



# ShuttleTracker

## Tutorial and overview

featuring version 1.3.0 (September 2019)

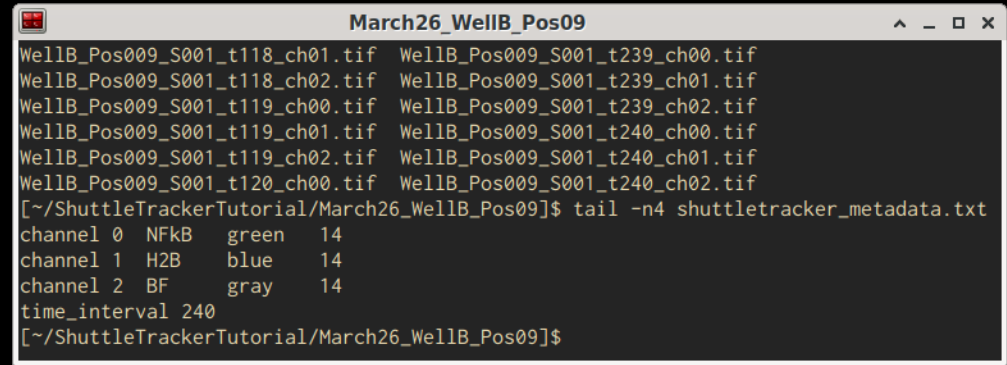
<http://pmbm.ippt.pan.pl/software/shuttletracker>

In this tutorial, we are going to process images in which nuclear translocation of a transcription factor, NF- $\kappa$ 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](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:

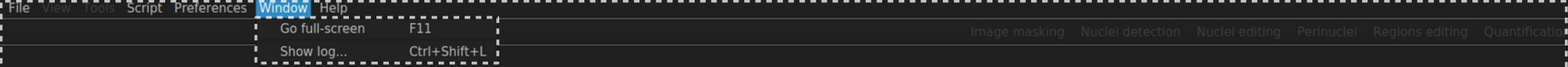
- suffix ch00 – GFP-labeled NF- $\kappa$ B subunit,
- suffix ch01 – nuclear signal from a fluorescently labeled histone,
- suffix ch02 – bright-field view.

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

A terminal window titled "March26\_WellB\_Pos09" showing a list of TIFF files and the output of a tail command on a metadata file. The files are organized in two columns, showing time points t118, t119, t120, t239, and t240 for channels ch00, ch01, and ch02. The metadata output shows channel 0 (NFkB, green, 14 bits), channel 1 (H2B, blue, 14 bits), channel 2 (BF, gray, 14 bits), and a time interval of 240.

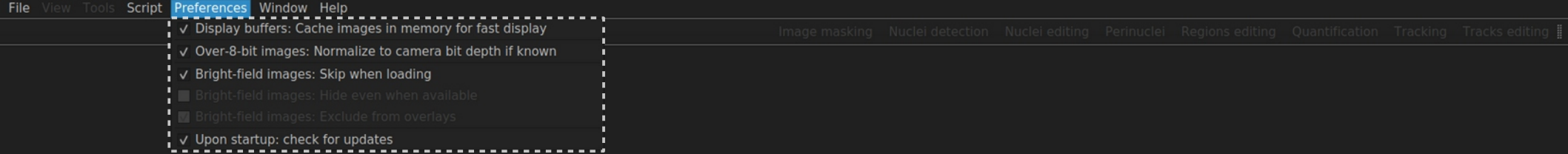
```
March26_WellB_Pos09
WellB_Pos009_S001_t118_ch01.tif  WellB_Pos009_S001_t239_ch00.tif
WellB_Pos009_S001_t118_ch02.tif  WellB_Pos009_S001_t239_ch01.tif
WellB_Pos009_S001_t119_ch00.tif  WellB_Pos009_S001_t239_ch02.tif
WellB_Pos009_S001_t119_ch01.tif  WellB_Pos009_S001_t240_ch00.tif
WellB_Pos009_S001_t119_ch02.tif  WellB_Pos009_S001_t240_ch01.tif
WellB_Pos009_S001_t120_ch00.tif  WellB_Pos009_S001_t240_ch02.tif
[~/ShuttleTrackerTutorial/March26_WellB_Pos09]$ tail -n4 shuttletracker_metadata.txt
channel 0  NFkB   green   14
channel 1  H2B    blue    14
channel 2  BF     gray    14
time_interval 240
[~/ShuttleTrackerTutorial/March26_WellB_Pos09]$
```

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).

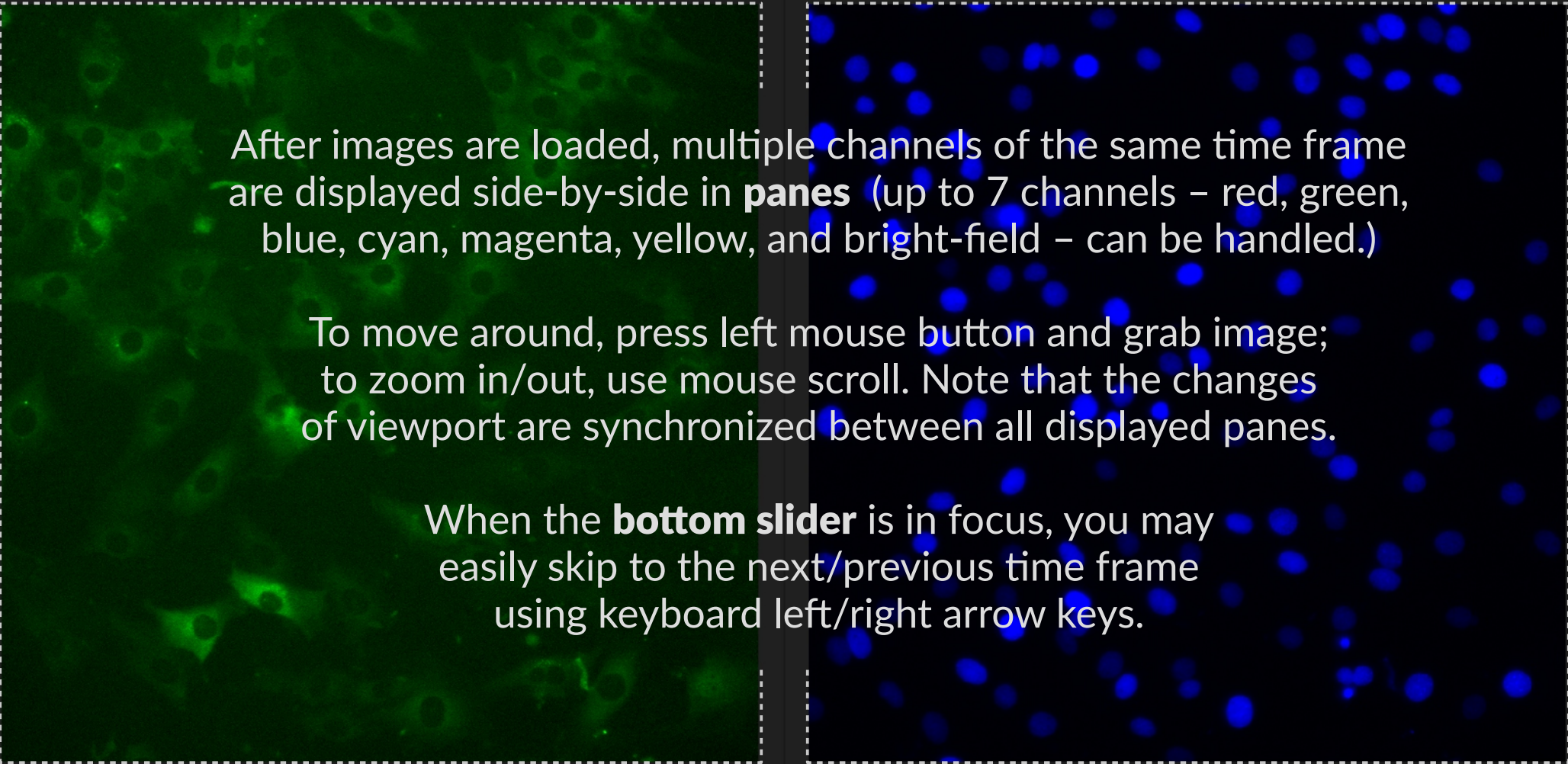


Before we load images, in **menu Preferences**, we may:

- 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.)

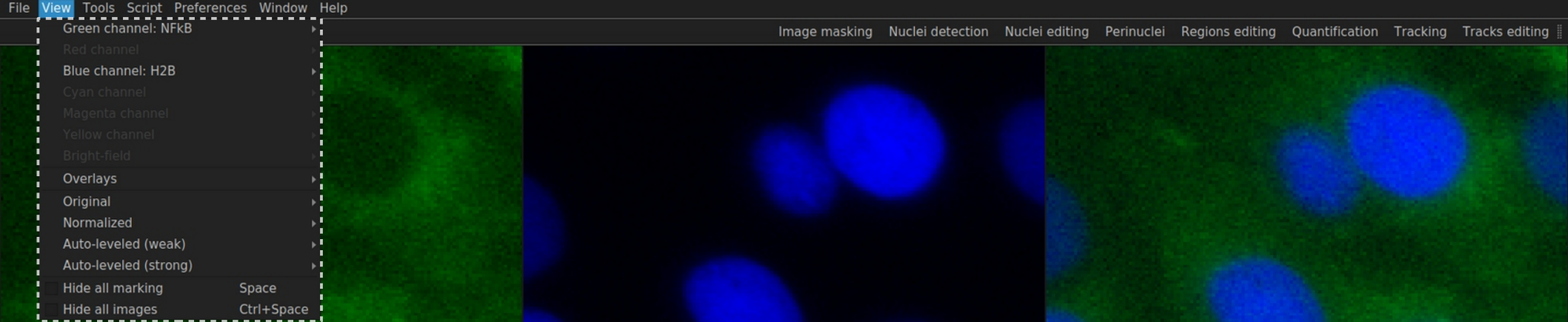




After images are loaded, multiple channels of the same time frame are displayed side-by-side in **panes** (up to 7 channels – red, green, blue, cyan, magenta, yellow, and bright-field – can be handled.)

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.



Display style can be altered using **menu View**, where you can, *e.g.*, stretch contrast, show all channels in grayscale, and compose overlays.



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

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).

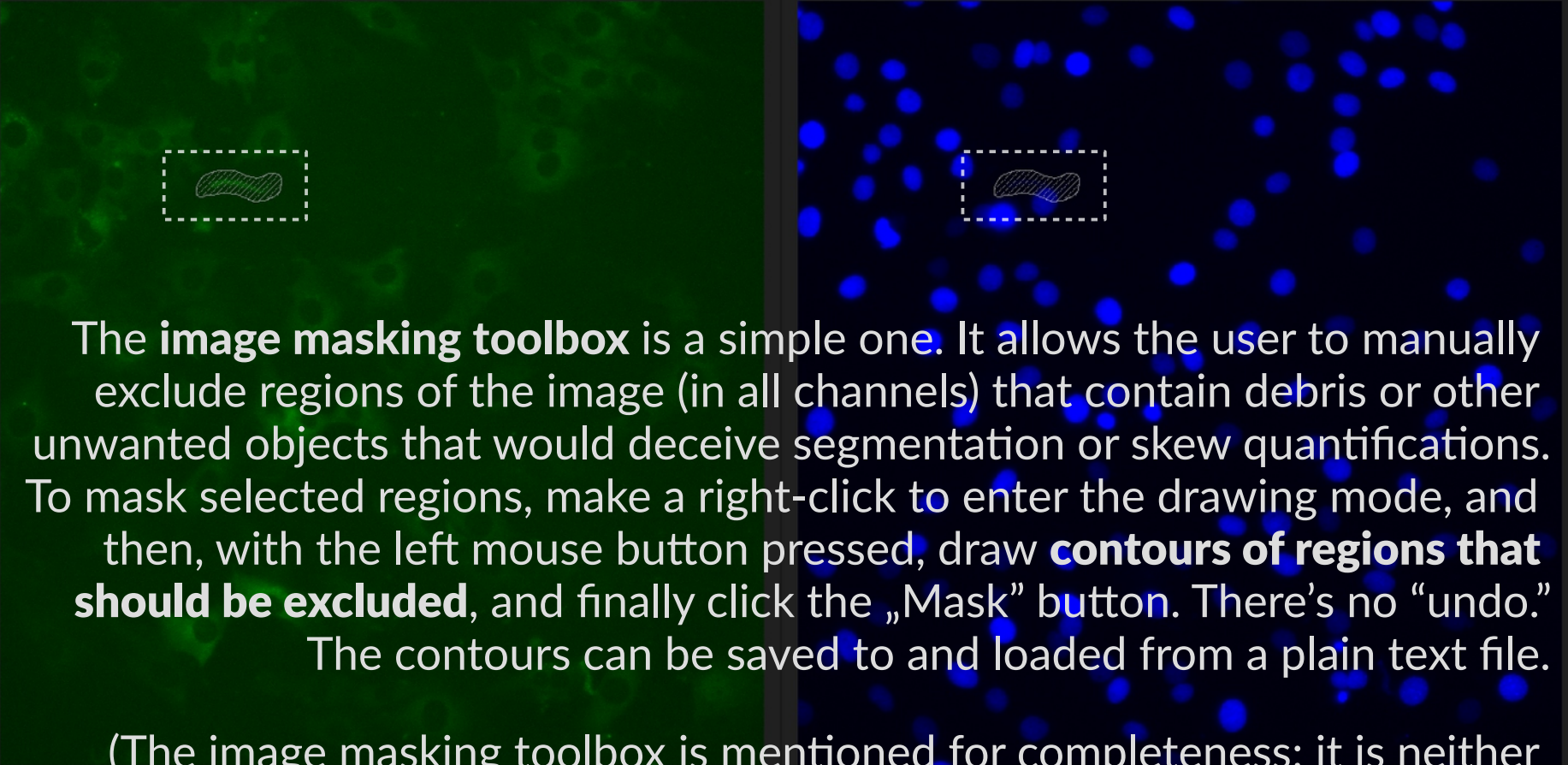
## ShuttleTracker Manual

## ShuttleTracker – User's manual

## Table of contents:

- Overview
  - Capabilities
- Launching
  - Launching with GUI
  - Launching from terminal
- Input images
  - File names
  - Image formats
  - Bit depth
  - Metadata
- Viewing
  - Navigation
  - Enhanced view
- Toolboxes
  - Image masking
  - Nuclei detection
  - Nuclei editing
  - Perinuclei derivation
  - Regions editing
  - Quantification
  - Tracking
  - Tracks editing
- Scripting
- Preferences
- Further analysis
- Getting help
- Credits
- Appendix A: Programming interface
- Appendix B: Hints and troubleshooting

(As this tutorial is 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.)

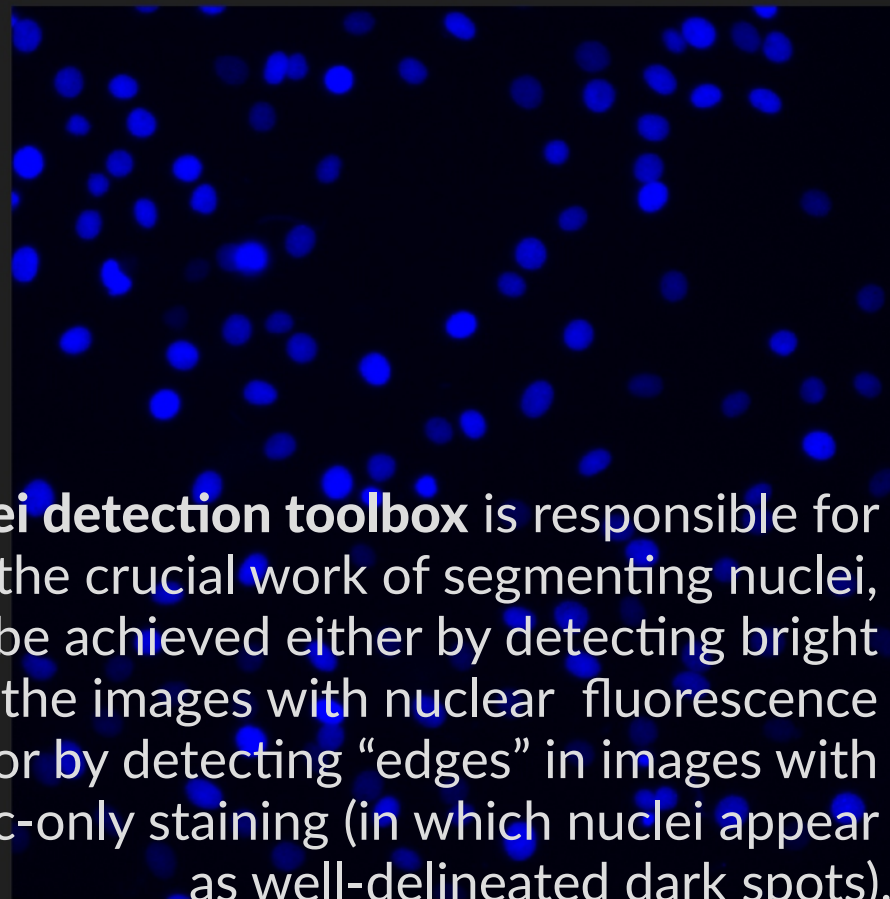
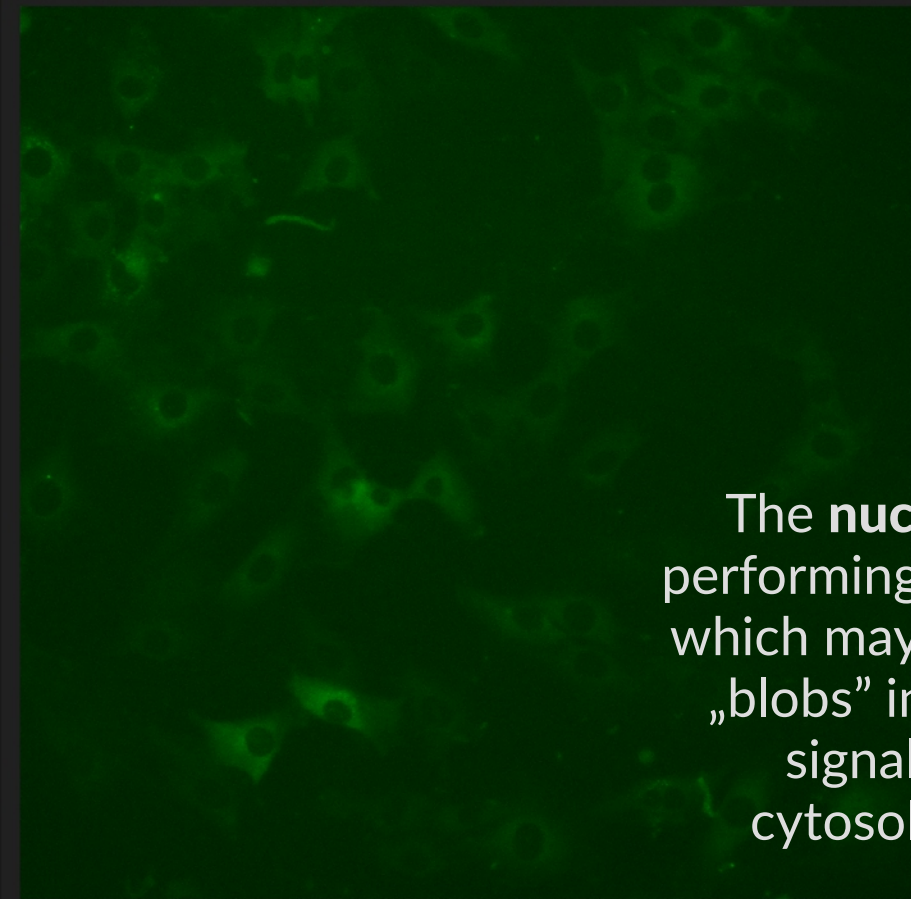


The **image masking toolbox** is a simple one. It allows the user to manually exclude regions of the image (in all channels) that contain debris or other unwanted objects that would deceive segmentation or skew quantifications. To mask selected regions, make a right-click to enter the drawing mode, and 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.)





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).

**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

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).

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]

**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

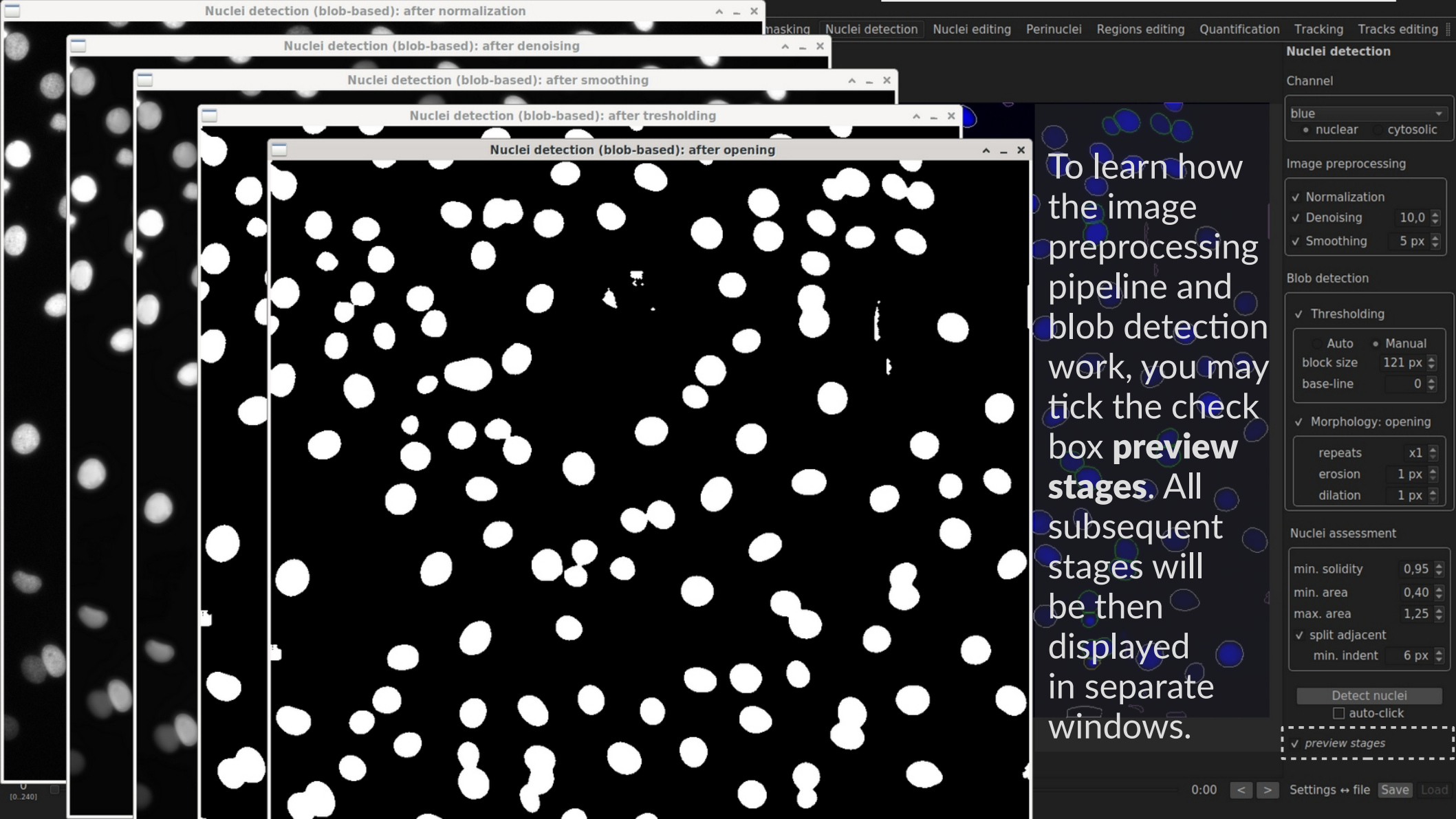
max. area 1,25

☒ split adjacent

min. indent 6 px

Detect nuclei

☐ auto-click☐ preview stages



Nuclei detection (blob-based): after normalization

Nuclei detection (blob-based): after denoising

Nuclei detection (blob-based): after smoothing

Nuclei detection (blob-based): after thresholding

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

To learn how the image preprocessing pipeline and blob detection work, you may tick the checkbox **preview stages**. All subsequent stages will be then displayed in separate windows.



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 **thresholding parameters**, 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.

## 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 px

#steps 8

base-line -9

from -10

up to 2

every 1

objective max solidity

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

## Nuclei detection

Channel

blue

☒ nuclear ☐ cytosolic

Image preprocessing

☒ Normalization☐ Denoising

1.0

☒ Smoothing

3 px

Blob detection

☒ ThresholdingAuto ☒ 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.



## Nuclei detection

Channel

blue

☒ nuclear ☐ cytosolic

Image preprocessing

☒ Normalization☐ Denoising

1.0

☒ Smoothing

3 px

Blob detection

☒ Thresholding

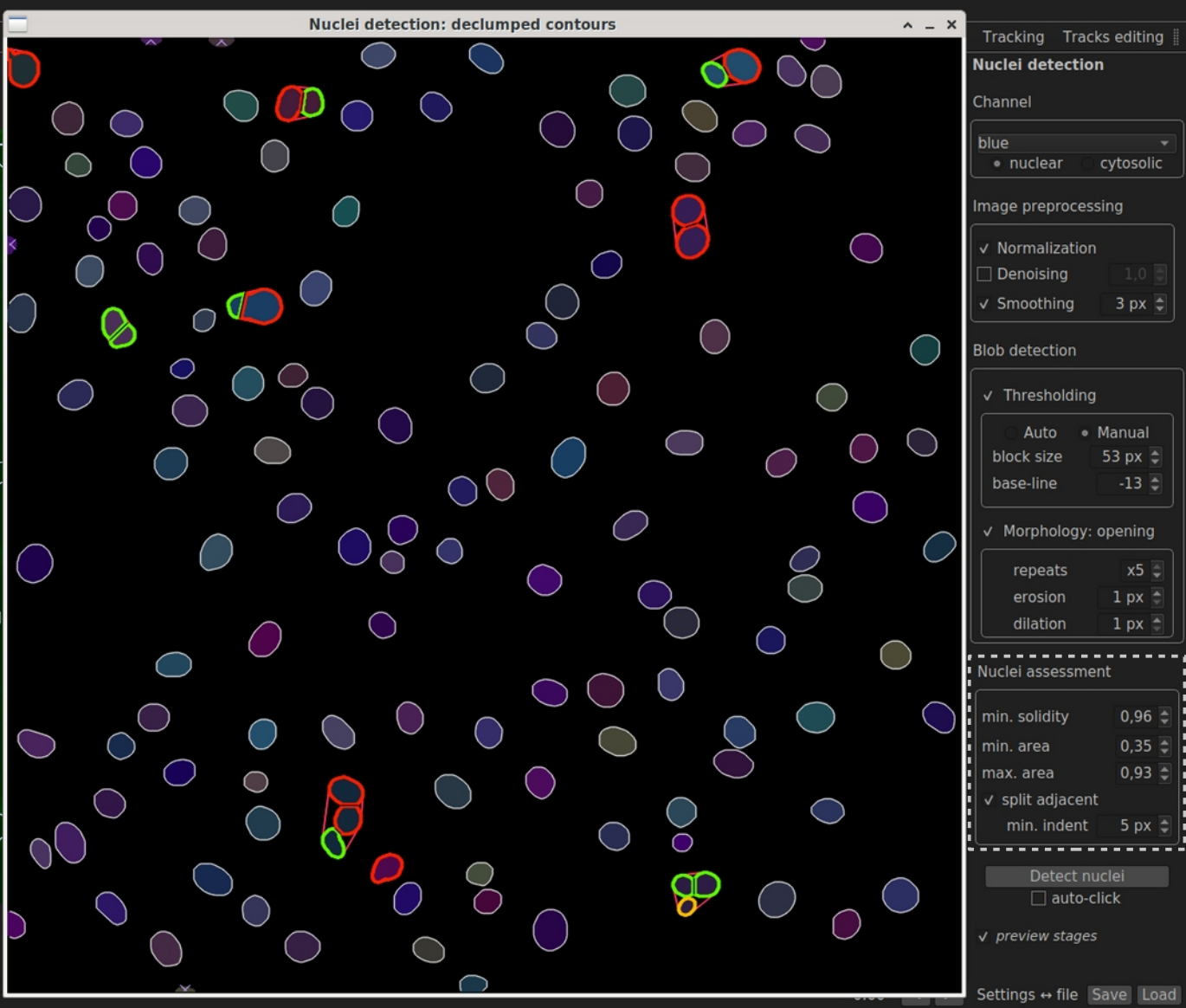
The nuclei detection settings can be **saved** to/**loaded** from a plain text file. The name of the file is displayed in the **status bar**.

By the way, you may use menu Window → Show log... to display **all messages** that ever appeared in the status bar.

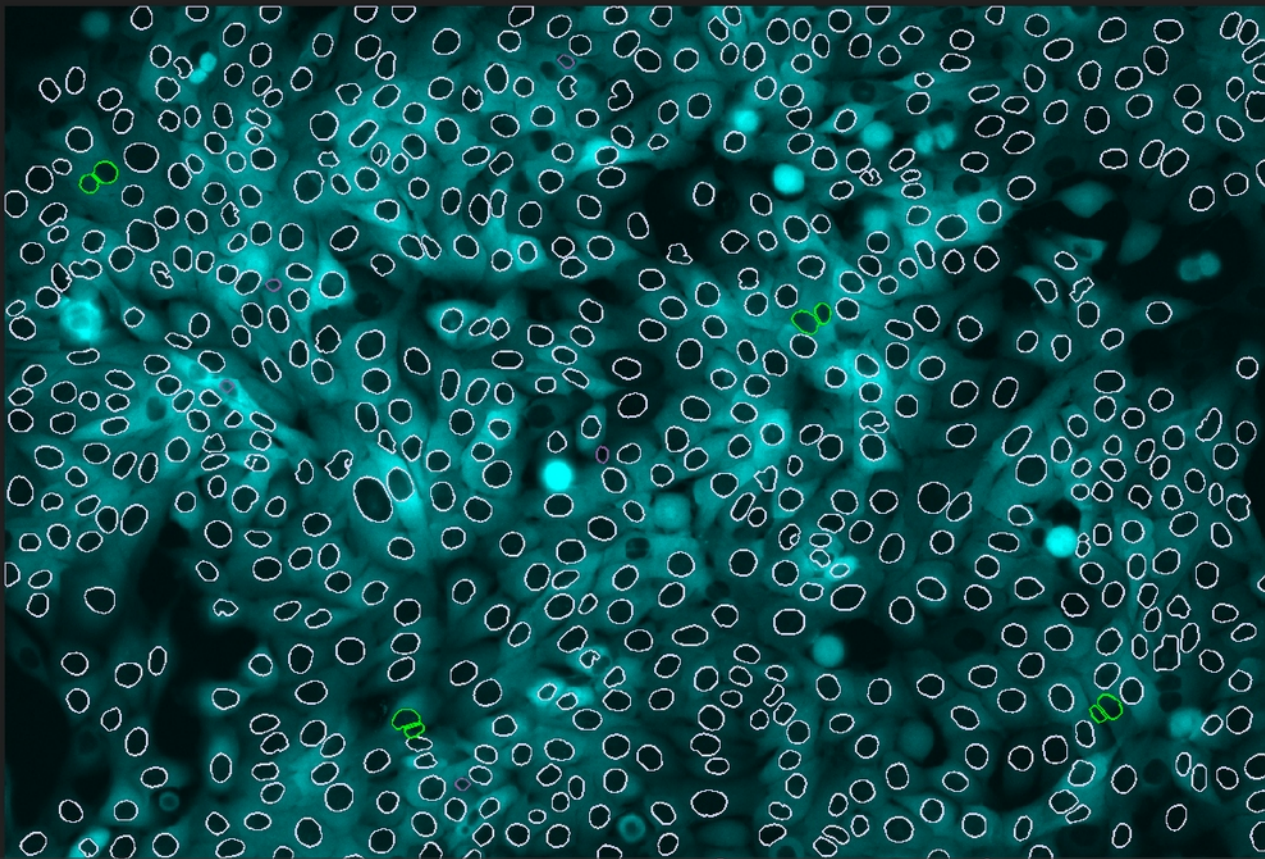
## ShuttleTracker - Logged messages

```
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 Sp0:1 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 Sp0:1 D:4 [26 ms]
```

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

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

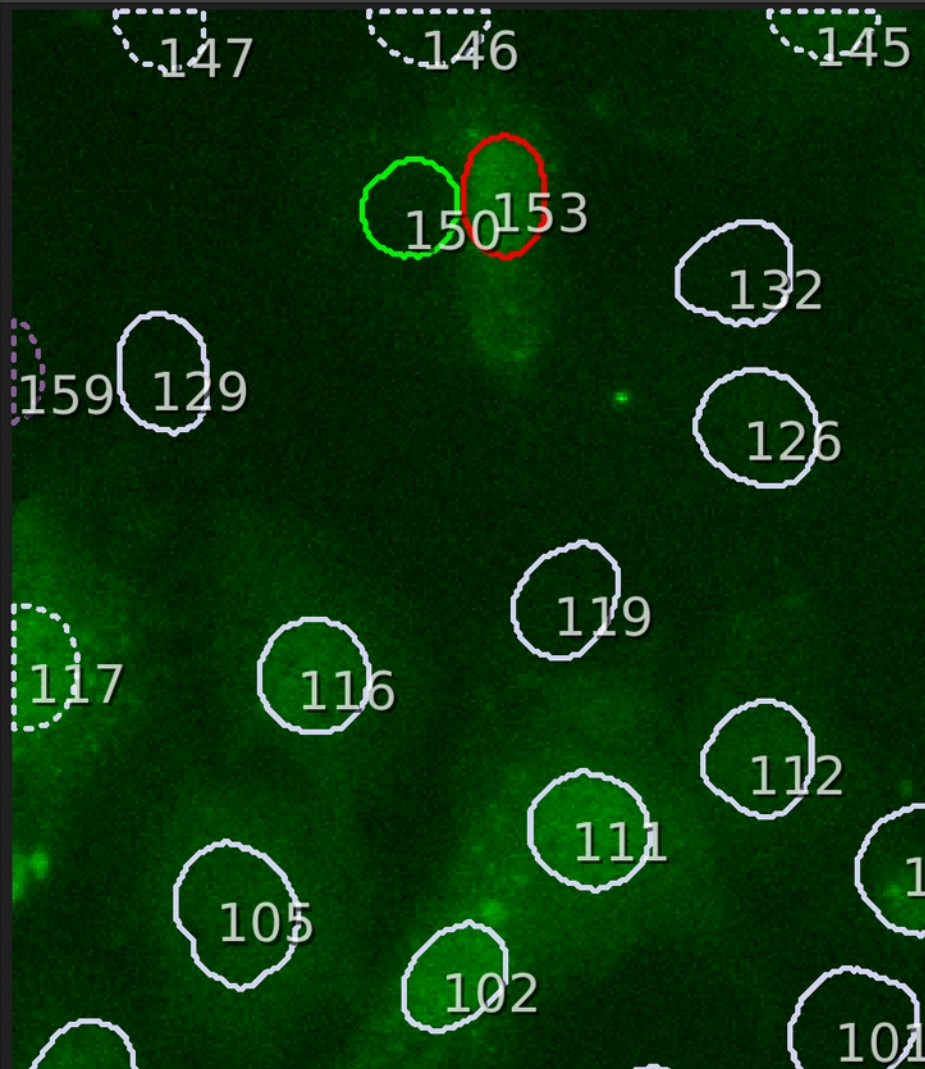
min. solidity 0.93  
min. area 0.40  
max. area 1.35  
✓ split adjacent  
min. indent 3 px

Detect nuclei

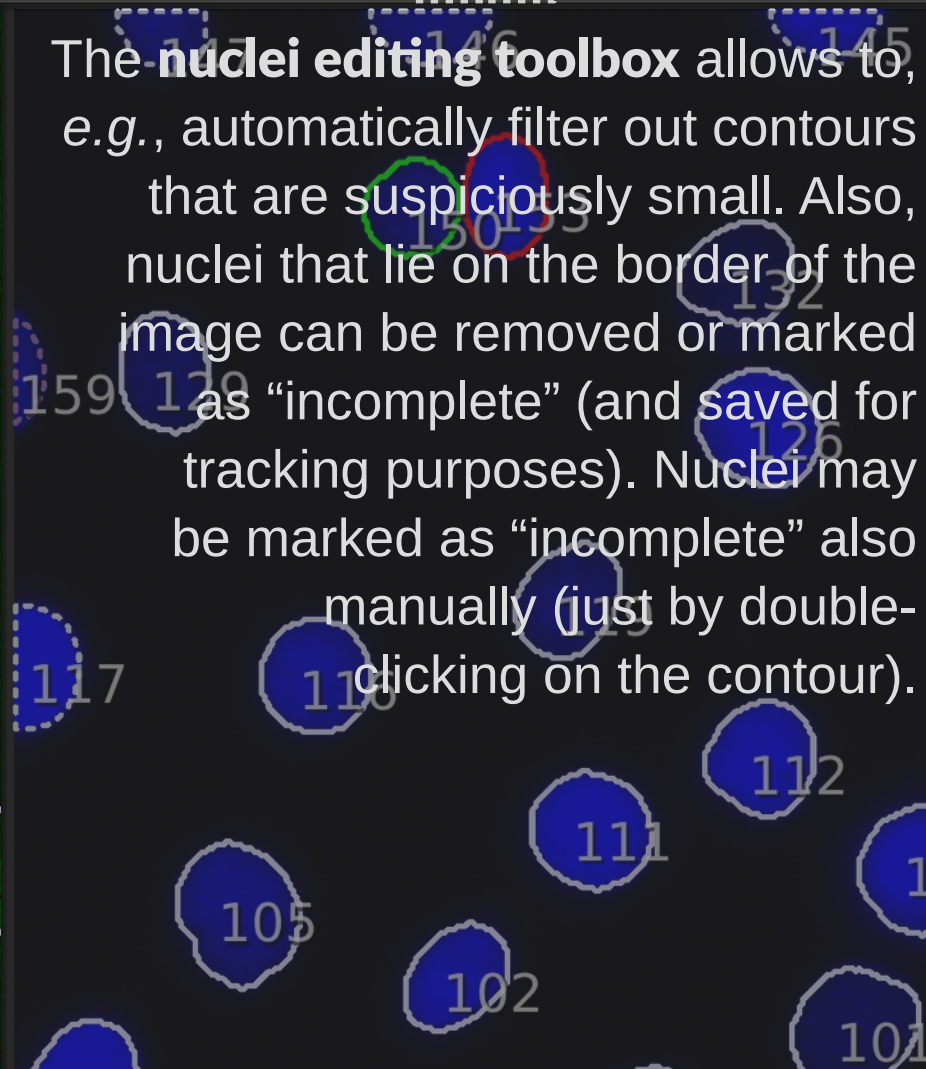
☐ auto-click☐ preview stages

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).





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 x

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 ☐

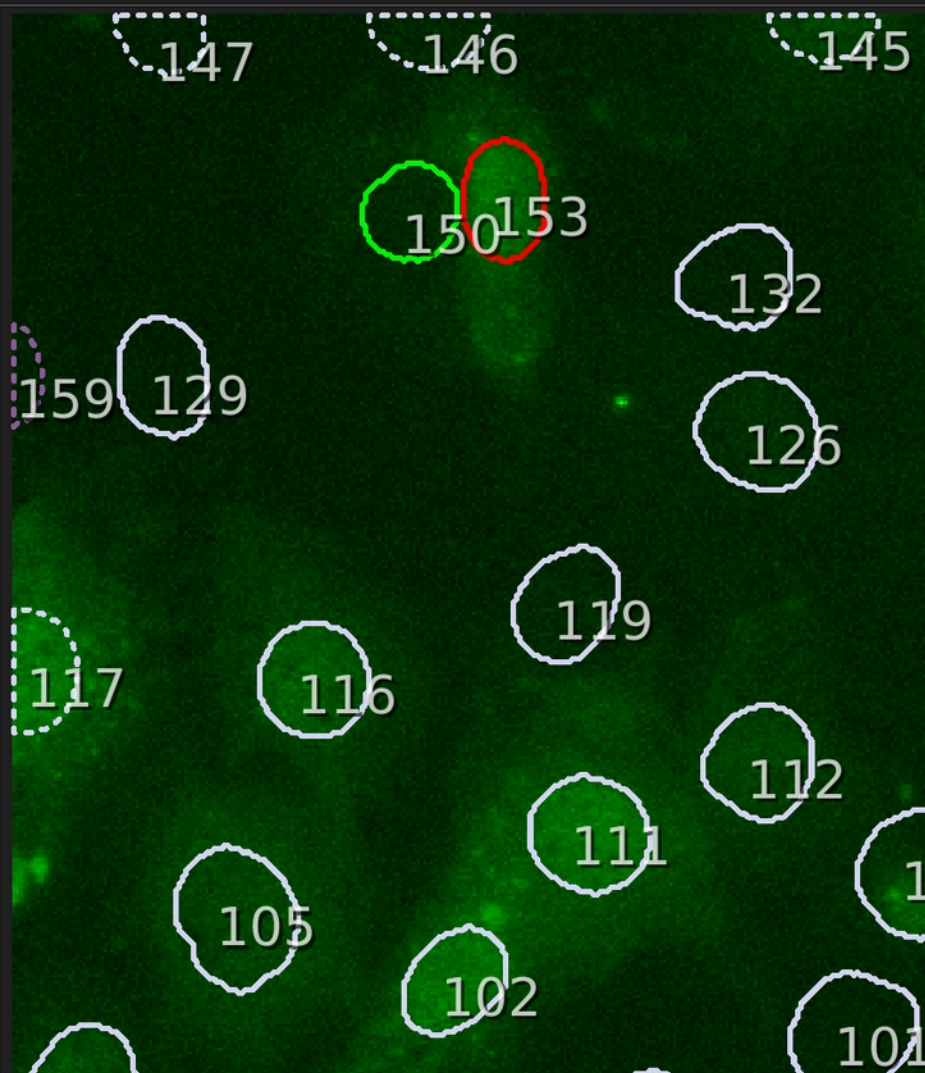
Annotations

☒ numbers

Contour properties

color: by origin

width: 1,6 px



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 x

View/Edit by origin

Nuclei:

- ✓ solo (detected)
- ✓ split (detected)
- ✓ manual

Non-nuclei (detected):

- ✓ non-splittable x
- ✓ split orphans x
- ✓ debris x

Edit by location

- border: any = x
- interior: debris x
- interior: orph's x

Annotations

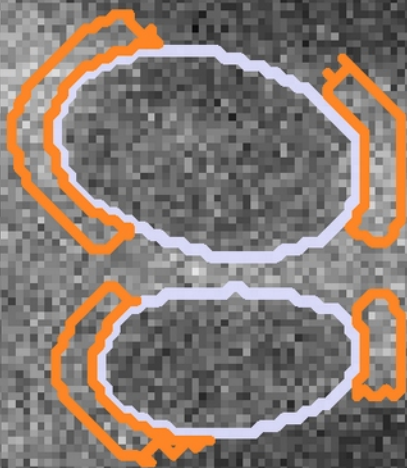
- ✓ numbers

Contour properties

- color by origin v
- width 1,6 px



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



## Perinuclei

all x

## 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  $\sigma$ 

smoothing 1 px

## Contour properties

width 1,4 px

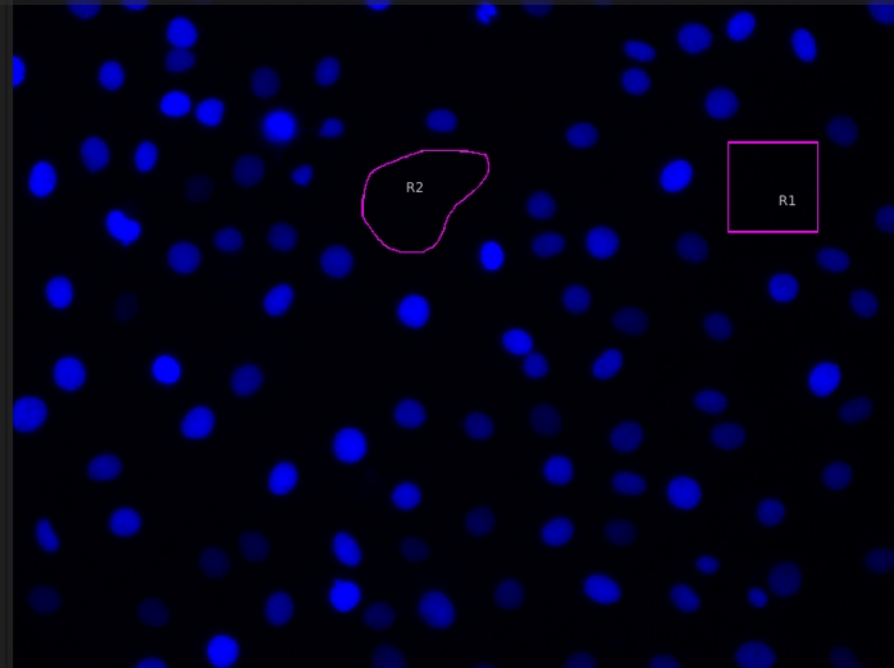
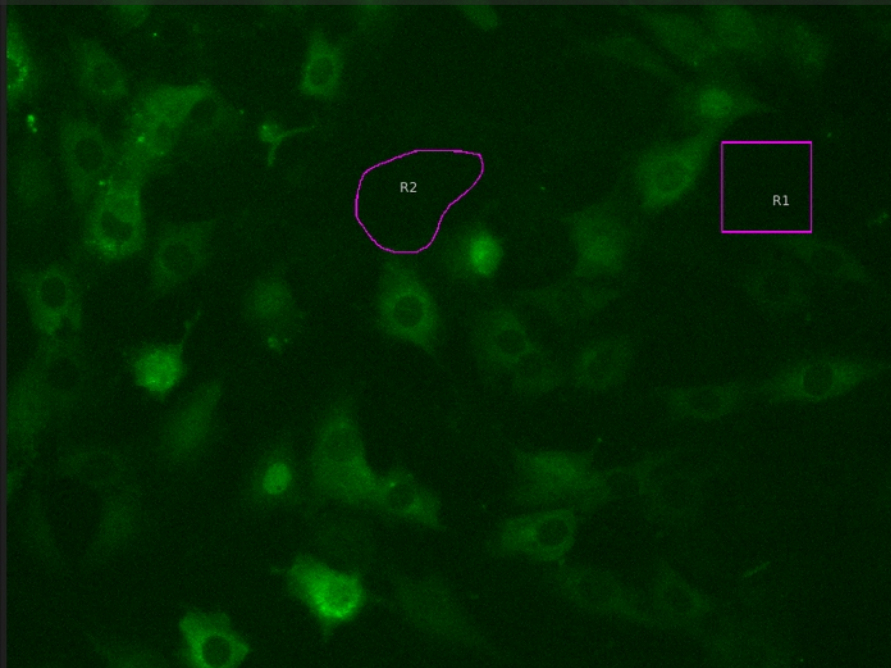
Derive

☒ 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.



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



## Regions editing

all x

## Background

subdivisions 10

subshifts 1

count 1

Sample

## Nuclei-centered Voronoi

scaling 1.00

erosion 2 px

neighborhoods → file

Tessellate

## Annotations

✓ numbers

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 x

Background

subdivisions 10

subshifts 1

count 1

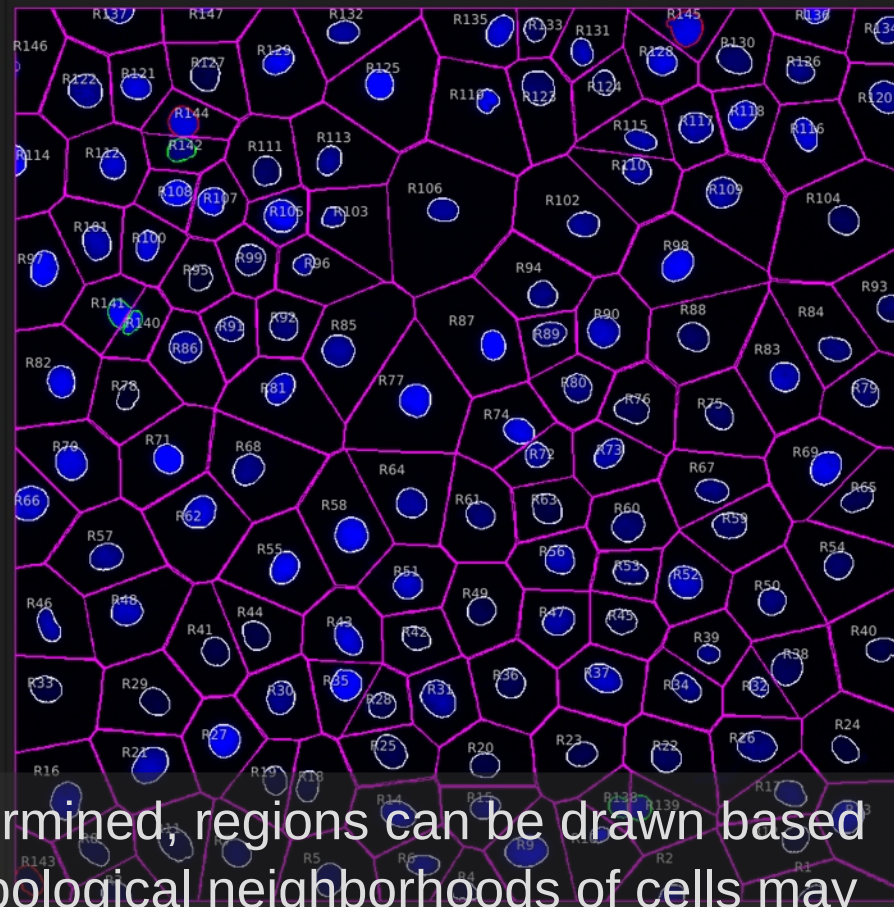
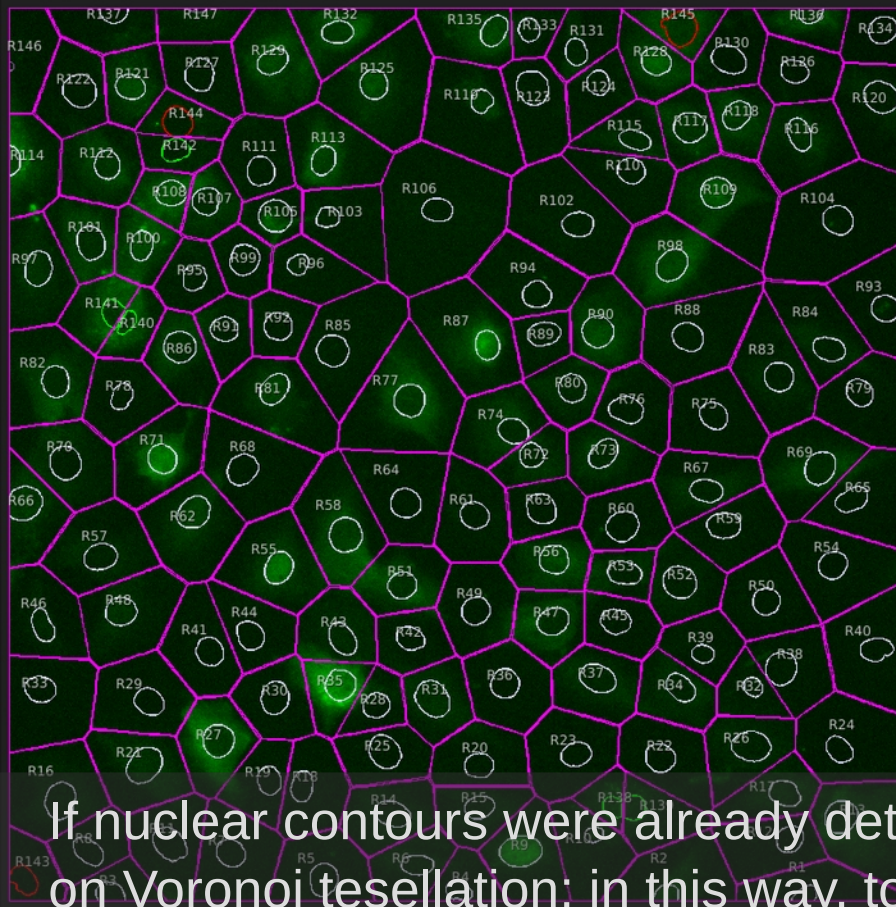
Sample

Nuclei-centered Voronoi

☐ scaling 1.00☐ erosion 2 px☒ neighborhoods → file

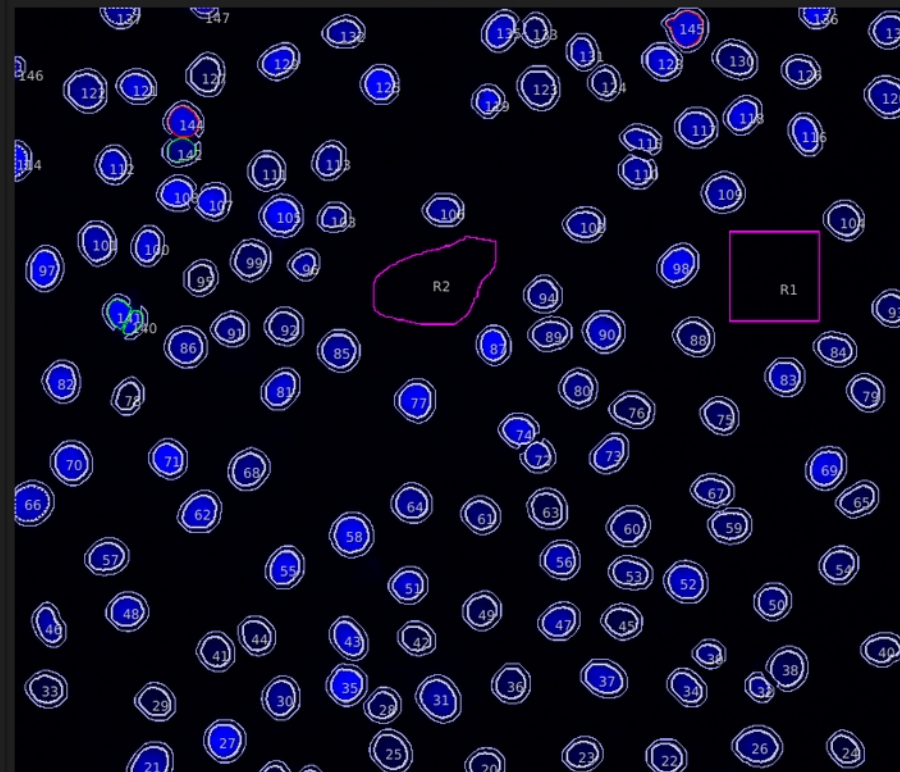
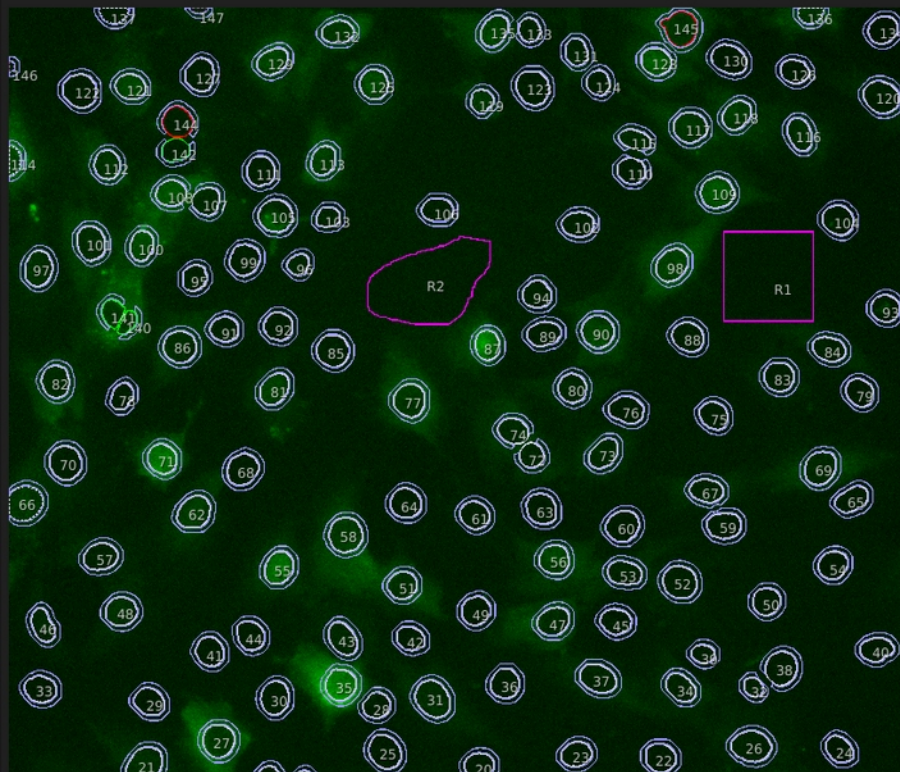
Tessellate

Annotations

☒ numbers

If nuclear contours were already determined, regions can be drawn based on Voronoi tessellation; in this way, topological neighborhoods of cells may be found (and saved to a file).





## Quantification

## Channel descriptions

☒ green NfκB☒ blue H2B

## Image

☒ non-masked only☒ csv

Export

## Nuclei

☐ png ☒ csv

Export

## Perinuclei

☒ csv

Export

## Regions

☐ png ☒ csv

Export

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...

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

QuantMinidemo - Mozilla Firefox

QuantMinidemo x +

localhost:8888/notebooks/QuantMinidemo 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()
```

	area	masked_area	eccentricity	is_complete	center_x	center_y	NFkB_intensity_min	NFkB_intensity_max	NFkB_intensity_mean	NFkB_intensity_stddev	NFkB_intensities_sum	H2B_intensity_min	H2B_intensity_max	H2B_intensity_mean	H2B_intensity_stddev	H2B_intensities_sum
nucleus_id																
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 x 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\_mean, NFkB\_intensity\_stddev, NFkB\_intensities\_sum, H2B\_intensity\_min, H2B\_intensity\_max, H2B\_intensity\_mean, H2B\_intensity\_stddev, H2B\_intensities\_sum'



## 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
---------------	-----

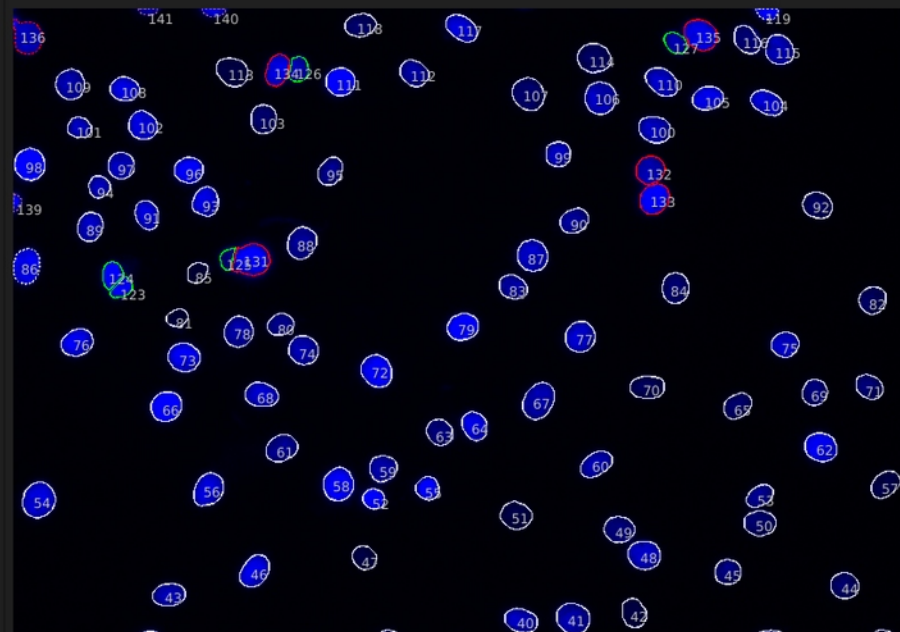
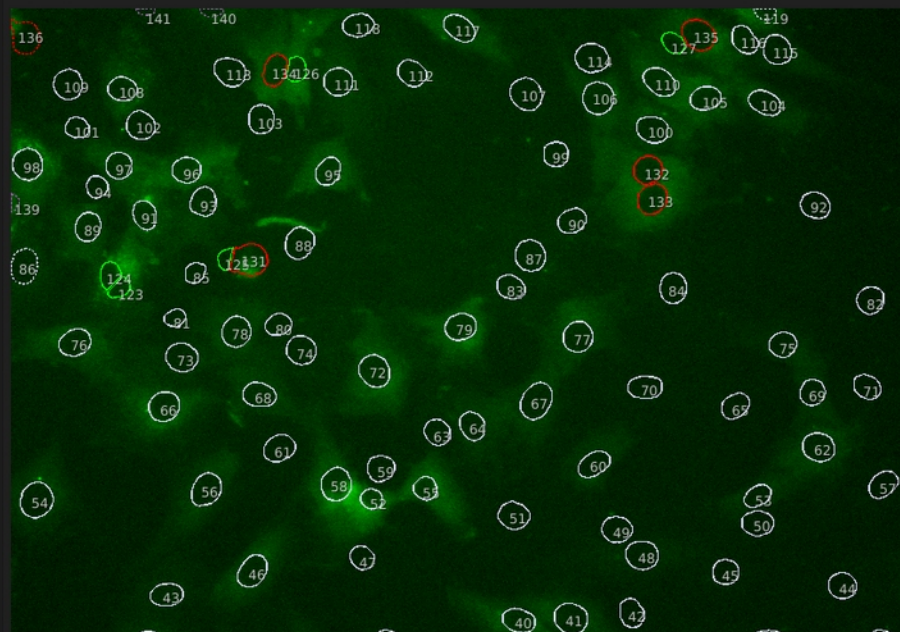
Nuclei  $\Delta xy$  cutoff

running median x10,0	
----------------------	--

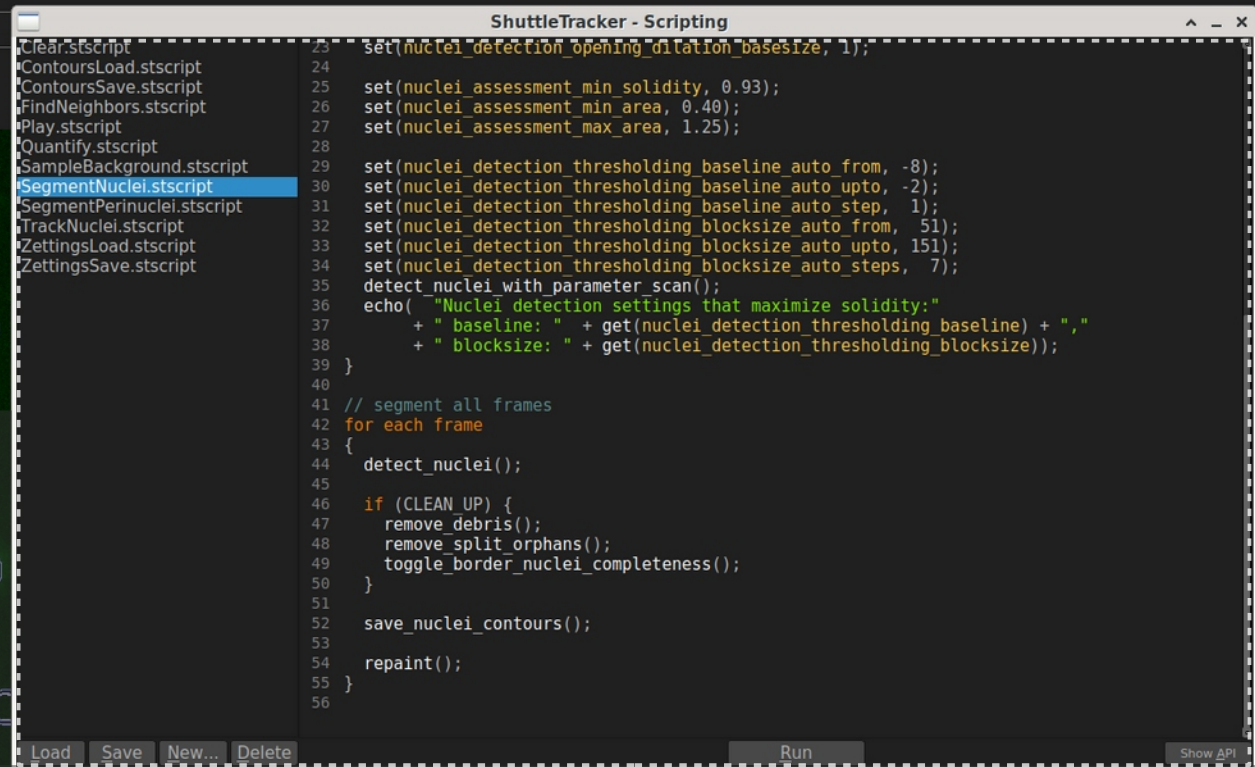
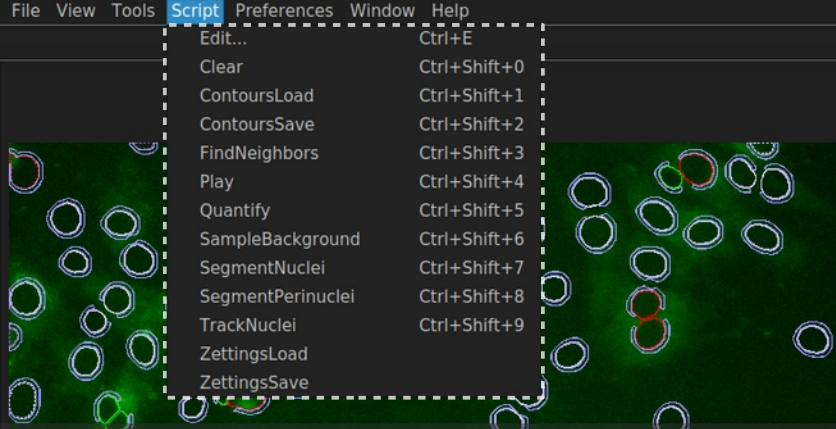
✓ Track breaking

min. area drop	50%
----------------	-----

Track (this frame)

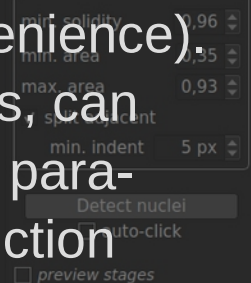
☐ auto-redraw traces

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 at the tracking toolbox after we become acquainted with the embedded script interpreter.

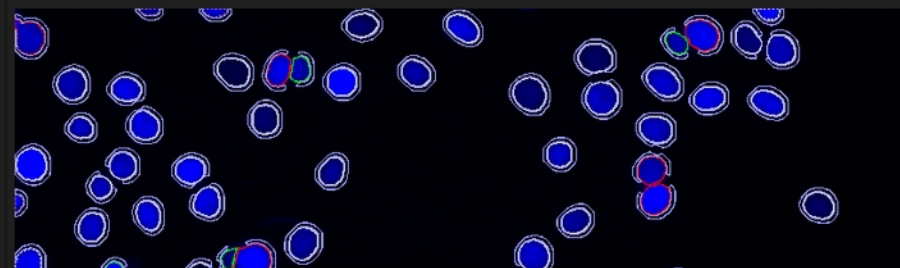
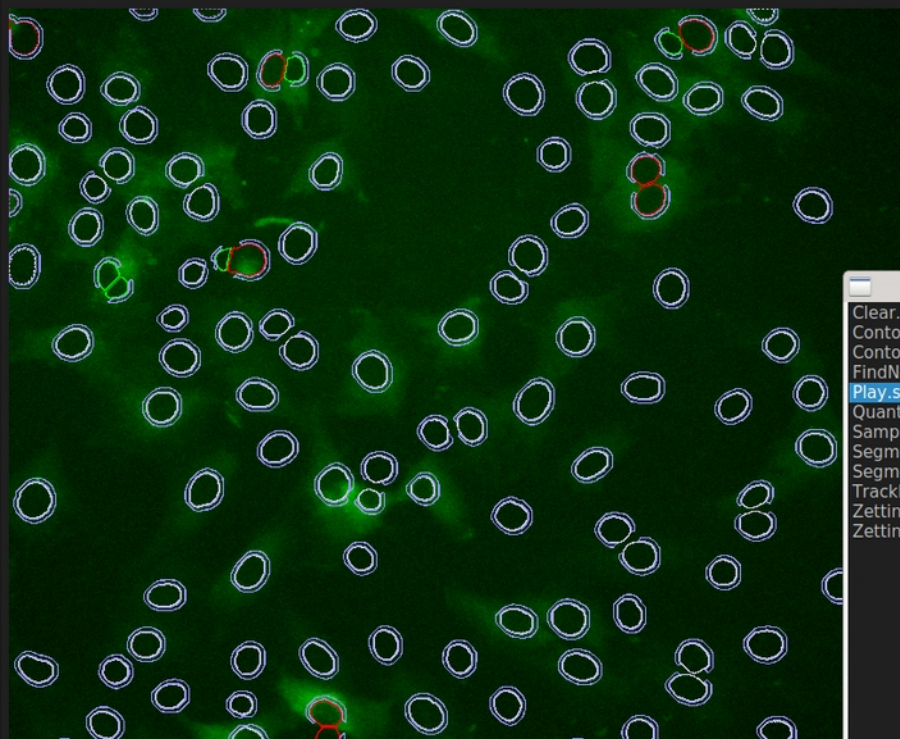


**Menu Script** allows one to call scripts to handle most typical repetitive tasks. The scripts can be modified and new ones can be created using a **Script Editor**.

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 programmatically 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).







## Nuclei detection

Channel

blue  
• nuclear • cytosolic

## Image preprocessing

☒ Normalization☐ Denoising 1.0☒ Smoothing 3 px

Blob detection

## ShuttleTracker - Scripting

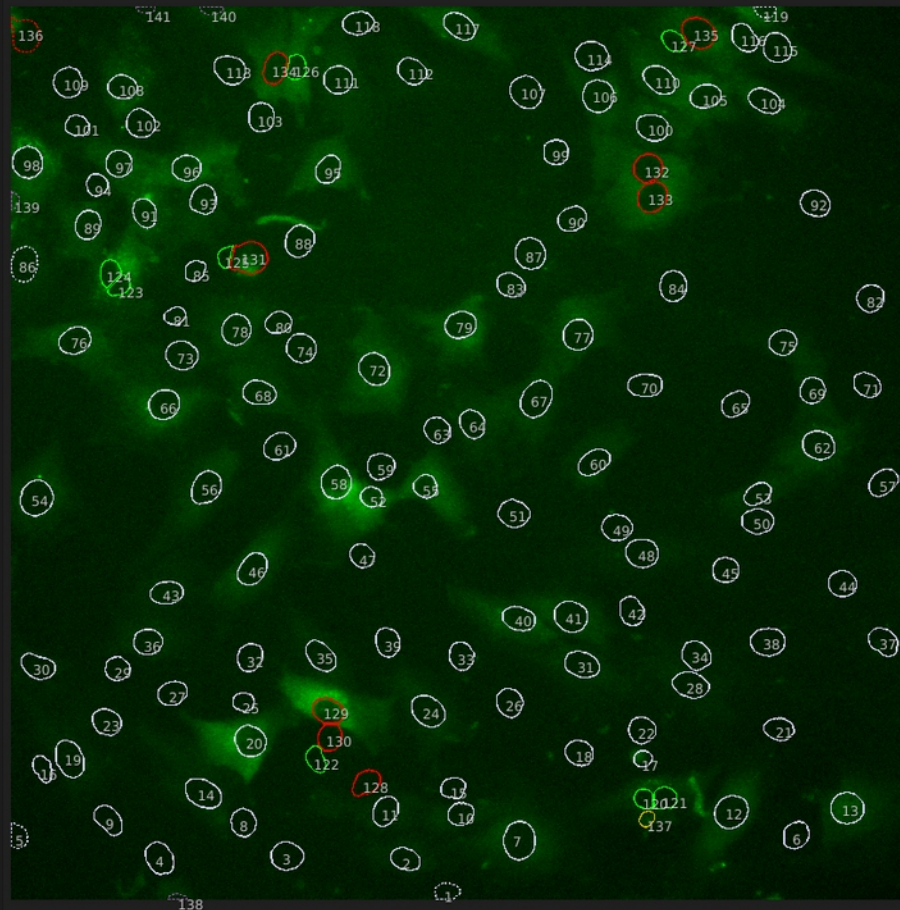
```
1 1 ///
2 2 /// Purpose: Play the whole image sequence.
3 3 ///
4 4
5 5 // The smaller the interval, the faster are images displayed.
6 6 var frames_per_second = 15; // intended speed
7 7 const ms_per_second = 1000; // 1 s = 1000 ms
8 8 const interval = ms_per_second/frames_per_second; // unit: millisecond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

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

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**

## ✓ 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  $\Delta xy$  cutoff

running median $\times 10,0$	
------------------------------	--

## ✓ Track breaking

min. area drop	50%
----------------	-----

Track (this frame)

☐ auto-redraw traces



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 editing toolbox** can be used to view and 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.

## Tracks editing

all ×

non-revised ×

## Trace properties

☐ Show only selected

Color by random

Selexn width 1,7 px

Others width 1,6 px

☒ Clip behind 30☒ Clip ahead 0☒ Show stubs

## Range Length OK

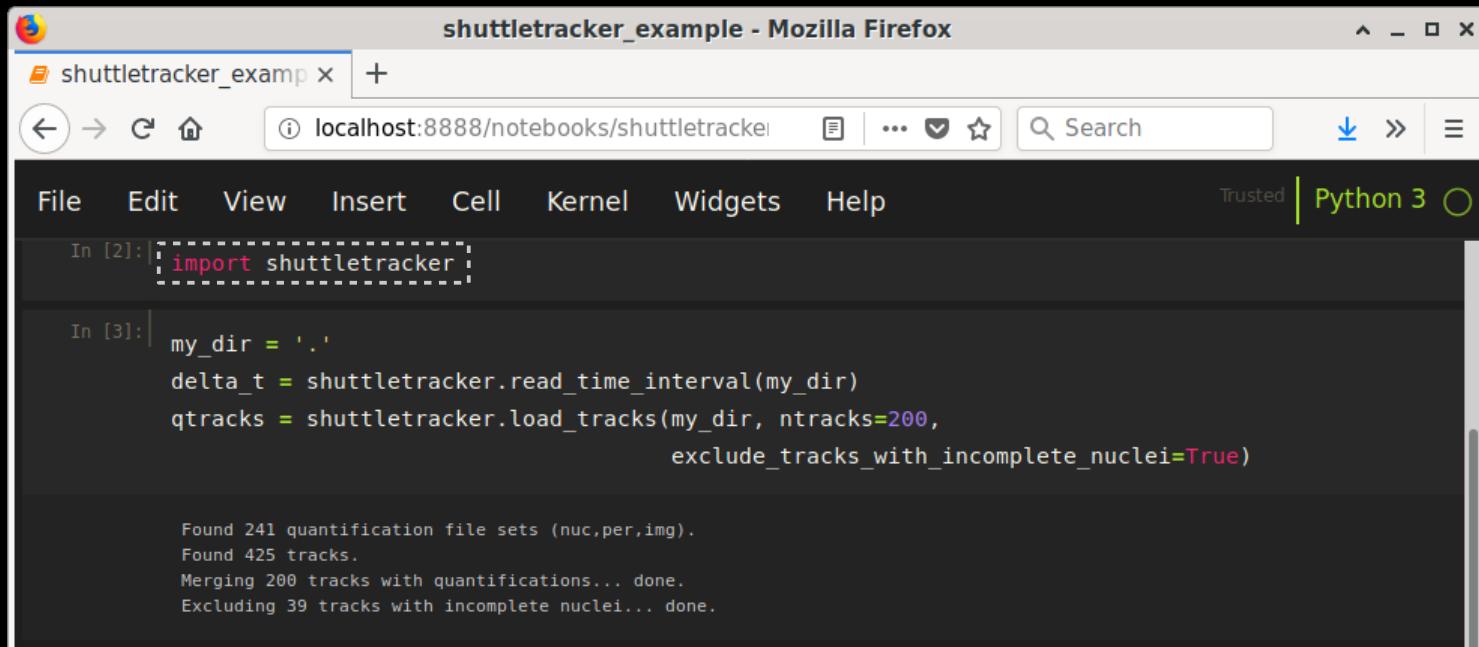
123	110:240	131
124	30:160	131
125	0:129	130
126	0:128	129
127	0:128	129
128	20:147	128
129	10:137	128
130	82:206	125
131	0:121	122
132	120:240	121
133	120:240	121
134	0:119	120
135	122:240	119
136	124:240	117
137	0:115	116
138	91:205	115
139	102:214	113
140	0:112	113
141	90:202	113
142	129:240	112
143	129:240	112

☐ Save only revised

Tracks ↔ file Save Load

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.



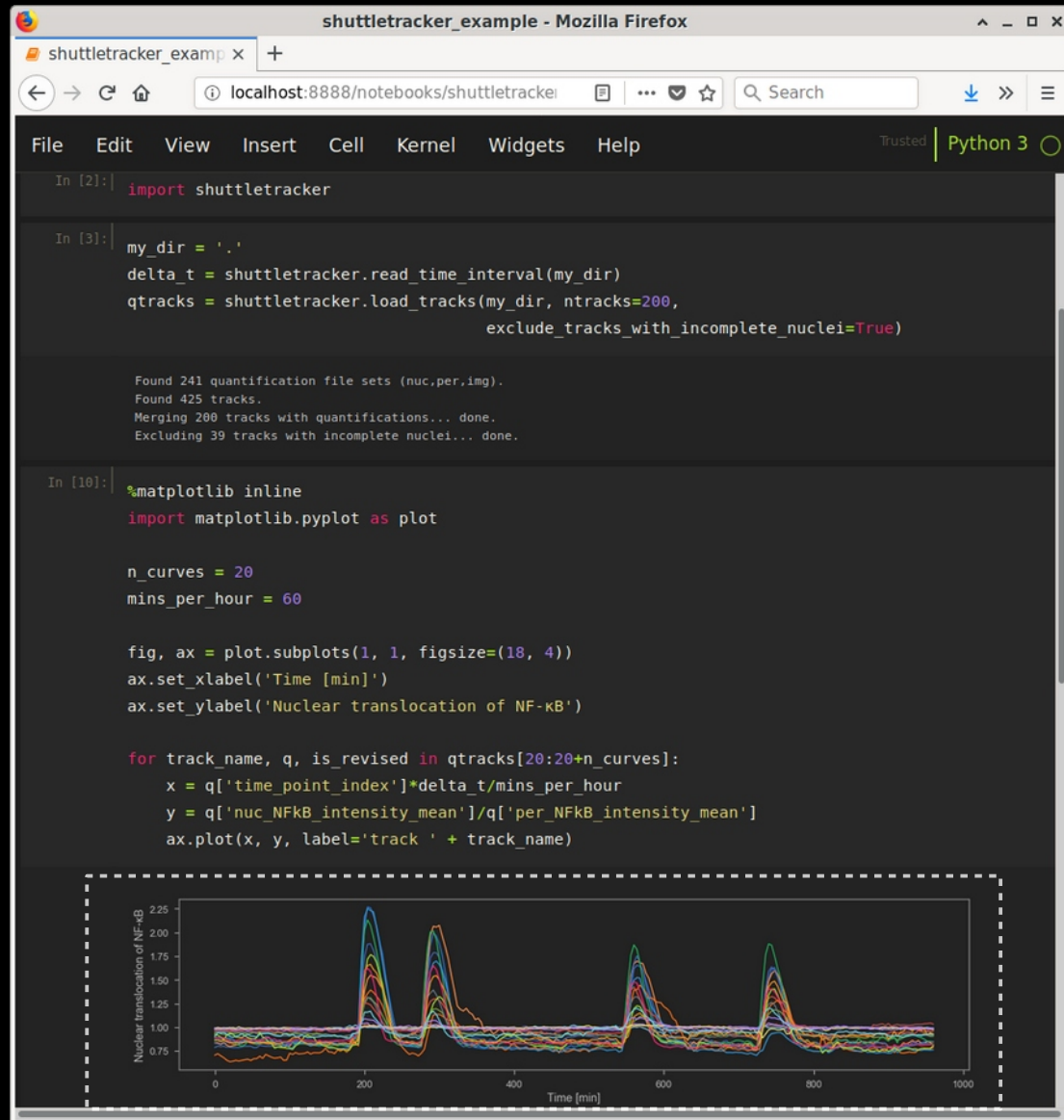
The screenshot shows a web browser window titled "shuttletracker\_example - Mozilla Firefox" displaying a Jupyter Notebook. The address bar shows "localhost:8888/notebooks/shuttletracker...". The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a status bar indicating "Trusted Python 3". The code is executed in two cells. The first cell contains the command to import the shuttletracker module. The second cell contains commands to set the working directory, read the time interval, load tracks (limiting to 200), and exclude tracks with incomplete nuclei. The output of the second cell shows the results of these operations.

```
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)
```

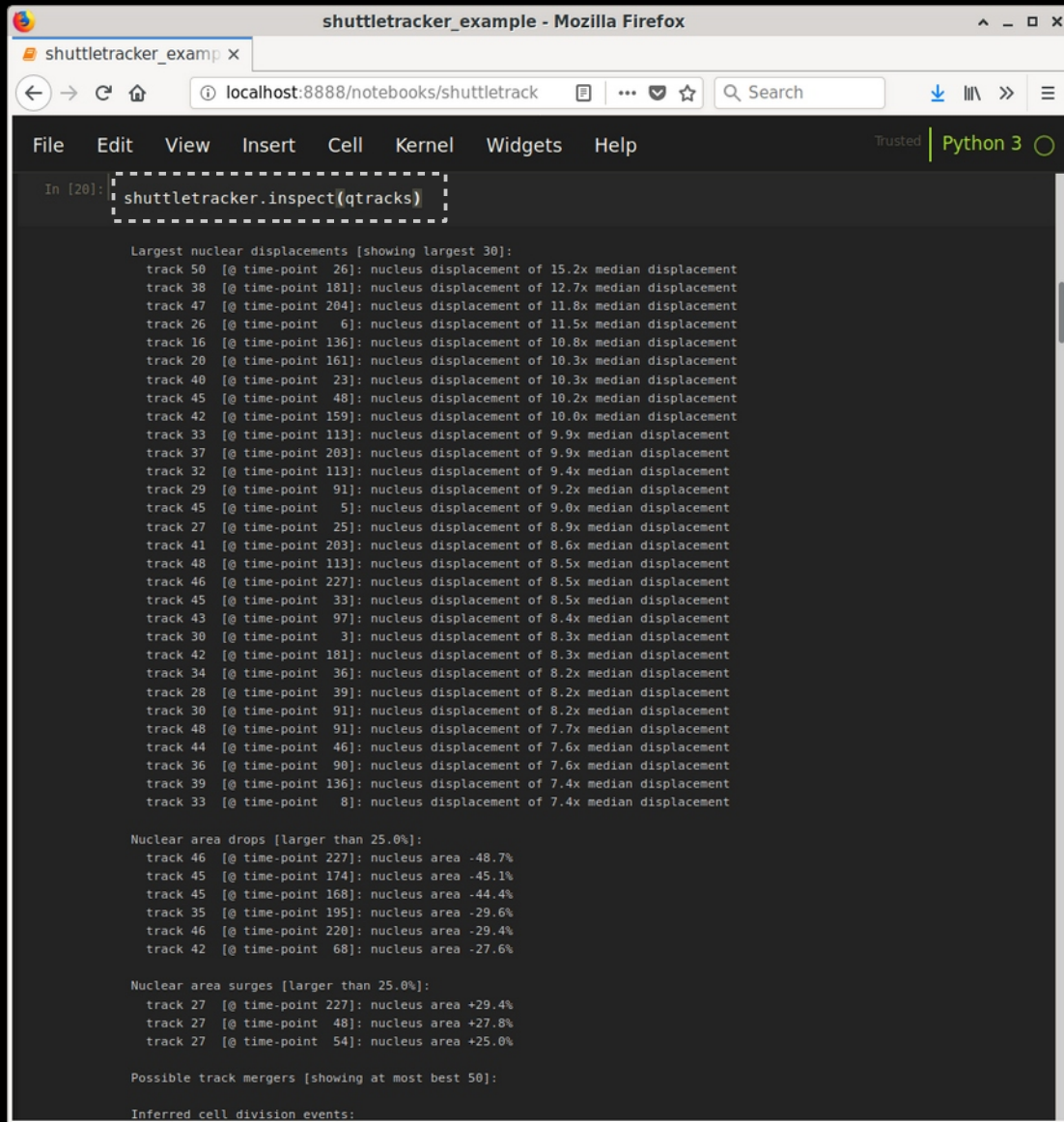
```
Found 241 quantification file sets (nuc,per,img).
Found 425 tracks.
Merging 200 tracks with quantifications... done.
Excluding 39 tracks with incomplete nuclei... done.
```

Four NF- $\kappa$ B nuclear translocation events can be distinguished in a **plot** of  $\langle \text{NF-}\kappa\text{B nuclear} \rangle / \langle \text{NF-}\kappa\text{B perinuclear} \rangle$ .



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.



```
In [20]: shuttletracker.inspect(qtracks)
```

```
Largest nuclear displacements [showing largest 30]:
track 50 [@ time-point 26]: nucleus displacement of 15.2x median displacement
track 38 [@ time-point 181]: nucleus displacement of 12.7x median displacement
track 47 [@ time-point 204]: nucleus displacement of 11.8x median displacement
track 26 [@ time-point 6]: nucleus displacement of 11.5x median displacement
track 16 [@ time-point 136]: nucleus displacement of 10.8x median displacement
track 20 [@ time-point 161]: nucleus displacement of 10.3x median displacement
track 40 [@ time-point 23]: nucleus displacement of 10.3x median displacement
track 45 [@ time-point 48]: nucleus displacement of 10.2x median displacement
track 42 [@ time-point 159]: nucleus displacement of 10.0x median displacement
track 33 [@ time-point 113]: nucleus displacement of 9.9x median displacement
track 37 [@ time-point 203]: nucleus displacement of 9.9x median displacement
track 32 [@ time-point 113]: nucleus displacement of 9.4x median displacement
track 29 [@ time-point 91]: nucleus displacement of 9.2x median displacement
track 45 [@ time-point 5]: nucleus displacement of 9.0x median displacement
track 27 [@ time-point 25]: nucleus displacement of 8.9x median displacement
track 41 [@ time-point 203]: nucleus displacement of 8.6x median displacement
track 48 [@ time-point 113]: nucleus displacement of 8.5x median displacement
track 46 [@ time-point 227]: nucleus displacement of 8.5x median displacement
track 45 [@ time-point 33]: nucleus displacement of 8.5x median displacement
track 43 [@ time-point 97]: nucleus displacement of 8.4x median displacement
track 30 [@ time-point 3]: nucleus displacement of 8.3x median displacement
track 42 [@ time-point 181]: nucleus displacement of 8.3x median displacement
track 34 [@ time-point 36]: nucleus displacement of 8.2x median displacement
track 28 [@ time-point 39]: nucleus displacement of 8.2x median displacement
track 30 [@ time-point 91]: nucleus displacement of 8.2x median displacement
track 48 [@ time-point 91]: nucleus displacement of 7.7x median displacement
track 44 [@ time-point 46]: nucleus displacement of 7.6x median displacement
track 36 [@ time-point 90]: nucleus displacement of 7.6x median displacement
track 39 [@ time-point 136]: nucleus displacement of 7.4x median displacement
track 33 [@ time-point 8]: nucleus displacement of 7.4x median displacement

Nuclear area drops [larger than 25.0%]:
track 46 [@ time-point 227]: nucleus area -48.7%
track 45 [@ time-point 174]: nucleus area -45.1%
track 45 [@ time-point 168]: nucleus area -44.4%
track 35 [@ time-point 195]: nucleus area -29.6%
track 46 [@ time-point 220]: nucleus area -29.4%
track 42 [@ time-point 68]: nucleus area -27.6%

Nuclear area surges [larger than 25.0%]:
track 27 [@ time-point 227]: nucleus area +29.4%
track 27 [@ time-point 48]: nucleus area +27.8%
track 27 [@ time-point 54]: nucleus area +25.0%

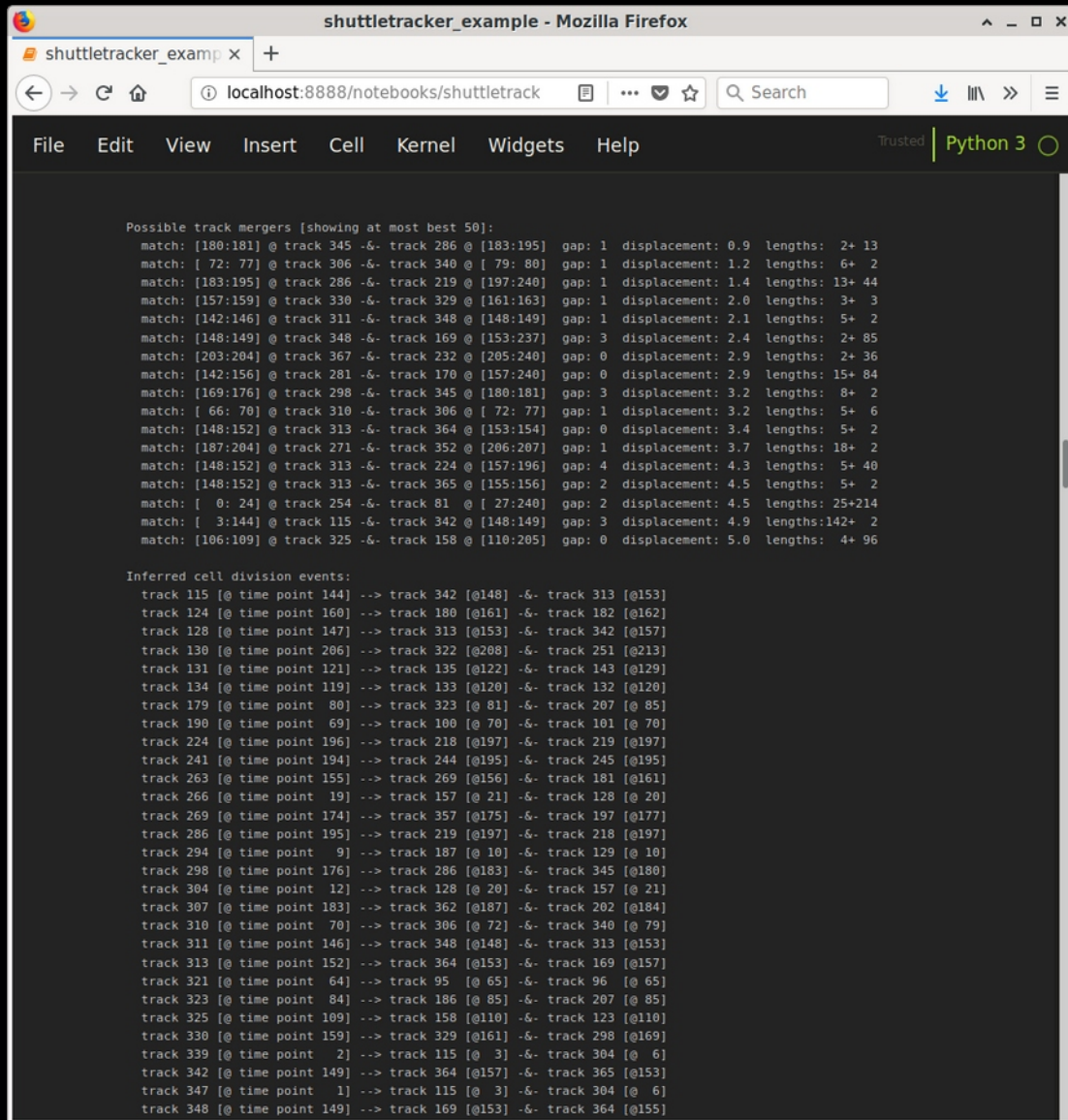
Possible track mergers [showing at most best 50]:

Inferred cell division events:
```



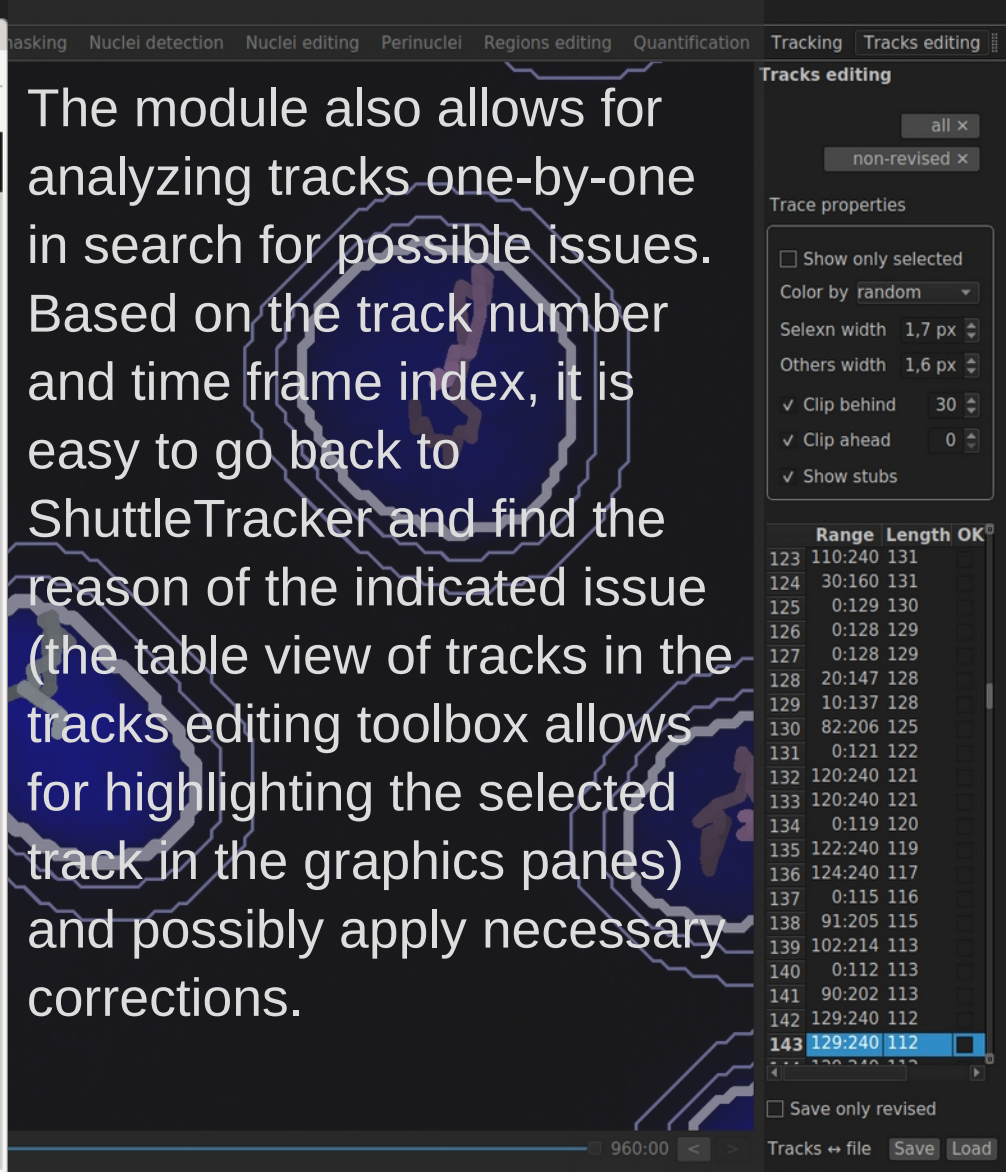
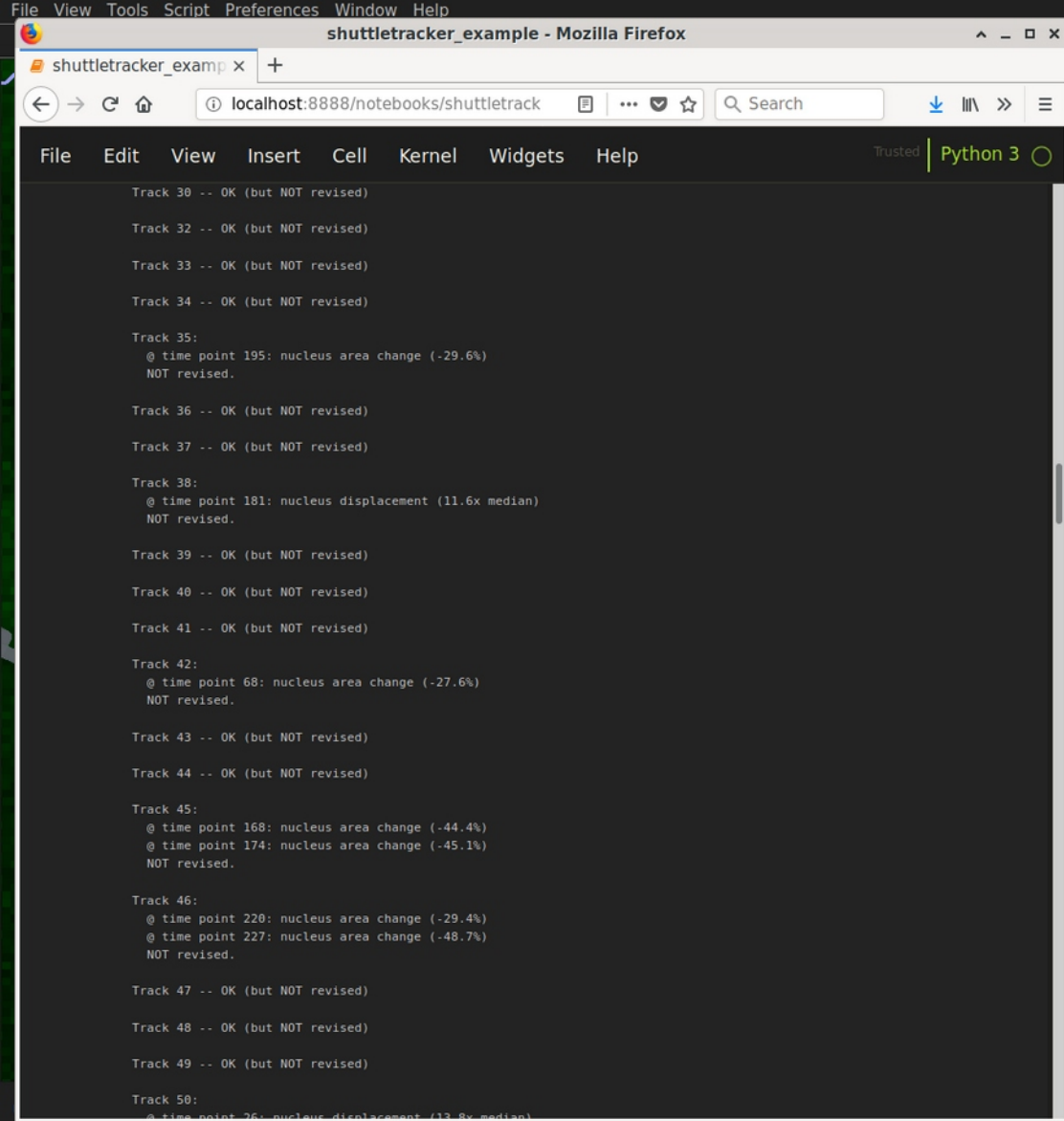
Gaps in tracks, that require manual intervention, are suggested.

Based on quantified features, cell lineage is deduced.



```
Possible track mergers [showing at most best 50]:
match: [180:181] @ track 345 -&- track 286 @ [183:195] gap: 1 displacement: 0.9 lengths: 2+ 13
match: [ 72: 77] @ track 306 -&- track 340 @ [ 79: 80] gap: 1 displacement: 1.2 lengths: 6+ 2
match: [183:195] @ track 286 -&- track 219 @ [197:240] gap: 1 displacement: 1.4 lengths: 13+ 44
match: [157:159] @ track 330 -&- track 329 @ [161:163] gap: 1 displacement: 2.0 lengths: 3+ 3
match: [142:146] @ track 311 -&- track 348 @ [148:149] gap: 1 displacement: 2.1 lengths: 5+ 2
match: [148:149] @ track 348 -&- track 169 @ [153:237] gap: 3 displacement: 2.4 lengths: 2+ 85
match: [203:204] @ track 367 -&- track 232 @ [205:240] gap: 0 displacement: 2.9 lengths: 2+ 36
match: [142:156] @ track 281 -&- track 170 @ [157:240] gap: 0 displacement: 2.9 lengths: 15+ 84
match: [169:176] @ track 298 -&- track 345 @ [180:181] gap: 3 displacement: 3.2 lengths: 8+ 2
match: [ 66: 70] @ track 310 -&- track 306 @ [ 72: 77] gap: 1 displacement: 3.2 lengths: 5+ 6
match: [148:152] @ track 313 -&- track 364 @ [153:154] gap: 0 displacement: 3.4 lengths: 5+ 2
match: [187:204] @ track 271 -&- track 352 @ [206:207] gap: 1 displacement: 3.7 lengths: 18+ 2
match: [148:152] @ track 313 -&- track 224 @ [157:196] gap: 4 displacement: 4.3 lengths: 5+ 40
match: [148:152] @ track 313 -&- track 365 @ [155:156] gap: 2 displacement: 4.5 lengths: 5+ 2
match: [ 0: 24] @ track 254 -&- track 81 @ [ 27:240] gap: 2 displacement: 4.5 lengths: 25+214
match: [ 3:144] @ track 115 -&- track 342 @ [148:149] gap: 3 displacement: 4.9 lengths:142+ 2
match: [106:109] @ track 325 -&- track 158 @ [110:205] gap: 0 displacement: 5.0 lengths: 4+ 96

Inferred cell division events:
track 115 @ time point 144 --> track 342 @148] -&- track 313 @153]
track 124 @ time point 160 --> track 180 @161] -&- track 182 @162]
track 128 @ time point 147 --> track 313 @153] -&- track 342 @157]
track 130 @ time point 206 --> track 322 @208] -&- track 251 @213]
track 131 @ time point 121 --> track 135 @122] -&- track 143 @129]
track 134 @ time point 119 --> track 133 @120] -&- track 132 @120]
track 179 @ time point 80 --> track 323 @ 81] -&- track 207 @ 85]
track 190 @ time point 69 --> track 100 @ 70] -&- track 101 @ 70]
track 224 @ time point 196 --> track 218 @197] -&- track 219 @197]
track 241 @ time point 194 --> track 244 @195] -&- track 245 @195]
track 263 @ time point 155 --> track 269 @156] -&- track 181 @161]
track 266 @ time point 19 --> track 157 @ 21] -&- track 128 @ 20]
track 269 @ time point 174 --> track 357 @175] -&- track 197 @177]
track 286 @ time point 195 --> track 219 @197] -&- track 218 @197]
track 294 @ time point 9 --> track 187 @ 10] -&- track 129 @ 10]
track 298 @ time point 176 --> track 286 @183] -&- track 345 @180]
track 304 @ time point 12 --> track 128 @ 20] -&- track 157 @ 21]
track 307 @ time point 183 --> track 362 @187] -&- track 202 @184]
track 310 @ time point 70 --> track 306 @ 72] -&- track 340 @ 79]
track 311 @ time point 146 --> track 348 @148] -&- track 313 @153]
track 313 @ time point 152 --> track 364 @153] -&- track 169 @157]
track 321 @ time point 64 --> track 95 @ 65] -&- track 96 @ 65]
track 323 @ time point 84 --> track 186 @ 85] -&- track 207 @ 85]
track 325 @ time point 109 --> track 158 @110] -&- track 123 @110]
track 330 @ time point 159 --> track 329 @161] -&- track 298 @169]
track 339 @ time point 2 --> track 115 @ 3] -&- track 304 @ 6]
track 342 @ time point 149 --> track 364 @157] -&- track 365 @153]
track 347 @ time point 1 --> track 115 @ 3] -&- track 304 @ 6]
track 348 @ time point 149 --> track 169 @153] -&- track 364 @155]
```



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](mailto:shuttletracker.software@gmail.com)

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