



DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Research Software Management

Robert Haase

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



Diese Maßnahme wird gefördert durch die Bundesregierung
aufgrund eines Beschlusses des Deutschen Bundestages.
Diese Maßnahme wird mitfinanziert durch Steuermittel auf
der Grundlage des von den Abgeordneten des Sächsischen
Landtags beschlossenen Haushaltes.

Quiz (recap)

- Which of these four is no FAIR principle

Findable



Accessible



Inoperable



Reusable



Quiz (recap)

- If I combine two works licensed CC-BY-SA and CC-BY-ND, what license do I have to use?

CC-BY-ND-SA



CC-BY-SA



CC-BY-ND



(not possible)



Research Software Management

- Counterpart to Research Data Management
 - Relatively new term.
- Special:
 - Version control: git
 - Software environments
conda/python

scientific **data**

Explore content ▾ About the journal ▾ Publish with us ▾

[nature](#) > [scientific data](#) > [articles](#) > article

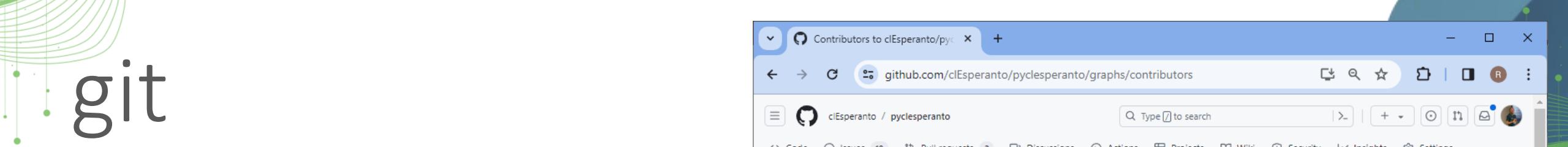
Article | [Open access](#) | Published: 14 October 2022

Introducing the FAIR Principles for research software

Michelle Barker  , Neil P. Chue Hong, Daniel S. Katz, Anna-Lena Lamprecht, Carlos Martinez-Ortiz, Fotis Psomopoulos, Jennifer Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez & Tom Honeyman

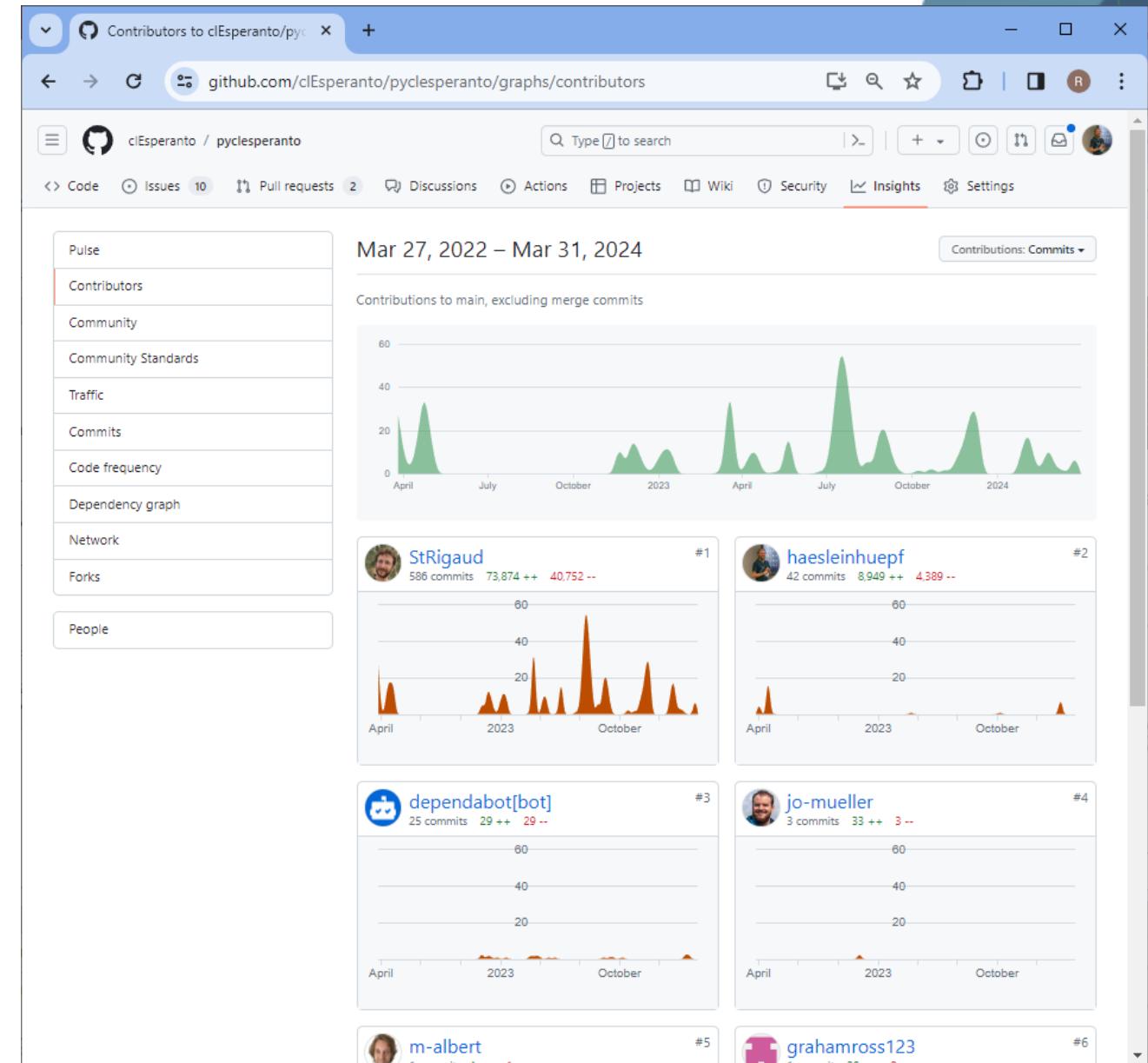
[Scientific Data](#) 9, Article number: 622 (2022) | [Cite this article](#)

20k Accesses | 60 Citations | 230 Altmetric | [Metrics](#)



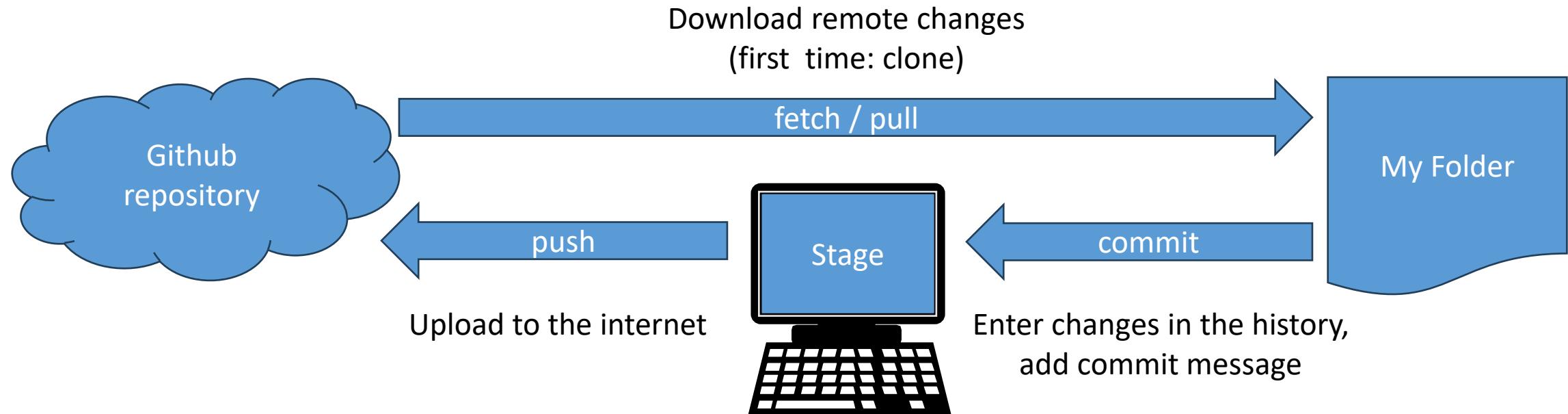
git

- Version control is key element of data scientist's toolbox
- Distributed file system with sophisticated logging mechanisms
- Control about what becomes part of a repository and what not



git

- Git makes file modifications a more active / involved process (making people think about)



git

- Who wrote this code
- when and
- why?

The screenshot shows two browser tabs. The left tab displays the commit history for the repository `haesleinhuepf/example_image_analysis_script`. It lists several commits, with the commit titled "bugfix: threshold_otsu" highlighted by an orange arrow. The right tab shows a detailed view of the changes made in that commit, specifically the file `my_library.py`. The diff view highlights the addition of the `threshold_otsu` function with a green background.

```
diff --git a/my_library.py b/my_library.py
index 3e3f3..65c07 100644
--- a/my_library.py
+++ b/my_library.py
@@ -6,7 +6,8 @@ def segment_image(image):
    blurred = gaussian(image, sigma=2)
    # binarize the image
-   binary = threshold_otsu(blurred)
+   threshold = threshold_otsu(blurred)
+   binary = blurred > threshold
    # label connected components
    result = label(binary)
```

git

- History
- Track recent changes

The screenshot shows two GitHub windows. The left window displays the commit history for the repository 'ScaDS/BIDS-lecture-2024'. It lists three commits:

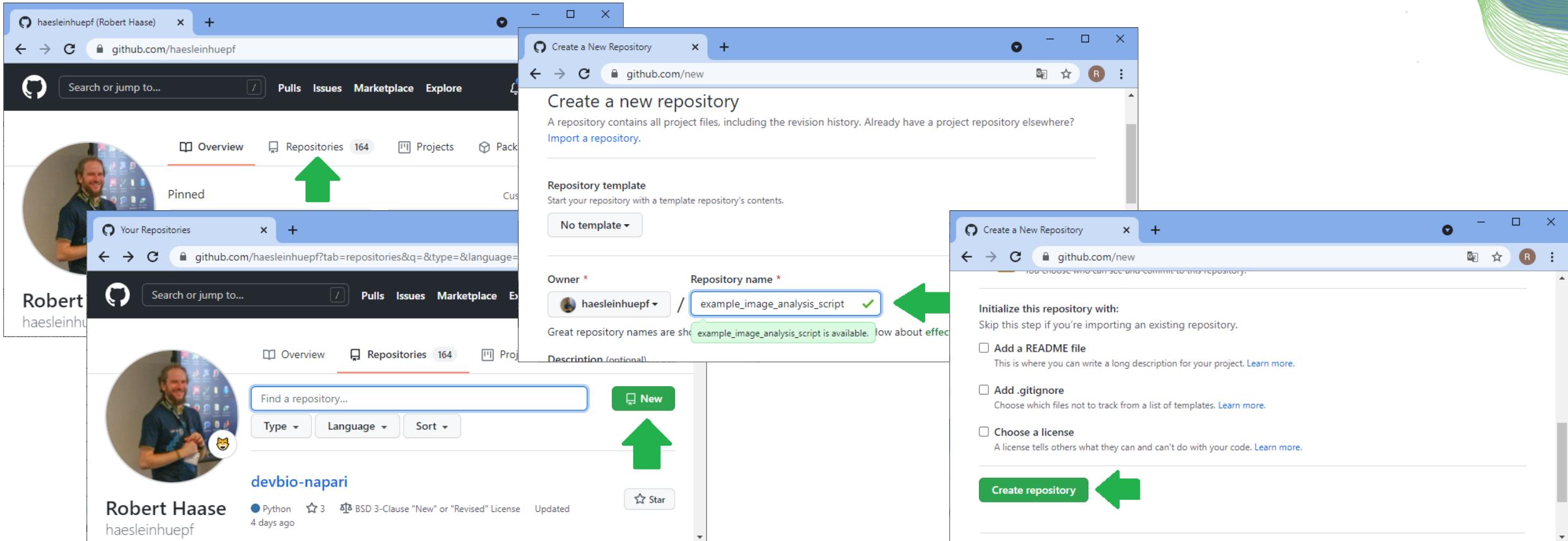
- 'add backwards compatibility exercise' by haesleinhuepf committed yesterday
- 'move pull-request exercise from week 2 to week 3' by haesleinhuepf committed yesterday
- 'fix issue with ../../ folder locations' by haesleinhuepf committed yesterday

A blue arrow points from the third commit to the right window, which shows a detailed view of the commit 'fix issue with ../../ folder locations'. The window title is 'fix issue with ../../ folder location'. It shows a diff of the file '02a_remote_files/exploring_bioimage_archive.ipynb'. The diff highlights 16 additions and 14 deletions. The code changes are as follows:

```
@@ -233,9 +233,9 @@
233 233     "    if not os.path.exists(folder_path):\n",
234 234         "        os.makedirs(folder_path)\n",
235 235     "\n",
236 -     "base_folder = f'../../data/{accession}'\n",
237 -     "raw_folder = f'../../data/{accession}/images'\n",
238 -     "groundtruth_folder = f'../../data/{accession}/groundtruth'\n",
236 +     "base_folder = f'data/{accession}'\n",
237 +     "raw_folder = f'data/{accession}/images'\n",
238 +     "groundtruth_folder = f'data/{accession}/groundtruth'\n",
239 239     "\n",
240 240     "ensure_folder_exists(base_folder)\n"
```

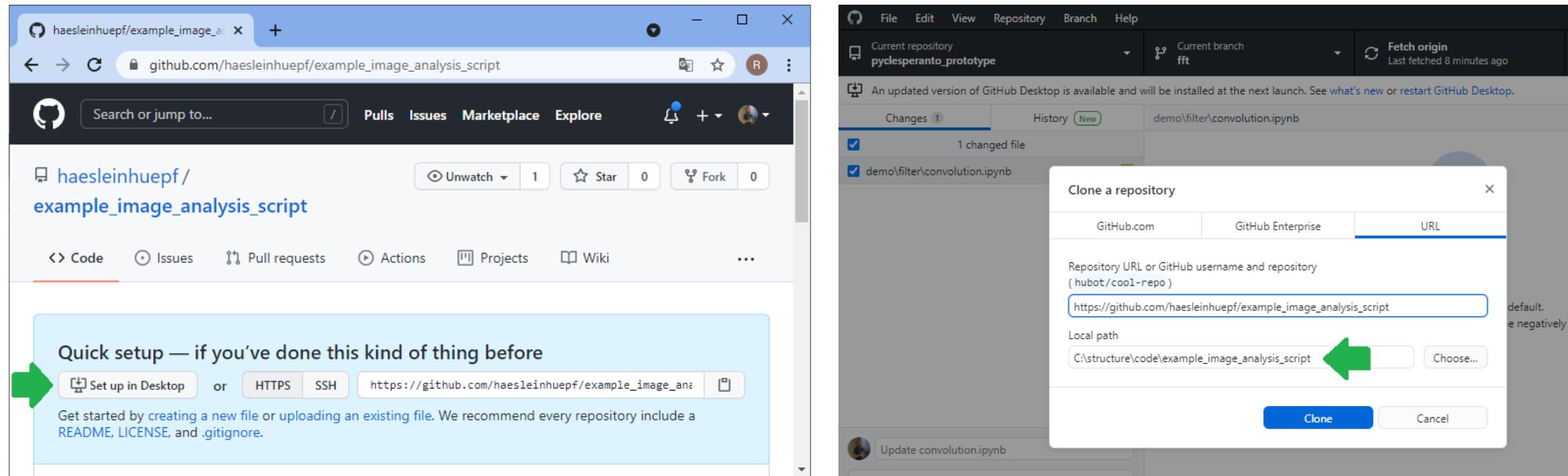
github – creating repositories

- Add a new, empty repository



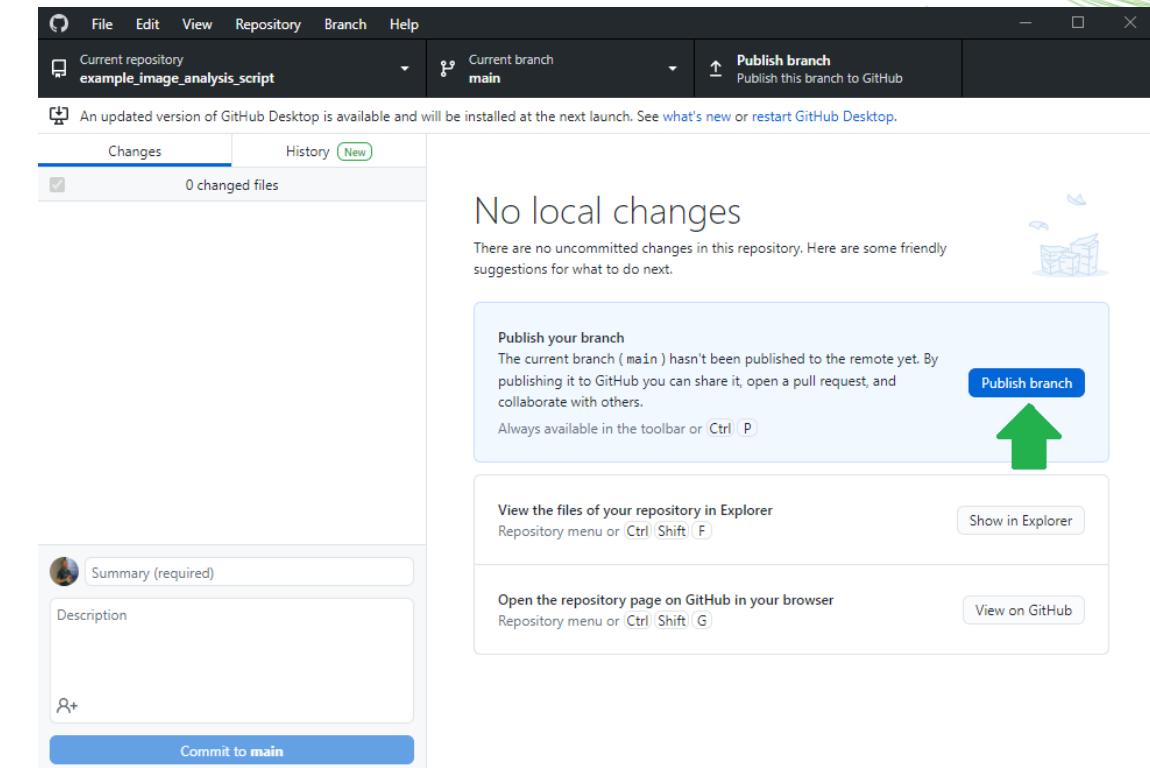
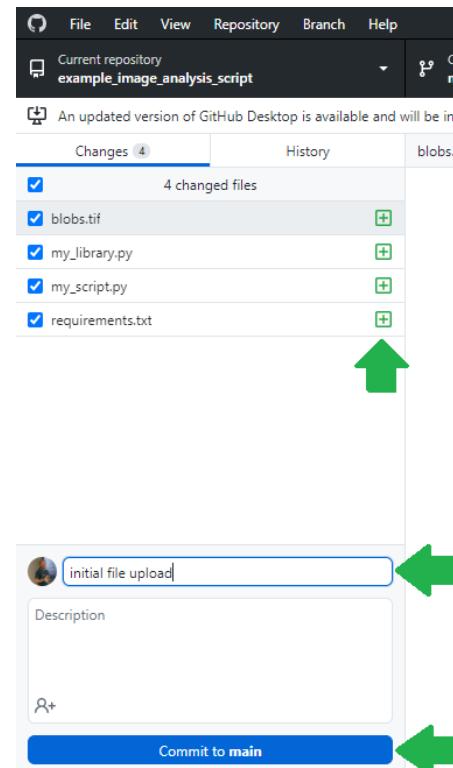
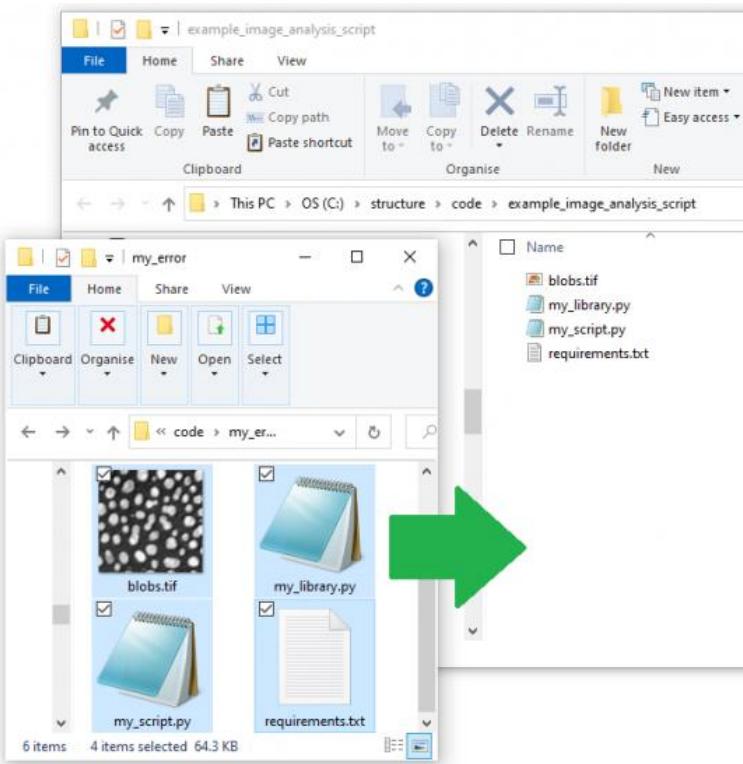
github – clone repositories

- git clone <https://github.com/organization/repository>
- Or: Use the Github Desktop app



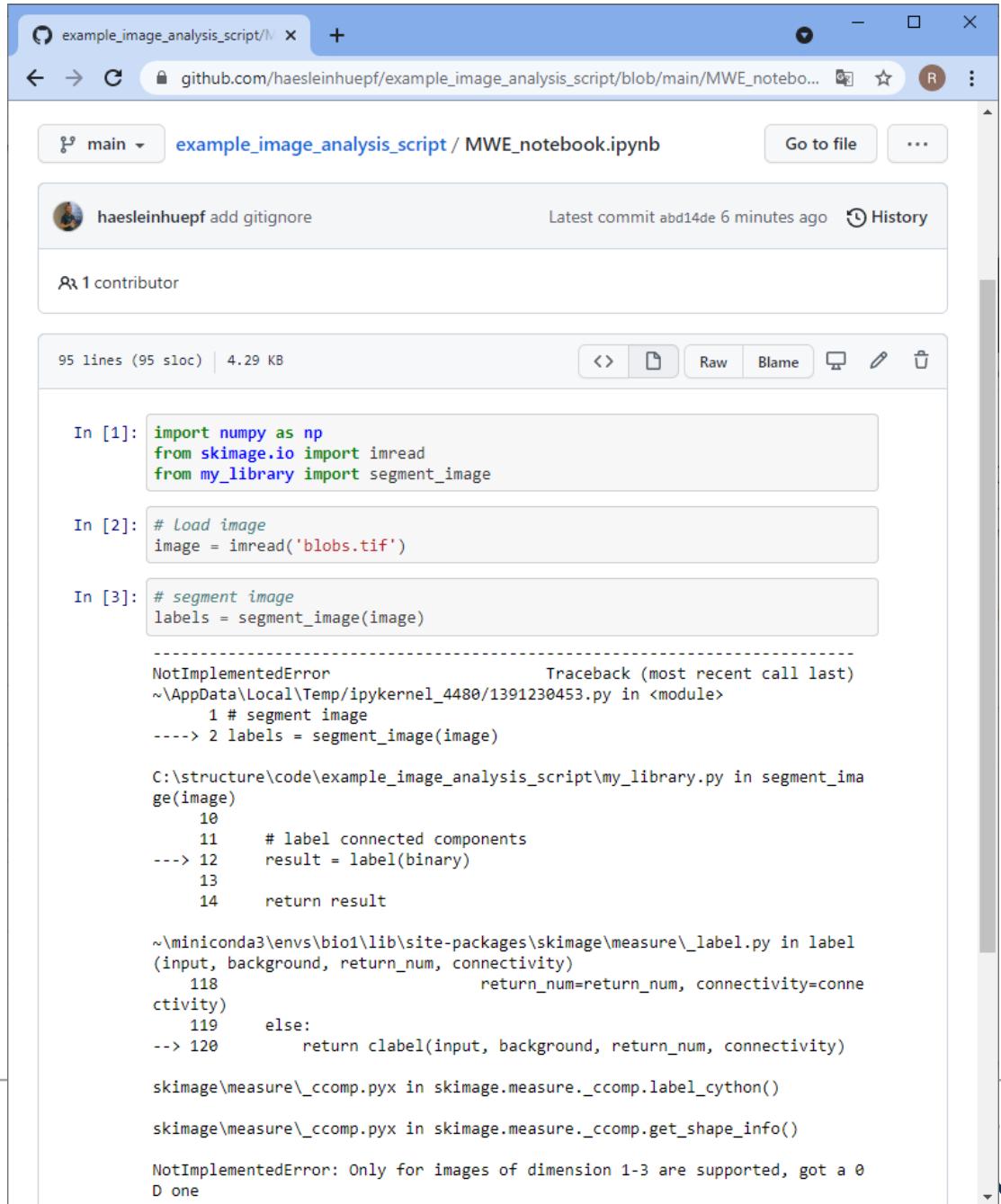
github - uploading

- git [add], commit, push



github

- Ease of reading notebooks online
- No need to download and execute code



The screenshot shows a GitHub browser window displaying a Jupyter notebook titled "example_image_analysis_script / MWE_notebook.ipynb". The notebook has 95 lines (95 sloc) and a size of 4.29 KB. It contains three code cells:

```
In [1]: import numpy as np
from skimage.io import imread
from my_library import segment_image

In [2]: # Load image
image = imread('blobs.tif')

In [3]: # segment image
labels = segment_image(image)

NotImplementedError Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4480\1391230453.py in <module>
      1 # segment image
----> 2 labels = segment_image(image)

C:\structure\code\example_image_analysis_script\my_library.py in segment_image
      10
      11     # label connected components
----> 12     result = label(binary)
      13
      14     return result

~\miniconda3\envs\bio1\lib\site-packages\skimage\measure\_label.py in label
    (input, background, return_num, connectivity)
        118                                         return_num=return_num, connectivity=connectivity)
    119     else:
--> 120         return clabel(input, background, return_num, connectivity)

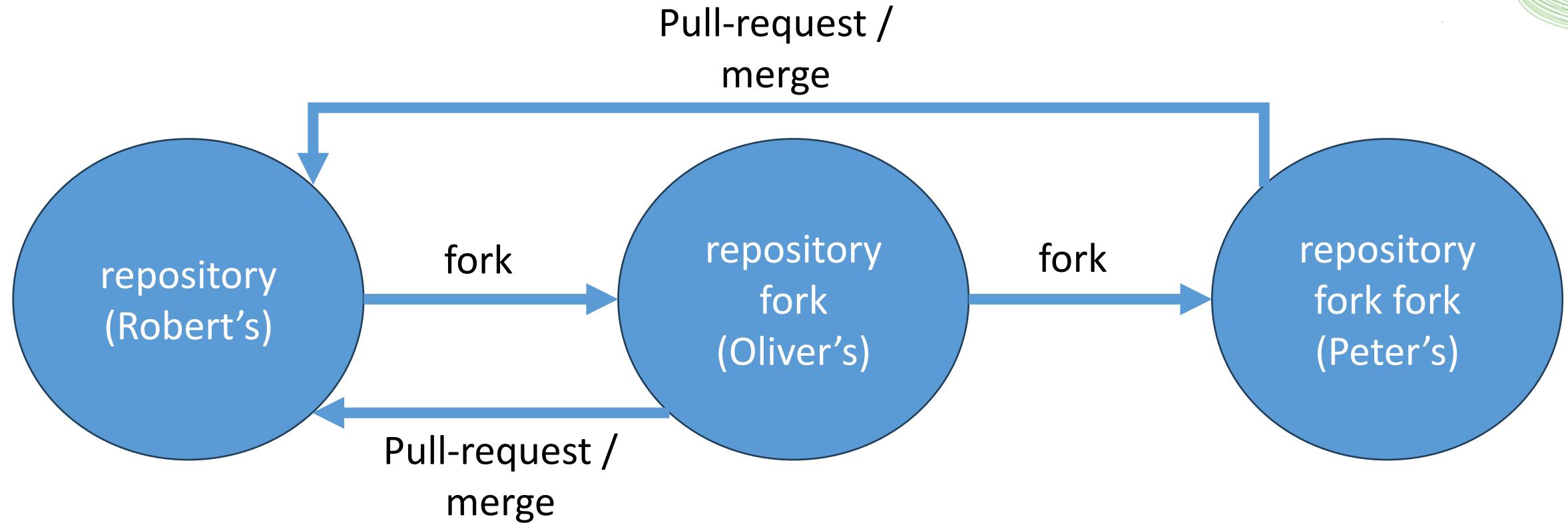
skimage\measure\_ccomp.pyx in skimage.measure._ccomp.label_cython()

skimage\measure\_ccomp.pyx in skimage.measure._ccomp.get_shape_info()

NotImplementedError: Only for images of dimension 1-3 are supported, got a 0D one
```

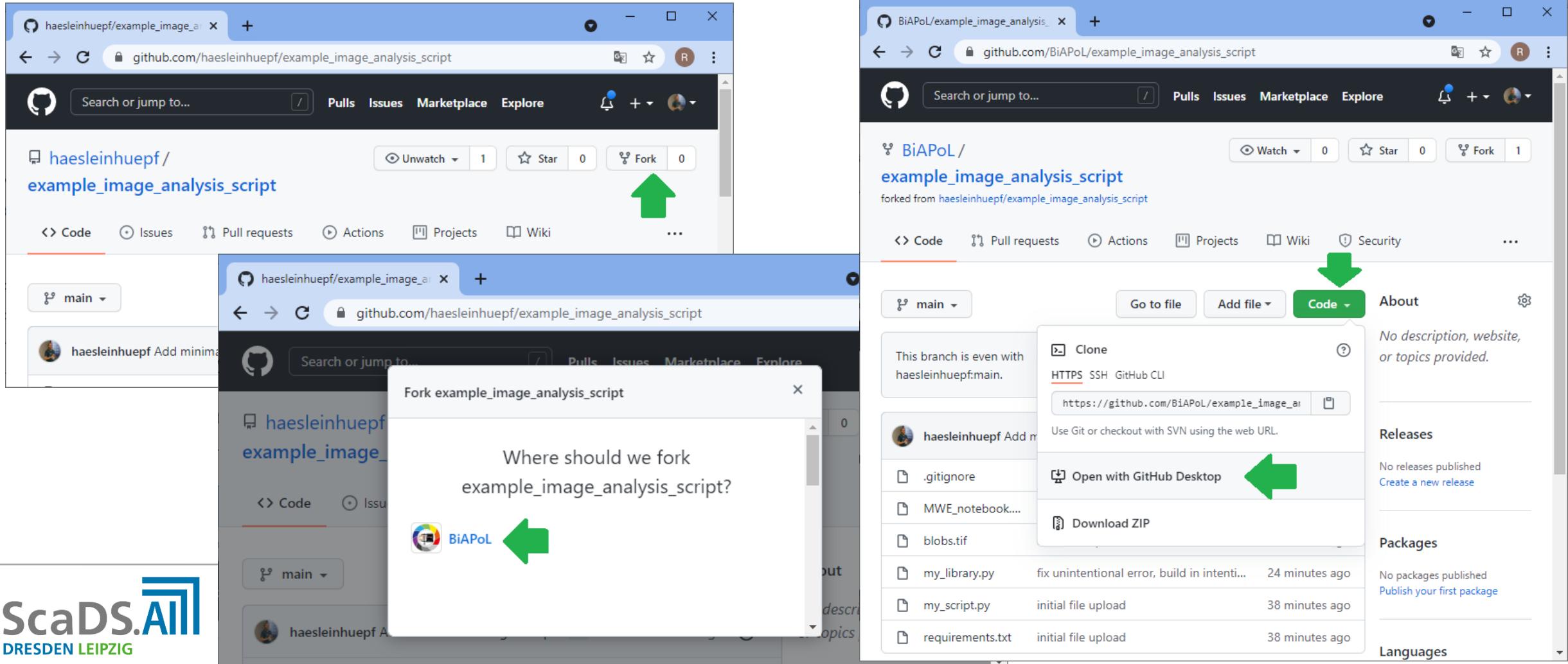
git - forking

- Making a copy where we have edit rights



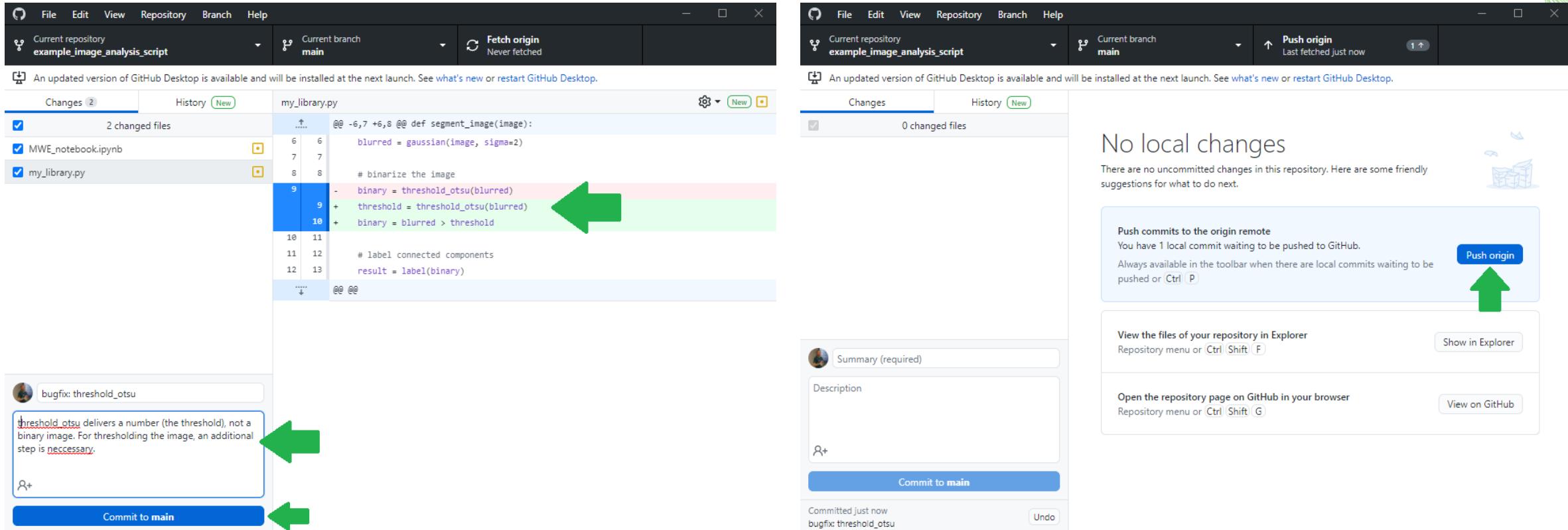
github - forking

- Making a copy where we have edit rights



github – uploading (again)

- After fixing a bug, we upload the changes to our fork



Github – pull requests

- Contribute to open-source projects

The image displays two side-by-side screenshots of GitHub interfaces, illustrating the process of contributing to an open-source project via pull requests.

Screenshot 1: BiAPoL/example_image_analysis_script Repository

This screenshot shows the repository page for `BiAPoL/example_image_analysis_script`. The main navigation bar includes `Pulls`, `Issues`, `Marketplace`, and `Explore`. Below the repository name, it shows it was forked from `haesleinhuepf/example_image_analysis_script`. The `Code` tab is selected. A green arrow points to the `Open pull request` button, which is highlighted in red. Another green arrow points to the `Contribute upstream` dropdown menu, which also has a red highlight.

Screenshot 2: Pull Request Comparison Page

This screenshot shows the "Open a pull request" page for comparing changes between branches. It includes fields for `base repository` (`haesleinhuepf/example_image...`) and `head repository` (`BiAPoL/example_image_analysi...`). A green checkmark indicates "Able to merge". The pull request body contains a message from Robert Haase:

```
bugfix: threshold_otsu

Dear Robert,

here comes a bug fix for your image segmentation function. threshold_otsu delivers a number (the threshold), not a binary image. For thresholding the image, an additional step is necessary.

Best,
Robert
```

At the bottom right of the pull request body is a `Create pull request` button.

Github – pull requests

- Reviewer perspective

The image shows two screenshots of a GitHub pull request interface. The left screenshot displays the pull request details for 'bugfix: threshold_otsu #1' from 'haesleinhuepf' to 'haesleinhuepf/main'. It shows 2 files changed, with a green arrow pointing to the 'Files changed' button. The right screenshot shows the review interface with a comment from a user named 'Robert' saying 'Thank you Robert! That's great 😊'. The interface includes sections for 'Continuous integration has not been set up', 'This branch has no conflicts with the base branch', and a 'Merge pull request' button.

Github – pull requests

- Reviewer perspective

The screenshot shows a GitHub pull request page for a repository named 'example_image_analysis_script'. The pull request is titled 'bugfix: threshold_otsu #1' and has been merged. The commit message is 'bugfix: threshold_otsu ...' with a SHA of '65c074a'. A review comment from 'haesleinhuepf' says:

Dear Robert,
here comes a bug fix for your image segmentation function. `threshold_otsu` delivers a number (the threshold), not a binary image. For thresholding the image, an additional step is necessary.
Best,
Robert

Below the comment, another review comment from 'haesleinhuepf' says:

Thank you Robert! That's great 😊

On the right side of the page, there are sections for 'Reviewers', 'Assignees', 'Labels', 'Projects', 'Milestone', and 'Linked issues'.

The screenshot shows a GitHub repository page for 'haesleinhuepf/example_image_analysis_script'. The repository has 1 unwatched star, 0 forks, and 1 pull request. The pull request is titled 'bugfix: threshold_otsu ...' and was merged by 'haesleinhuepf' 17 minutes ago. The commit message is 'bugfix: threshold_otsu ...' with a SHA of '65c074a'. The repository has 1 contributor.

On the right, there is an inline code editor showing a Jupyter notebook cell with the following code:

```
In [1]: import numpy as np
from skimage.io import imread
from my_library import segment_image

In [2]: # Load image
image = imread('blobs.tif')

In [3]: # segment image
labels = segment_image(image)

In [4]: # count objects
number_of_objects = labels.max()
print('Number of objects', number_of_objects)
```

The output of the code is 'Number of objects 61'. To the right of the code editor, a green box contains the text 'Problem solved :-)'.

Quiz (recap)

- Sending my modifications to a git-server is done using the command git ...

submit



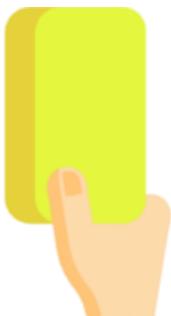
push



pull



commit



Github

- If this was too fast...

The screenshot shows a web browser window displaying a blog post from the FocalPlane website. The header features the FocalPlane logo and the text "Where biology meets microscopy". Below the header is a navigation bar with links to Home, About us, Topics, Gallery, Jobs, Events, Resources, Network, Contact us, and Log in/register. A search icon is also present. The main content area shows a breadcrumb trail: Home / How to / Collaborative bio-image analysis script ... The title of the post is "Collaborative bio-image analysis script editing with git". It was posted by Robert Haase on 4 September 2021. The post begins with a TL;DR summary: "I'm a computer scientist who often collaborates with biologists on bio-image analysis scripts. We are using more and more git, a version control program, for working on code collaboratively. When using git, we speak about repositories, commits and pushing to the origin. We also make forks, send pull-requests and merge code. This blog post explains these terms and demonstrates how a typical collaborative bio-image analysis scripting project looks like." The text continues with a personal anecdote about writing a script that counts cells and needing help from experts.

The screenshot shows a web browser window displaying a GitHub profile page for the user "haesleinhuepf". The profile page includes a bio image of a smiling man, a pinned item, and sections for Overview, Repositories (164), Projects, and Packages. A green arrow points upwards towards the "Repositories" tab, indicating where to click to demonstrate the process of creating a new repository for sharing code.



DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Conda environments

Robert Haase

GEFÖRDERT VOM



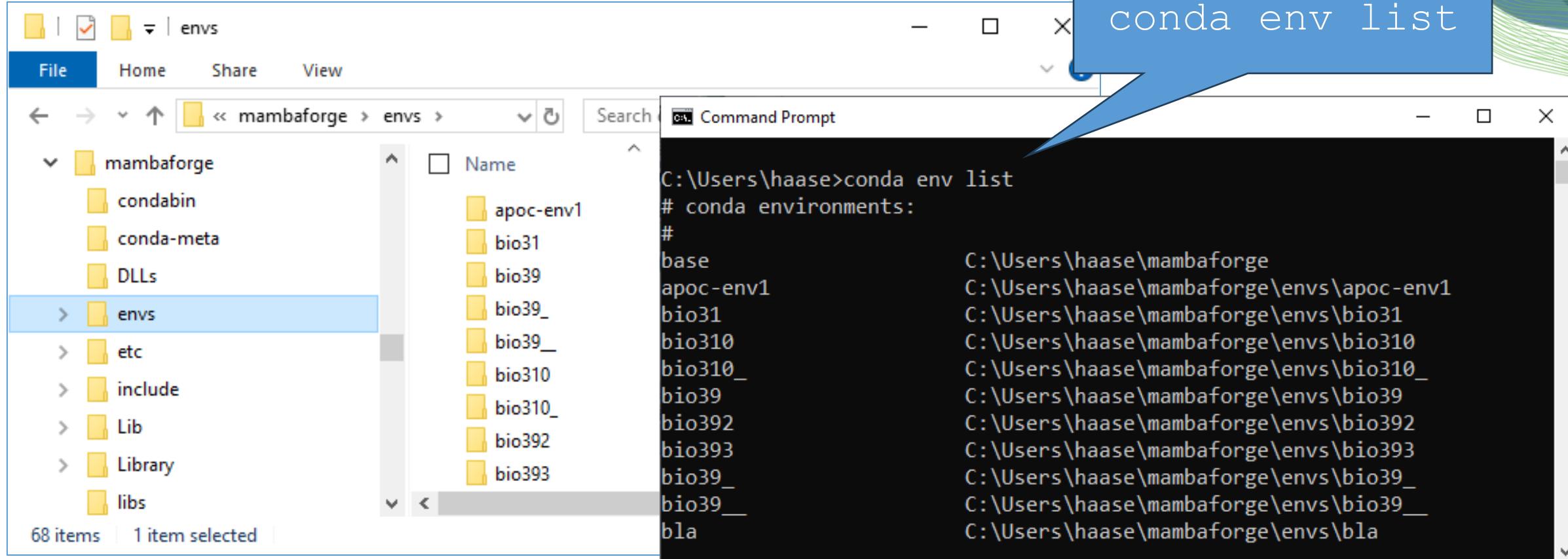
Bundesministerium
für Bildung
und Forschung

Conda environments

- Conda is a package manager
 - Allows to install Python packages
 - Allows to install other stuff (git, JDK)
- Conda is an environment manager
 - Virtual environments
 - Import/export/distribute environments
- Pip can only install Python packages (also into conda environments)

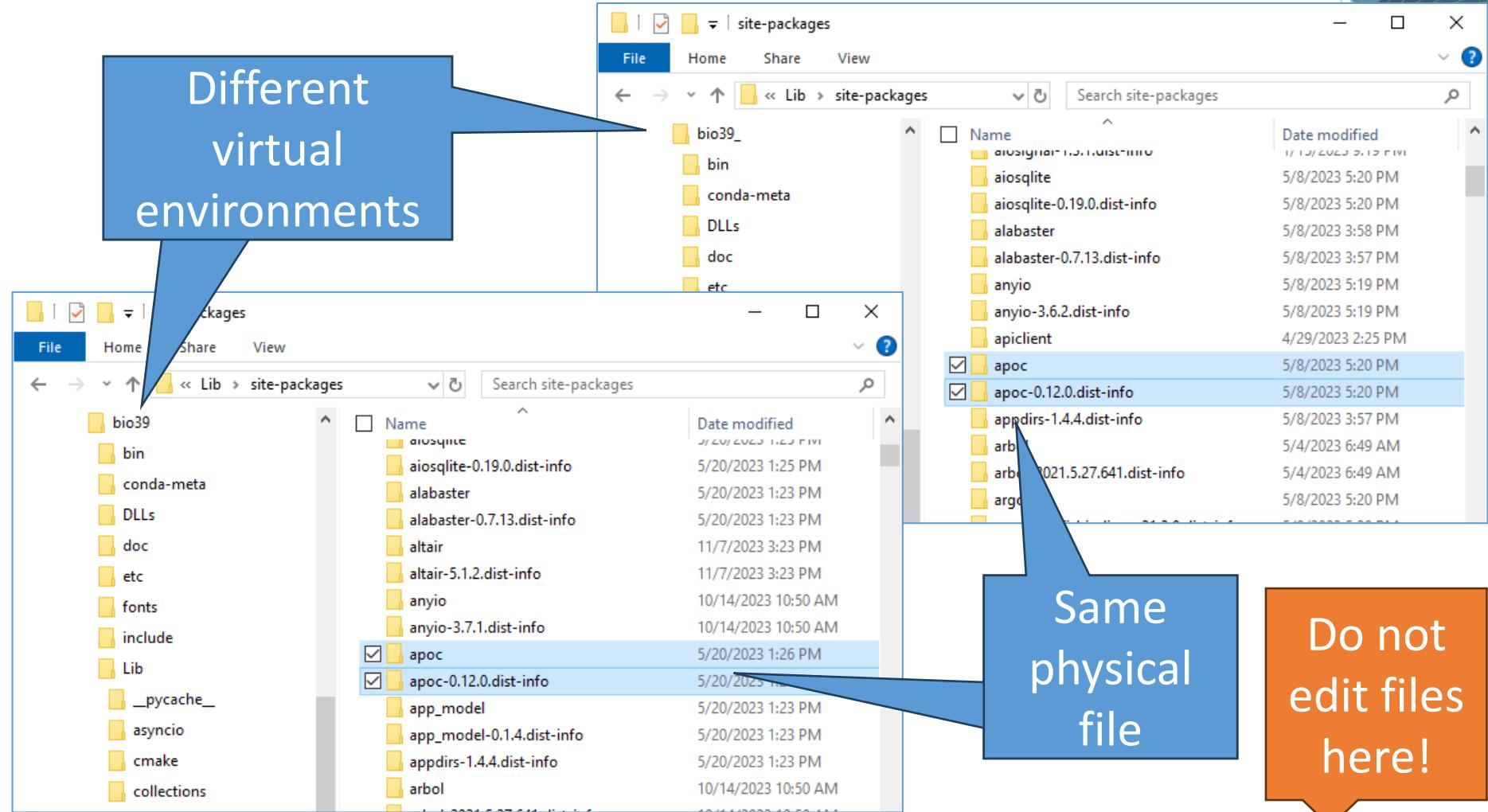
Conda environments

- A conda environment is *just a folder*



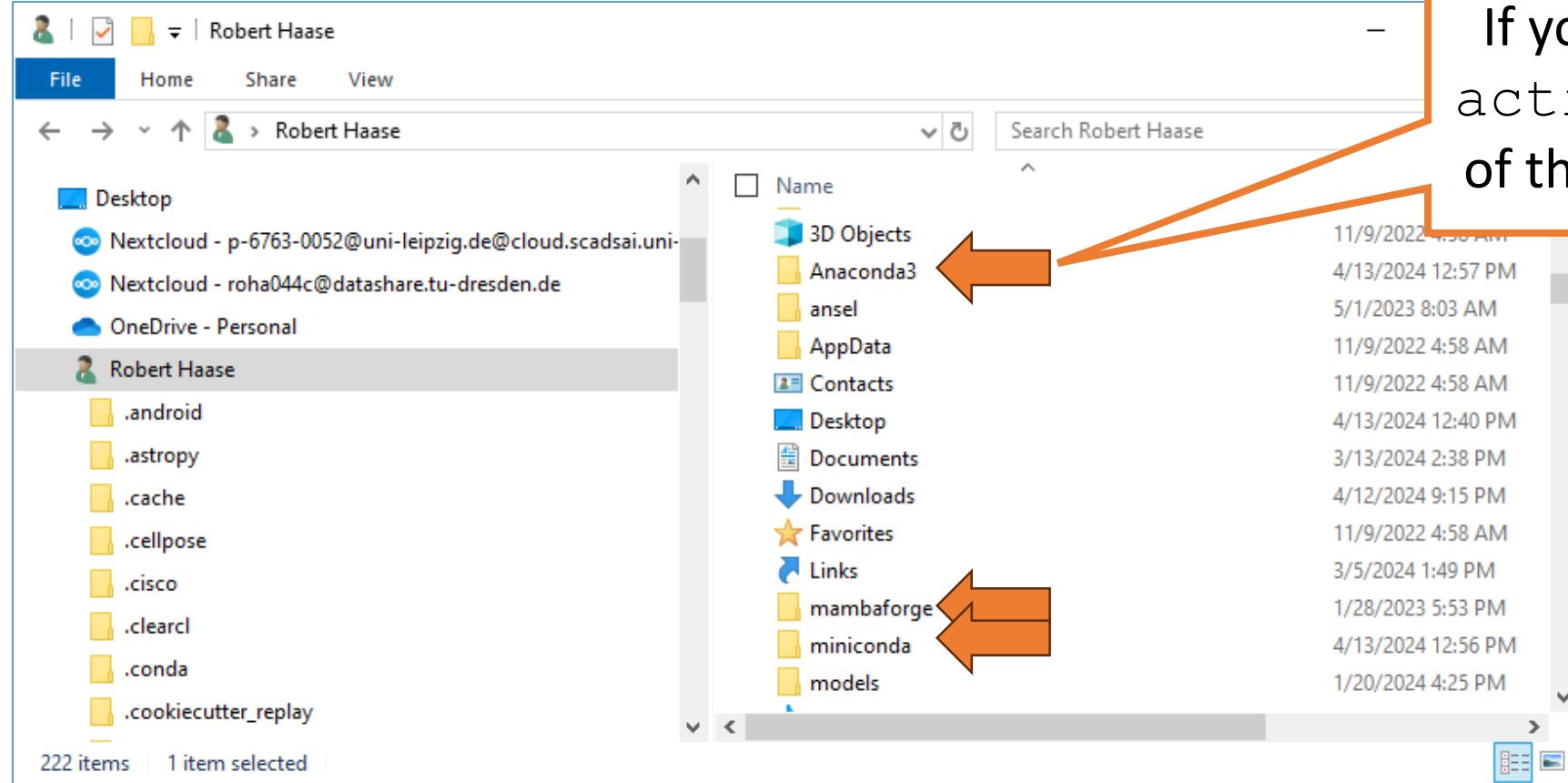
Conda environments

- Packages within the folders are just linked
- If you change a file in env1, the same file in env2 is changed.



Multiple Python installations

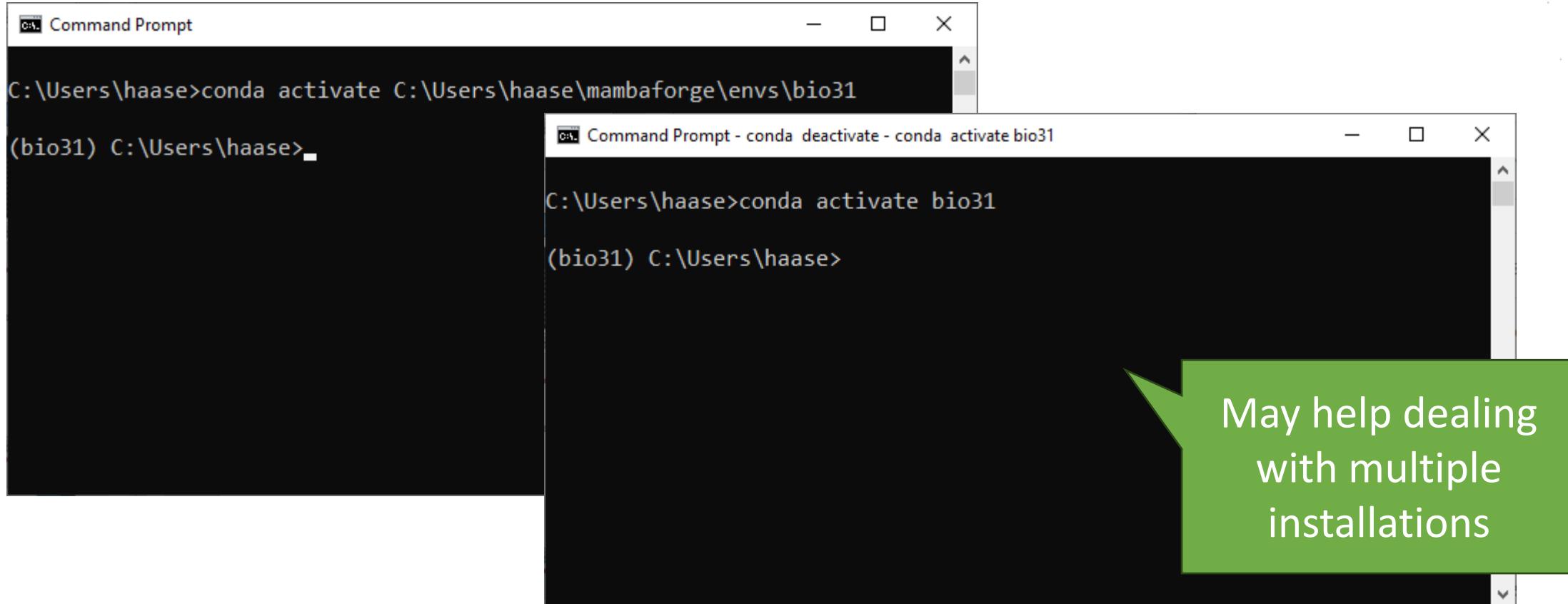
- On some computers you find this



If you enter `conda activate ...`, which of those is executed?

Conda environments

- Conda environments can be activated from anywhere



The image shows two separate Command Prompt windows. The left window has a title bar 'Command Prompt' and contains the command 'conda activate C:\Users\haase\mambaforge\envs\bio31'. The right window has a title bar 'Command Prompt - conda deactivate - conda activate bio31' and contains the command 'conda activate bio31'. Both windows show the prompt '(bio31) C:\Users\haase>' indicating the environment is active.

May help dealing
with multiple
installations

Broken [conda] environments

- Common case of confusion in the Python ecosystem
 - Happens to everyone – it's just a matter of time.

Background

- The base-environment is special.
- You install packages into the base environment, if you do not run `conda activate my_env`
- You can run base-software from within other environments

Base

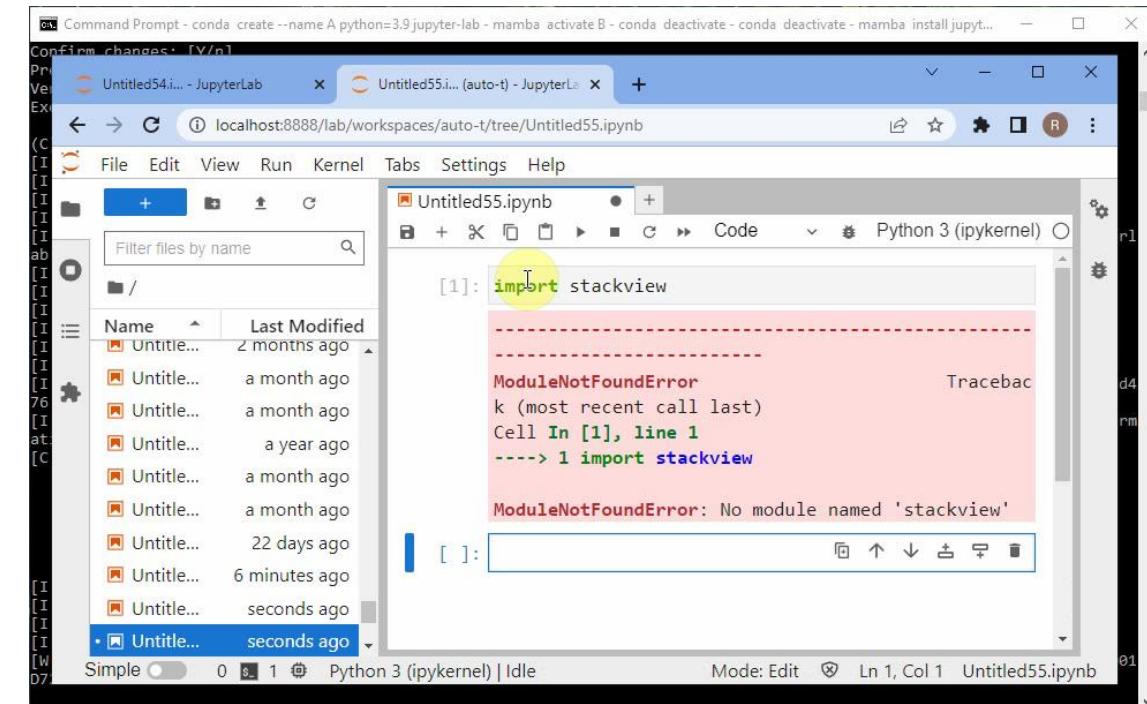
Installed packages:

Jupyter lab

my_env

Installed packages:

stackview



The screenshot shows a Jupyter Notebook interface with two tabs: 'Untitled54.ipynb' and 'Untitled55.ipynb'. The 'Untitled55.ipynb' tab is active, displaying a code cell with the following content:

```
[1]: import stackview
```

An error message is visible in the output area:

```
ModuleNotFoundError
k (most recent call last)
Cell In [1], line 1
----> 1 import stackview

ModuleNotFoundError: No module named 'stackview'
```

The 'File Explorer' sidebar on the left shows a list of files in the current directory, including several 'Untitled...' files.

Broken [conda] environments

- The only cure: Uninstall and reinstall [Ana]conda.



- Alternatively:
 - Create a new environment for every project.
 - `conda activate my_env`
 - Do not install project-specific stuff in the base-environment.

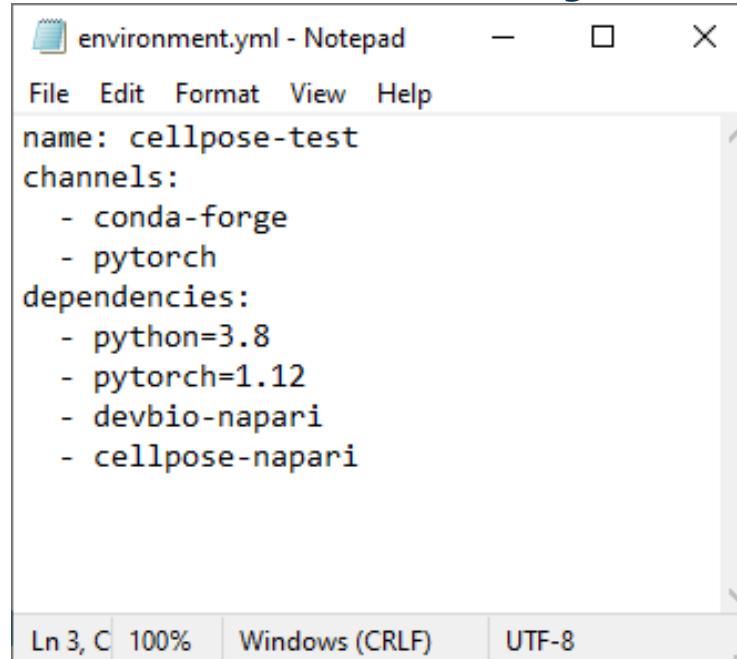
Broken [conda] environments

Do not install anything in the base environment

Always call **conda activate ...** before doing anything

Documenting dependencies

- Maintain a document with the dependencies (and versions) you need in your project!
 - The conda way
 - The pip way

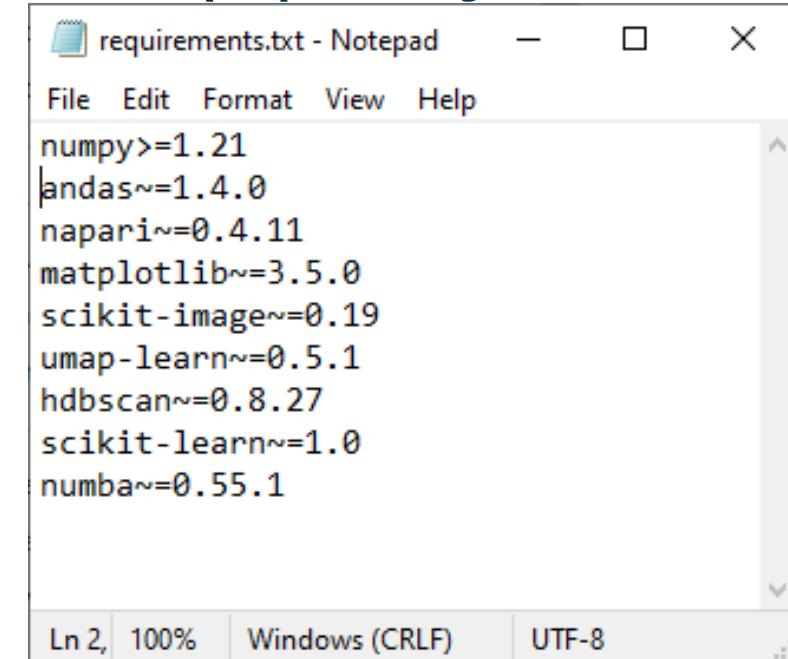


```
environment.yml - Notepad
File Edit Format View Help
name: cellpose-test
channels:
- conda-forge
- pytorch
dependencies:
- python=3.8
- pytorch=1.12
- devbio-napari
- cellpose-napari

Ln 3, C 100% Windows (CRLF) UTF-8
```

conda env create -f environment.yml

In case your
environment is screwed
up, you can rebuild it
any time.



```
requirements.txt - Notepad
File Edit Format View Help
numpy>=1.21
andas~=1.4.0
napari~=0.4.11
matplotlib~=3.5.0
scikit-image~=0.19
umap-learn~=0.5.1
hdbscan~=0.8.27
scikit-learn~=1.0
numba~=0.55.1

Ln 2, 100% Windows (CRLF) UTF-8
```

pip install -r requirements.txt

Documenting dependencies

- ... the complete way.

```
conda env export > environment.yml
```

```
environment.yml - Notepad
File Edit Format View Help
name: bio_39
channels:
- conda-forge
- defaults
dependencies:
- alabaster=0.7.12=py_0
- anyio=3.6.1=pyhd8ed1ab_1
- aom=3.5.0=h63175ca_0
- apoc-backend=0.10.0=pyhd8ed1ab_0
- appdirs=1.4.4=pyh9f0ad1d_0
- argon2-cffi=21.3.0=pyhd8ed1ab_0
- argon2-cffi-bindings=21.2.0=py39hb82d6ee_2
- asciitree=0.3.3=py_2
- asttokens=2.0.8=pyhd8ed1ab_0
- attrs=22.1.0=pyh71513ae_1
- autopep8=1.7.0=pyhd8ed1ab_0
- babel=2.10.3=pyhd8ed1ab_0
- backcall=0.2.0=pyh9f0ad1d_0
backgrounds=1.0=py_2
Ln 391, Col 46 100% Windows (CRLF) UTF
```

```
environment.yml - Notepad
File Edit Format View Help
- win_inet_pton=1.1.0=py39hcbf5309_4
- winpty=0.4.3=4
- wrapt=1.14.1=py39hb82d6ee_0
- x264=1!164.3095=h8ffe710_2
- x265=3.5=h2d74725_3
- xorg-libxau=1.0.9=hcd874cb_0
- xorg-libxdmcp=1.1.3=hcd874cb_0
- xz=5.2.6=h8d14728_0
- yaml=0.2.5=h8ffe710_2
- zarr=2.13.3=pyhd8ed1ab_0
- zeromq=4.3.4=h0e60522_1
- zfp=1.0.0=h0e60522_1
- zict=2.2.0=pyhd8ed1ab_0
- zipp=3.9.0=pyhd8ed1ab_0
- zlib-ng=2.0.6=h8ffe710_0
- zstd=1.5.2=h7755175_4
prefix: C:\Users\rober\mambaforge\envs\bio_39
Ln 391, Col 46 100% Windows (CRLF) UTF-8
```

Excellent way to document which dependencies were *actually used...*

It is *questionable* if re-creating an environment from this yml file works.

Installing dependencies

- A difference between pip and conda:

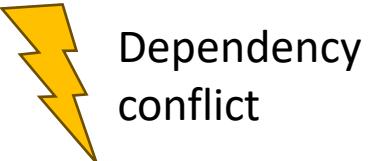
```
pip install package_a
```

```
conda install package_a
```

Depends on:
numpy \leq 1.22.0

```
pip install package_b
```

```
conda install package_b
```



Depends on:
numpy \geq 1.22.0

passes

fails

Because the environment
cannot be solved

Installing dependencies

- A difference between pip and conda:

pip install git

fails

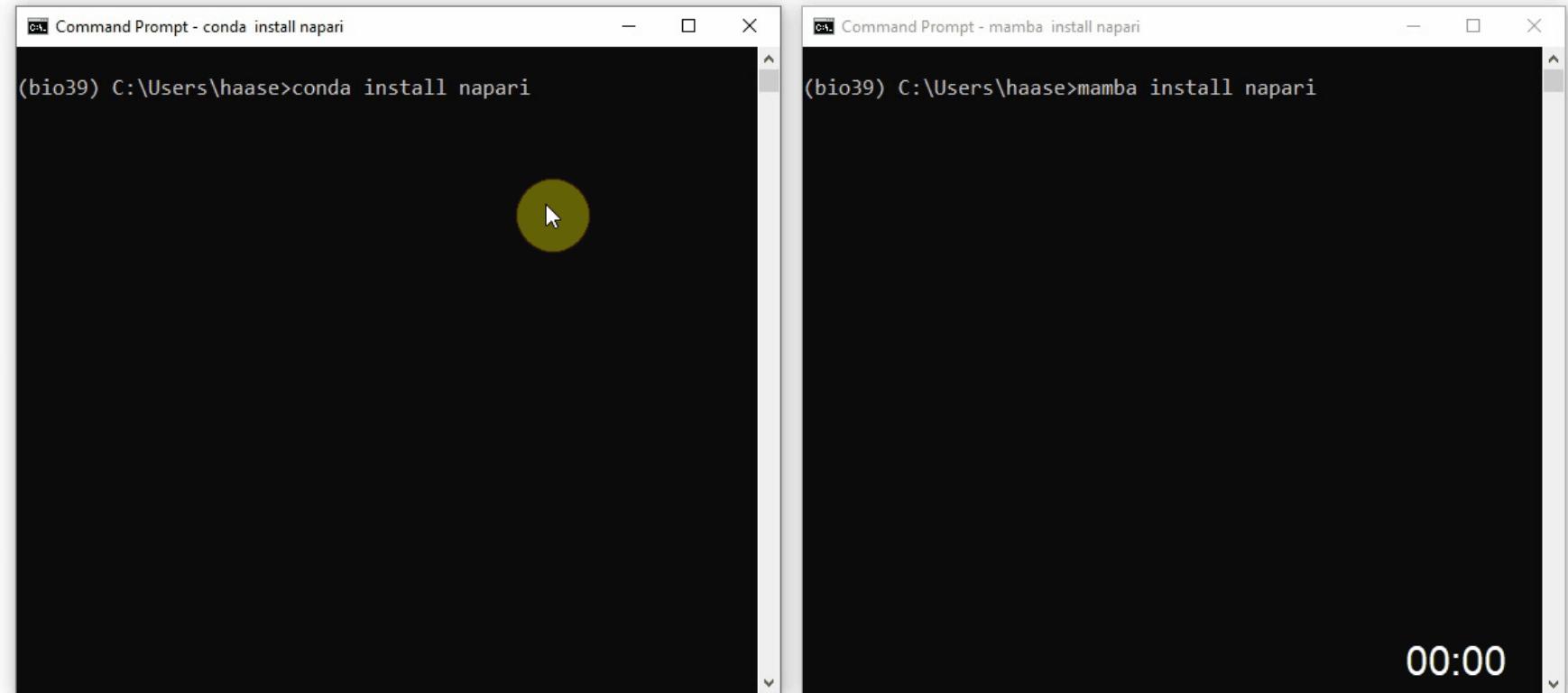
Because git is not a
python package

conda install git

passes

Conda versus mamba

- mamba is a “drop in” replacement that accepts the same commands as conda.
- mamba is much faster though.



Quiz

conda install package_a

Depends on:
numpy \leq 1.22.0

pip install package_b

Depends on:
numpy \geq 1.22.0



Dependency
conflict

fails



works

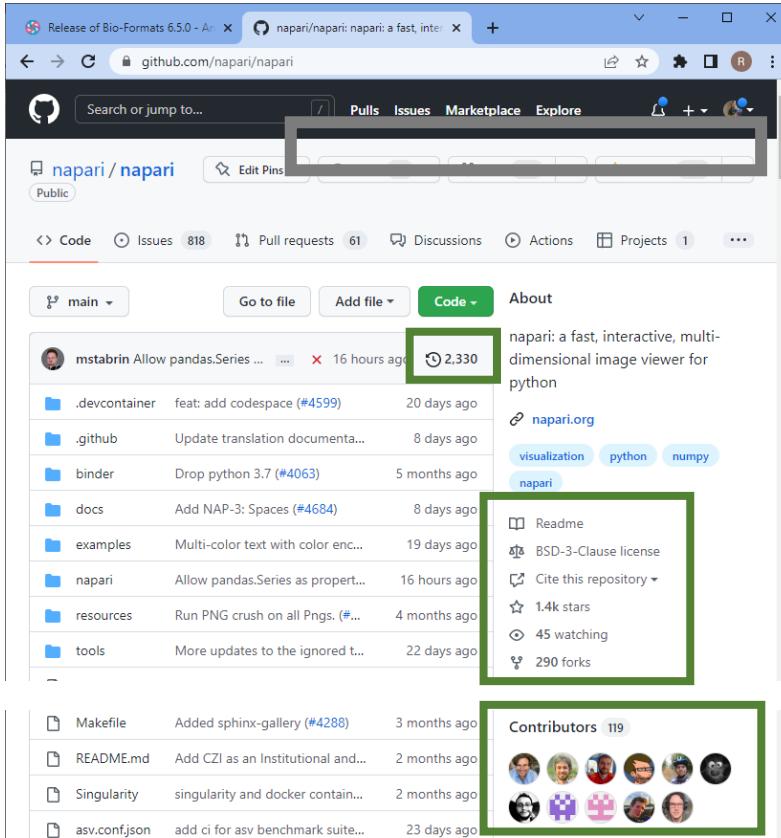


works but...

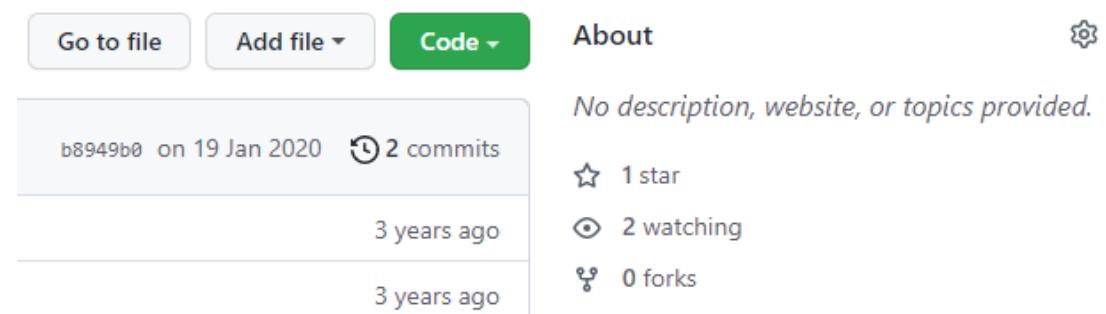


Software quality indicators

- Visit the project's github or gitlab page and review indicators.

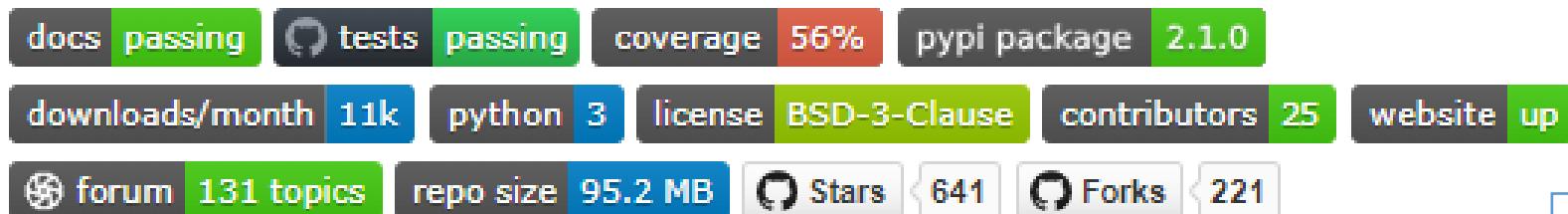


- **Stars:** People like software, similarly to tweets on Twitter
- **Watching:** People receive updates for new releases
- **Forks:** People made a copy of the code, e.g. to contribute to the project
- **Contributors:** People who contributed to the code
- **Commits:** Changes to the code



Software quality indicators

- Visit the project's github or gitlab page and review indicators.

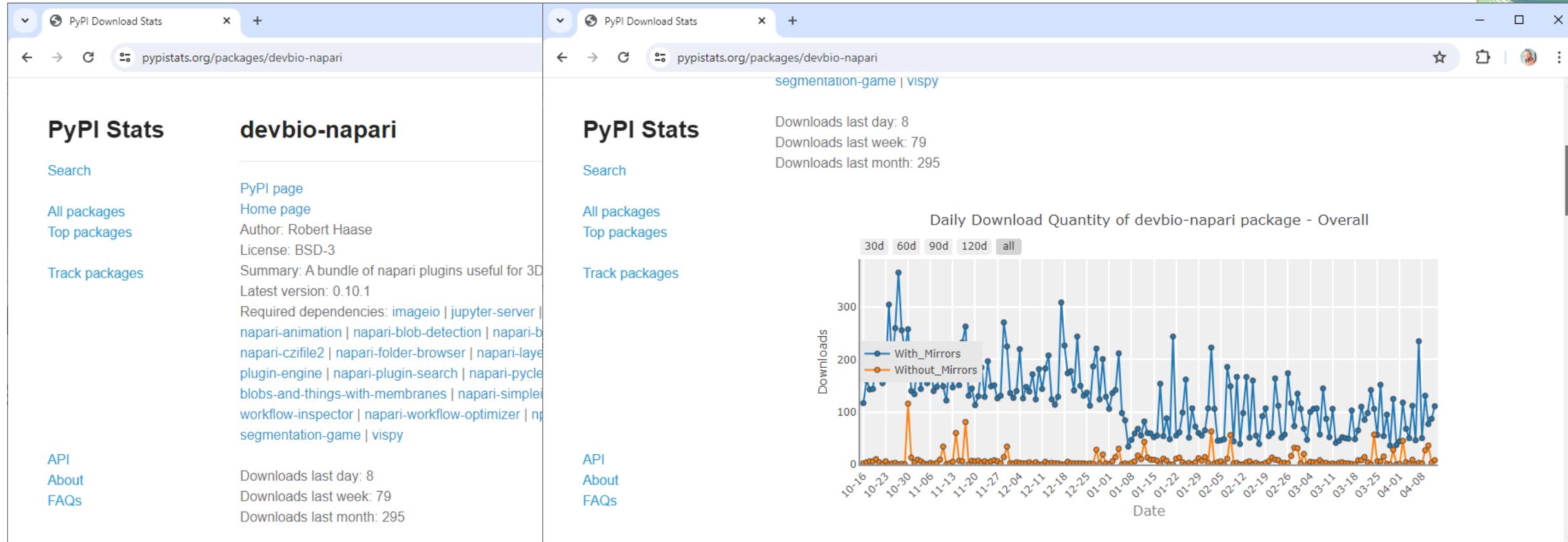


Note, github badges
cannot be *deserved*.
Developers put
them there



Software quality indicators

- Download statistics: pypi



Software quality indicators

- Download statistics: conda

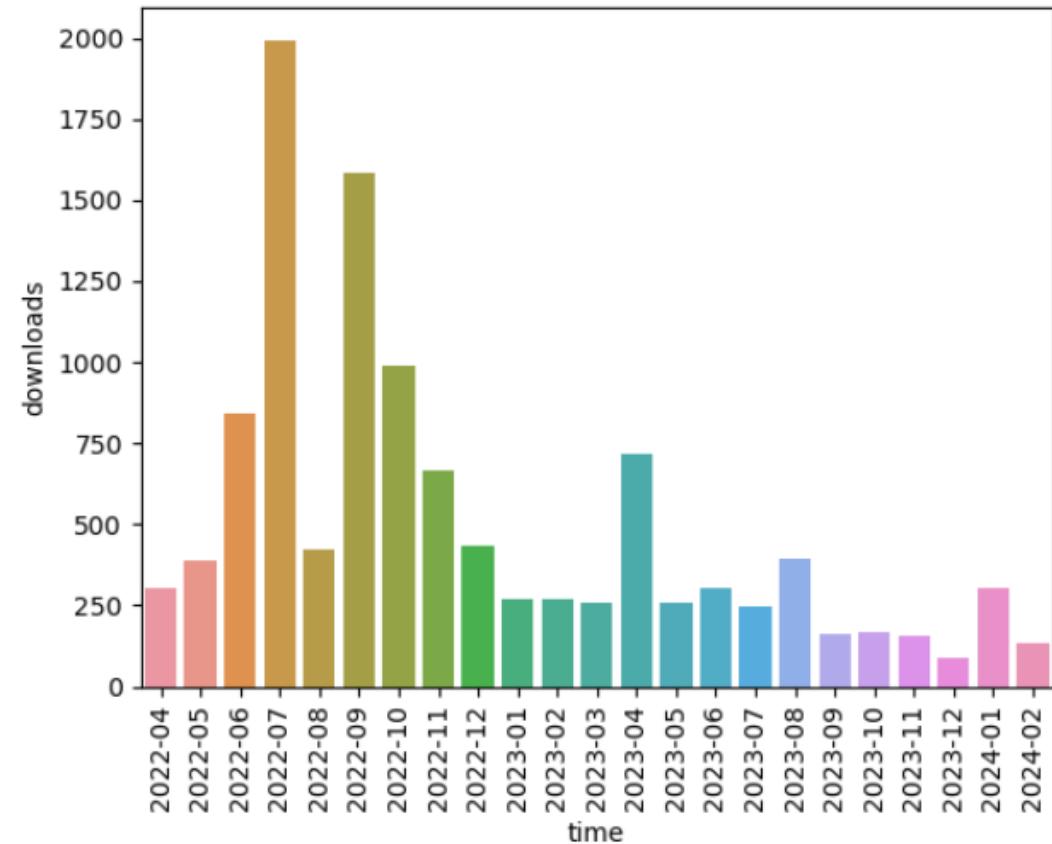
```
[2]: p = pkg_python('devbio-napari', monthly=True)
```

```
C:\Users\haase\mambaforge\envs\bio39\lib\site-packages\condastats\cli.py:153:  
FutureWarning: The default of observed=False is deprecated and will be changed  
to True in a future version of pandas. Pass observed=False to retain current b  
ehavior or observed=True to adopt the future default and silence this warning.  
agg = df.groupby(["pkg_name", "time", "column]).counts.sum()
```

Some reformatting is necessary to get this in a processable format.

```
[3]: df = pd.DataFrame({  
    'time':[t[1] for t in p.index],  
    'downloads':p.tolist()  
})  
df
```

| | time | downloads |
|---|---------|-----------|
| 0 | 2022-04 | 304 |
| 1 | 2022-05 | 391 |
| 2 | 2022-06 | 842 |
| 3 | 2022-07 | 1994 |
| 4 | 2022-08 | 423 |
| 5 | 2022-09 | 1583 |



Software quality indicators

- Scientific publications

The screenshot shows a GitHub repository page for "MouseLand/cellpose: a generalist algorithm for cellular segmentation". The page includes a README file, a BSD-3-Clause license, and several sample images of cell segmentation results. A section titled "CITATION" provides instructions for citing the software. It includes a citation for Cellpose 1.0 and a note about the human-in-the-loop training for Cellpose 2.0.

If you use Cellpose 1, 2 or 3, please cite the Cellpose 1.0 paper:
Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1), 100-106.

If you use the human-in-the-loop training, please also cite the Cellpose 2.0 paper:
Pachitariu, M. & Stringer, C. (2022). Cellpose 2.0: how to train your own model. *Nature methods*, 1-8.

The screenshot shows a Nature Methods article page for "Cellpose 2.0: how to train your own model". The page includes the journal header, navigation links, and the article title. Below the title, it lists the authors (Marius Pachitariu & Carsen Stringer), the journal issue (Nature Methods 19, 1634–1641 (2022)), and citation metrics (55k Accesses, 120 Citations, 116 Altmetric). Two arrows point to the "Accesses" and "Citations" metrics.

Article | Open access | Published: 07 November 2022

Cellpose 2.0: how to train your own model

Marius Pachitariu & Carsen Stringer

Nature Methods 19, 1634–1641 (2022) | Cite this article

55k Accesses | 120 Citations | 116 Altmetric | Metrics



DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Image Processing

Robert Haase

Reusing materials from Mauricio Rocha Martins
(Norden lab, MPI CBG); Dominic Waithe (Oxford
University); Alex Bird, Dan White (MPI CBG)

GEFÖRDERT VOM



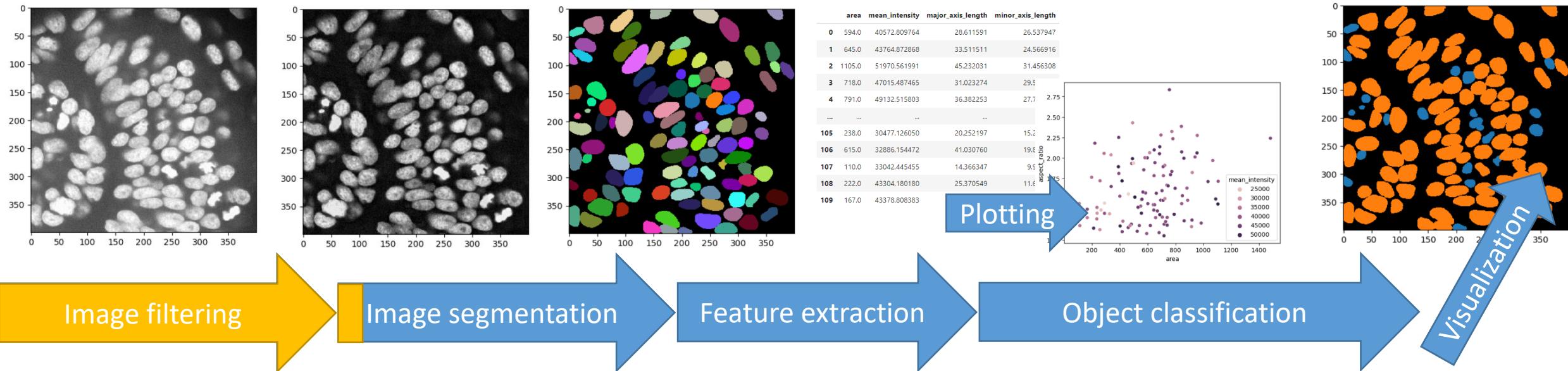
Bundesministerium
für Bildung
und Forschung



Diese Maßnahme wird gefördert durch die Bundesregierung
aufgrund eines Beschlusses des Deutschen Bundestages.
Diese Maßnahme wird mitfinanziert durch Steuermittel auf
der Grundlage des von den Abgeordneten des Sächsischen
Landtags beschlossenen Haushaltes.

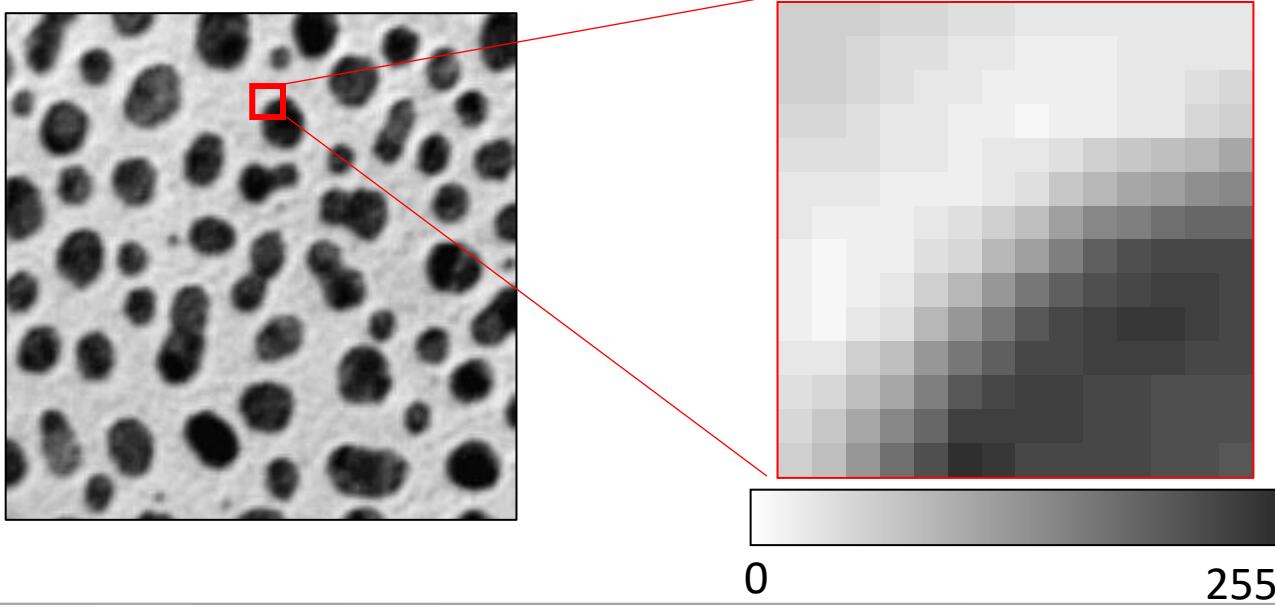
Lecture overview: Bio-image Analysis

- Image Data Analysis workflows
- Goal: Quantify observations, substantiate conclusions with numbers



Images and pixels

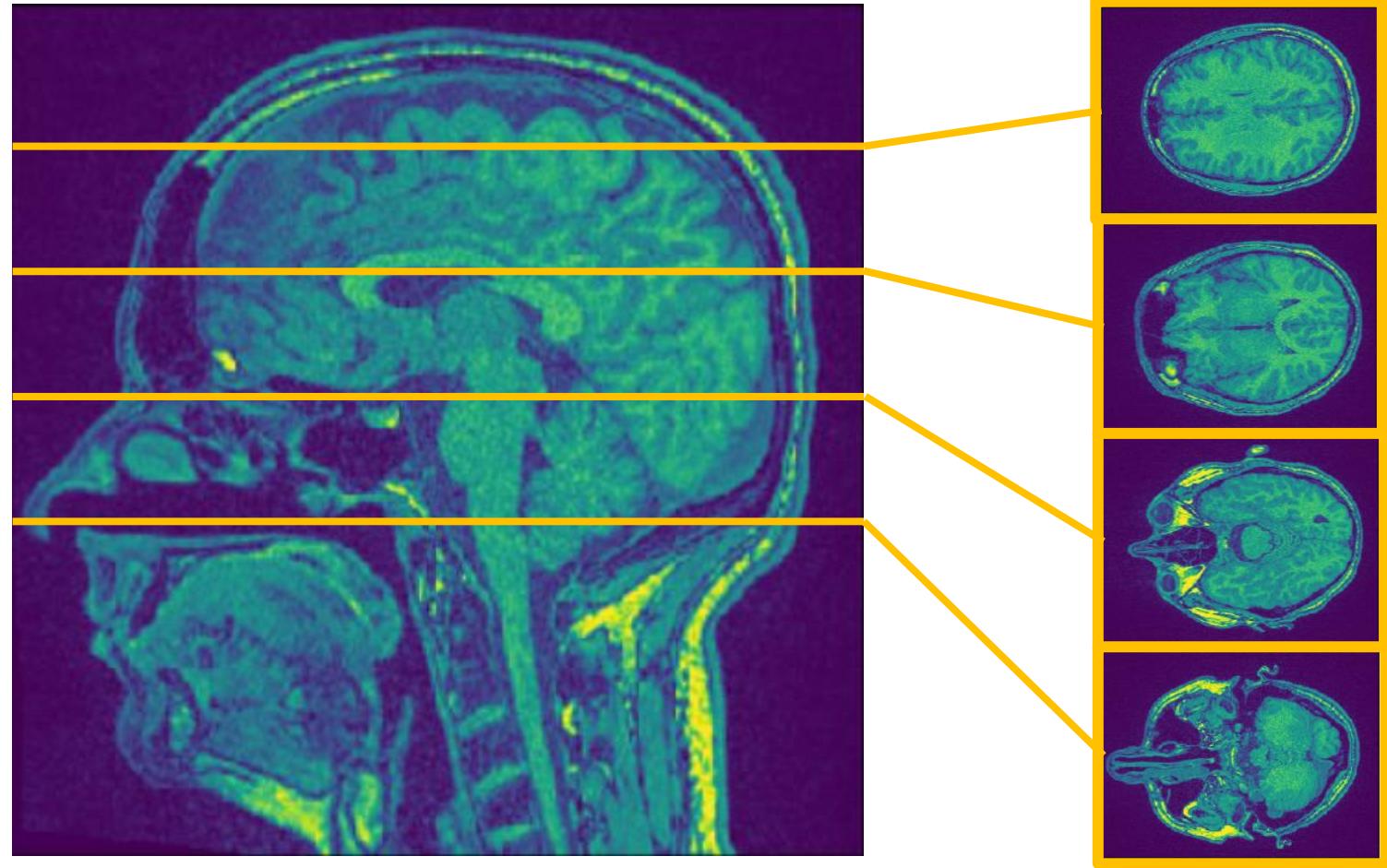
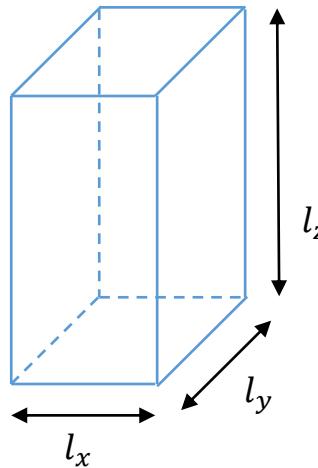
- An image is just a matrix of numbers: pixels: “picture element”
- The edges between pixels are an artefact of the imaging / digitization. They are not real!



| | | | | | | | | | | | | | |
|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 48 | 48 | 48 | 40 | 40 | 32 | 32 | 24 | 24 | 24 | 24 | 24 | 24 | 24 |
| 48 | 48 | 40 | 32 | 32 | 24 | 24 | 16 | 16 | 16 | 16 | 24 | 24 | 24 |
| 48 | 48 | 40 | 32 | 24 | 24 | 16 | 16 | 16 | 16 | 16 | 24 | 24 | 32 |
| 40 | 40 | 32 | 24 | 24 | 16 | 16 | 8 | 16 | 16 | 16 | 24 | 24 | 48 |
| 32 | 32 | 32 | 24 | 24 | 16 | 24 | 24 | 32 | 48 | 56 | 64 | 72 | 88 |
| 24 | 24 | 24 | 16 | 16 | 16 | 24 | 32 | 56 | 72 | 88 | 96 | 112 | 120 |
| 24 | 16 | 16 | 16 | 24 | 32 | 48 | 64 | 96 | 120 | 128 | 144 | 152 | 152 |
| 16 | 8 | 16 | 16 | 32 | 40 | 72 | 96 | 128 | 160 | 176 | 184 | 184 | 184 |
| 16 | 8 | 16 | 24 | 48 | 72 | 104 | 136 | 160 | 176 | 184 | 192 | 192 | 184 |
| 16 | 8 | 24 | 32 | 72 | 104 | 136 | 168 | 184 | 192 | 200 | 200 | 192 | 184 |
| 24 | 24 | 48 | 64 | 104 | 136 | 160 | 184 | 184 | 192 | 192 | 192 | 184 | 184 |
| 32 | 40 | 64 | 88 | 128 | 168 | 184 | 192 | 192 | 184 | 184 | 176 | 176 | 176 |
| 40 | 56 | 88 | 120 | 152 | 192 | 192 | 192 | 192 | 184 | 184 | 176 | 176 | 176 |
| 48 | 64 | 104 | 144 | 176 | 208 | 200 | 184 | 184 | 184 | 176 | 176 | 176 | 168 |

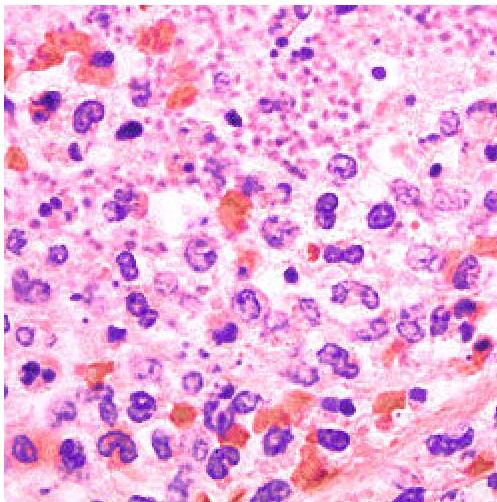
Image stacks and voxels

- 3-dimensional images consisting of voxels
- “Image stack”
- Often *anisotropic* (not equally large in all directions)

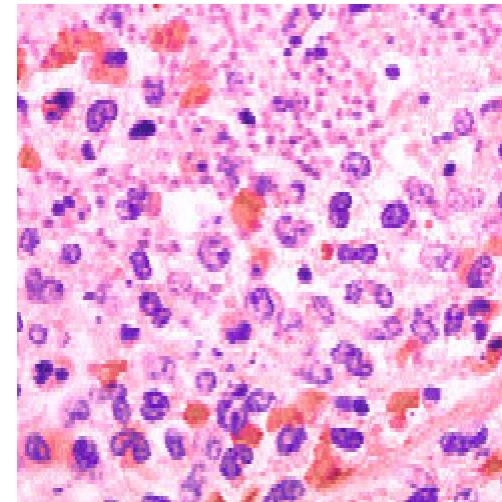


Anisotropy

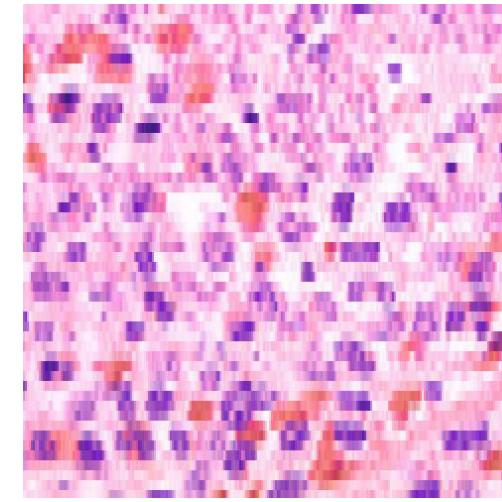
- Voxel size has immediate impact on image quality and thus, on processing / analysis results.



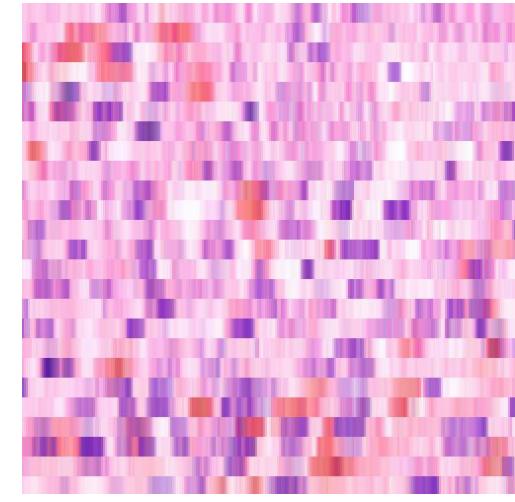
1:1
250 x 250 px



1:2
250 x 125 px



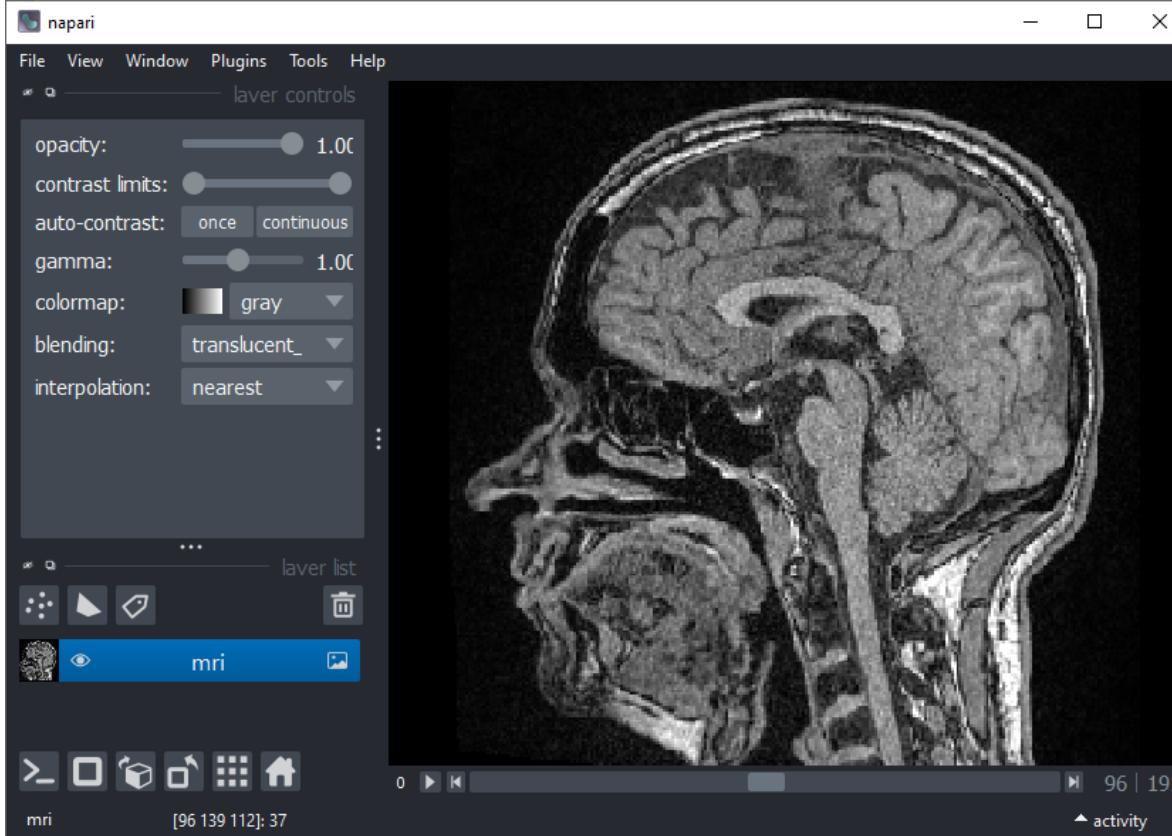
1:5
250 x 50 px



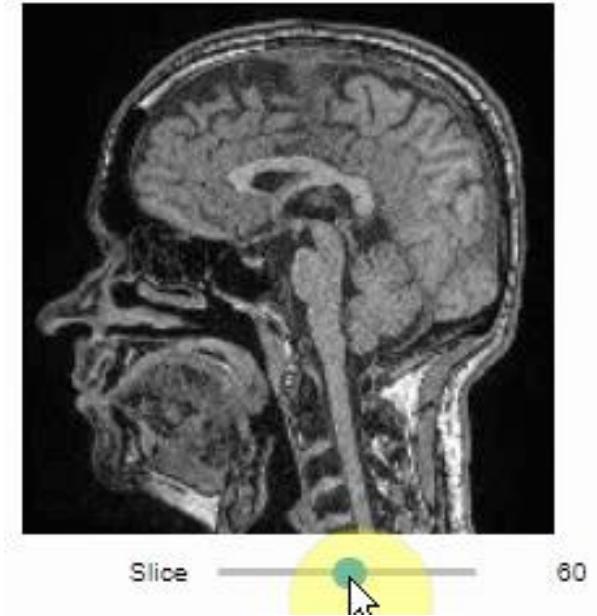
1:10
250 x 25 px

Image stacks and voxels

Interactive tools available

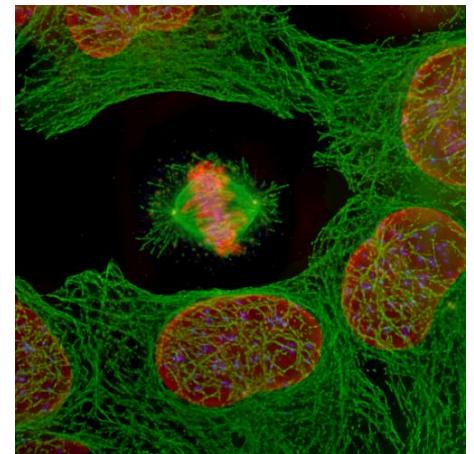
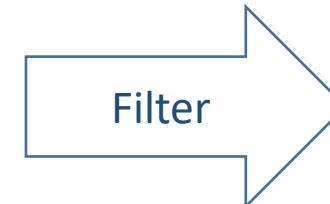
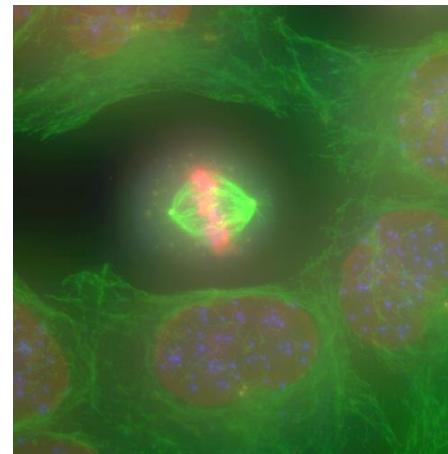


```
import stackview  
  
stackview.slice(mri_image)
```



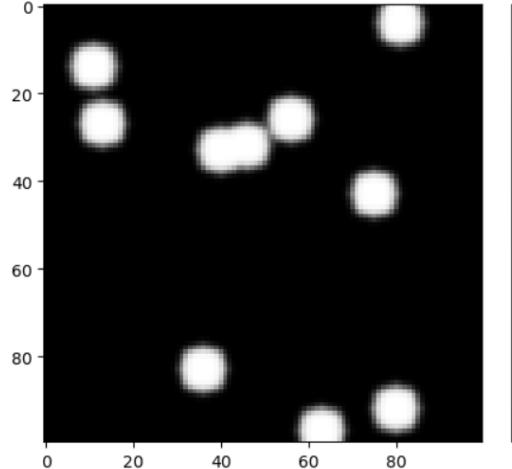
Filters

- An image processing filter is an operation on an image.
- It takes an image and produces a new image out of it.
- 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
 - Artefact-removal
 - Contrast enhancement
 - Correct uneven illumination

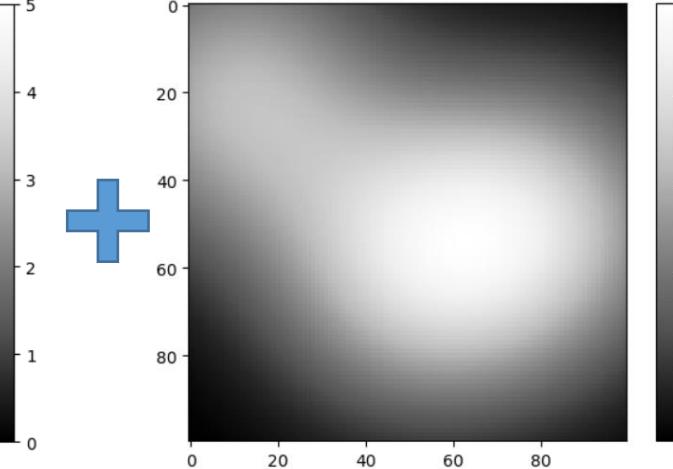


Effects harming image quality

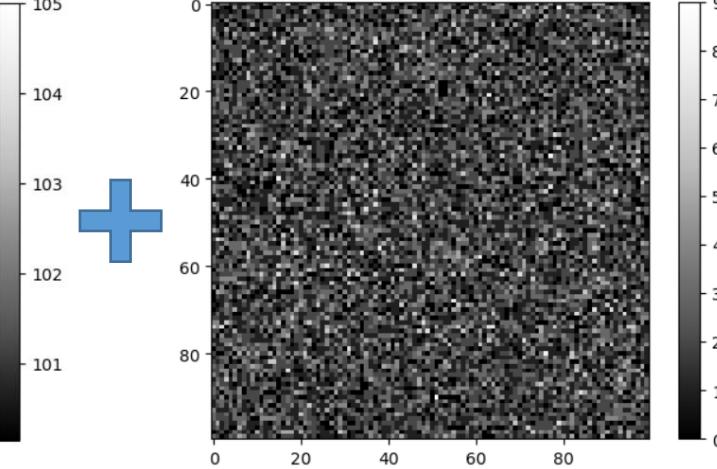
“nuclei”



“background”



“noise”



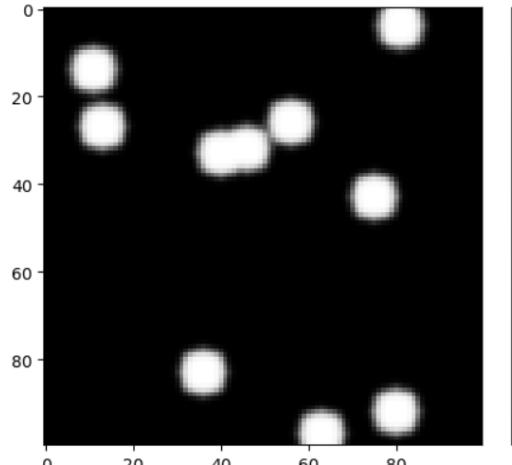
- Aberrations, defocus
- Motion blur

- Light from objects behind and in front of the scene (out-of-focus light)
- Dirt on the object slide
- Camera offset

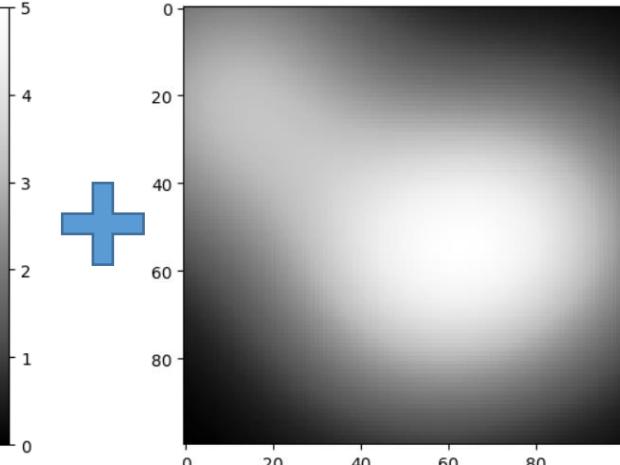
- Shot noise (arriving photons)
- Dark noise (electrons made from photons)
- Read-out-noise (electronics)

Effects harming image quality

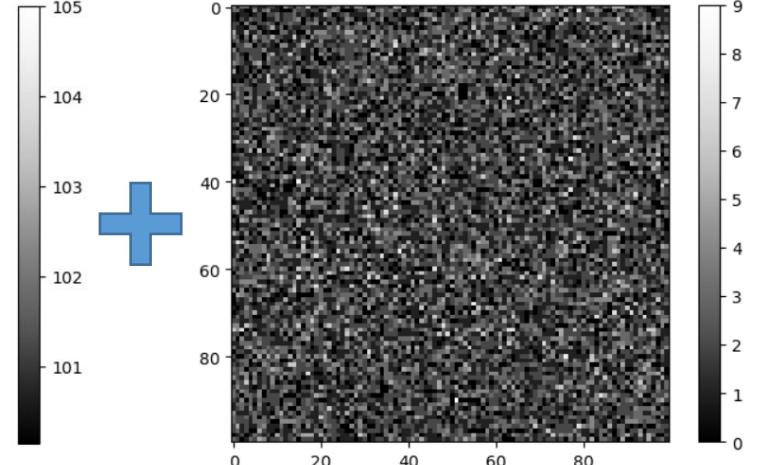
“nuclei”



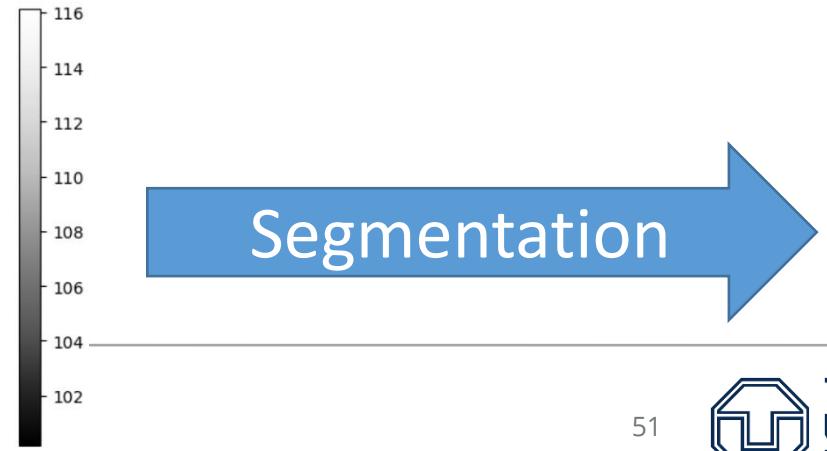
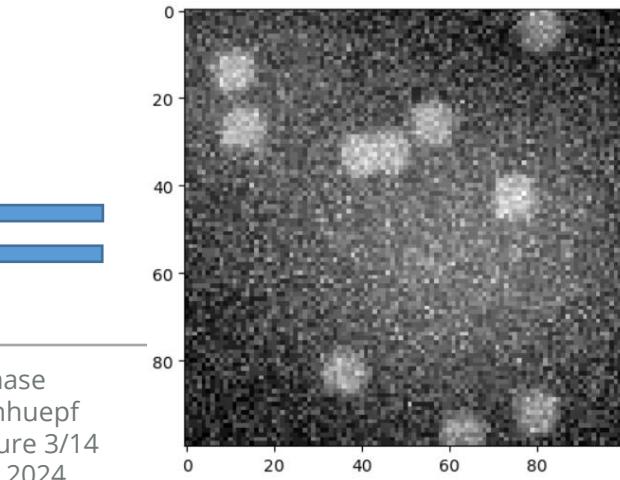
“background”



“noise”



Segmentation



51

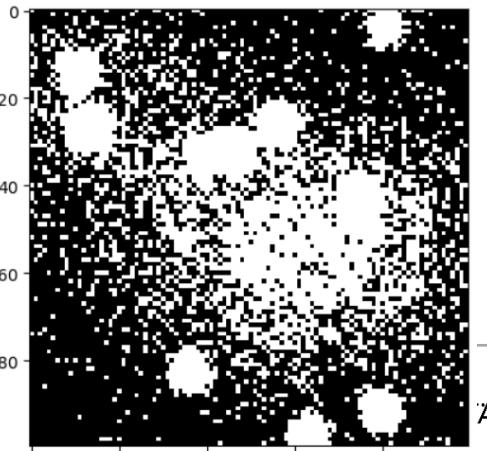


Image filtering

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

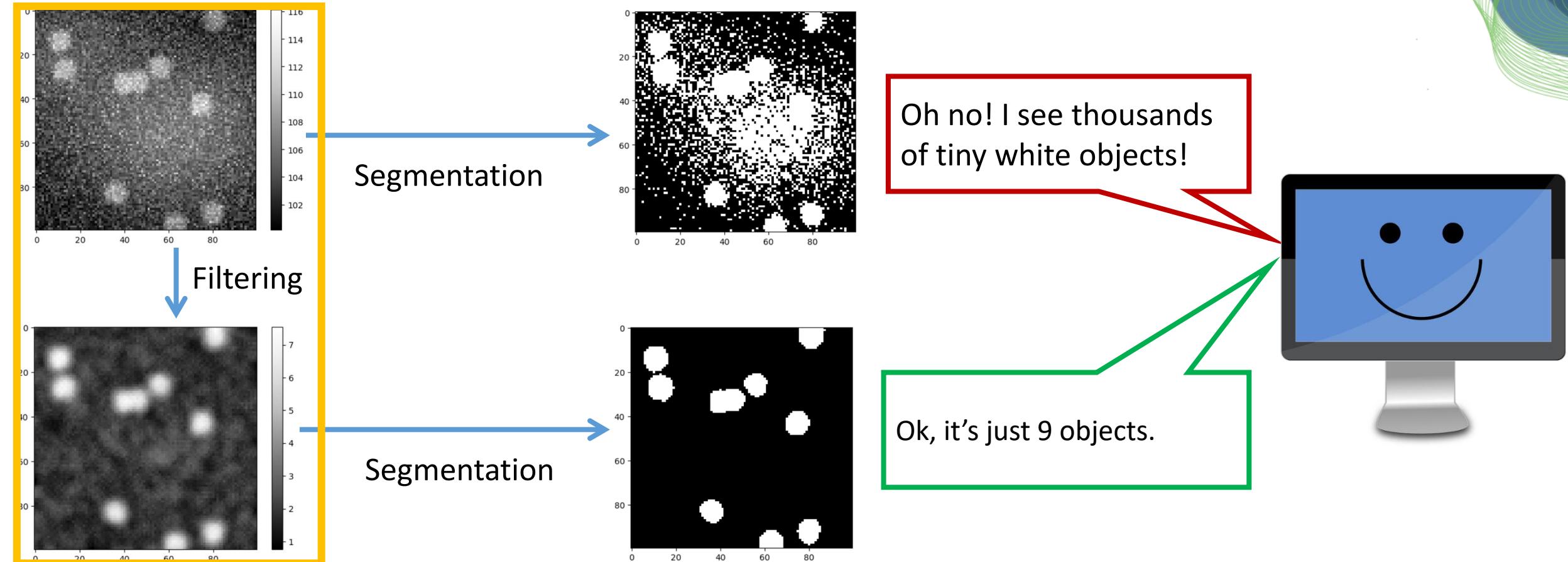


Image filtering

- Attempt to invert / “undo” processes disturbing image quality

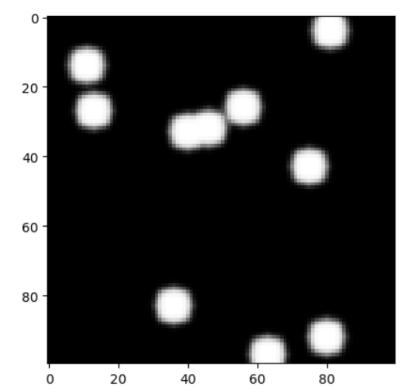
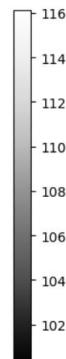
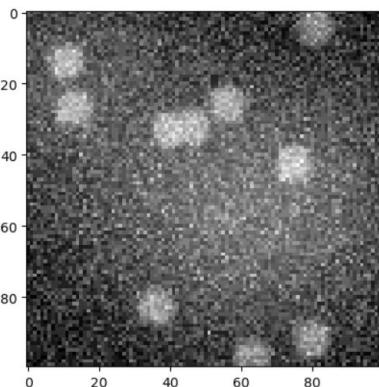
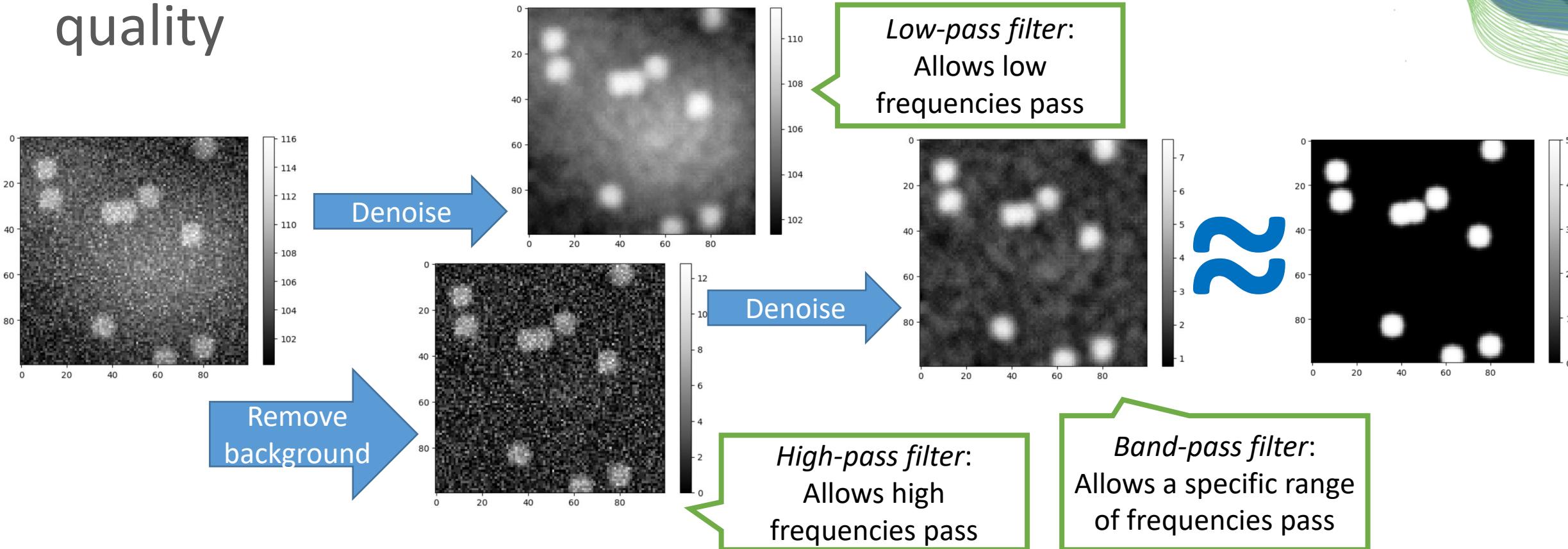


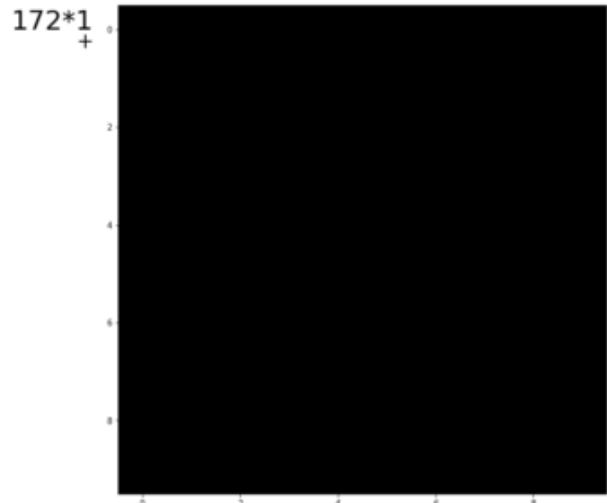
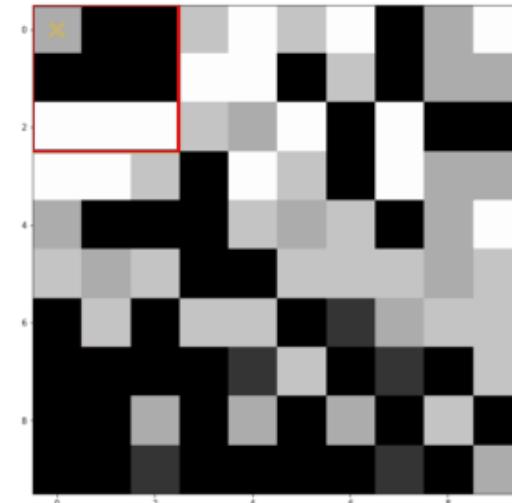
Image filtering

- Attempt to invert / “undo” processes disturbing image quality



Linear Filters

- *Linear filters* replace each pixel value with a weighted linear combination of surrounding pixels
- Filter *kernels* are matrices describing a linear filter
- This multiplication of surrounding pixels according to a matrix is called *convolution*



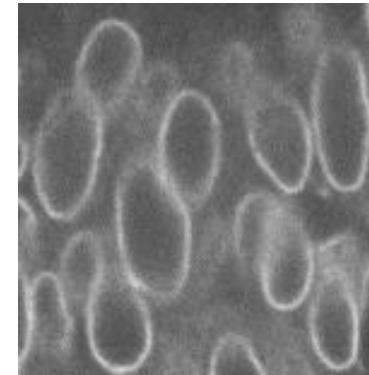
Mean filter, 3x3 kernel

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

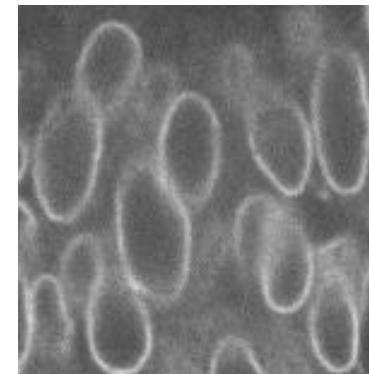
Linear filters

- Terminology:
 - “We convolve an image with a kernel.”
 - Convolution operator: *

- Examples
 - Mean
 - Gaussian blur
 - Sobel-operator
 - Laplace-filter



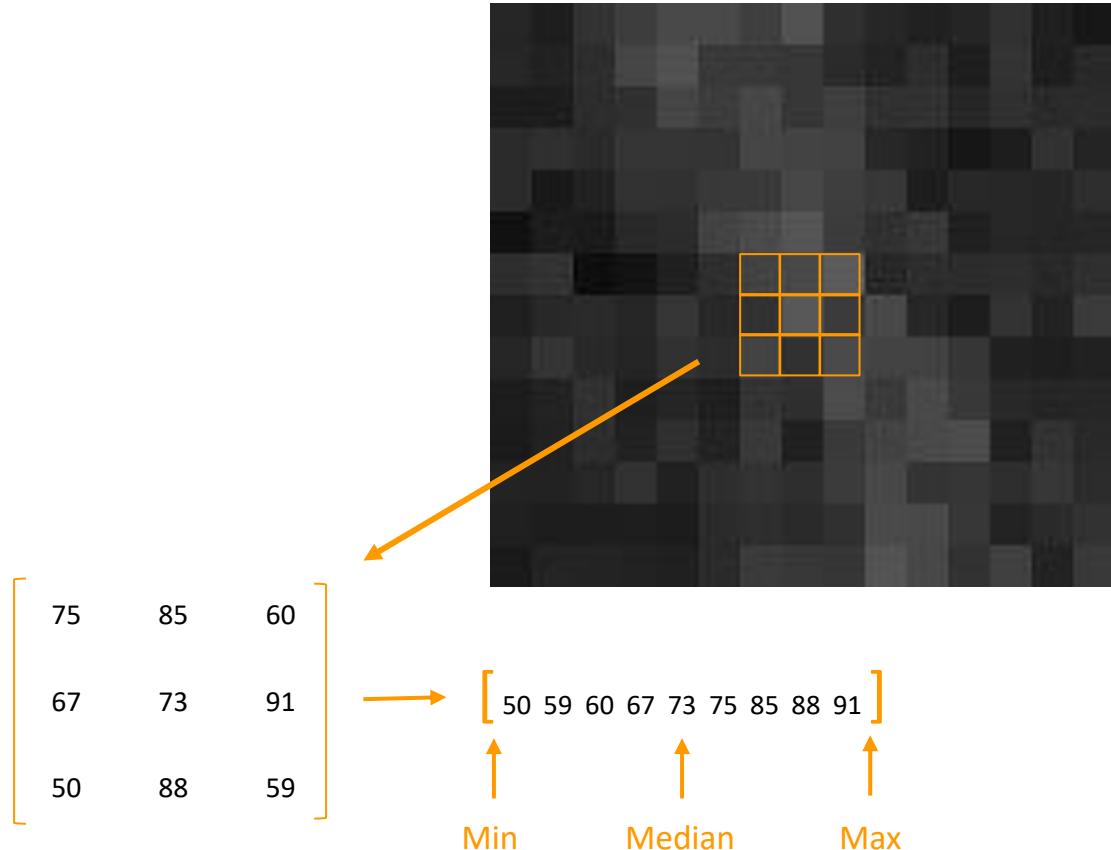
$$\text{Input} \quad * \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \text{Output}$$



$$\text{Input} \quad * \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \text{Output}$$

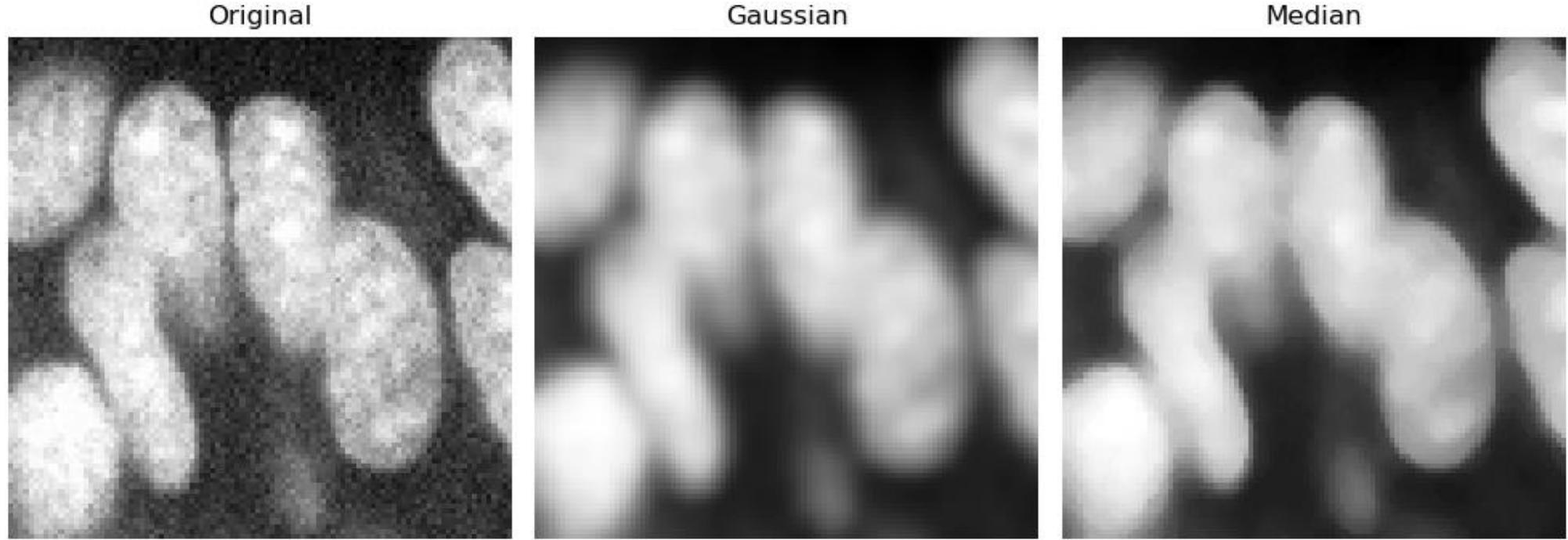
Non-linear Filters

- Non-linear filters also replace pixel value inside a rolling window but using a non-linear function.
- Examples: order statistics filters
 - Min
 - Median
 - Max
 - Variance
 - Standard deviation



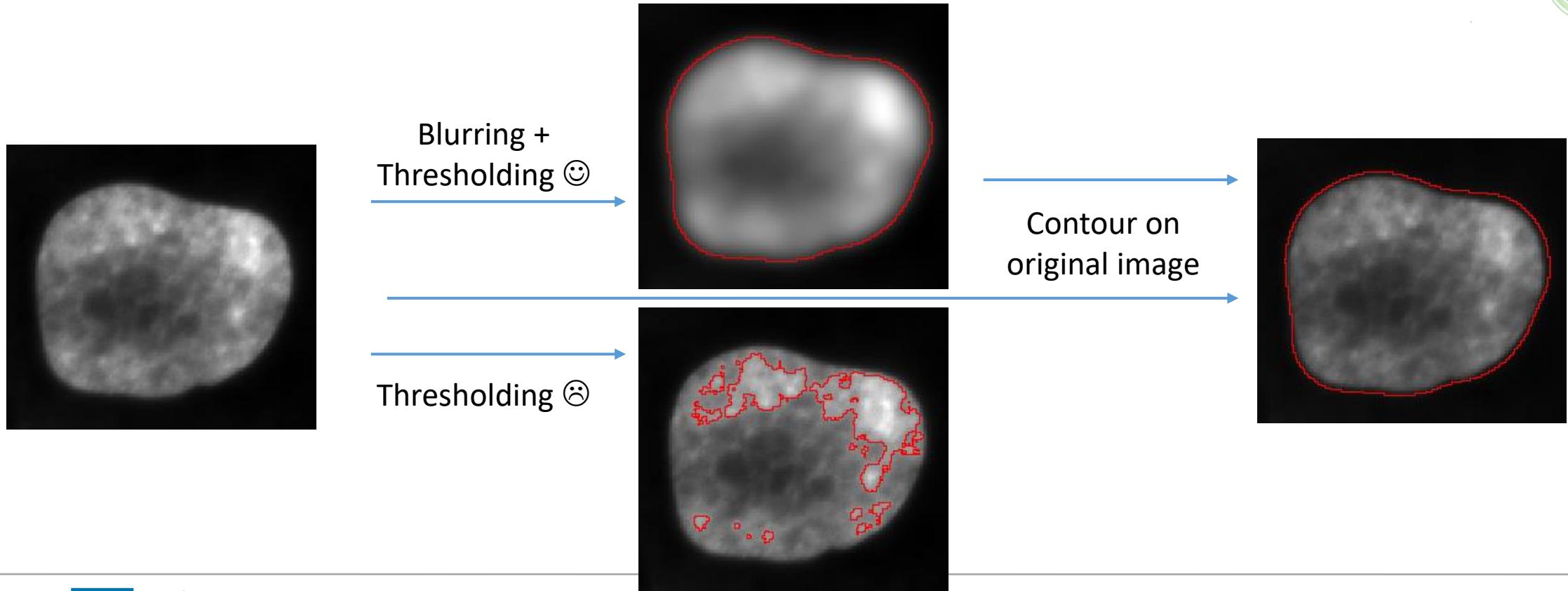
Noise removal

- Gaussian filter
- Median filter (computationally expensive)



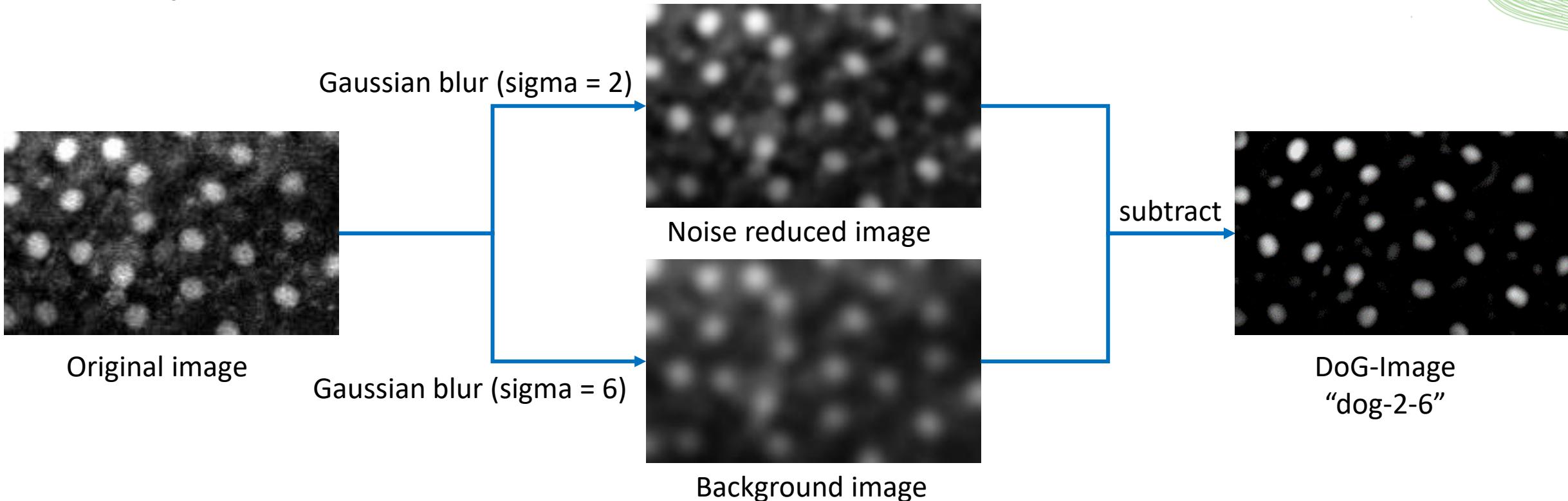
Filtering for improving thresholding results

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



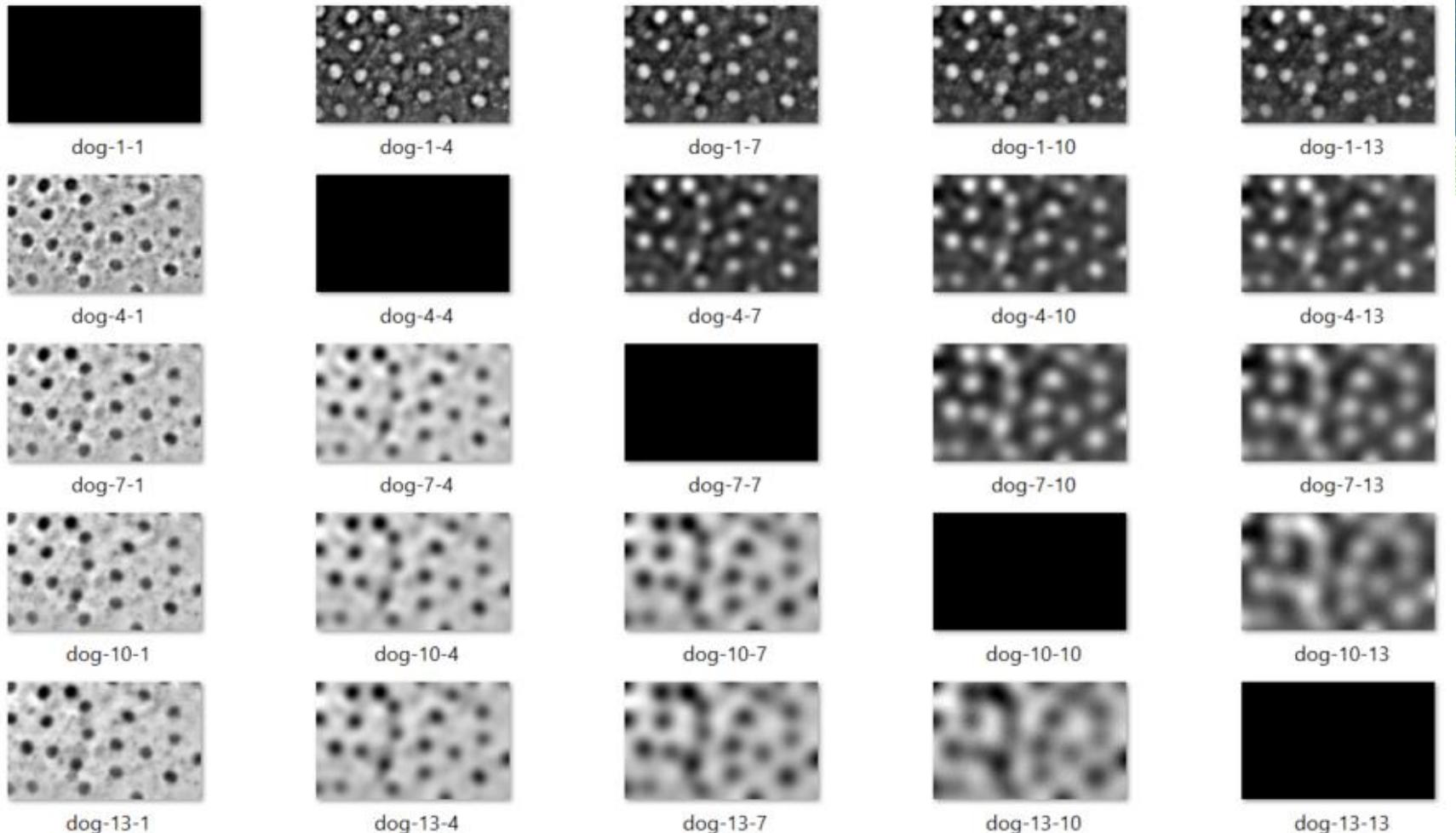
Difference-of-Gaussian (DoG)

- Improve image in order to detect bright objects.
- Band-pass filter



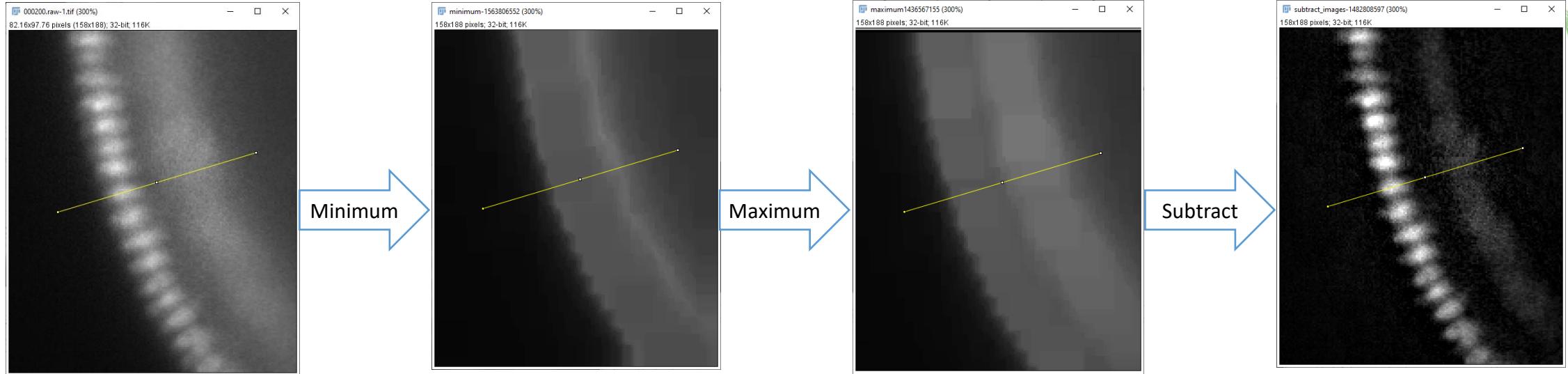
Difference-of-Gaussian (DoG)

- Example DoG images



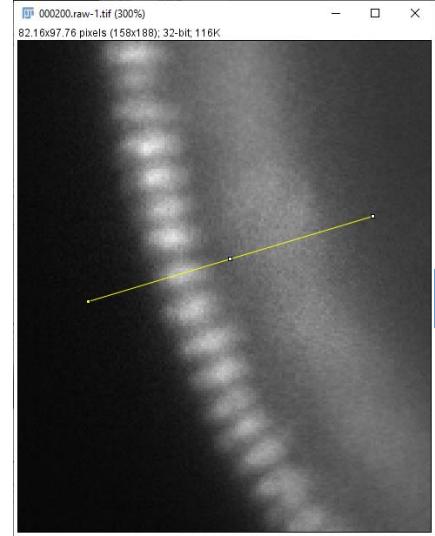
Top-hat filter

- Background subtraction

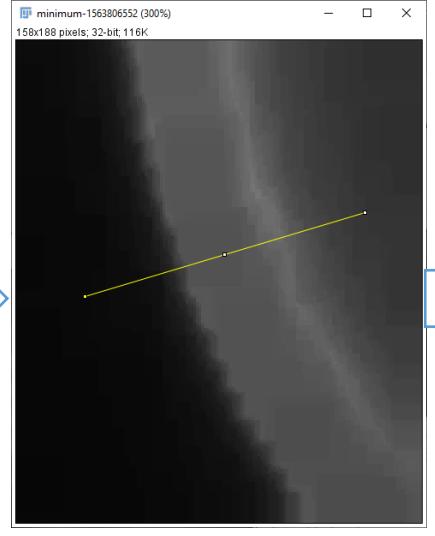


Top-hat filter

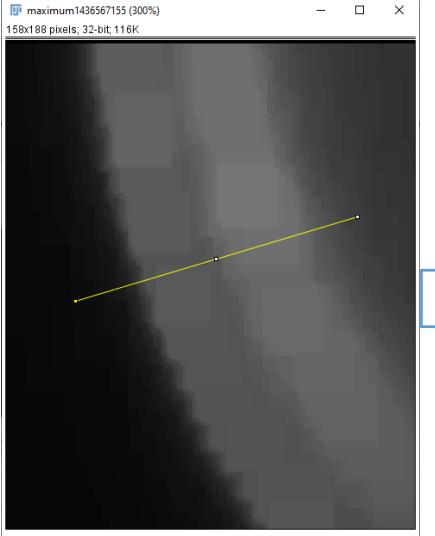
- Background subtraction



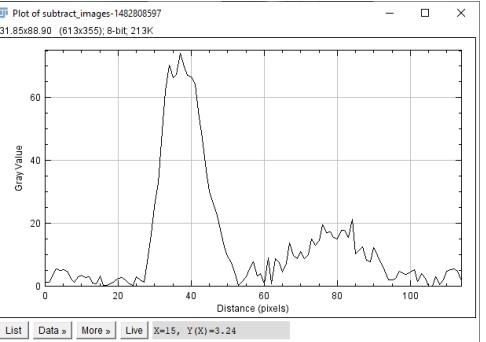
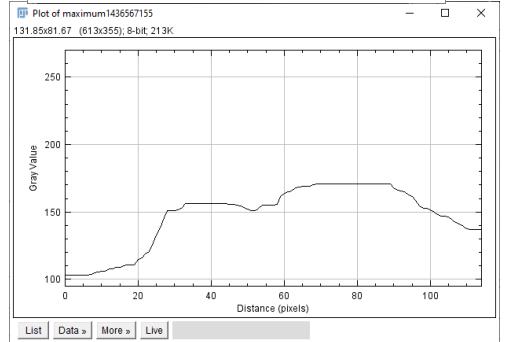
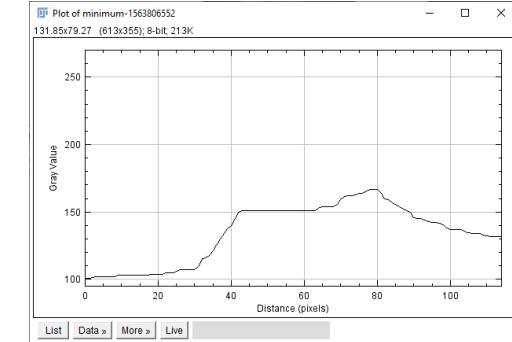
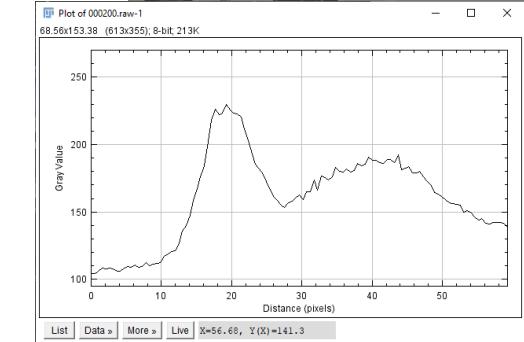
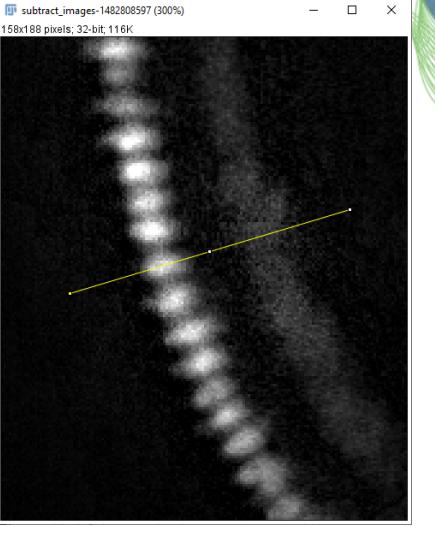
Minimum



Maximum



Subtract

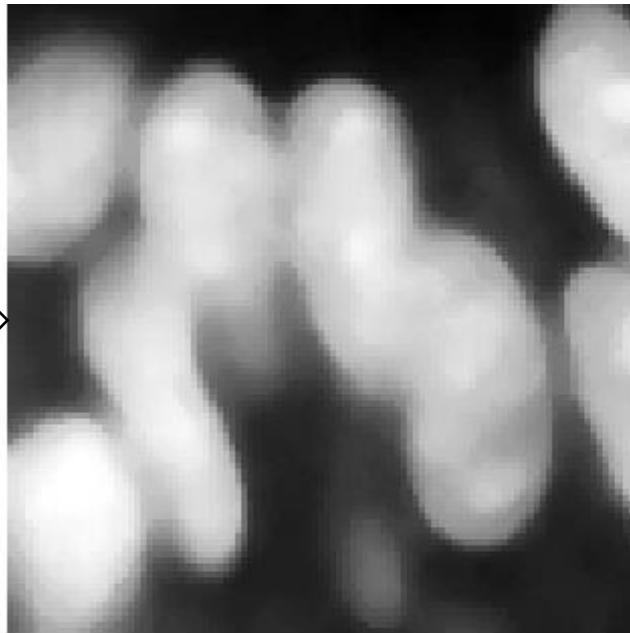


Quiz: Noise removal

- The median filter is a ...



Median

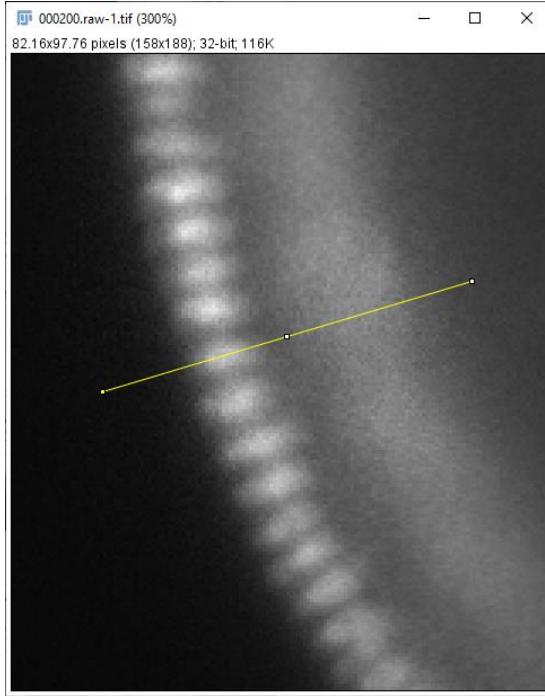


Linear filter

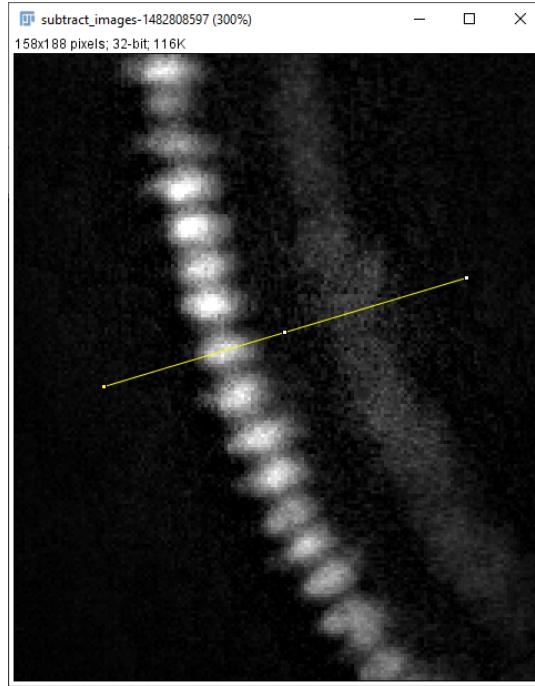
Non-linear filter

Background removal

- Removing background from an image is a ... ?



Top-hat



Low-pass
filter

High-pass
filter



DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Exercises

Robert Haase

GEFÖRDERT VOM



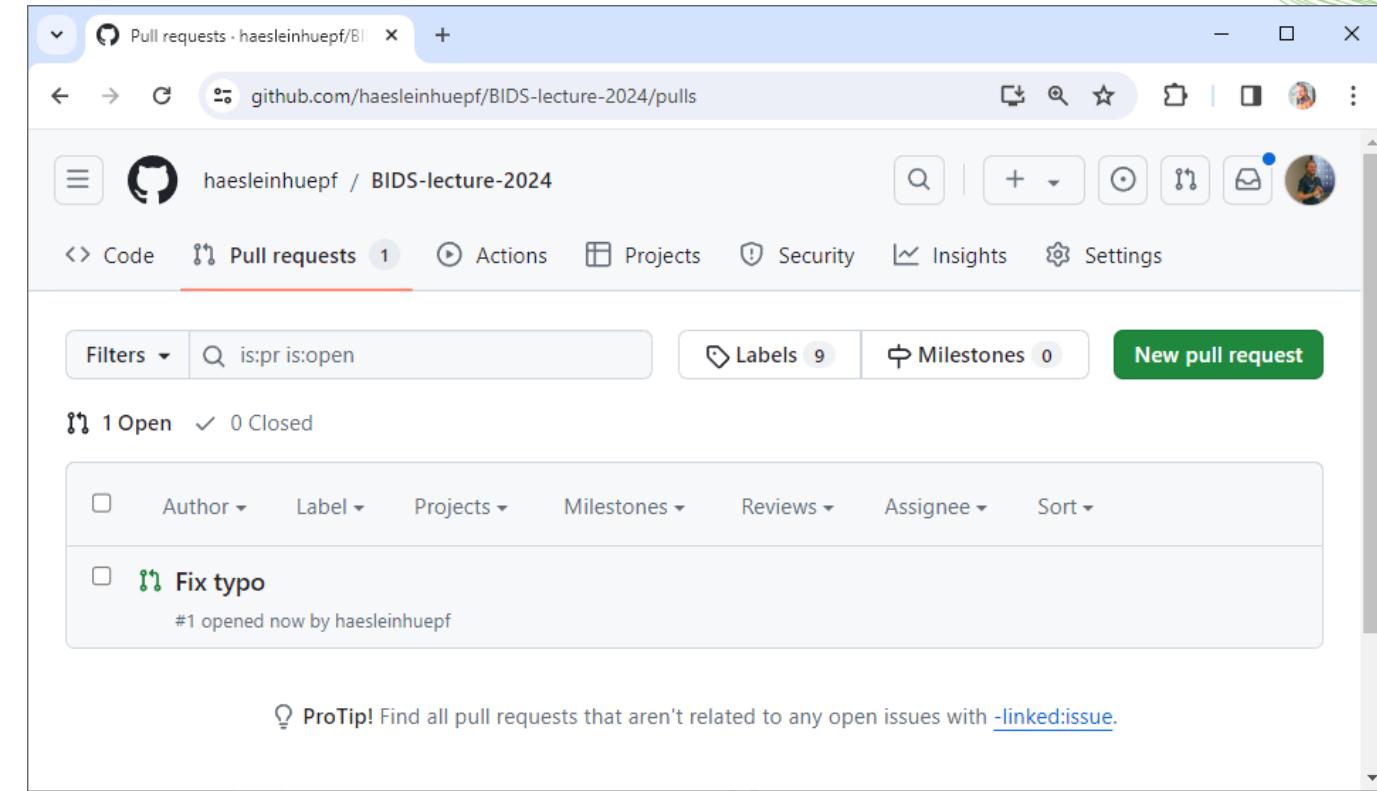
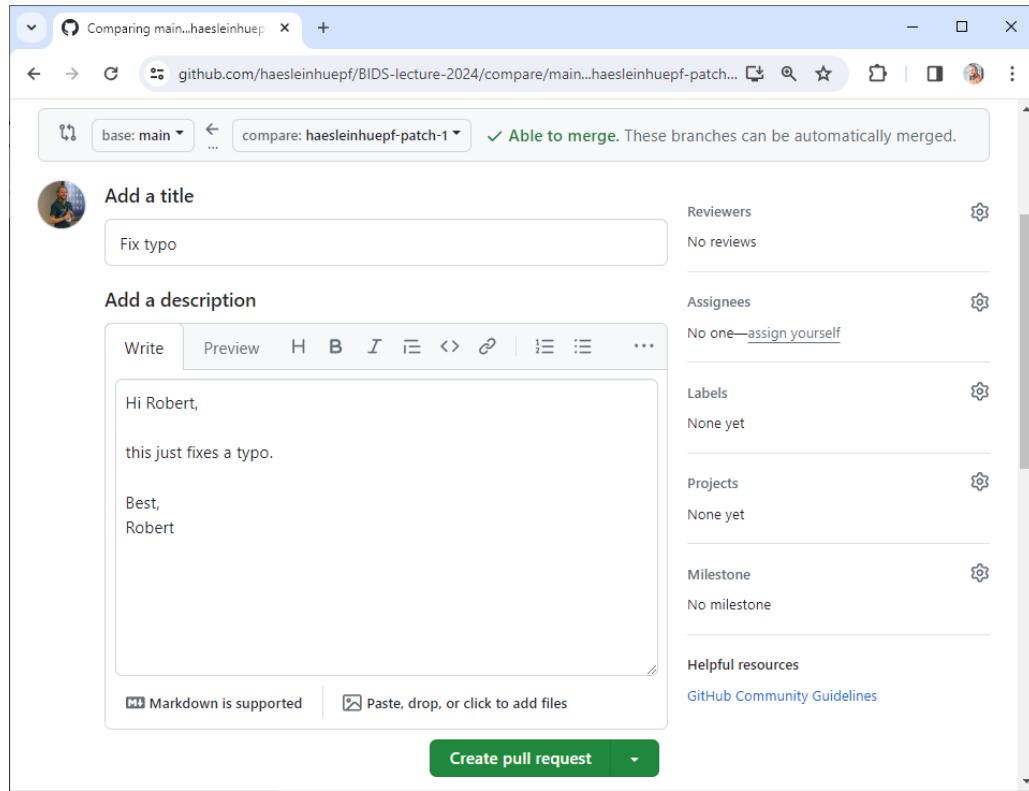
Bundesministerium
für Bildung
und Forschung



Diese Maßnahme wird gefördert durch die Bundesregierung
aufgrund eines Beschlusses des Deutschen Bundestages.
Diese Maßnahme wird mitfinanziert durch Steuermittel auf
der Grundlage des von den Abgeordneten des Sächsischen
Landtags beschlossenen Haushaltes.

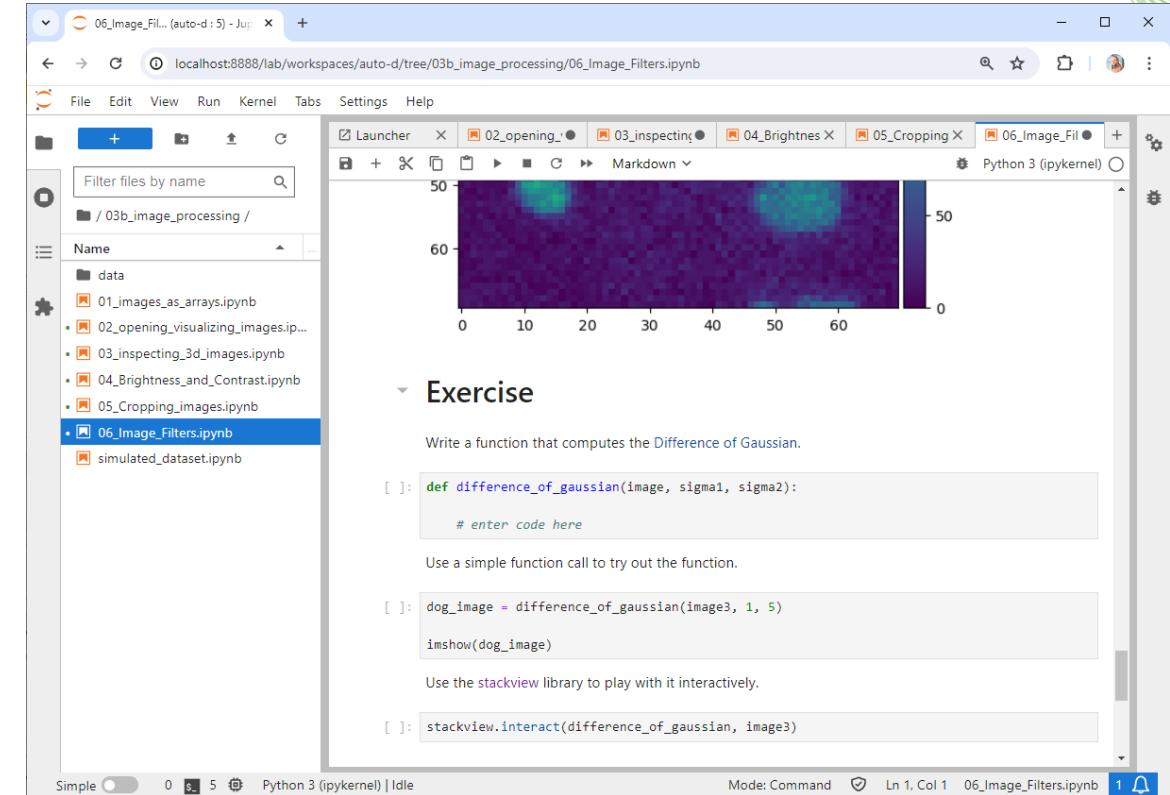
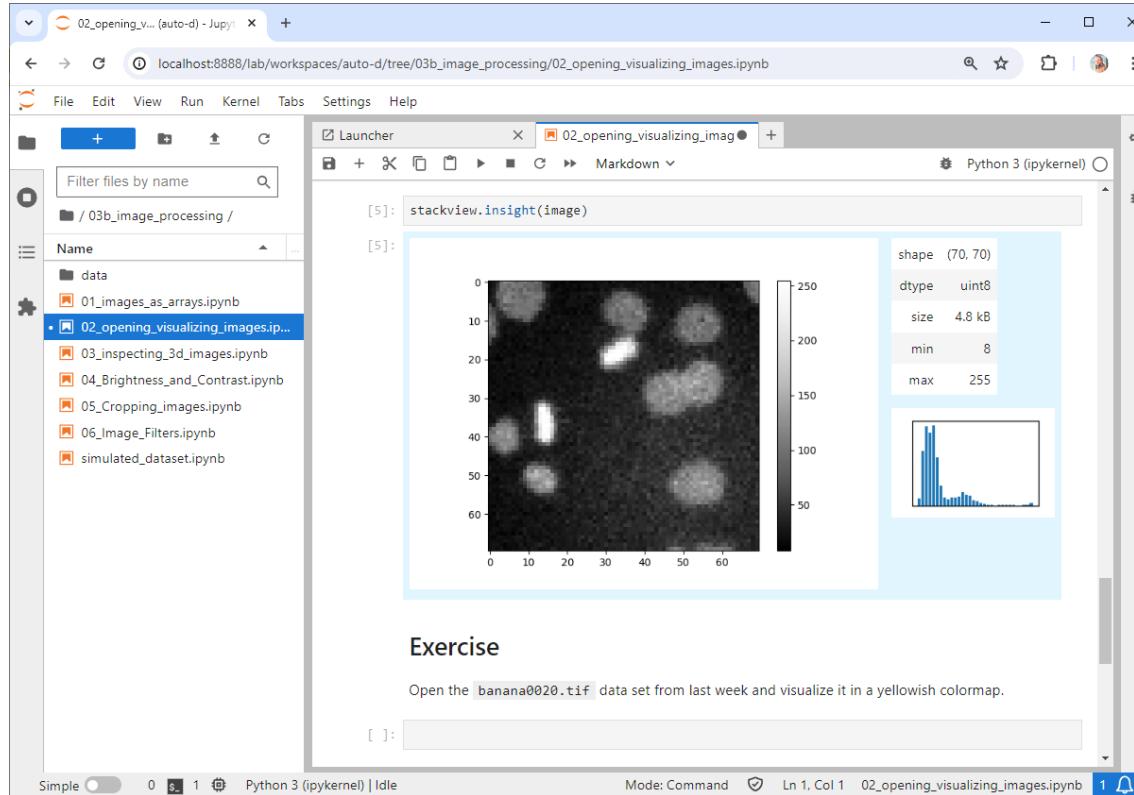
Exercise: pull-request

- Clone the training materials repository
- Fix the typo on this page, send a pull-request



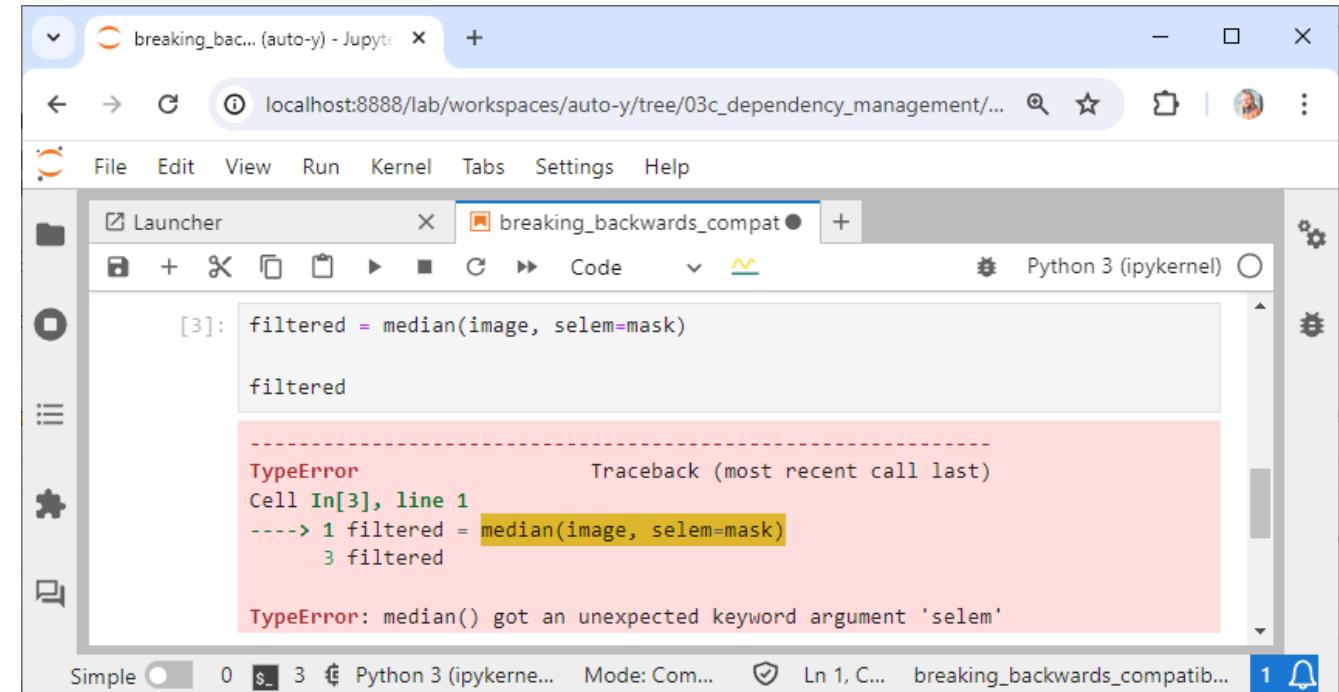
Exercise: image processing

- Get started with loading, viewing, cropping and processing images



Exercise: dependencies

- There is a Jupyter Notebook which doesn't work (anymore). Find out why.
- Fix it in two ways:
 - A) by changing the code
 - B) by not changing the code



A screenshot of a Jupyter Notebook interface. The title bar shows "breaking_bac... (auto-y) - Jupyter". The URL in the address bar is "localhost:8888/lab/workspaces/auto-y/tree/03c_dependency_management/breaking_backwards_compatibility.ipynb". The menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The left sidebar has a "Launcher" tab. The main area shows a code cell [3]:

```
filtered = median(image, selem=mask)
```

The output cell shows:

```
filtered
```

Below the code cell, a red box highlights the error:

```
-----  
TypeError                                     Traceback (most recent call last)  
Cell In[3], line 1  
----> 1 filtered = median(image, selem=mask)  
      3 filtered  
  
TypeError: median() got an unexpected keyword argument 'selem'
```