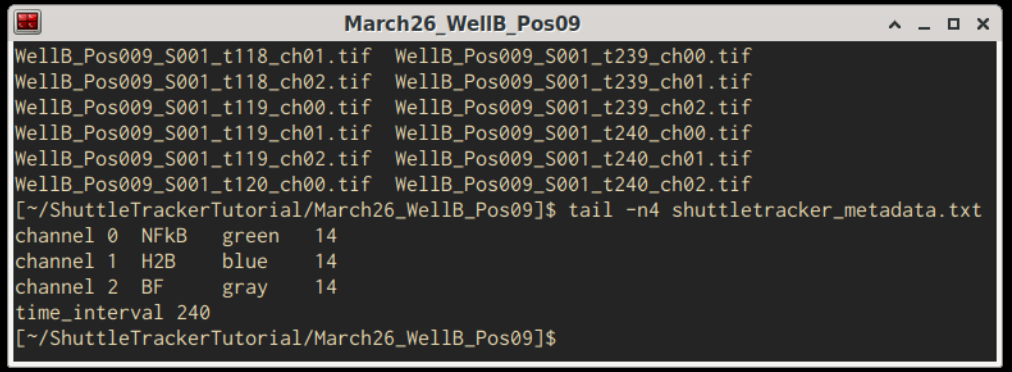
http://pmbm.ippt.pan.pl/software/shuttletracker

In this tutorial, we are going to process images in which nuclear translocation of a transcription factor, NF-$\varkappa$B, is triggered by irregular pulses of TNF administered with the use of a microfluidic device.

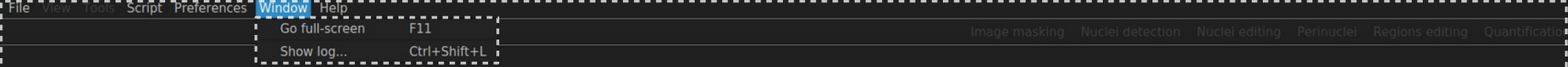
To proceed, an archive of the images of MEF cells should be retrieved from URL:
http://pmbm.ippt.pan.pl/software/shuttletracker/tutorial/March26_WellB_Pos09.zip
The archive contains 16-bit TIFF images of 241 time frames recorded in 3 channels:

- suffix ch00 – GFP-labeled NF-$\varkappa$B subunit,
- suffix ch01 – nuclear signal from a fluorescently labeled histone,
- suffix ch02 – bright-field view.

The archive contains also a text file, shuttletracker_metadata.txt, which provides a terse description of the images.

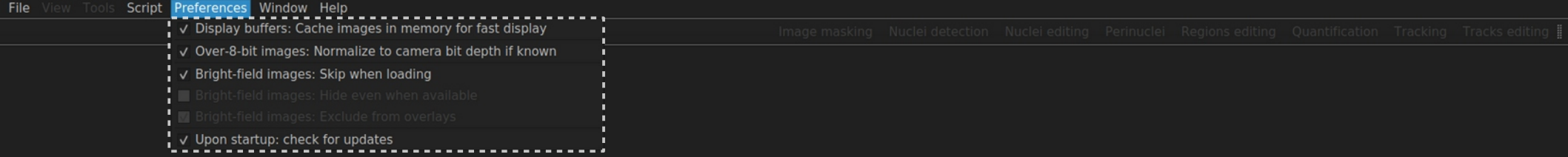



A quick peek into the extracted archive in terminal.
The last column is the microscope camera bit depth.




After starting ShuttleTracker, the **main window** with a standard menu bar and a toolbox bar (initially disabled) is displayed.

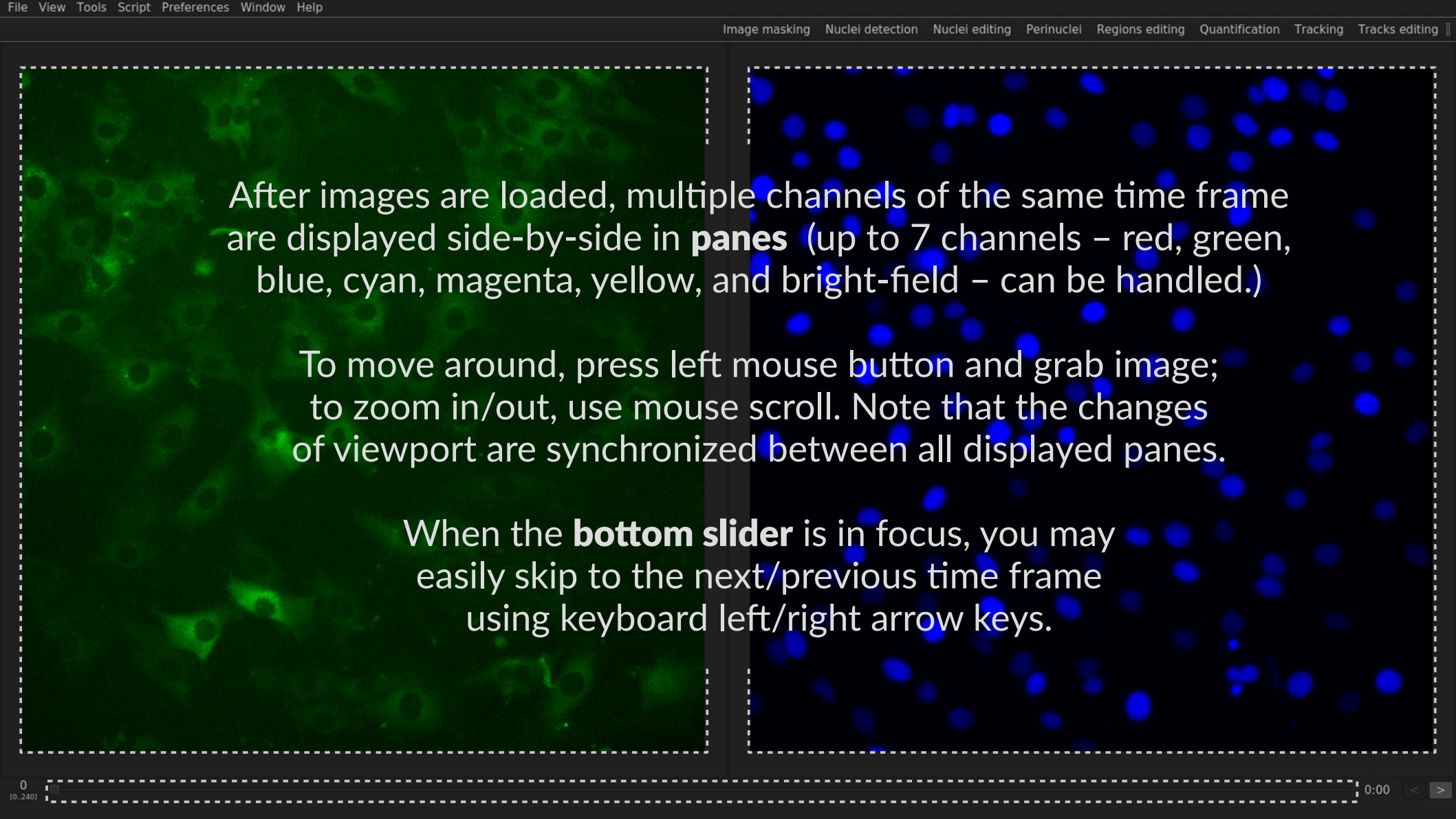
To go full-screen, you can press F11 (or select **menu Window** → Go full-screen).

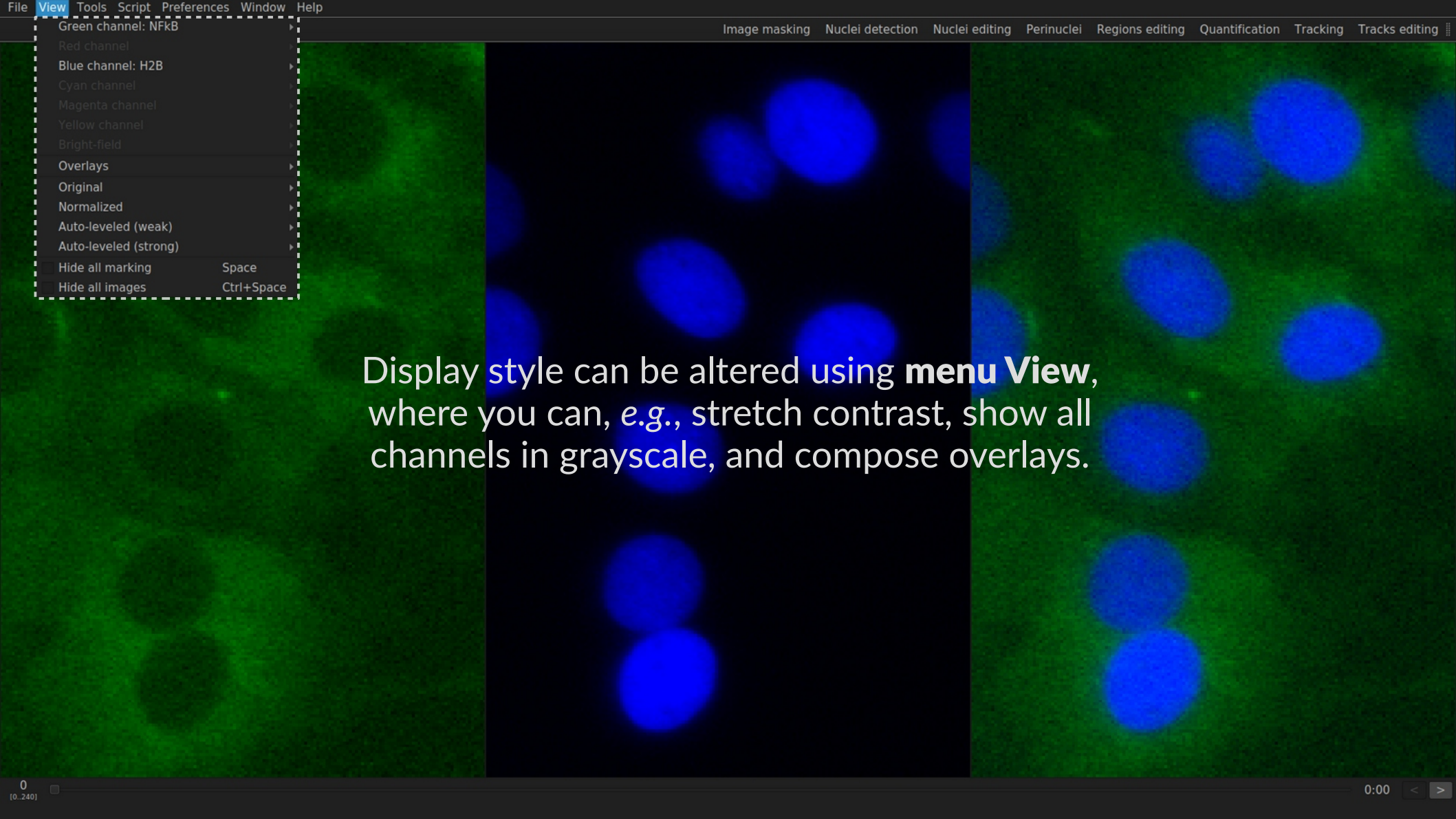


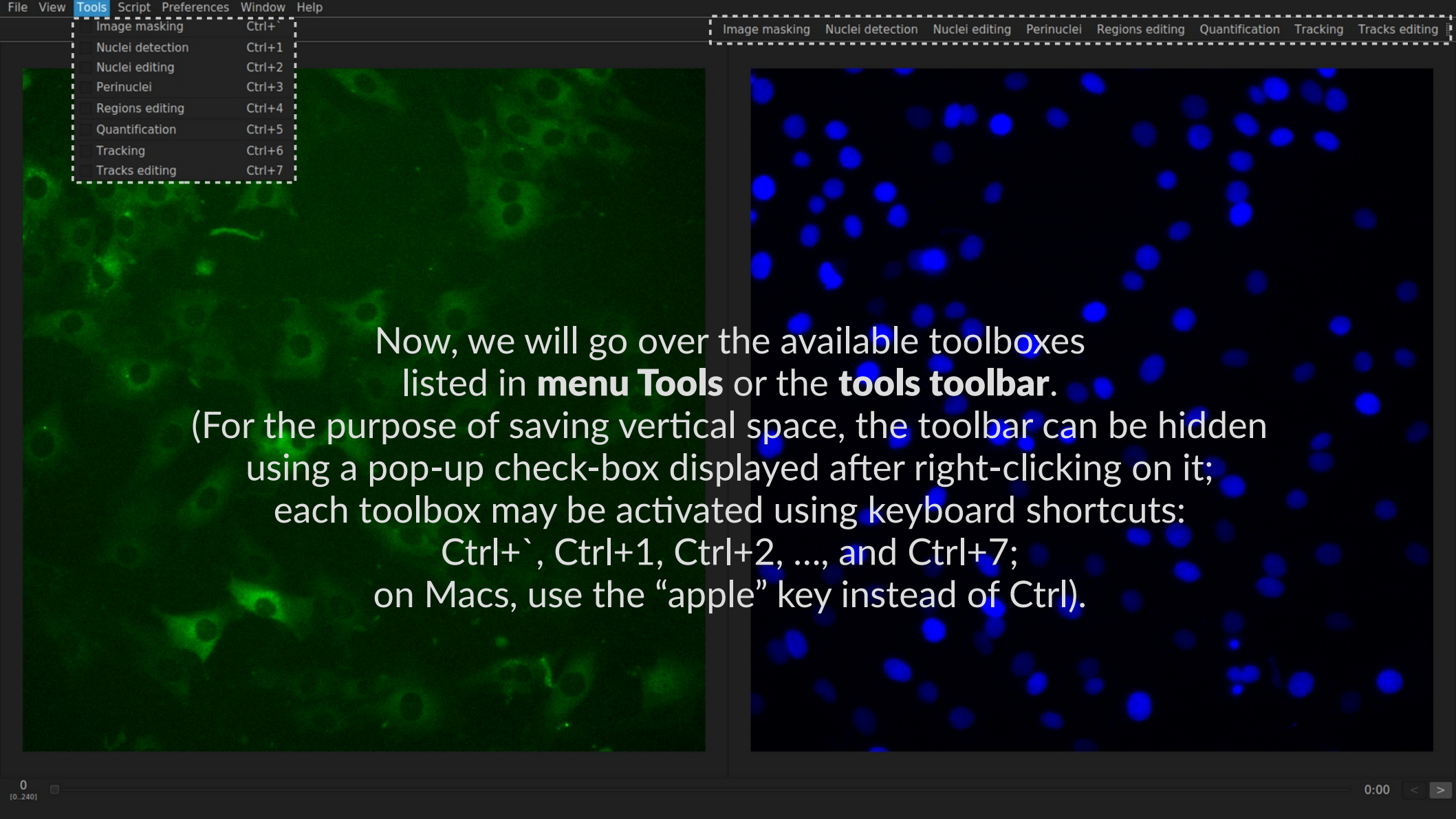

Before we load images, in **menu Preferences**, we may:

- decide to use fast display buffers (only if you have ~6 GB RAM free),
- want to normalize images to camera bit depth (which will cause that only 14 lower bits of 16-bit TIFFs will be internally mapped to 8 bits),
- and decide to hide or skip at all bright-field images (because they are not necessary for the analysis).

(These preferences will be saved upon exit and restored upon next launch.)


File View Tools Script Preferences Window Help
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
After images are loaded, multiple channels of the same time frame are displayed side-by-side in **panes** (up to 7 channels – red, green, blue, cyan, magenta, yellow, and bright-field – can be handled.)
To move around, press left mouse button and grab image; to zoom in/out, use mouse scroll. Note that the changes of viewport are synchronized between all displayed panes.
When the **bottom slider** is in focus, you may easily skip to the next/previous time frame using keyboard left/right arrow keys.
0
[0..240]
0:00


File View Tools Script Preferences Window Help
Green channel: NFkB
Red channel
Blue channel: H2B
Cyan channel
Magenta channel
Yellow channel
Bright-field
Overlays
Original
Normalized
Auto-leveled (weak)
Auto-leveled (strong)
Hide all marking Space
Hide all images Ctrl+Space
Image masking
Nuclei detection
Nuclei editing
Perinuclei
Regions editing
Quantification
Tracking
Tracks editing
Display style can be altered using **menu View**, where you can, *e.g.*, stretch contrast, show all channels in grayscale, and compose overlays.
0
[0..240]
0:00


File View Tools Script Preferences Window Help
Image masking Ctrl+`
Nuclei detection Ctrl+1
Nuclei editing Ctrl+2
Perinuclei Ctrl+3
Regions editing Ctrl+4
Quantification Ctrl+5
Tracking Ctrl+6
Tracks editing Ctrl+7
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
Now, we will go over the available toolboxes
listed in **menu Tools** or the **tools toolbar**.
(For the purpose of saving vertical space, the toolbar can be hidden
using a pop-up check-box displayed after right-clicking on it;
each toolbox may be activated using keyboard shortcuts:
Ctrl+`, Ctrl+1, Ctrl+2, ..., and Ctrl+7;
on Macs, use the “apple” key instead of Ctrl).
0
[0..240]
0:00

File View Tools Script Preferences Window Help

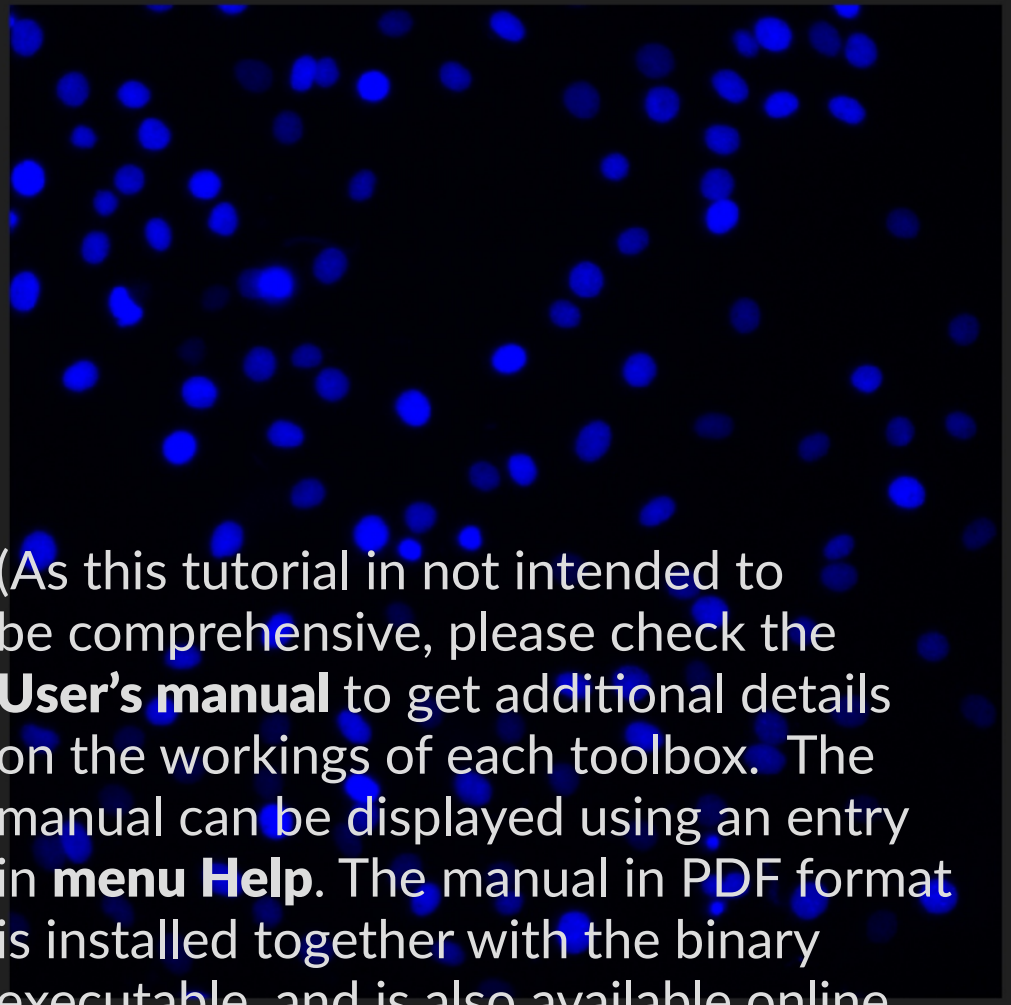
User's manual... Ctrl+Shift+M
About... Ctrl+Shift+V

ShuttleTracker Manual

# ShuttleTracker – User's manual

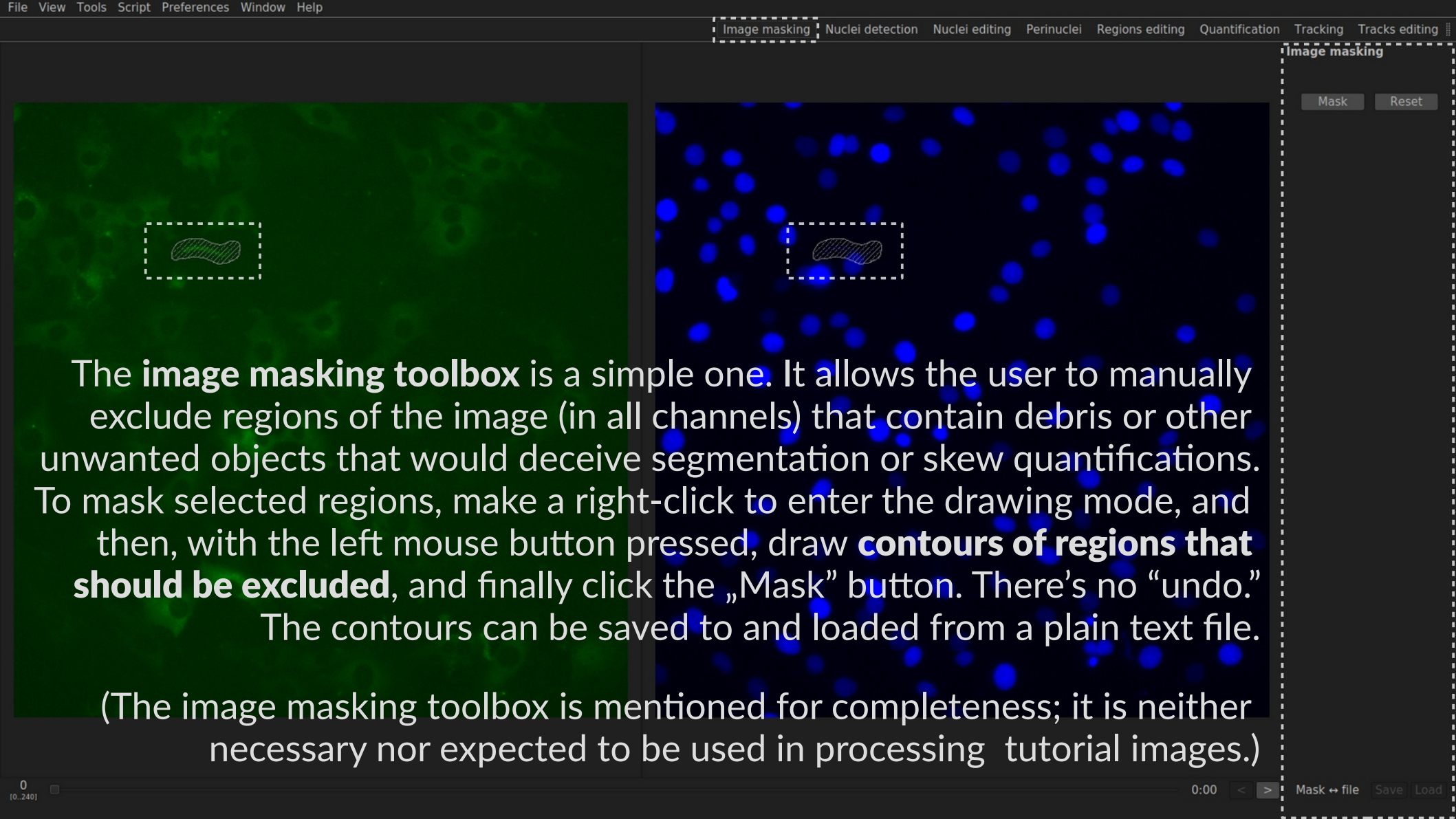
Table of contents:

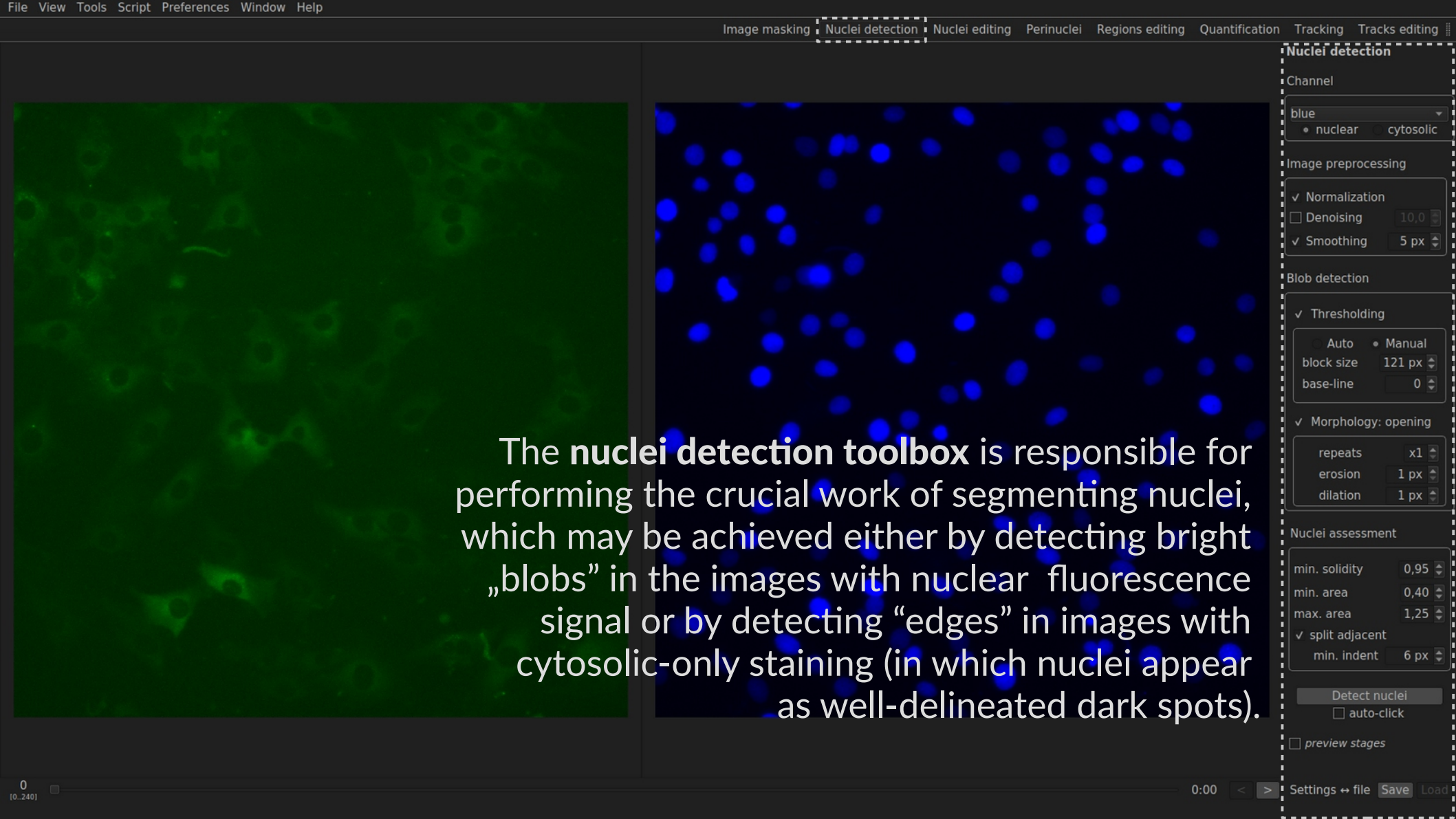


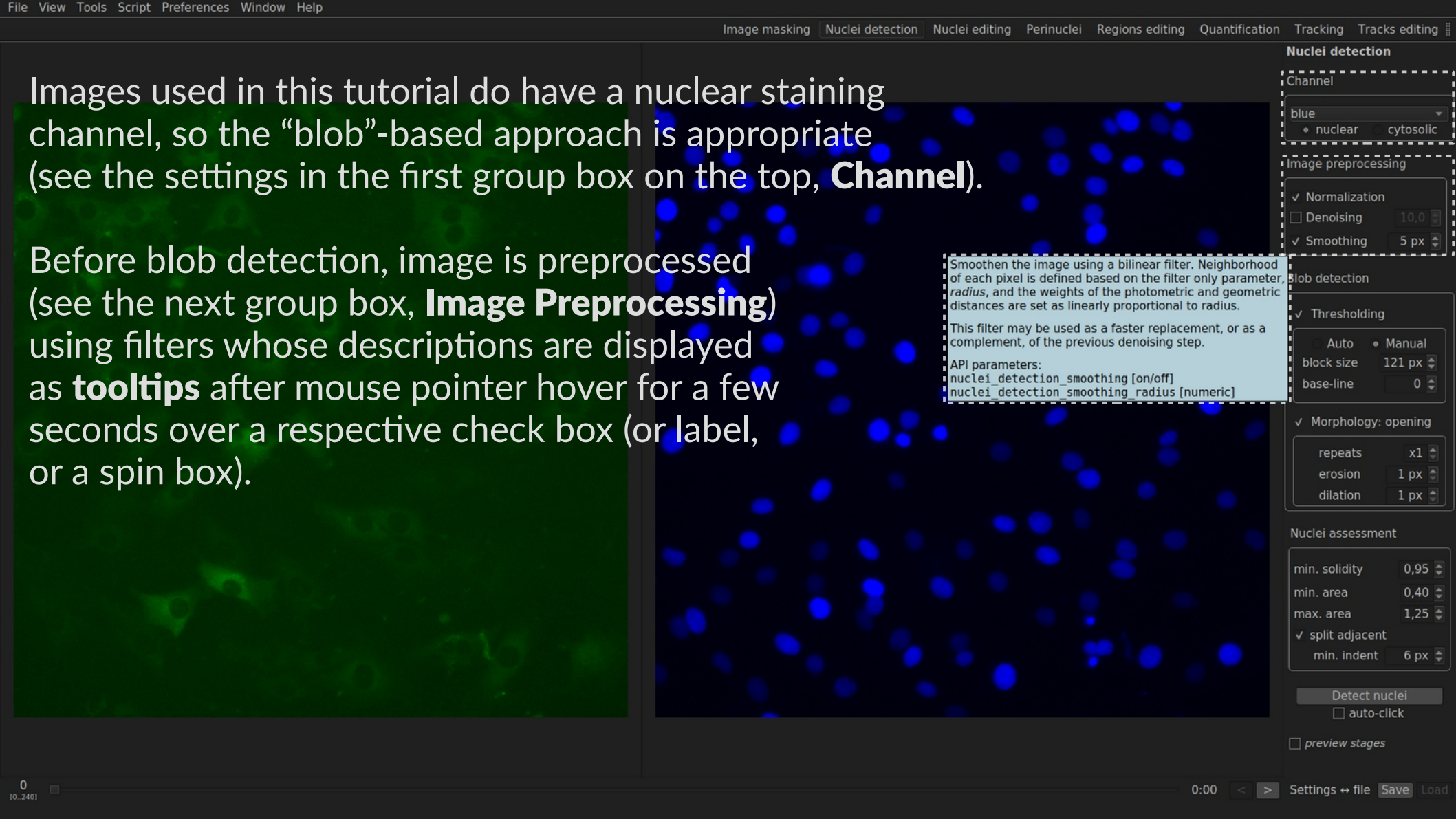


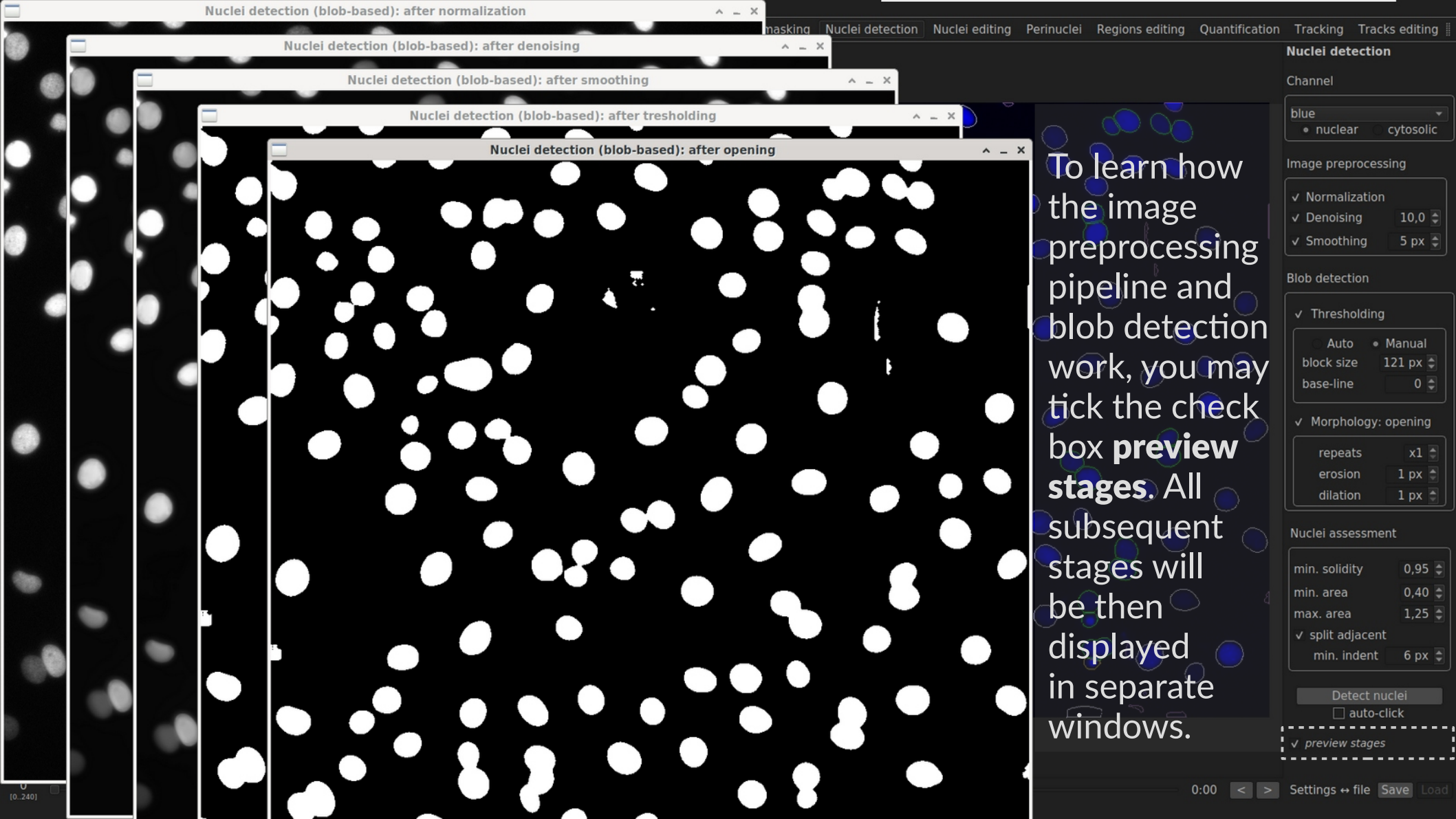
(As this tutorial in not intended to be comprehensive, please check the **User's manual** to get additional details on the workings of each toolbox. The manual can be displayed using an entry in **menu Help**. The manual in PDF format is installed together with the binary executable, and is also available online on the project homepage.)

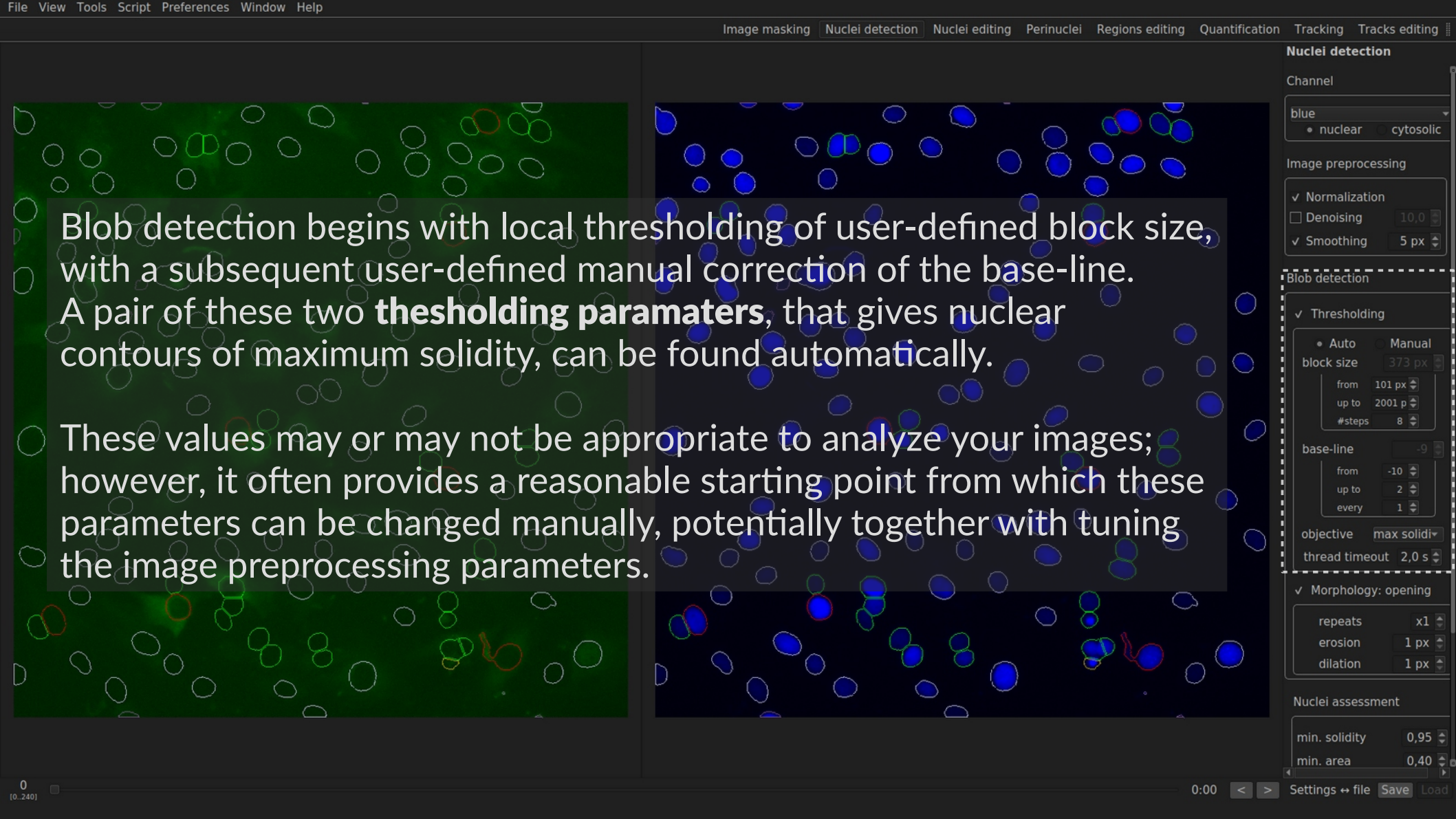
0:00

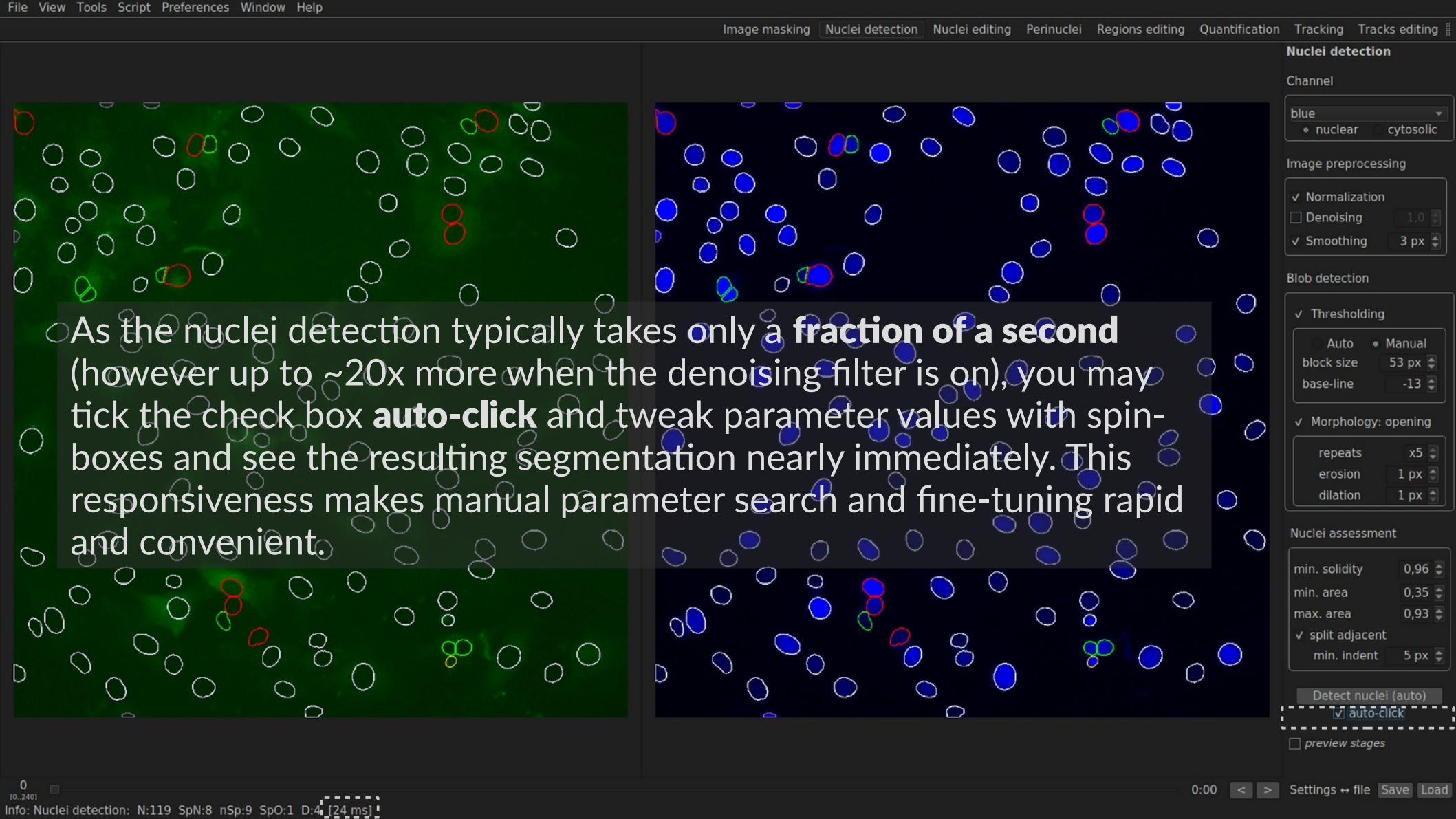

File View Tools Script Preferences Window Help
Image masking
Nuclei detection
Nuclei editing
Perinuclei
Regions editing
Quantification
Tracking
Tracks editing
Image masking
Mask
Reset
The **image masking toolbox** is a simple one. It allows the user to manually exclude regions of the image (in all channels) that contain debris or other unwanted objects that would deceive segmentation or skew quantifications. To mask selected regions, make a right-click to enter the drawing mode, and then, with the left mouse button pressed, draw **contours of regions that should be excluded**, and finally click the „Mask" button. There's no "undo." The contours can be saved to and loaded from a plain text file.
(The image masking toolbox is mentioned for completeness; it is neither necessary nor expected to be used in processing tutorial images.)
0
[0..240]
0:00
Mask ↔ file
Save
Load

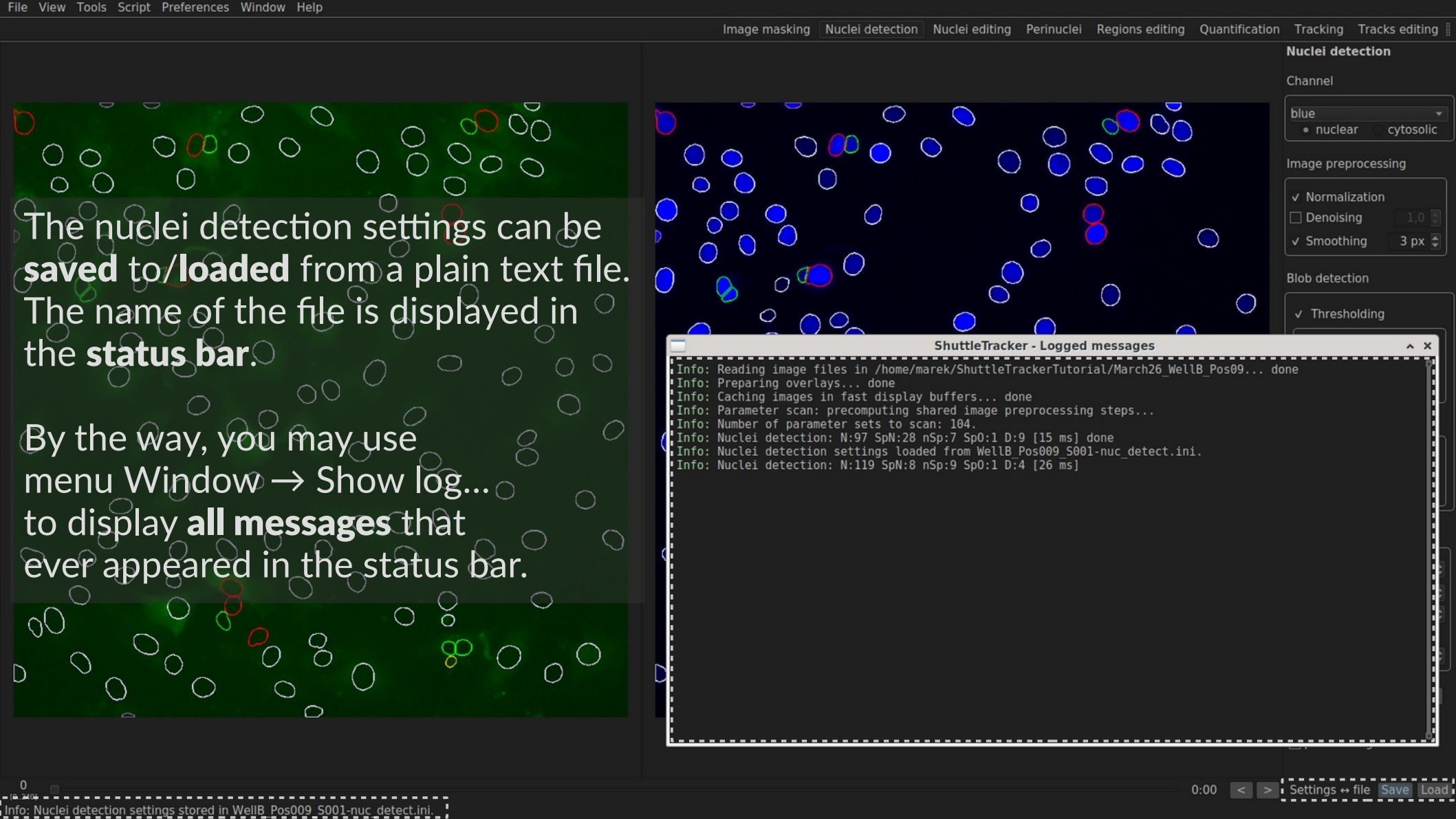

File View Tools Script Preferences Window Help
Image masking
Nuclei detection
Nuclei editing
Perinuclei
Regions editing
Quantification
Tracking
Tracks editing
Nuclei detection
Channel
blue
nuclear
cytosolic
Image preprocessing
Normalization
Denoising
10,0
Smoothing
5 px
Blob detection
Thresholding
Auto
Manual
block size
121 px
base-line
0
Morphology: opening
repeats
x1
erosion
1 px
dilation
1 px
Nuclei assessment
min. solidity
0,95
min. area
0,40
max. area
1,25
split adjacent
min. indent
6 px
Detect nuclei
auto-click
preview stages
0
[0..240]
0:00
Settings ↔ file
Save
Load
The nuclei detection toolbox is responsible for performing the crucial work of segmenting nuclei, which may be achieved either by detecting bright „blobs” in the images with nuclear fluorescence signal or by detecting “edges” in images with cytosolic-only staining (in which nuclei appear as well-delineated dark spots).


File View Tools Script Preferences Window Help
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
Images used in this tutorial do have a nuclear staining channel, so the “blob”-based approach is appropriate (see the settings in the first group box on the top, **Channel**).
Before blob detection, image is preprocessed (see the next group box, **Image Preprocessing**) using filters whose descriptions are displayed as **tooltips** after mouse pointer hover for a few seconds over a respective check box (or label, or a spin box).
Nuclei detection
Channel
blue
nuclear cytosolic
Image preprocessing
Normalization
Denoising 10,0
Smoothing 5 px
Smoothen the image using a bilinear filter. Neighborhood of each pixel is defined based on the filter only parameter, radius, and the weights of the photometric and geometric distances are set as linearly proportional to radius.
This filter may be used as a faster replacement, or as a complement, of the previous denoising step.
API parameters:
nuclei_detection_smoothing [on/off]
nuclei_detection_smoothing_radius [numeric]
Blob detection
Thresholding
Auto Manual
block size 121 px
base-line 0
Morphology: opening
repeats x1
erosion 1 px
dilation 1 px
Nuclei assessment
min. solidity 0,95
min. area 0,40
max. area 1,25
split adjacent
min. indent 6 px
Detect nuclei
auto-click
preview stages
0
[0..240]
0:00
Settings ↔ file Save Load

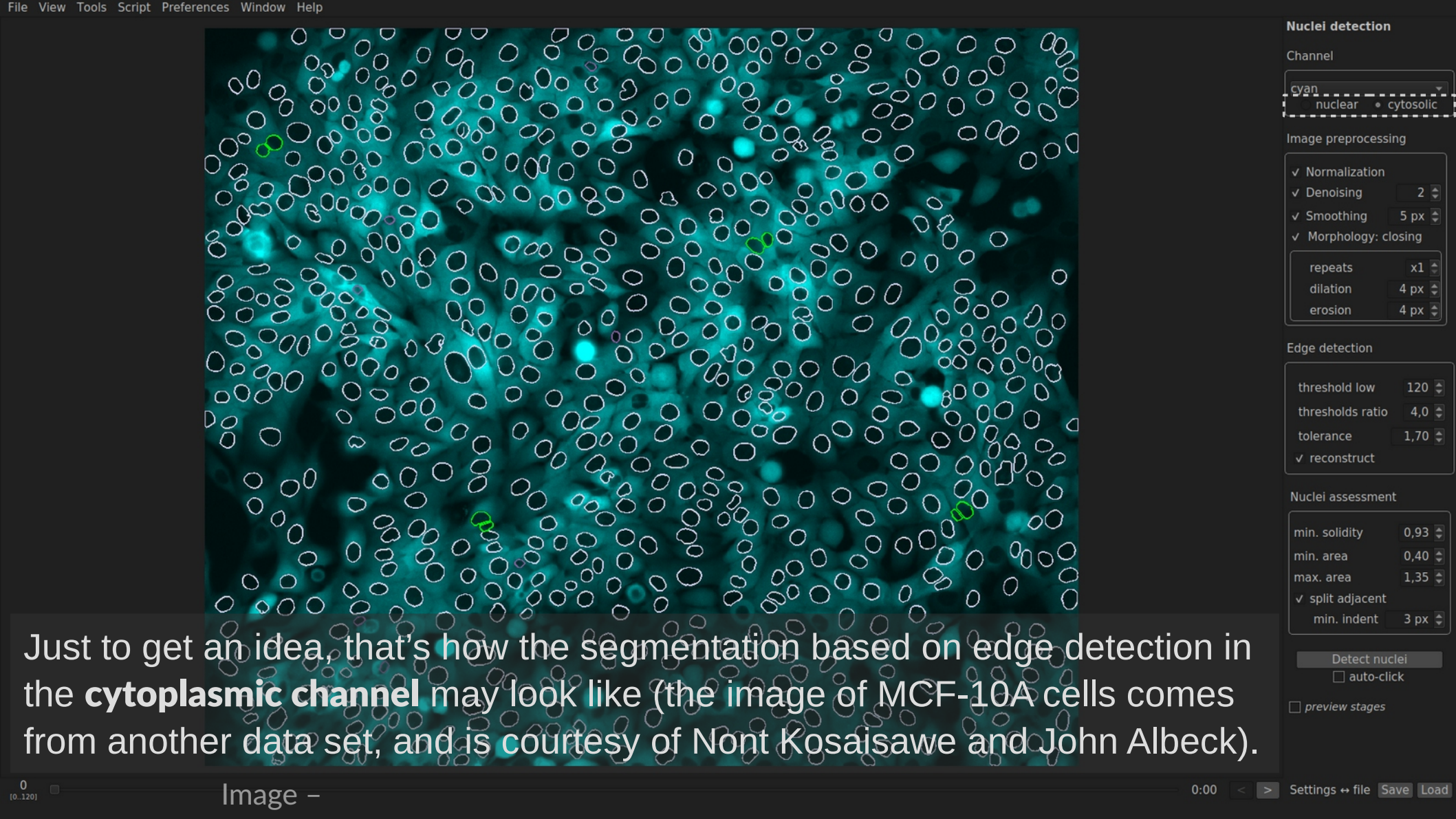

Nuclei detection (blob-based): after normalization
Nuclei detection (blob-based): after denoising
Nuclei detection (blob-based): after smoothing
Nuclei detection (blob-based): after tresholding
Nuclei detection (blob-based): after opening
masking
Nuclei detection
Nuclei editing
Perinuclei
Regions editing
Quantification
Tracking
Tracks editing
Nuclei detection
Channel
blue
nuclear
cytosolic
Image preprocessing
Normalization
Denoising 10,0
Smoothing 5 px
Blob detection
Thresholding
Auto
Manual
block size 121 px
base-line 0
Morphology: opening
repeats x1
erosion 1 px
dilation 1 px
Nuclei assessment
min. solidity 0,95
min. area 0,40
max. area 1,25
split adjacent
min. indent 6 px
Detect nuclei
auto-click
preview stages
0:00
Settings ↔ file
Save
Load
To learn how the image preprocessing pipeline and blob detection work, you may tick the check box **preview stages**. All subsequent stages will be then displayed in separate windows.

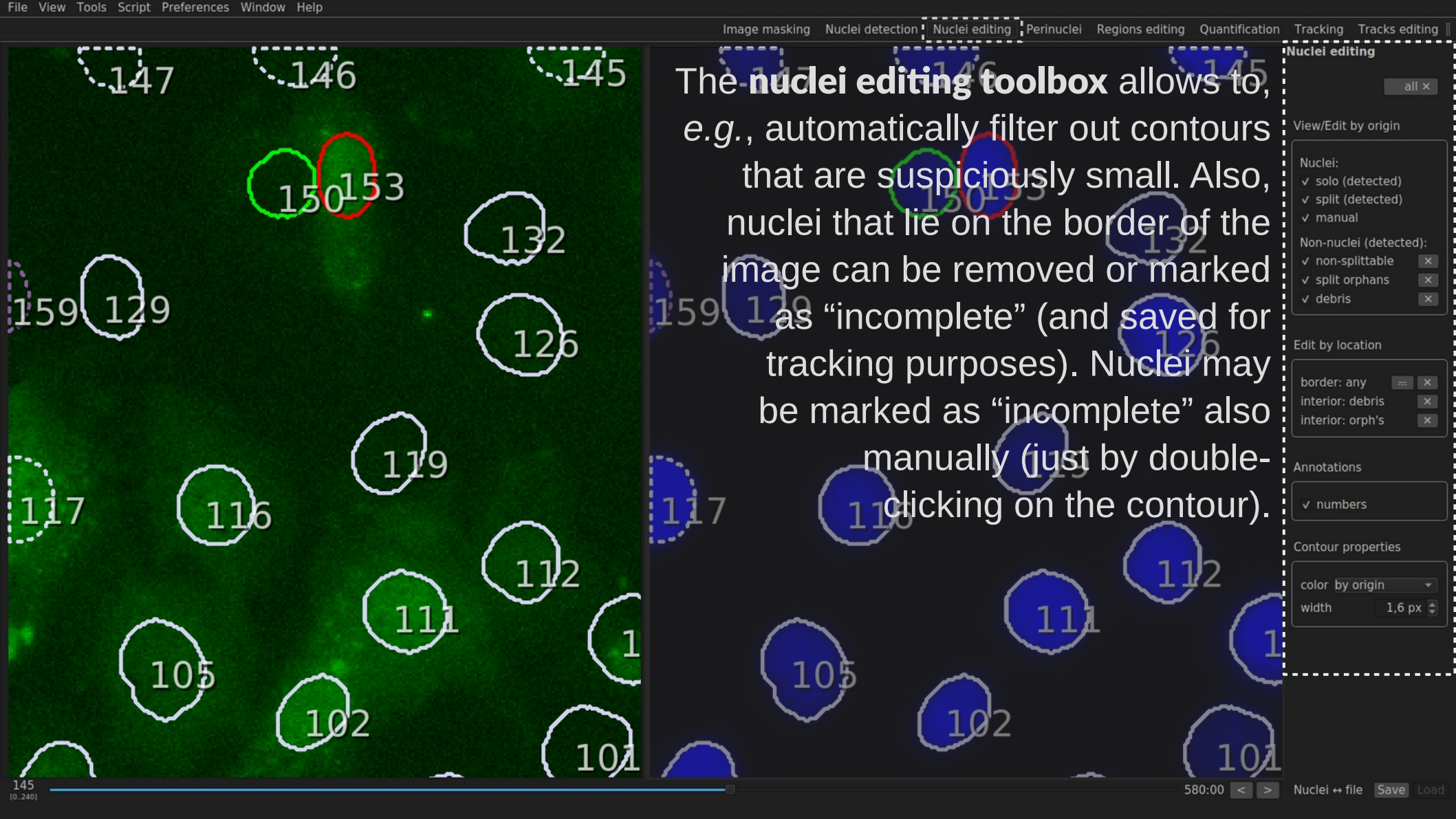

File View Tools Script Preferences Window Help
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
Nuclei detection
Channel
blue
nuclear cytosolic
Image preprocessing
Normalization
Denoising 10,0
Smoothing 5 px
Blob detection
Thresholding
Auto Manual
block size 373 px
from 101 px
up to 2001 p
#steps 8
base-line -9
from -10
up to 2
every 1
objective max solidi
thread timeout 2,0 s
Morphology: opening
repeats x1
erosion 1 px
dilation 1 px
Nuclei assessment
min. solidity 0,95
min. area 0,40
Blob detection begins with local thresholding of user-defined block size, with a subsequent user-defined manual correction of the base-line. A pair of these two **thesholding paramaters**, that gives nuclear contours of maximum solidity, can be found automatically.
These values may or may not be appropriate to analyze your images; however, it often provides a reasonable starting point from which these parameters can be changed manually, potentially together with tuning the image preprocessing parameters.
0
[0..240]
0:00
Settings ↔ file Save Load

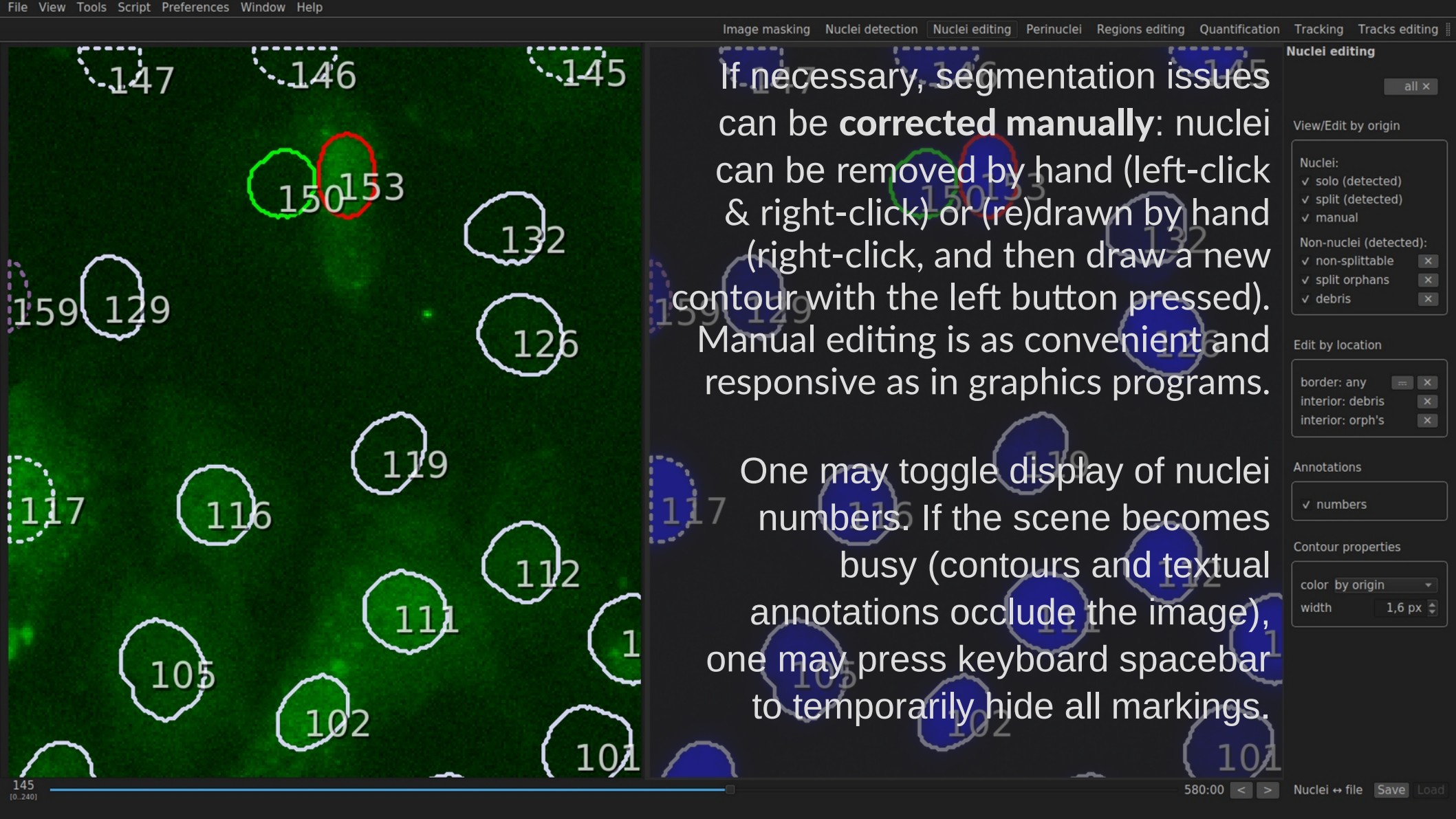

File View Tools Script Preferences Window Help
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
Nuclei detection
Channel
blue
nuclear cytosolic
Image preprocessing
Normalization
Denoising 1,0
Smoothing 3 px
Blob detection
Thresholding
Auto Manual
block size 53 px
base-line -13
Morphology: opening
repeats x5
erosion 1 px
dilation 1 px
Nuclei assessment
min. solidity 0,96
min. area 0,35
max. area 0,93
split adjacent
min. indent 5 px
Detect nuclei (auto)
auto-click
preview stages
As the nuclei detection typically takes only a fraction of a second (however up to ~20x more when the denoising filter is on), you may tick the check box auto-click and tweak parameter values with spin-boxes and see the resulting segmentation nearly immediately. This responsiveness makes manual parameter search and fine-tuning rapid and convenient.
0
[0..240]
0:00
Settings ↔ file
Save
Load
Info: Nuclei detection: N:119 SpN:8 nSp:9 SpO:1 D:4 [24 ms]


File View Tools Script Preferences Window Help
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
Nuclei detection
Channel
blue
nuclear cytosolic
Image preprocessing
Normalization
Denoising 1.0
Smoothing 3 px
Blob detection
Thresholding
The nuclei detection settings can be **saved** to/**loaded** from a plain text file. The name of the file is displayed in the **status bar**.
By the way, you may use menu Window → Show log... to display **all messages** that ever appeared in the status bar.
ShuttleTracker - Logged messages
Info: Reading image files in /home/marek/ShuttleTrackerTutorial/March26_WellB_Pos09... done
Info: Preparing overlays... done
Info: Caching images in fast display buffers... done
Info: Parameter scan: precomputing shared image preprocessing steps...
Info: Number of parameter sets to scan: 104.
Info: Nuclei detection: N:97 SpN:28 nSp:7 SpO:1 D:9 [15 ms] done
Info: Nuclei detection settings loaded from WellB_Pos009_S001-nuc_detect.ini.
Info: Nuclei detection: N:119 SpN:8 nSp:9 SpO:1 D:4 [26 ms]
0:00
Settings ↔ file Save Load
Info: Nuclei detection settings stored in WellB_Pos009_S001-nuc_detect.ini.


File View Tools Script Preferences Window Help
Nuclear contours are drawn in different colors based on their handling by the splitting (“declumping”) procedure. The procedure works by exploiting convexity defects. Its behavior is controlled by widgets in the **Nuclei assessment** group box. Please consult the user’s manual to learn more.
Nuclei detection: declumped contours
Tracking
Tracks editing
Nuclei detection
Channel
blue
nuclear
cytosolic
Image preprocessing
Normalization
Denoising
1,0
Smoothing
3 px
Blob detection
Thresholding
Auto
Manual
block size
53 px
base-line
-13
Morphology: opening
repeats
x5
erosion
1 px
dilation
1 px
Nuclei assessment
min. solidity
0,96
min. area
0,35
max. area
0,93
split adjacent
min. indent
5 px
Detect nuclei
auto-click
preview stages
Settings ↔ file
Save
Load
0
[0..240]


File View Tools Script Preferences Window Help
Nuclei detection
Channel
cyan
nuclear • cytosolic
Image preprocessing
✓ Normalization
✓ Denoising 2
✓ Smoothing 5 px
✓ Morphology: closing
repeats x1
dilation 4 px
erosion 4 px
Edge detection
threshold low 120
thresholds ratio 4,0
tolerance 1,70
✓ reconstruct
Nuclei assessment
min. solidity 0,93
min. area 0,40
max. area 1,35
✓ split adjacent
min. indent 3 px
Detect nuclei
auto-click
preview stages
Just to get an idea, that's how the segmentation based on edge detection in the **cytoplasmic channel** may look like (the image of MCF-10A cells comes from another data set, and is courtesy of Nont Kosaisawe and John Albeck).
0
[0..120]
Image –
0:00
< >
Settings ↔ file
Save
Load

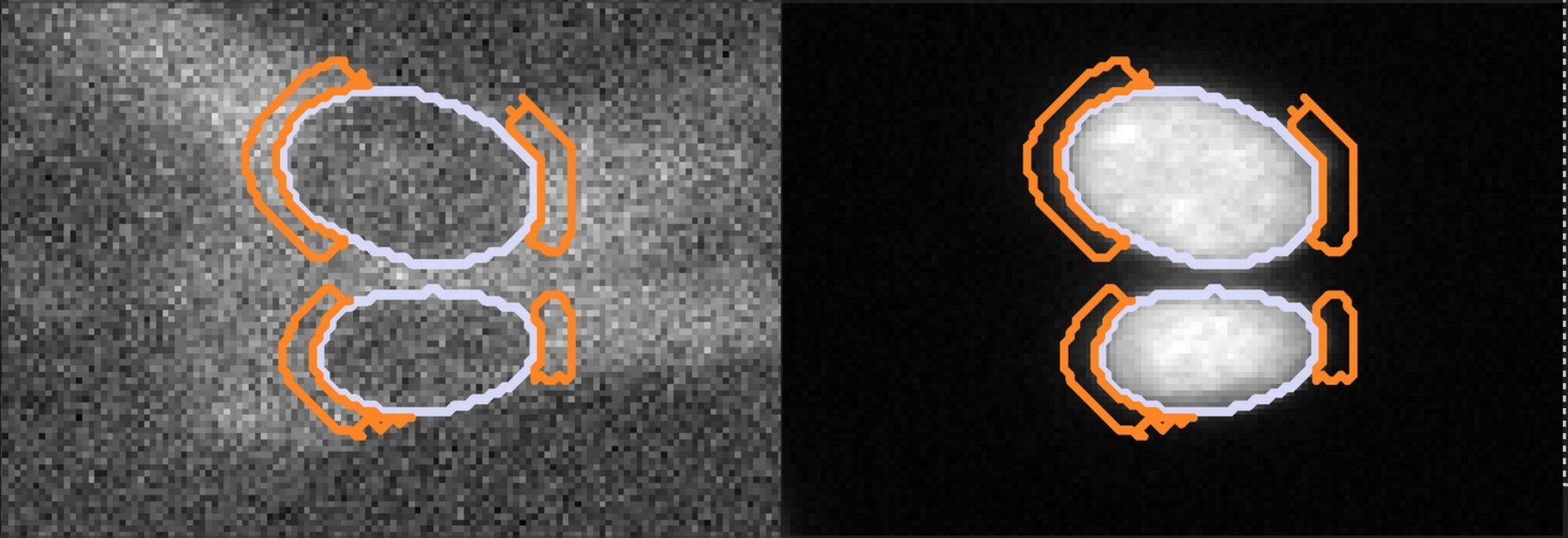

File View Tools Script Preferences Window Help
Image masking
Nuclei detection
Nuclei editing
Perinuclei
Regions editing
Quantification
Tracking
Tracks editing
The nuclei editing toolbox allows to, e.g., automatically filter out contours that are suspiciously small. Also, nuclei that lie on the border of the image can be removed or marked as “incomplete” (and saved for tracking purposes). Nuclei may be marked as “incomplete” also manually (just by double-clicking on the contour).
Nuclei editing
all
View/Edit by origin
Nuclei:
solo (detected)
split (detected)
manual
Non-nuclei (detected):
non-splittable
split orphans
debris
Edit by location
border: any
interior: debris
interior: orph's
Annotations
numbers
Contour properties
color by origin
width 1,6 px
145
[0..240]
580:00
Nuclei ↔ file
Save
Load


File View Tools Script Preferences Window Help
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
147 146 145
150 153 132
159 129 126
119 117 116 112 111 105 102 101
If necessary, segmentation issues can be **corrected manually**: nuclei can be removed by hand (left-click & right-click) or (re)drawn by hand (right-click, and then draw a new contour with the left button pressed). Manual editing is as convenient and responsive as in graphics programs.
One may toggle display of nuclei numbers. If the scene becomes busy (contours and textual annotations occlude the image), one may press keyboard spacebar to temporarily hide all markings.
Nuclei editing
all ×
View/Edit by origin
Nuclei:
✓ solo (detected)
✓ split (detected)
✓ manual
Non-nuclei (detected):
✓ non-splittable
✓ split orphans
✓ debris
Edit by location
border: any
interior: debris
interior: orph's
Annotations
✓ numbers
Contour properties
color by origin
width 1,6 px
145
[0..240]
580:00 < >
Nuclei ↔ file Save Load

File View Tools Script Preferences Window Help

Image masking | Nuclei detection | Nuclei editing | Perinuclei | Regions editing | Quantification | Tracking | Tracks editing

The **perinuclei toolbox** can generate configurable perinuclear annuli that are self-avoiding and also, optionally, background-avoiding.



Perinuclei

all ×

Settings

inner offset 0 px
ring width 6 px
neighbor avoid 3
overlap avoid 1
min. area 10 px
✓ Background avoid
channel green
expansion 3,5 σ
smoothing 1 px

Contour properties

width 1,4 px

Derive
✓ auto-click

"Derivation" of perinuclei takes only a **fraction of a second** so you may tick the check box **auto-click** and tweak parameter values with spin-boxes and see the resulting perinuclear contours nearly immediately. Final settings can be saved to/loaded from a text file.

224 [0..240]

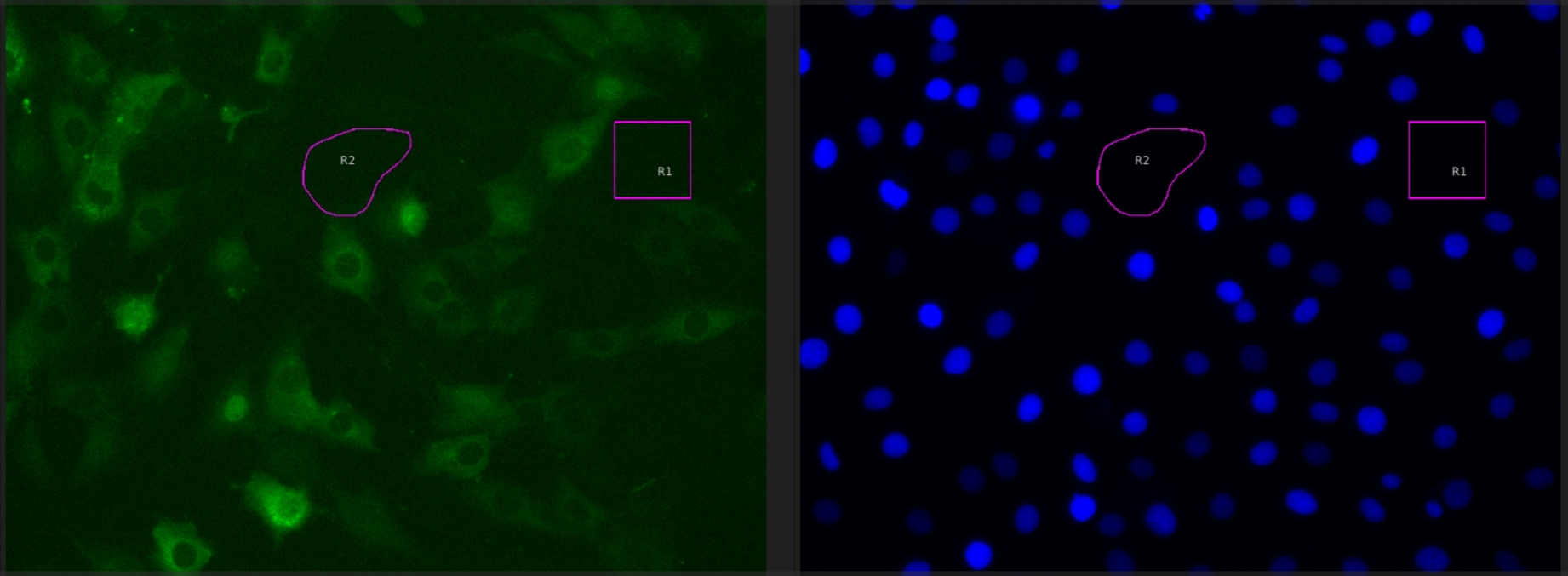
896:00 < > Settings ↔ file Save Load

Info: Perinuclei created (125 ms)

File View Tools Script Preferences Window Help

Image masking | Nuclei detection | Nuclei editing | Perinuclei | Regions editing | Quantification | Tracking | Tracks editing

The **regions toolbox** allows for marking any arbitrary regions of the image. Most often, the regions are used to sample image background.




The contours may be drawn by hand (right-click, then left-press) or, if background is to be sampled automatically, they can be proposed by the program. Regions' contours can be saved to/loaded from a text file.

**Regions editing**

all ×

Background

subdivisions 10
subshifts 1
count 1
Sample

Nuclei-centered Voronoi

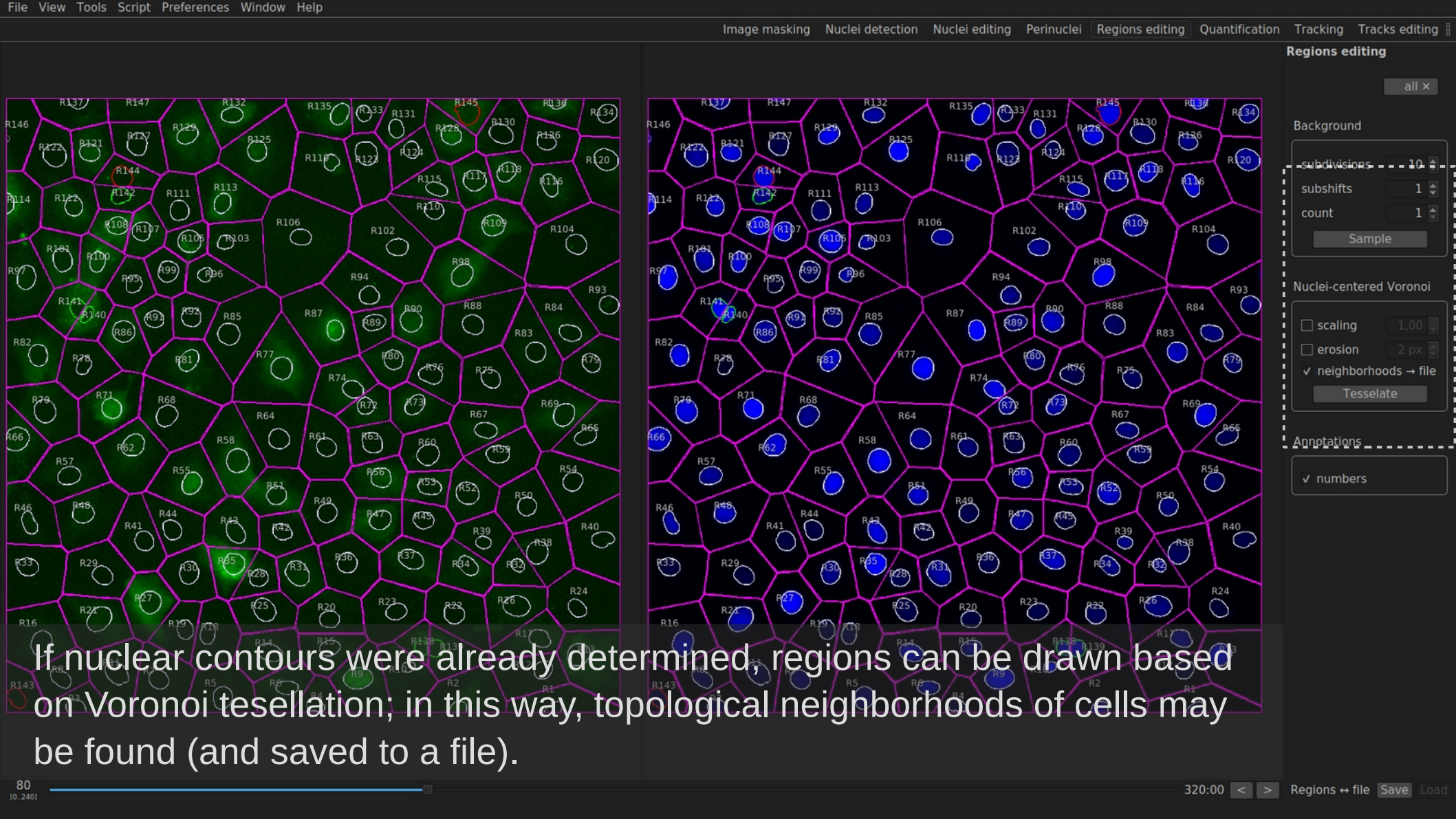
scaling 1,00
erosion 2 px
neighborhoods → file
Tesselate

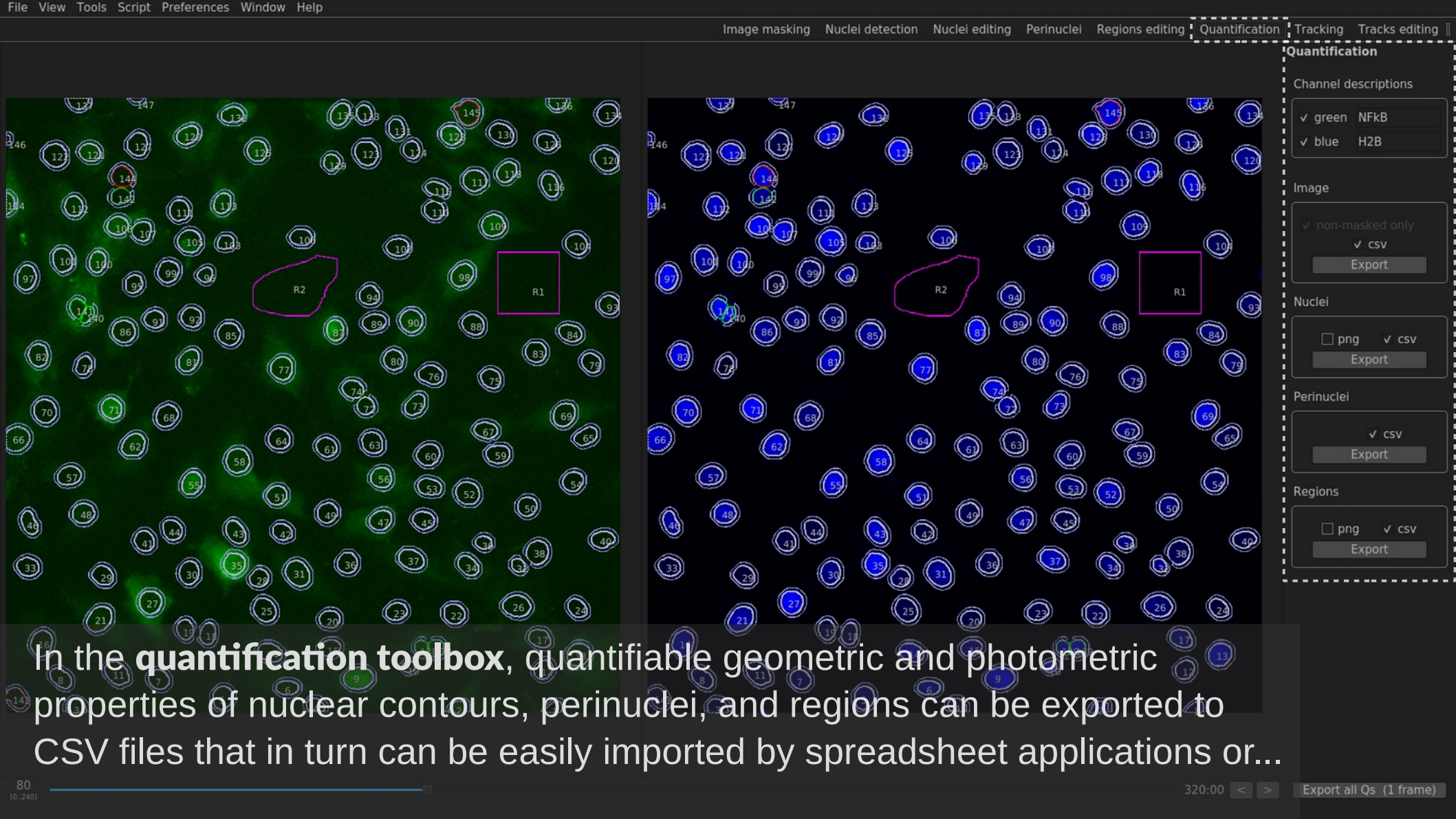
Annotations

✓ numbers

80 [0..240]

320:00 < > Regions ↔ file Save Load


File View Tools Script Preferences Window Help
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
Regions editing
all ×
Background
subdivisions 10
subshifts 1
count 1
Sample
Nuclei-centered Voronoi
scaling 1,00
erosion 2 px
✓ neighborhoods → file
Tesselate
Annotations
✓ numbers
If nuclear contours were already determined, regions can be drawn based on Voronoi tesellation; in this way, topological neighborhoods of cells may be found (and saved to a file).
80
[0..240]
320:00 < >
Regions ↔ file Save Load


File View Tools Script Preferences Window Help
Image masking
Nuclei detection
Nuclei editing
Perinuclei
Regions editing
Quantification
Tracking
Tracks editing
Quantification
Channel descriptions
green NFkB
blue H2B
Image
non-masked only
csv
Export
Nuclei
png
csv
Export
Perinuclei
csv
Export
Regions
png
csv
Export
R1
R2
In the quantification toolbox, quantifiable geometric and photometric properties of nuclear contours, perinuclei, and regions can be exported to CSV files that in turn can be easily imported by spreadsheet applications or...
80
[0..240]
320:00
Export all Qs (1 frame)

...or by your favorite data analysis environment (here, a Jupyter notebook is shown).

QuantsMinidemo - Mozilla Firefox

QuantsMinidemo

localhost:8888/notebooks/Quants

90%

Search

In [1]:
```
!pwd
```

```
/home/marek/ShuttleTrackerTutorial/March26_WellB_Pos09
```

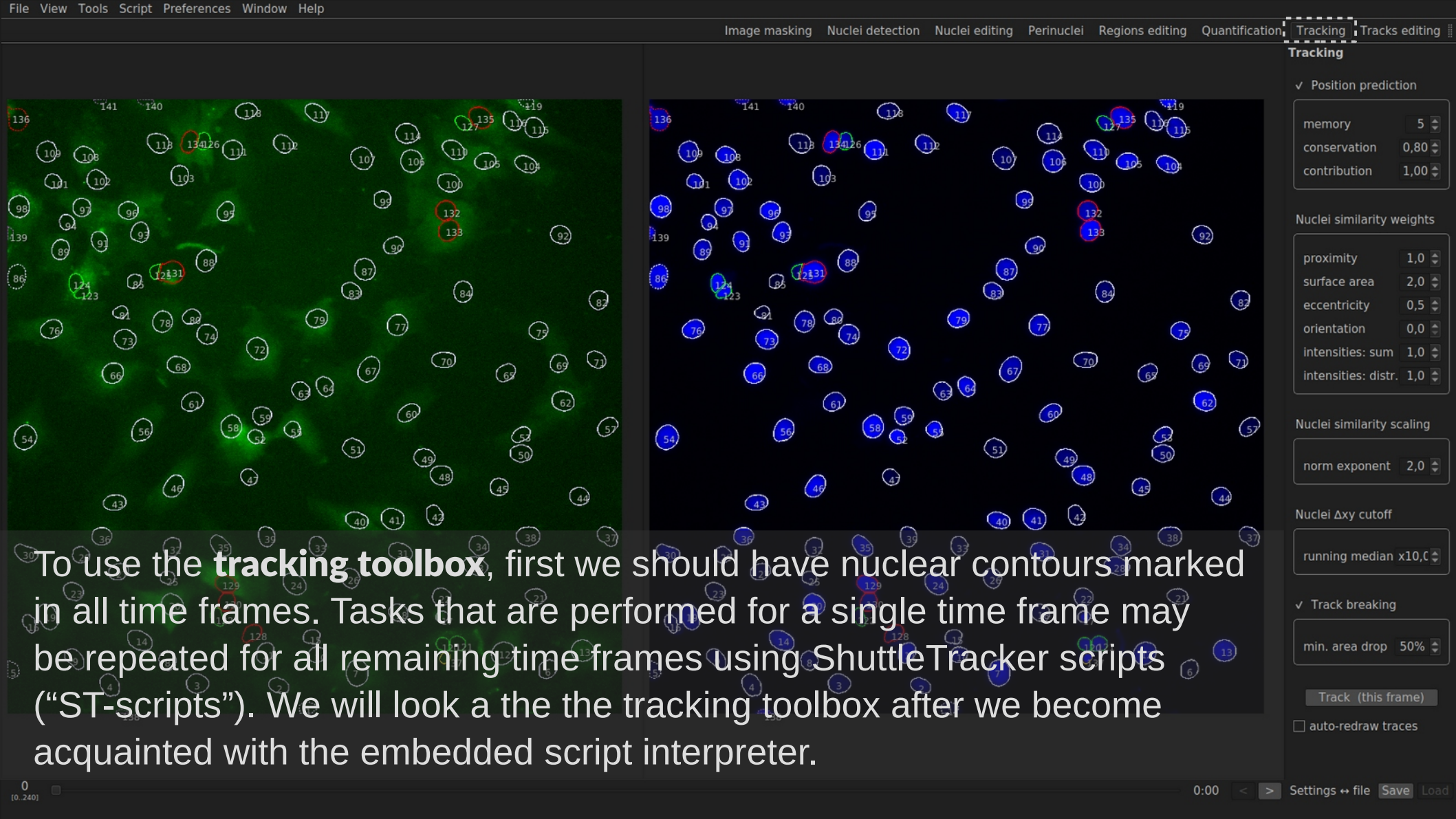
In [2]:
```
import pandas as pd
d = pd.read_csv('WellB_Pos009_S001_t080-nuc_quant.csv')
d.set_index('nucleus_id', inplace=True)
d.head()
```

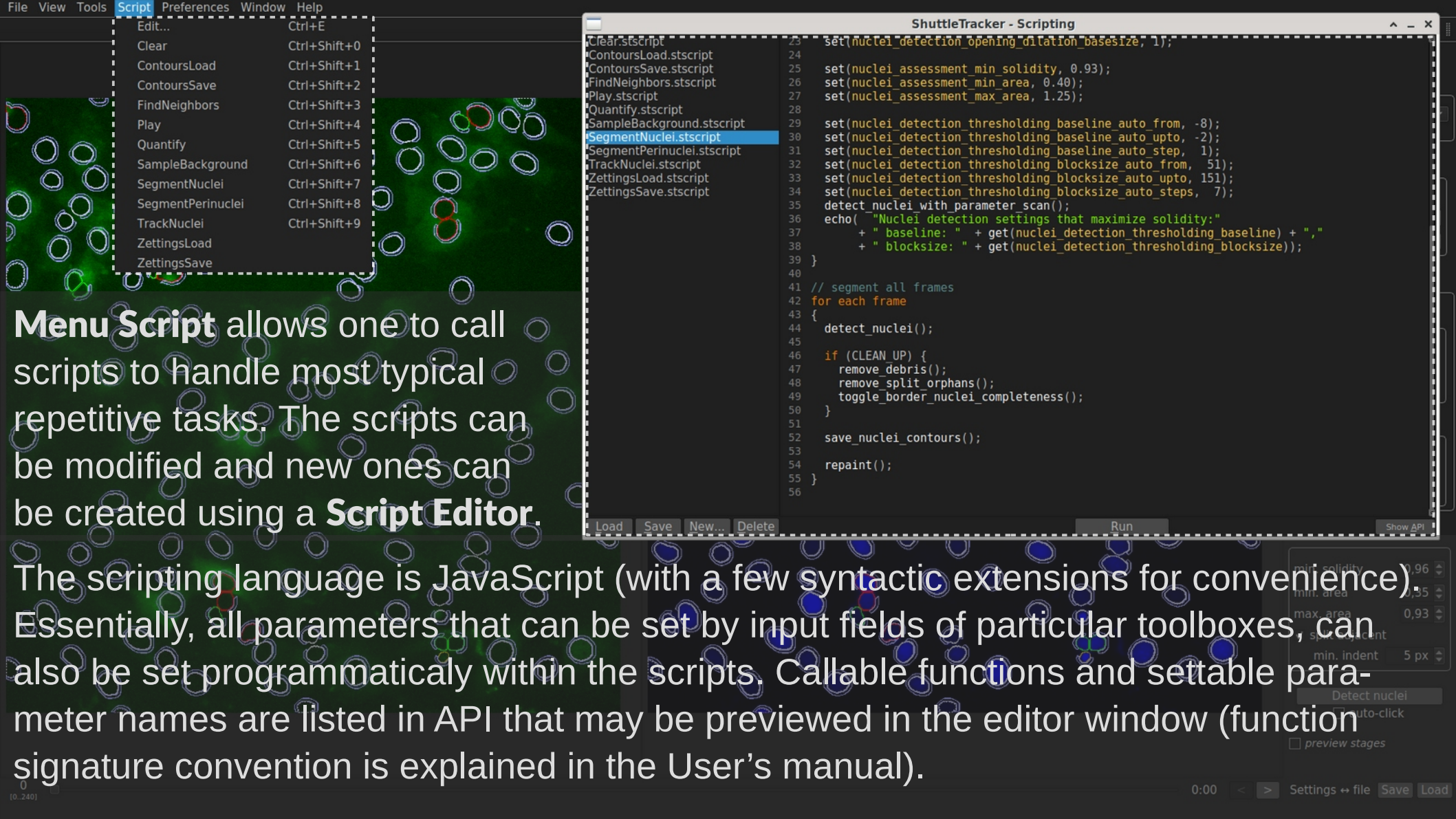
| nucleus_id | area | masked_area | eccentricity | is_complete | center_x | center_y | NFkB_intensity_min | NFkB_inte |
|---|---|---|---|---|---|---|---|---|
| 1 | 363.5 | 0.0 | 0.270 | 0 | 891.93 | 991.51 | 17 | 31 |
| 2 | 429.5 | 0.0 | 0.232 | 0 | 735.90 | 990.98 | 27 | 46 |
| 3 | 581.5 | 0.0 | 0.212 | 0 | 112.74 | 989.45 | 18 | 31 |
| 4 | 486.5 | 0.0 | 0.187 | 1 | 503.01 | 987.49 | 18 | 33 |
| 5 | 916.0 | 0.0 | 0.087 | 1 | 352.03 | 975.01 | 17 | 33 |

5 rows × 22 columns

In [4]:
```
', '.join(d.keys())
```

```
'area, masked_area, eccentricity, is_complete, center_x, center_y, NFkB_intensity_min, NFkB_intensity_max, NFkB_intensity
_median, NFkB_intensity_quartile1, NFkB_intensity_quartile3, NFkB_intensity_mean, NFkB_intensity_stddev, NFkB_intensities
_sum, H2B_intensity_min, H2B_intensity_max, H2B_intensity_median, H2B_intensity_quartile1, H2B_intensity_quartile3, H2B_i
ntensity_mean, H2B_intensity_stddev, H2B_intensities_sum'
```


File View Tools Script Preferences Window Help
Image masking Nuclei detection Nuclei editing Perinuclei Regions editing Quantification Tracking Tracks editing
Tracking
✓ Position prediction
memory 5
conservation 0,80
contribution 1,00
Nuclei similarity weights
proximity 1,0
surface area 2,0
eccentricity 0,5
orientation 0,0
intensities: sum 1,0
intensities: distr. 1,0
Nuclei similarity scaling
norm exponent 2,0
Nuclei Δxy cutoff
running median x10,0
✓ Track breaking
min. area drop 50%
Track (this frame)
auto-redraw traces
0
[0..240]
0:00
Settings ↔ file
Save
Load
To use the tracking toolbox, first we should have nuclear contours marked in all time frames. Tasks that are performed for a single time frame may be repeated for all remaining time frames using ShuttleTracker scripts ("ST-scripts"). We will look a the the tracking toolbox after we become acquainted with the embedded script interpreter.




**Menu Script** allows one to call scripts to handle most typical repetitive tasks. The scripts can be modified and new ones can be created using a **Script Editor**.

The scripting language is JavaScript (with a few syntactic extensions for convenience). Essentially, all parameters that can be set by input fields of particular toolboxes, can also be set programmaticaly within the scripts. Callable functions and settable parameter names are listed in API that may be previewed in the editor window (function signature convention is explained in the User's manual).

File View Tools Script Preferences Window Help

Image masking | Nuclei detection | Nuclei editing | Perinuclei | Regions editing | Quantification | Tracking | Tracks editing

**Nuclei detection**

Channel

blue

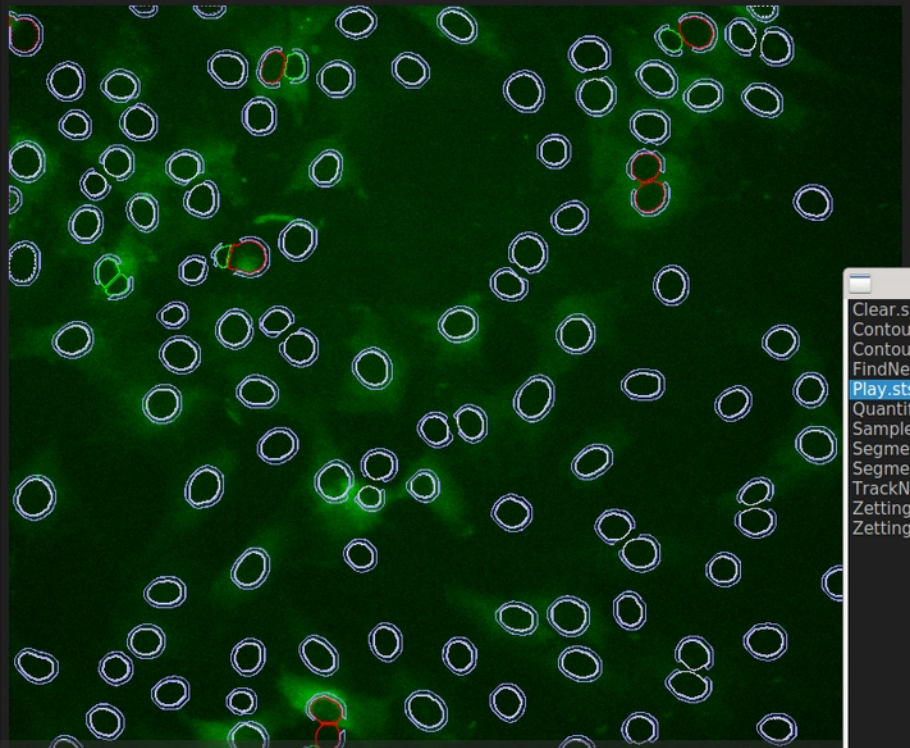
• nuclear ○ cytosolic

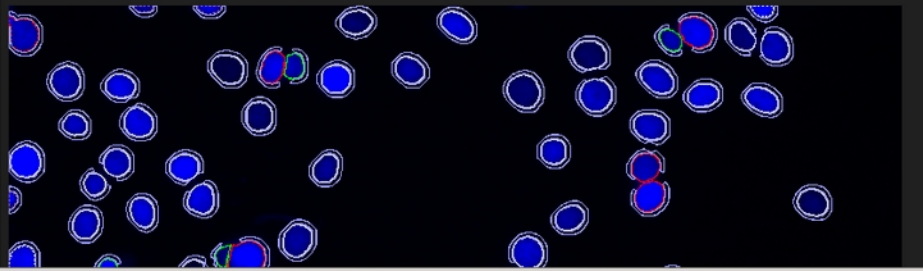
Image preprocessing

✓ Normalization

☐ Denoising 1,0

✓ Smoothing 3 px





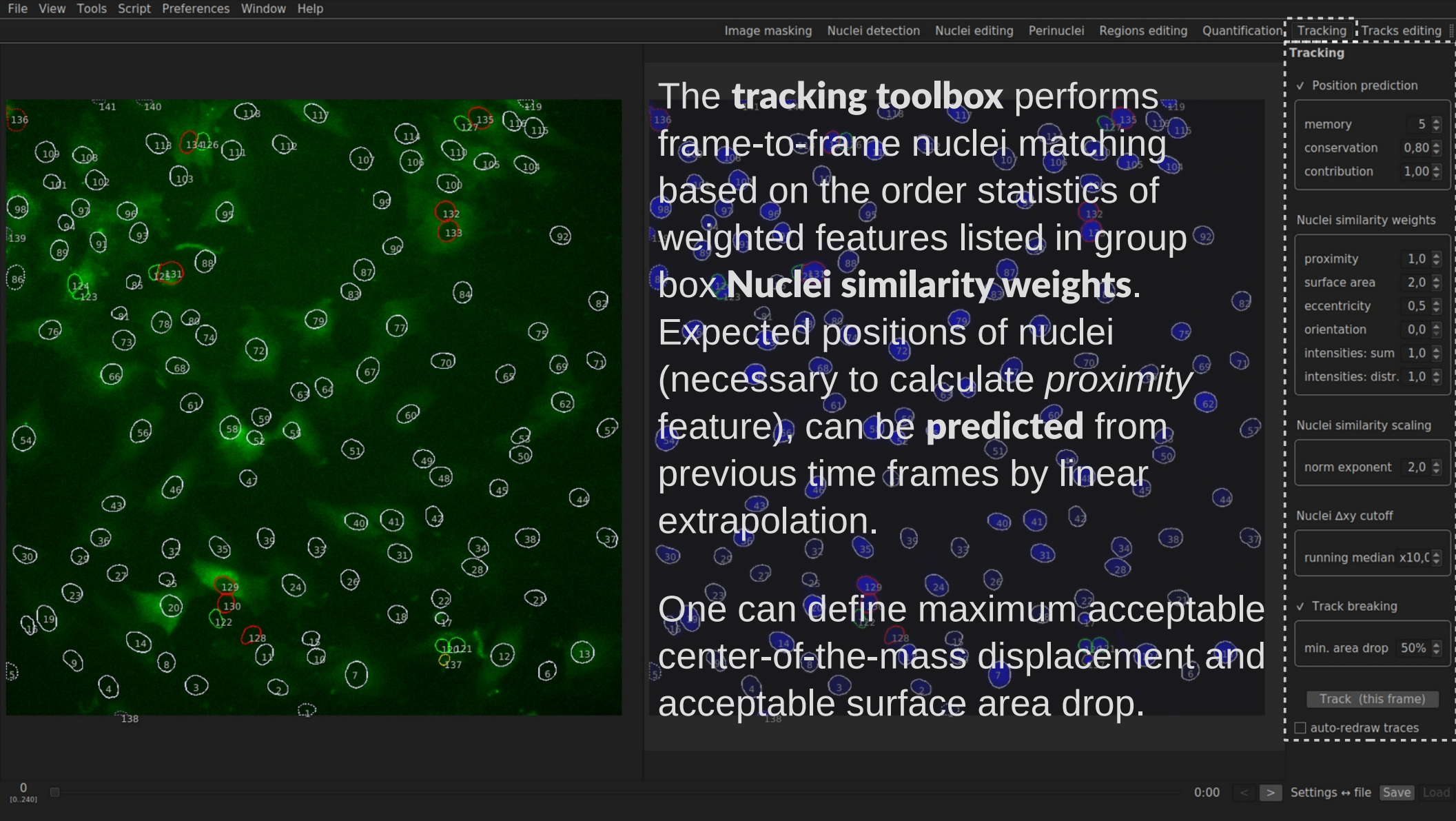
Of note, the interpreter has access to all standard JavaScript functionalities, not only these explicitly specified in API.

**ShuttleTracker - Scripting**

Clear.stscript
ContoursLoad.stscript
ContoursSave.stscript
FindNeighbors.stscript
Play.stscript
Quantify.stscript
SampleBackground.stscript
SegmentNuclei.stscript
SegmentPerinuclei.stscript
TrackNuclei.stscript
ZettingsLoad.stscript
ZettingsSave.stscript

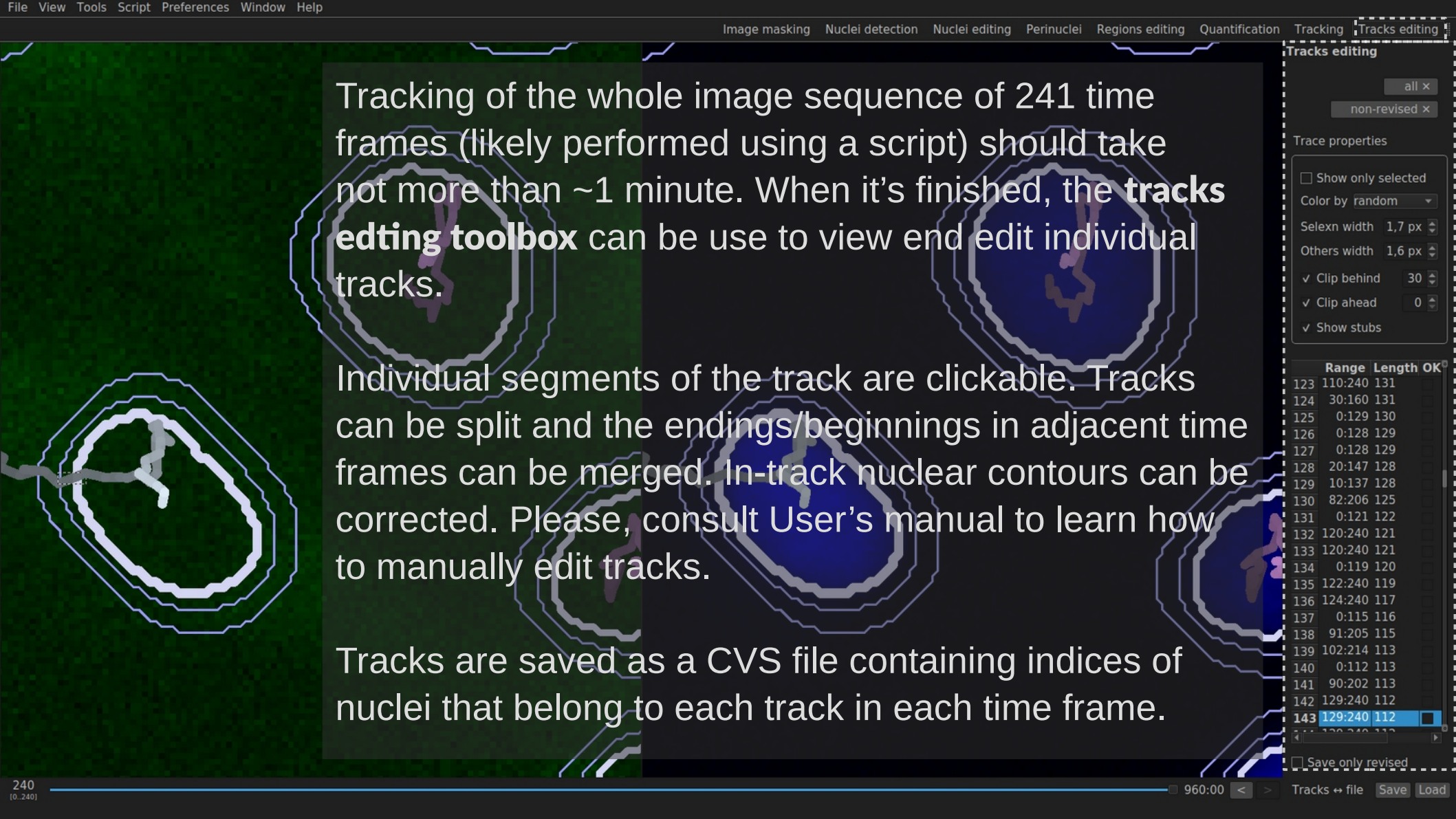
```
1  ////
2  //// Purpose: Play the whole image sequence.
3  ////
4
5  // The smaller the interval, the faster are images displayed.
6  var frames_per_second = 15;                        // intended speed
7  const ms_per_second = 1000;                        // 1 s = 1000 ms
8  const interval = ms_per_second/frames_per_second; // unit: millisecond
9
10 // Remember the frame at which the script has been called.
11 const current = current_frame();
12 const t_start = (new Date()).getTime();  // tic!
13
14 for each frame {
15   //echo("Displaying frame " + current_frame());
16   repaint();
17   millisleep(interval);
18 }
19
20 // Return to the frame at which the script has been called.
21 const t_stop = (new Date()).getTime();  // toc!
22 go_to_frame(current);
23
24 const t_interval = t_stop - t_start;
25 if (t_interval > 0) {
26   const measured_fps = frames_count()/(t_interval/ms_per_second);
27   echo("Image display effective speed: ~" + Math.round(measured_fps) + " FPS.");
28 }
29
```

Load | Save | New... | Delete | Run | Show API




The **tracking toolbox** performs frame-to-frame nuclei matching based on the order statistics of weighted features listed in group box **Nuclei similarity weights**. Expected positions of nuclei (necessary to calculate *proximity* feature), can be **predicted** from previous time frames by linear extrapolation.

One can define maximum acceptable center-of-the-mass displacement and acceptable surface area drop.



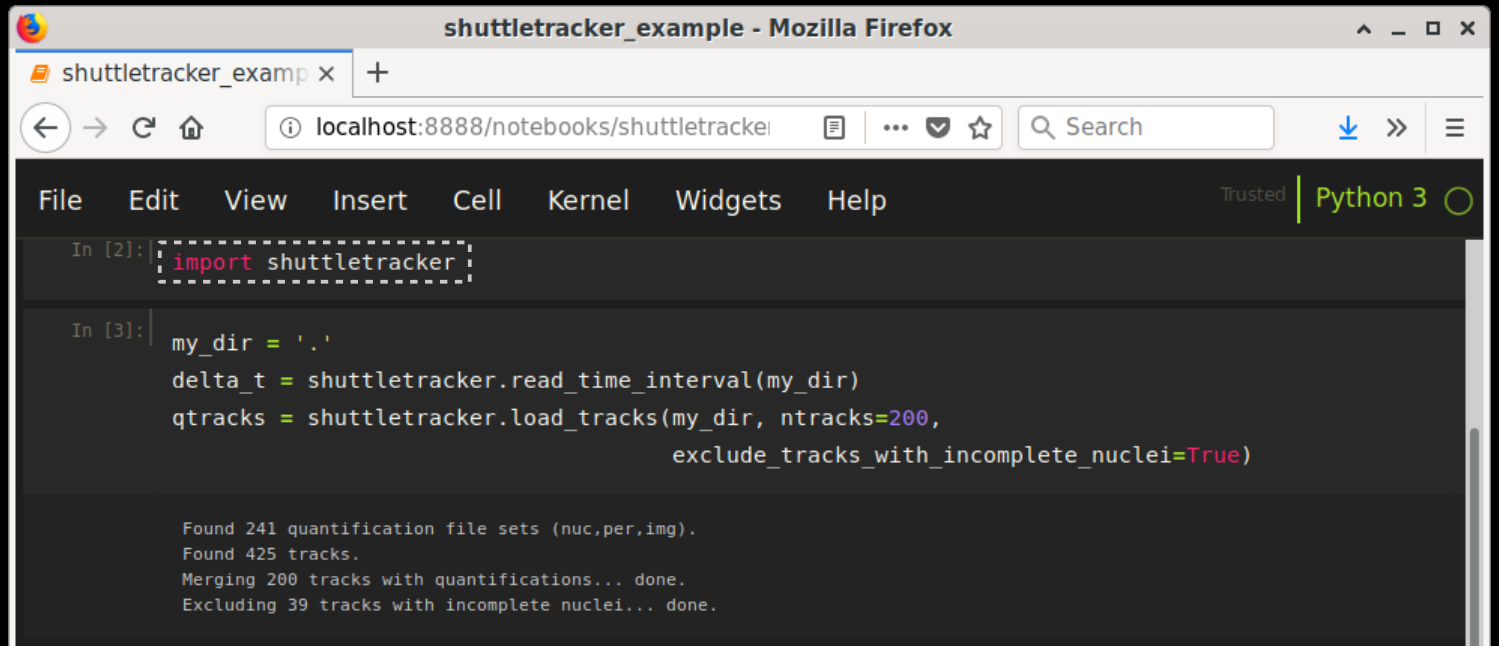

Tracking of the whole image sequence of 241 time frames (likely performed using a script) should take not more than ~1 minute. When it's finished, the **tracks edting toolbox** can be use to view end edit individual tracks.

Individual segments of the track are clickable. Tracks can be split and the endings/beginnings in adjacent time frames can be merged. In-track nuclear contours can be corrected. Please, consult User's manual to learn how to manually edit tracks.

Tracks are saved as a CVS file containing indices of nuclei that belong to each track in each time frame.

Each track can be easily inspected visually and marked as revised in the table view of the tracks editing toolbox (column ‘OK’). Comprehensive analysis of tracks in the context of quantified features of nuclei and other contours has been delegated to external scripts, which grants the user full flexibility in looking at the data.

Tracks and nuclei quantifications can be joined and analyzed using a **Python module** that is distributed with the binary executables and the source code package.



Four NF-$\varkappa$B nuclear translocation events can be distinguished in a **plot** of ⟨NF-$\varkappa$B nuclear⟩/⟨NF-$\varkappa$B perinuclear⟩.

shuttletracker_example - Mozilla Firefox

shuttletracker_examp

localhost:8888/notebooks/shuttletracke

Search

File Edit View Insert Cell Kernel Widgets Help

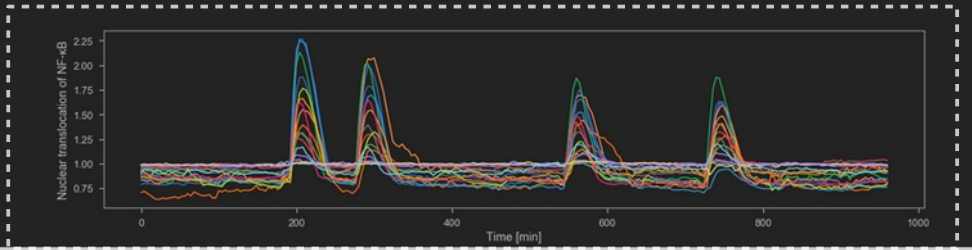
Trusted | Python 3

In [2]:
```
import shuttletracker
```

In [3]:
```
my_dir = '.'
delta_t = shuttletracker.read_time_interval(my_dir)
qtracks = shuttletracker.load_tracks(my_dir, ntracks=200,
                                     exclude_tracks_with_incomplete_nuclei=True)
```

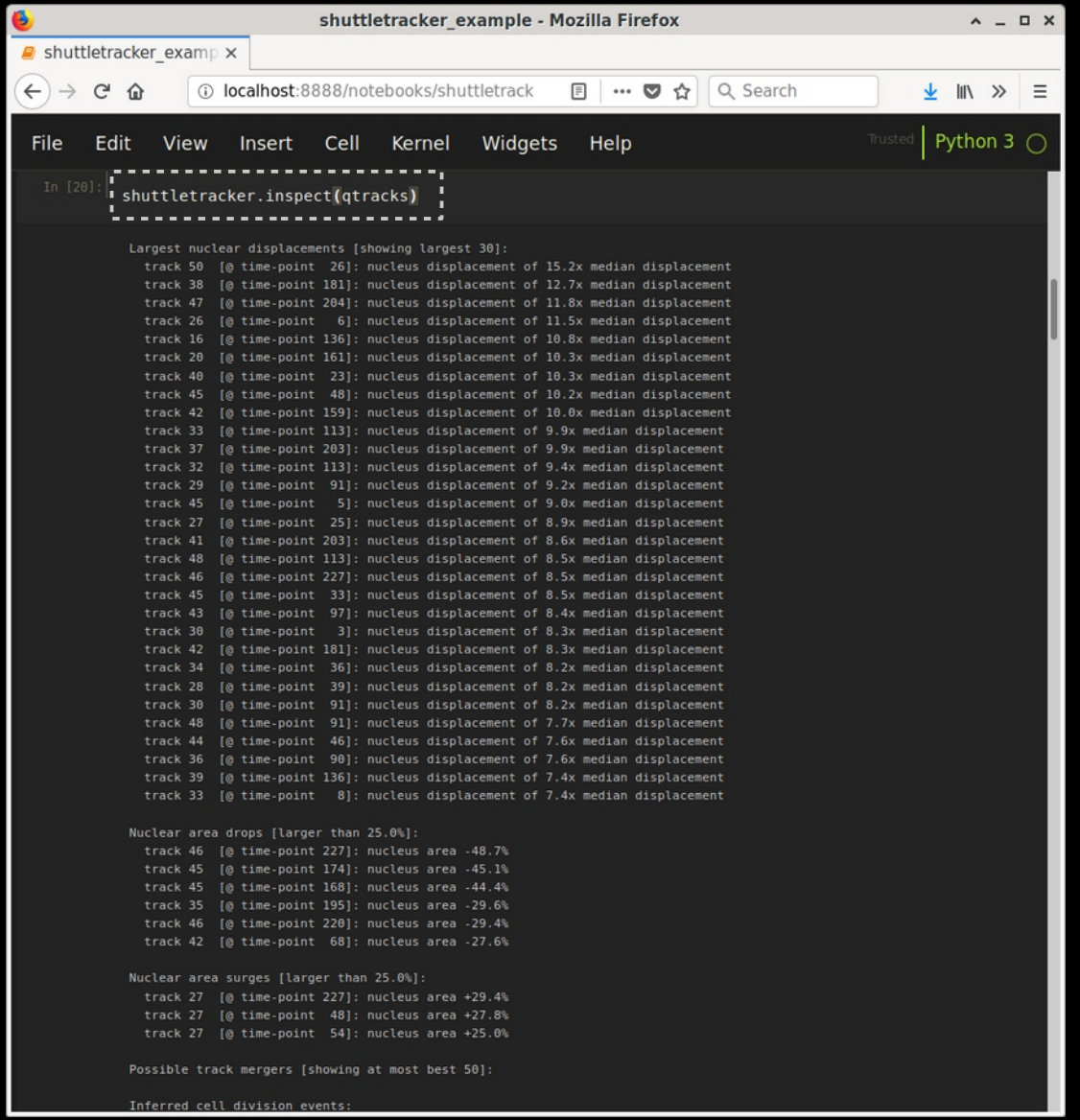
```
Found 241 quantification file sets (nuc,per,img).
Found 425 tracks.
Merging 200 tracks with quantifications... done.
Excluding 39 tracks with incomplete nuclei... done.
```

In [10]:
```
%matplotlib inline
import matplotlib.pyplot as plot

n_curves = 20
mins_per_hour = 60

fig, ax = plot.subplots(1, 1, figsize=(18, 4))
ax.set_xlabel('Time [min]')
ax.set_ylabel('Nuclear translocation of NF-κB')

for track_name, q, is_revised in qtracks[20:20+n_curves]:
    x = q['time_point_index']*delta_t/mins_per_hour
    y = q['nuc_NFkB_intensity_mean']/q['per_NFkB_intensity_mean']
    ax.plot(x, y, label='track ' + track_name)
```



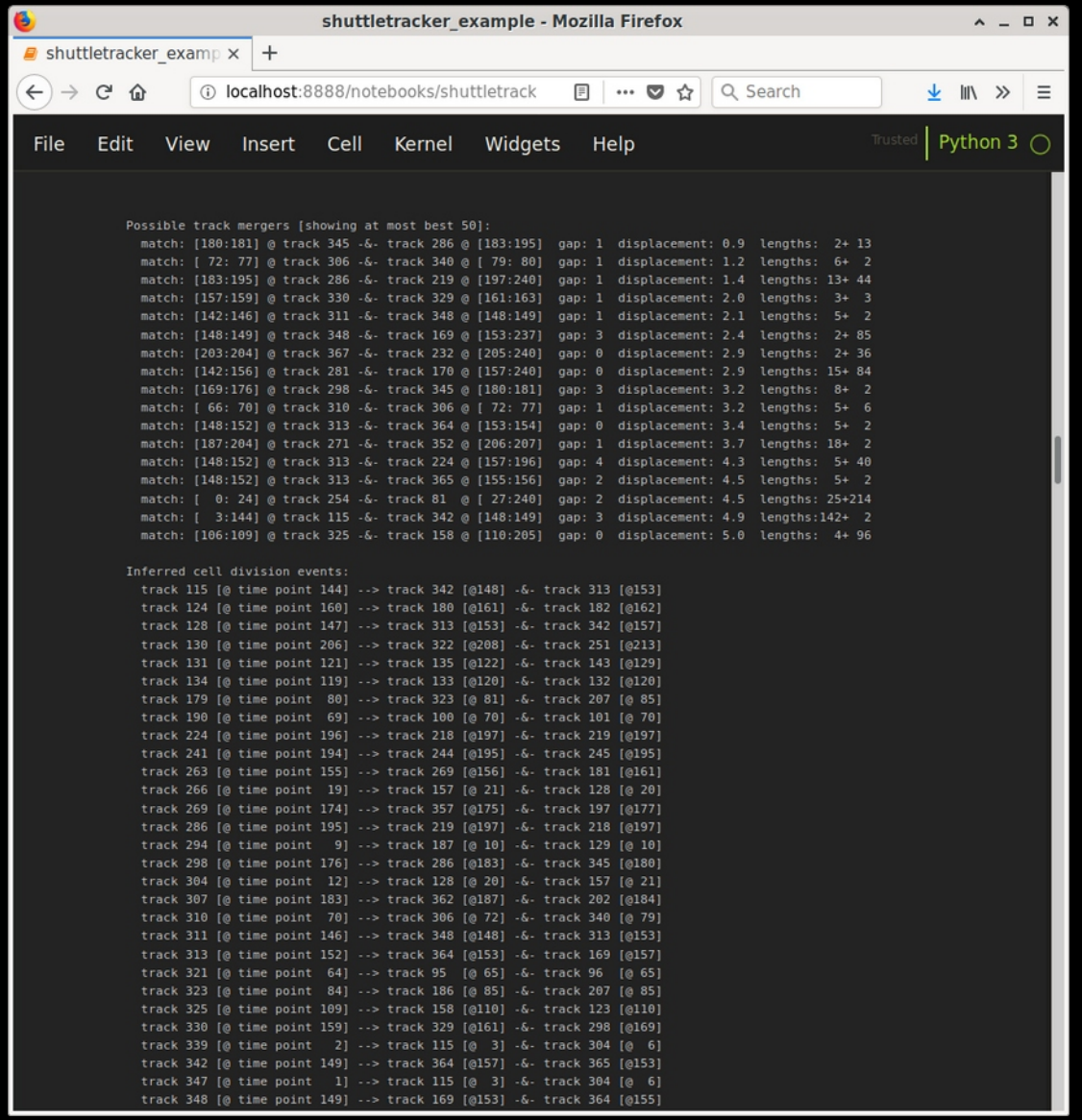
The module aids priority fixing of the most outstanding errors in segmentation or tracking, such as suspiciously large displacements or unusual nuclear contour surface area variation.

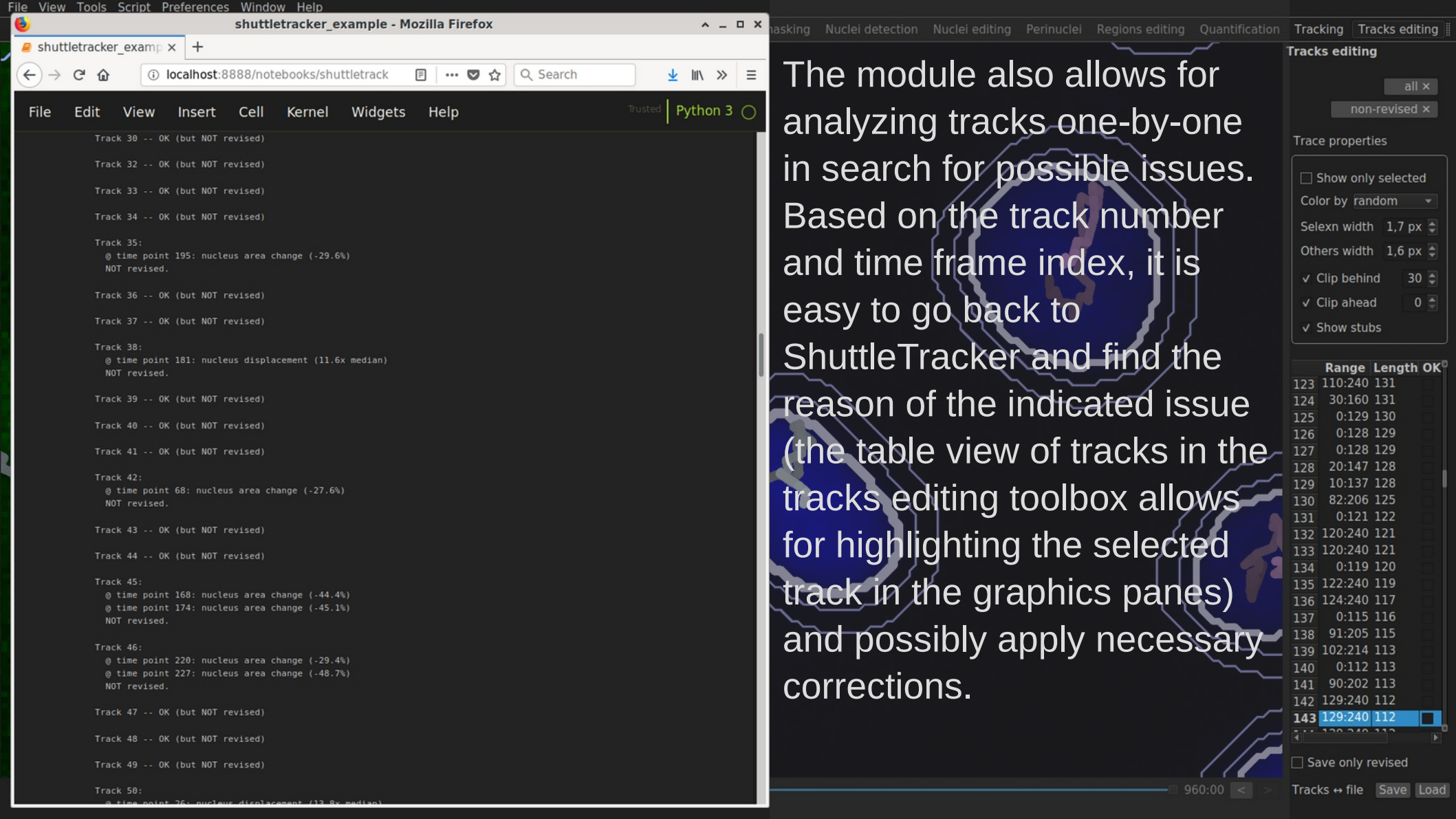
Also, based on quantified features, cell lineage can be deduced.



Gaps in tracks, that require manual intervention, are suggested.

Based on quantified features, cell lineage is deduced.






The module also allows for analyzing tracks one-by-one in search for possible issues. Based on the track number and time frame index, it is easy to go back to ShuttleTracker and find the reason of the indicated issue (the table view of tracks in the tracks editing toolbox allows for highlighting the selected track in the graphics panes) and possibly apply necessary corrections.

# Thank you for your interest in ShuttleTracker!

Questions, bug reports, and feature requests should be directed to:
shuttletracker.software@gmail.com

http://pmbm.ippt.pan.pl/software/shuttletracker