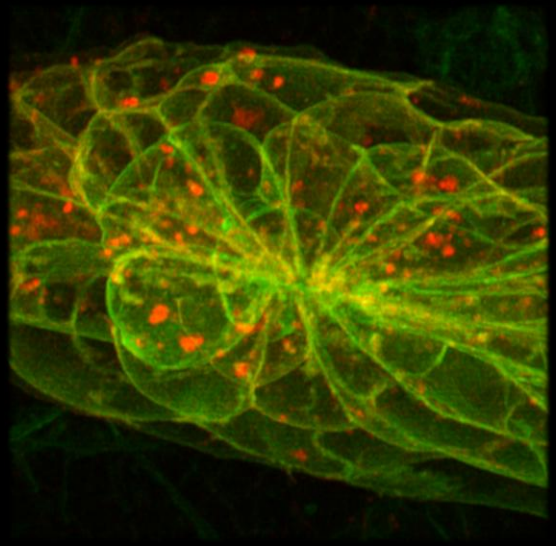
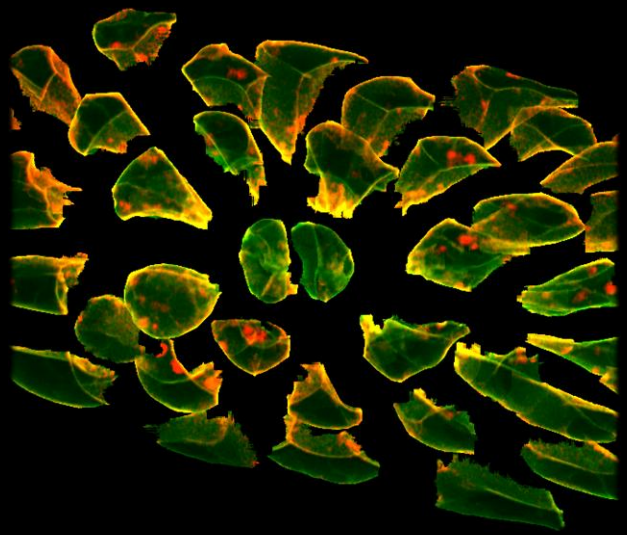
Python BioImage Analysis Tutorial

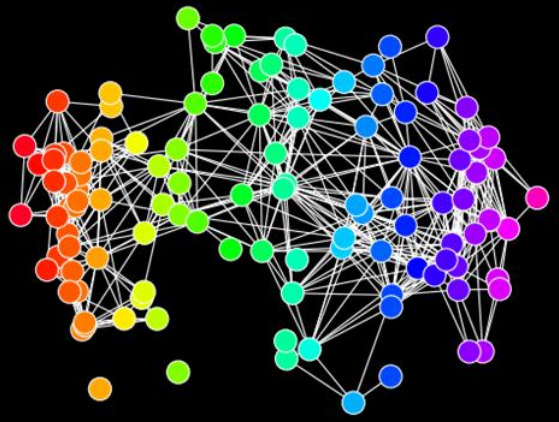
EMBL Bio-IT/ALMF Course

Image Analysis with Python 2018

Sessions 3 – 5







Jonas Hartmann

Gilmour group, EMBL Heidelberg

Agenda

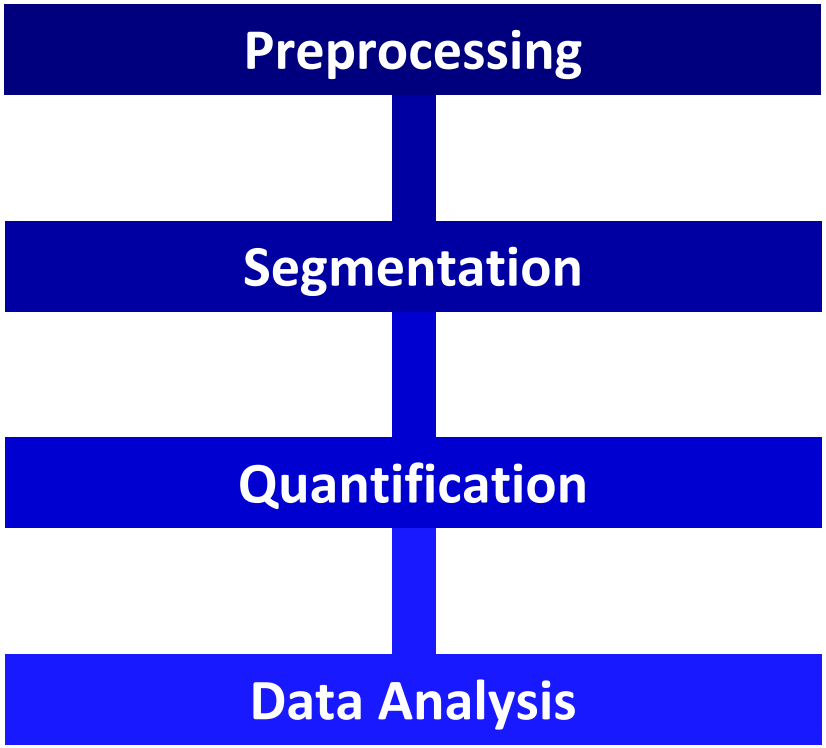
- Intro
- Image filters & convolution (?)
- Intro to tutorial
- You: work on tutorial

Agenda

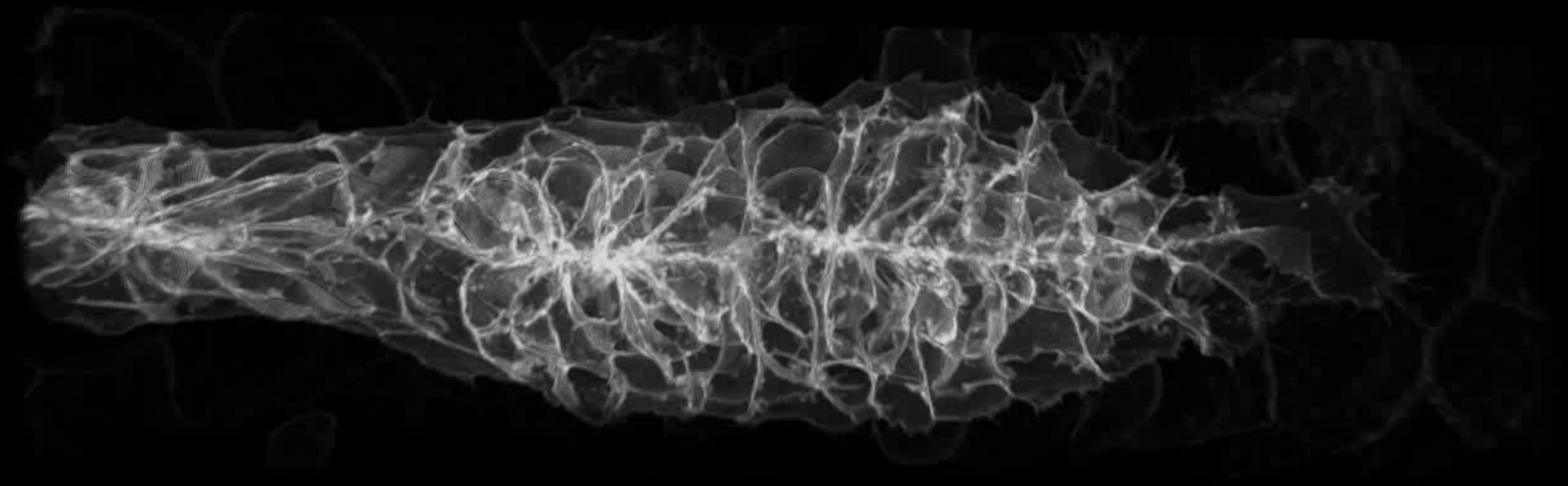
- Intro
- Image filters & convolution (?)
- Intro to tutorial
- You: work on tutorial
- You: work on tutorial
- You: work on tutorial
- You: work on tutorial
- You: work on tutorial
- You: work on tutorial
- You: work on tutorial

Intro

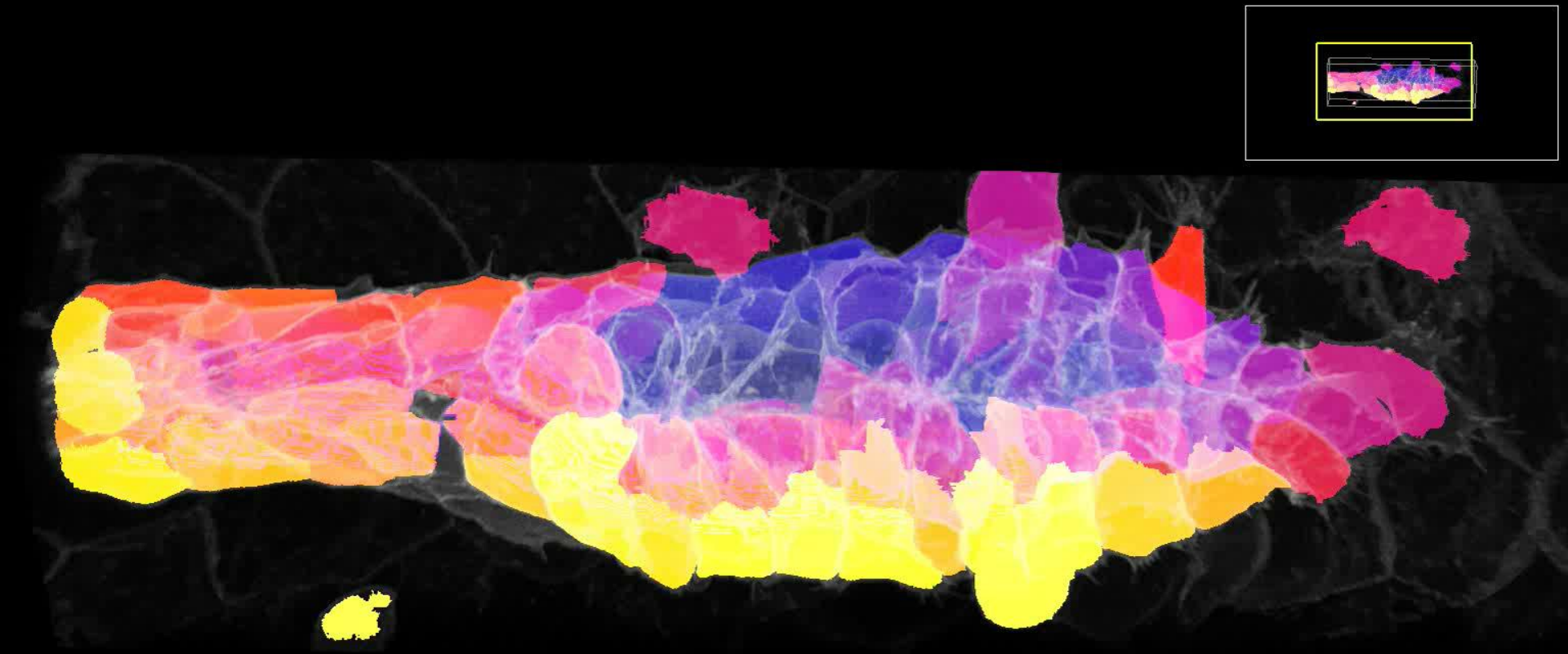
- **A typical image analysis workflow**



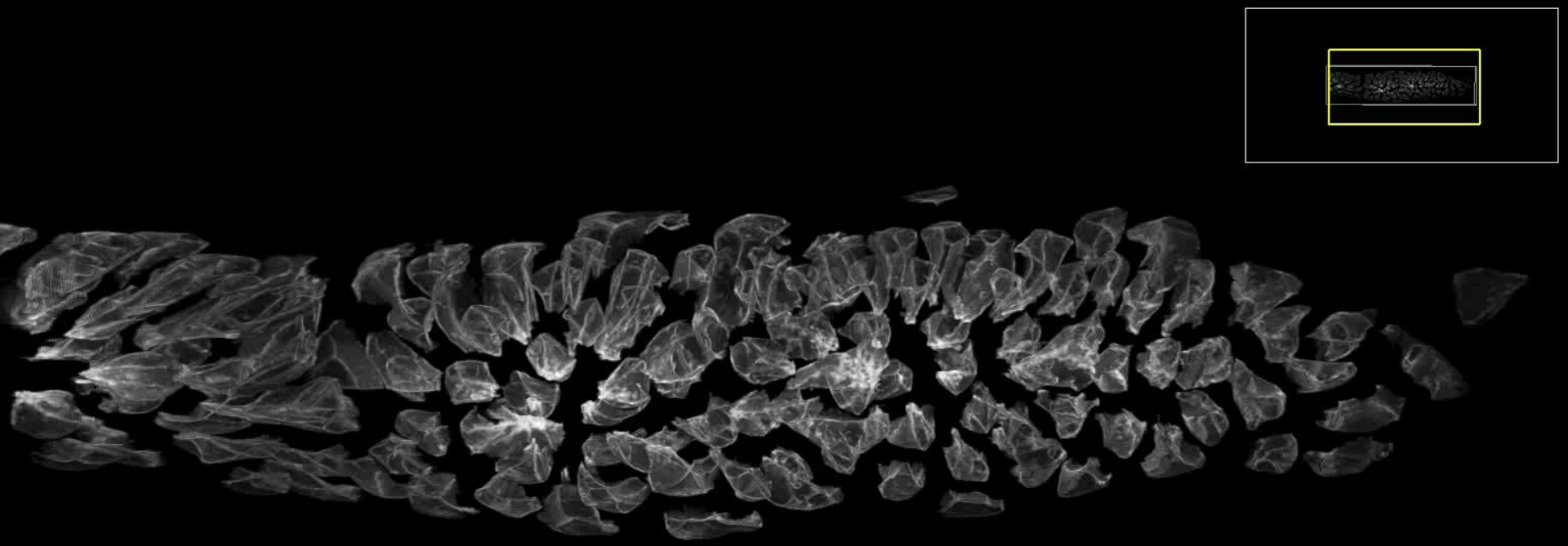
Intro



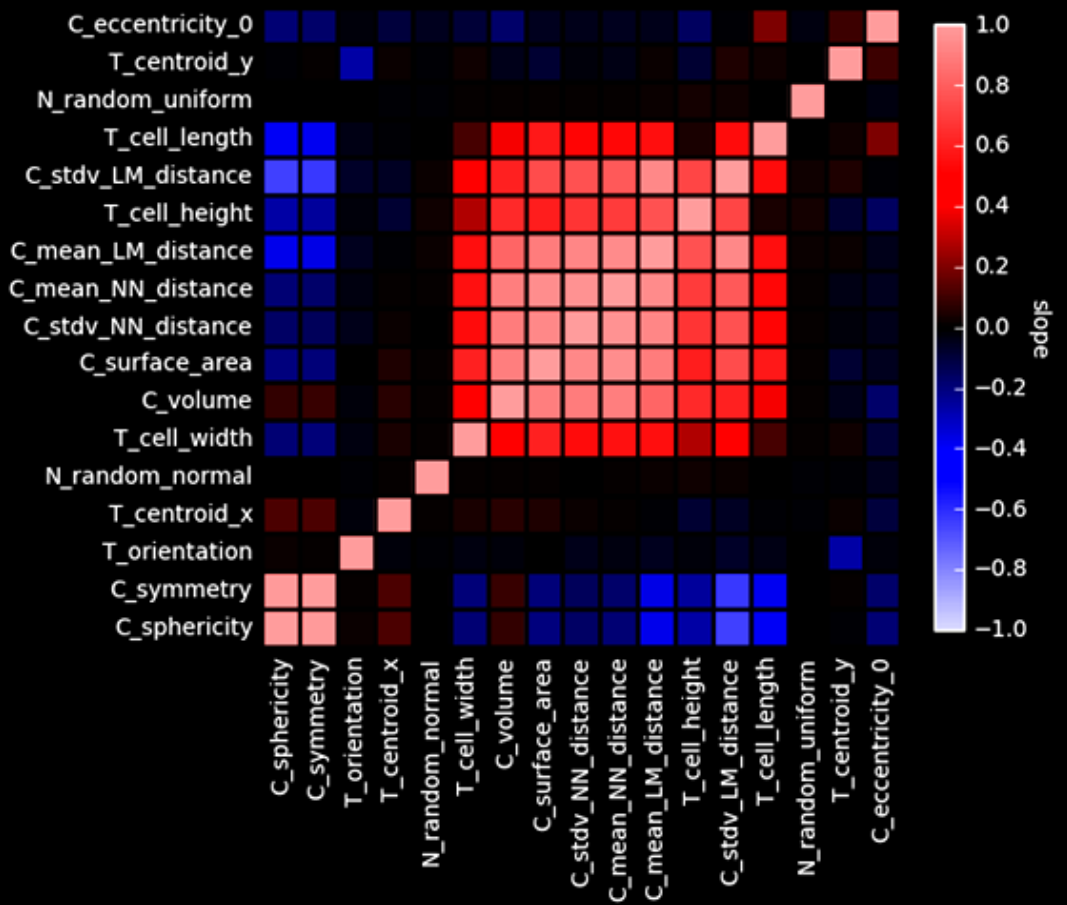
Intro



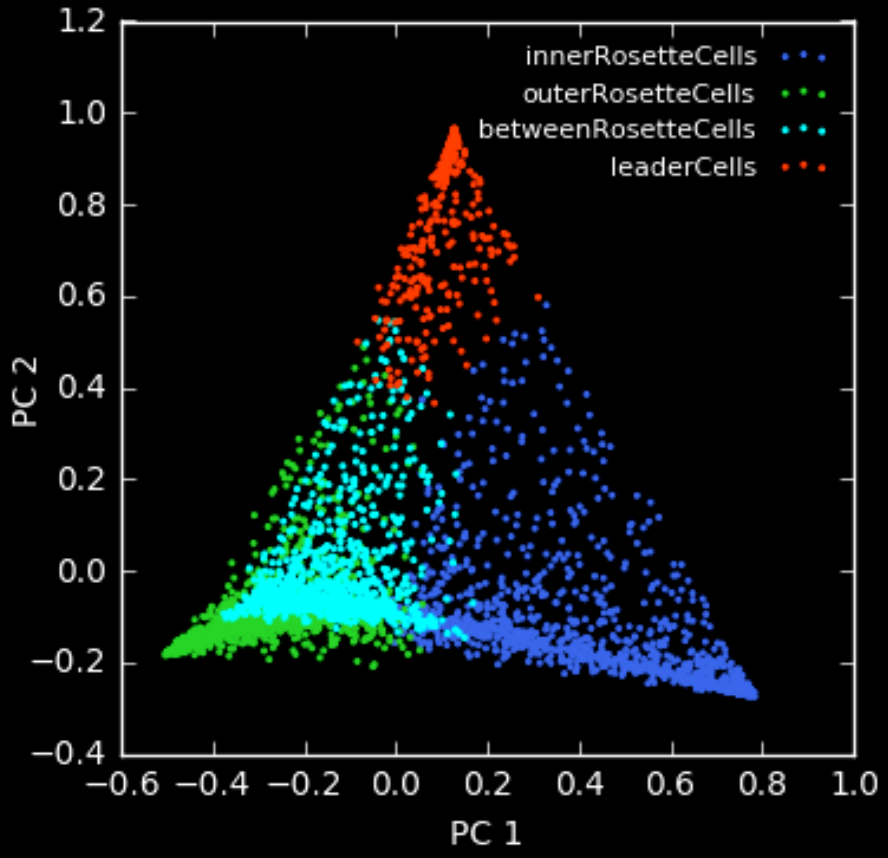
Intro

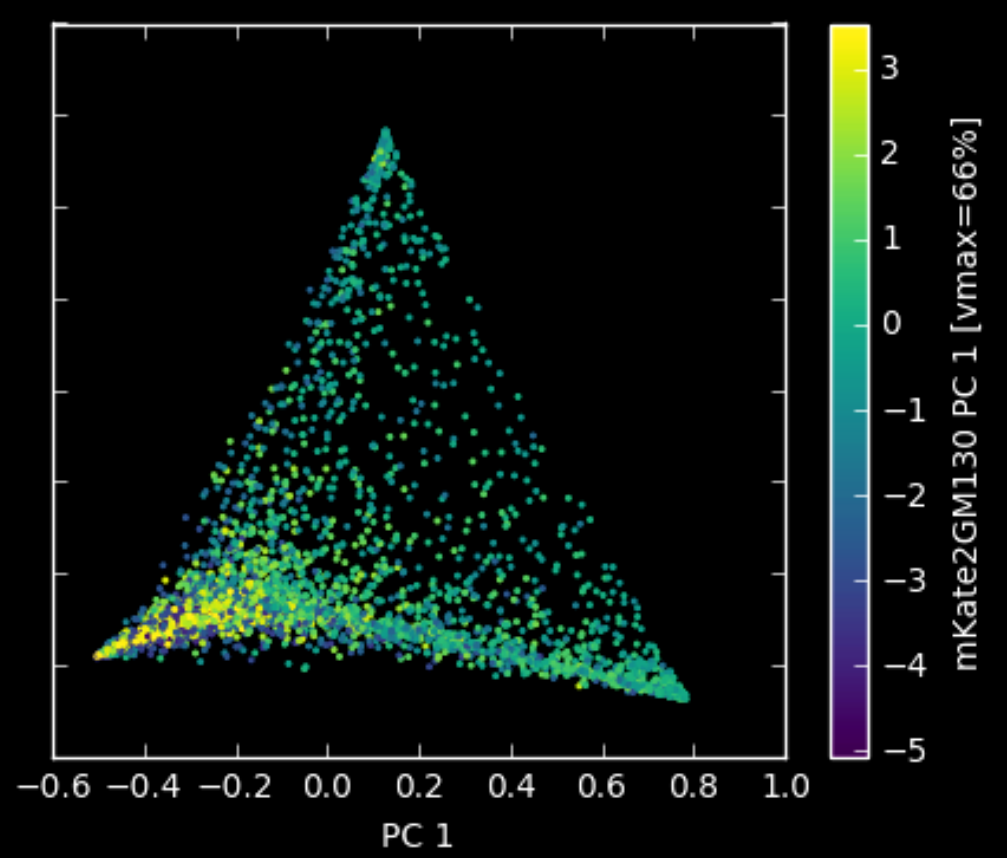


Intro


C_eccentricity_0
T_centroid_y
N_random_uniform
T_cell_length
C_stdv_LM_distance
T_cell_height
C_mean_LM_distance
C_mean_NN_distance
C_stdv_NN_distance
C_surface_area
C_volume
T_cell_width
N_random_normal
T_centroid_x
T_orientation
C_symmetry
C_sphericity
C_sphericity
C_symmetry
T_orientation
T_centroid_x
N_random_normal
T_cell_width
C_volume
C_surface_area
C_stdv_NN_distance
C_mean_NN_distance
C_mean_LM_distance
T_cell_height
C_stdv_LM_distance
T_cell_length
N_random_uniform
T_centroid_y
C_eccentricity_0
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
slope

Intro


innerRosetteCells
outerRosetteCells
betweenRosetteCells
leaderCells
PC 2
PC 1
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6 −0.4 −0.2 0.0 0.2 0.4 0.6 0.8 1.0


PC 1
−0.6 −0.4 −0.2 0.0 0.2 0.4 0.6 0.8 1.0
3
2
1
0
−1
−2
−3
−4
−5
mKate2GM130 PC 1 [vmax=66%]

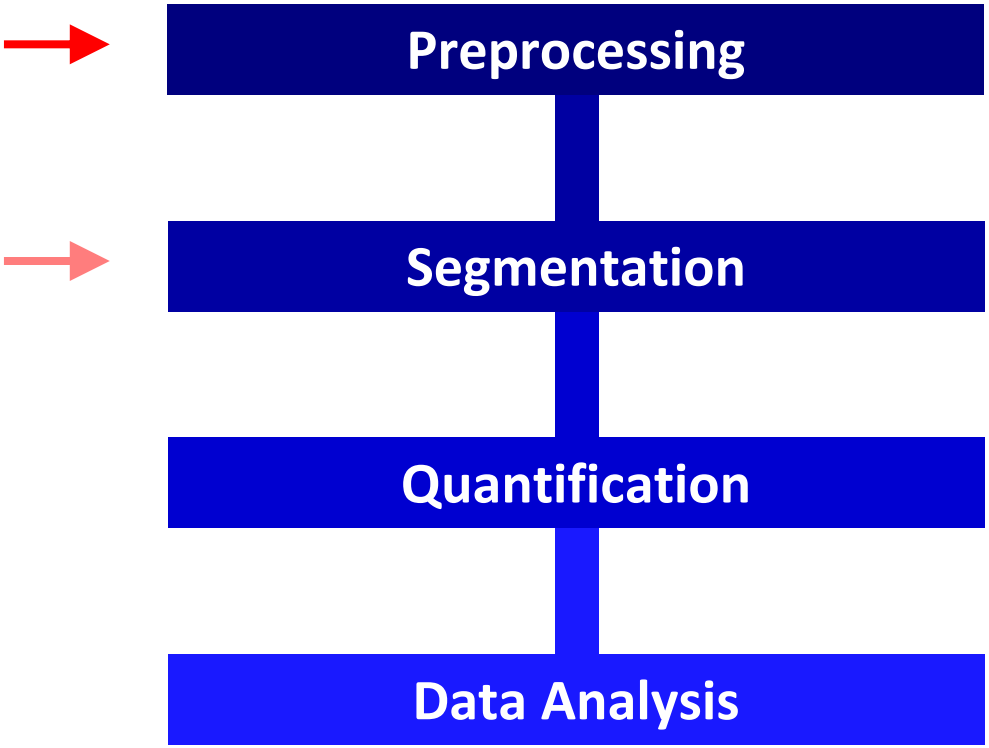
Goal of Sessions 3-5

Learning the basics of all this with a hands-on tutorial

But first...

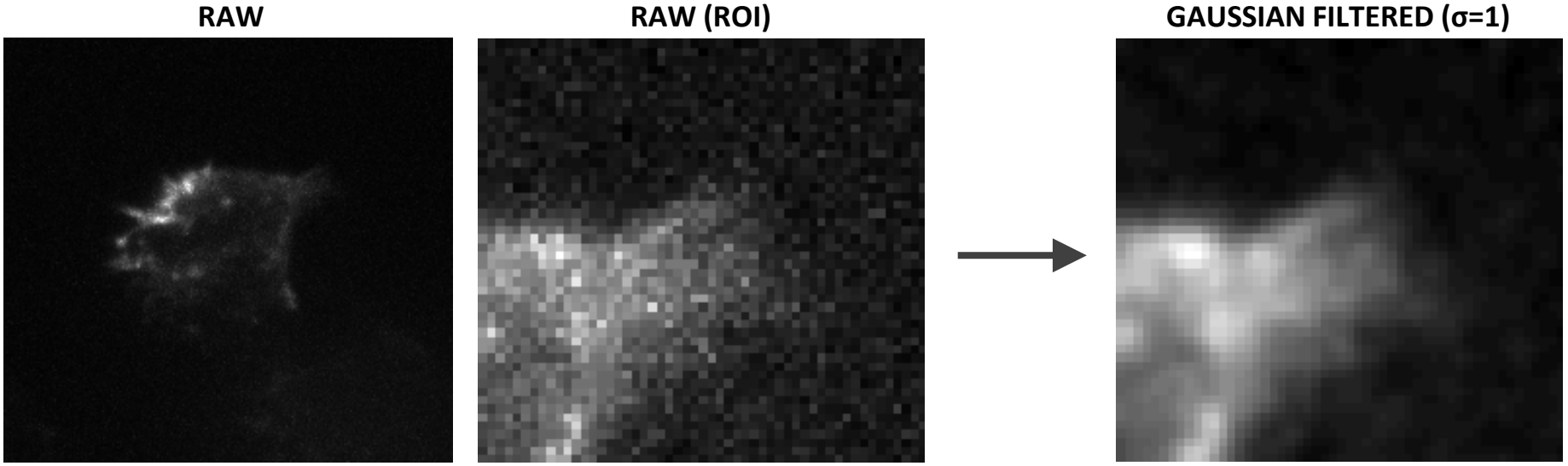
Filtering & Convolution

- A key tool for image analysis: convolutional filters



Filtering & Convolution

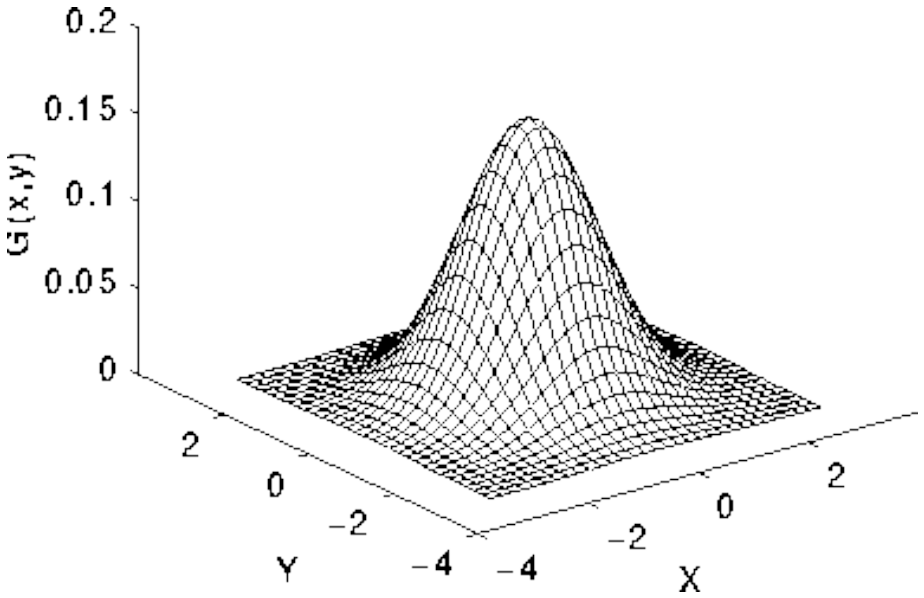
- Goal: removing noise whilst preserving or enhancing structure
- Common filters
  - Gaussian filter (smoothing, general noise reduction)
  - Median filter (removing shot noise)
  - LoG filter (dots), Sobel filter (edges)
- Example: Gaussian filter



Filtering & Convolution

- **How it works: kernels & convolution**

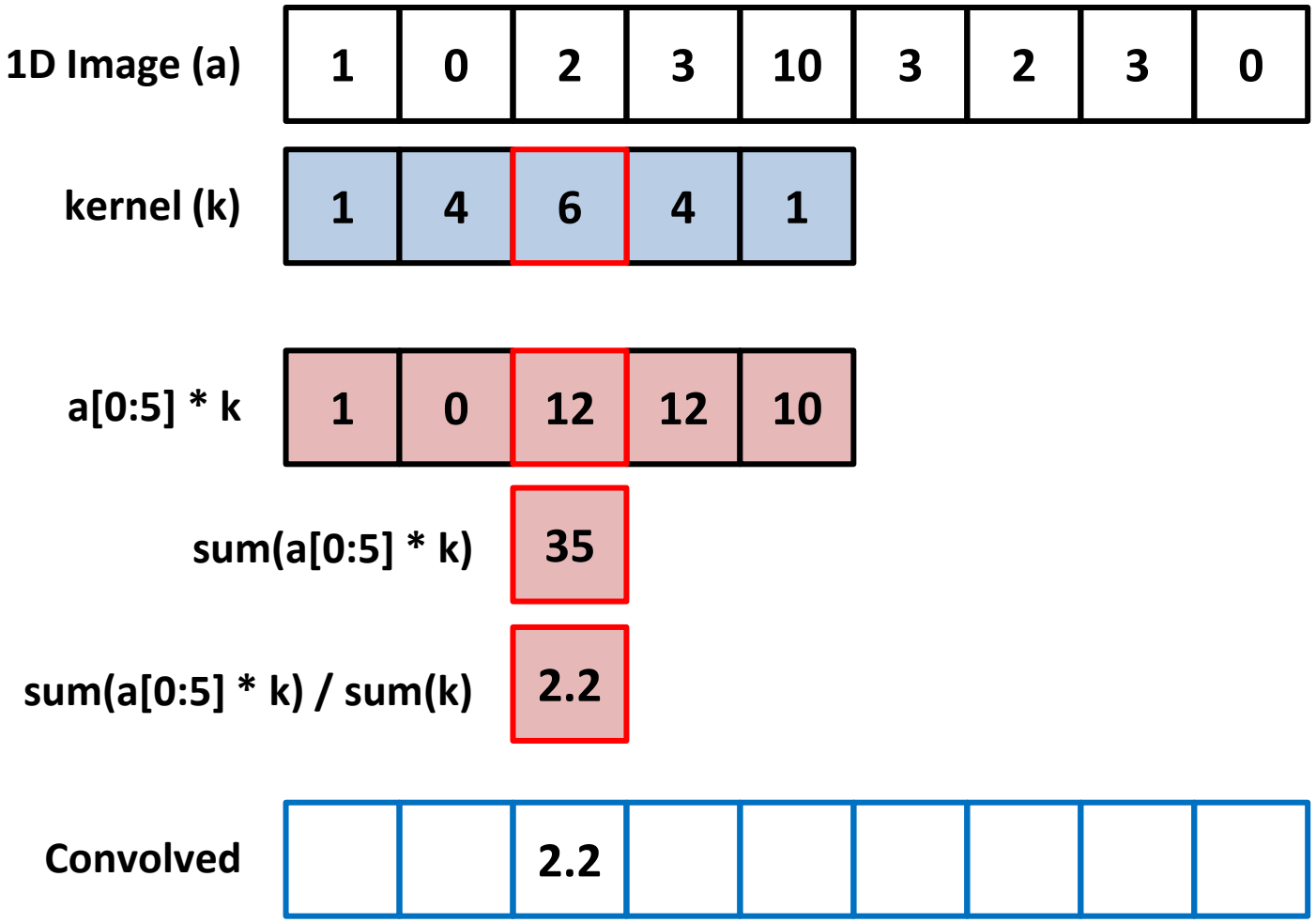
2D Gaussian kernel



0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	10	10	10	0	0	0	0
0	0	10	10	20	30	20	10	10	0	0
0	0	10	30	50	50	50	30	10	0	0
0	10	20	50	70	80	70	50	20	10	0
0	10	30	60	80	90	80	60	30	10	0
0	10	20	50	70	80	70	50	20	10	0
0	0	10	30	50	50	50	30	10	0	0
0	0	10	10	20	30	20	10	10	0	0
0	0	0	0	10	10	10	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

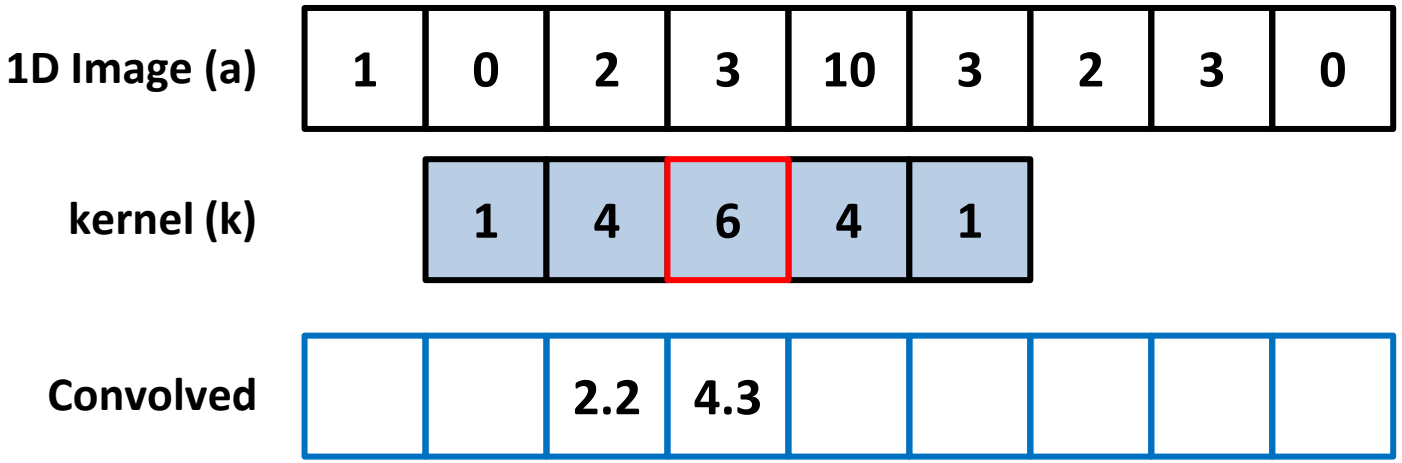
Filtering & Convolution

- How it works: kernels & convolution



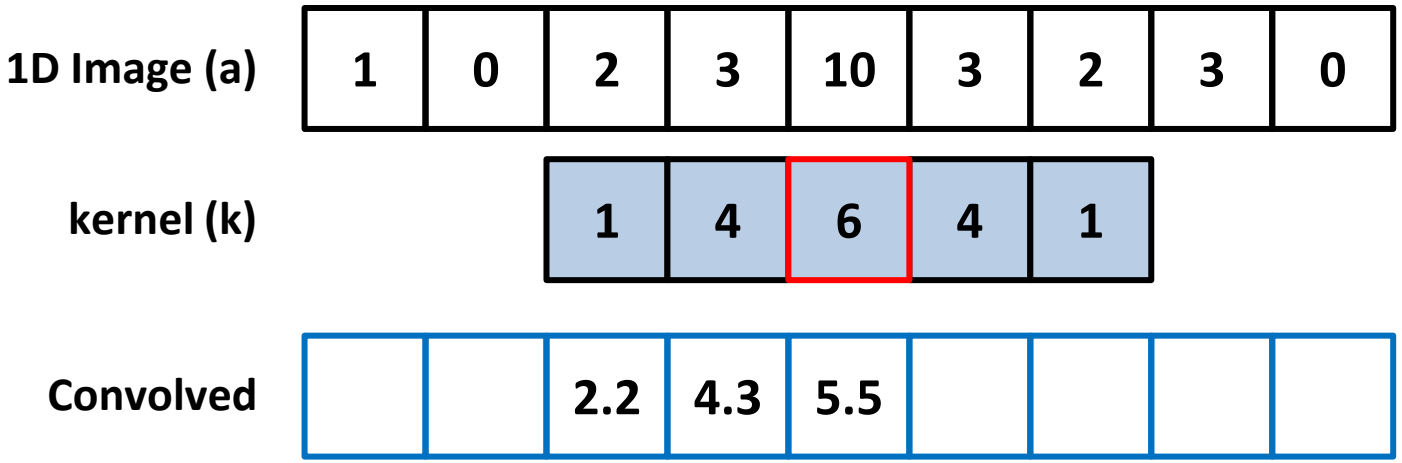
Filtering & Convolution

- How it works: kernels & convolution



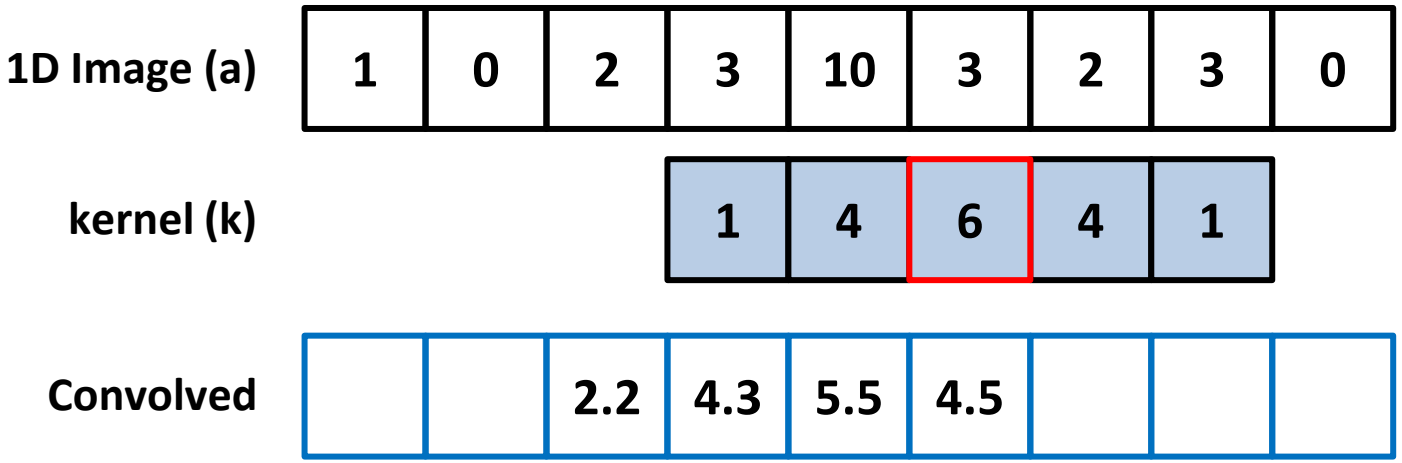
Filtering & Convolution

- How it works: kernels & convolution



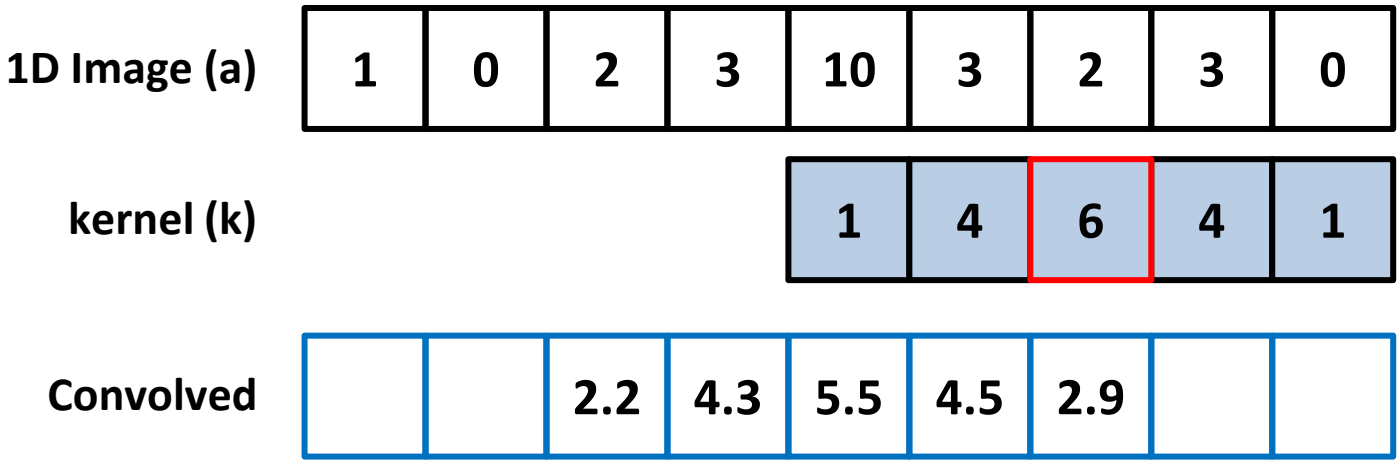
Filtering & Convolution

- How it works: kernels & convolution



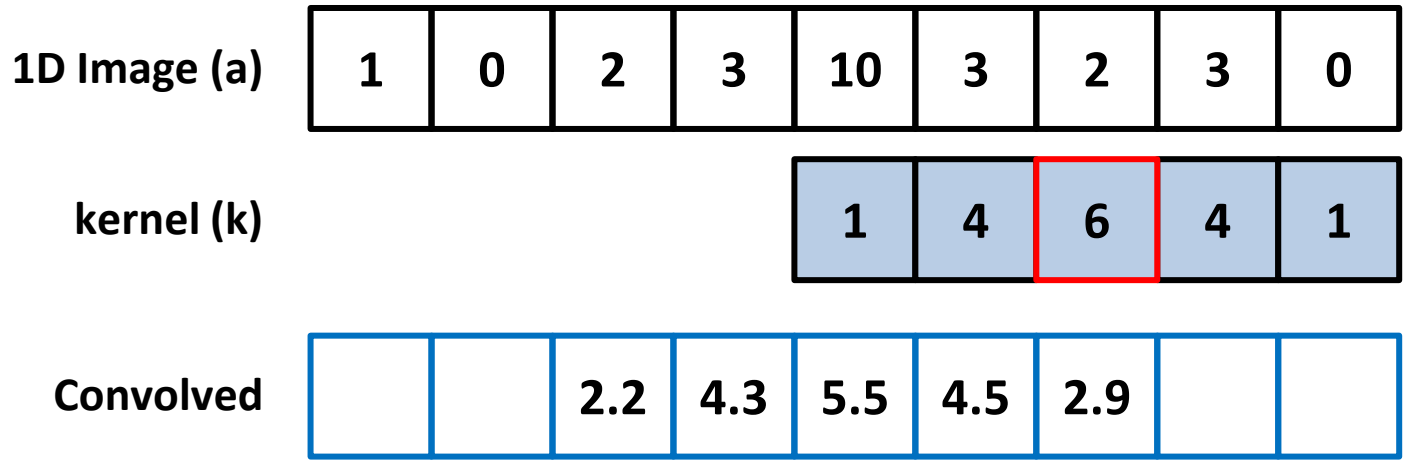
Filtering & Convolution

- How it works: kernels & convolution



Filtering & Convolution

- How it works: kernels & convolution

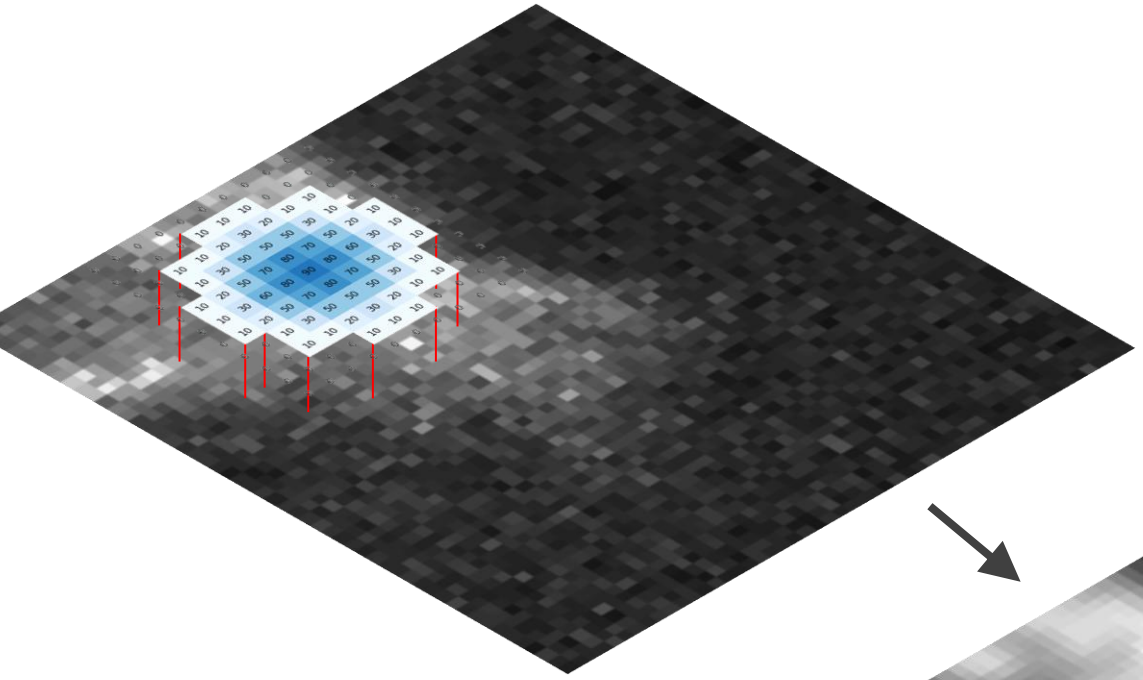


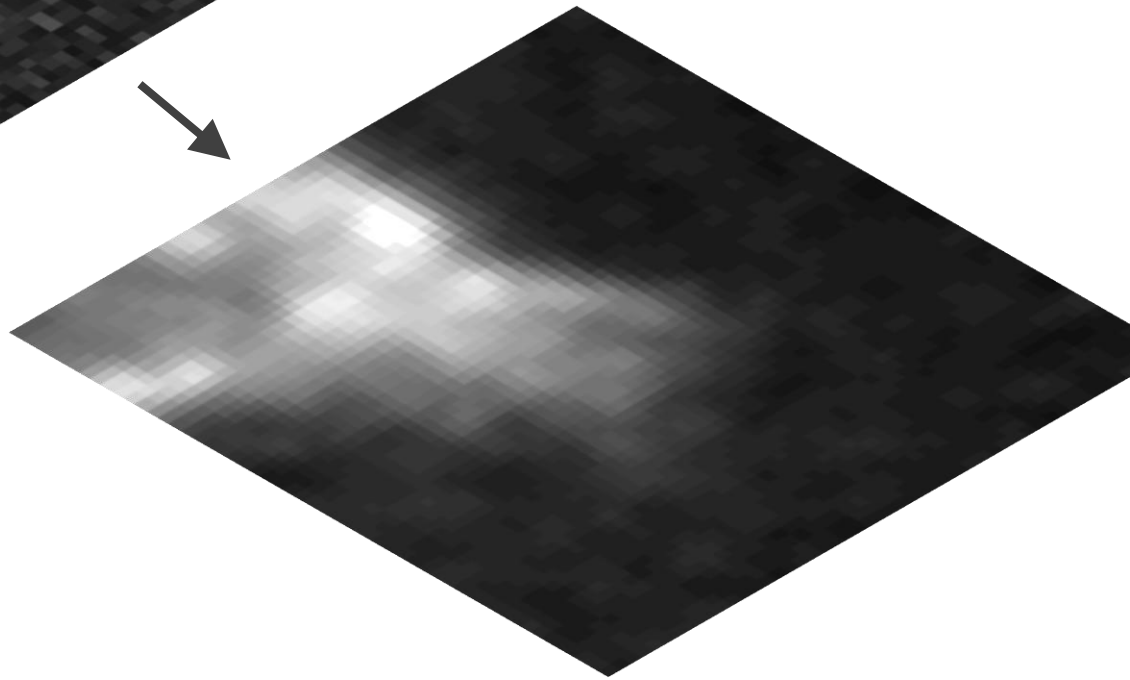
Note: Behavior at edges is undefined. Default in *scipy* is `reflect`.

Filtering & Convolution

- How it works: kernels & convolution

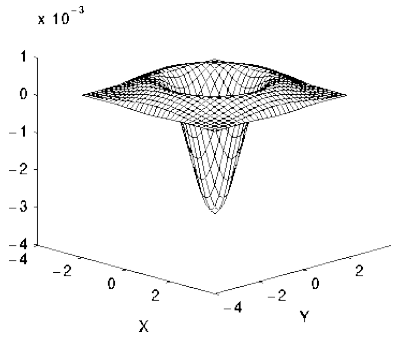
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	10	10	10	0	0	0	0
0	0	10	10	20	30	20	10	10	0	0
0	0	10	30	50	50	50	30	10	0	0
0	10	20	50	70	80	70	50	20	10	0
0	10	30	60	80	90	80	60	30	10	0
0	10	20	50	70	80	70	50	20	10	0
0	0	10	30	50	50	50	30	10	0	0
0	0	10	10	20	30	20	10	10	0	0
0	0	0	0	10	10	10	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0





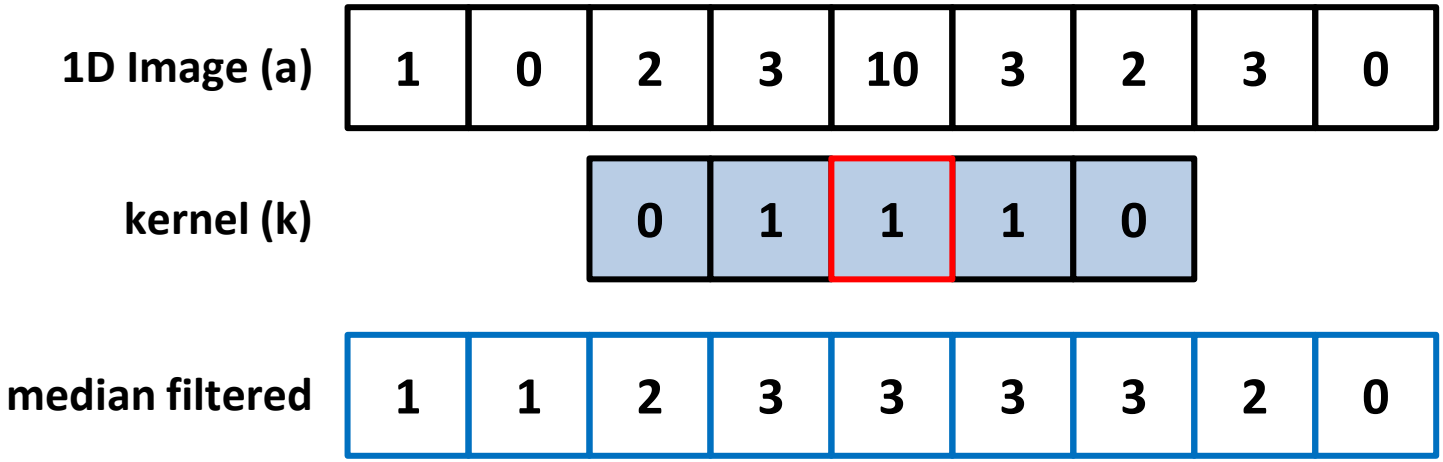
Filtering & Convolution

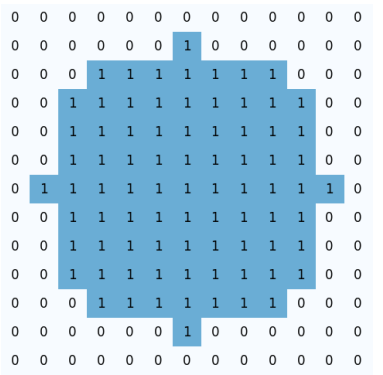
- Other filters have different kernels
  - e.g. LoG filter



0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

- Or they perform different operations
  - e.g. median filter

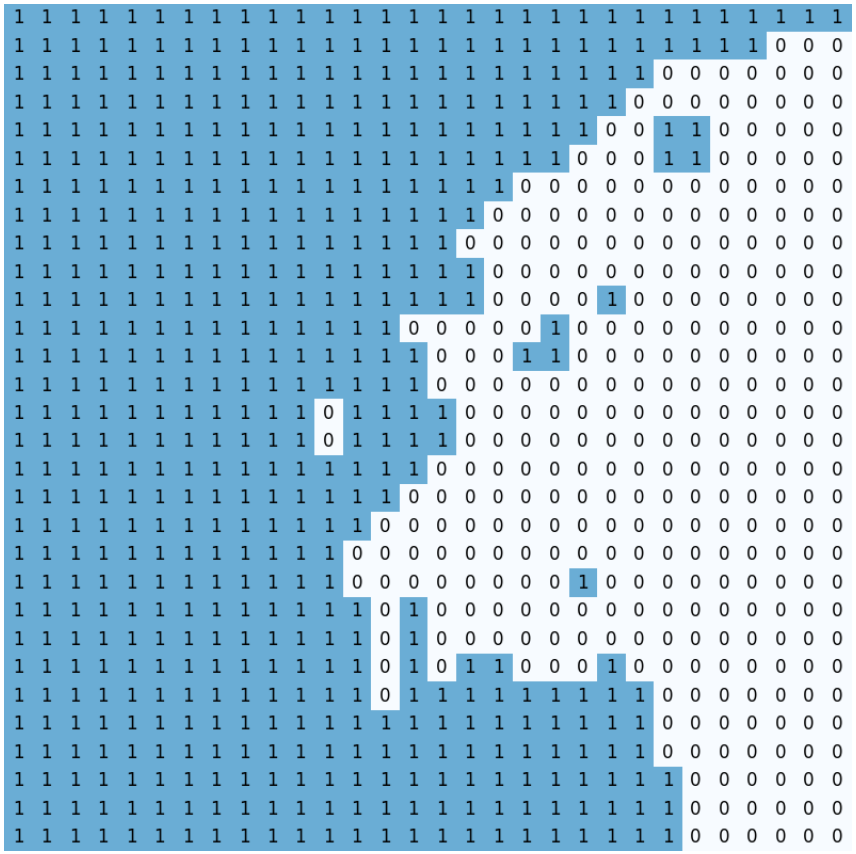




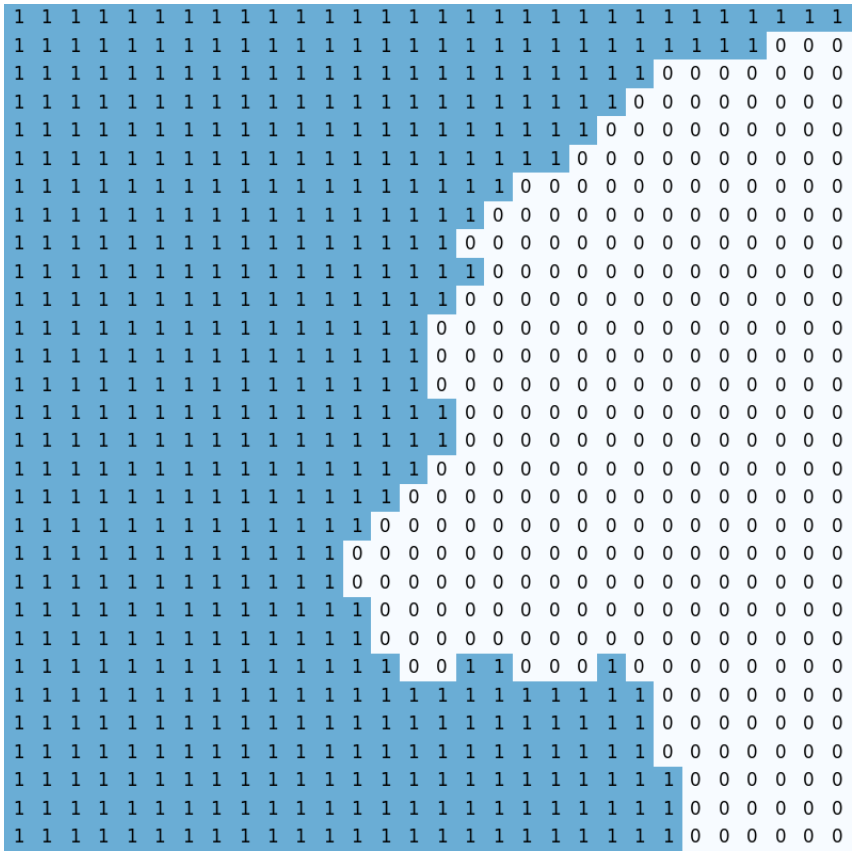
Filtering & Convolution

- Filters can also be used to improve/correct binary masks
- This is referred to as ‘morphological operations’

THRESHOLDED

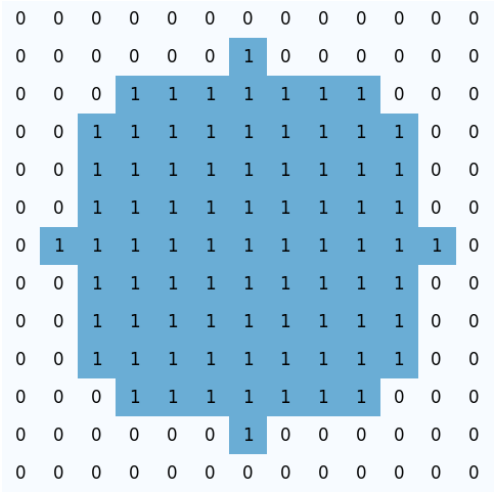


MORPHOLOGICALLY PROCESSED



Filtering & Convolution

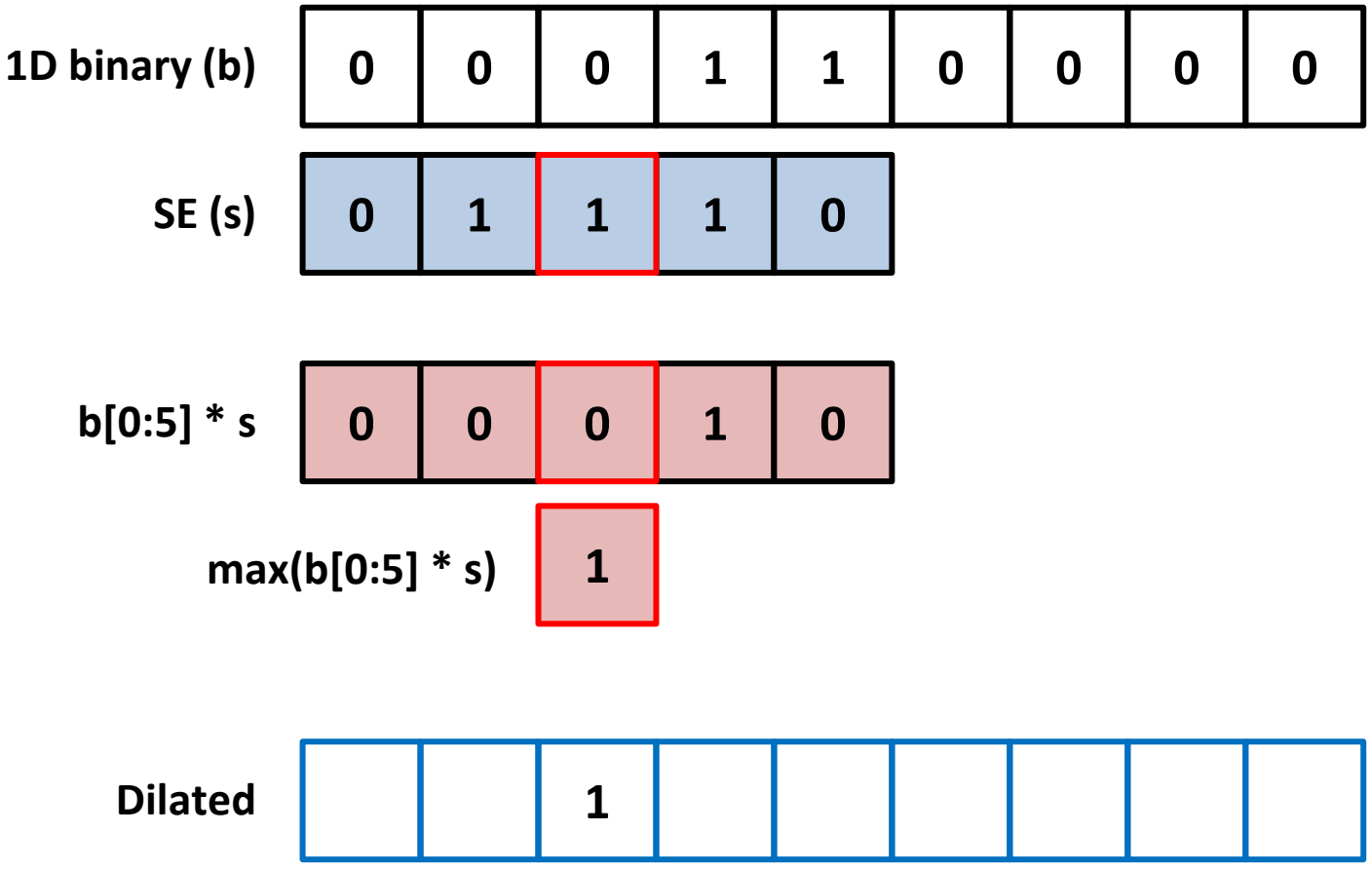
- **Common morphological operations**
  - **Erosion & Dilation**
  - **Opening & Closing**
  - **Hole filling**
- **Principle very much the same as in filtering**
  - **Use of a `structural element` (SE); basically the same as a kernel**



DISC-SHAPED SE

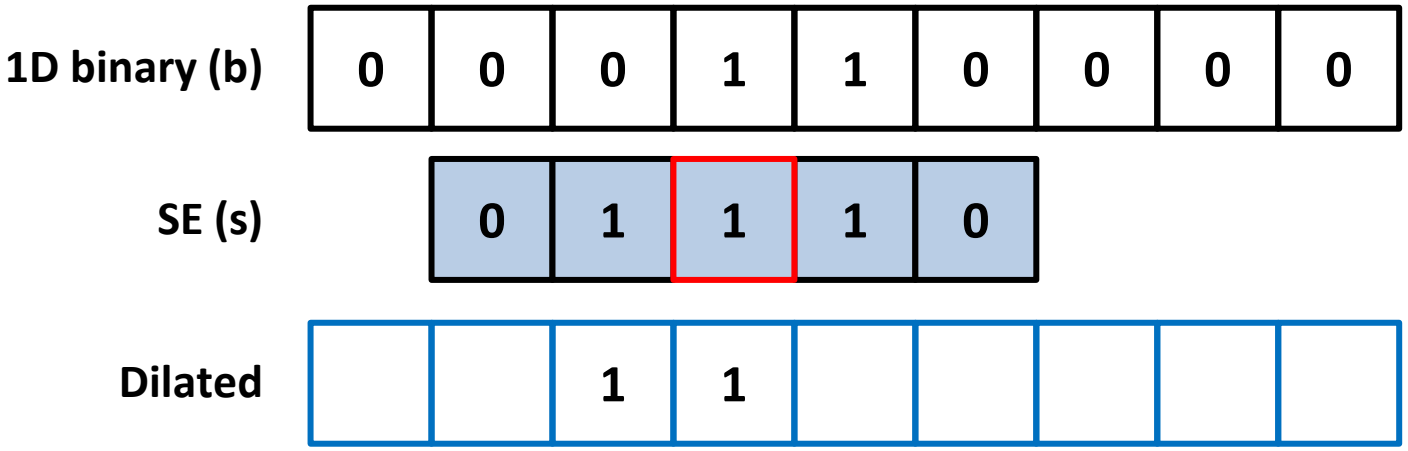
Filtering & Convolution

- **Dilation: expanding masks**



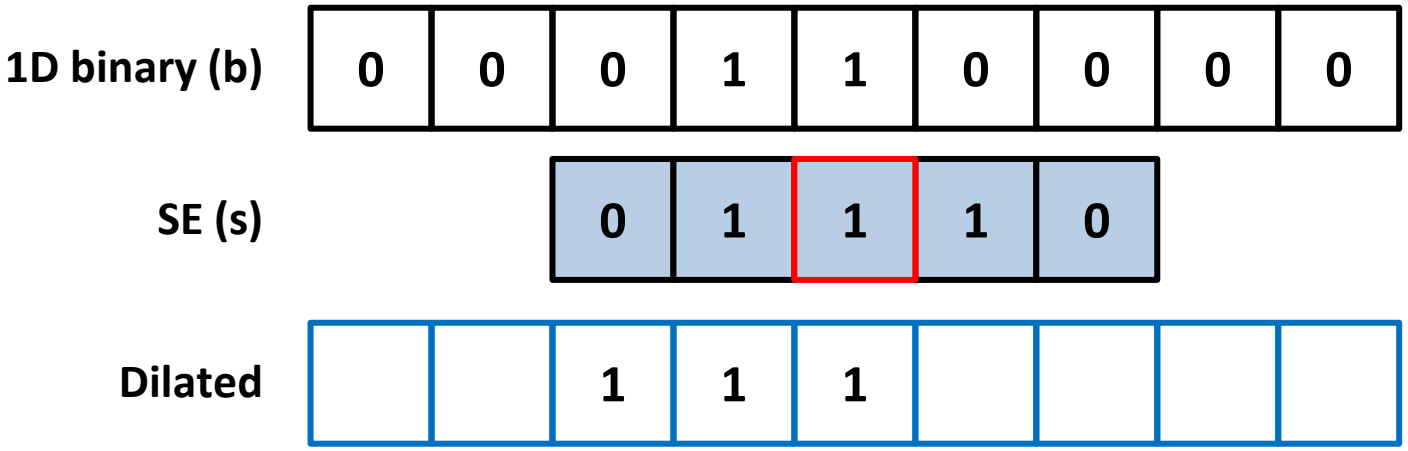
Filtering & Convolution

- **Dilation: expanding masks**



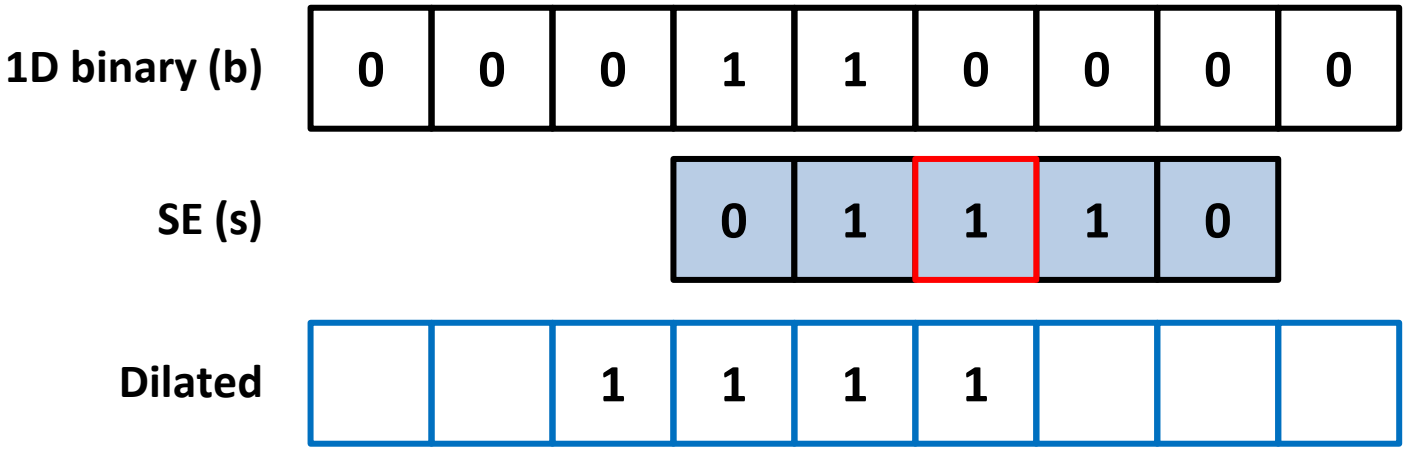
Filtering & Convolution

- **Dilation: expanding masks**



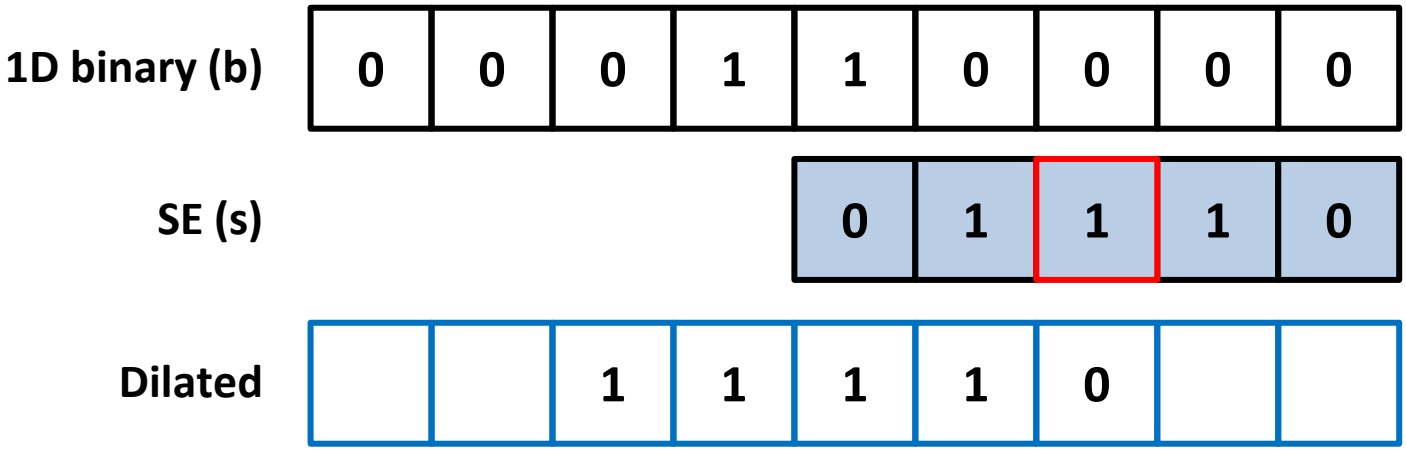
Filtering & Convolution

- **Dilation: expanding masks**



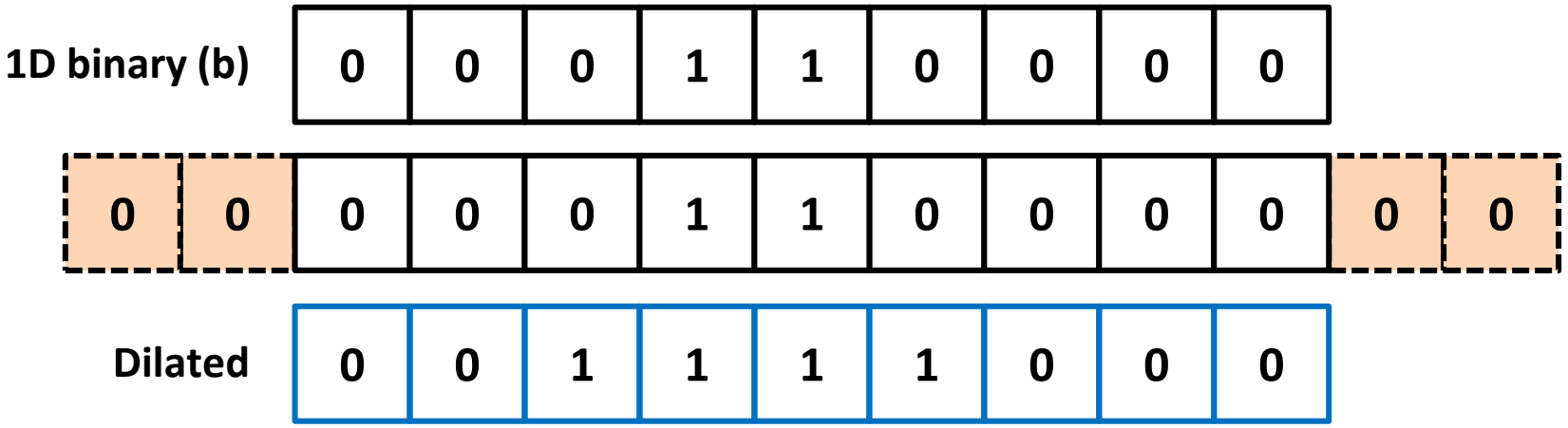
Filtering & Convolution

- **Dilation: expanding masks**



Filtering & Convolution

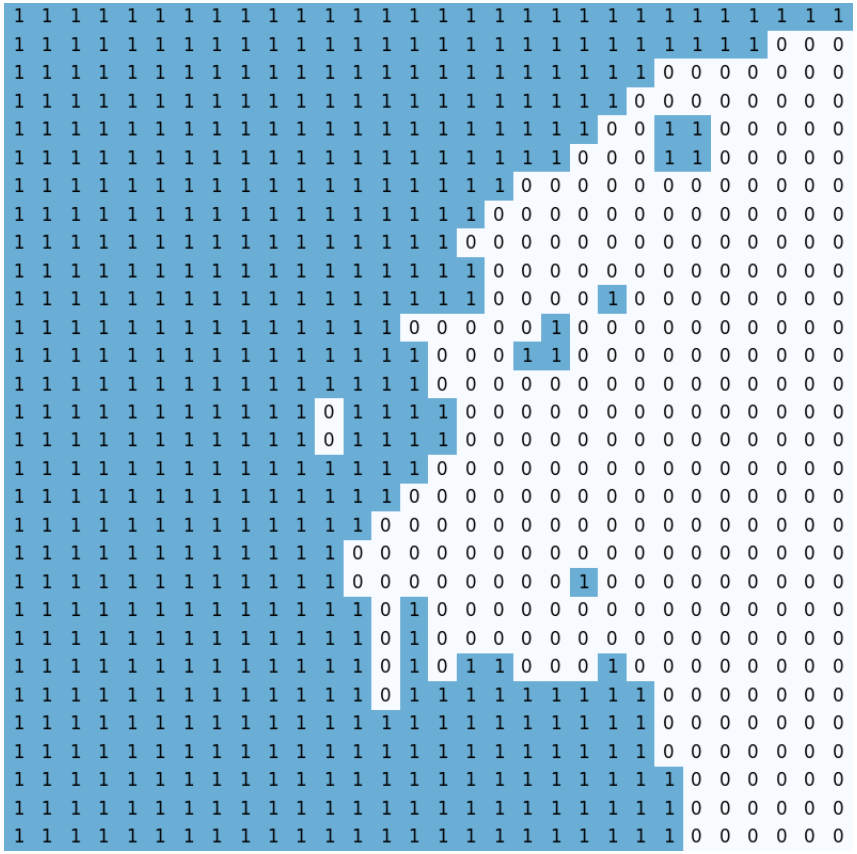
- **Dilation: expanding masks**



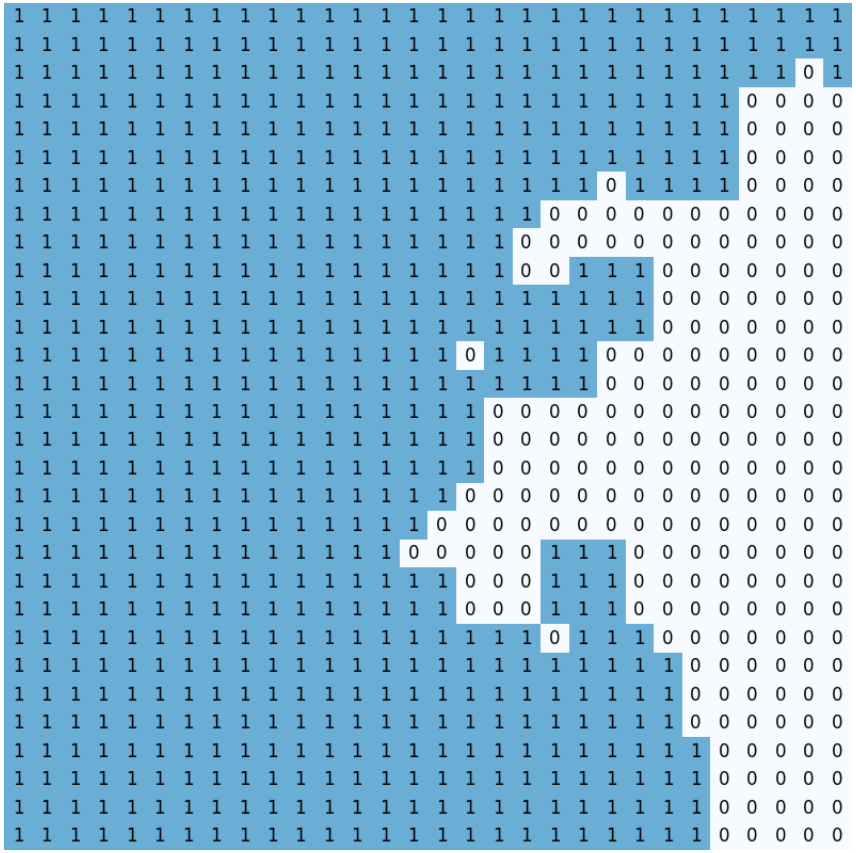
Filtering & Convolution

- **Dilation: expanding masks**

THRESHOLDED



DILATED (SE np.ones((3,3)))



Foreground Detection: Morphological Operations

- **Common morphological operations**

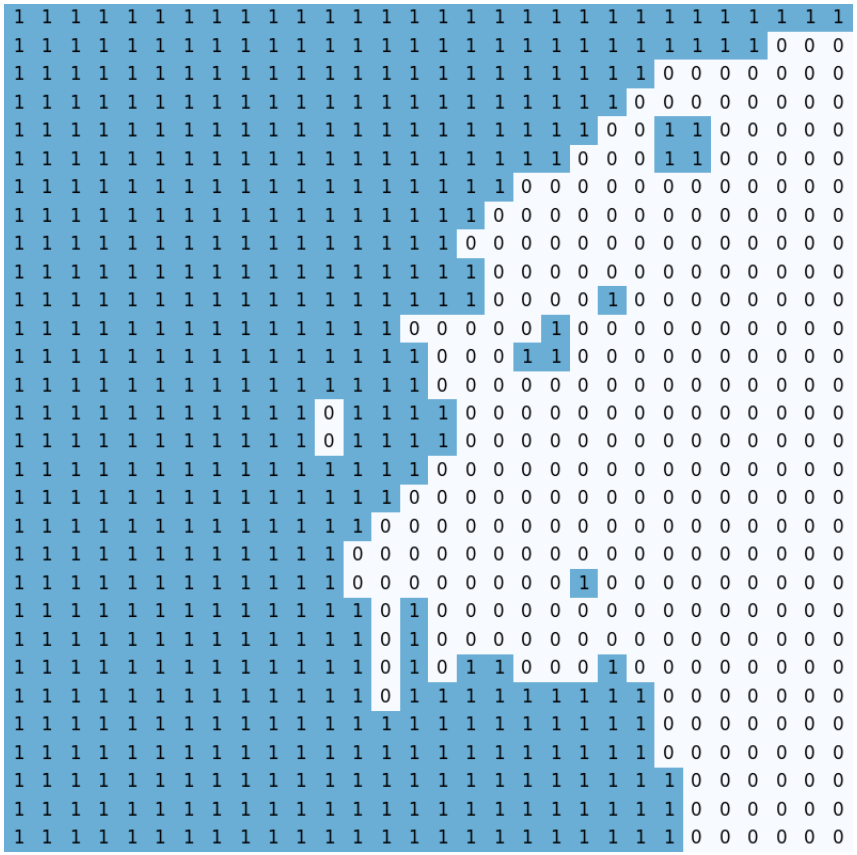
```
Dilation:      maxConv(b,s)
Erosion:       minConv(b,s)
Closing:       dilation(erosion(b,s))
Opening:       erosion(dilation(b,s))
Hole filling:  [more complicated]
```

- **Some notes**
  - Closing and opening more or less preserve mask area
  - The shape of the SE matters (disc-shapes are usually preferred)
  - Combine morphological operations to get the desired effect

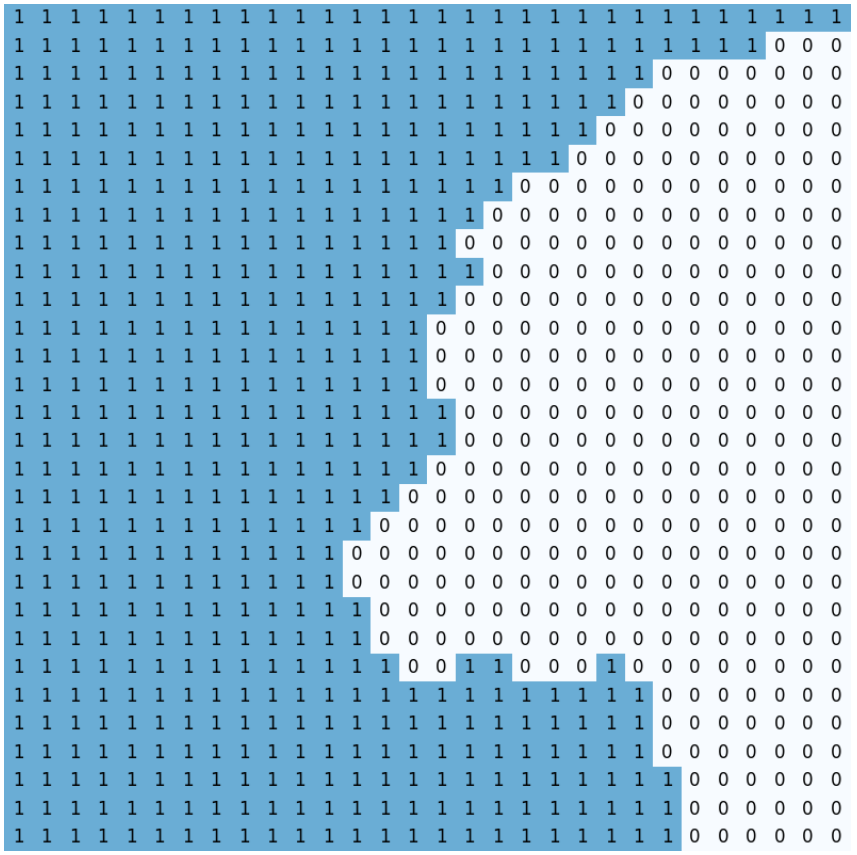
Filtering & Convolution

- **Combine morphological operations to get the desired effect**

THRESHOLDED



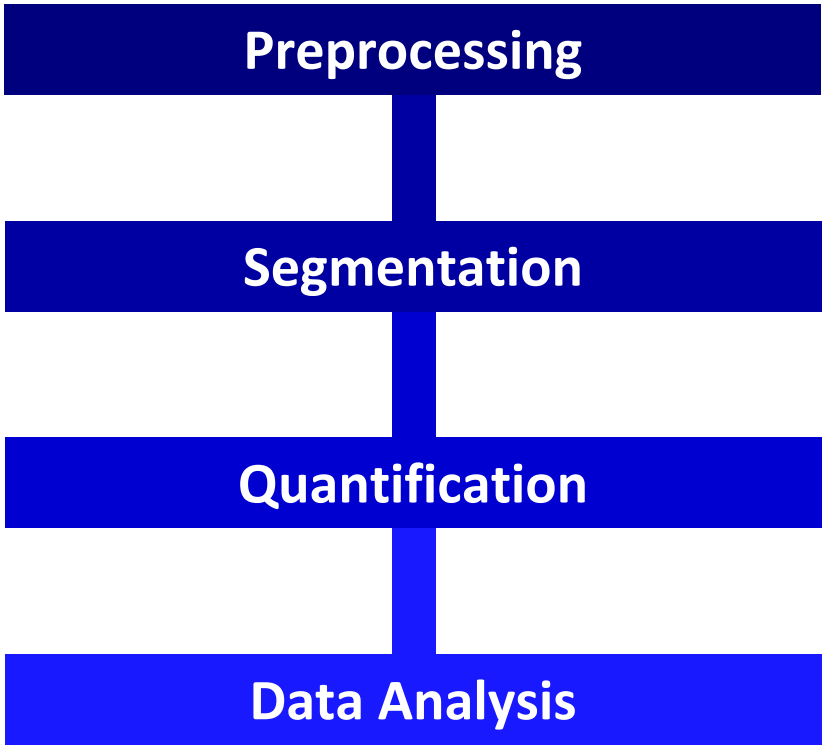
CLOSING(OPENING(THRESHOLDED))



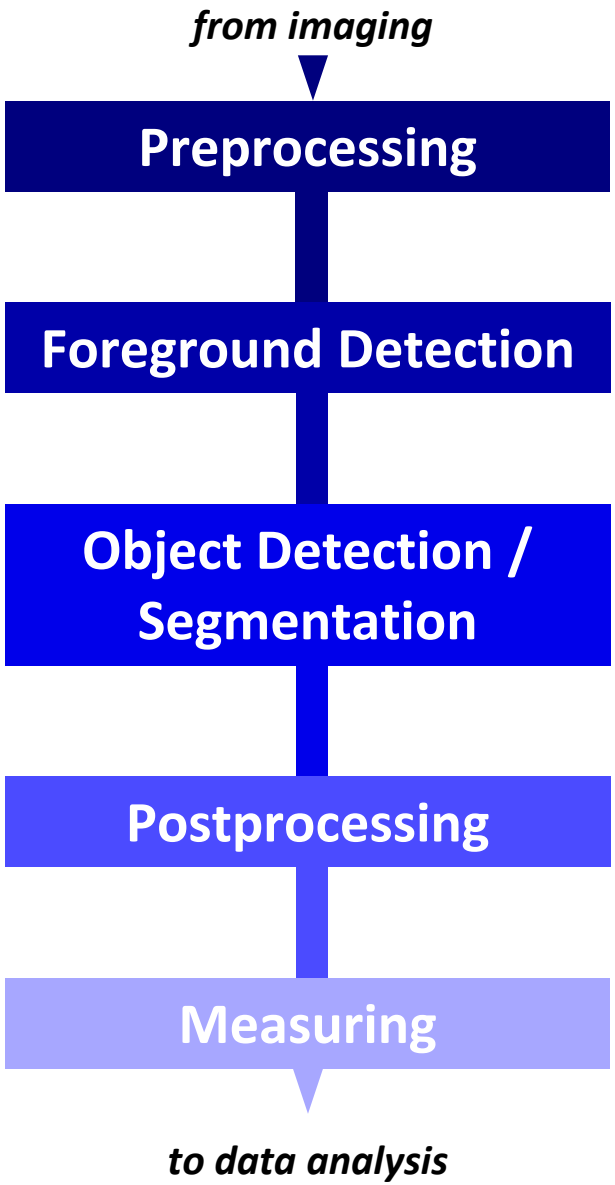
It‘s time to get to work!

Tutorial Pipeline

- **A typical image analysis workflow**



Tutorial Pipeline: Outline



Import to numpy array

Gaussian smoothing

(Adaptive) thresholding

Morphological clean-up

Seeding with distance transform

Watershed

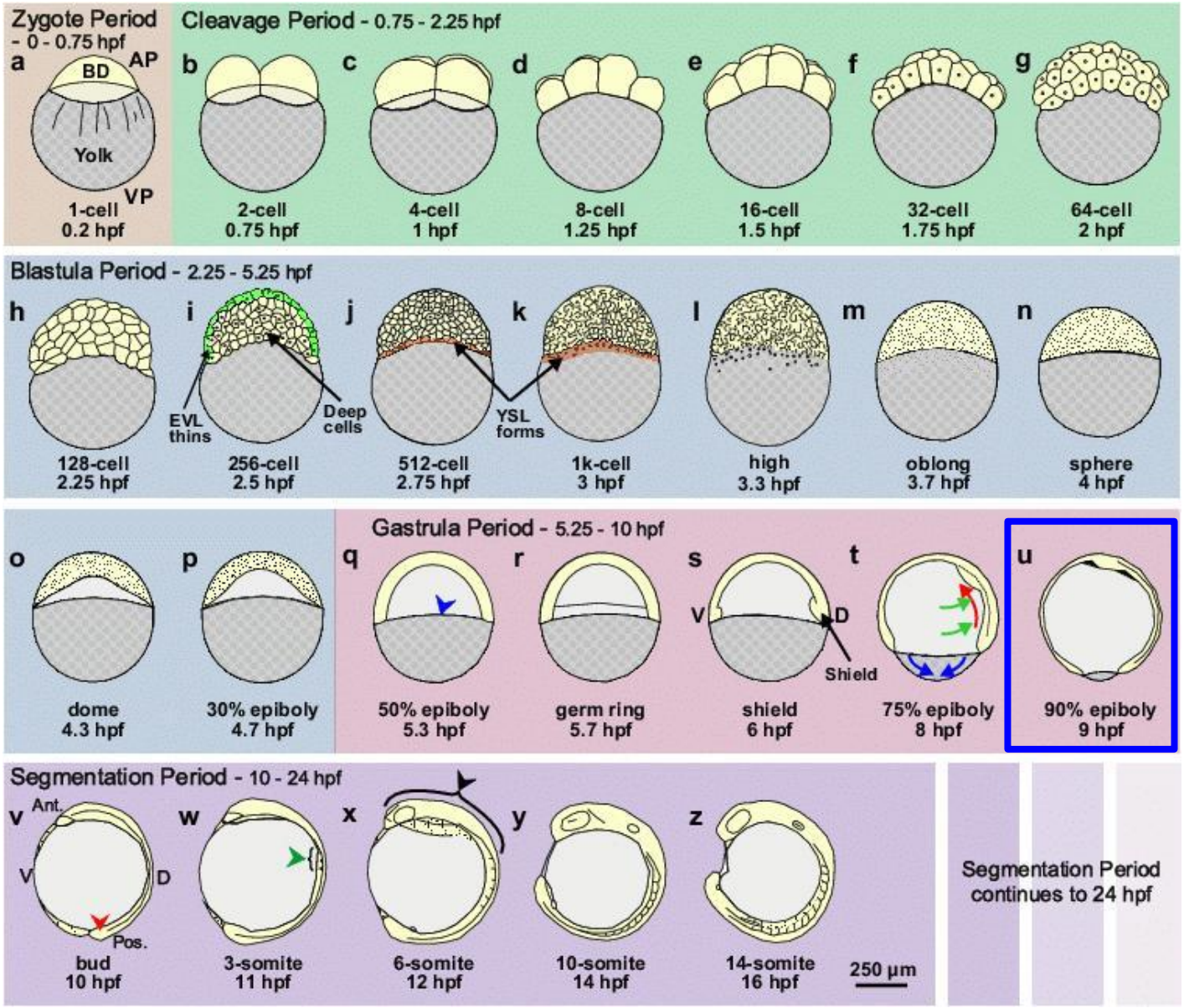
Filter objects at image borders

Basic intensity & shape measurements

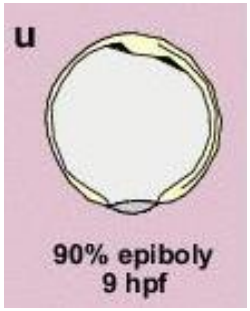
Simple analysis and visualization

Saving of data and figures

Tutorial Pipeline: Sample Images

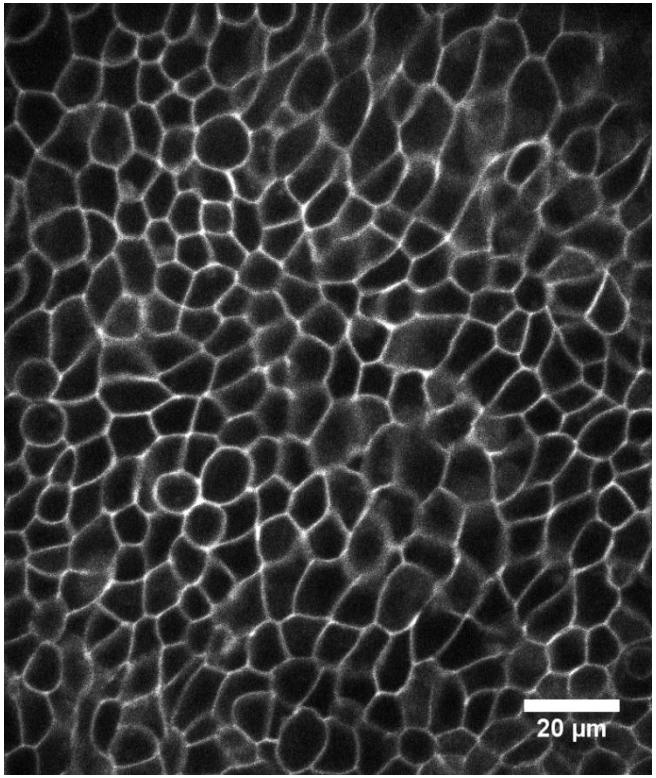


Tutorial Pipeline: Sample Images



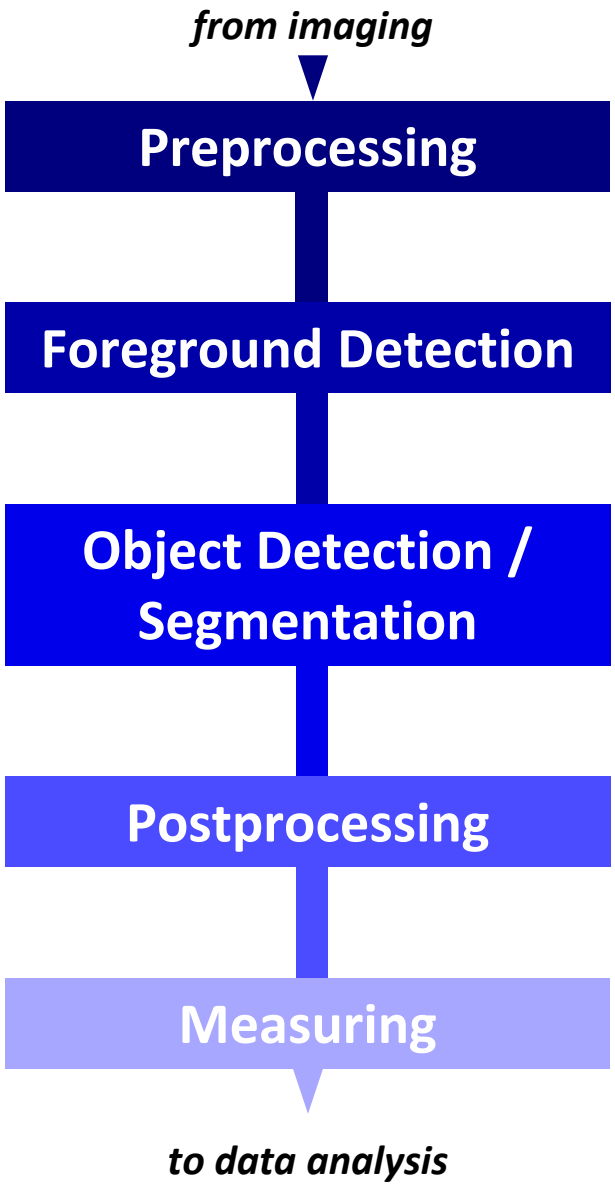
- Early zebrafish embryo
- An "*in vivo* cell culture"
- Observable: division, migration, differentiation, morphogenesis

- 40X spinning-disk confocal slice
- Label: *mNG:Gγ9 (a G-protein)*
- "Real-world data"



Good Luck! ;p

Tutorial Pipeline: Outline



Import to numpy array

Gaussian smoothing

(Adaptive) thresholding

Morphological clean-up

Seeding with distance transform

Watershed

Filter objects at image borders

Basic intensity & shape measurements

Simple analysis and visualization

Saving of data and figures