PoL Physics of Life TU Dresden

CENTER FOR SYSTEMS BIOLOGY DRESDEN

Bio-Image Analysis, Programming, Biostatistics and Machine Learning for Computational Biology
A Summary

Robert Haase

April-July 2021

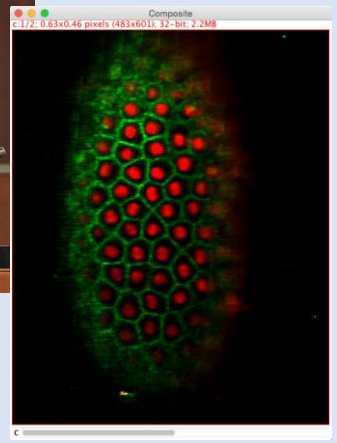
@haesleinhuepf

Image analysis is part of the experiment

Observation

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{j=1}^{n} [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}$$

Modeling



Imaging

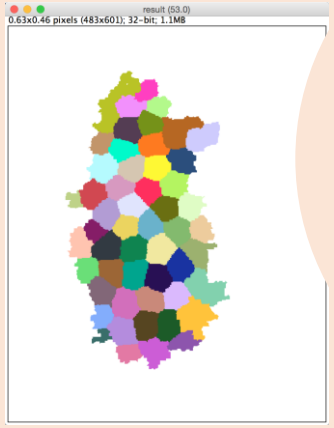
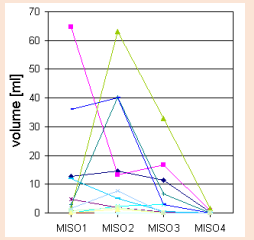


Image processing



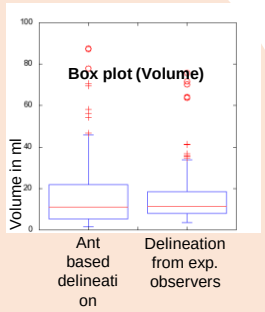
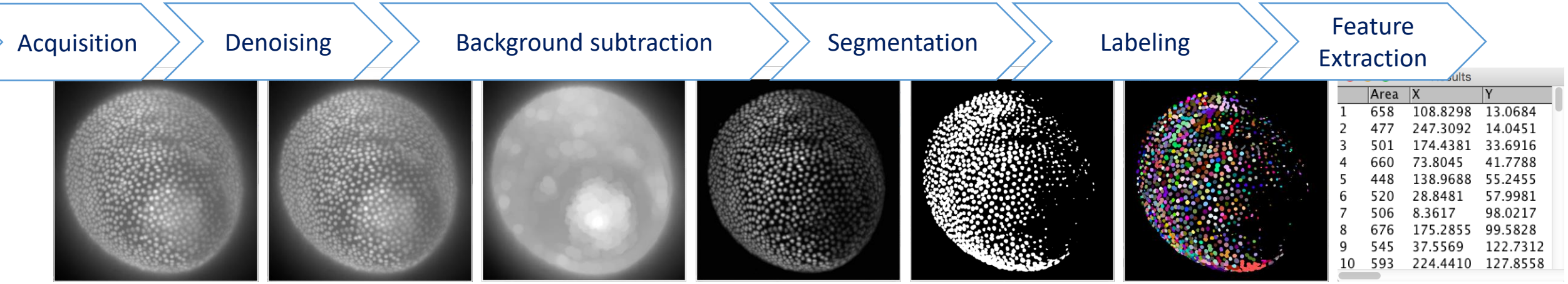


Image analysis
Bio-statistics

Image sources: Akanksha Jain, Tomancak lab, MPI CBG (SPIM image)
https://commons.wikimedia.org/wiki/User:Acagastya (Miroscope, public domain)
Muhammad Mahdi Karim (Drosophila, GFDL 1.2)

Image analysis is part of the experiment

- Going back allows us to improve
- We’re not “failing”



	Area	X	Y
1	658	108.8298	13.0684
2	477	247.3092	14.0451
3	501	174.4381	33.6916
4	660	73.8045	41.7788
5	448	138.9688	55.2455
6	520	28.8481	57.9981
7	506	8.3617	98.0217
8	676	175.2855	99.5828
9	545	37.5569	122.7312
10	593	224.4410	127.8558

Image data source: Daniela Vorkel, Myers lab, MPI CBG

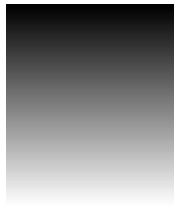
Introduction to bio-image analysis

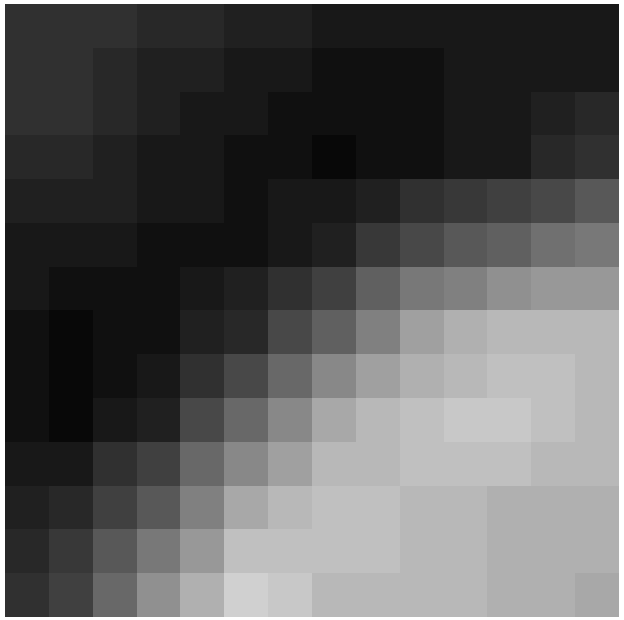
PoL Physics of Life TU Dresden

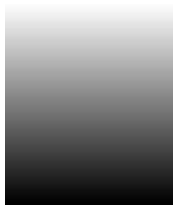
- Bio-image analysis is supposed to be
  - **Quantitative**
    - We derive numbers from images which describe physical properties of the observed sample.
  - **Objective**
    - The derived measurement does not depend on who did the measurement. The measurement is free of interpretation.
  - **Reliable / valid / trustworthy**
    - We are confident that the measurement is describing what it is supposed to describe.
  - **Reproducible**
    - Somebody else can do the experiment under *different conditions* and gets similar measurements. For this, documentation is decisive!
  - **Repeatable**
    - We can do the same experiment twice under the *same conditions* and get a similar measurements.

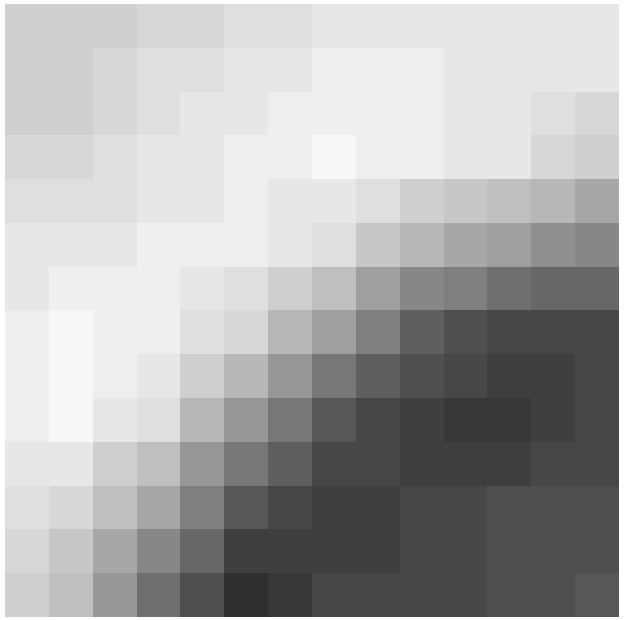
Lookup tables

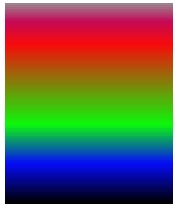
- The lookup table decides how the image is displayed on screen.
- Applying a different lookup table doesn't change the image. All pixel values stay the same, they just appear differently

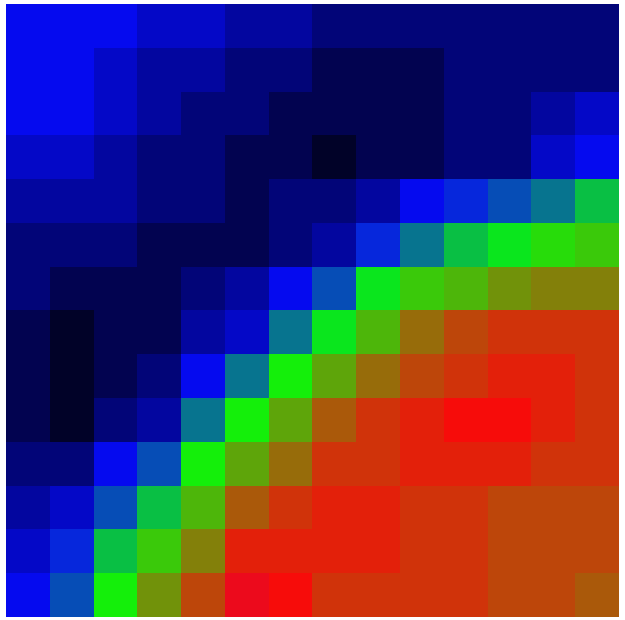
Pixel value	Display color
0 1 2 … 255	



Pixel value	Display color
0 1 2 … 255	

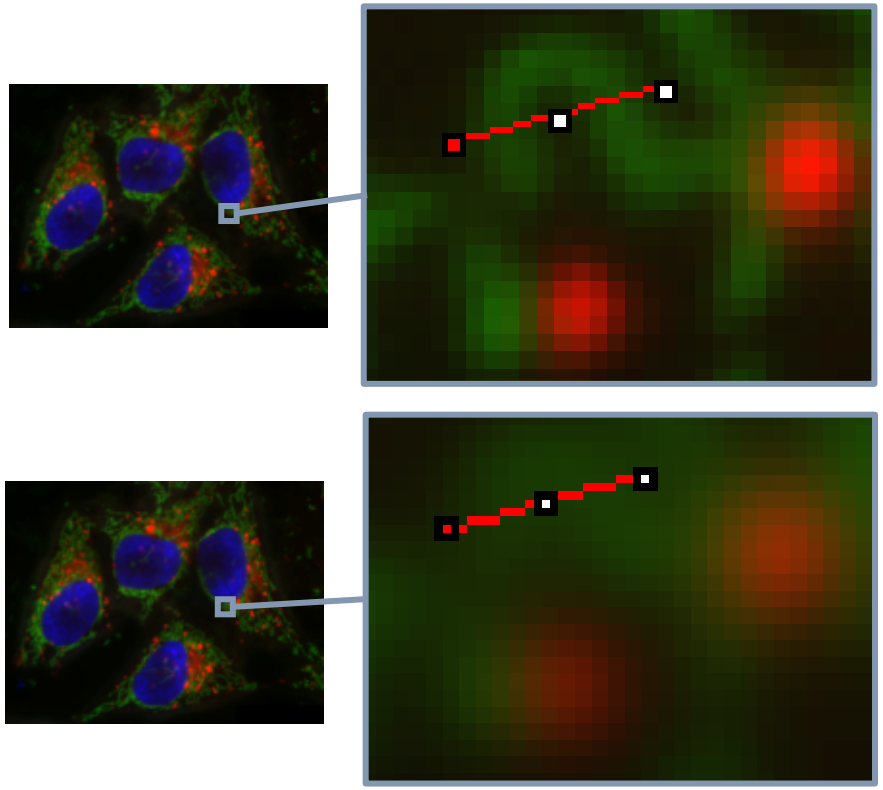


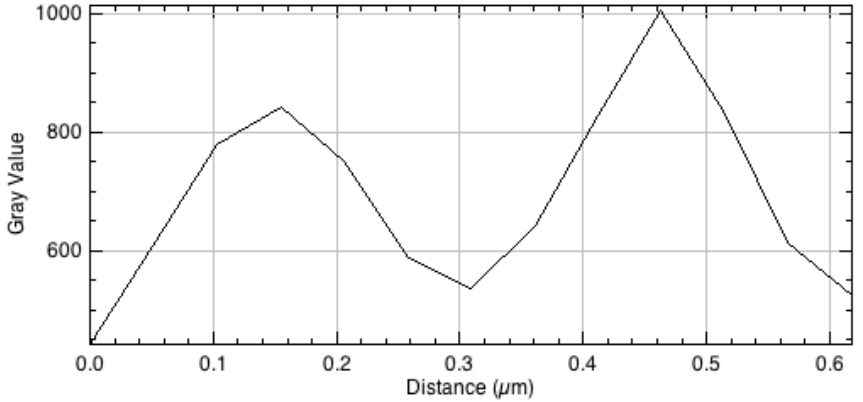
Pixel value	Display color
0 1 2 … 255	

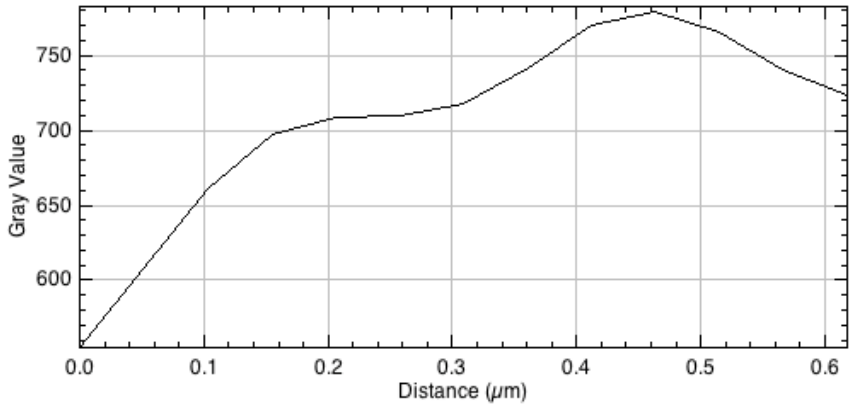


Pixel size versus resolution

- Resolution is a property of your imaging system.
- The measure of how close object can be in an image while still being differentiable, is called spatial resolution.







PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Filters don’t do magic

Robert Haase

July 2021

@haesleinhuepf

Filters

- An image processing filter is an operation on an image.
- It takes an image and produces a new image out of it.
- Filters change pixel values.

- There is no “best” filter. Which filter fits your needs, depends on the context.
- Filters do not do magic. They can not make things visible which are not in the image.

- Application examples
  - Noise-reduction
  - Background removal
  - Artefact-removal
  - Contrast enhancement
  - Correct uneven illumination

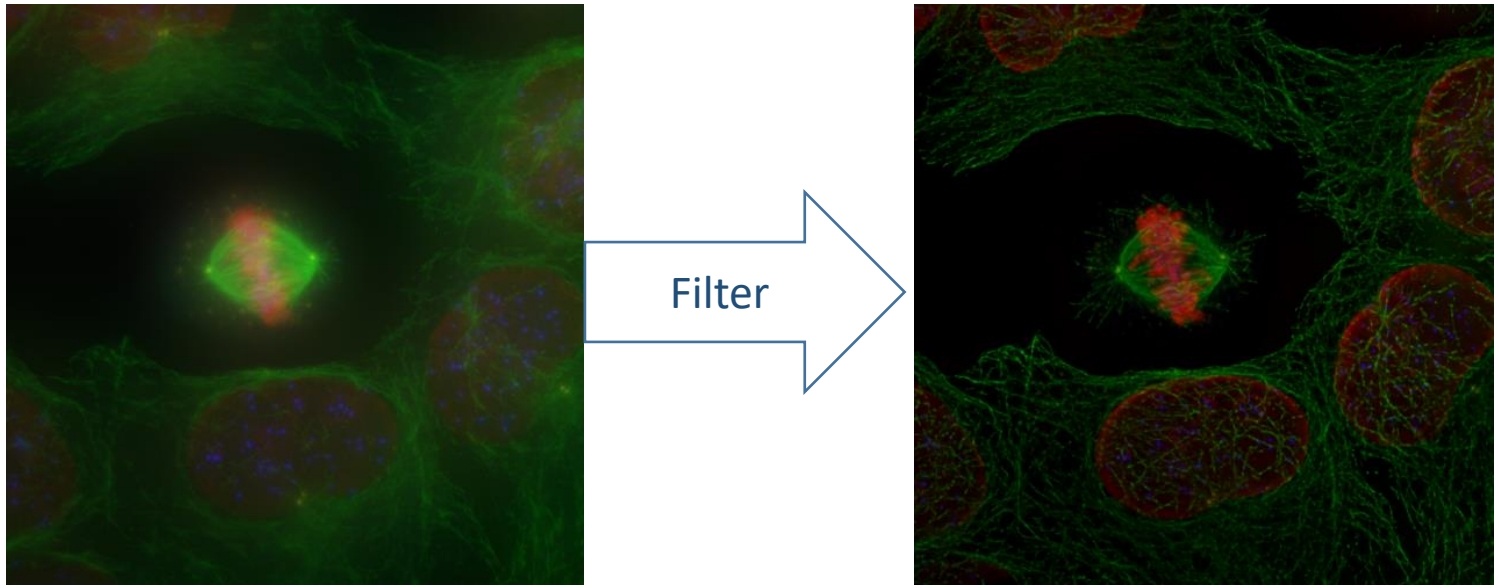
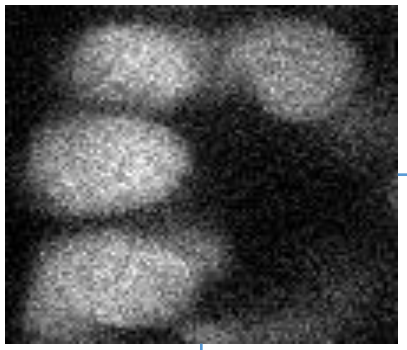


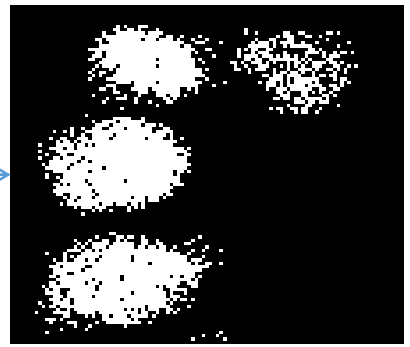
Image source: Alex Bird / Dan White MPI CBG

Why filters are used

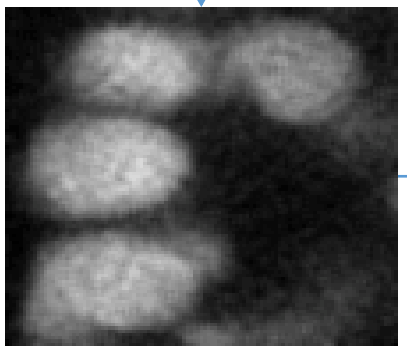
- We need to remove the noise to help the computer *interpreting* the image



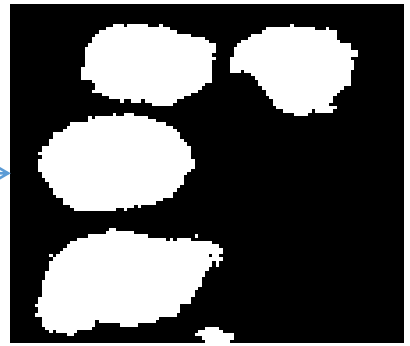
Thresholding



Blurring



Thresholding

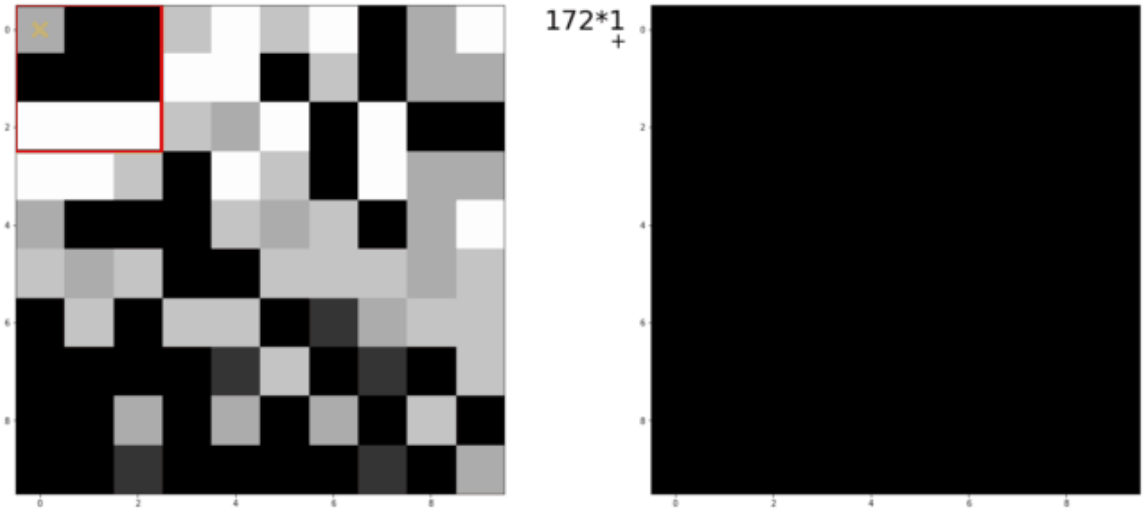


Oh no! I see thousands of tiny white objects!

Ok, it’s just 4 objects.

Linear Filters: Convolution

- Linear filters replace each pixel value with a linear combination of surrounding pixels



- Kernels are matrices describing a linear filter

- Convolution: Applying a kernel matrix to an image by pixel-wise multiplication and summation to retrieve new pixel values.

Mean filter, 3x3 kernel

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

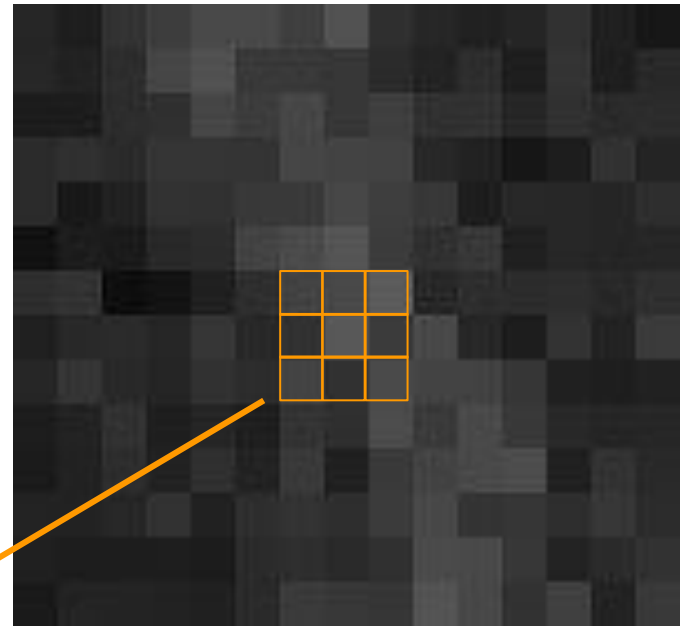
Question 20. How do linear filters work and what is a convolution?

Animation source: Dominic Waithe, Oxford University
https://github.com/dwaithe/generalMacros/tree/master/convolution_ani

Nonlinear Filters

- Non linear filters also replace pixel value inside as rolling window but in a non linear function.

- Examples: order statistics filters
  - Min
  - Median
  - Max
  - Variance
  - Standard deviation



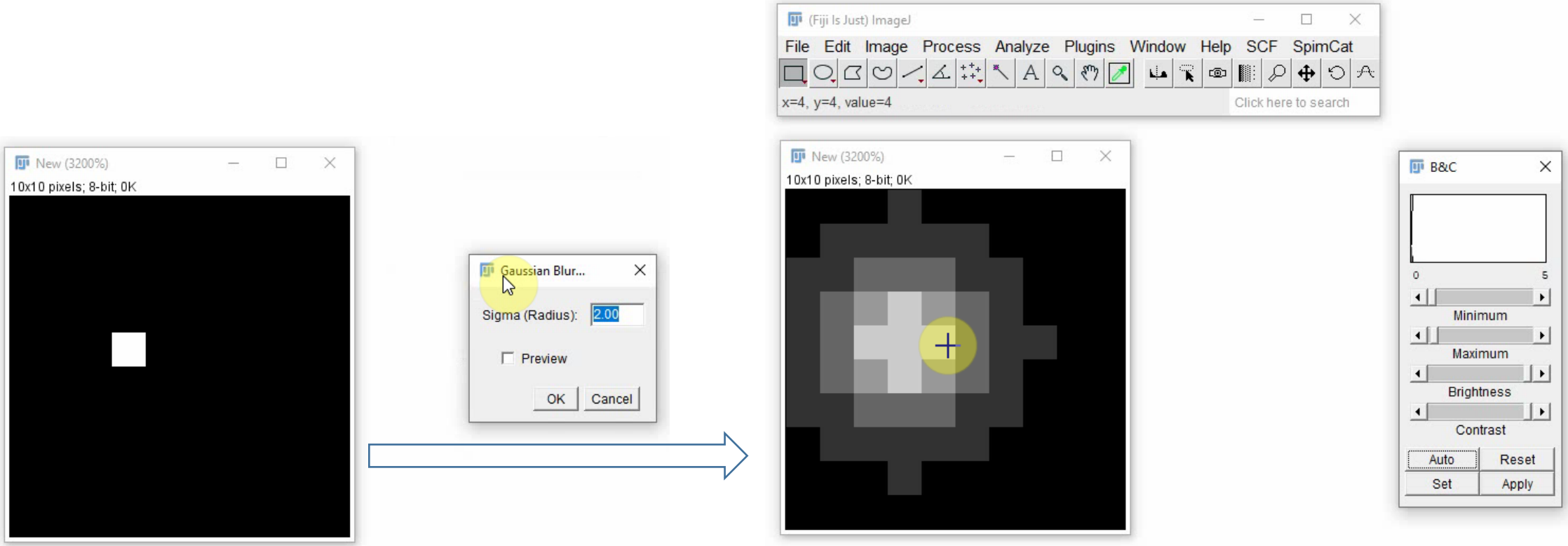
75	85	60
67	73	91
50	88	59

50 59 60 67 73 75 85 88 91

Min Median Max

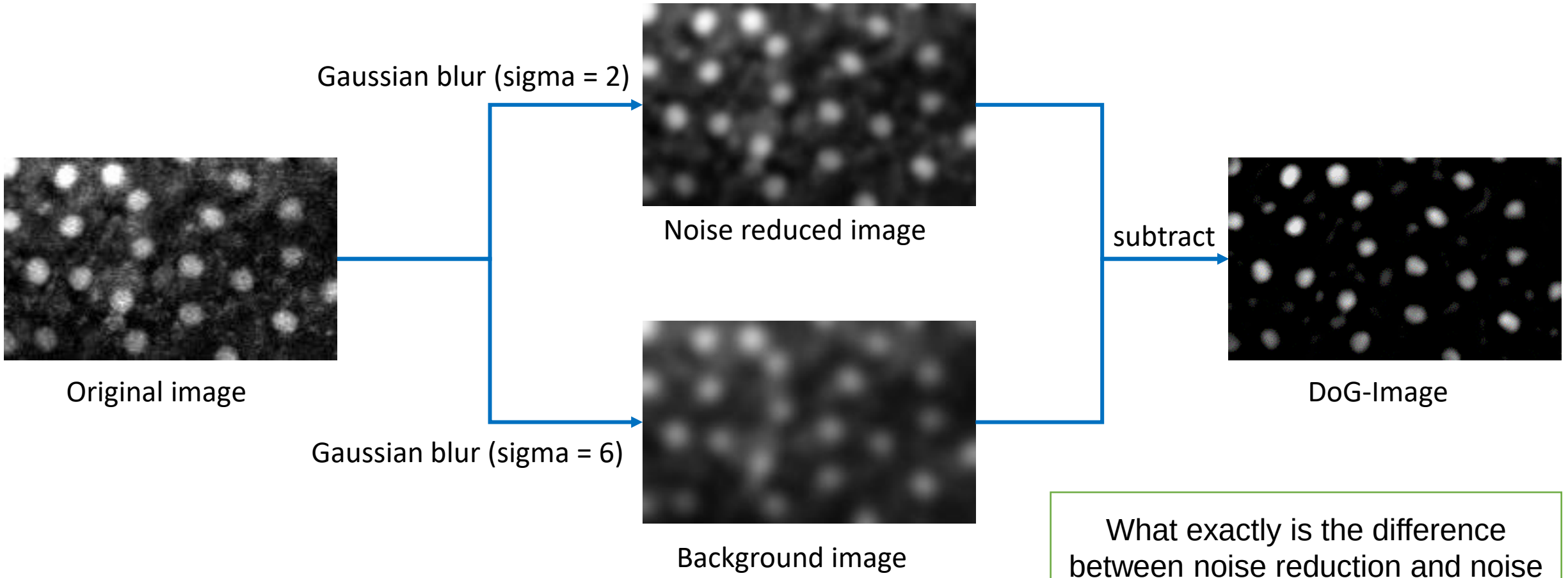
Determining what filters do

- Use images with single pixels > 0 and apply filters to understand what the filters do



Difference-of-Gaussian (DoG)

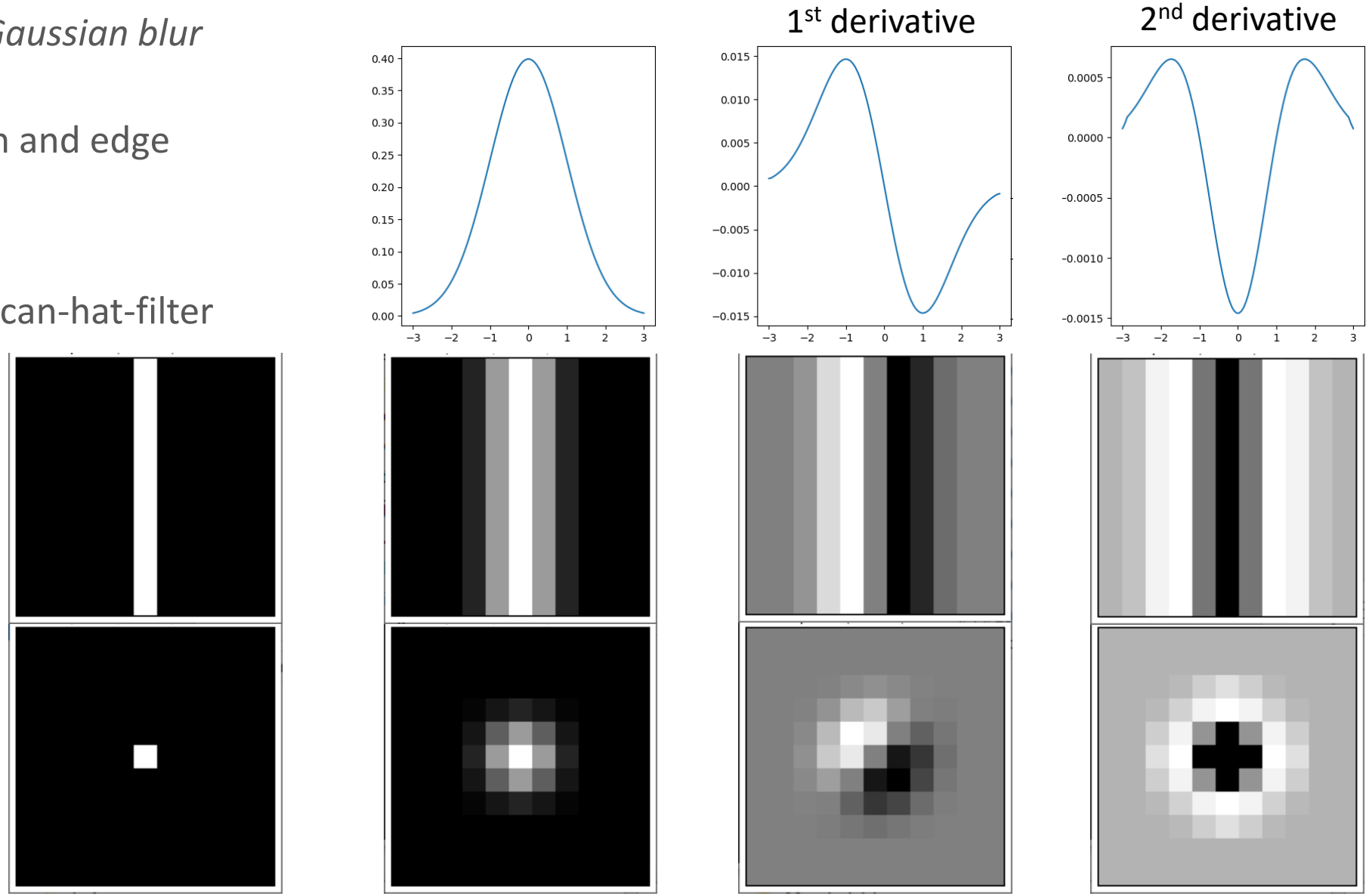
- Improve image in order to detect bright objects.



What exactly is the difference between noise reduction and noise removal (and background subtraction)?

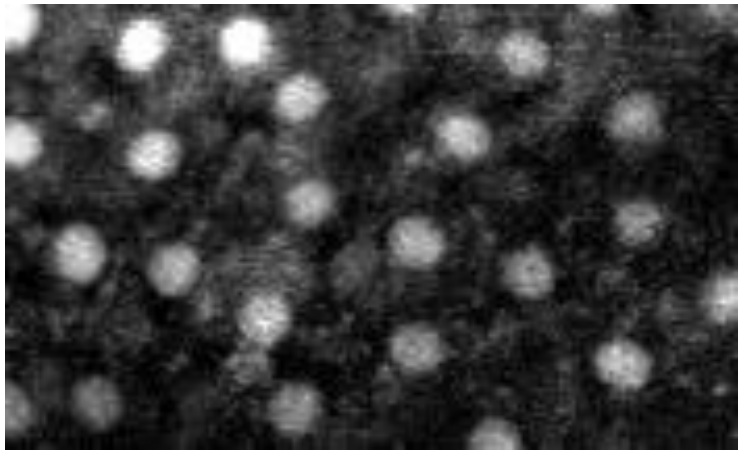
Laplace-filter

- *Second derivative of a Gaussian blur filter*
- Used for edge-detection and edge enhancement
- Also known as the Mexican-hat-filter

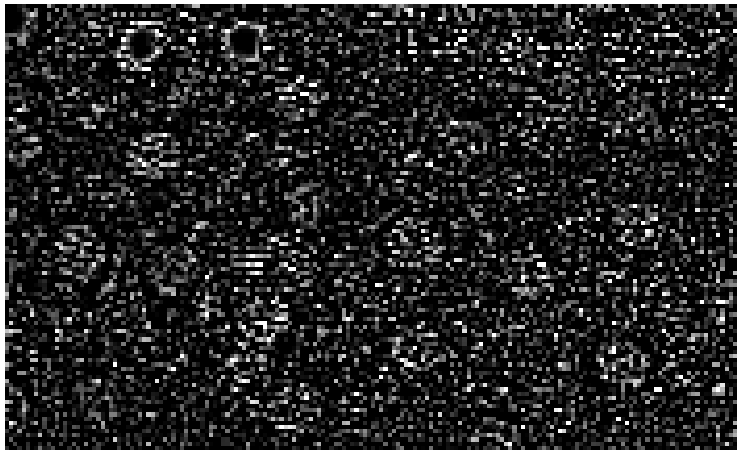


Laplacian-of-Gaussian (LoG)

Laplace filter



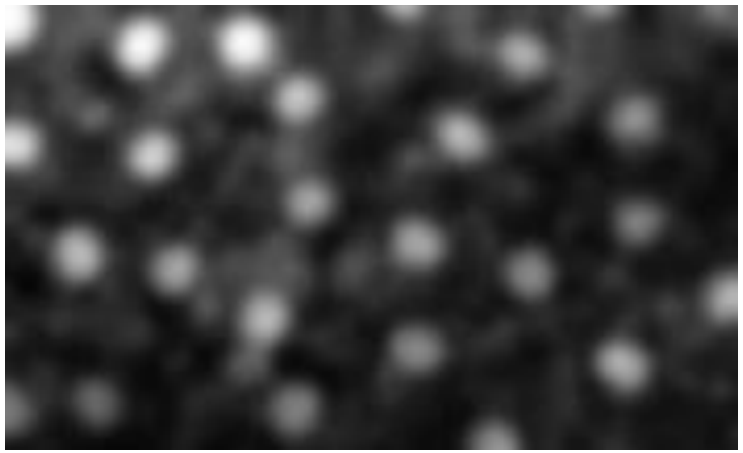
$$*\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$



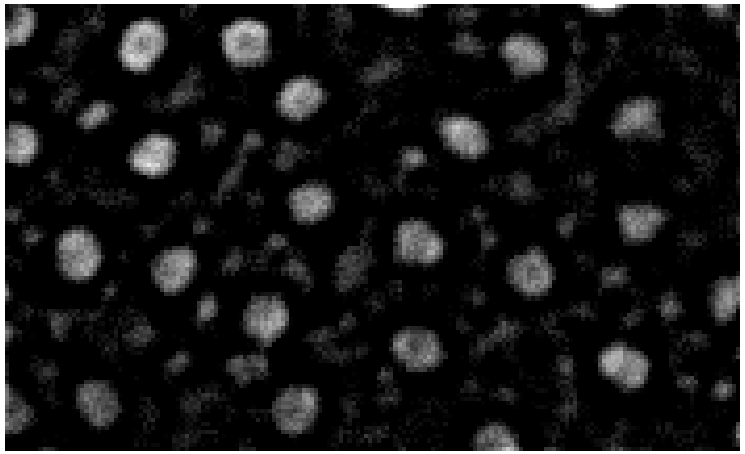
Laplace filtered image

Gaussian blur

Laplacian of Gaussian filter



$$*\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$



LoG image

PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Image Segmentation

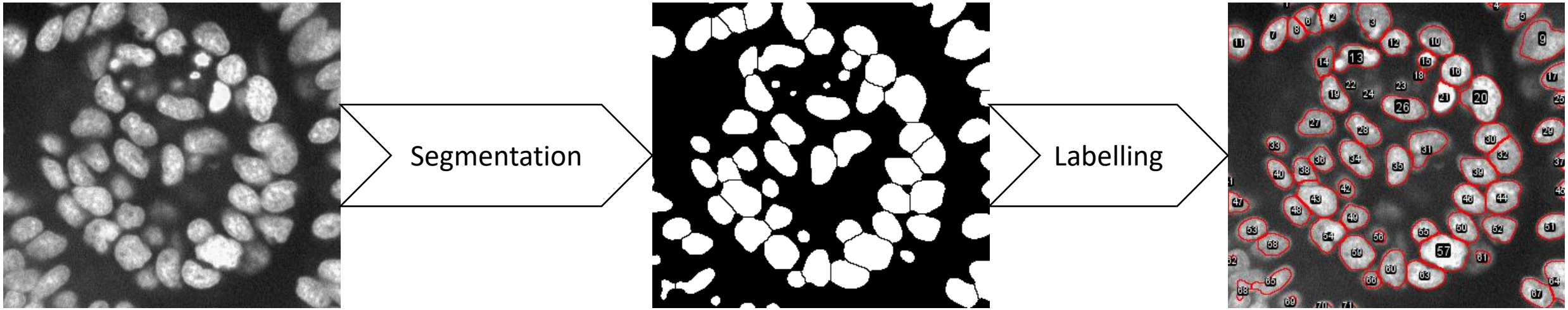
Robert Haase

July 2021

@haesleinhuepf

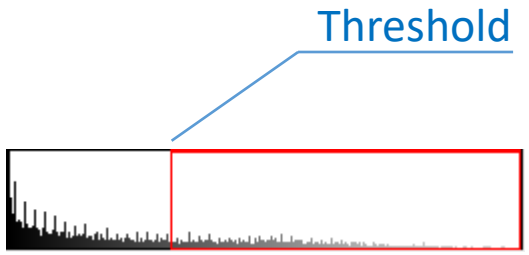
Segmentation and labelling

- In order to allow the computer analyzing individual objects in images, we need to segment and label them first.
- Methods
  - Thresholding
  - Spot detection + seeded watershed
  - Edge detection based
  - Machine learning

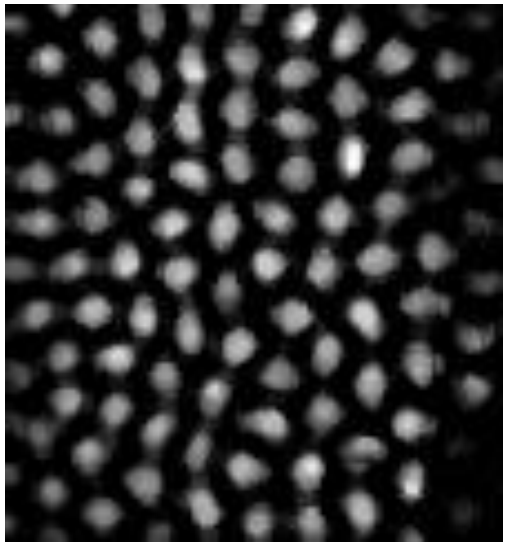


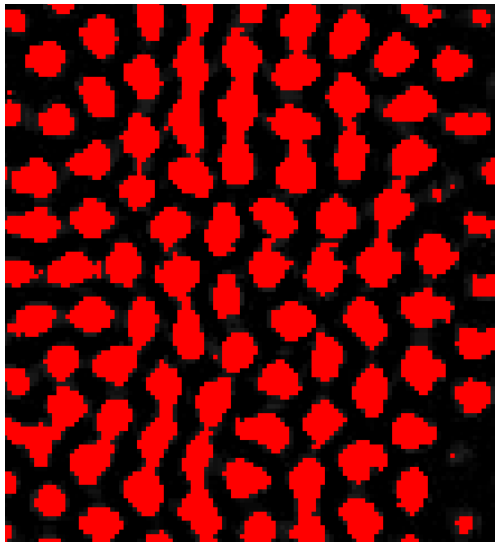
Thresholding

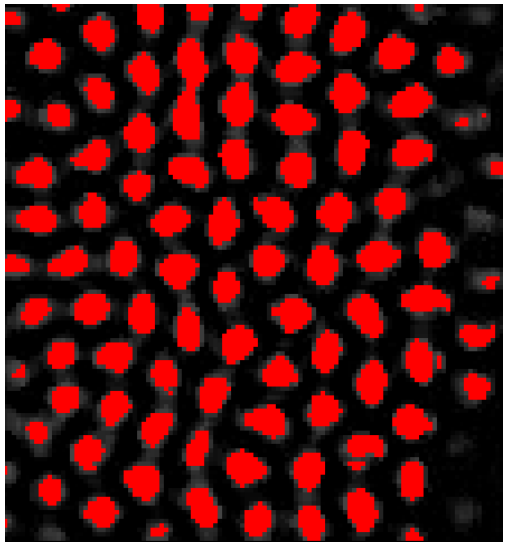
- Thresholding is the process of separating background from regions of interest by setting a minimum and/or maximum signal intensity as *threshold*.

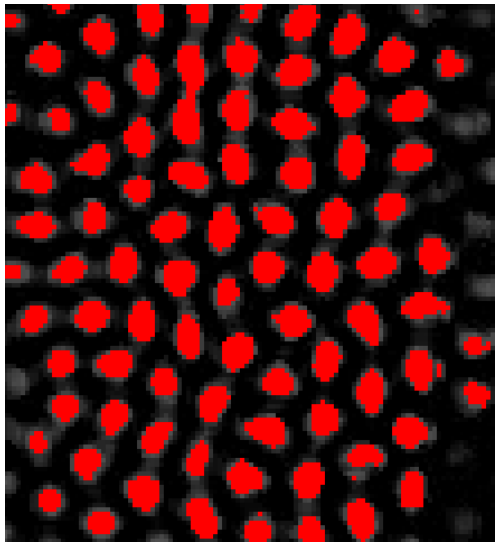


- When thresholding in a scientific context
  - Apply the same algorithm to all images in your project -> objectivity
  - Check carefully with many images when choosing the algorithm -> reliability
  - Document and cite which algorithm you used -> reproducibility


Original image

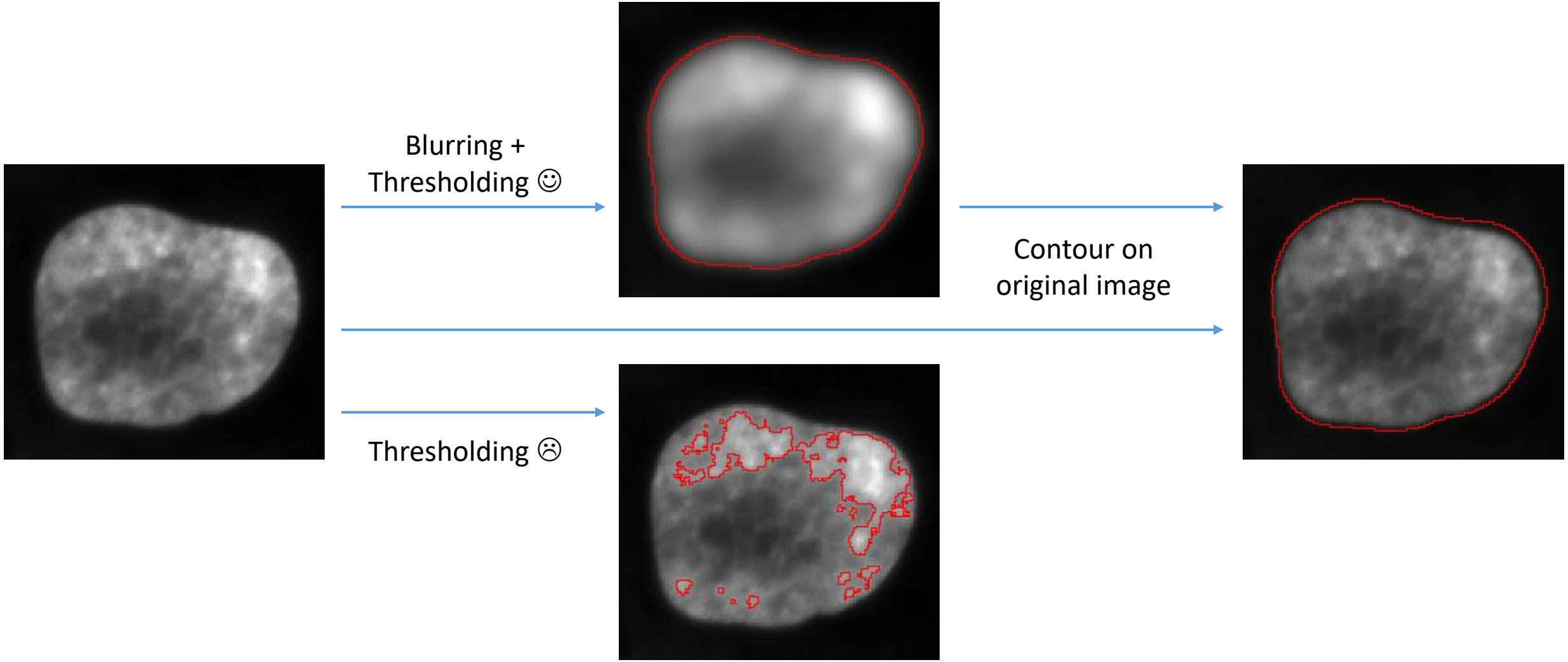

IsoData-threshold


Huang-threshold

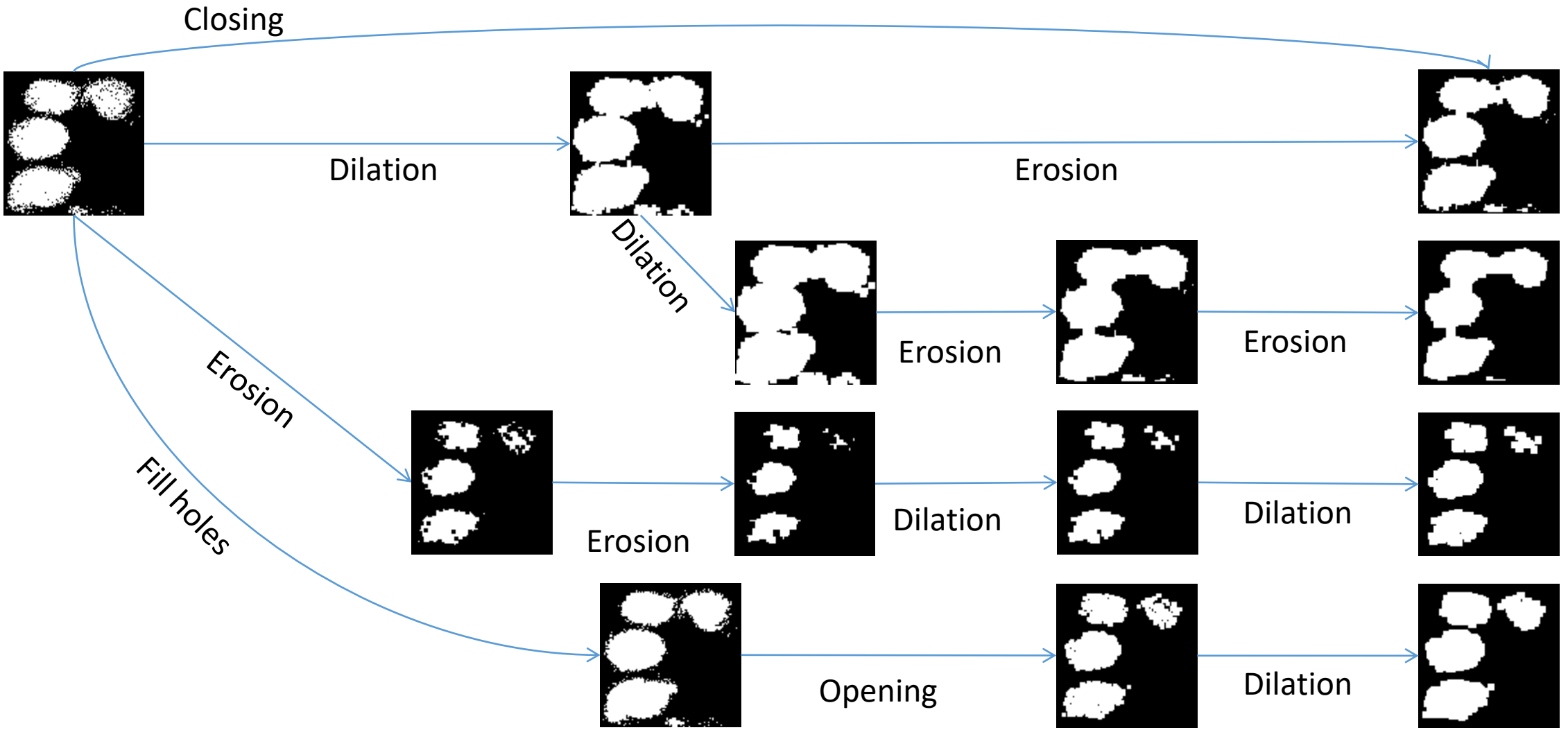

Otsu-threshold

Over-blurring to improve thresholding results

- In case thresholding algorithms outline the wrong structure, blurring may help.
- However: **Do not** continue processing the blurred image, continue with the original!

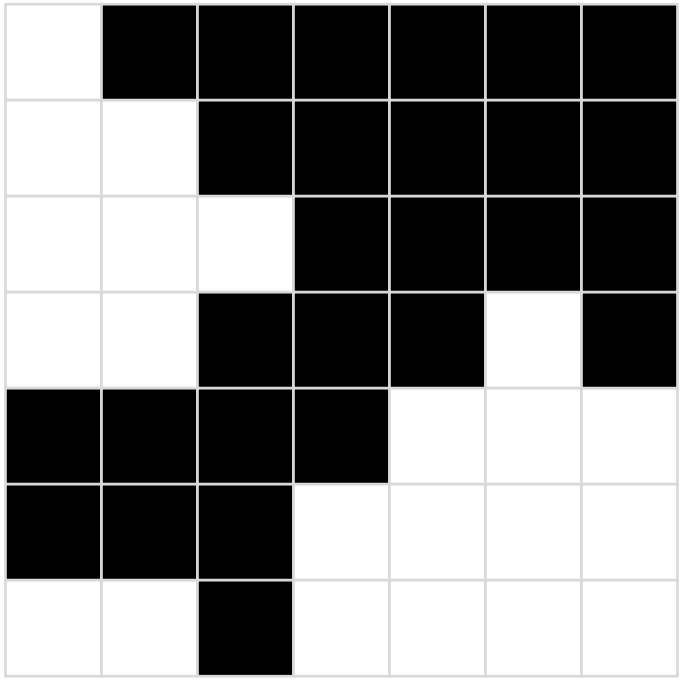


Refining masks: opening / closing



Connected components labelling

- In order to allow the computer differentiating objects, connected components analysis (CCA) is used to mark pixels belonging to different objects with different numbers
- Background pixels are marked with 0.
- The maximum intensity of a labelled map corresponds to the number of objects.



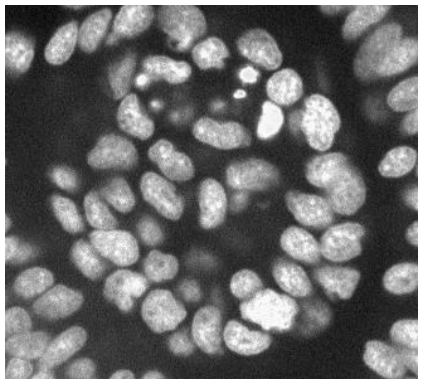
CCA

1	0	0	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0
1	1	0	0	0	3	0
0	0	0	0	3	3	3
0	0	0	3	3	3	3
2	2					

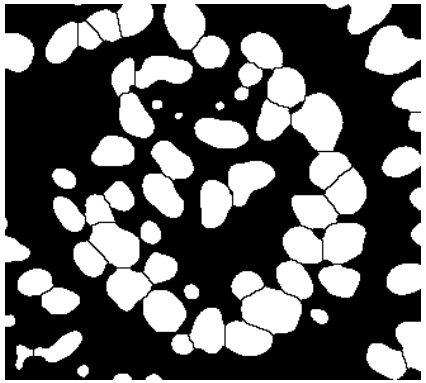
Is there a difference between connected components analysis and connected components labelling?

The particle analyser

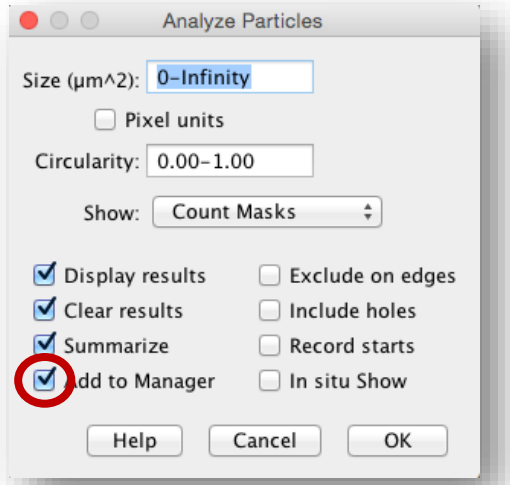
- Suggested workflow



Threshold, refine masks

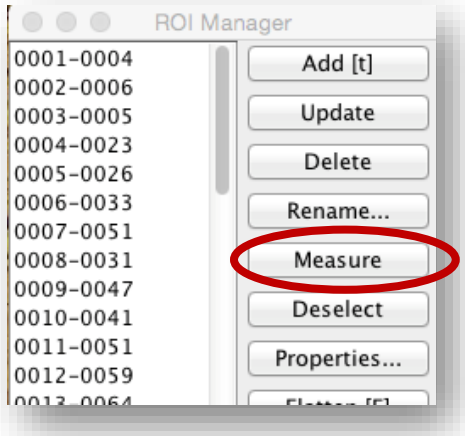


Particle analysis (CCA)

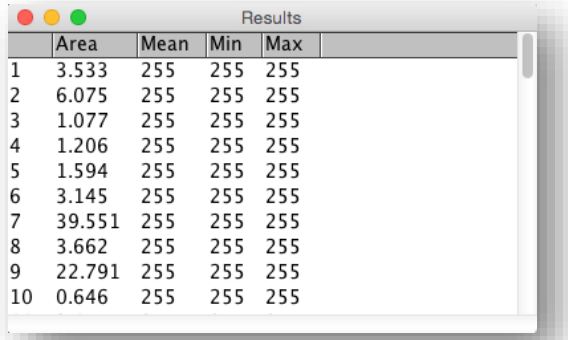


Add to manager

Select original image



Measure



Feature extraction

Robert Haase

July 2020

@haesleinhuepf

Feature extraction

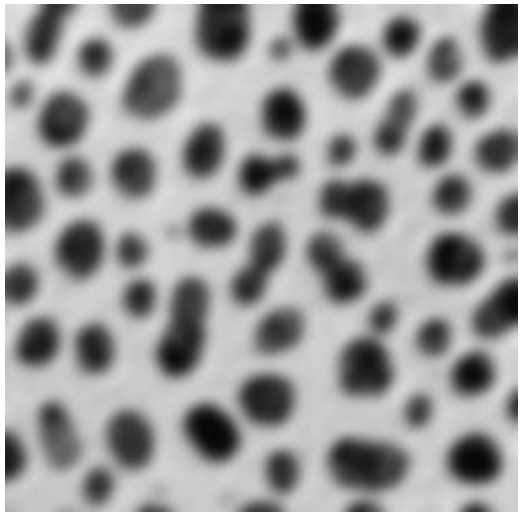
PoL Physics of Life TU Dresden

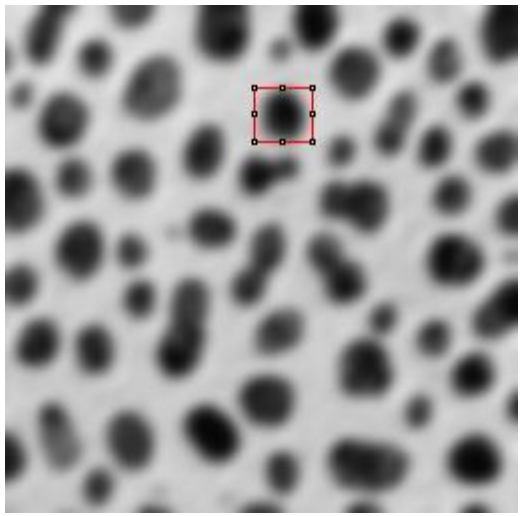
- A *feature* is a countable or measurable property of an image or object.
- Goal of feature extraction is finding a minimal set of features to describe an object well enough to differentiate it from other objects.

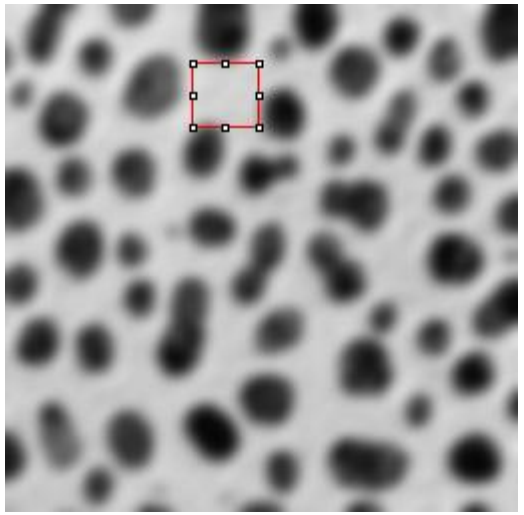
- Intensity based features
  - **Mean intensity**
  - **Standard deviation**
  - **Total intensity**
  - Textures
  - …
- Shape based / spatial features
  - Area / Volume
  - **Roundness**
  - **Solidity**
  - **Circularity /** Sphericity
  - Elongation
  - **Centroid**
  - **Bounding box**
  - …
- Spatio-temporal features
  - **Displacement,**
  - **Speed,**
  - Acceleration,
  - …
- Others
  - **Overlap**
  - Colocalisation
  - Network-analysis
  - …
- Mixed features
  - **Center of mass**
  - Local minima / maxima
  - …

Intensity based features

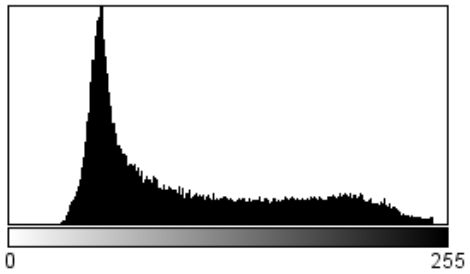
- Min / max
- Median
- Mean
- Mode
- Variance
- Standard deviation



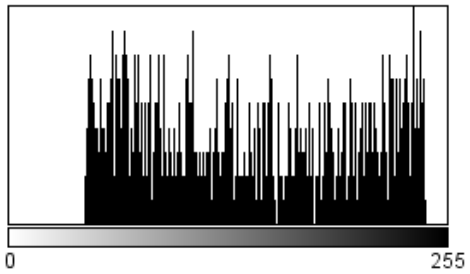




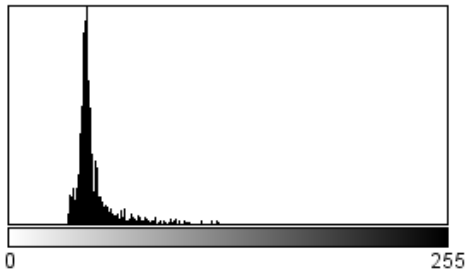
- Can be derived from pixel values
- Don’t take spatial relationship of pixels into account



Count: 65024
Mean: 103.301
StdDev: 57.991
Min: 29
Max: 248
Mode: 53 (1663)



Count: 783
Mean: 141.308
StdDev: 61.876
Min: 44
Max: 243
Mode: 236 (9)

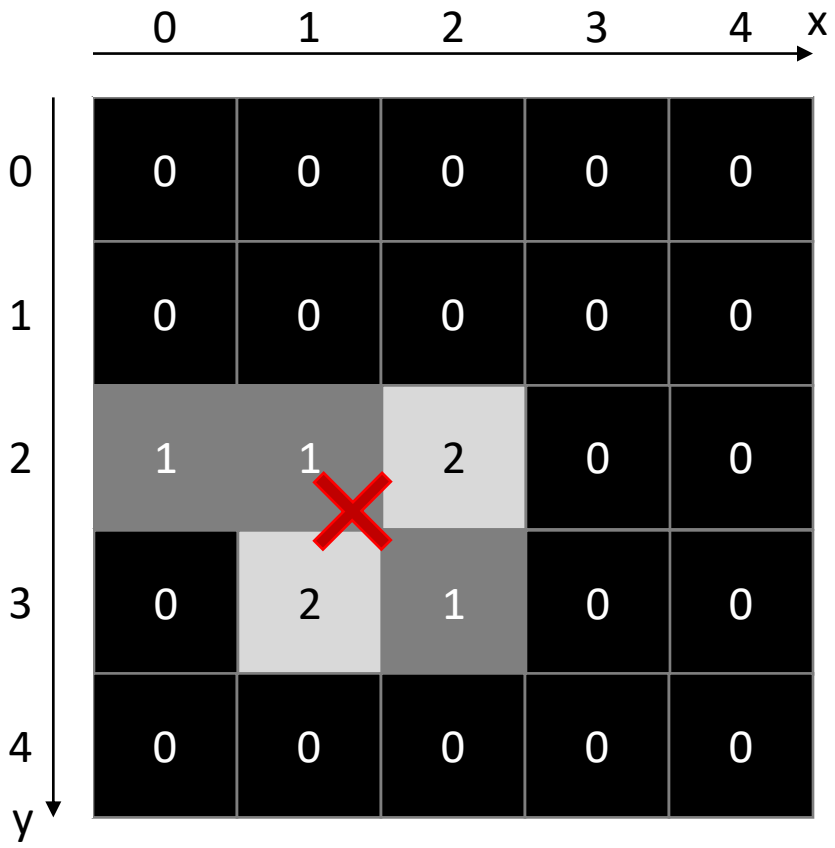


Count: 1056
Mean: 49.016
StdDev: 12.685
Min: 34
Max: 122
Mode: 45 (120)

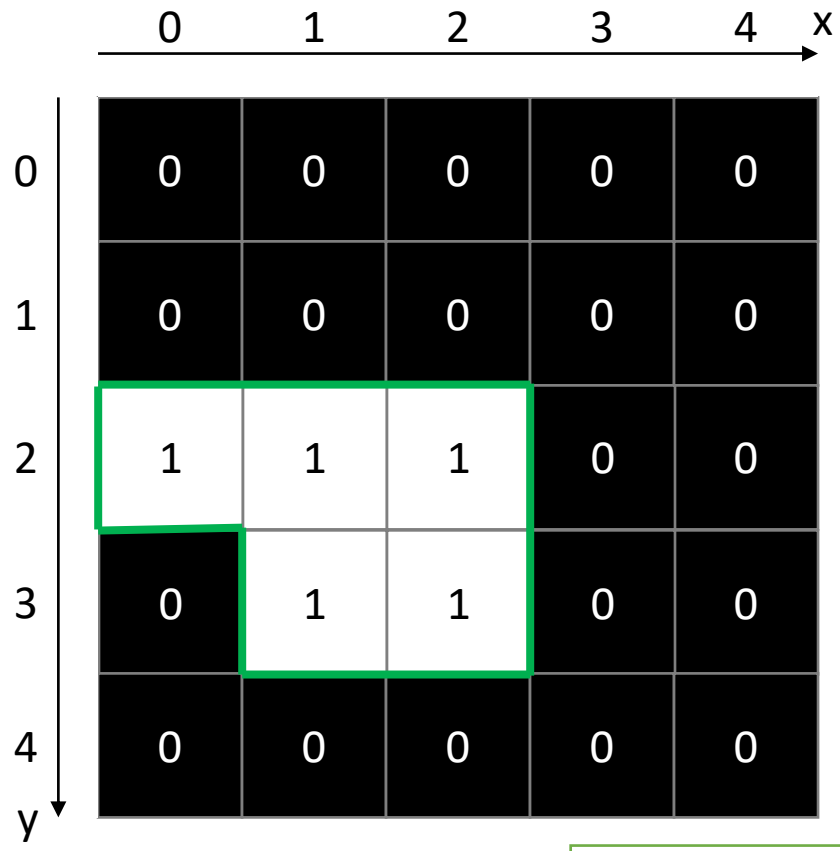
- See also:
  - descriptive statistics
  - histogram

Size / positional / geometrical features

- Where and how large are given objects?



- Center of mass / centroid



- Area, perimeter

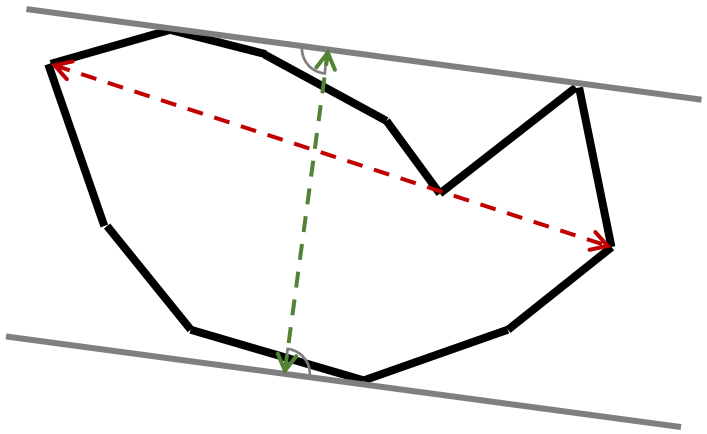
variable	value
x_b	0
y_b	2
w_b	3
h_b	2

- Bounding box

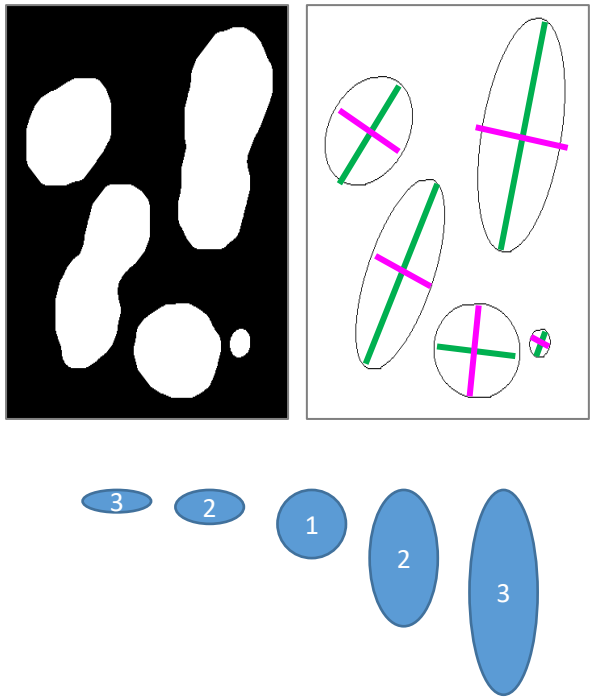
If the center of mass is a mixed feature, why is not the centroid (it in shape based/spatial features) if it's the center of mass for binary pictures? Thank you :)

Shape descriptors

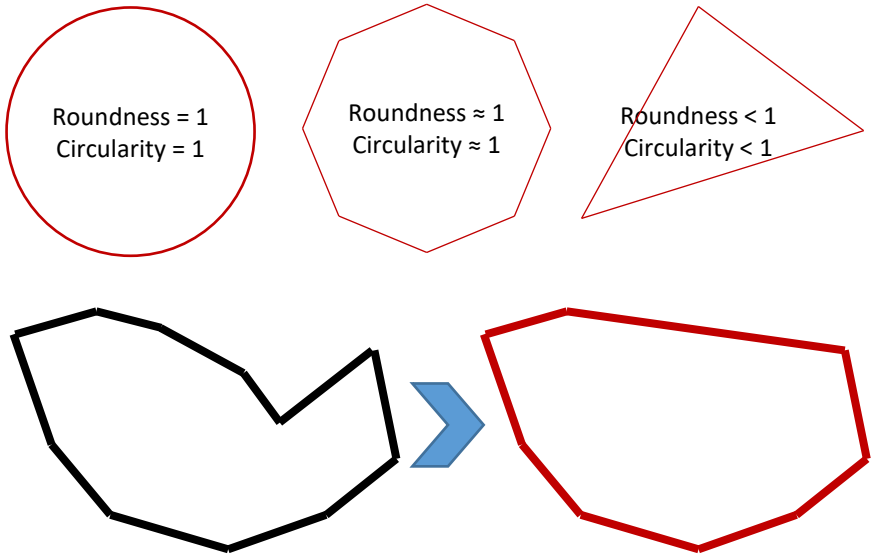
- How are objects shaped?



- Feret's diameter
- The minimum caliper ("Minimum Feret")



- Fit ellipse
  - Major axis … long diameter
  - Minor axis … short diameter
- Aspect ratio



- Roundness
- Circularity
- Solidity

PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Machine learning

Robert Haase, Myers lab, MPI CBG

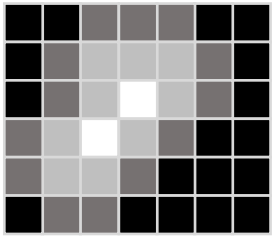
July 2020

@haesleinhuepf

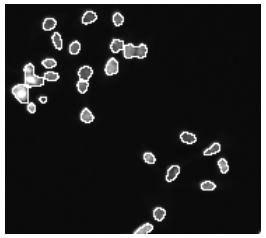
Machine learning

- Automatic construction of predictive models from given data

Pixels,

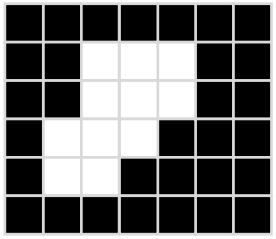


Objects,



Images, Audio, Text, Measurements, …

Dense Segmentation / Binarization



Object classification

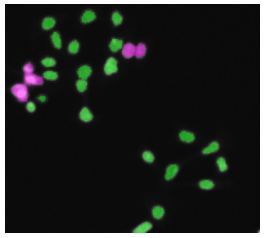
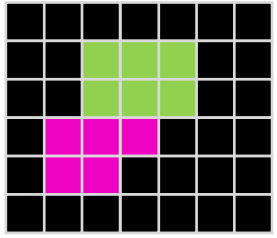


Image classification

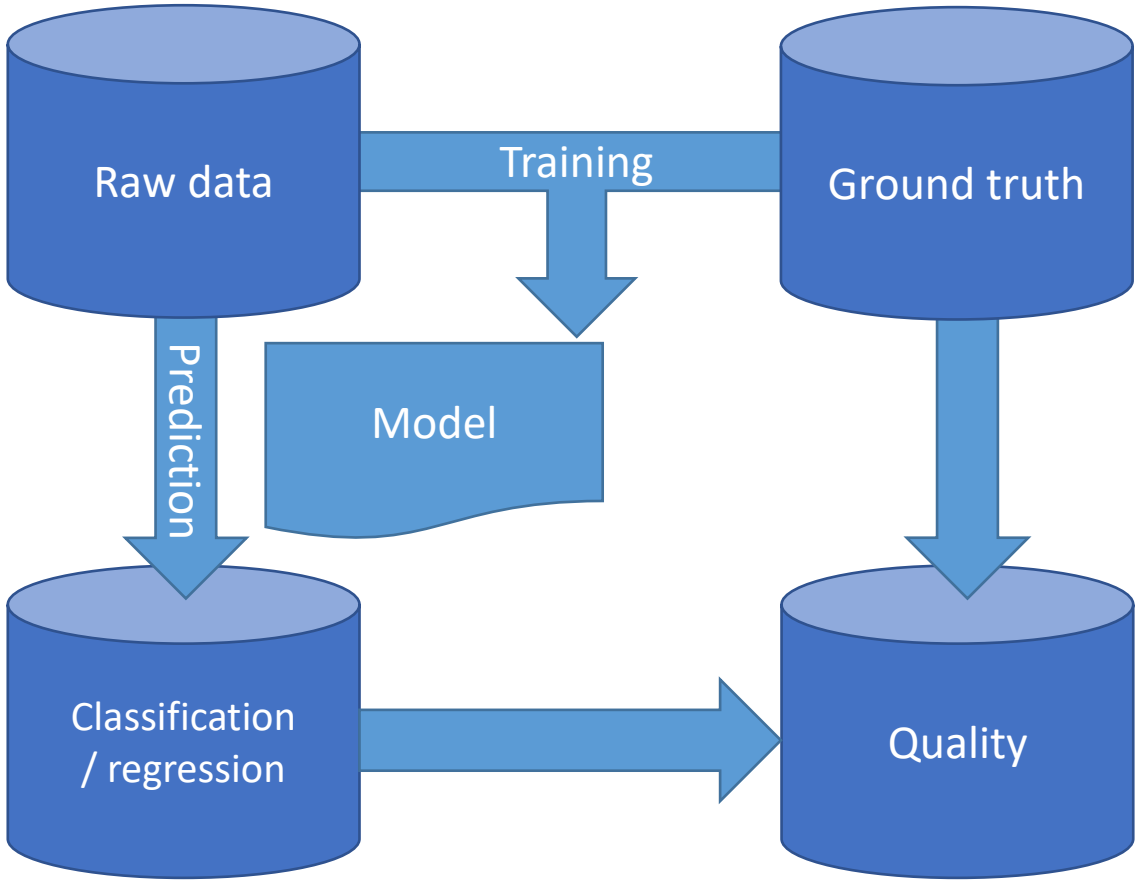
"Cat"

Instance segmentation



Cont. quantity

P_{Cat}= 0.5
$P_{Microscope}$= 0.4
Height = 80 cm

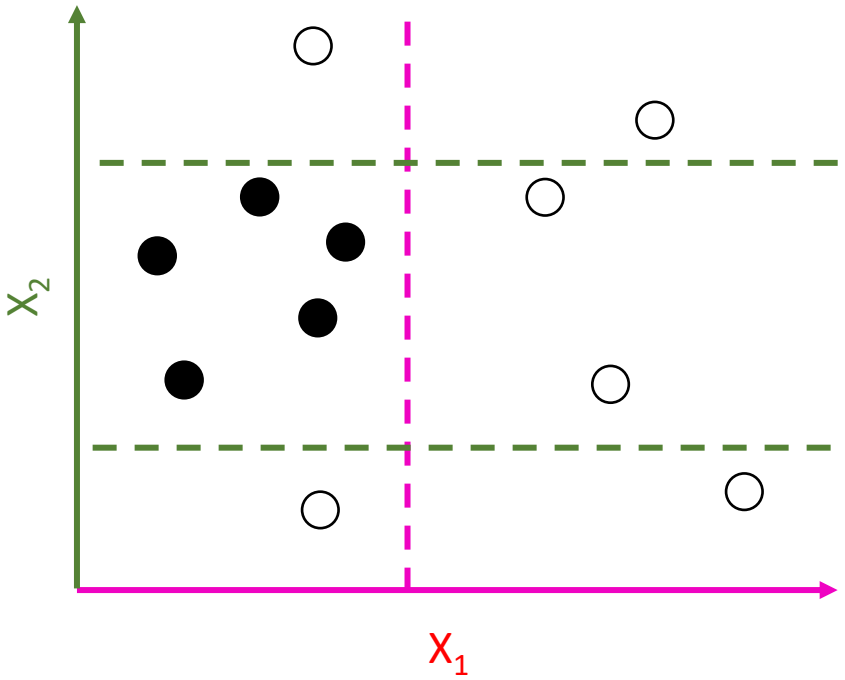


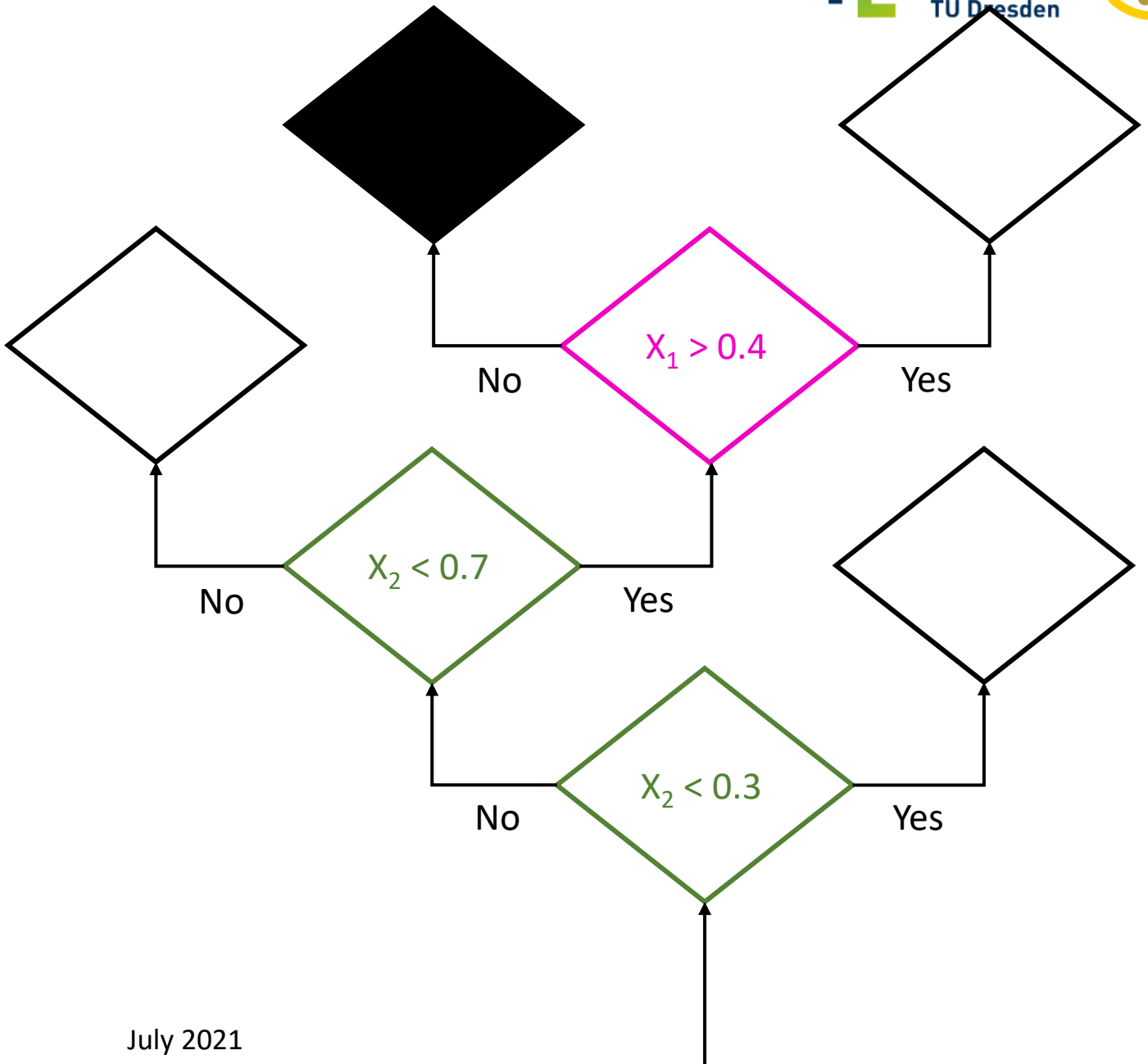
Annotated raw data, usually generated by humans

Precision, Recall

Deriving random decision trees

- Depending on sampling, the decision trees are different

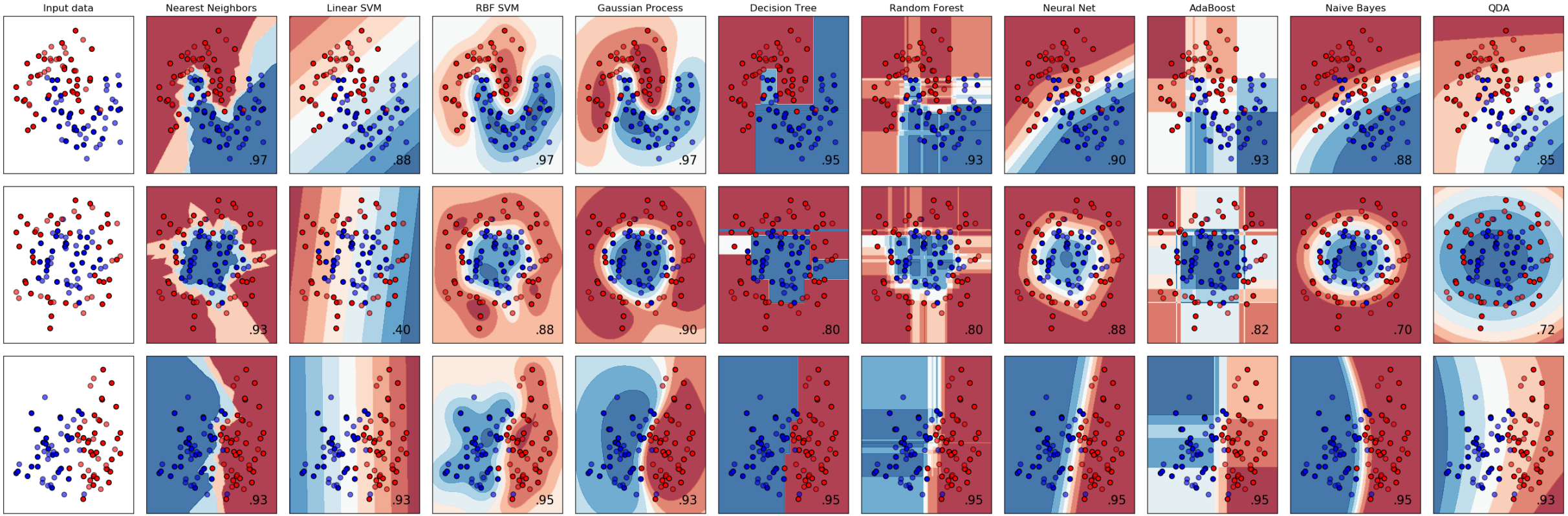




Approaches

PoL Physics of Life TU Dresden

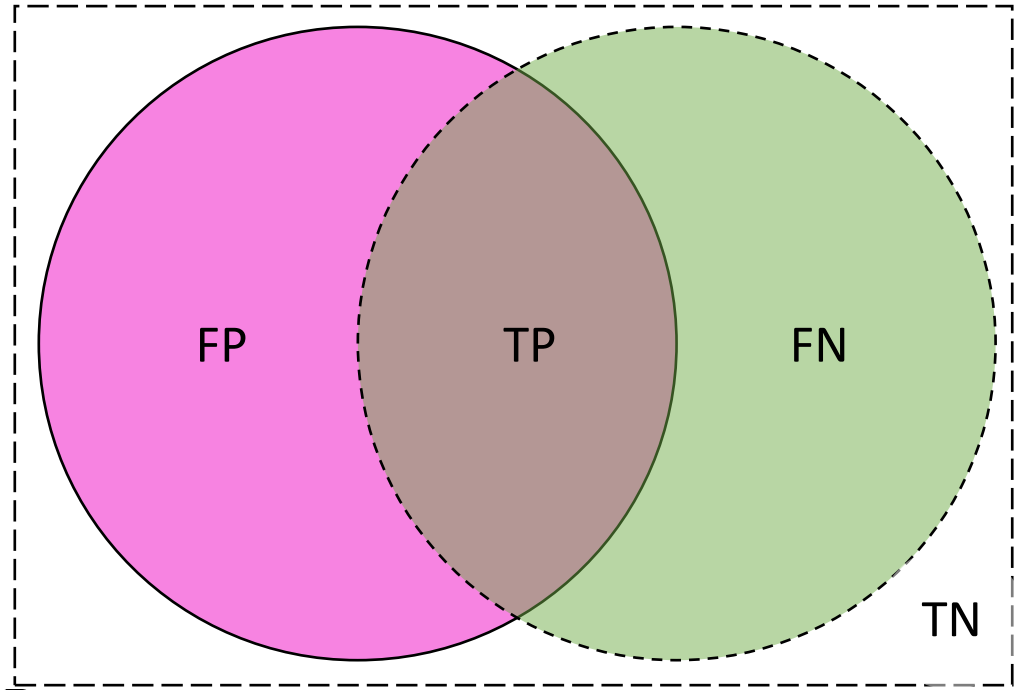
- The right approach depends on data, computational resources and desired quality



Adapted from https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Algorithm evaluation

- In general
  - Define what's positive and what's negative.
  - Compare with a reference to figure out what was true and false



A — Prediction A

B — Reference B (ground truth)

ROI — Region of interest

TP — True-positive

FN — False-negative

FP — False-positive

TN — True-negative

Overlap (a.k.a. Jaccard index) $\frac{TP}{TP + FN + FP}$

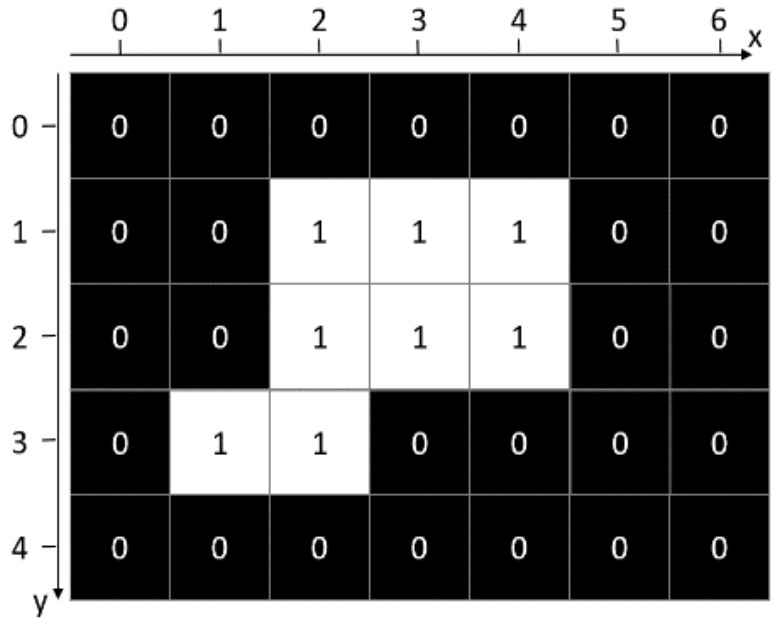
Precision $\frac{TP}{TP + FP}$ What fraction of points that were predicted as positives were really positive?

Recall (a.k.a. sensitivity) $\frac{TP}{TP + FN}$ What fraction of positives points were predicted as positives?

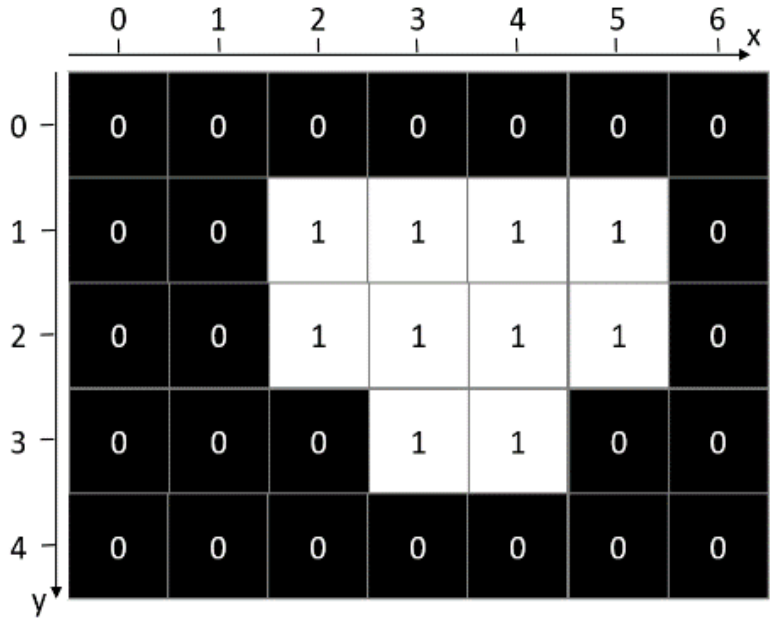
Algorithm evaluation

- Assume you are evaluating a binary segmentation algorithm by comparing its result to a given ground

Segmentation result



Ground truth



Jaccard index $\frac{TP}{TP+FN+FP}$

Precision $\frac{TP}{TP+FP}$

Recall $\frac{TP}{TP+FN}$

TP = 6
FP = 2
FN = 4
J = 6 / 12
P = 6 / 8
R = 6 / 10

PoL Physics of Life TU Dresden

CENTER FOR SYSTEMS BIOLOGY DRESDEN

Bio-statistics

Robert Haase, Myers lab, MPI CBG

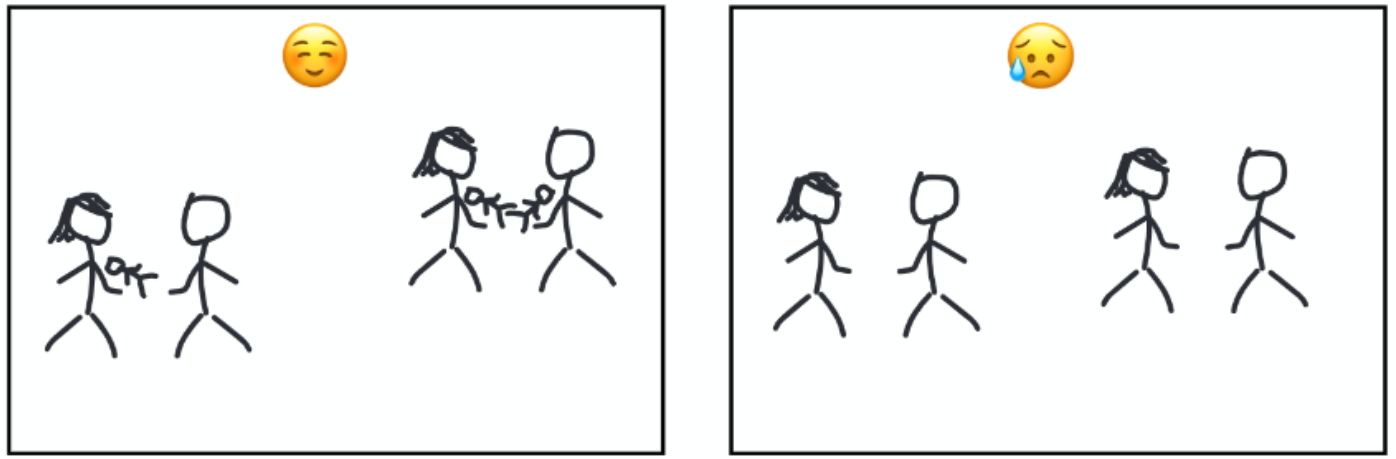
July 2020

@haesleinhuepf

Probability

Example: Success rate in in-vitro-fertilisation clinic

Do we count the babies or the mothers?



4 trials

3 babies:

2 successful attempts:

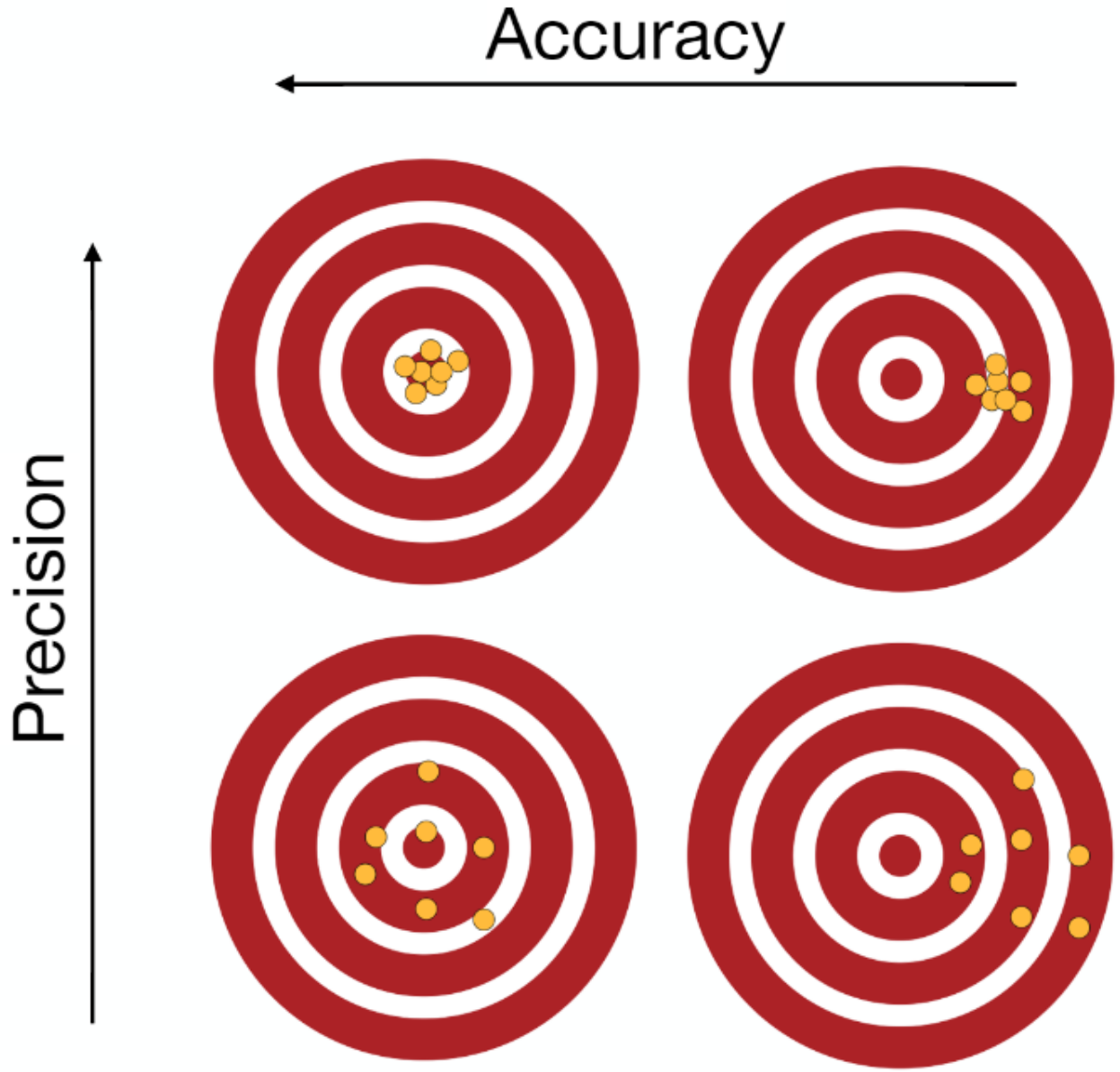
What is the success rate?

Confidence

- Which is wider, a 95% CI or 99%?
- Is it possible to generate a 100% confidence interval?
- Can you compute a confidence level, when you know the proportion, but not the sample size?
- Why is the confidence interval not symmetric?

Precision and accuracy

- What's the difference?



Summary parameters

- Discrete / continuous variables

Summary parameters

1 2 2 5 5 5 10 30

Min value: 1

Max value: 30

Parametric measures

Mean: (1+2+2+5+5+5+10+30)/8 = 7.5

Variance: (1+4+4+25+25+25+100+900)/8 = 90.57143

SD: square_root (variance) = 9.516902

SD = standard deviation = sigma

non-parametric measures:

	1	2	2	5	5	5	10	30
Ranks:	1	2	2	4	4	4	7	8

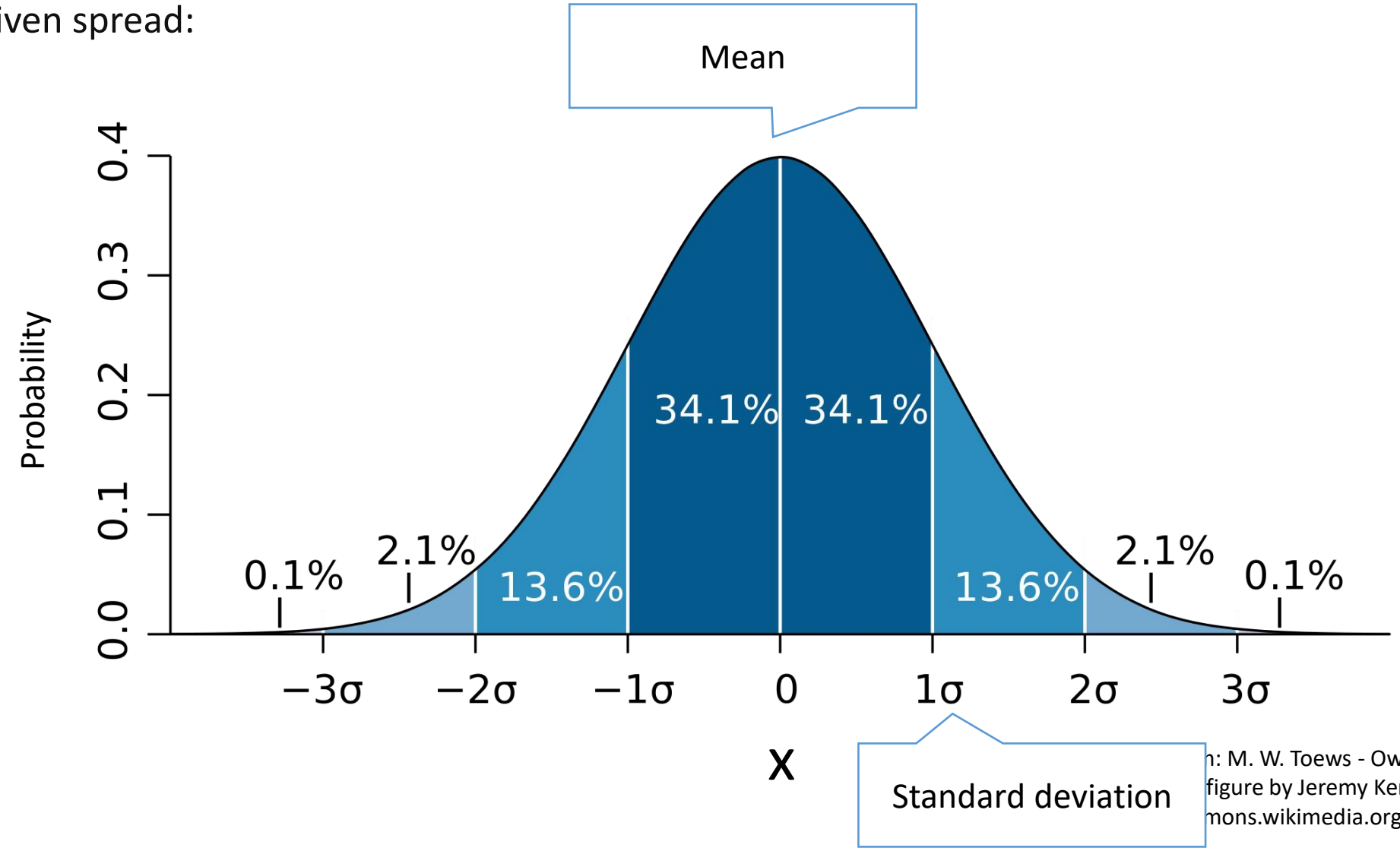
Median: the central value: 5

Quartiles: the value of the lower and upper quarter: 2, 6.25

Inter quartile range (IQR): 6.25-2

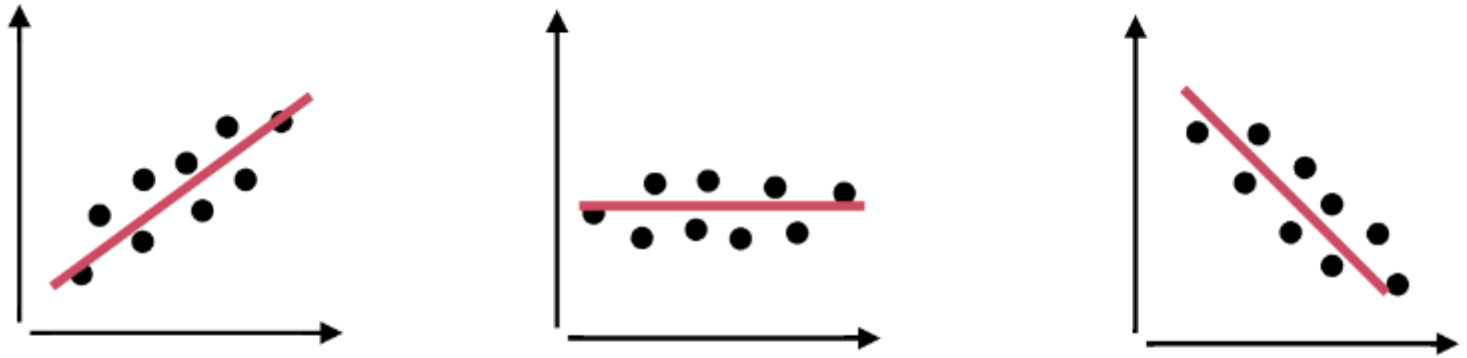
Normal distribution

- A population (or a sample) is normal distributed if observations are distributed around a determined mean with a given spread:



n: M. W. Toews - Own work, based (in concept) on figure by Jeremy Kemp, on 2005-02-09, CC BY 2.5, mons.wikimedia.org/w/index.php?curid=1903871

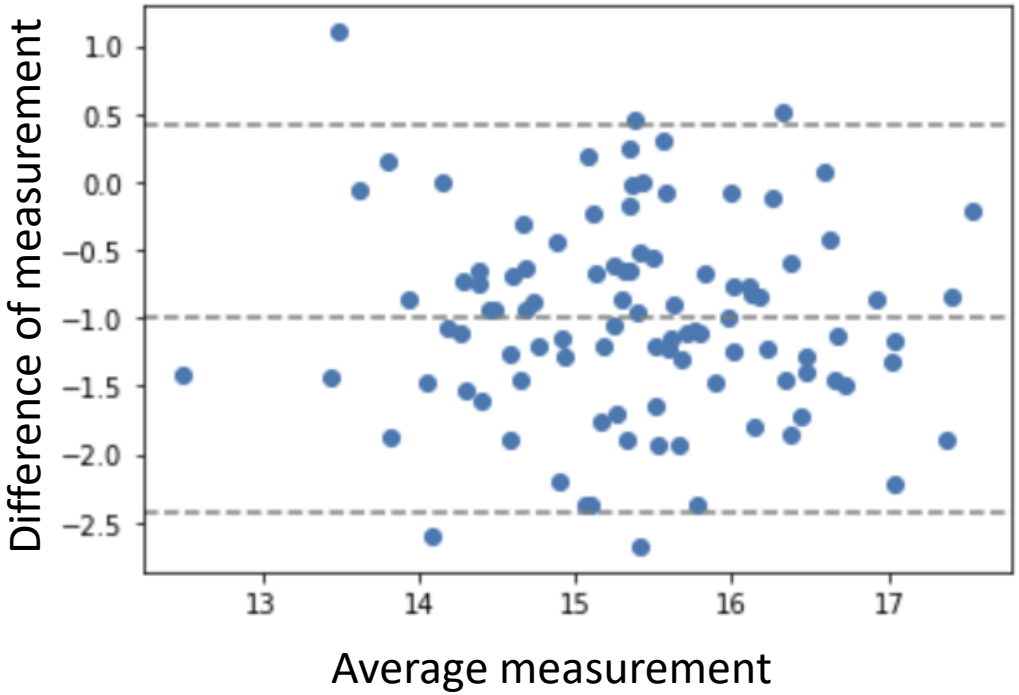
Correlation



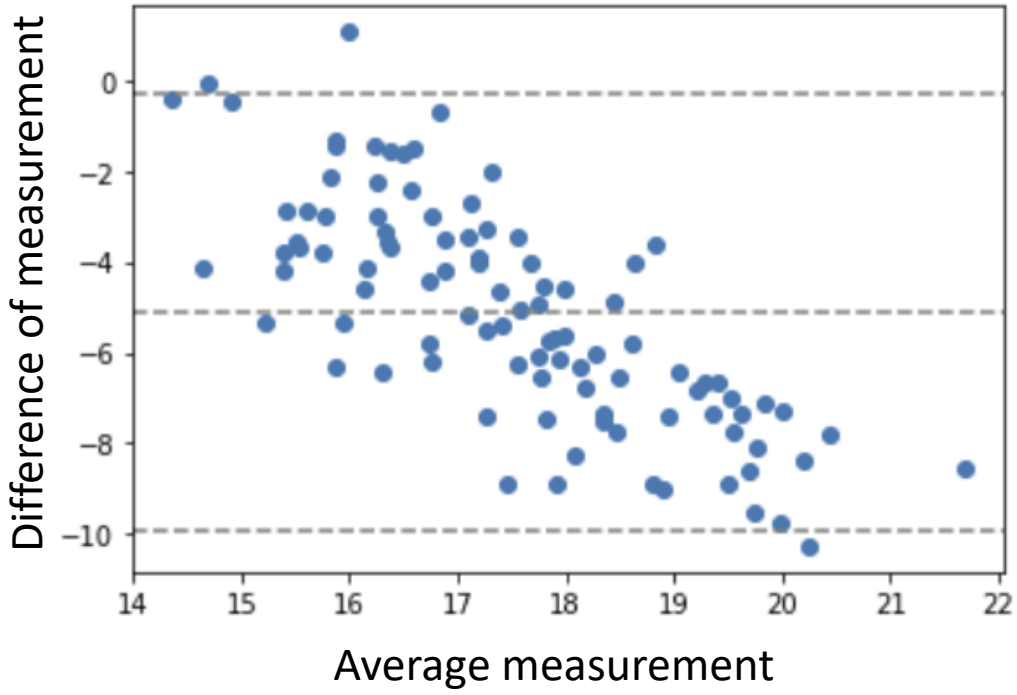
Bland-Altman analysis

- Bland-Altman plots allow us to differentiate various kinds of bias.

Agreement with additive error (offset)



Multiplicative bias



Programming

Robert Haase, Myers lab, MPI CBG

July 2020

@haesleinhuepf

Working with variables and string values

PoL Physics of Life TU Dresden

- Also strings as values for variables are supported
- When combining strings and numbers, you need to explicitly define what you want to do.

```
# mixing types

a = 5
b = "2"

print (a + b)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-4-51629e6a285f> in <module>
      4 b = "2"
      5
----> 6 print (a + b)

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
# mixing types

a = "5"
b = 2

print (a + b)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-85ae49867097> in <module>
      4 b = 2
      5
----> 6 print (a + b)

TypeError: can only concatenate str (not "int") to str
```

```
# mixing types to make numbers

a = 5
b = "2"

print (a + int(b))
```

```
7
```

```
# mixing types to make strings

a = "5"
b = 2

print (a + str(b))
```

```
52
```

- Conversion to a floating point number: float()

Generating arrays within for-loops

PoL Physics of Life TU Dresden

- There is a long and a short way for creating arrays with numbers.

```
# we start with an empty list
numbers = []

# and add elements
for i in range(0, 5):
    numbers.append(i * 2)

print(numbers)
```

```
[0, 2, 4, 6, 8]
```

```
numbers = [i * 2 for i in range(0, 5)]

print(numbers)
```

```
[0, 2, 4, 6, 8]
```

Generating arrays within for-loops

PoL Physics of Life TU Dresden

- Also a combination with the if-statement is possible

```
# we start with an empty list
numbers = []

# and add elements
for i in range(0, 5):
    # check if the number is odd
    if i % 2:
        numbers.append(i * 2)

print(numbers)
```

```
[2, 6]
```

```
numbers = [i * 2 for i in range(0, 5) if i % 2]

print(numbers)
```

```
[2, 6]
```

Subsets

```
# Arrays
numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(numbers)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

- Creating subsets of arrays

Starting at (including)

Ending at (excluding)

```
subset = numbers[2:4]
print(subset)
```

```
[2, 3]
```

Step size

```
subset_with_gaps = arr[1:8:2]
print(subset_with_gaps)
```

```
[1, 3, 5, 7]
```

Tables

- Tables can be dictionaries with lists as values

```
measurements_week = {
    'Monday':[2.3, 3.1, 5.6],
    'Tuesday':[1.8, 7.0, 4.3],
    'Wednesday':[4.5, 1.5, 3.2],
    'Thursday':[1.9, 2.0, 6.4],
    'Friday':[4.4, 2.3, 5.4]
}
```

```
measurements_week
```

```
]: {'Monday': [2.3, 3.1, 5.6],
    'Tuesday': [1.8, 7.0, 4.3],
    'Wednesday': [4.5, 1.5, 3.2],
    'Thursday': [1.9, 2.0, 6.4],
    'Friday': [4.4, 2.3, 5.4]}
```

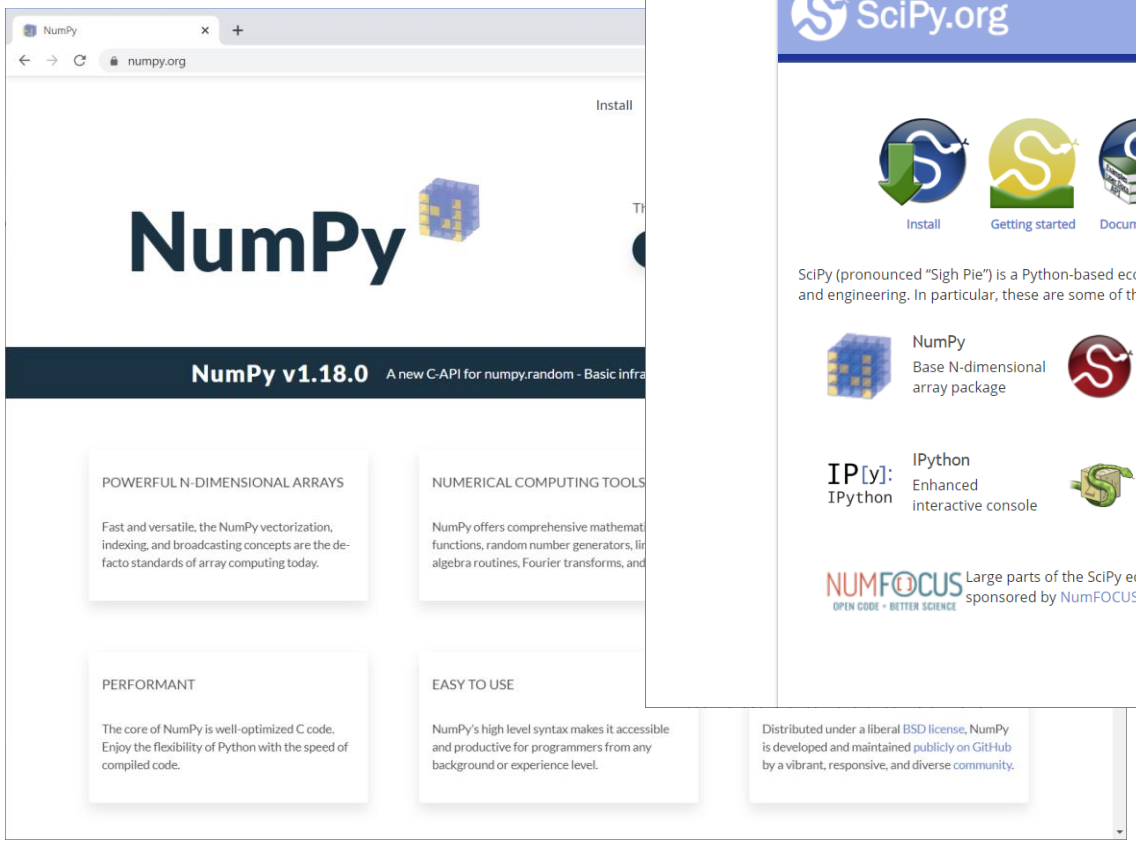
- Retrieve a column

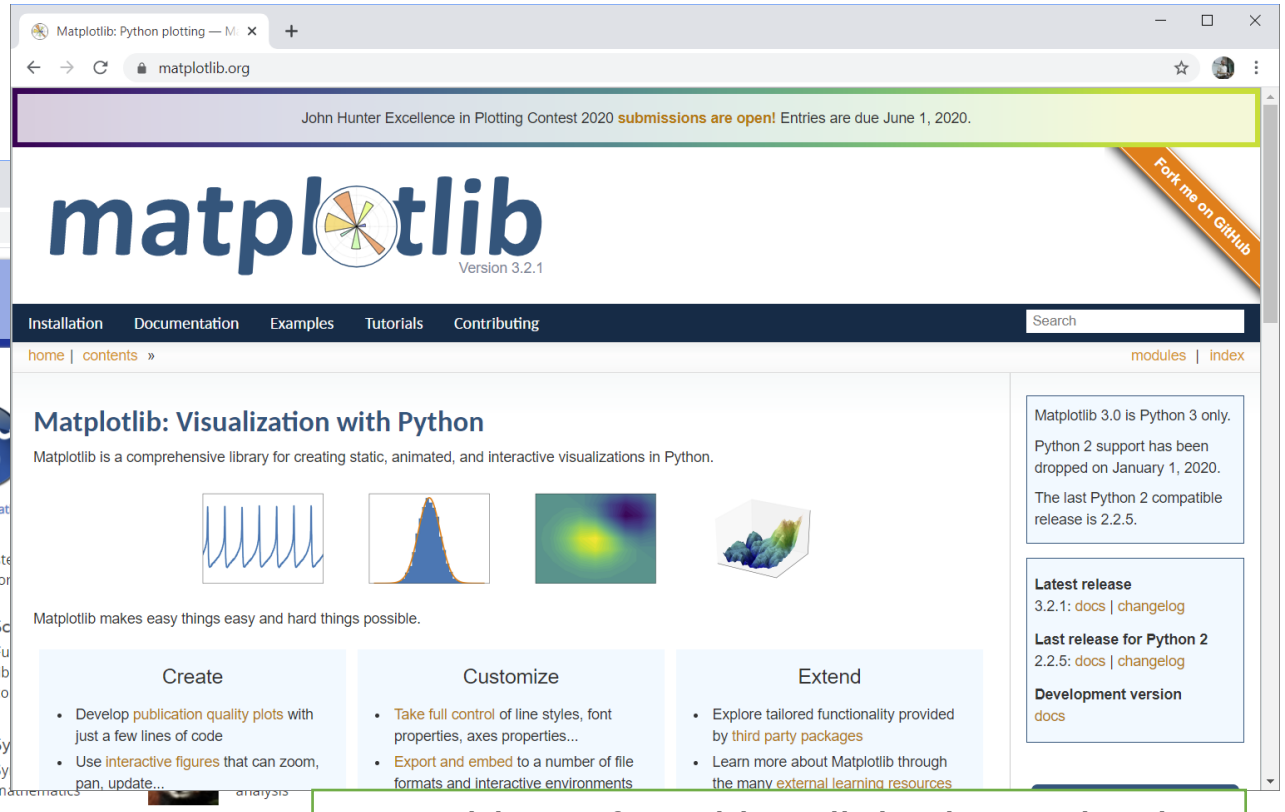
```
measurements_week['monday']
```

```
]: [2.3, 3.1, 5.6]
```

Python with numpy, scipy, matplotlib

- https://numpy.org
- https://scipy.org
- https://matplotlib.org





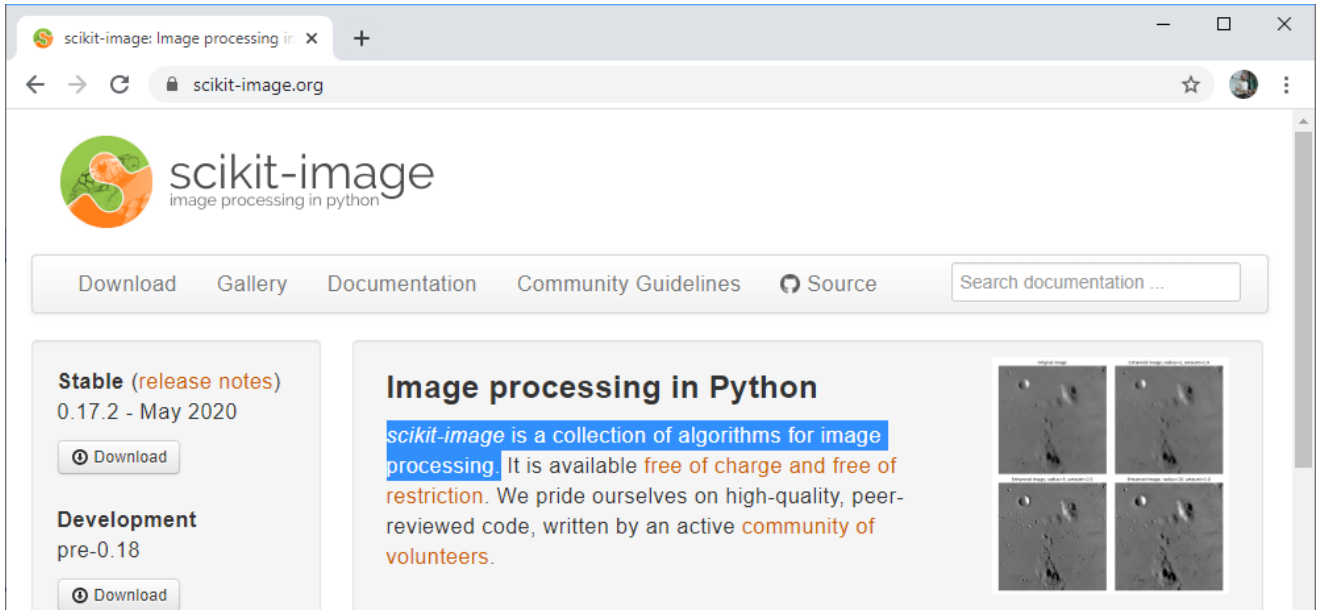
I'm bit confused by all the intorudced tools. Like for plotting or table processing, the possible solutions posted online are so different, either in codes or in the tools they used for each cases. And it's also mostly hard for me to know where the bug comes from, how to decide which to try?

Python: Image processing

- https://scikit-image.org/

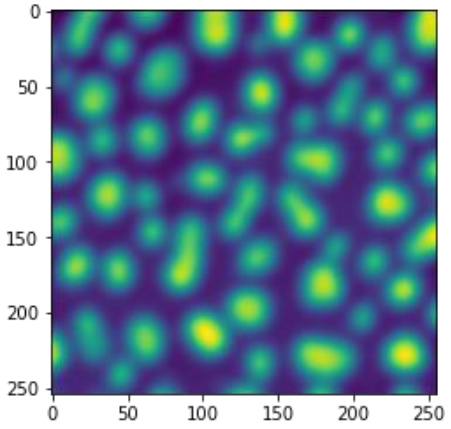
```
from skimage import filters

filters.
```

```
LPIFilter2D
median
meijering
prewitt
prewitt_h
prewitt_v
rank
rank_order
ridges
roberts
```

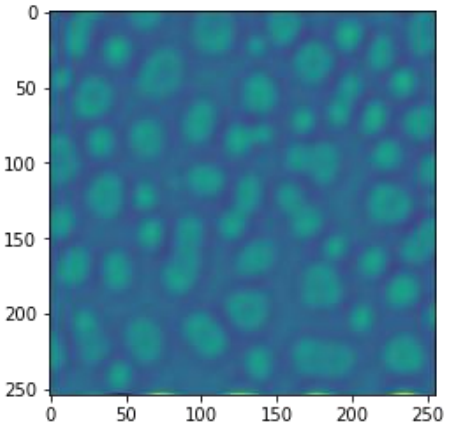



```
from skimage import filters

# Gaussian blur
gaussian_blurred_image = filters.gaussian(image, 5)
plt.imshow(gaussian_blurred_image)
plt.show()

# LoG
laplacian_of_gaussian = filters.laplace(gaussian_blurred_image)
plt.imshow(laplacian_of_gaussian)
plt.show()
```

Gaussian blur (sigma=5)



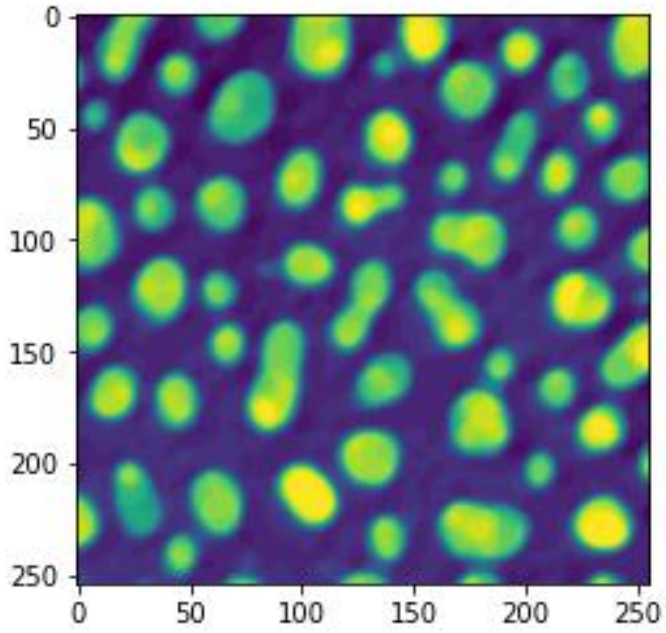
LoG

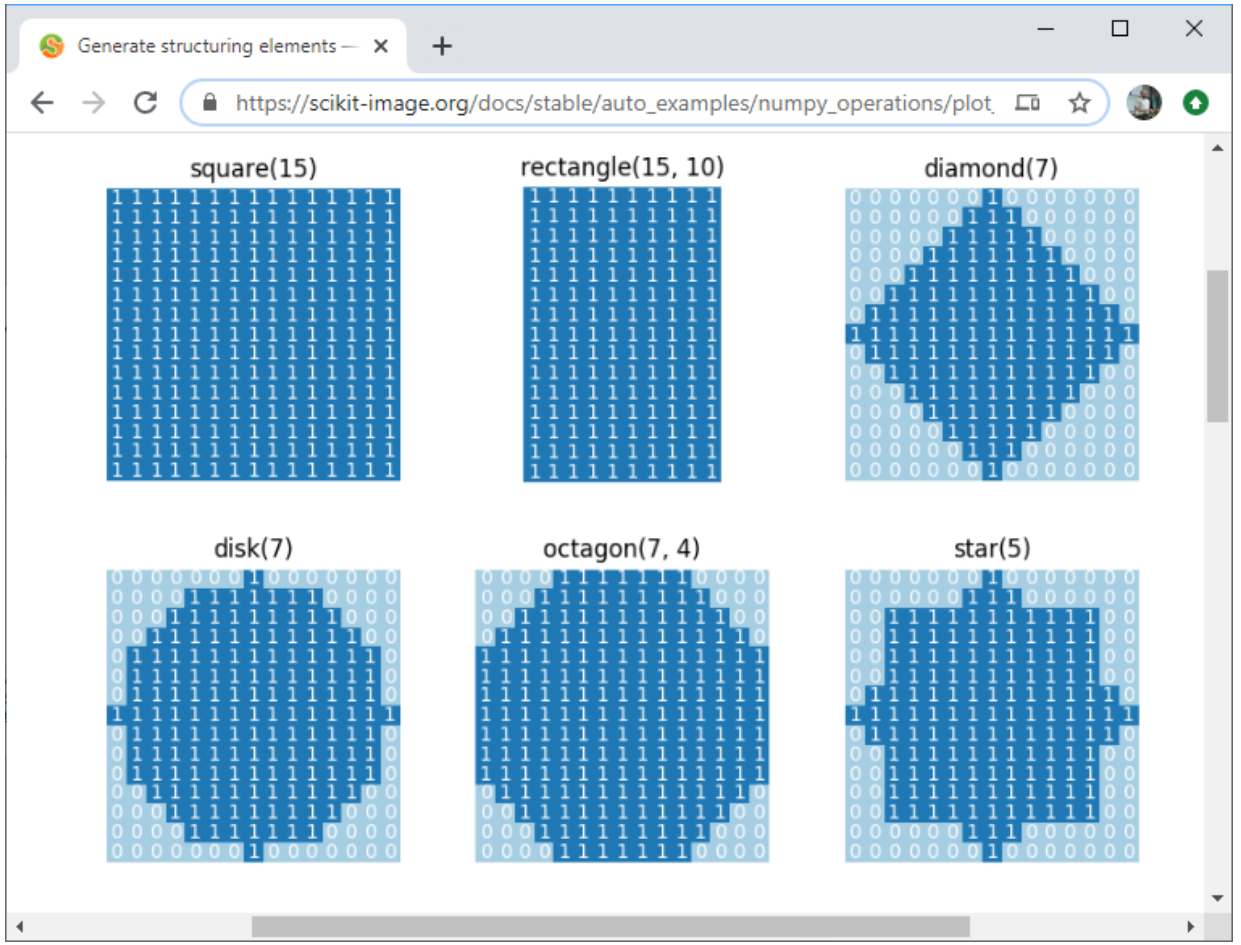


Filtering images: structuring elements

- Some filters ask for structuring elements

```
from skimage.morphology import disk

median_filtered = filters.median(image, disk(5))
plt.imshow(median_filtered)
plt.show()
```



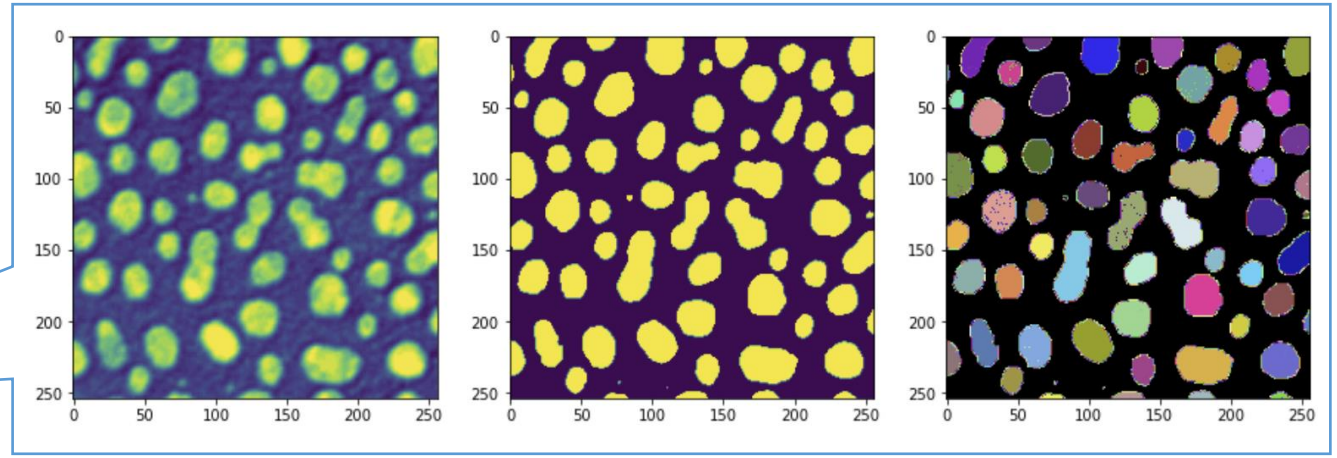


https://scikit-image.org/docs/stable/auto_examples/numpy_operations/plot_structuring_elements.html#sphx-glr-auto-examples-numpy-operations-plot-structuring-elements-py

Connected components analysis

- The label function applies connected component analysis.

```
# Load data
from skimage.io import imread
blobs = imread("blobs.tif")

# Thresholding
from skimage.filters import threshold_otsu
threshold = threshold_otsu(blobs)
binary_blobs = blobs > threshold

# Labeling
from skimage.measure import label
labeled_blobs = label(binary_blobs)

# Visualization
import matplotlib.pyplot as plt
fig, axs = plt.subplots(1, 3, figsize=(15,15))

axs[0].imshow(blobs)
axs[1].imshow(binary_blobs)
axs[2].imshow(labeled_blobs, cmap=label_cmap)
```

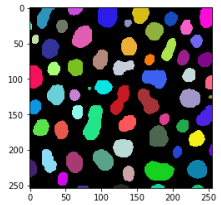


https://scikit-image.org/docs/dev/api/skimage.measure.html#skimage.measure.label

Measurements with scikit-image: regionprops

PoL Physics of Life TU Dresden

- Regionprops to collect measurements



We start with a label image

```
from skimage import measure


# analyse objects
properties = measure.regionprops(label_image, intensity_image=image)
```

- Reorganize in a dictionary of lists

```
statistics = {
    'area':       [p.area                for p in properties],
    'mean':       [p.mean_intensity      for p in properties],
    'major_axis': [p.major_axis_length   for p in properties]
}
```

- Visualize as table, a.k.a. pandas DataFrame

```
import pandas as pd
dataframe = pd.DataFrame(statistics)
```

	area	mean	major_axis	aspect_ratio
0	429	191.440559	34.779230	2.088249
1	183	179.846995	20.950530	1.782168
2	658	205.604863	30.198484	1.067734
3	433	217.515012	24.508791	1.061942
4	472	213.033898	31.084766	1.579415
...	...	...	...	...
57	213	184.525822	18.753879	1.296143
58	79	184.810127	18.287489	3.173540
59	88	182.727273	21.673692	4.021193
60	52	189.538462	14.335104	2.839825
61	48	173.833333	16.925660	4.417297

- Save to disk

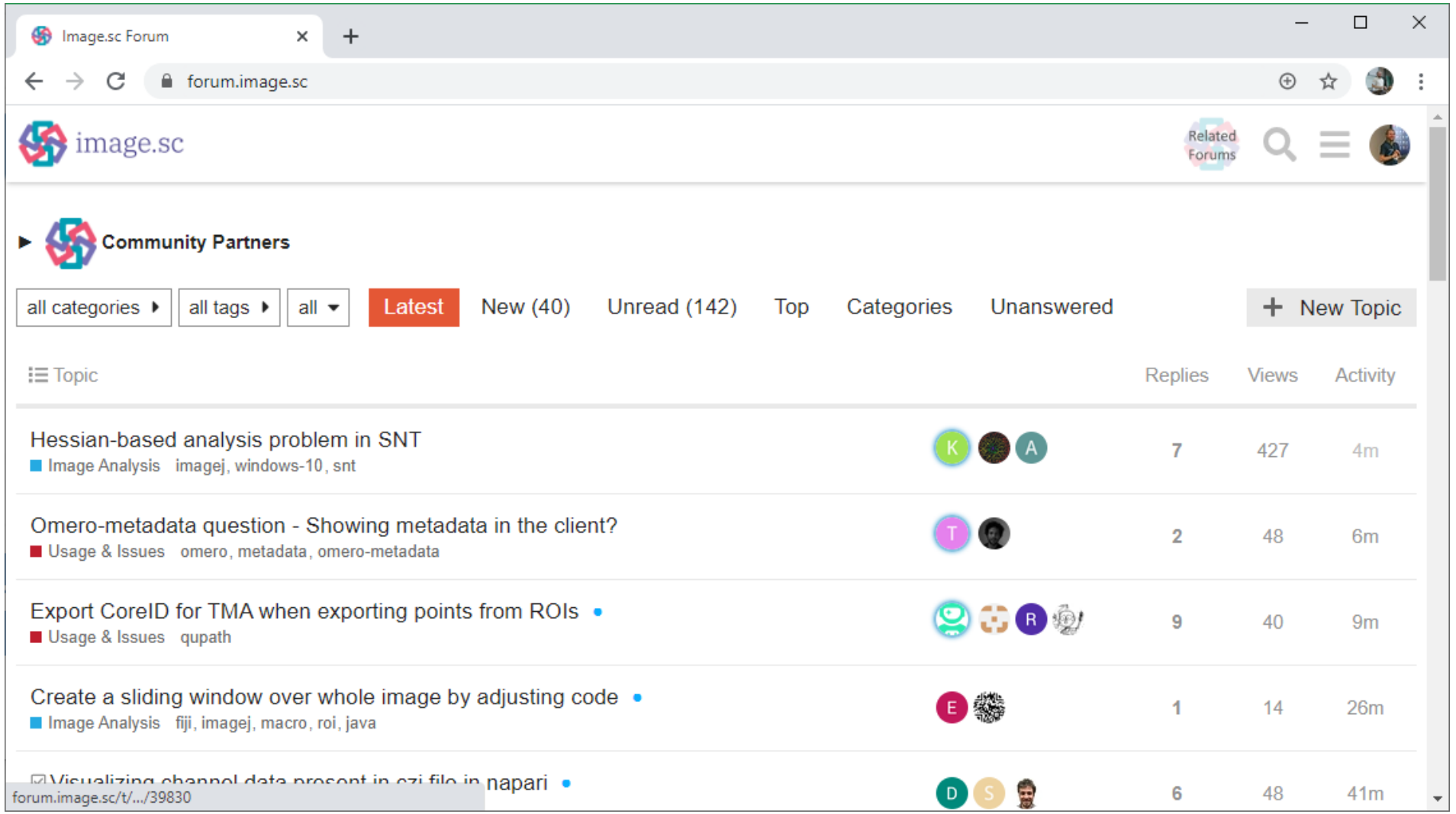
```
dataframe.to_csv("blobs_analysis.csv")
```

https://pandas.pydata.org/

ImageJ, Fiji & friends: community

PoL Physics of Life TU Dresden

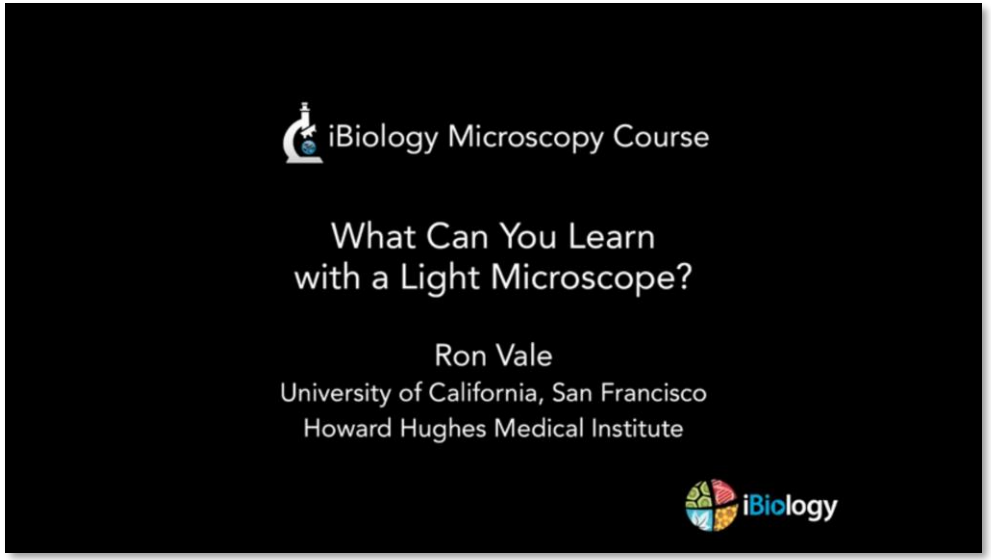
- Visit http://forum.image.sc !



What's next?

Microscopy

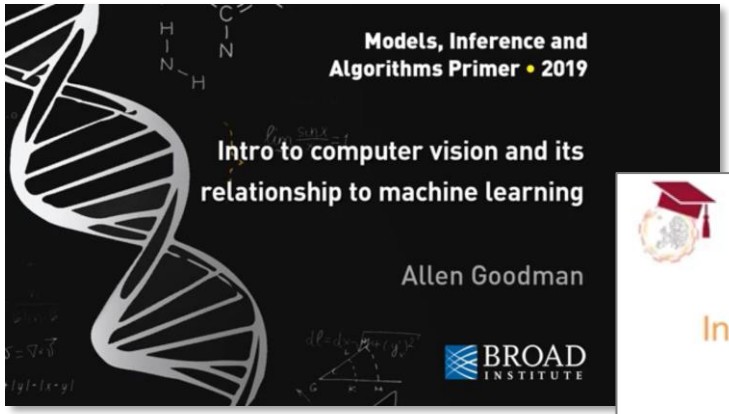
- BioDIP Dresden Light Microscopy Course: https://youtu.be/60_jgZtyR6U
- Microcourses: https://youtu.be/Tkc_GOCjx7E
- iBiology Microscopy Course: https://youtu.be/4c5ILWQmqRY

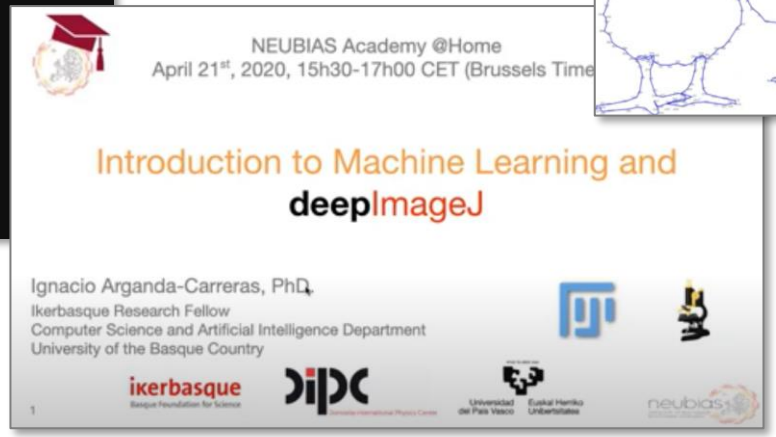


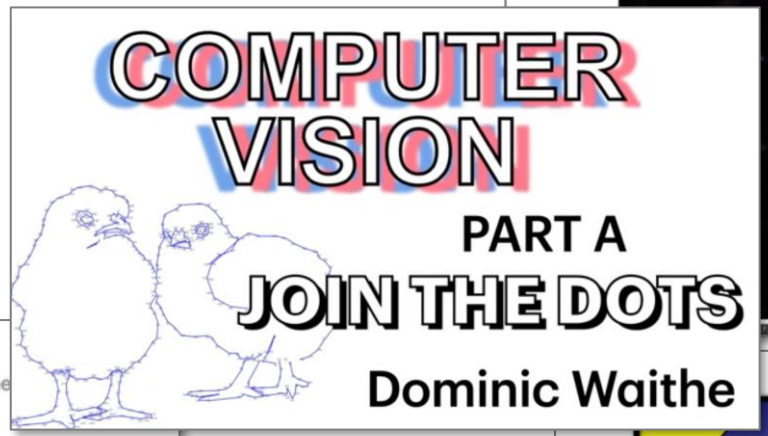
What's next?

PoL Physics of Life TU Dresden

- More machine learning for Bio-Image analysis?
  - Computer Vision / machine learning: https://youtu.be/Kzb5vTpvDBM
  - Computer vision: https://youtu.be/Smw3suzynho
  - DeepImageJ: https://youtu.be/0vTbsO8Vnuo
  - CSBDeep: https://youtu.be/ipp0mxfjhwY
  - StarDist: https://youtu.be/Amn_eHRGX5M
  - ilastik: https://www.youtube.com/ilastikTeam

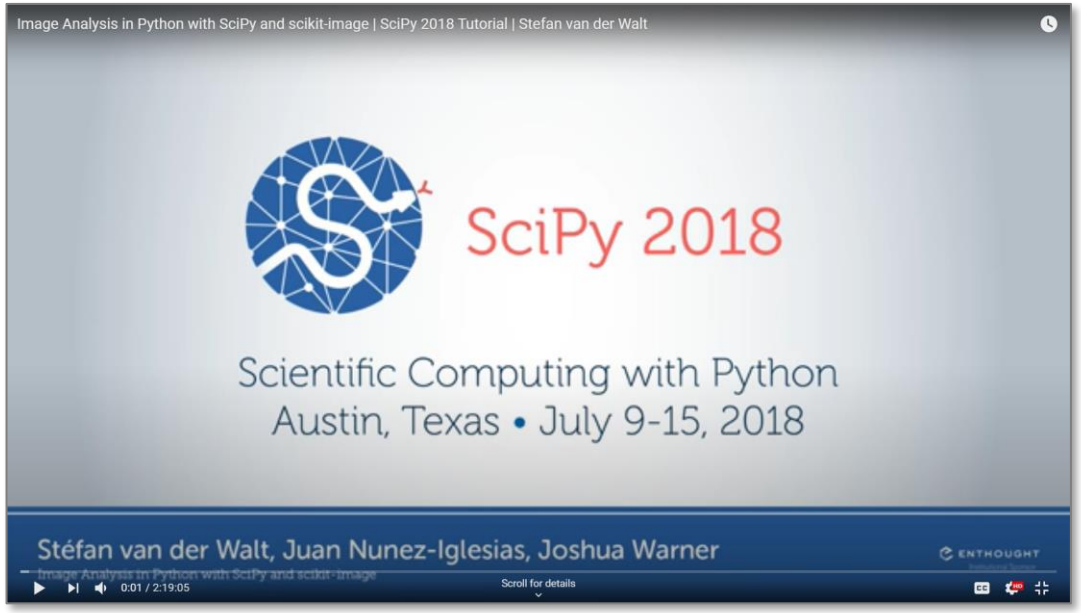


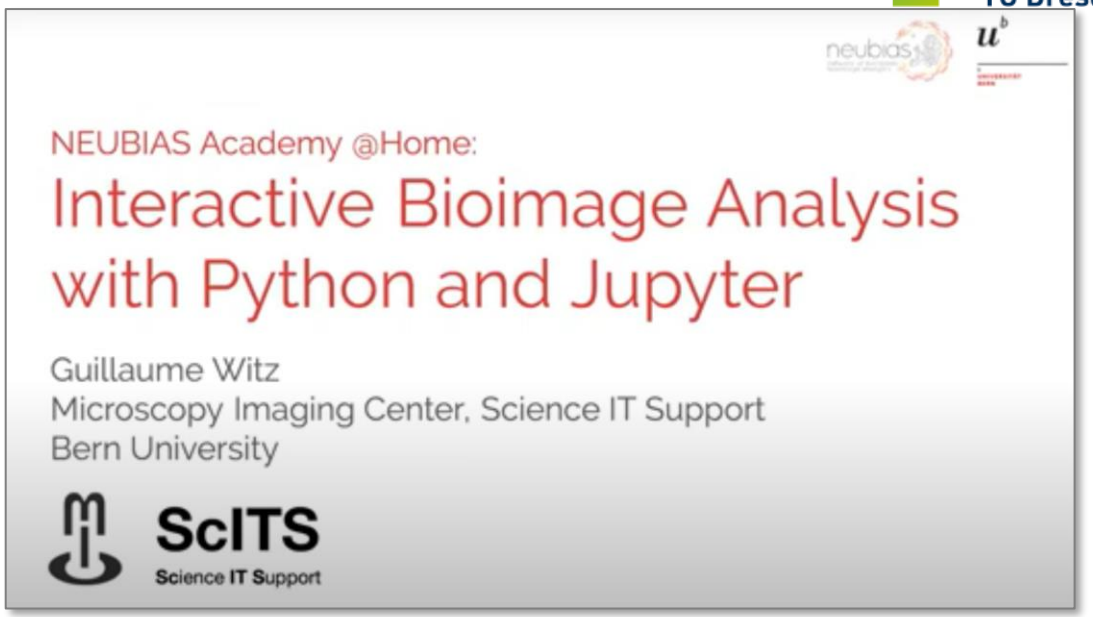




What's next?

- Image Analysis with Python
  - Python & Jupyter
    - https://youtu.be/2KF8vBrp3Zw
    - https://youtu.be/Y3pB3wnOivE
  - Scikit-image
    - https://youtu.be/pZATswy_IsQ
    - https://youtu.be/d1CIV9irQAY
  - Napari
    - https://youtu.be/VgvDSq5aCDQ





What's next?

PoL Physics of Life TU Dresden

More general

- Image data integrity: https://youtu.be/c_Oi2HKom_Y
- Coloc: https://youtu.be/cOrCz4qc8DI
- Automated microscopy: https://youtu.be/w0ERCrKx4gk