

Smart Microscopy Workshop

Day 1 – Image Analysis in ZEN



Marion Lang

Munich, 05/17/2021

Image Analysis Wizard incl. Zone Of Influence (ZOI)

BioApps

Introduction Macro Environment (OAD) in ZEN

Scripting: Image Analysis/BioApps

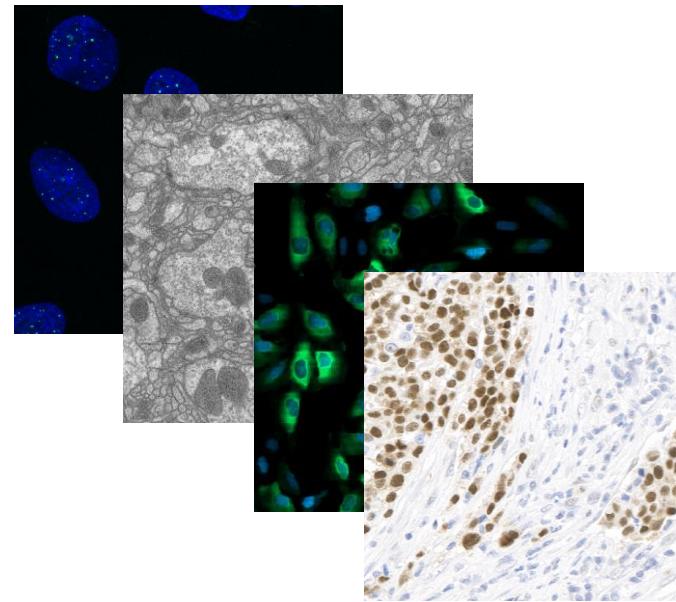
Microscopy

It's all about Numbers!



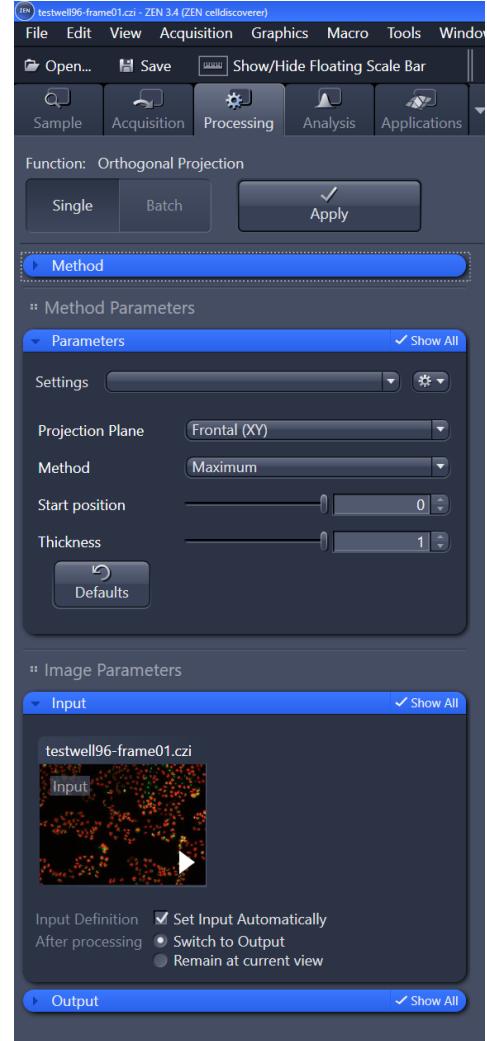
Customer goals:

- How many cells are in my sample?
- What area percentage is covered by cells?
- How many cells are positive/negative?
-

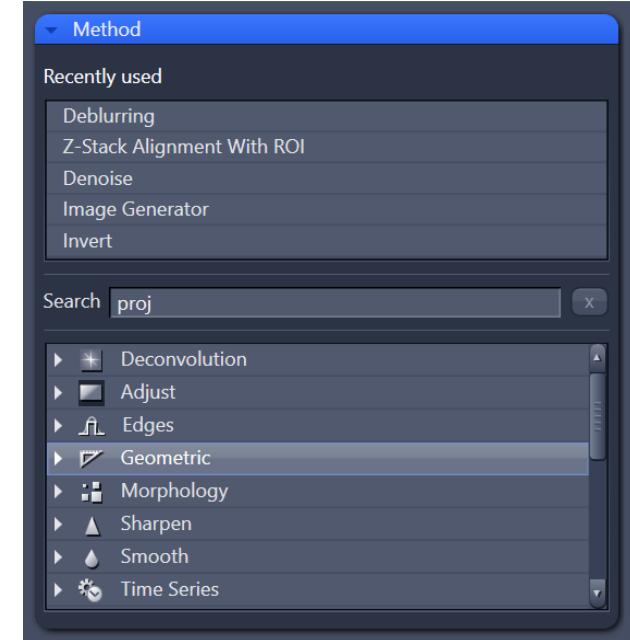


Processing in ZEN

Image Processing Tab



Select Image Processing Method

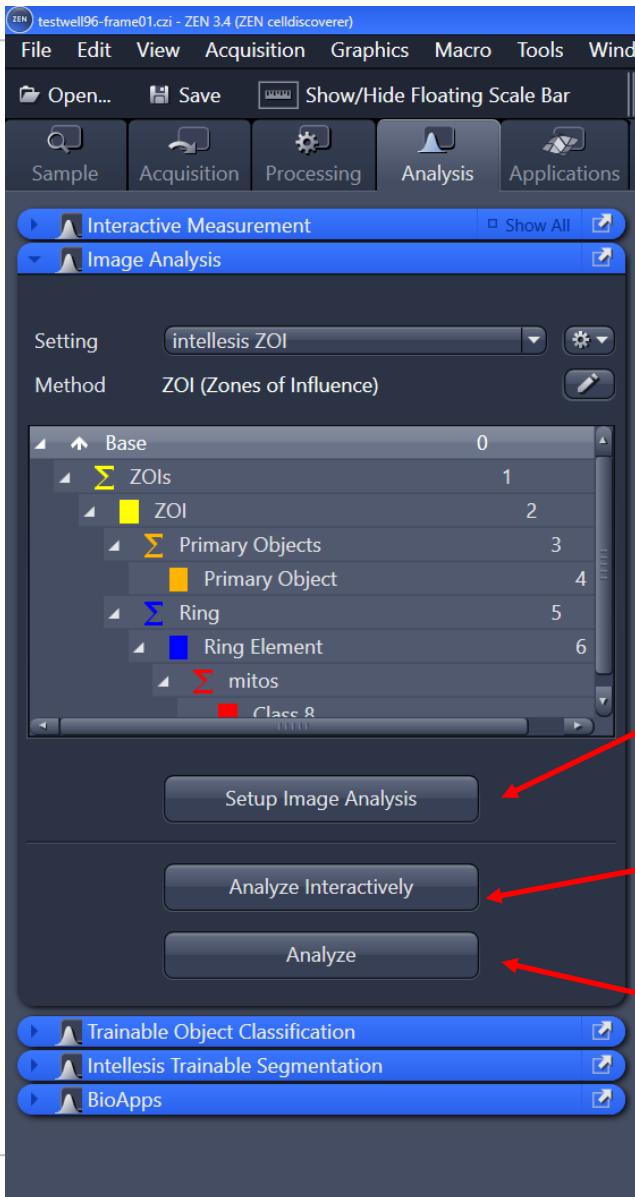


Set parameters for processing (depending on the input)

Define Input

Image Analysis with ZEN

Image Analysis Module



Wizard to guide you through the **setup** step-by-step

Analyze the images interactively

(if you have defined “Interactive” steps during setup)

When “Interactive” is active in a step, the analysis workflow will pause for you to interactively adapt parameters for the current image.

Analyze the complete .czi (without pause)

Image Analysis using ZEN

Image Analysis Wizard

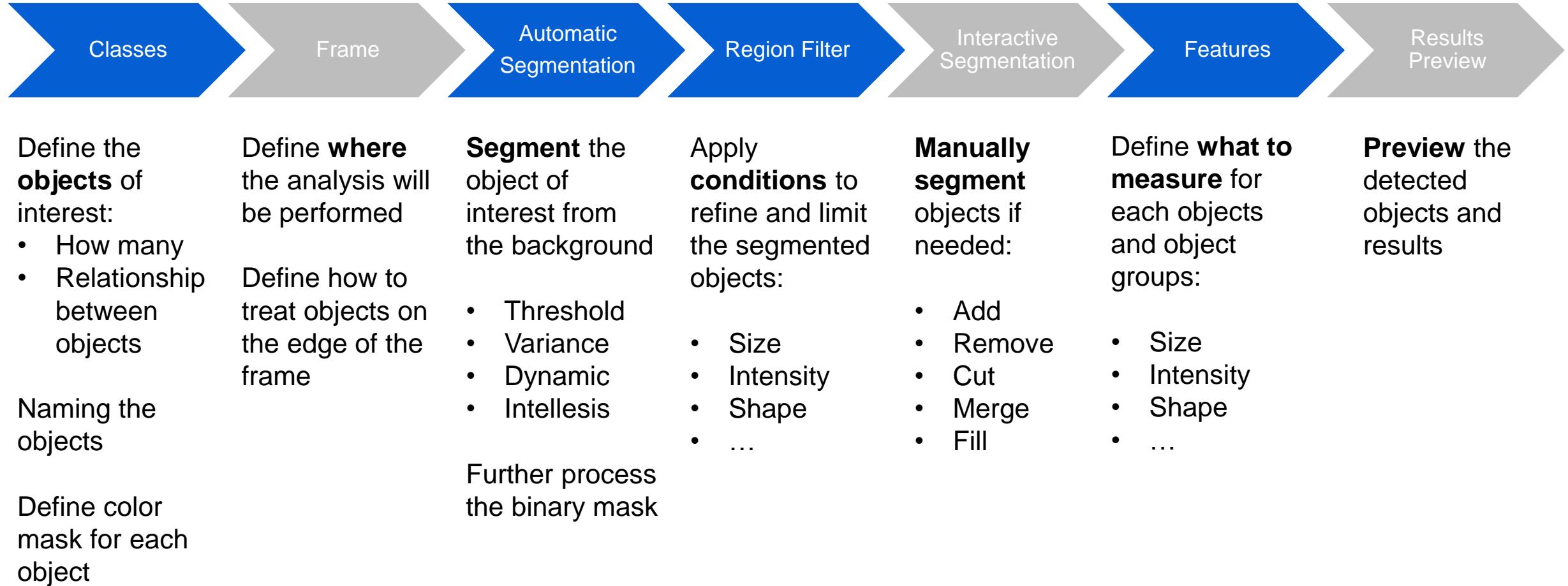


Image Analysis Step 1: Define Objects (Classes)

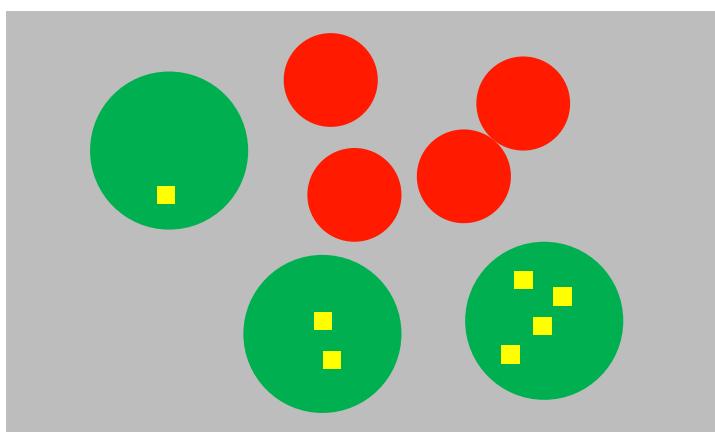
Overview



1/7 Classes

Class	Count
Base	0
Σ Classes 1	1
Class 2	DAPI
Σ Classes 3	3
Class 4	EGFP

Add Class Add Subclass Remove Class



Classes vs. Class (always created in pairs!)

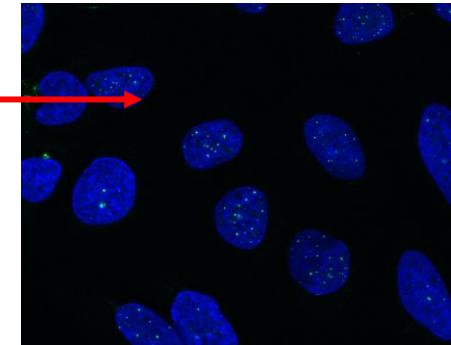
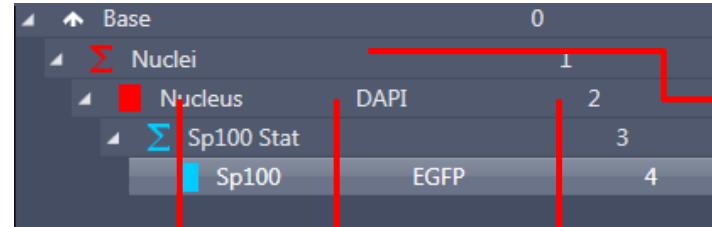
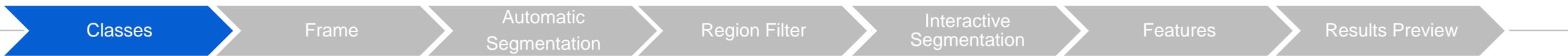
- “**Class**” contains **single objects**: e.g. a single nucleus.
You can measure the mean intensity, area, compactness of this single nucleus.
- “**Classes**” contains **a group of objects**: e.g. all the nuclei in a frame.
You can measure the number, total area, total intensity of nuclei inside a frame.

Class vs Subclass:

- Each “Class” represents a object of interest, and **each Class is independent**
E.g. white and red blood cells in a blood smear.
- **A “Subclass” is a object that is part of the primary “Class”:**
E.g. a FISH image where the nucleus is stained with DAPI, as the primary “Class”; while the HER2 dots is a “Subclass” for each nucleus.
You can specifically count how many HER2 dots “Subclass” are there in each nucleus “Class”.

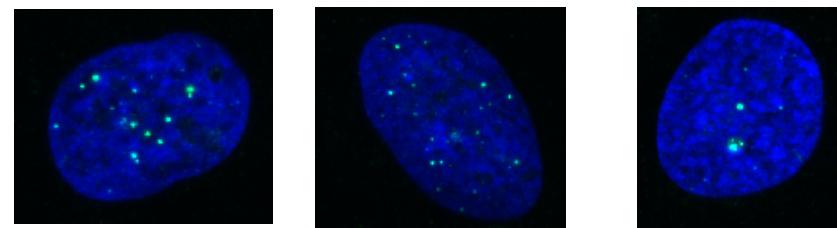
Image Analysis: Class/Classes

Classes vs. Class



Nuclei: Statistical Features of all single nuclei

Number of all objects
Mean Area of all objects
Mean Intensity of all objects
....



Nucleus: Features of each individual nucleus

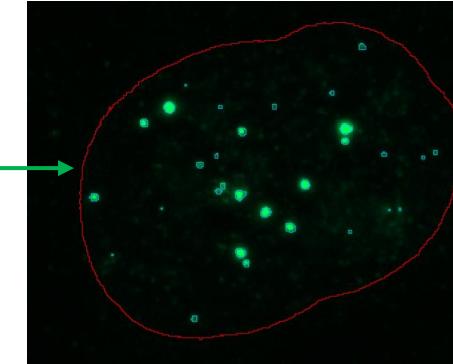
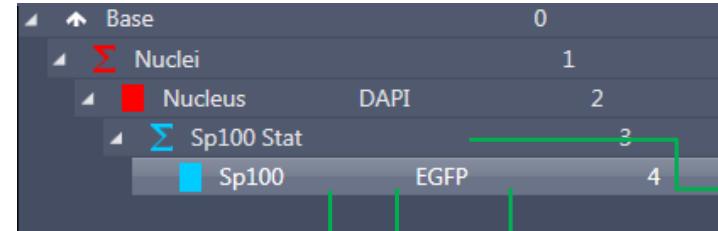
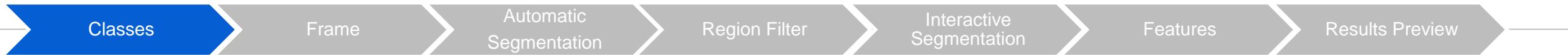
Area 1
Intensity 1
Circularity 1
...

Area 2
Intensity 2
Circularity 2
...

Area n
Intensity n
Circularity n
...

Image Analysis: Class/Classes

Sub-class



Sp100 Stat: Statistical features of all green spots in one cell

Number of Spots
Mean Area of Spots
Mean Intensity of Spots
...



Sp100: Features of each individual spot

Area 1
Intensity 1
...

Area 2
Intensity 2
...

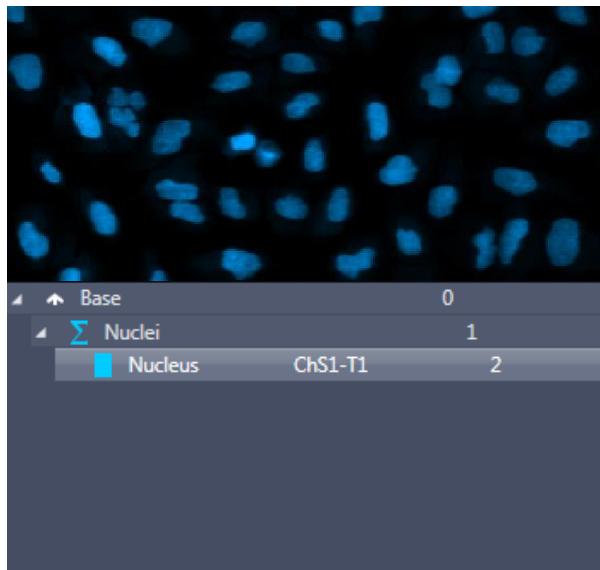
Area n
Intensity n
...

Image Analysis Step 1: Define Objects (Classes)

Classes examples



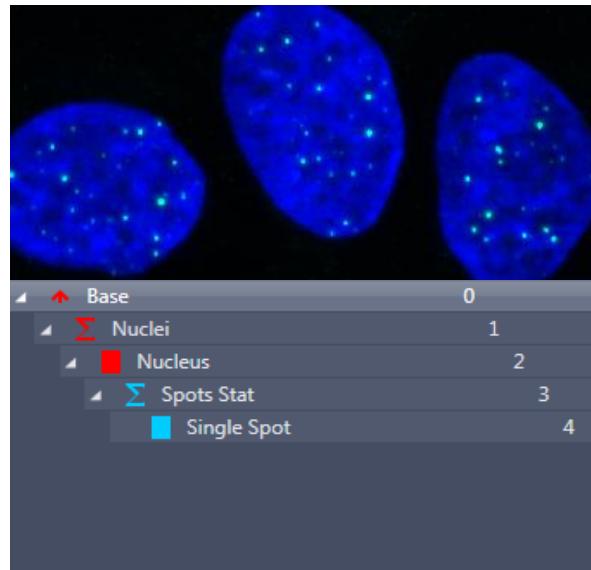
One class



Two independent classes



One class and one subclass

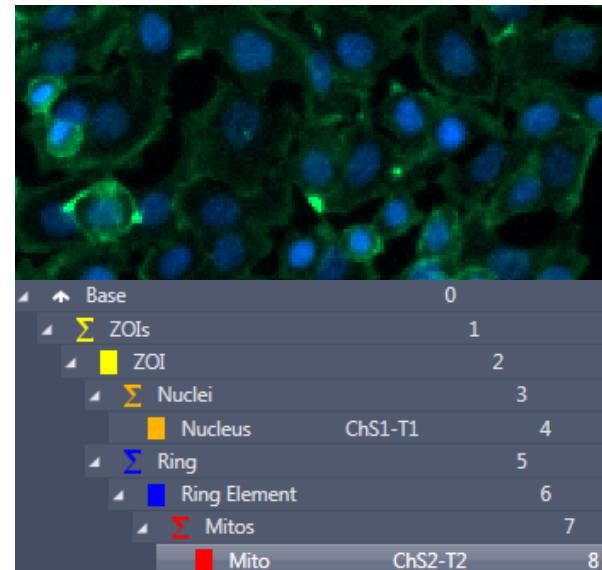


- Cell counting
- Wound healing

- Calculate white blood cell/red blood cell ratio
- Calculate ratio tissue vs. blood vessels

- Quantify HER2/PML bodies/SP100 dots per cell nucleus

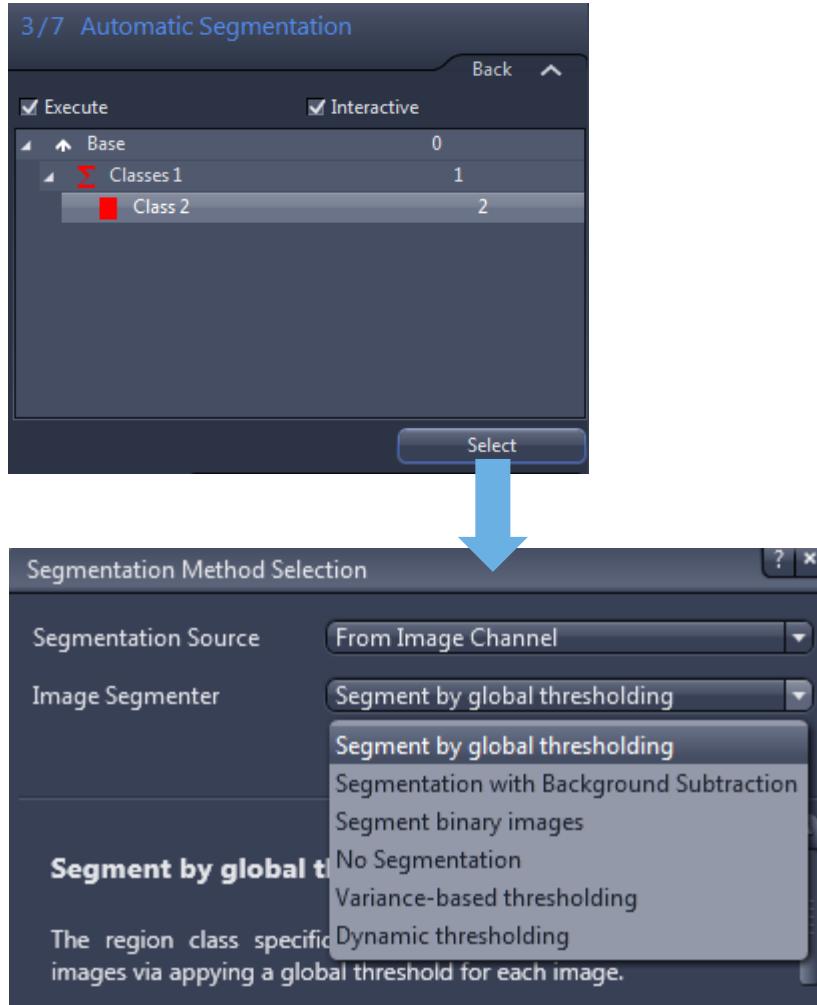
ZOI (Zone Of Influence)



- Measure mitochondria expression level per cell
- Calculate translocation ratio

Image Analysis Step 3: Segmentation

The most critical step



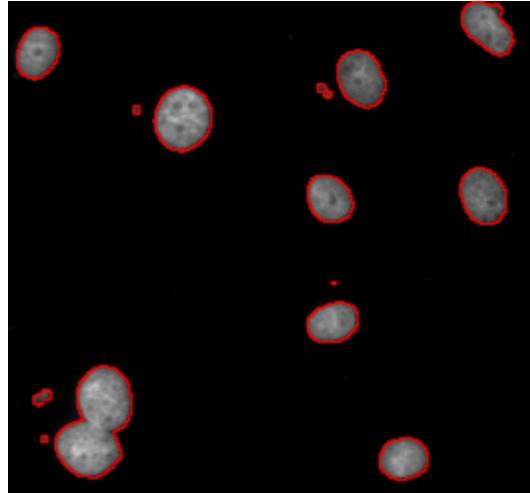
Segmentation: partitioning a digital image into multiple segments.
Binary process: the object of interest is 1, the rest is 0.
The result is a “mask” of the segmented objects.

Available segmentation methods:

- Segmentation by global thresholding
One global thresholding is performed for the whole image
- Segmentation with background subtraction
background subtraction is performed before thresholding
- Variance-based thresholding
edge detection: detects changes in pixel intensities, independent of the absolute intensity
- Segment binary images
simple binary segmentation
- Dynamic thresholding
adaptive thresholding to the surrounding of the object
- Intellesis
machine learning

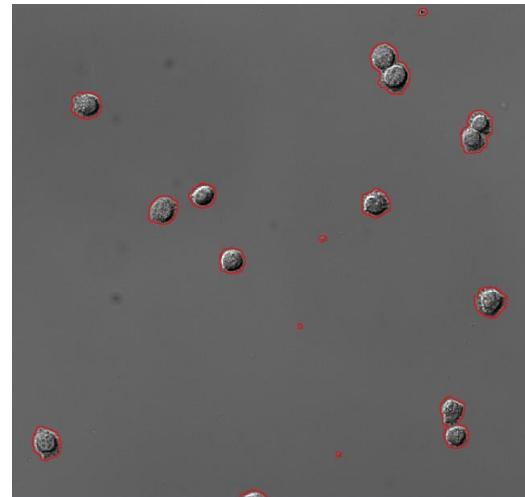
Image Analysis Step 3: Segmentation

Examples for different segmentation methods



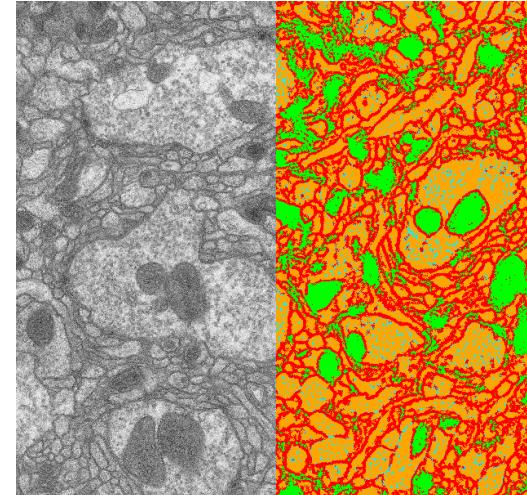
Threshold
(intensity based)

- Fluorescence images



Variance
(change in intensity)

- Brightfield images

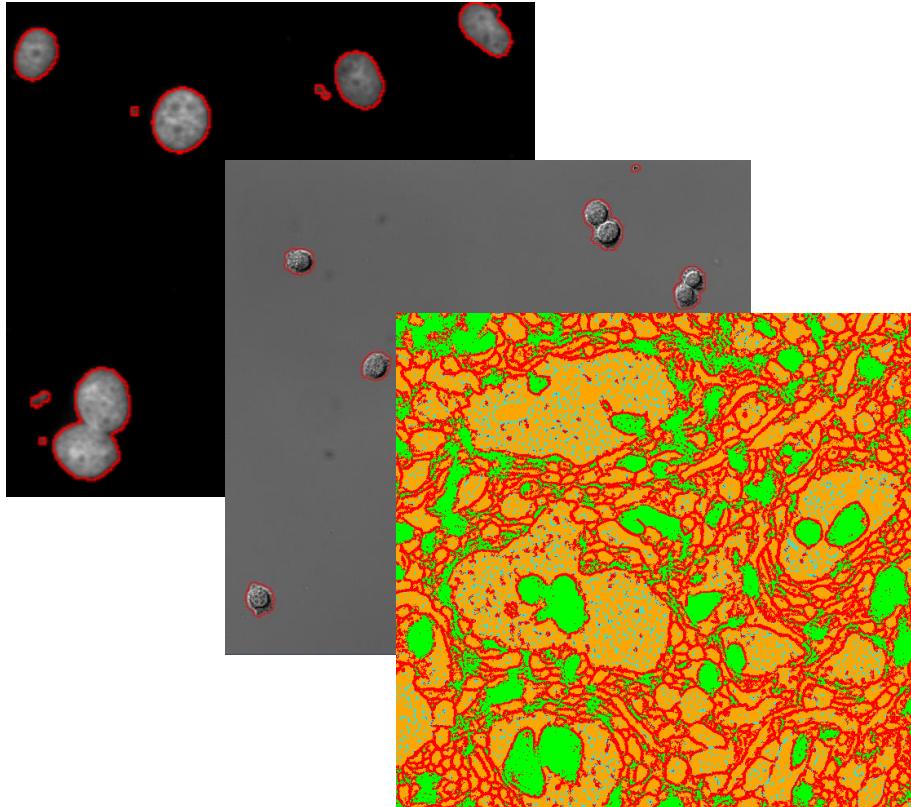


Intellessis
(machine learning)

- Everything (but slow)
- Nothing else works
- Ease of use

Image Analysis Step 3: Segmentation

The most critical step



Further refine the segmentation mask

- **Min. Area:** Remove small objects
- **Binary Options**
 - Dilate
 - Erode
 - Open
 - Close
- **Separate Objects**
 - Watershed
 - Morphology

Image Analysis Step 3: Segmentation

Separate objects



Separate objects that are connected
Acts only on the binary image.

- **Morphology**

Erode then dilate, objects that are separated won't be merged again

- **Watersheds**

Separate objects assuming they are roughly the same size, might wrongly split elongated object

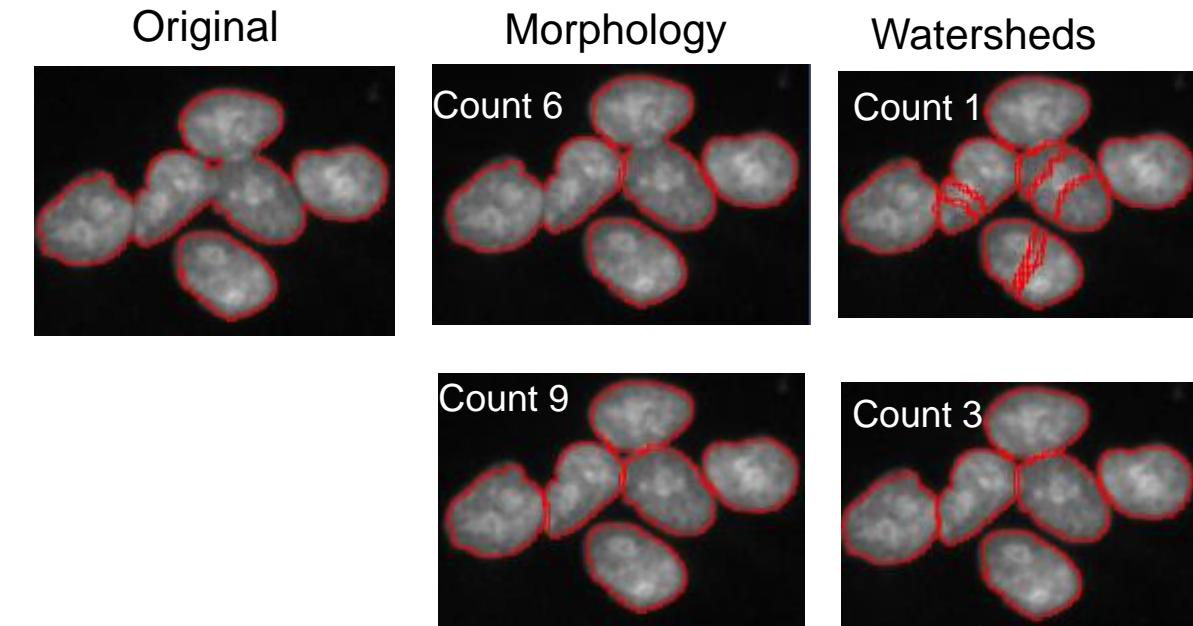


Image Analysis Step 3: Segmentation

Binary options



The “segmented mask” can be refined by various processes. This affects only the mask itself, but not the original image.

Binary: Performs morphological operations on the mask

- **Dilate:** *enlarge the boundary of the mask by counts/pixels*
- **Erode:** *shrink the boundary of the mask by counts/pixels*
- **Open:** *Erode then dilate: smoothes and removes isolated pixels.*
- **Close:** *Dilate then erode, smoothes and fills small holes.*

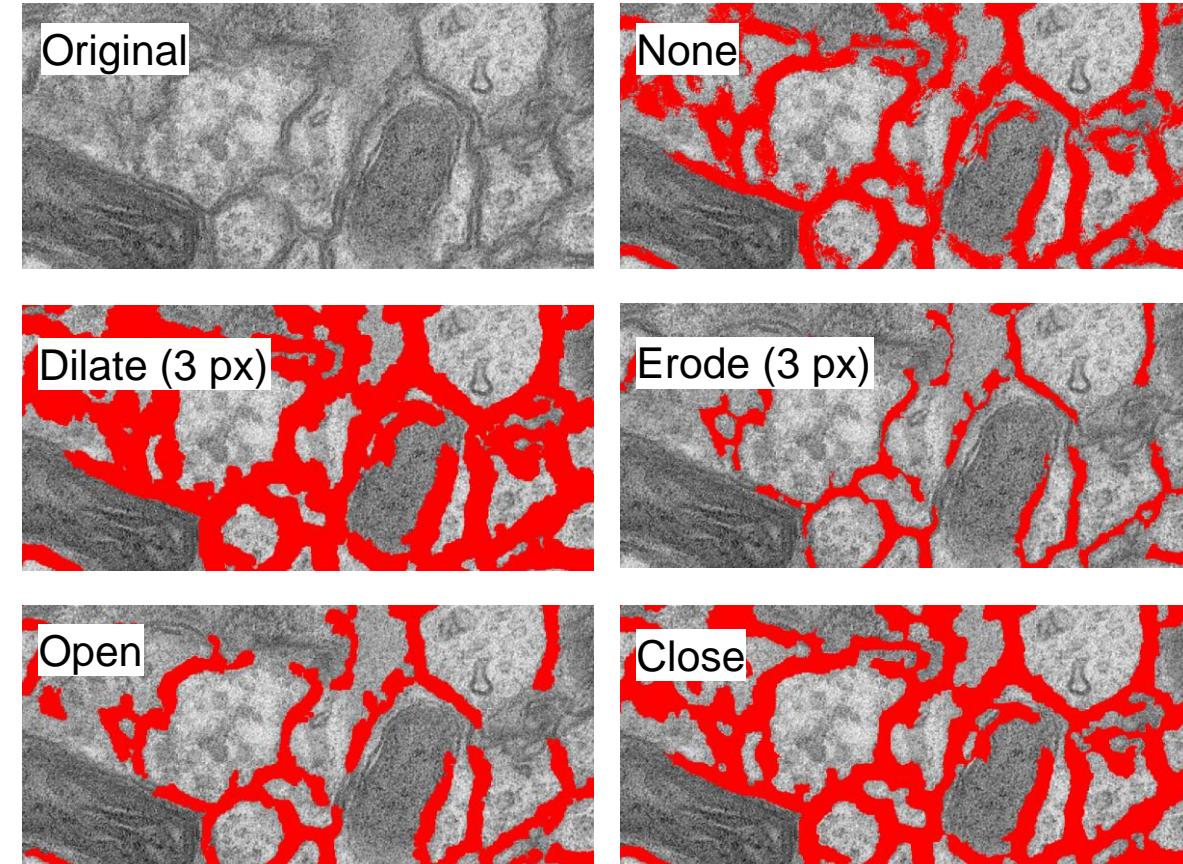
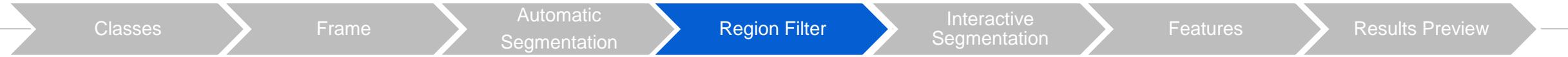
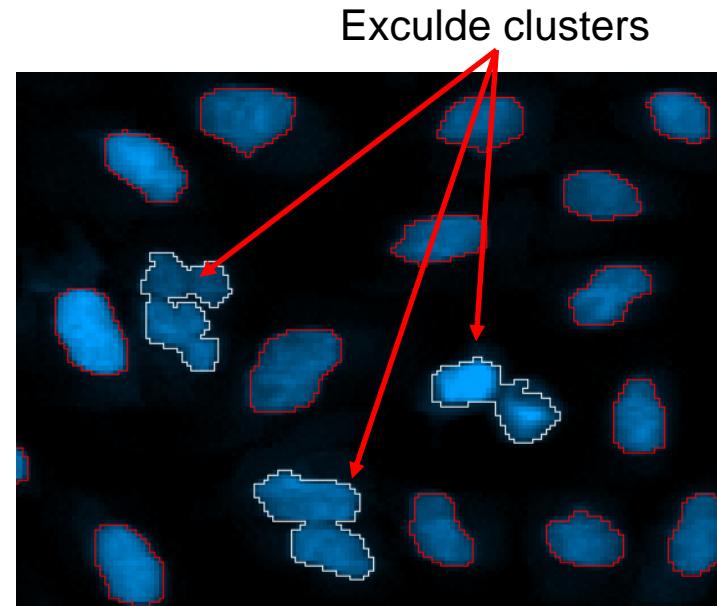
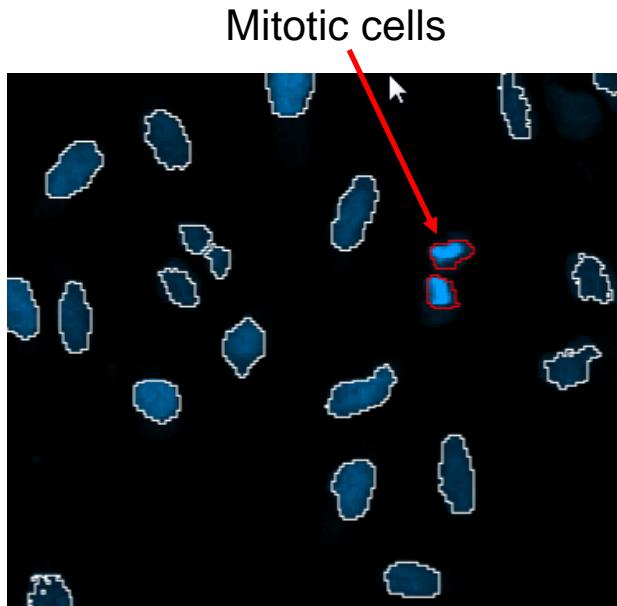


Image Analysis Step 4: Region Filter

Refine/Limit detected objects by conditions



- Region Filters allow to further refine/limit the segmented objects.
- Detect only specific objects, that fulfill certain criteria rather than all segmented objects
 - Only mitotic cells
 - Only nicely separated cells, not clusters
 - ...



4/7 Region Filter

Back

Execute Interactive

Base 0
Σ Nuclei 1
Nucleus DAPI 2

Define region filters for segmented objects

Edit Copy to All

Name	Minimum	Maximum
Area	0.660	2.710
Circularity	0.662	1.000

Undo Redo Reset

This screenshot of the software interface shows the 'Region Filter' dialog box. It includes a tree view of filters applied ('Base', 'Σ Nuclei', 'Nucleus DAPI'), execution checkboxes, and a table for defining region filters based on area and circularity. Two images below the dialog show examples of how these filters are applied to a cell population.

Define Features

Define what to measure for the detected objects



6/7 Features

Back ^

Base 0
Σ Nuclei 1
Nucleus DAPI 2

Features of individual regions

Edit Copy to All

Name	Display
ID	<input type="checkbox"/>
Area	<input type="checkbox"/>
Intensity Mean Value of channel 'DAPI'	<input type="checkbox"/>
Circularity	<input type="checkbox"/>

Edit Copy to All

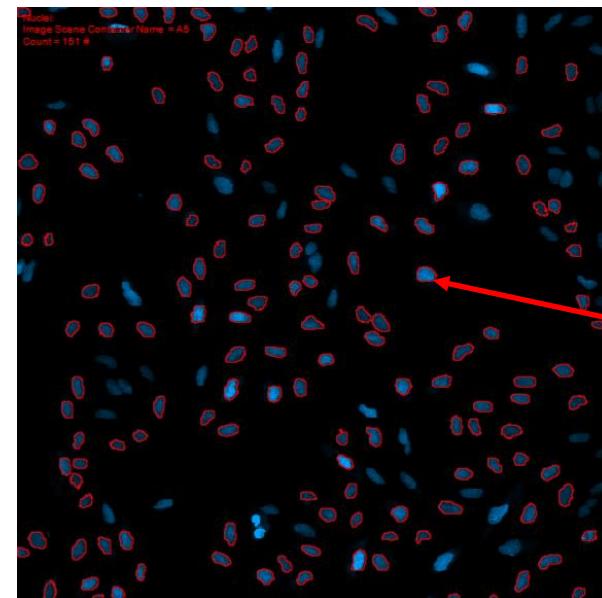
Name

Annotation Options

Color

You can define features for “Nuclei” and “Nucleus” independently
Select features from a list of ~ 100 features

- Geometry
- Intensity
- Image
- Position



“Nuclei”

- Image Scene Container Name: A5
- Count: 151

“Nucleus”

- Area: $2.49 \text{ }\mu\text{m}^2$
- Mean Intensity: 154
- Roundness: 0.8

Define Features

Define what to measure for the detected objects



6/7 Features

The screenshot shows the 'Features' interface with a tree view on the left and a table of feature settings on the right. The tree view includes categories like 'Base', 'Σ Nuclei', 'Nucleus', 'Σ Sp100 Stat', 'Sp100', and 'EGFP'. The table on the right lists various features such as 'Name', 'ID of the parent', 'ID', 'Count', 'Intensity Mean Value of channel 'EGFP'', 'Annotation Options', and 'Color'.

Features of all regions

Edit

Features for „Sp100 Stat“

Feature Selection

Name	Display	Copy
ID of the parent	<input type="checkbox"/>	<input type="button" value="▼"/>
ID	<input type="checkbox"/>	<input type="button" value="▼"/>
Count	<input type="checkbox"/>	<input type="button" value="▼"/> Nucleus
Intensity Mean Value of channel 'EGFP'	<input type="checkbox"/>	<input type="button" value="▼"/> Nucleus

In case of more complex Classes/Class structures, use the “copy-to” functionality to copy features from one class to another (collect all results in one table)

Result table for „Nucleus“

ID	Area [µm²]	Sp100 Stat...	Sp100 Stat...
A	B	C	D
1	14	319.654	14
2	12	212.405	30
3	16	247.973	52
4	18	183.832	27
5	4	27.415	2
6	19	295.594	41
7	7	193.159	32
8	9	195.454	41
		261.781	931.038
			862.518

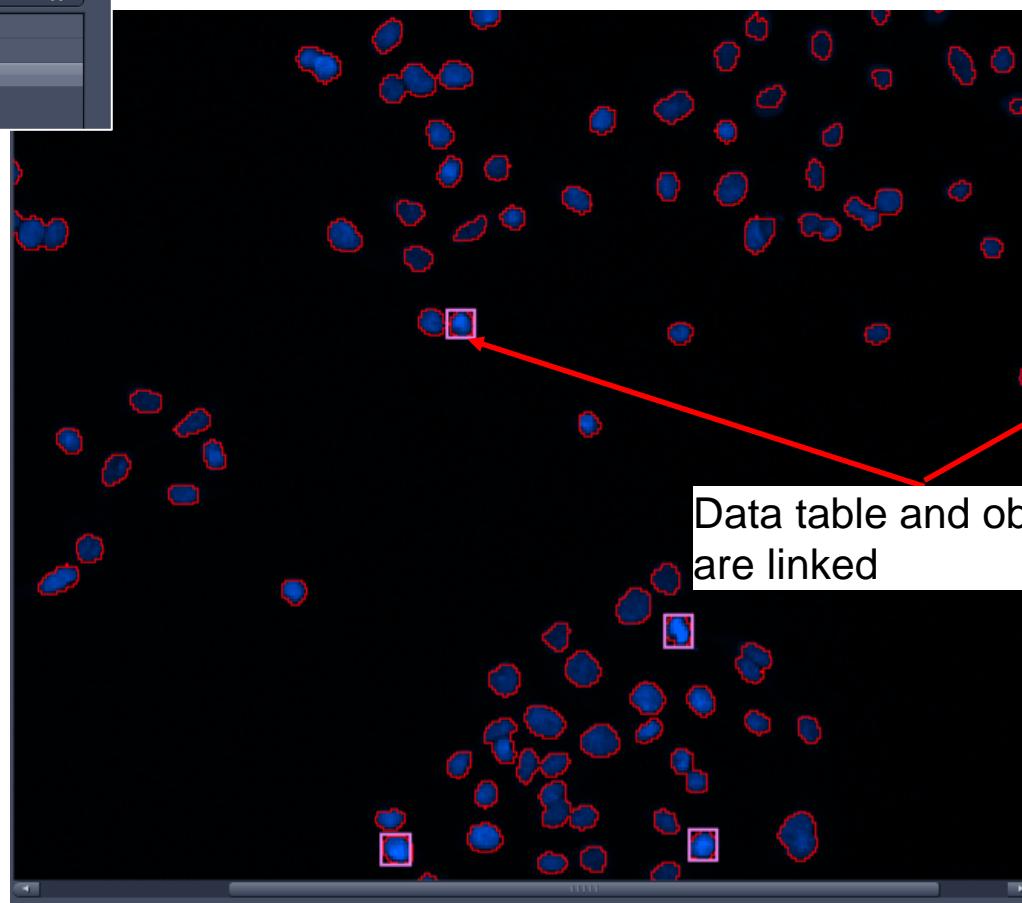
View results after Image Analysis

Link between table and detected objects



The screenshot shows the software's control panel with various checkboxes and dropdowns. A red arrow points to the 'Nucleus' checkbox under the 'Show Objects' section. Another red arrow points to the 'Create Table(s)' button.

Highlight „Nucleus“ / „Nuclei“ to see corresponding tables



Sort the data to see, e.g. the nuclei with highest mean intensity

ID	Area [µm²]	Intensity...	Circularity
A	B	C	D
1	137	0.970	57.268
2	110	0.890	55.348
3	140	0.880	52.443
4	28	0.890	49.854
5	88	0.700	49.171
6	146	1.130	49.088
7	136	1.080	48.787
8	89	4.430	48.456
9	42	1.360	48.287
10	149	3.080	46.789
11	118	0.830	46.313
12	107	0.630	45.984
13	144	1.780	45.202
14	82	1.150	44.678
15	61	0.720	44.250
16	106	1.060	44.226
17	57	1.080	43.694
18	117	1.090	43.651
19	96	0.550	43.600
20	36	0.850	43.541
21	54	0.450	43.156
22	129	0.640	42.953
23	53	0.730	42.384
24	25	0.500	41.400
25	26	0.530	41.321
26	98	0.790	41.114

Export data after Image Analysis

Generate result tables



The screenshot shows the software's control panel with various checkboxes and a dropdown menu labeled "Create Table(s)". The "Create Table(s)" menu is open, displaying three options: "Nucleus", "All classes (separately)", and "All class/classes (concatenated)". A red arrow points from the "Nucleus" option to the explanatory text above.

Highlight a specific “Classes”/“Class” to export data for this “Classes”/“Class” only (one table only)

Export data all individual “Classes” and “Class” in separate tables.

In this case, 4 separate data sets:

The screenshot shows the software's interface with four tabs at the top: "IA.czi", "IA Regions Nuclei*", "IA Regions Greens*", "IA Region Nucleus*", and "IA Region Green*". Below the tabs, a table is displayed with columns "ID" and "Count [#]". The first row shows ID 1 with Count 10. A red arrow points from the "All classes (separately)" option in the dropdown menu to this table.

ID	Count [#]
1	10

The subsequent data can be saved as .csv
If the input data has higher dimension, e.g. time series,
multi-positions, the final data will be concatenated!

Export two data sets where all tables for “Classes” and “Class”, respectively, are concatenated (2 data tables)

The screenshot shows the software's interface with tabs "IA.czi", "IA Region*", and "IA Regions*". Below the tabs, two tables are displayed side-by-side. The first table has columns "ID", "Count [#]", and "Area [µm²]". The second table has columns "ID", "Count [#]", and "Intensity Sum...". The first row of the first table shows ID 1 with Count 10 and Area 22.36. The second row shows ID 2 with Count 12 and Intensity Sum 260,913.00. A red arrow points from the "All class/classes (concatenated)" option in the dropdown menu to this section.

ID	Count [#]	Area [µm²]
1	10	22.36
2	12	260,913.00

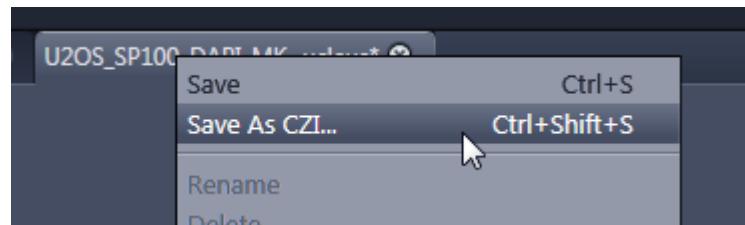
ID	Count [#]	Intensity Sum...
1	12	260,913.00

Export data after Image Analysis

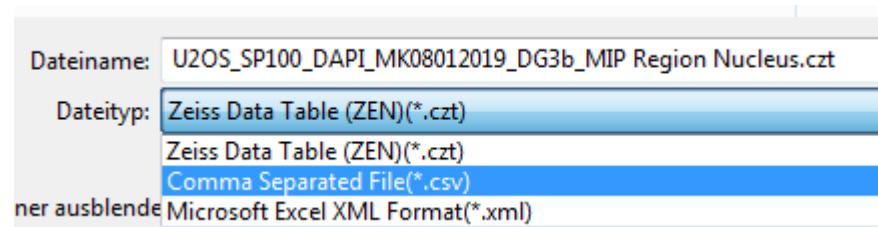
Export into *.csv format



Right-click on tab containing the table



Select “Save As CZI...”



Select *.csv

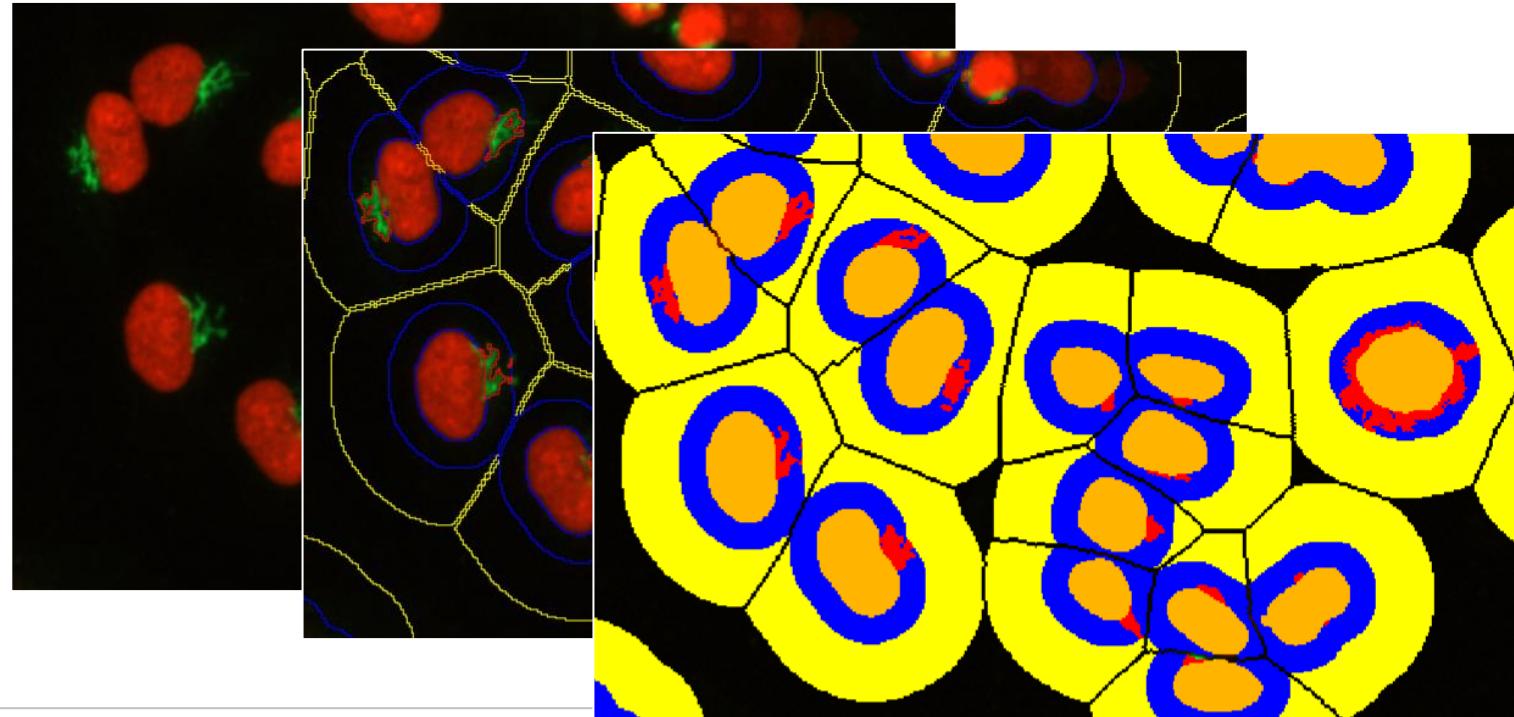
Typical ZOI Applications



Applications in cell biology, drug discovery, in-vitro assays, endpoint assays,
That require to detect objects outside of the object used for segmentation

e.g.:

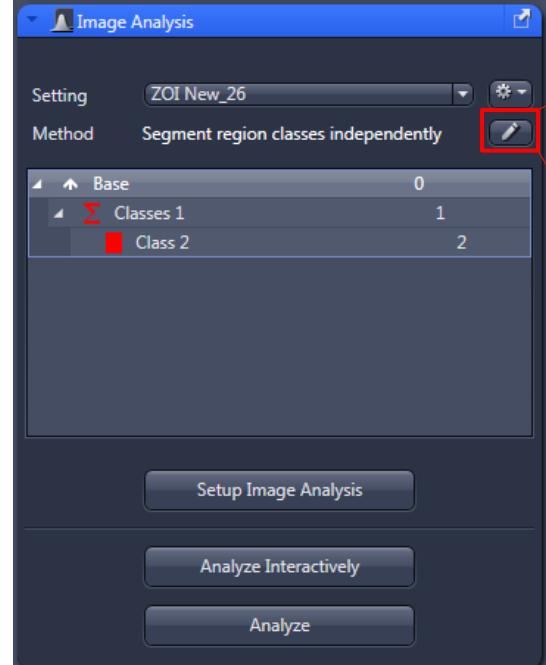
- Cytoplasm-Nucleus Translocation
- Protein Localization
- Actin, Mitochondria....



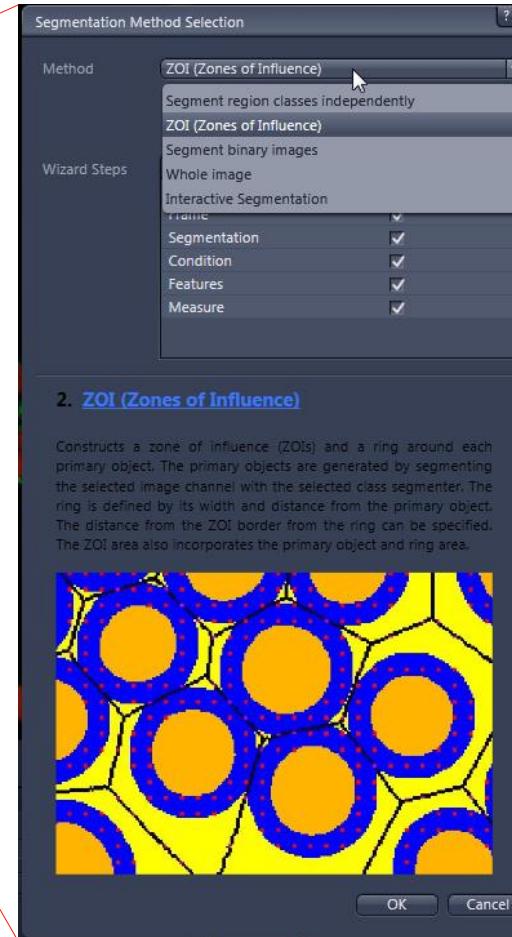
e.g. quantify the total green
structures (area and
intensity) per cell

Creating an Image Analysis Setting using ZOI

Select the ZOI (Zones of Influence) Method



Create a new Image Analysis Setting and choose ZOI (Zones of Influence) as Segmentation Method

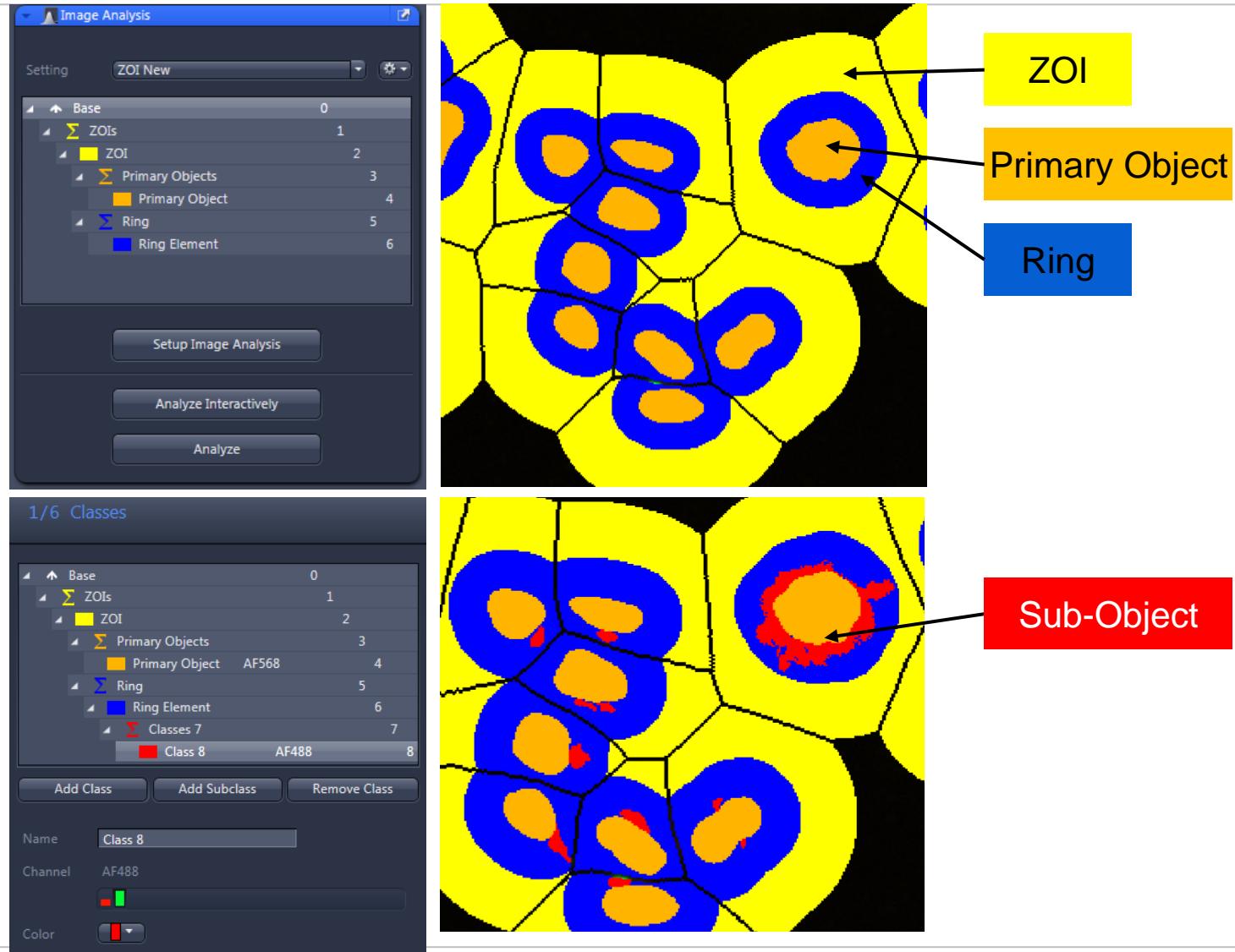


Set up an Image Analysis Setting using ZOI



The ZOI-method will create the necessary classes automatically:

- **ZOIs/ZOI**: the area (zone of influence) that is attributed to the primary objects
- **Primary Objects/Primary Object**: the objects that identify the cell (e.g. nuclei)
- **Ring/Ring Element**: automatically generated around each primary object to measure parameters or to detect sub-objects
- **Optional: Ring sub-object**: objects you want to measure per cell other than the nucleus



Adjust Ring Parameters

Set Width and Distance



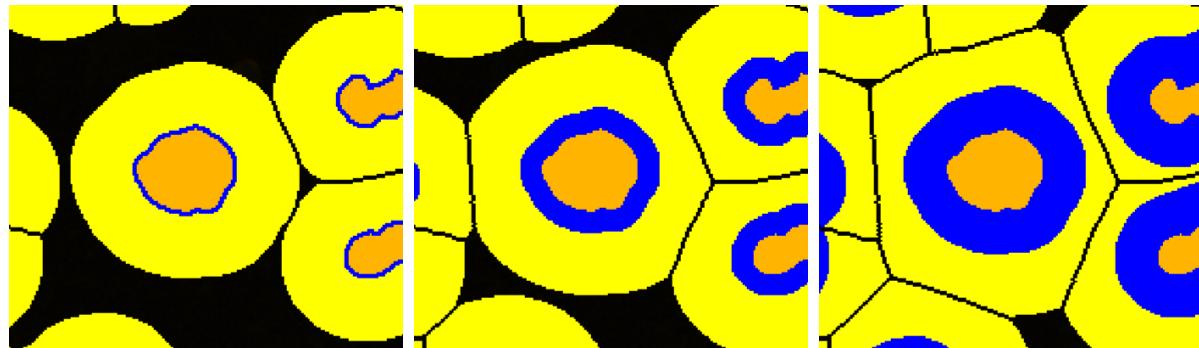
3/6 Automatic Segmentation

Execute Interactive

- Base
- ZOIs
 - ZOI
 - Primary Objects
 - Primary Object AF568
 - Ring
 - Ring Element
 - Classes 7
 - Class 8 AF568

Ring Distance: 0

Ring Width: 10



Width: 2

Width: 10

Width: 20

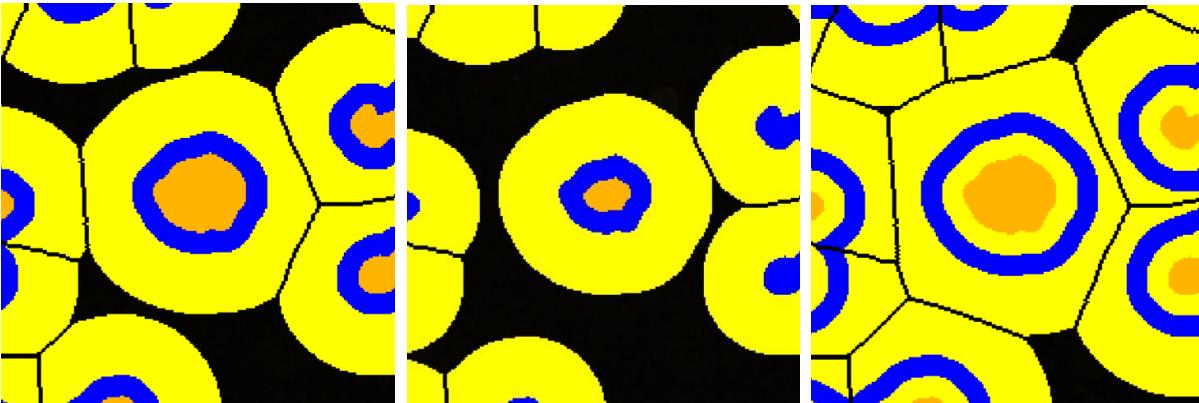
3/6 Automatic Segmentation

Execute Interactive

- Base
- ZOIs
 - ZOI
 - Primary Objects
 - Primary Object AF568
 - Ring
 - Ring Element
 - Classes 7
 - Class 8 AF568

Ring Distance: 0

Ring Width: 10



Distance: 0

Negative values

Positive values

Attribute Ring-Features to Primary Object

Use the „copy“-Function



Copy „Ring“/„Actin_stat“ features to the nucleus

5/6 Features

Back ^

Base	0
Σ ZOIs	1
└ ZOI	2
└ Primary Objects	3
└ Nucleus	H3258
└ Ring	4
└ Ring Element	5
└ Actin_stat	6
└ Actin	EGFP
	8

Features of all regions

Edit Copy to All

Name	Display
ID of the parent	<input type="checkbox"/>
ID	<input type="checkbox"/>
Area	<input type="checkbox"/>
Intensity Mean Value of channel 'EGFP'	<input type="checkbox"/>

Annotation Options

Color

Features Ring

Name	Display	Copy
ID of the parent	<input type="checkbox"/>	<input type="button" value="Nucleus"/>
ID	<input type="checkbox"/>	<input type="button" value="Nucleus"/>
Area	<input type="checkbox"/>	<input type="button" value="Nucleus"/>
Count	<input type="checkbox"/>	<input type="button" value="Nucleus"/>

1 Σ ZOIs
1 ZOI
→ Primary Objects
→ Nucleus

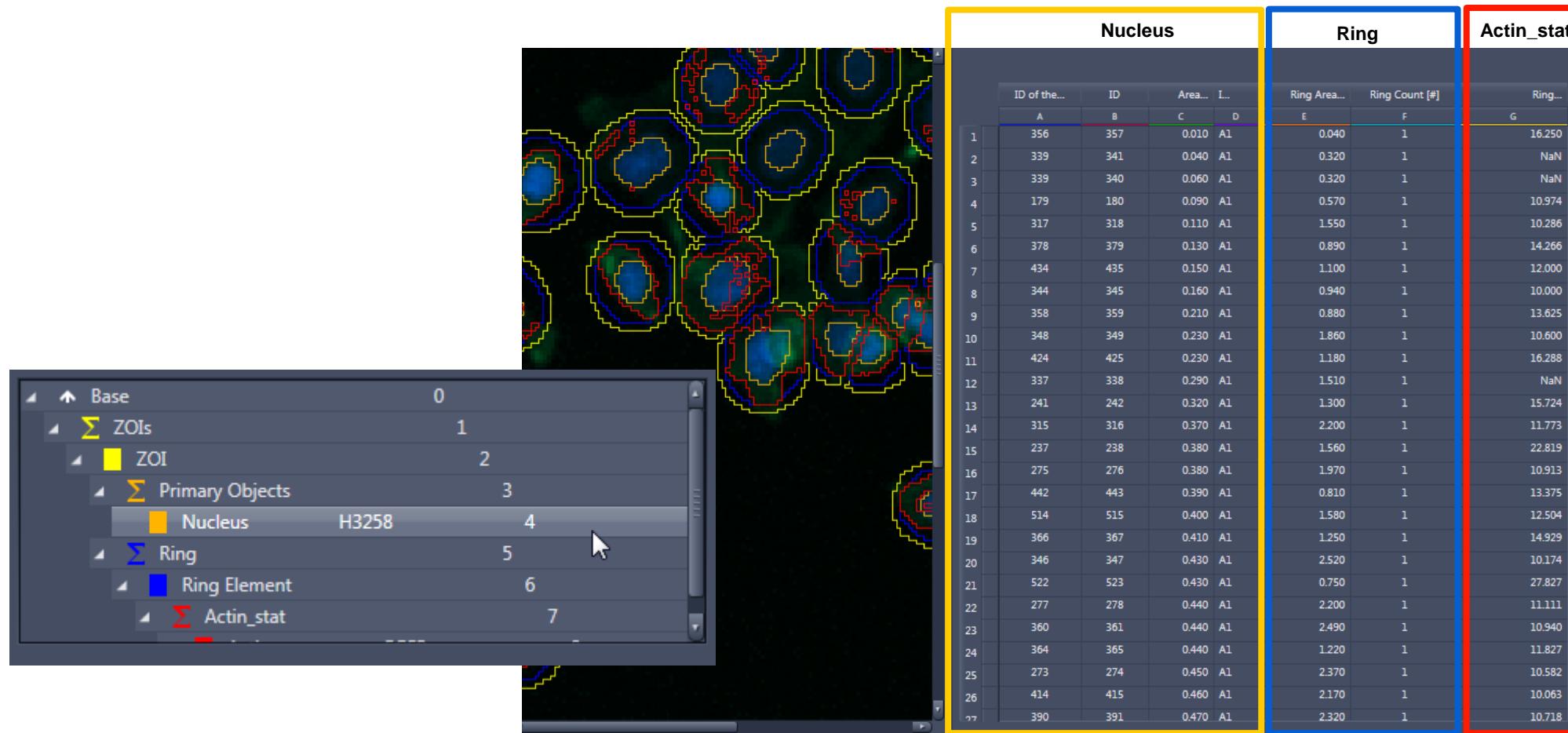
Features Actin_stat

Name	Display	Copy
ID of the parent	<input type="checkbox"/>	<input type="button" value="Nucleus"/>
ID	<input type="checkbox"/>	<input type="button" value="Nucleus"/>
Area	<input type="checkbox"/>	<input type="button" value="Nucleus"/>
Intensity Mean Value of channel 'EGFP'	<input type="checkbox"/>	<input type="button" value="Nucleus"/>

r Σ Ring
r Ring Element
r Primary Objects
r Nucleus

Results

Features for „Nucleus“



The results table for „Nucleus“ also contains the copied features of „Ring“ and „Actin_stat“

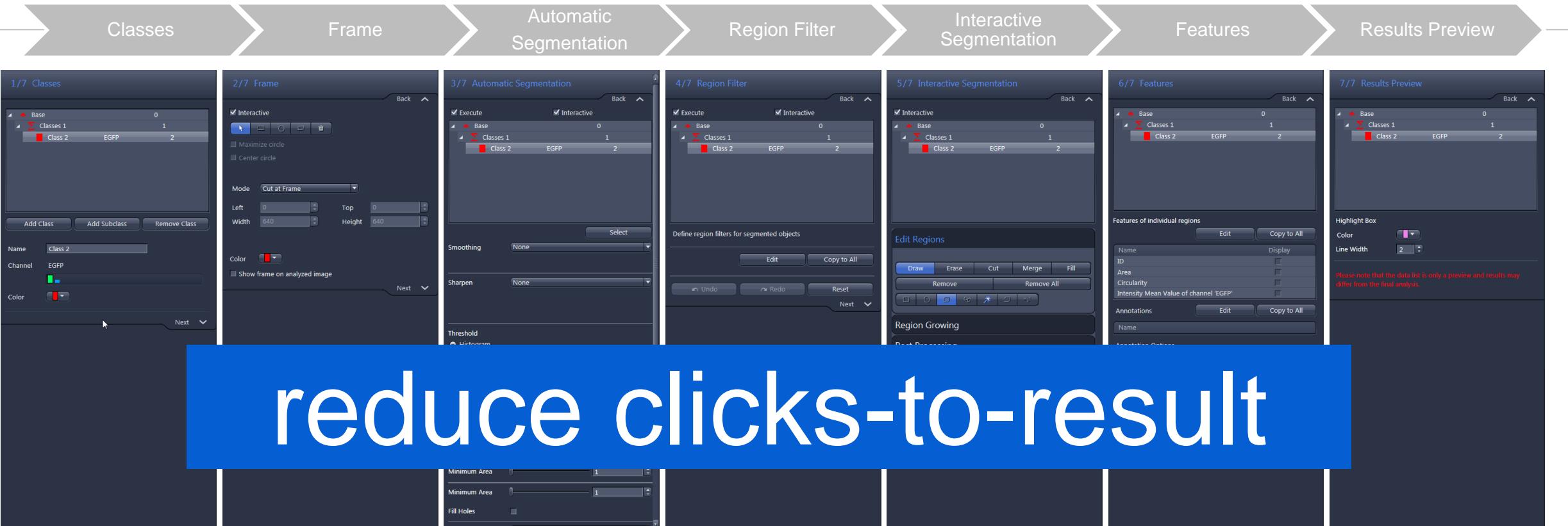
Image Analysis Wizard incl. Zone Of Influence (ZOI)

BioApps

Introduction Macro Environment (OAD) in ZEN

Scripting: Image Analysis/BioApps

Image Analysis in ZEN blue Status Quo



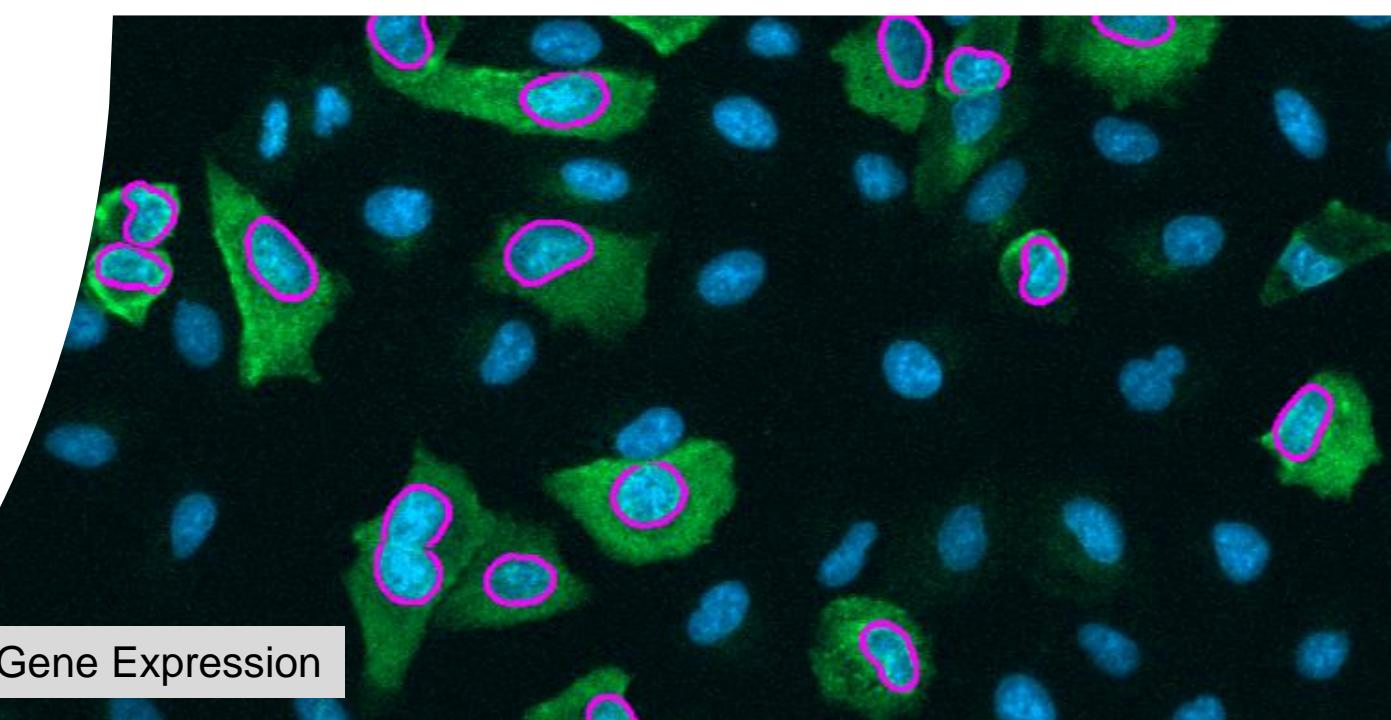
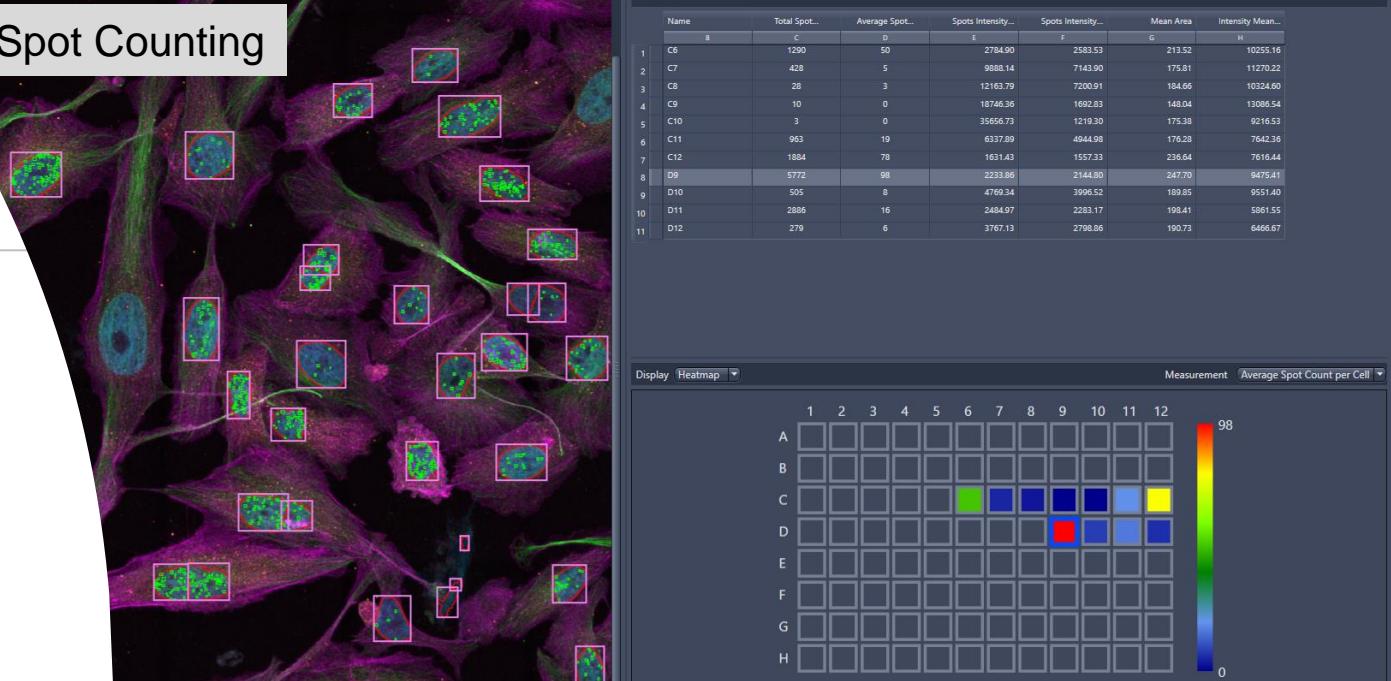
The ZEN Image Analysis Wizard is powerful but often overwhelming

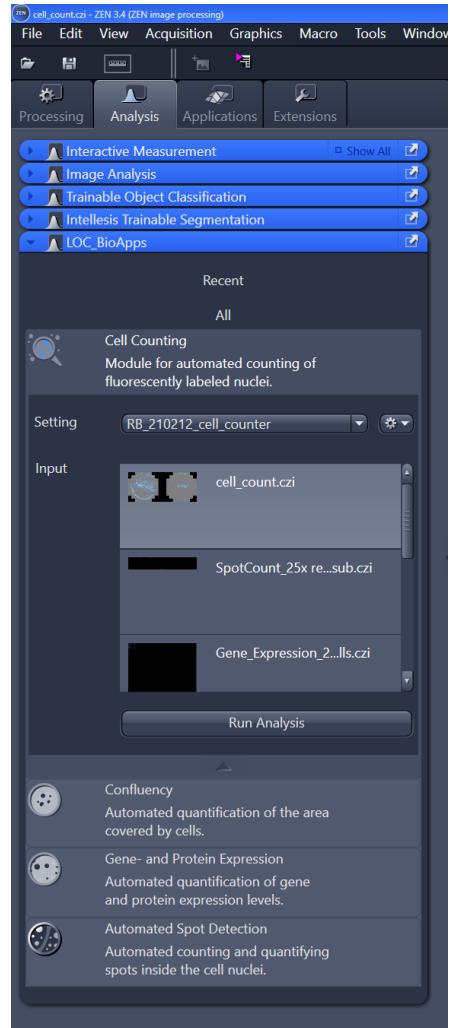
- Seven steps to perform an image analysis
- different segmenters (e.g. ZOI)
- segmentation options (Threshold, Edge, Intellesis,...)
- Many measurement features

New BioApps Modules Dedicated Image Analysis Solutions

Simple to use BioApps:

- „Out of the box“ image analysis solutions for common assays
- Optimized for measurements in screening applications with multi-well setups
- Simple setup of the analysis
- Predefined parameters for ease-of-use
- Dedicated and interactive result display: links segmented objects, list entries and plots for quality control and data exploration.





New BioApps will can be found the BioApps tool on the Analysis tab in ZEN blue

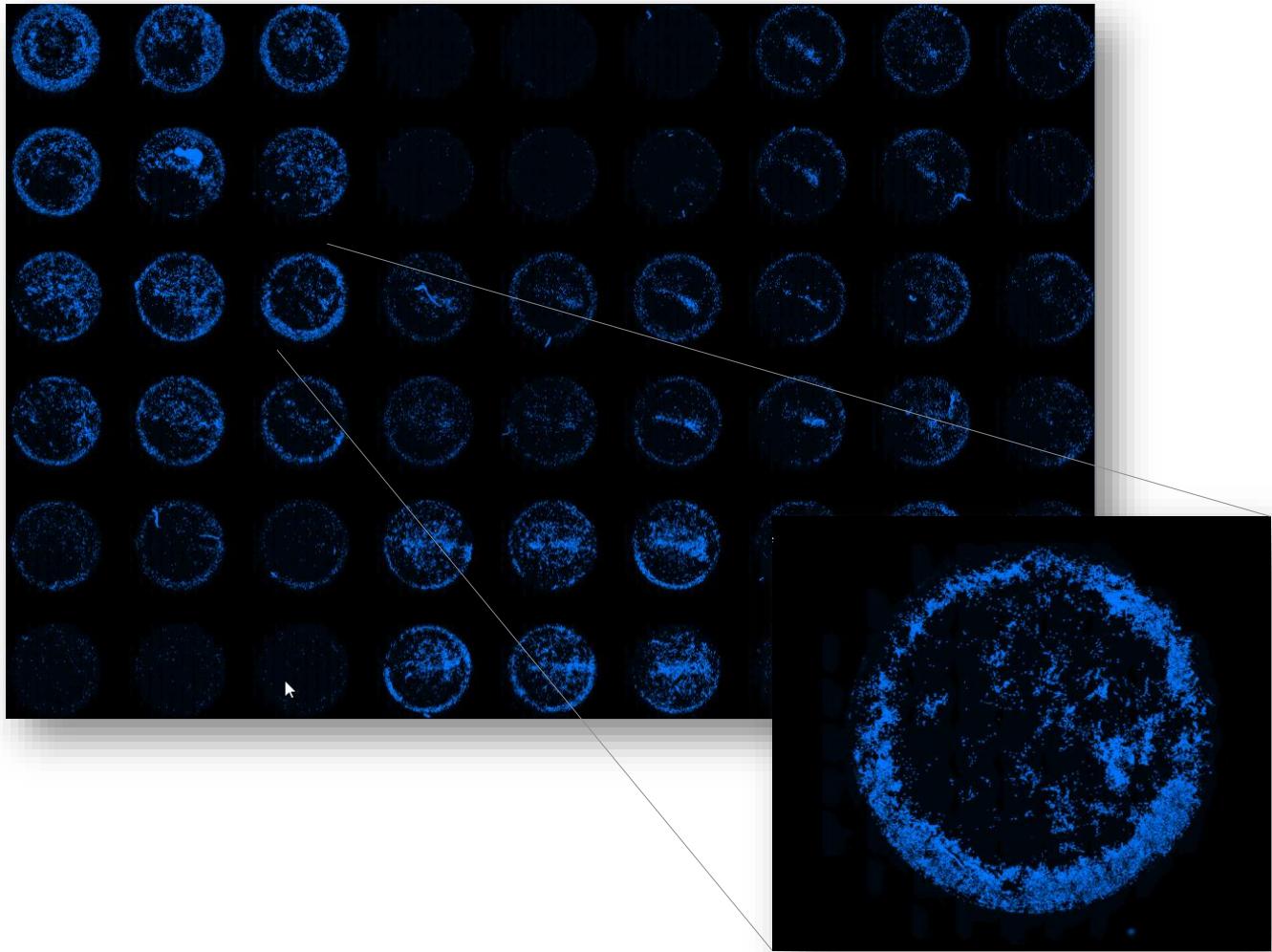
BioApps available with ZEN blue 3.4:

- **Cell Counting**
- **Confluency**
- **Gene- and Protein Expression**
- **Automated Spot Detection**

Compatibility with the Image Analysis Wizard:

- All BioApp settings can be extracted from an analyzed image and converted into a setting for the Image Analysis Wizard

Cell Counting Applications



Cell Counting:

Simple automated image analysis workflow customized for counting of fluorescently labeled cell nuclei

Typical applications:

- automatic monitoring of cell numbers
- Proliferation, e.g., under the influence of different compound effects

Measurement features:

- Nuclei count
- density of the nuclei
- mean intensity (nuclear channel)
- mean area

Cell Counting Setup & Result View



BioApps

Define the cells you would like to count.

Name: Cell

Channel: DAPI

Color: Red

Info: Use a different color than the channel color to make the masked cells visible.

Select the method for segmentation:

Threshold-based (selected) AI-based

Apply Background Subtraction

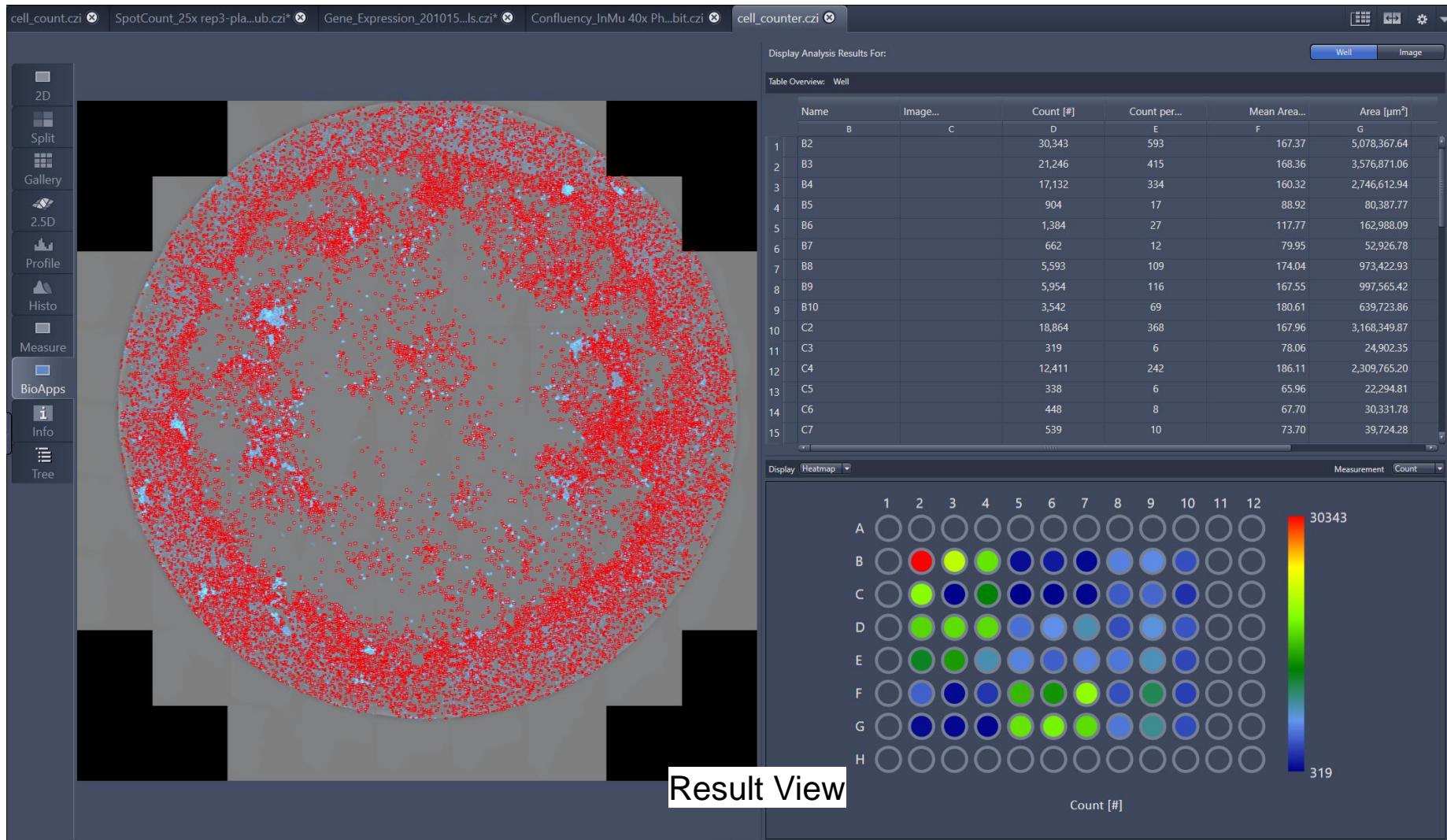
Threshold: Low (17595) High (34874)

Area (μm^2): Low (20) High (1169)

Circularity: Low (0.5) High (1.0)

Manual Automatic

Setup



BioApps

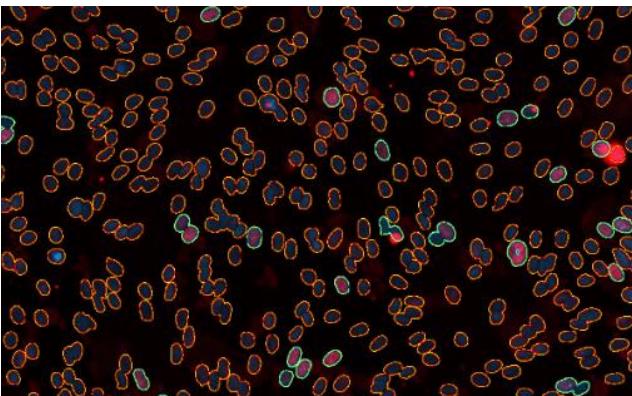
Quick Result Export



Display Objects Export

Image Result png Well csv Chart Heatmap png Other Processing Info. txt

Export...



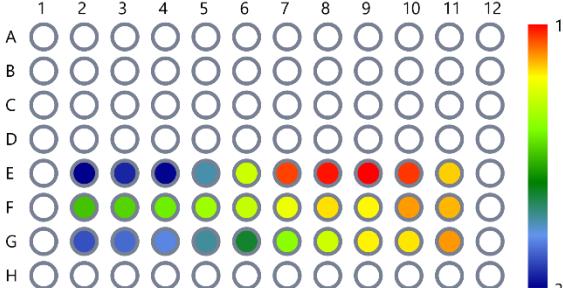
Class: Cell
Channel Name: H3342

Preprocessing:
Smoothing: None
Background Reduction: None
Sharpening: None

Segmentation: Threshold
Mode: Click
Lower Value: 32
Higher Value: 50

Postprocessing:
Min Area: 10
Fill Holes: true
Separate: Watersheds
Count: 6
Suppress Invalid Image Regions: false

Region Filter:
Distance Unit: µm
Area
Minimum: 20
Maximum: 500
Circularity
Minimum: 0.5
Maximum: 1



1390
337

All Cells Count [#]

SLR22_53BP1_EdU_images_and_analysis

	D	E	F	G
Cell Transfection Intensity Measured	IntensitySum1_AF647::Intensity%			
74	21,958457	12121,4495	2105386687	
52	15,8567775	11198,1262	1714141963	
69	20,1166181	11769,6455	2062136054	
17	19,338843	11434,0799	3747302455	
08	9,8540146	12883,0841	4301661782	
91	6,8164794	12742,8324	3399392657	
91	6,62299854	14882,3879	4231003352	
32	9,49640288	13441,2353	6330633658	
31	9,73977695	12756,6847	5640610505	
34	10,9926169	12917,872	5324772680	
98	11,2385321	12782,9526	3548752172	
06	11,7908788	10075,9206	3135525739	
26	13,3616119	10104,8299	3986567609	

Export options:

- Currently displayed image
- Currently displayed chart (Heatmap / XY Plot / Histogram) as png, jpg, bmp, tif)
- Currently displayed table with measurement features (csv)
- Processing information (txt)

Image Analysis Wizard incl. Zone Of Influence (ZOI)

BioApps

Introduction Macro Environment (OAD) in ZEN

Scripting: Image Analysis/BioApps

ZEN is only part of the workflow Open Application Development



ZEN Connect

OAD

Python

Excel

Scripting

MATLAB

ZEN

KNIME

Analysis

Fiji

Automation

Machine Learning

Extensions

Simplify

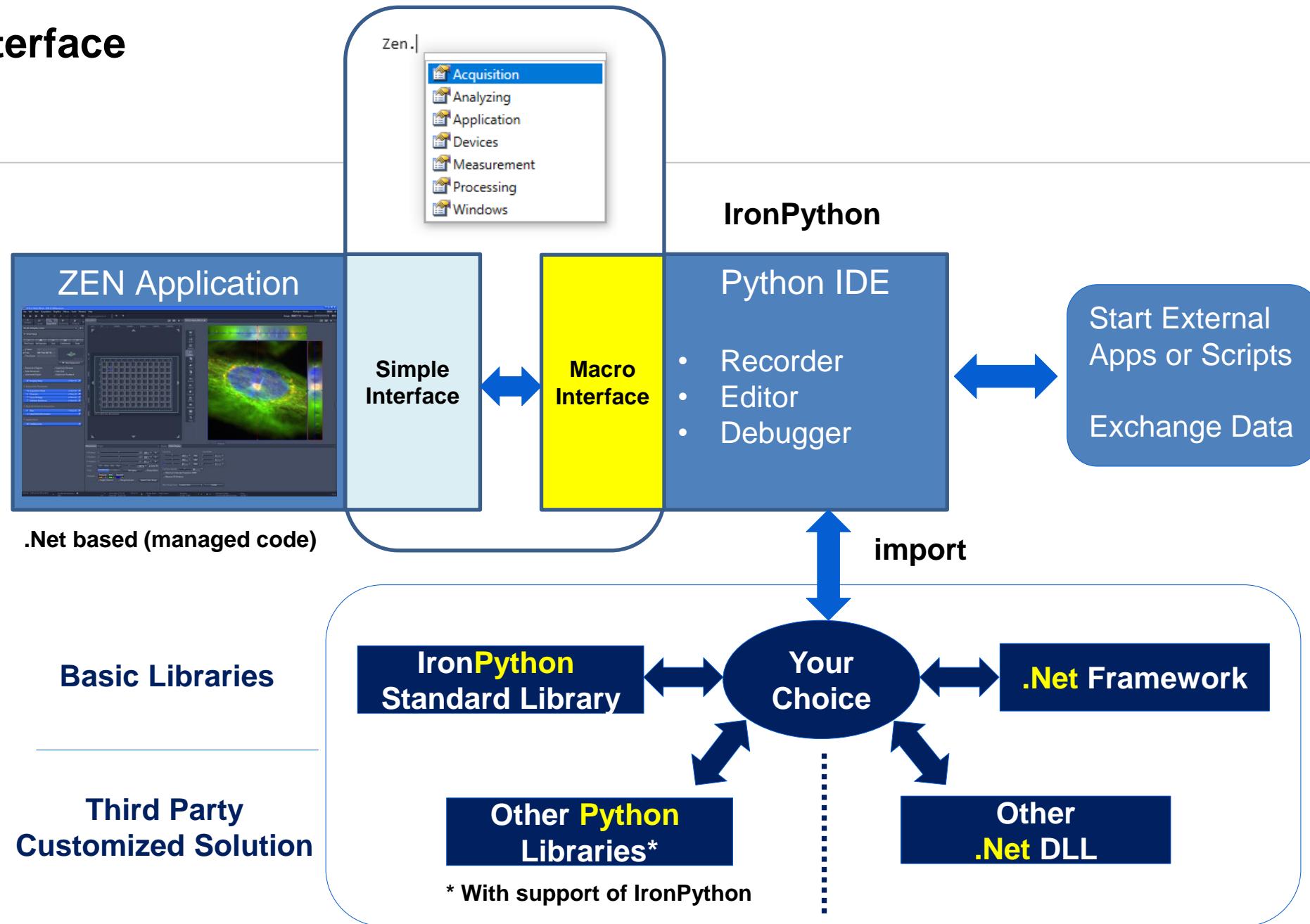
BioFormats

Open Application Development (OAD) General Concept and Key Features



- Macro environment: simplify, customize and automate your workflows using Python scripts
- Remote control of ZEN via TCP/IP and COM interface
- BioFormats (CZIReader) allow easy access to CZI files from many external applications.
- Analyze and Exchange data with applications like Fiji, Python, Knime, CellProfiler, Icy, MATLAB, Excel, etc.
- Create “smart” experiments with Experiment Feedback: modify the acquisition parameters on-the-fly based on Online Image Analysis and external inputs
- Read CZIs on any platform:
 - libCZI (C++) for cross-platform applications (Windows, Linux, MacOS)
 - ZeissImgLib (c#) for Windows only
- Reading CZI in Python aicsimageio / aicspylibcz

Macro Interface



Macro Interface – General Remarks

- ZEN uses IronPython as its internal scripting language
→ this allows using ZEN functionality directly, but not SciPy, NumPy etc.
- Not everything can be recorded, but that does not mean, there is not command for a specific functionality
- script editor has a built-in help file
- Almost everything in ZEN is “done” via settings
- Built-in intellisense auto-completion

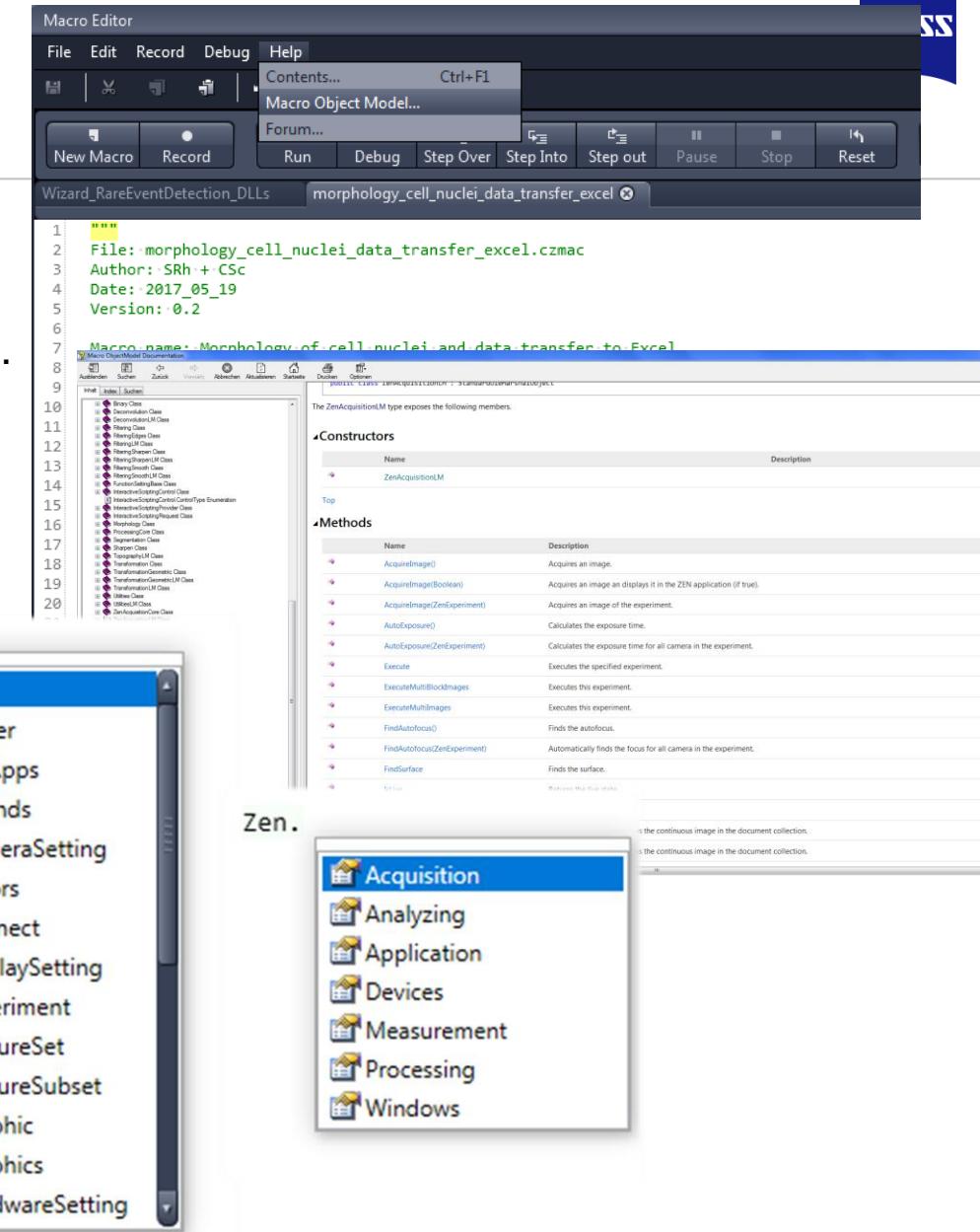


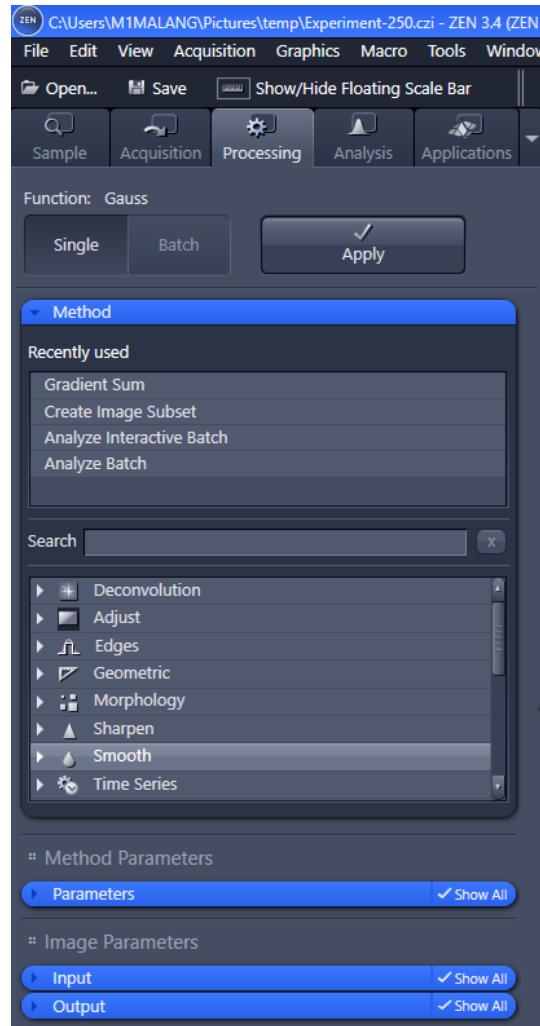
Image Analysis Wizard incl. Zone Of Influence (ZOI)

BioApps

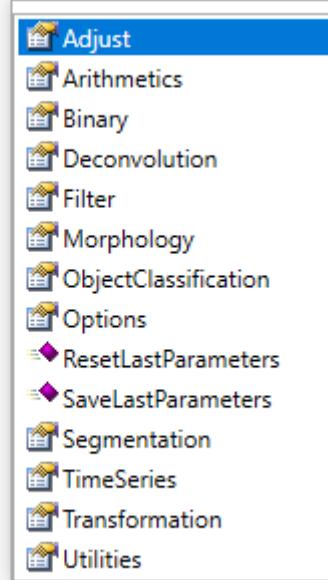
Introduction Macro Environment (OAD) in ZEN

Scripting: Image Analysis/BioApps

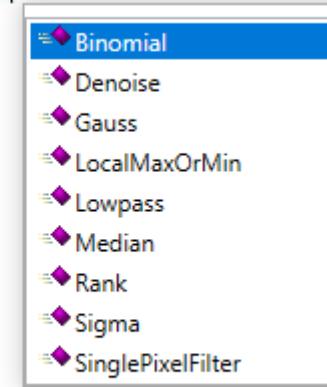
Image Processing in OAD



Zen.Processing.|



Zen.Processing.Filter.Smooth.|

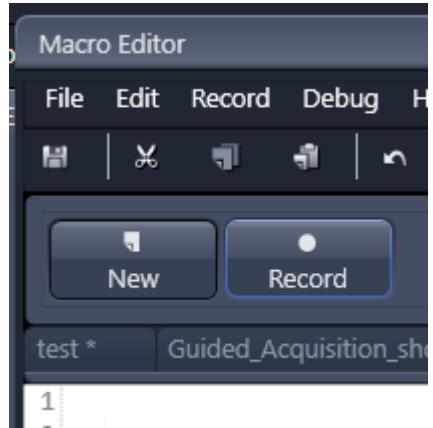
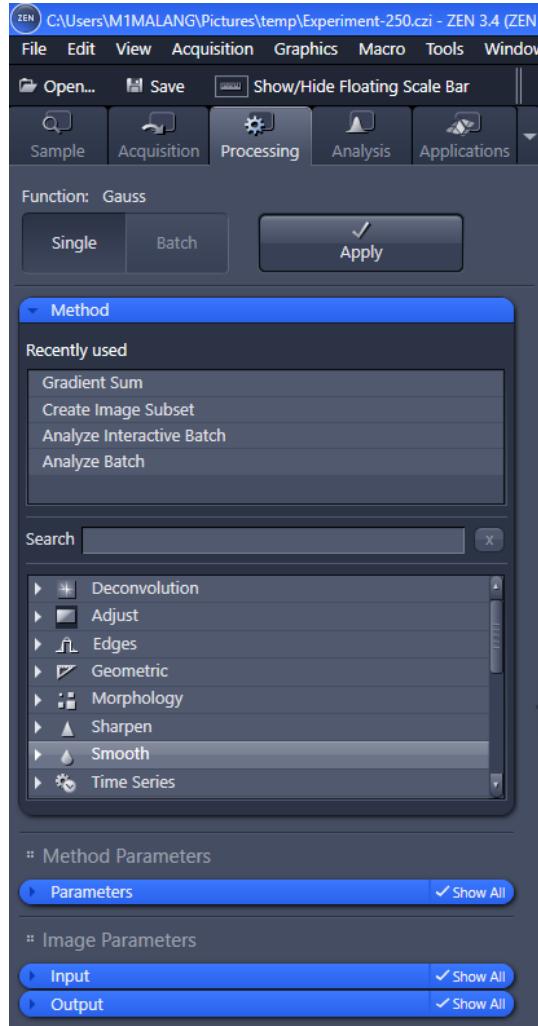


Zen.Processing.Filter.Smooth.Gauss()

```
(ZenImage input, double sigma, bool interactive) : ZenImage
(ZenImage input, double sigma) : ZenImage
(ZenImage input) : ZenImage
(ZenImage input, bool interactive) : ZenImage
(ZenImage input, double sigmaX, double sigmaY, double sigma3rd, ZenThirdProcessingDimension dimension3rd, bool interactive) : ZenImage
(ZenImage input, double sigmaX, double sigmaY, double sigma3rd, ZenThirdProcessingDimension dimension3rd) : ZenImage
(ZenImage input, GaussSetting parameter) : ZenImage
(ZenImage input, GaussSetting parameter, bool interactive) : ZenImage
```

Image Processing in OAD

Record Image Processing Functions



```
# ***** Recorded Code Block *****
image1 = Zen.Application.Documents.GetByName("C:\...\temp\Experiment-250.czi")
image2 = Zen.Processing.Filter.Smooth.Gauss(image1, 1.3, 1.3, 1.3,
ZenThirdProcessingDimension.None, True)
Zen.Application.Documents.Add(image2)
# ***** End of Code Block *****
```

BioApps in OAD Automate Worflows



Automate your workflows with basic OAD Support for all BioApps

```
# get image
img = Zen.Application.ActiveDocument

# load setting for Cell Counting
cellcountsetting = ZenBioApps.Settings.CellCountingSetting("test.czccct")

# Analyze image
ZenBioApps.CellCounting.Analyze(img, cellcountsetting)
```

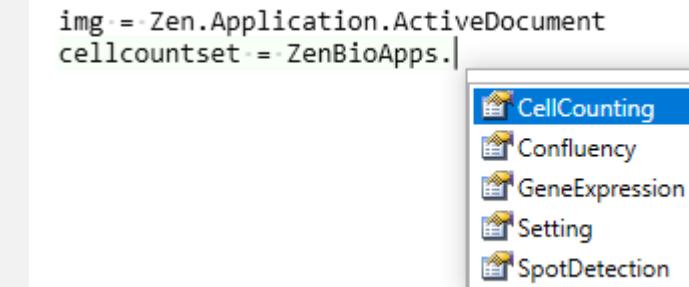


Image Analysis in OAD

Execute Image Analysis and generate Tables



```
ias = ZenImageAnalysisSetting()  
ias.Load('Translocation')  
Zen.Analyzing.Analyze(image, ias)
```

Zen.Analyzing.

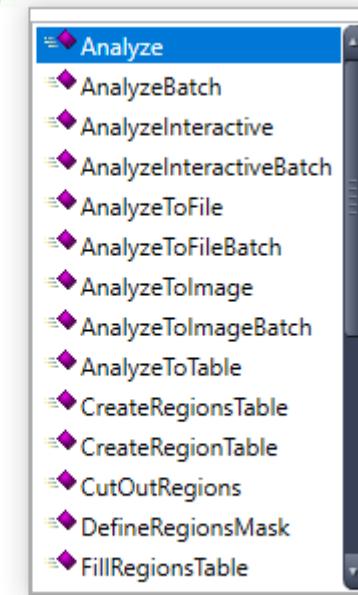


Image Analysis in OAD

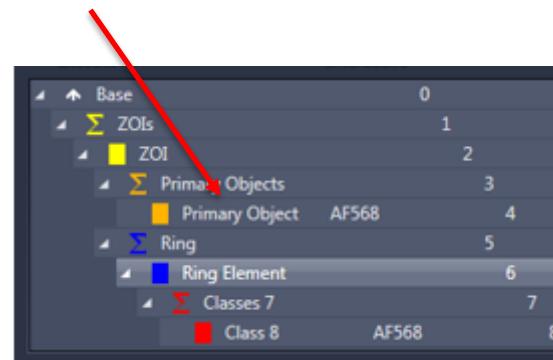
Execute Image Analysis and generate Tables



```
ias = ZenImageAnalysisSetting()  
ias.Load('Translocation')  
Zen.Analyzing.Analyze(image, ias)
```

Zen.Analyzing.

```
# Create data list with results for the Statistics class "primary objects"  
table_all = Zen.Analyzing.CreateRegionsTable(image, "Primary Objects")  
Zen.Application.Documents.Add(table_all)  
  
# Create data list with results for each individual object of the class "primary object"  
table_single = Zen.Analyzing.CreateRegionTable(image, "Primary Object")  
Zen.Application.Documents.Add(table_single)
```



Base	0
Σ ZOIs	1
└ ZOI	2
└ Primary Objects	3
└ Primary Object	4
└ Ring	5
└ Ring Element	6
└ Classes	7
└ Class 8	8

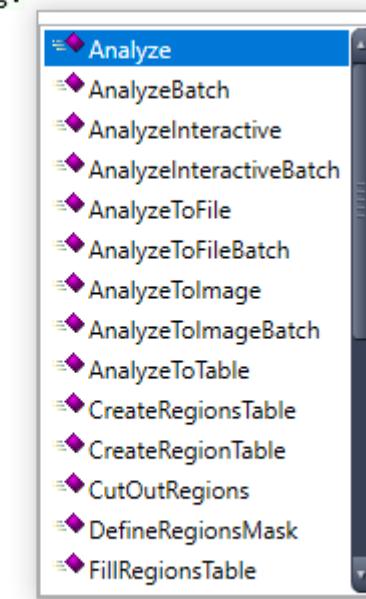


Image Analysis in OAD

Get Polygons for Regions



```
# Get regions
regs = Zen.Analyzing.GetRegions(img, "Primary Object")

# for all regions get the polygon
for reg in regs:
    poly = reg.GetPolygon()
```

Try out BioApps

Set up an Image Analysis in ZEN

Scripting: BioApps & Image Analysis

Automation

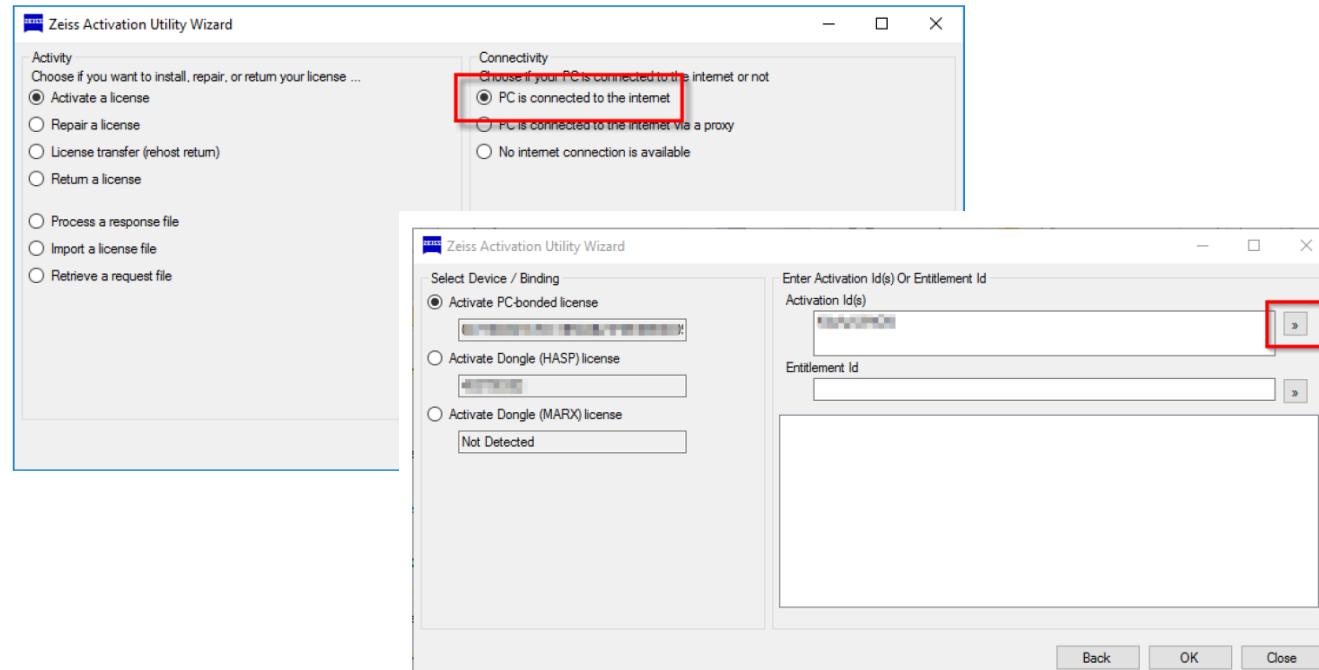
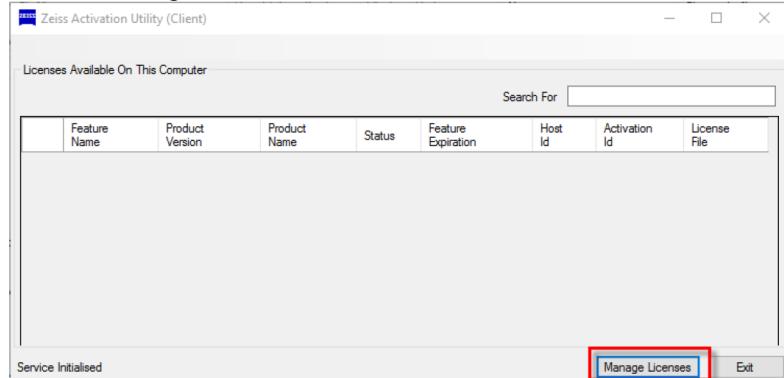
Prerequisites



- Is ZEN installed inclusive the 3rd party python packages?
- Is ZEN licenced?

1. Start the **License Activation Tool** in the **Carl Zeiss Microscopy** folder from the Windows Start menu.

2. Click on the **Manage Licenses** button.



Use the following activation ID:

7KMfq6tTcKva

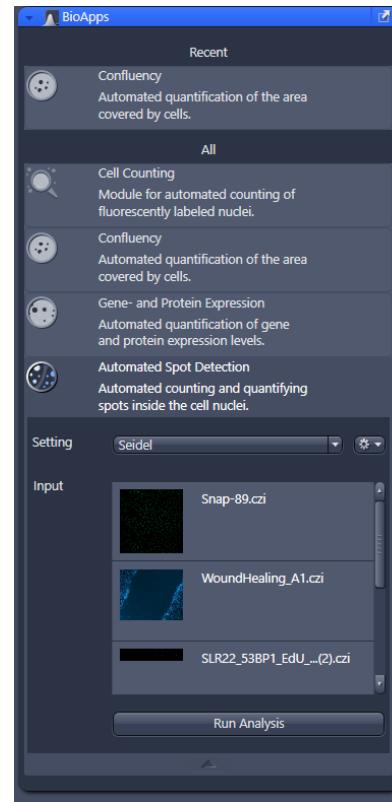
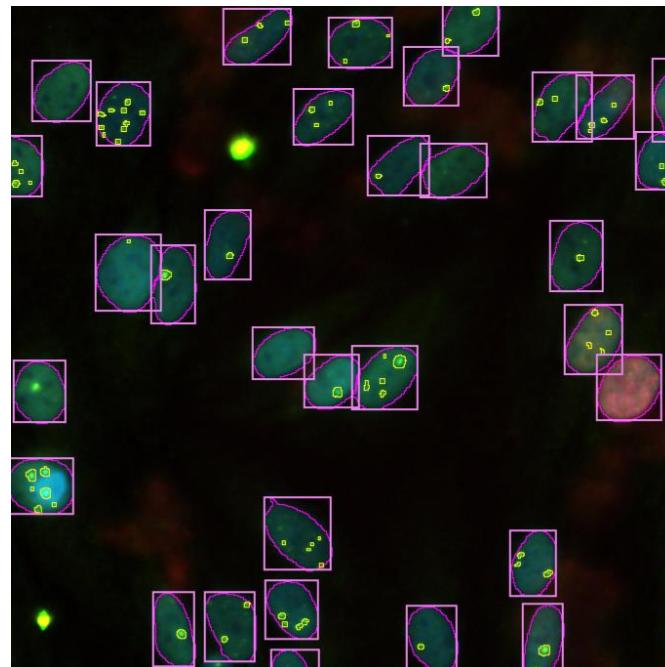
Detailed description under
www.zeiss.com/license-activation

Hands on Exercise 1

Set up Image analysis on an example using BioApps



1. Start ZEN
2. Open Image **SLR22_53BP1_EdU.czi**
3. Set up an analysis using the **Automatic Spot Counting** Bio App
4. Execute the analysis & check results

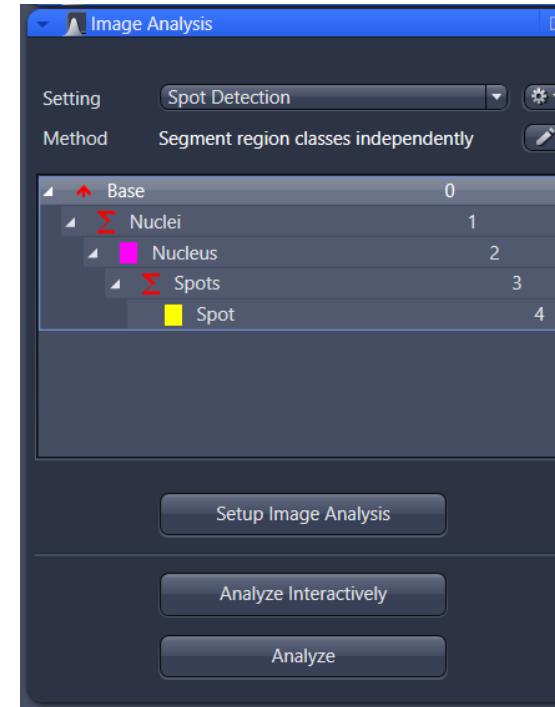


Hands on Exercise 2

Set up Image analysis on an example using the Image Analysis Wizard



1. Open Image **SLR22_53BP1_EdU.czi**
2. Set up an analysis using the **Image Analysis Wizard**
 - Define Classes
 - Set segmentation for classes
 - Define Region Filters
 - Define Measurement Features
3. Execute the Image Analysis & check results



Hands on Exercise 3

Run both created Image analysis using OAD



1. Open Image **SLR22_53BP1_EdU.czi**
2. Open the Macro Editor
3. Use the previously defined settings to execute via OAD the BioApps Analysis

```
# get image
img = Zen.Application.ActiveDocument

# load setting for SpotCounting BioApp and Analyze
setting = ZenBioApps.Settings.SpotDetectionSetting("YourBioAppSetting.czccct")
ZenBioApps.SpotDetection.Analyze(img, setting)
```

Hands on Exercise 3

Run both created Image analysis using OAD



1. Open Image **SLR22_53BP1_EdU.czi**
2. Open the Macro Editor
3. Use the previously defined settings to execute via OAD the BioApps Analysis

```
# get image
img = Zen.Application.ActiveDocument

# load setting for SpotCounting BioApp and Analyze
setting = ZenBioApps.Settings.SpotDetectionSetting("YourBioAppSetting.czccct")
ZenBioApps.SpotDetection.Analyze(img, setting)
```

4. Image Analysis Wizard Analysis
- ```
load setting for Image Analysis Wizard and Analyze
ias = ZenImageAnalysisSetting()
ias.Load("YourImageAnalysisSetting")
Zen.Analyzing.Analyze(image, ias)
```

# Hands on Exercise 4: Image Processing via OAD



1. Load an image / acquire an image
2. Open the macro editor
3. Start the recorder
4. Apply a Gauss filter to the image
5. Stop the recording
6. Execute the macro

```
***** Recorded Code Block *****
image1 =
Zen.Application.Documents.GetByName("C:\...\temp\Experiment-
250.czi")
image2 = Zen.Processing.Filter.Smooth.Gauss(image1, 1.3, 1.3, 1.3,
ZenThirdProcessingDimension.None, True)
Zen.Application.Documents.Add(image2)
***** End of Code Block *****
```

# Hands on Exercise 5:

## Prerequisites: Set up ZEN with simulated hardware



### 1. Set up a simulated hardware

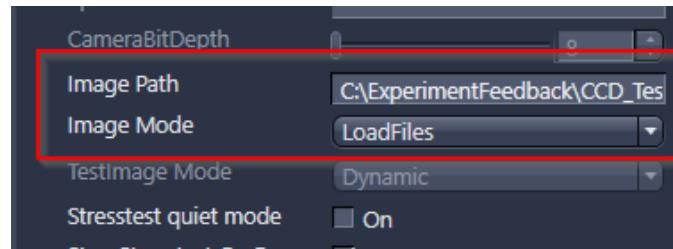
1. Start the MTBKonfig ("C:\Program Files\Carl Zeiss\MTB 2011 - 3.2.1.0\MTB Configuration\MTBConfig.exe") and load the provided MTB *Castor Simulation WF + LSM.mtb*

2. Make sure that the checkbox **simulate all devices** is active. Click **Apply** and **OK**

3. Start ZEN with the **CellDiscoverer** profile

4. In the acquisition tab go to **Acquisition Mode (Show all) – Model Specific**

1. Enter the path to the images you want to load as **Image Path** (place the images from 96well.zip somewhere on the harddrive)
2. For **Image Mode** select **Load Files**



5. Test if it works as expected by snapping an image.

# Hands on Exercise 6: Create a script to run an IA after Image Acquisition



1. Define an experiment
2. Snap an image and setup a suitable Image Analysis
3. Create a macro that performs Image Analysis and saves the result table

```
import
import sys
from System.IO import FileInfo

get active document
img = Zen.Application.ActiveDocument

get name and extract filename
nameandpath = img.FileName
fileInfo = FileInfo(nameandpath)
filename = fileInfo.Name
filepath = fileInfo.DirectoryName
```

```
perform image analysis
analysissetting1 = ZenImageAnalysisSetting()
analysissetting1.Load("YourImageAnalysisSetting")
Zen.Analyzing.Analyze(img, analysissetting1)

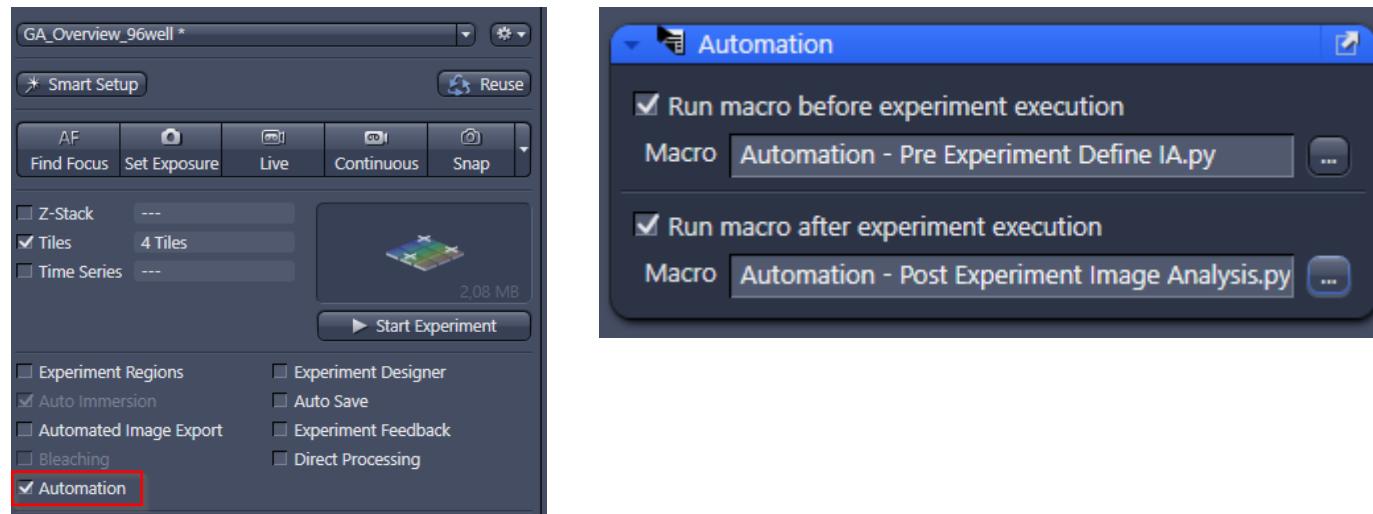
Create Zen table with results for all detected objects
OverviewTable = Zen.Analyzing.CreateRegionsTable(img)

save analyzed image and table
newname = filepath + "\\" + filename[:-4] + "_results.csv"
img.Save()
OverviewTable.Save(newname)
```

# Hands on Exercise 6: Use „automation“ to automatically run an IA after Image Acquisition



1. Use the previously created experiment
2. Activate the checkbox **Automation**
3. Select the previously created Macro to **Run macro after experiment execution**
4. **Start the experiment**



# Optional: Exercise 6 – Extended Version: Use „automation“ to automatically run an IA after Image Acquisition Pre-Acquisition Macro



1. Create a macro that will be executed before each experiment that lists the available IA settings
2. Save the selected IA setting in a TXT file

```
Import the system library
import System.DateTime
activate IO library
from System.IO import File, Directory, FileInfo

get Image Analysis folder
doc = Zen.Application.Environment.GetFolderPath(ZenSpecialFolder.Documents)
IAPath = doc + '\\\\' + 'Carl Zeiss\\ZEN\\Documents\\\\Image Analysis Settings'
fileNames = []
IAtype = '*.czias'
files = Directory.GetFiles(IAPath,IAtype)
for i in range(0,files.Length):
 file = files[i]
 fileInfo = FileInfo(file)
 IAOrgFileNameWE = fileInfo.Name.Substring(0,fileInfo.Name.Length-6)
 fileNames.append(IAOrgFileNameWE)
```

```
create setup dialog
window = ZenWindow()
window.AddDropDown('IAval','Select Image Analysis',', fileNames, 0)

show dialog and get results from dialog
result=window.Show()

check if cancelled was selected
if result.HasCanceled == True:
 sys.exit('Macro aborted with Cancel!')

write selected IA setting to temp file (IA_selected.txt) in C:\\Temp folder
IA = result.GetValue('IAval')
fname = 'c:\\\\temp\\\\IA_selected.txt'
f = open(fname, 'w')
f.write(IA)
f.close()
```

# Optional: Exercise 6 – Extended Version: Use „automation“ to automatically run an IA after Image Acquisition Post-Acquisition Macro

1. Create a macro that will be executed after each experiment that reads the name of the IA setting from the TXT file
2. Execute the selected Image Analysis
3. Create a table with results and show in ZEN

```
Import the system library
import System.DateTime
##
activate IO library
from System.IO import File, Directory, FileInfo
```

```
image = Zen.Application.ActiveDocument
```

```
import IA from previously created text file
fname = 'c:\\temp\\IA_selected.txt'
rFile = open(fname,'r')
IA = rFile.read()
```

```
print IA
#rFile.Close()

perform image analysis
analysissetting1 = Zen.ImageAnalysisSetting()
analysissetting1.Load(IA)
Zen.Analyzing.Analyze(image, analysissetting1)

Create Zen table with results for all detected objects (parent class)
OverviewTable = Zen.Analyzing.CreateRegionsTable(image)
Zen.Application.Documents.Add(OverviewTable)
```



# Exercice 7: Start a Fiji macro for Processing



```
from System.Diagnostics import Process

define image
img = Zen.Application.ActiveDocument

get name and extract filename
nameandpath = img.FileName

IMAGEJ = "C:\\Fiji.app\\ImageJ-win64.exe"
IMAGEJDIR = Path.GetDirectoryName(IMAGEJ)

start Fiji script to perform processing
app = Process()
app.StartInfo.FileName = IMAGEJ
macro = r'-macro C:\\Training\\Kuwahara.ijm'
app.StartInfo.Arguments = macro + ' ' + nameandpath
app.Start()

wait until the script is finished
app.WaitForExit()
excode = app.ExitCode

filename as defined in Fiji macro:
processedname = nameandpath[:-4] + "_result.tif"

load result in Zen
if File.Exists(processedname):
 processedimage =
 Zen.Application.LoadImage(processedname, False)
 Zen.Application.Documents.Add(processedimage)

else:
 print 'Saved figure not found.'
```

# Exercice 7: Start a Fiji macro for Processing (Fiji macro)



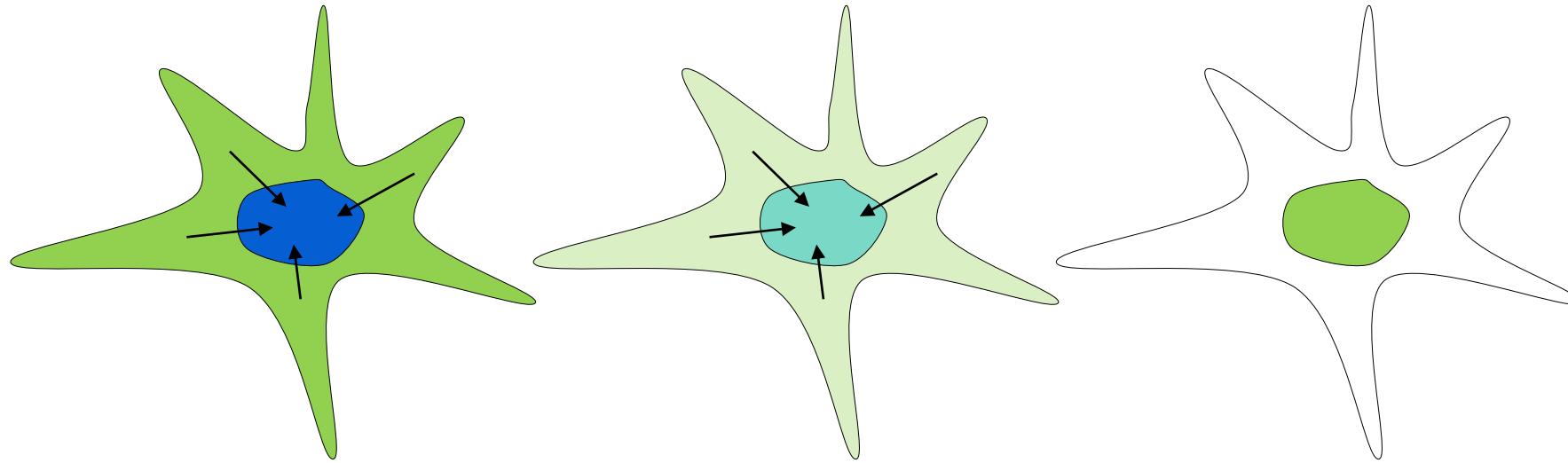
```
// read the filename that is passed from ZEN
name= getArguments();
// error message if there was no filename
if (name=="") exit("No file selected!");

// Import the CZI file with Bio-formats
run("Bio-Formats Importer", "open=[" + name + "] autoscale
 color_mode=Default open_all_series rois_import=[ROI manager]
 view=Hyperstack stack_order=XYCZT");

// Run Processing
run("Kuwahara Filter", "sampling=5");

// save result
originalNameWithoutExt = replace(name , ".czi" , "");
resultName = originalNameWithoutExt + "_result.tif";
saveAs("Tiff", resultName);
close();
```

# Application Example: Translocation



Cytoplasm-to-Nucleus Translocation

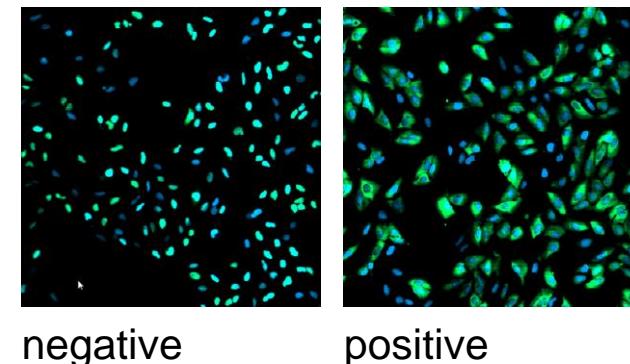
# Human U2OS cells cytoplasm–nucleus translocation



- 96-well plate, human osteosarcoma cells (U2OS), nuclei stained with DRAQ
- Cytoplasm to nucleus translocation of the Forkhead (FKHR-EGFP) fusion protein
- In proliferating cells, FKHR is localized in the cytoplasm (constantly moving into the nucleus, but is transported out again by export proteins).
- Upon inhibition of nuclear export, FKHR accumulates in the nucleus
- Export is inhibited by blocking PI3 kinase / PKB with **Wortmannin** or **LY294002**.

|   | 96-well plate |       |       |      |      |      |       |       |      |     |     |           |
|---|---------------|-------|-------|------|------|------|-------|-------|------|-----|-----|-----------|
|   | 1             | 2     | 3     | 4    | 5    | 6    | 7     | 8     | 9    | 10  | 11  | 12        |
| A | Neg. Ctrl     | 39688 | 0.977 | 1.95 | 3.91 | 7.81 | 15.63 | 31.25 | 62.5 | 125 | 250 | Pos. Ctrl |
| B | Neg. Ctrl     | empty | 0.977 | 1.95 | 3.91 | 7.81 | 15.63 | 31.25 | 62.5 | 125 | 250 | Pos. Ctrl |
| C | Neg. Ctrl     | empty | 0.977 | 1.95 | 3.91 | 7.81 | 15.63 | 31.25 | 62.5 | 125 | 250 | Pos. Ctrl |
| D | Neg. Ctrl     | empty | 0.977 | 1.95 | 3.91 | 7.81 | 15.63 | 31.25 | 62.5 | 125 | 250 | Pos. Ctrl |
| E | Pos. Ctrl     | empty | 0.31  | 0.63 | 1.25 | 2.5  | 5     | 10    | 20   | 40  | 80  | Neg. Ctrl |
| F | Pos. Ctrl     | empty | 0.31  | 0.63 | 1.25 | 2.5  | 5     | 10    | 20   | 40  | 80  | Neg. Ctrl |
| G | Pos. Ctrl     | empty | 0.31  | 0.63 | 1.25 | 2.5  | 5     | 10    | 20   | 40  | 80  | Neg. Ctrl |
| H | Pos. Ctrl     | empty | 0.31  | 0.63 | 1.25 | 2.5  | 5     | 10    | 20   | 40  | 80  | Neg. Ctrl |

Wortmannin in nM      LY294.002 in  $\mu$ M



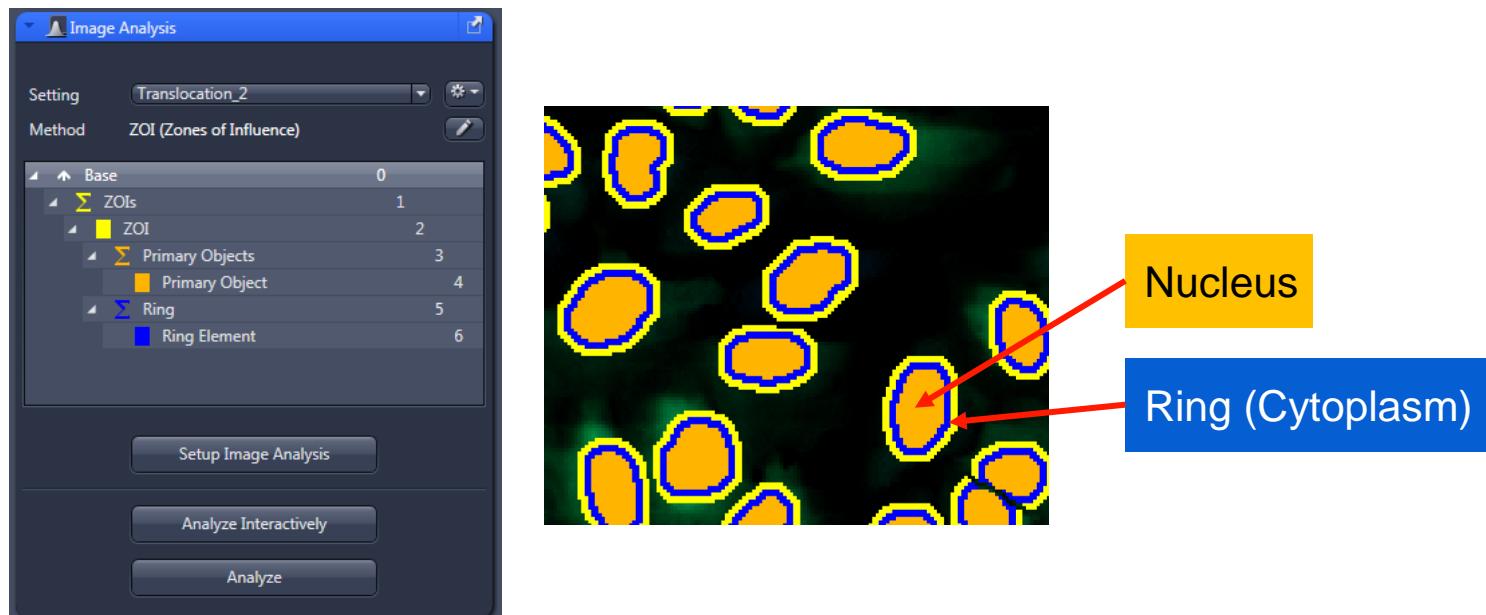
Data set [BBBC013v1](#) by Ilya Ravkin, available from the Broad Bioimage Benchmark Collection [[Ljosa et al., Nature Methods, 2012](#)]

# Human U2OS cells cytoplasm–nucleus translocation



## ZOI – Segmentation:

- Segmentation based on the DRAQ-channel (nuclei)
- The Ring is used to measure the EGFP signal in the cytoplasm



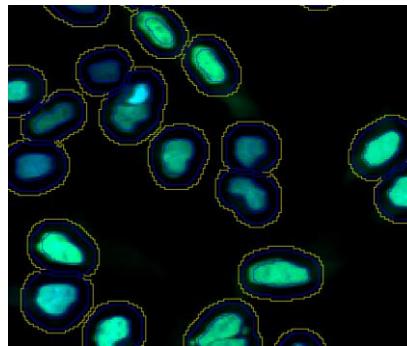
Data set [BBBC013v1](#) by Ilya Ravkin, available from the Broad Bioimage Benchmark Collection [[Ljosa et al., Nature Methods, 2012](#)]

# Human U2OS cells cytoplasm–nucleus translocation

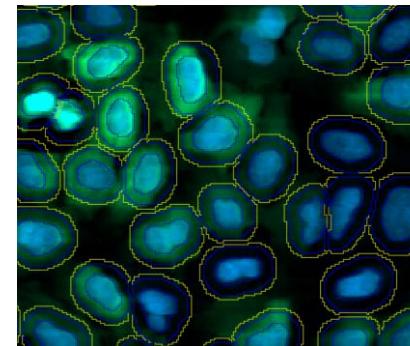


ZOI – Segmentation:

- Segmentation based on the DRAQ-channel (nuclei)
- The Ring is used to measure the EGFP signal in the cytoplasm
- Determine measurement features for nuclei and ring around the nuclei:
  - Nucleus Mean Intensity EGFP
  - Ring Mean Intensity EGFP



negative



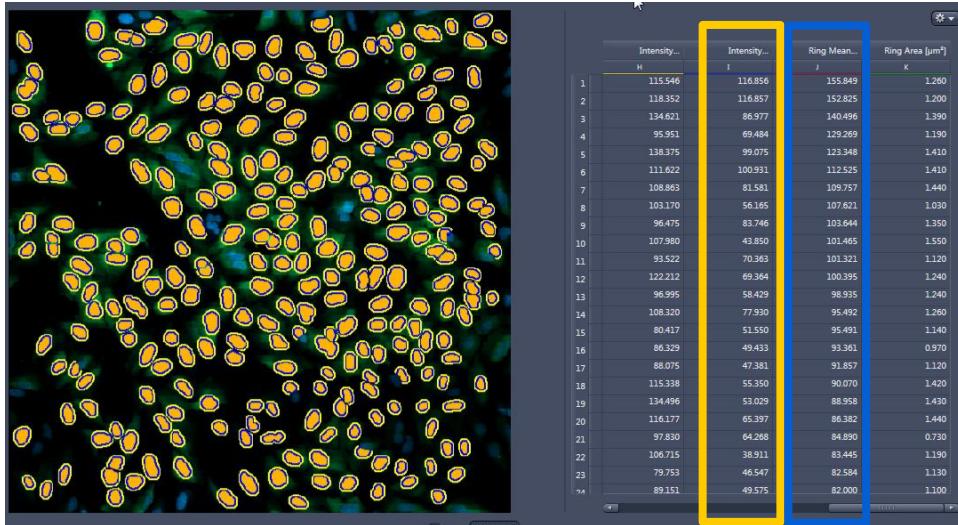
positive

Data set [BBBC013v1](#) by Ilya Ravkin, available from the Broad Bioimage Benchmark Collection [[Ljosa et al., Nature Methods, 2012](#)]

# Human U2OS cells cytoplasm–nucleus translocation



- Calculate the Translocation-Ratio (T) for each cell
- Calculate the mean value for each well  
(e.g. via OAD and Python)



$$T = \frac{I_{\text{Mean, Nucleus}}}{I_{\text{Mean, Ring}}}$$

Data set [BBBC013v1](#) by Ilya Ravkin, available from the Broad Bioimage Benchmark Collection [[Ljosa et al., Nature Methods, 2012](#)]

## Tasks to perform:



1. Load load the image file (\*.csv) and image analysis setting (\*.czias)
2. Run the image analysis
3. Extract the image analysis results as \*.csv
4. Start the python script (test\_wellplate\_from\_ZEN.PY)
  
5. Read data from \*.csv file
6. Calculate the translocation Ratio
7. Generate heatmaps for different features (e.g. Translocation Ratio)
8. Save heatmaps as PNG files
  
9. Load and display PNG files in ZEN



Images and Scripts can be found in [Translocation\\_Demo\\_Zeiss.zipx](#)

```
from System.Diagnostics import Process
from System.IO import File, Path, Directory
import time

define the external plot script or tool
pythonexe = r'C:\Anaconda3\python.exe'
script = r'C:\...\test_wellplate_from_ZEN.PY'

load image and add it to ZEN
image_to_analyze = r'C:\...\Translocation_comb_96_5ms.czi'
image = Zen.Application.LoadImage(image_to_analyze)
Zen.Application.Documents.Add(image)

get the image path
outputpath = Path.GetDirectoryName(image_to_analyze)
resultname = Path.GetFileNameWithoutExtension(image.Name)
```

[https://github.com/zeiss-microscopy/OAD/tree/master/Scripts/Image\\_Analysis](https://github.com/zeiss-microscopy/OAD/tree/master/Scripts/Image_Analysis)

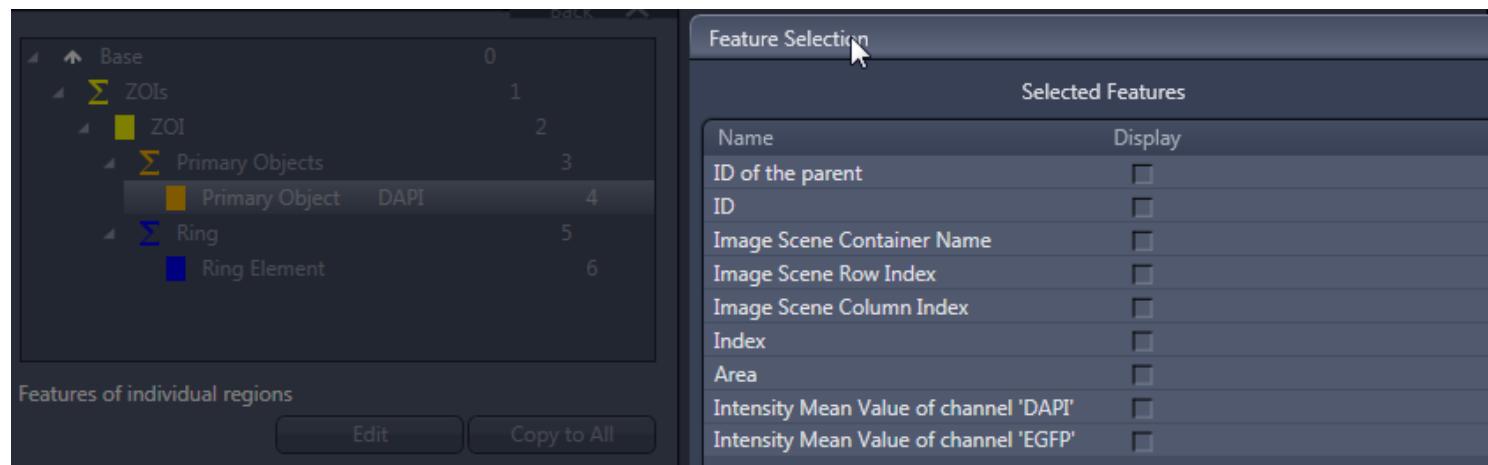
# OAD Script: Run Image Analysis



Features defined in the image analysis setting:

ID parent | ID | Image Scene container Name | Image Scene Row | Image Scene Column | Index # | Area | NucMeanDapi | NucMeanGFP | RingMeanGFP | RingArea

```
Load image analysis setting and perform image analysis
iasfilename = r'C:\...\Image Analysis Settings\Translocation_2.czias'
ias = ZenImageAnalysisSetting()
ias.Load(iasfilename)
Zen.Analyzing.Analyze(image,ias)
```



# OAD Script: Write CSV File



```
For ZOI-Image Analysis Settings need to get the results for the Primary Objects
Create data list with results for each primary object
table_single = Zen.Analyzing.CreateRegionTable(image, "Primary Object")
#Zen.Application.Documents.Add(table_single)

Save both data list as CSV files
table_single_filename = Path.Combine(outputpath, resultname + '_Single.csv')
table_single.Save(table_single_filename)

close the image and image analysis setting
image.Close()
ias.Close()
```

| A  | B            | C         | D          | E          | F          | G                          | H           | I           | J           | K                          | L    |
|----|--------------|-----------|------------|------------|------------|----------------------------|-------------|-------------|-------------|----------------------------|------|
| 1  | ParentID::ID | ID::ID!!I | ImageScene | ImageScene | ImageScene | Index::Index Area::Area!!I | IntensityMe | IntensityMe | CopyRingInt | CopyRingArea::Ring Area!!R |      |
| 2  |              |           |            |            |            | µm <sup>2</sup>            | Gray        | Gray        | Gray        | µm <sup>2</sup>            |      |
| 3  | 269          | 270 A1    |            | 1          | 1          | 1                          | 2.64        | 133.337121  | 88.0530303  | 141.167832                 | 1.43 |
| 4  | 271          | 272 A1    |            | 1          | 1          | 2                          | 1.84        | 79.2880435  | 0           | 0.02962963                 | 1.35 |
| 5  | 273          | 274 A1    |            | 1          | 1          | 3                          | 3.04        | 168.177632  | 9.81907895  | 9.22058824                 | 1.36 |
| 6  | 275          | 276 A1    |            | 1          | 1          | 4                          | 2.81        | 106.231317  | 38.7935943  | 46.6071429                 | 1.96 |
| 7  | 277          | 278 A1    |            | 1          | 1          | 5                          | 2.47        | 146.546559  | 52.7773279  | 65.0347222                 | 1.44 |
| 8  | 279          | 280 A1    |            | 1          | 1          | 6                          | 2.01        | 96.6865672  | 33.4079602  | 68.3777778                 | 1.35 |
| 9  | 281          | 282 A1    |            | 1          | 1          | 7                          | 1.58        | 94.3734177  | 18.7468354  | 28.1842105                 | 1.14 |
| 10 | 283          | 284 A1    |            | 1          | 1          | 8                          | 1.81        | 87.6574586  | 31.0607735  | 53.5683453                 | 1.39 |
| 11 | 285          | 286 A1    |            | 1          | 1          | 9                          | 2.25        | 114.826667  | 0           | 0                          | 1.36 |
| 12 | 287          | 288 A1    |            | 1          | 1          | 10                         | 2.72        | 95.4669118  | 0.11397059  | 0.625                      | 1.44 |
| 13 | 289          | 290 A1    |            | 1          | 1          | 11                         | 1.95        | 96.4564103  | 13.2974359  | 22.0735294                 | 1.36 |
| 14 | 291          | 292 A1    |            | 1          | 1          | 12                         | 2.23        | 104.753363  | 0.07623318  | 0.10218978                 | 1.37 |

# OAD Script: Start Python with Parameters



```
define the actual CSV file and the parameters
 csvfile = Path.Combine(outputpath, table_single_filename)

this depends on the actual CZIAS and the import of the CSV table in python
 #parameter2display = 'CellCount'
 parameter2display = 'Ratio'
 params = ' -f ' + csvfile + ' -w 96' + ' -p ' + parameter2display + ' -sp False -dpi 100 -xlsx True'

start the data display script as an external application
Process.Start(script, params)
```

```
C:\Anaconda3\python.exe
CSU Filename: C:\testdata\Broadinstitute\Translocation_com_96_5ms_Single.csv
PlateType: 96
Parameter to display: Ratio
DPI: 100
Columns : Index<I'ParentID::ID of the parent!!I', 'ID::ID!!I',
 'ImageSceneContainerName::Image Scene Container Name',
 'ImageSceneRow::Image Scene Row Index!!I',
 'ImageSceneColumn::Image Scene Column Index!!I', 'Index::Index!!I',
 'Area::Area!!R',
 'IntensityMean_DAPI::Intensity Mean Value of channel 'DAPI'!!R',
 'IntensityMean_EGFP::Intensity Mean Value of channel 'EGFP'!!R',
 'CopyRingIntensityMean1::Ring Mean Intensity 1!!R',
 'CopyRingArea::Ring Area!!R'],
 dtype='object')
Number of Object Parameters: 6
wellID_key : WellID
Found keys:
Index<I'ParentID', 'ID', 'WellID', 'RowID', 'ColumnID', 'Index', 'Area',
 'NucMeanDapi', 'NucMeanGFP', 'RingMeanGFP', 'RingArea', 'Ratio'],
 dtype='object')
FloatProgress(value=1.0, description='Processing Wells', max=96.0, min=1.0)
```

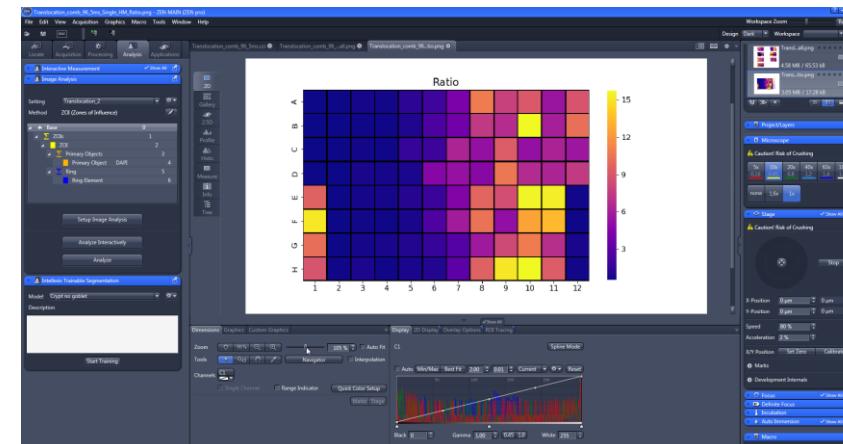
# OAD Script: Load PNG Files



```
define filenames of PNG files
savename_all = Path.Combine(Path.GetDirectoryName(image_to_analyze), Path.GetFileNameWithoutExtension(image_to_analyze) +
'_Single_HM_all.png')
savename_single = Path.Combine(Path.GetDirectoryName(image_to_analyze), Path.GetFileNameWithoutExtension(image_to_analyze) +
'_Single_HM_' + parameter2display + '.png')
print 'Showing saved figure in ZEN.'

Check if fileneame exists, and Load images in ZEN
if File.Exists(savename_all):
 plotfigure1 = Zen.Application.LoadImage(savename_all, False)
 plotfigure2 = Zen.Application.LoadImage(savename_single, False)
 Zen.Application.Documents.Add(plotfigure1)
 Zen.Application.Documents.Add(plotfigure2)
else:
 print 'Saved figure not found.'

print 'Done.'
```



# Results (heatmap)



Extract relevant features and plot them as a heatmap for the 96 well plate  
(e.g. via OAD and Python)

