



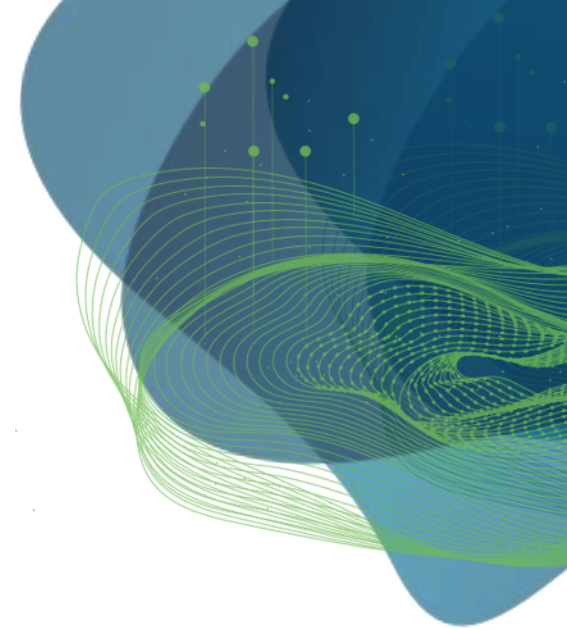
ScaDS.AI

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 and CC-BY-SA, what license do I have to use?

CC-BY



CC-BY-SA



CC-BY-ND



(not possible)



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

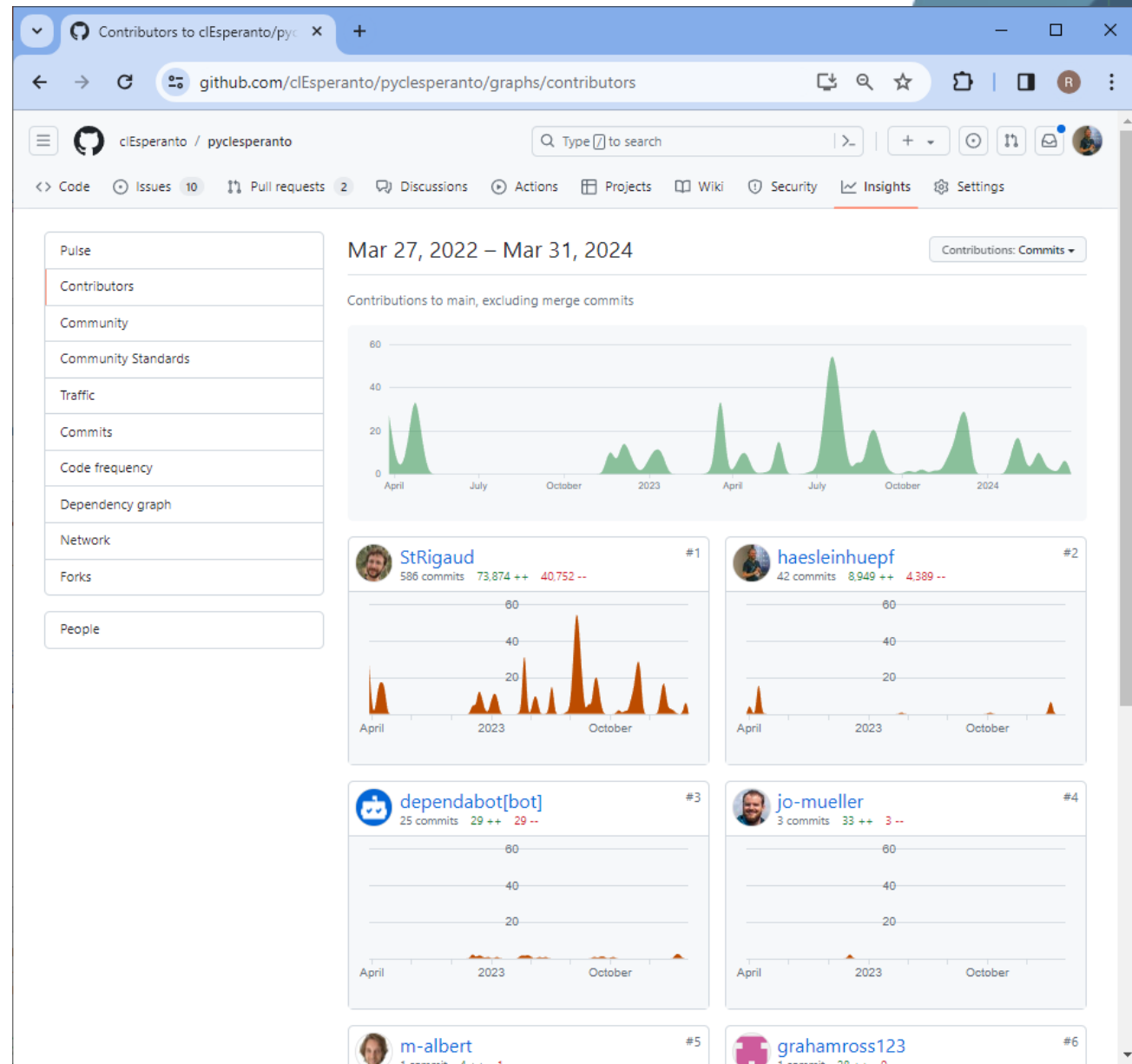
[Michelle Barker](#) , [Neil P. Chue Hong](#), [Daniel S. Katz](#), [Anna-Lena Lamprecht](#), [Carlos Martinez-Ortiz](#), [Fotis Psomopoulos](#), [Jennifer Harrow](#), [Leyla Jael Castro](#), [Morane Gruenpeter](#), [Paula Andrea Martinez](#) & [Tom Honeyman](#)

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

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

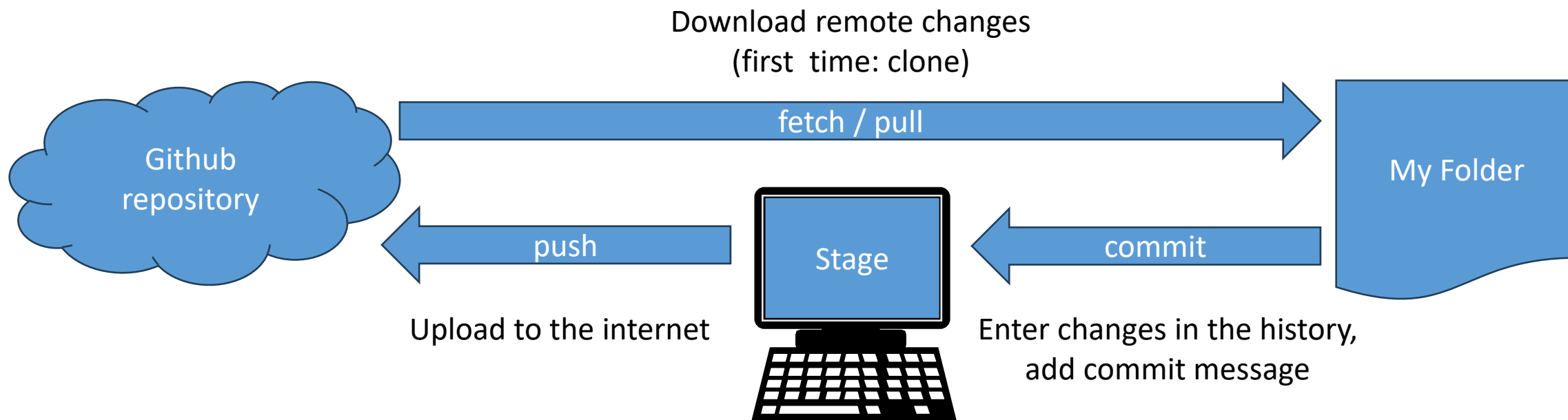
git

- 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 image shows two browser windows from GitHub. The left window displays the commit history for the repository `haesleinhuepf/example_image_analysis_script`. The commit list includes:

- Merge pull request #1 from BiAPoL/main
- bugfix: threshold_otsu** (highlighted with an orange arrow)
- Add minimal working example
- add gitignore

The right window shows the diff for the commit `bugfix: threshold_otsu`. It displays changes to the file `my_library.py`. The diff shows a function `segment_image` with several lines of code. Lines 9 and 10 are highlighted in green, indicating additions:

```
@@ -6,7 +6,8 @@ def segment_image(image):
6 6     blurred = gaussian(image, sigma=2)
7 7
8 8     # binarize the image
9 +     binary = threshold_otsu(blurred)
10 +    threshold = threshold_otsu(blurred)
11     binary = blurred > threshold
12     # label connected components
13     result = label(binary)
```


git

- History
- Track recent changes

The screenshot displays the GitHub interface for the repository 'ScaDS / BIDS-lecture-2024'. On the left, the 'Commits' section shows a list of recent commits on the 'main' branch, including 'add backwards compatibility exercise', 'move pull-request exercise from week 2 t', and 'fix issue with ../../ folder locations'. A blue arrow points to the 'fix issue with ../../ folder locations' commit. The main area shows a diff view for the commit 'c5dc3bf8b6...', displaying changes to the file '02a_remote_files/exploring_bioimage_archive.ipynb'. The diff shows that lines 236, 237, and 238 were removed (indicated by a red background) and replaced with new code (indicated by a green background). The removed code used relative paths like '../..' for folder locations, while the new code uses absolute paths like 'data/'.

Line	Change	Code
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

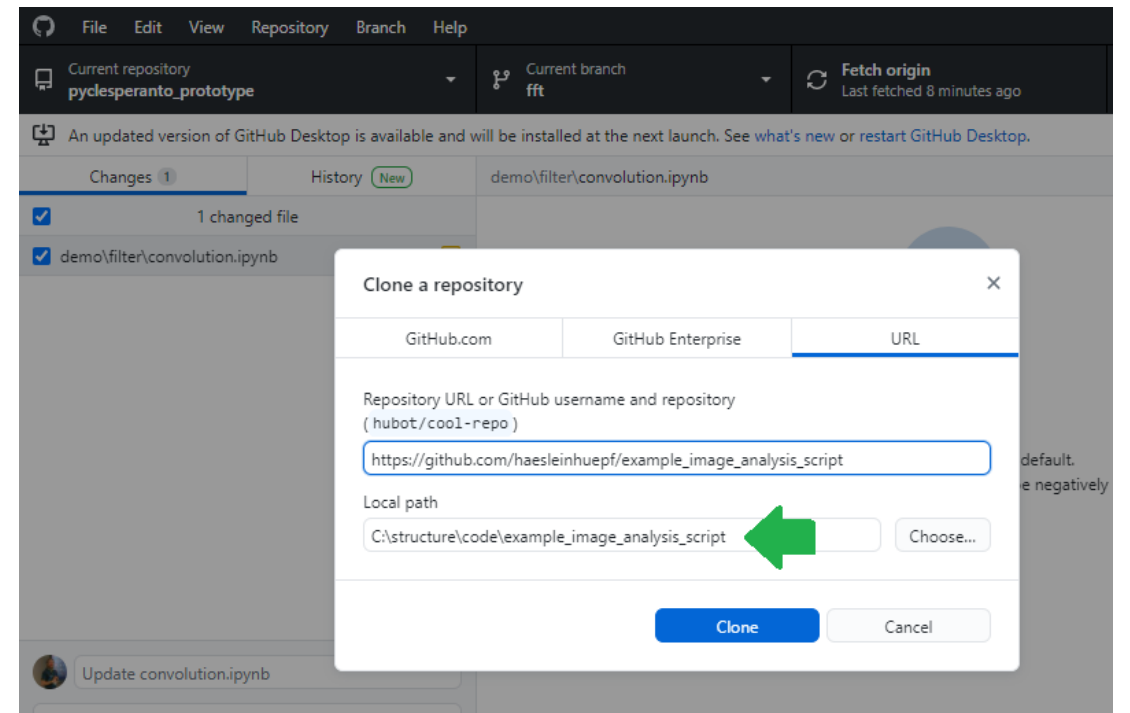
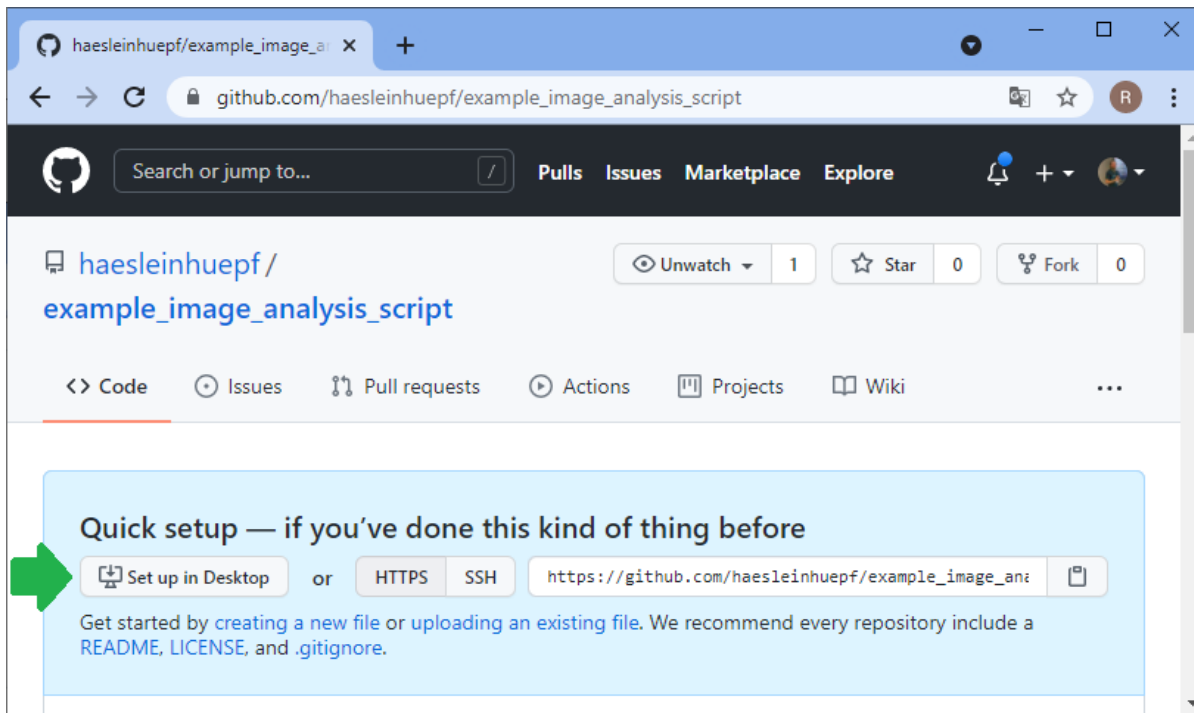
- Add a new, empty repository

The image displays three overlapping browser windows illustrating the steps to create a new repository on GitHub:

- Top-left window:** Shows the GitHub profile page for Robert Haase. A green arrow points to the 'Repositories' tab, which shows 164 repositories.
- Top-right window:** Shows the 'Create a New Repository' page. A green arrow points to the 'Repository name' field, which contains 'example_image_analysis_script' and is marked as available.
- Bottom-right window:** Shows the 'Create a New Repository' page with the 'Create repository' button highlighted by a green arrow.

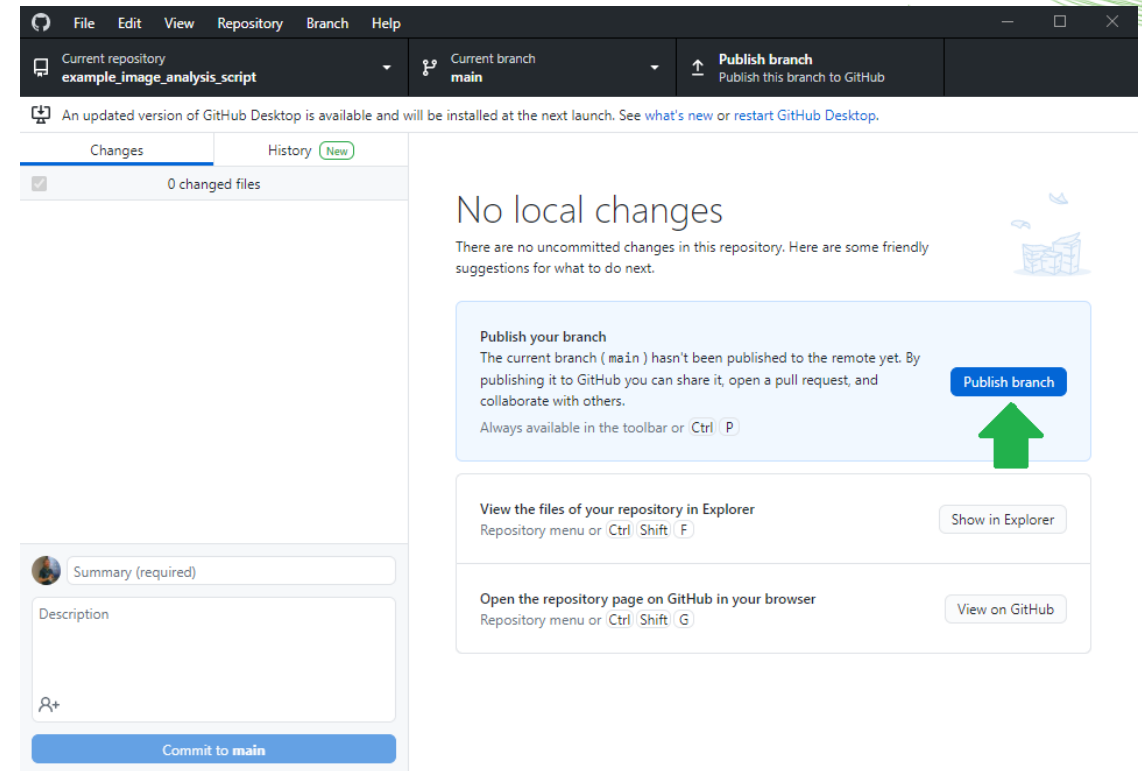
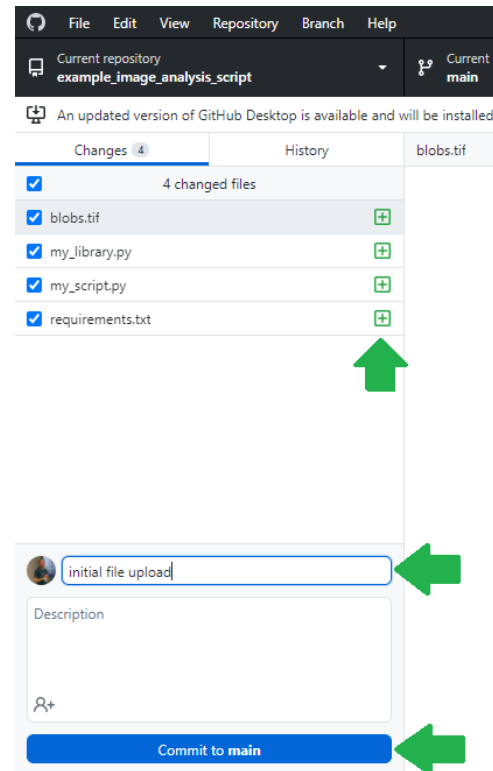
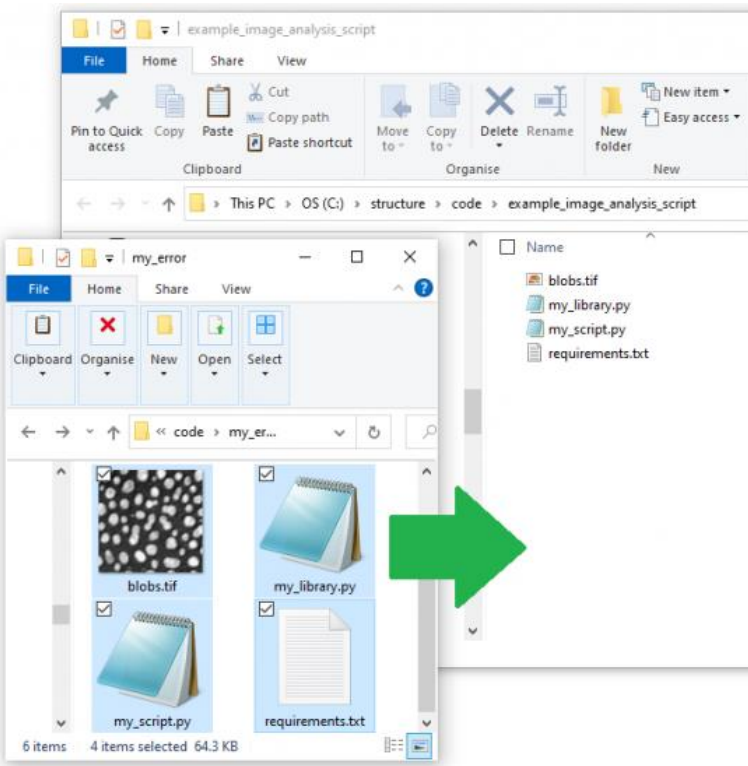
github – clone repositories

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



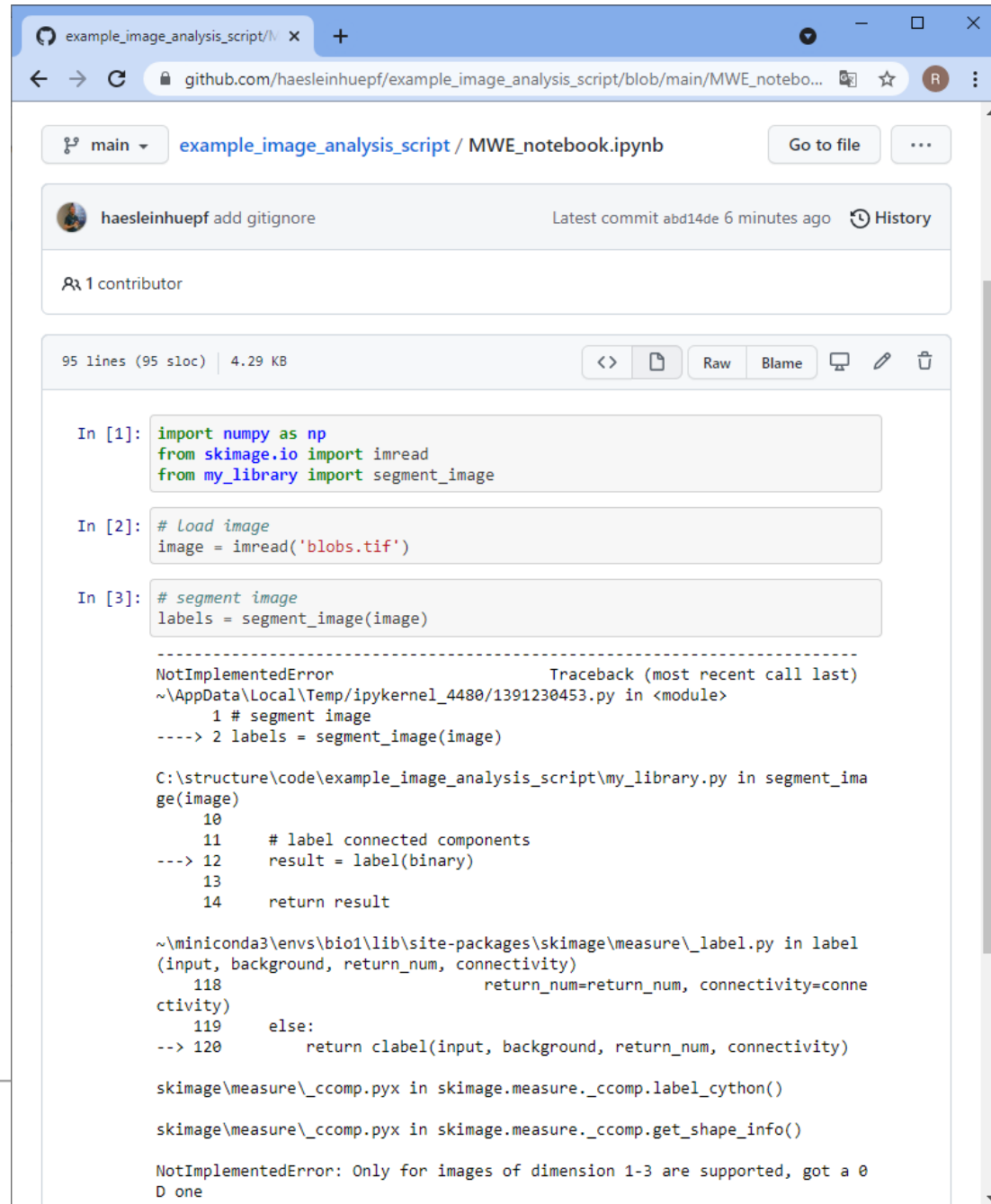
github - uploading

- git [add], commit, push



github

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



The screenshot shows a GitHub repository page for 'example_image_analysis_script' by user 'haesleinhuepf'. The file 'MWE_notebook.ipynb' is displayed, containing three code cells. The first cell imports 'numpy', 'skimage.io', and 'my_library'. The second cell loads an image from 'blobs.tif'. The third cell attempts to segment the image. The output shows a 'NotImplementedError' with a traceback pointing to 'my_library.py' in the 'segment_image' function, which calls 'label(binary)'. The error message states: 'NotImplementedError: Only for images of dimension 1-3 are supported, got a 0 D one'.

```
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(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)

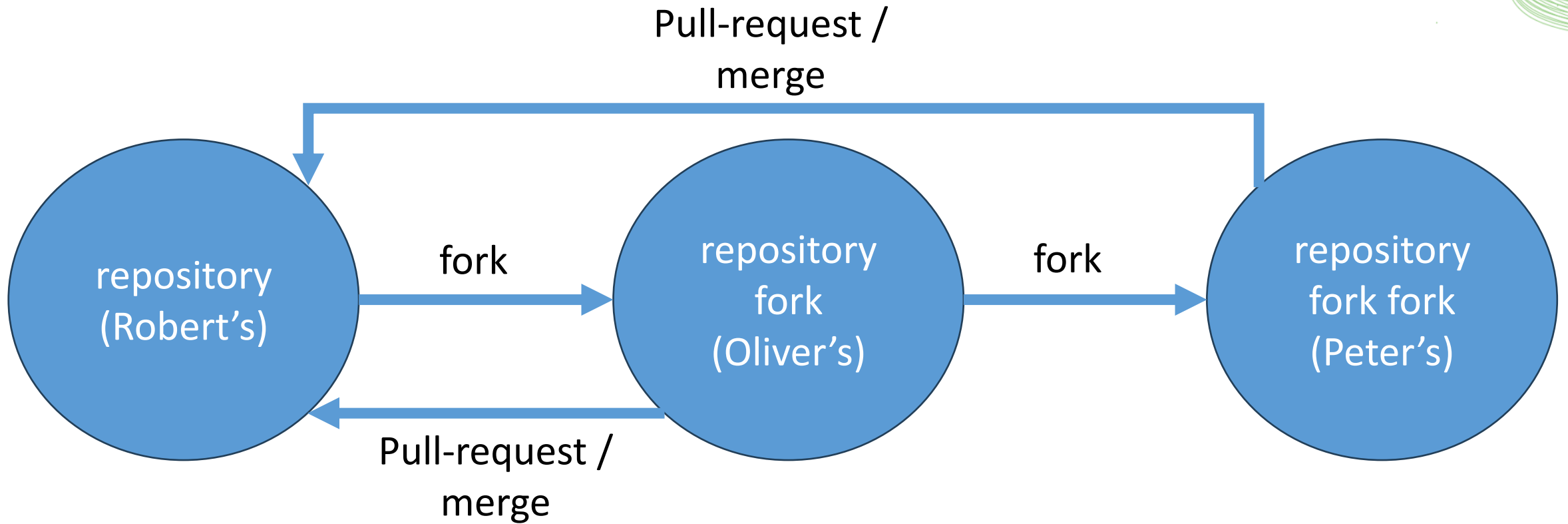
skimage\measure\_ccomp.pyx in skimage.measure._ccomp.label_cython()

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

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

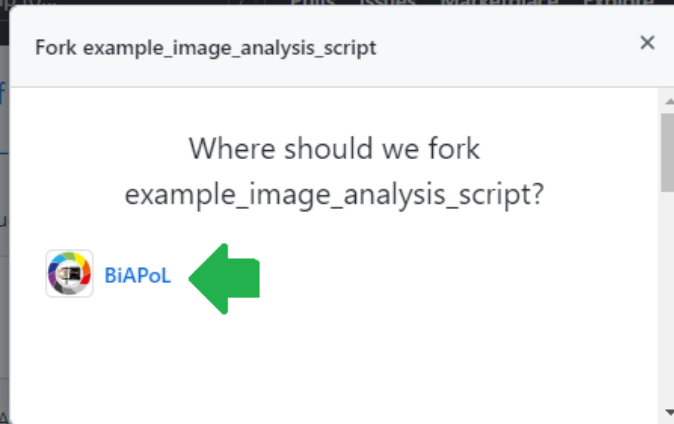
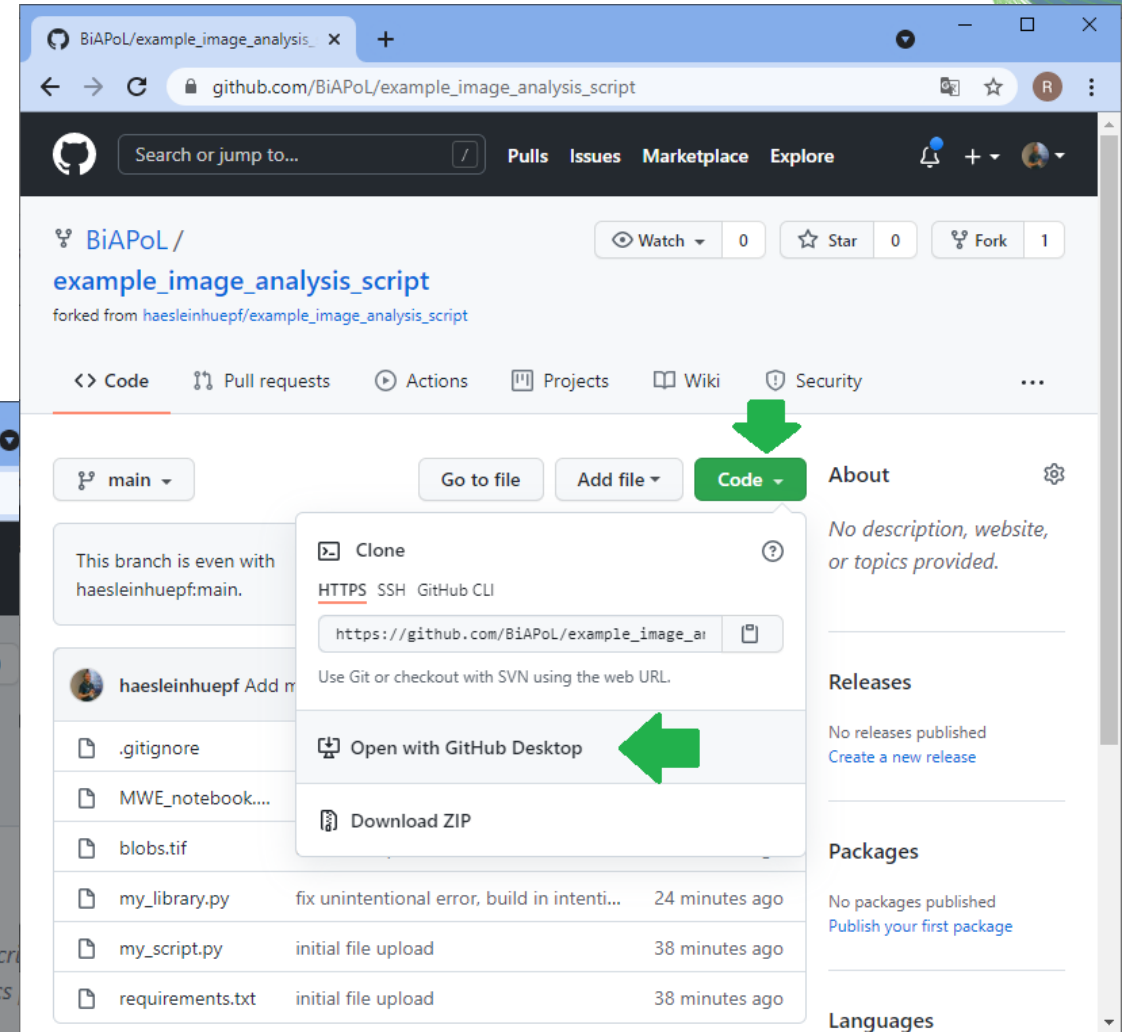
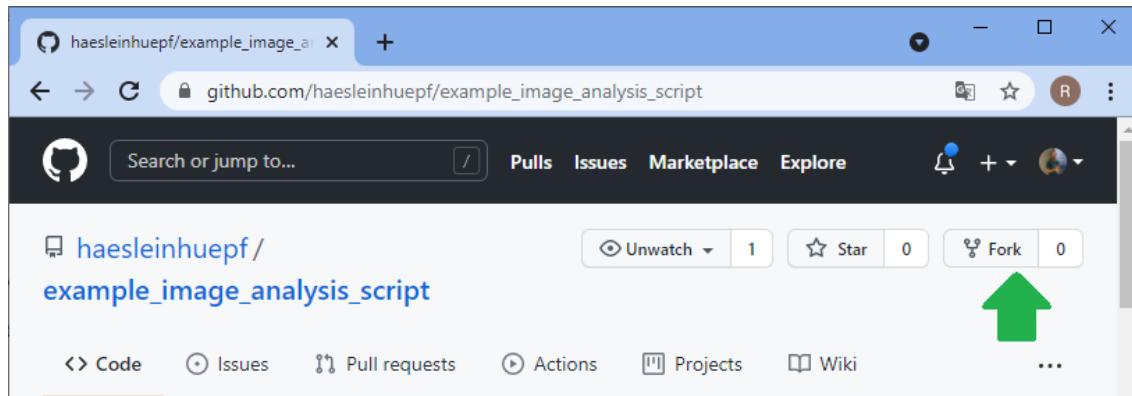
git - forking

- 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

File Edit View Repository Branch Help

Current repository: example_image_analysis_script | Current branch: main | Fetch origin: Never fetched

Changes: 2 changed files

- MWE_notebook.ipynb
- my_library.py

```
@@ -6,7 +6,8 @@ def segment_image(image):
6 6     blurred = gaussian(image, sigma=2)
7 7
8 8     # binarize the image
9 -     binary = threshold_otsu(blurred)
9 +     threshold = threshold_otsu(blurred)
10 +     binary = blurred > threshold
10 11
11 12     # label connected components
12 13     result = label(binary)
@@ @@
```

Commit message: bugfix: threshold_otsu

Commit description: threshold_otsu delivers a number (the threshold), not a binary image. For thresholding the image, an additional step is necessary.

Commit to main

File Edit View Repository Branch Help

Current repository: example_image_analysis_script | Current branch: main | Push origin: Last fetched just now

Changes: 0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote

You have 1 local commit waiting to be pushed to GitHub.

Always available in the toolbar when there are local commits waiting to be pushed or **Ctrl | P**

View the files of your repository in Explorer

Repository menu or **Ctrl | Shift | F**

Open the repository page on GitHub in your browser

Repository menu or **Ctrl | Shift | G**

Commit to main

Committed just now: bugfix: threshold_otsu

Github – pull requests

- Contribute to open-source projects

BiAPoL / example_image_analysis_script

forked from haesleinhuepf/example_image_analysis_script

main

This branch is 1 commit ahead of haesleinhuepf:main.

Open a pull request to contribute your changes upstream.

1 minute ago 7

29 minutes ago

1 minute ago

1 hour ago

1 hour ago

1 hour ago

1 hour ago

Open pull request

blobs.tif	initial file upload	1 hour ago
my_library.py	bugfix: threshold_otsu	1 minute ago
my_script.py	initial file upload	1 hour ago
requirements.txt	initial file upload	1 hour ago

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: haesleinhuepf/example_image... base: main

head repository: BiAPoL/example_image_analys... compare: main

✓ Able to merge. These branches can be automatically merged.

bugfix: threshold_otsu

Write Preview

Dear Robert,

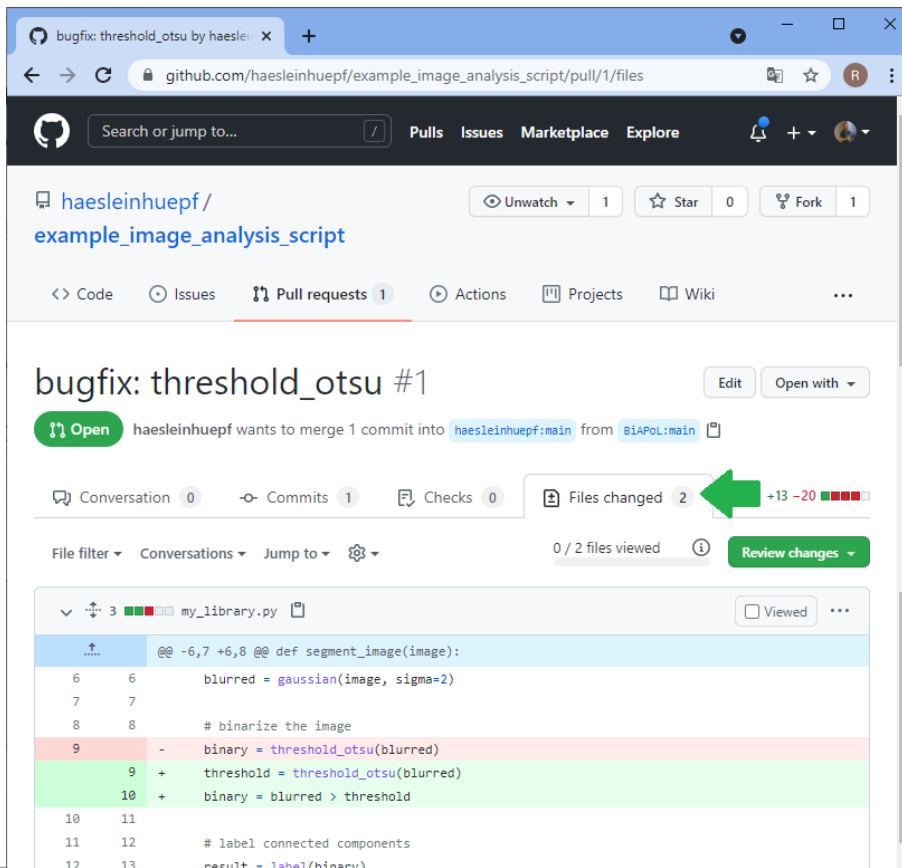
here comes a bug fix for your image segmentation function. threshold_otsu delivers a number (the threshold), not a binary image. For thresholding the image, an additional step is necessary.

Best,
Robert

Create pull request

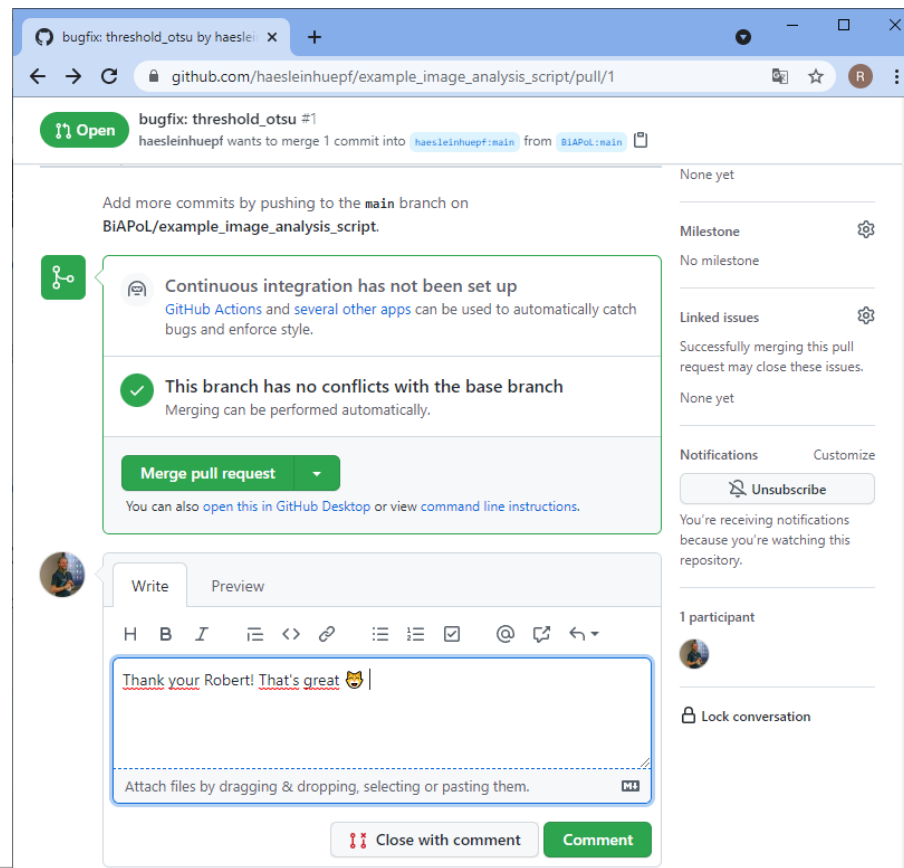
Github – pull requests

- Reviewer perspective



The screenshot shows the GitHub pull request interface for a repository named 'haesleinhuepf/example_image_analysis_script'. The pull request is titled 'bugfix: threshold_otsu #1' and is being reviewed by 'haesleinhuepf'. The interface shows the 'Files changed' section with a diff view of the file 'my_library.py'. A green arrow points to the 'Files changed' section, which shows 2 files changed (+13 -20). The diff view shows the following code changes:

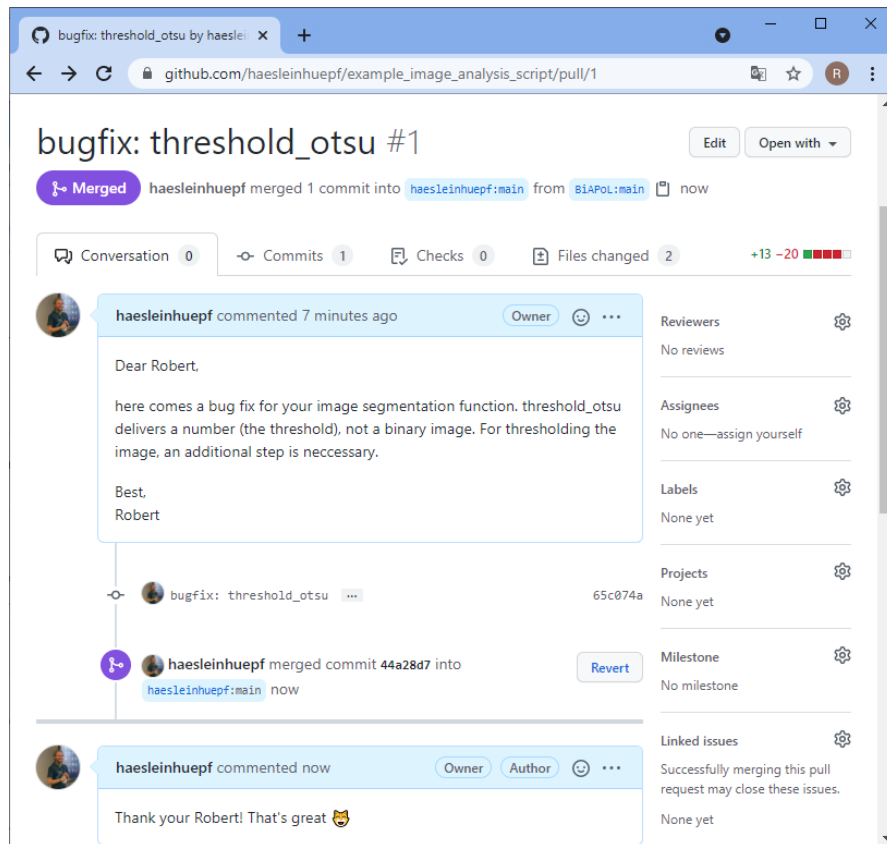
```
@@ -6,7 +6,8 @@ def segment_image(image):
6 6     blurred = gaussian(image, sigma=2)
7 7
8 8     # binarize the image
9 -     binary = threshold_otsu(blurred)
9 +     threshold = threshold_otsu(blurred)
10 +     binary = blurred > threshold
10 11
11 12     # label connected components
12 13     result = label(binary)
```



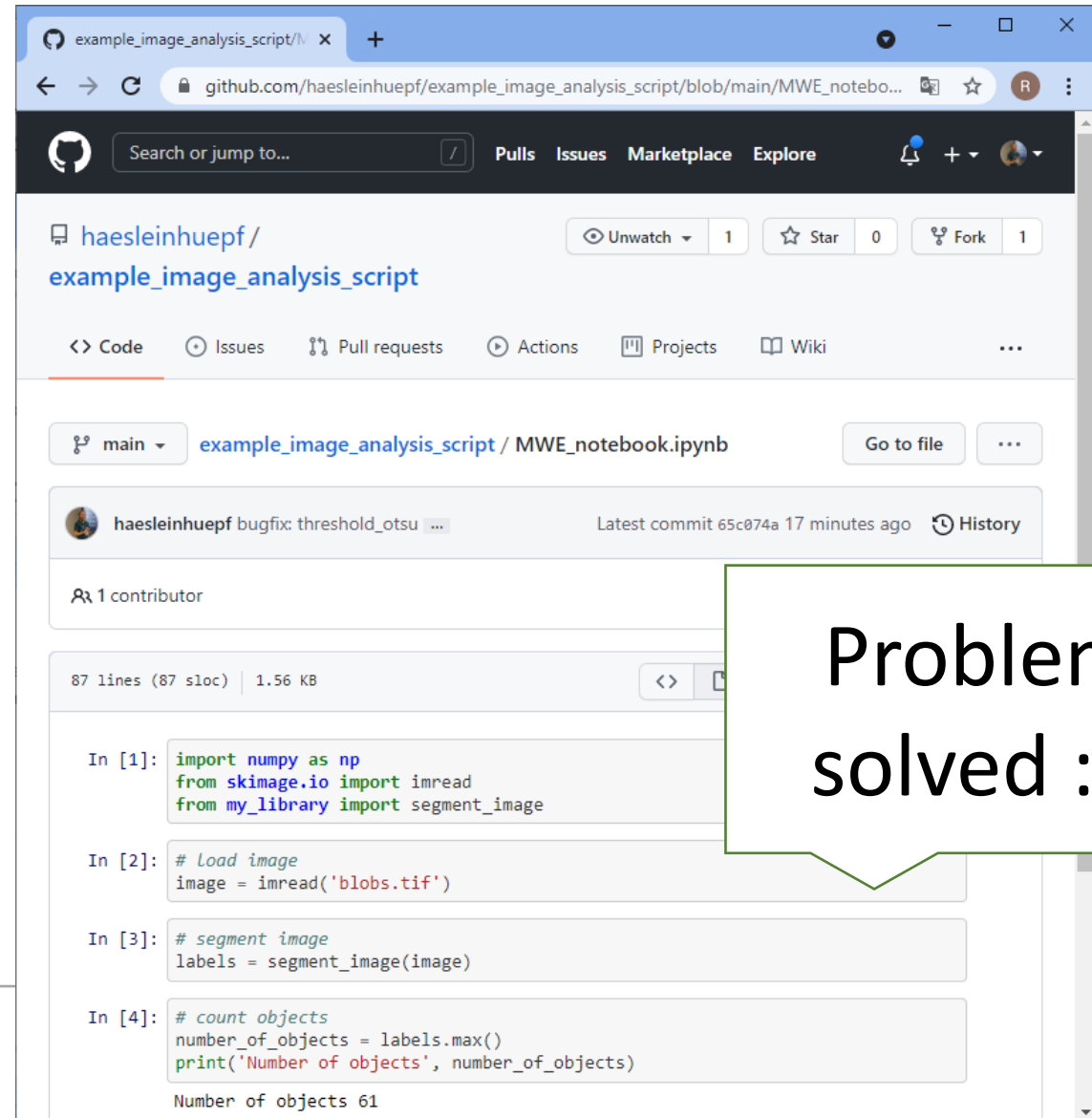
The screenshot shows the GitHub pull request interface for a repository named 'haesleinhuepf/example_image_analysis_script'. The pull request is titled 'bugfix: threshold_otsu #1' and is being reviewed by 'haesleinhuepf'. The interface shows the 'Merge pull request' button and the 'Close with comment' button. The review text is: 'Thank you Robert! That's great 🐱'. The interface also shows the 'Merge pull request' button and the 'Close with comment' button.

Github – pull requests

- Reviewer perspective



The screenshot shows a GitHub pull request titled "bugfix: threshold_otsu #1". The pull request is merged, and the commit message is "haesleinhuepf merged 1 commit into haesleinhuepf:main from BIAPOL:main". The pull request is owned by haesleinhuepf. The conversation shows a comment from haesleinhuepf: "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". The pull request is merged into the main branch.



The screenshot shows a GitHub repository page for "haesleinhuepf/example_image_analysis_script". The file "MWE_notebook.ipynb" is selected. The code in the notebook is as follows:

```
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)

Number of objects 61
```

A callout box with a green border contains the text "Problem solved :-)".

Quiz (recap)

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

`submit`



`push`



`pull`

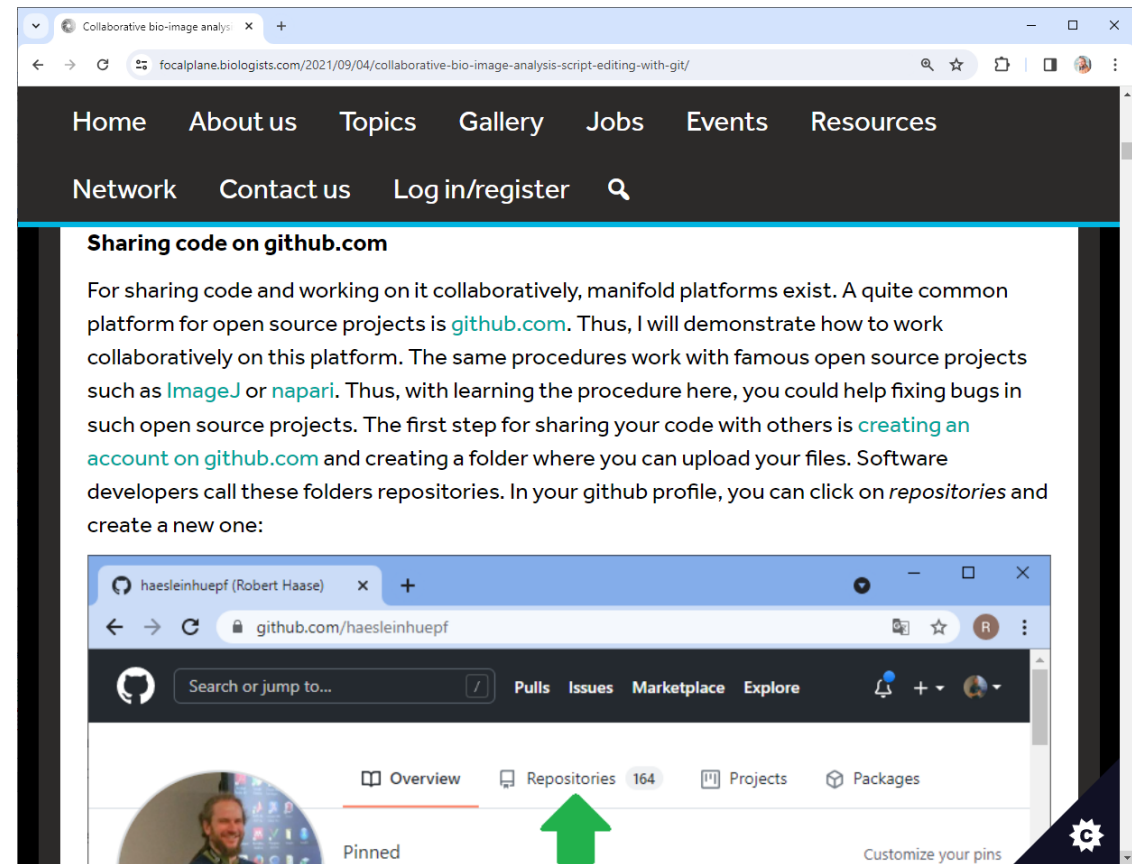


`commit`



Github

- If this was too fast...





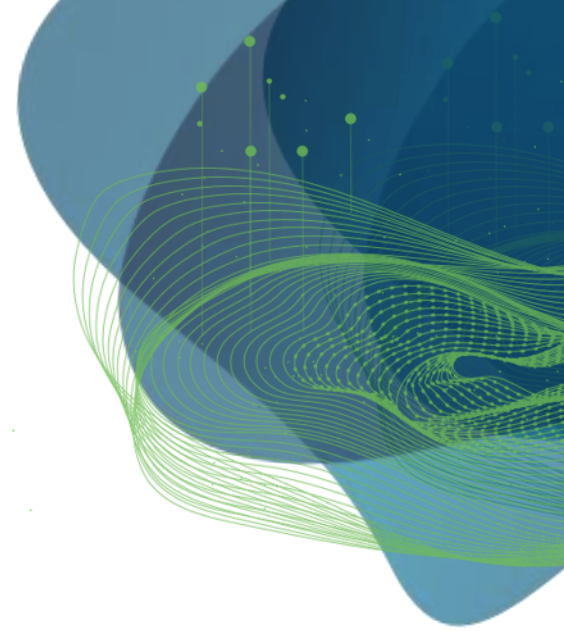
ScaDS.AI

DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Conda environments

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.

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*

The screenshot displays a Windows File Explorer window showing the directory structure of a conda environment. The path is `C:\Users\haase\mambaforge\envs`. The 'envs' folder is selected, and its contents are listed in the right pane:

- apoc-env1
- bio31
- bio39
- bio39_
- bio310
- bio310_
- bio39
- bio310_
- bio392
- bio393
- bio39_
- bio39_
- bla

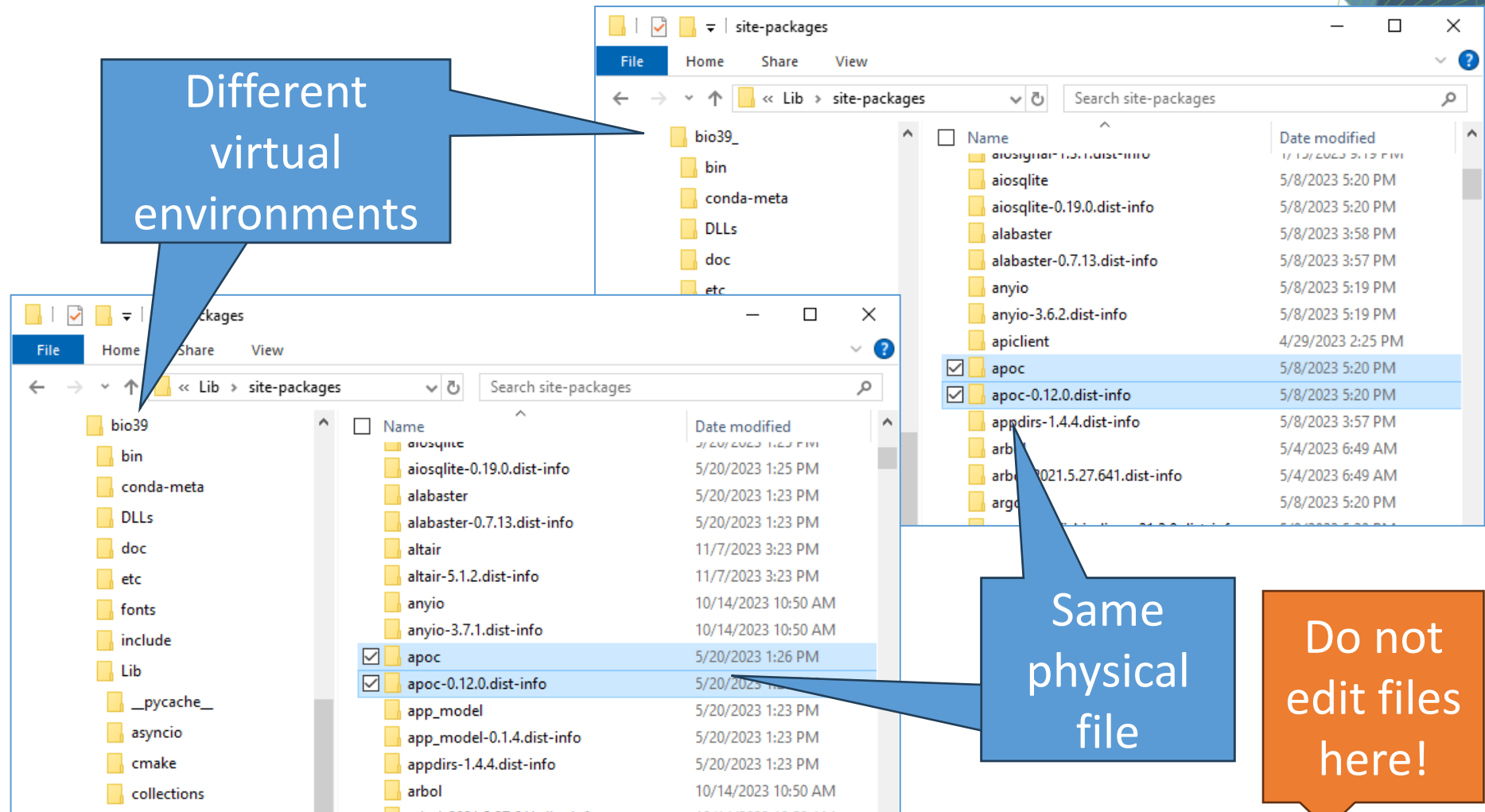
Overlaid on the right is a Command Prompt window showing the command `conda env list` and its output:

```
C:\Users\haase>conda env list
# conda environments:
#
base                                C:\Users\haase\mambaforge
apoc-env1                           C:\Users\haase\mambaforge\envs\apoc-env1
bio31                                C:\Users\haase\mambaforge\envs\bio31
bio310                               C:\Users\haase\mambaforge\envs\bio310
bio310_                             C:\Users\haase\mambaforge\envs\bio310_
bio39                                C:\Users\haase\mambaforge\envs\bio39
bio392                              C:\Users\haase\mambaforge\envs\bio392
bio393                              C:\Users\haase\mambaforge\envs\bio393
bio39_                              C:\Users\haase\mambaforge\envs\bio39_
bio39_                              C:\Users\haase\mambaforge\envs\bio39_
bla                                 C:\Users\haase\mambaforge\envs\bla
```

A blue speech bubble points to the Command Prompt with the text `conda env list`.

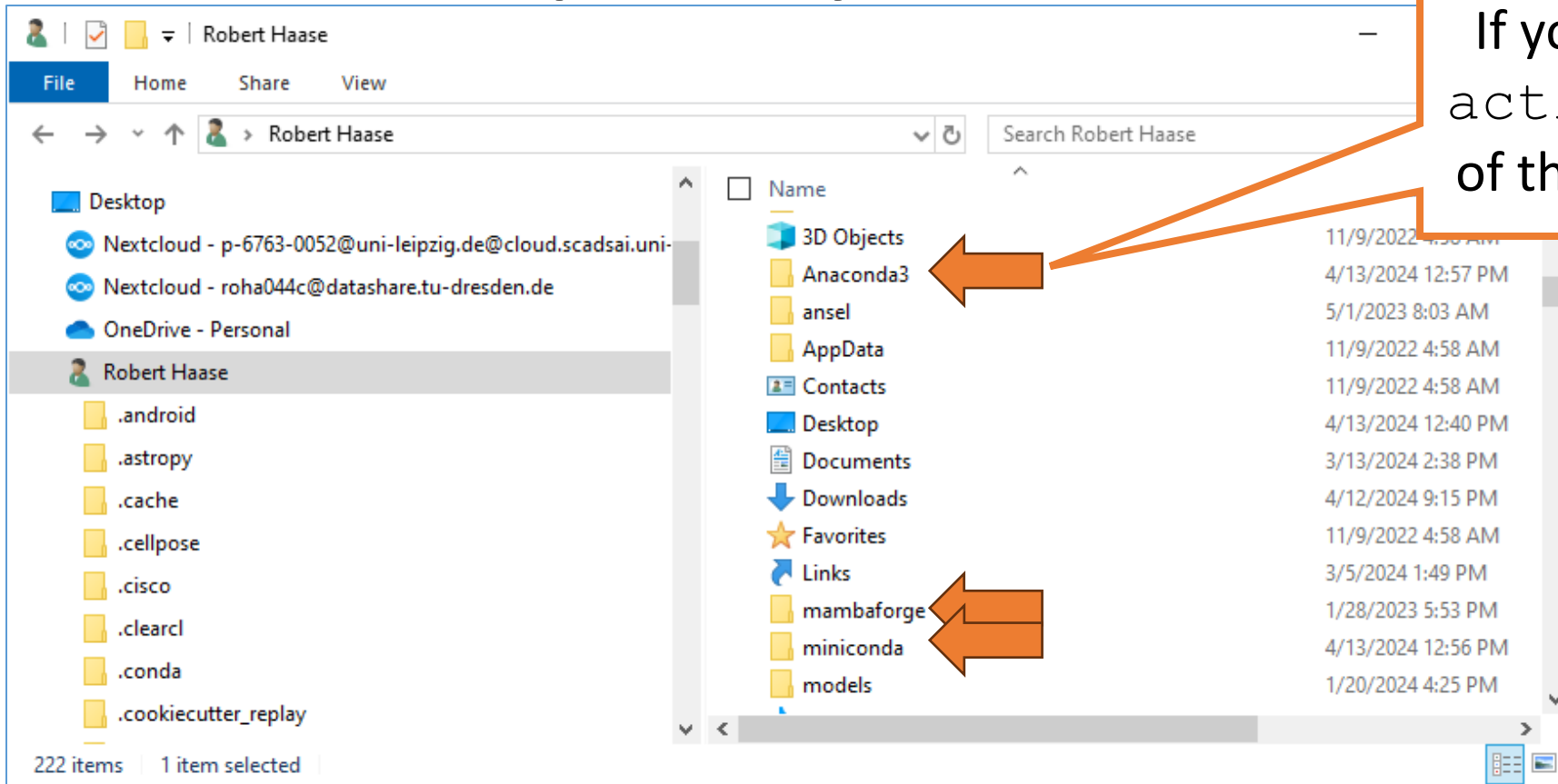
Conda environments

- 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



Conda environments

- Conda environments can be activated from anywhere

```
Command Prompt
C:\Users\haase>conda activate C:\Users\haase\mambaforge\envs\bio31
(bio31) C:\Users\haase>
```

```
Command Prompt - conda deactivate - conda activate bio31
C:\Users\haase>conda activate bio31
(bio31) C:\Users\haase>
```

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.

The image consists of three overlapping screenshots from a Windows environment:

- Left Screenshot:** Shows the MAMBA logo and the text "mamba (0.25.0) supported by @QuantStack". It also includes GitHub and Twitter links.
- Middle Screenshot:** A terminal window showing the output of a conda/mamba command. It lists various packages being installed, such as toolz, tornado, traitlets, typing_extensions, tzdata, ucrt, unicodedata2, vc, vs2015_runtime, wcwidth, wheel, widgetsnbextension, xorg-libxau, xorg-libxdmcp, xz, yaml, zeromq, zipp, zlib-ng, and zstd. It also shows a confirmation prompt "Confirm changes: [Y/n]" and the start of a transaction.
- Right Screenshot:** A JupyterLab interface showing a code cell with the command `import stackview`. Below the code, a red error message is displayed: `ModuleNotFoundError: No module named 'stackview'`. The error message also includes a traceback: `k (most recent call last) Cell In [1], line 1 ----> 1 import stackview`.

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

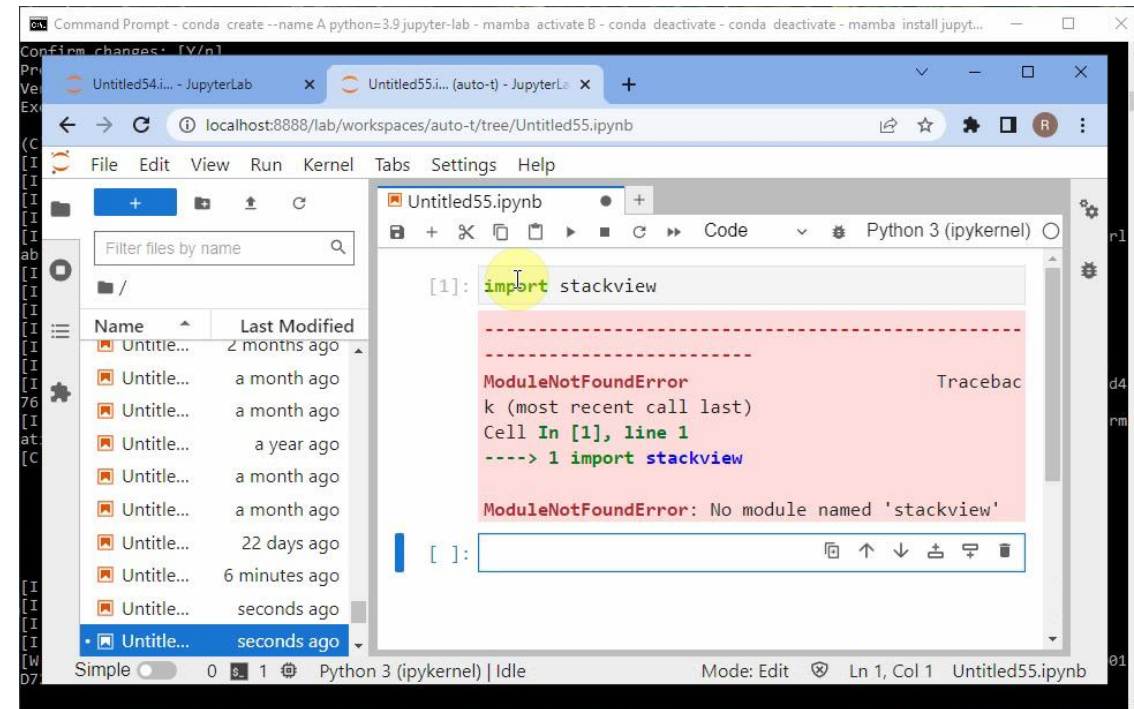
Installed packages:

Jupyter lab

my_env

Installed packages:

stackview



Broken [conda] environments

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

Huge effort.

Little effort.

- 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**

```
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
```

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

`conda env create -f environment.yml`

- **The pip way**

```
requirements.txt - Notepad
File Edit Format View Help
numpy>=1.21
pandas~=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
- backports=1.0=py_2
```

```
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
```

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



Dependency
conflict

`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

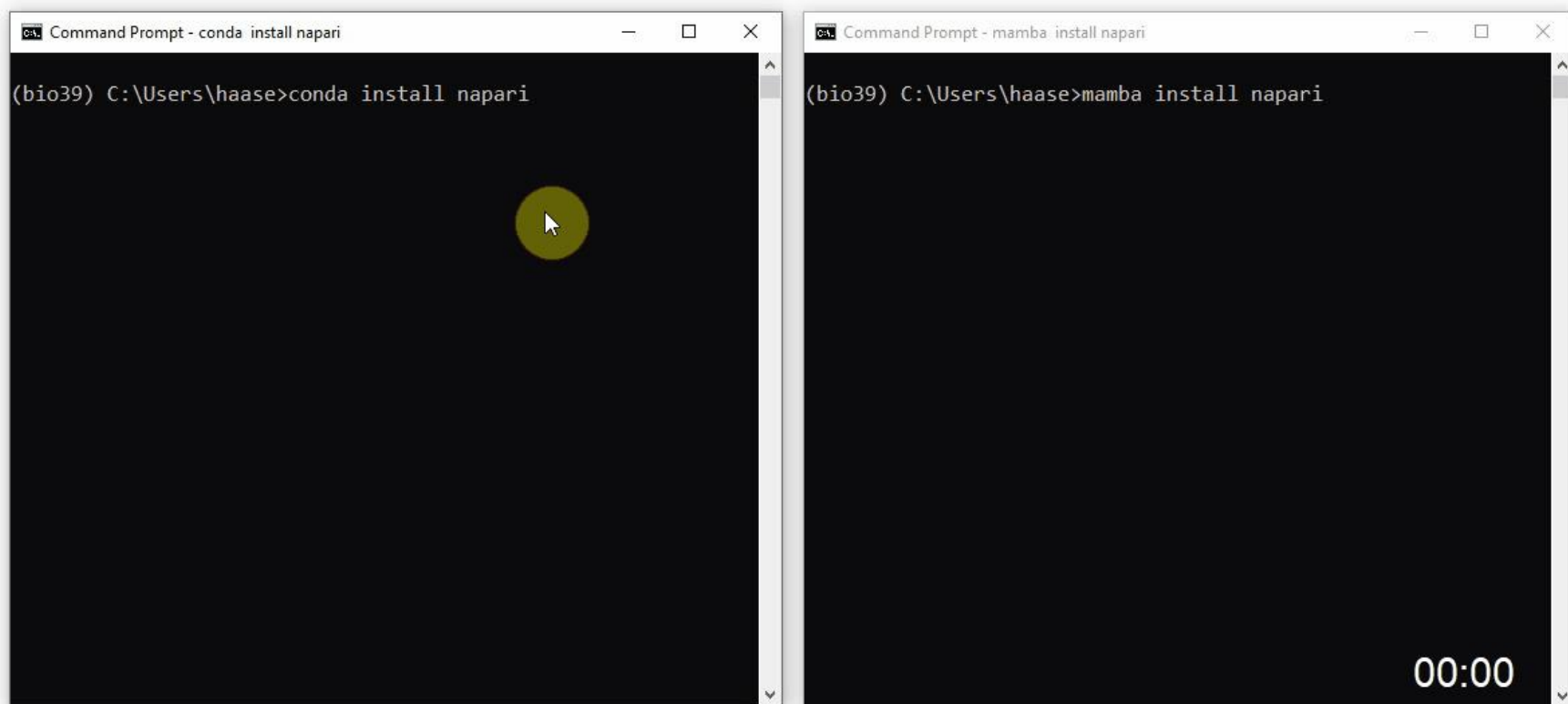
Because git is not a
python package

```
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<=1.22.0

```
pip install package_b
```

Depends on:
numpy>=1.22.0



Dependency
conflict

fails



works



works but...



Software quality indicators

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

Release of Bio-Formats 6.5.0 - A... x napari/napari: napari: a fast, inter... x

github.com/napari/napari

Search or jump to...

Public

Code Issues 818 Pull requests 61 Discussions Actions Projects 1

main Go to file Add file Code About

mstabrin Allow pandas.Series ... 16 hours ago 2,330

napari: a fast, interactive, multi-dimensional image viewer for python

visualization python numpy napari

Readme BSD-3-Clause license Cite this repository 1.4k stars 45 watching 290 forks

Contributors 119

- **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

Go to file Add file Code About

No description, website, or topics provided.

b8949b0 on 19 Jan 2020 2 commits

3 years ago

3 years ago

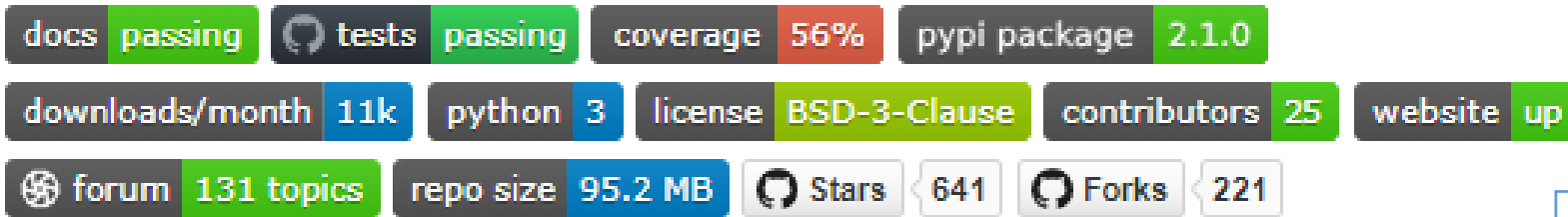
1 star

2 watching

0 forks

Software quality indicators

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

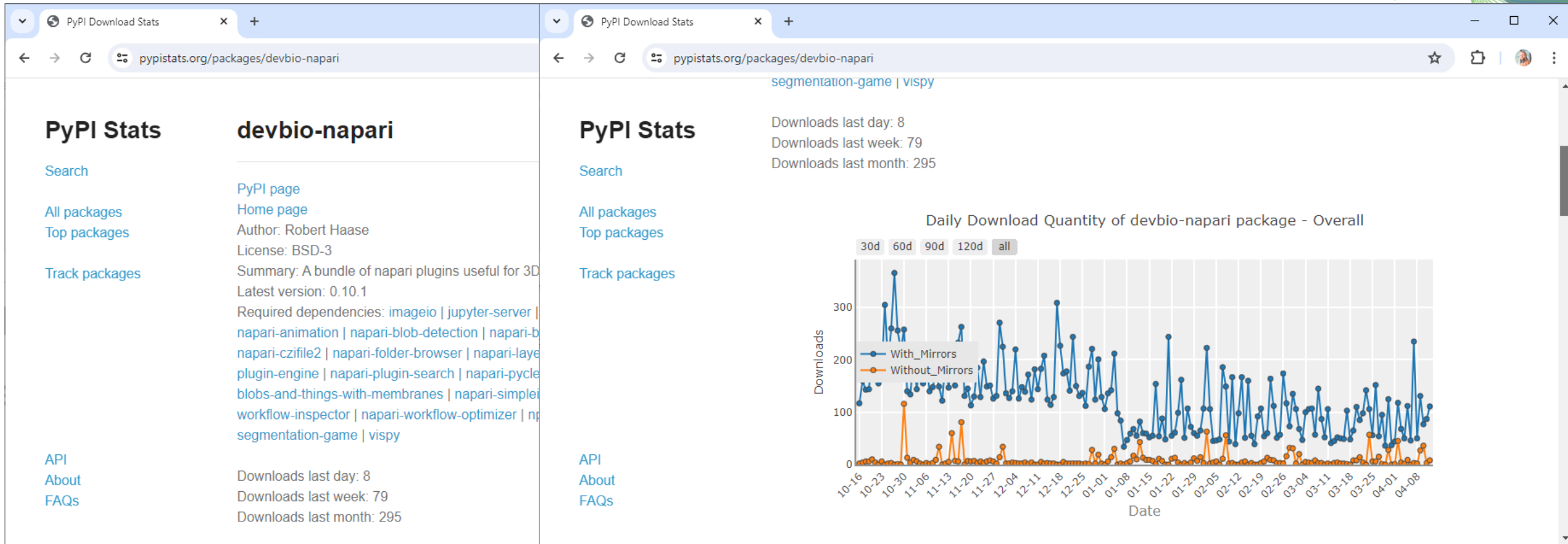


Note, github badges cannot be *deserved*.
Developers put them there



Software quality indicators

- Download statistics: pypi



Software quality indicators

- Download statistics: conda

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

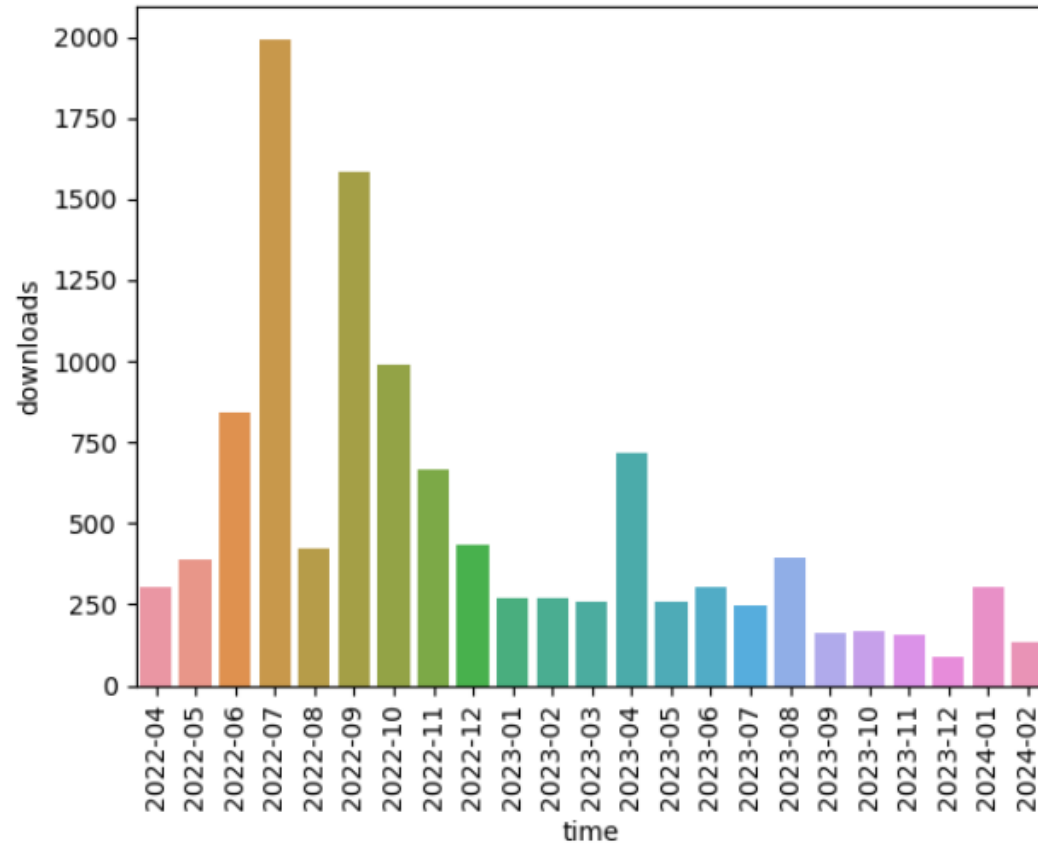
```
C:\Users\haase\mambaforge\envs\bio39\lib\site-packages\condastats\cli.py:153:  
FutureWarning: The default of observed=False is deprecated and will be changed  
to True in a future version of pandas. Pass observed=False to retain current b  
ehavior or observed=True to adopt the future default and 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
```

```
[3]:
```

	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

MouseLand/cellpose: a generalist algorithm for cellular segmentation

README BSD-3-Clause license

CITATION

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.

Cellpose 2.0: how to train your own model

Article | [Open access](#) | Published: 07 November 2022

Marius Pachitariu & Carsen Stringer

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

55k Accesses | 120 Citations | 116 Altmetric | [Metrics](#)



ScaDS.AI

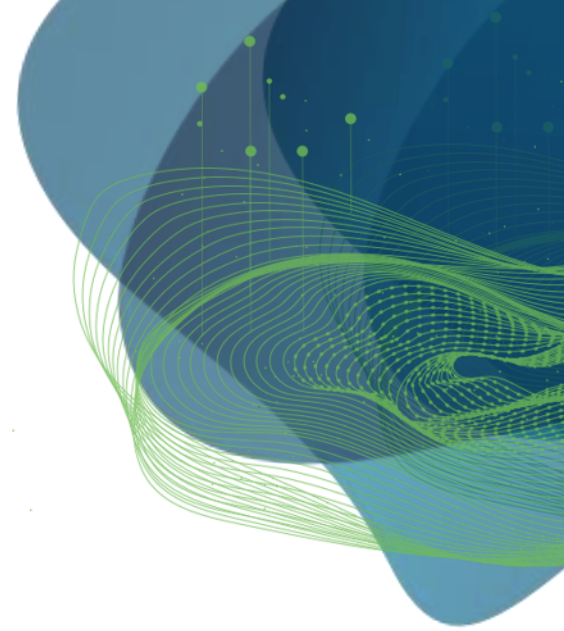
DRESDEN LEIPZIG

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



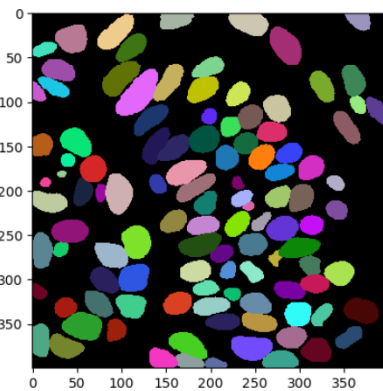
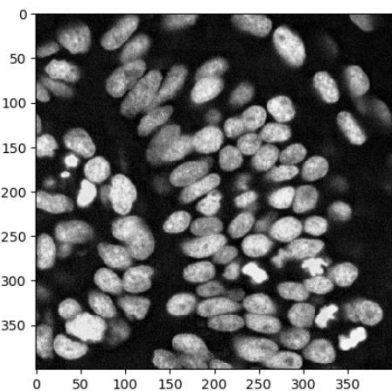
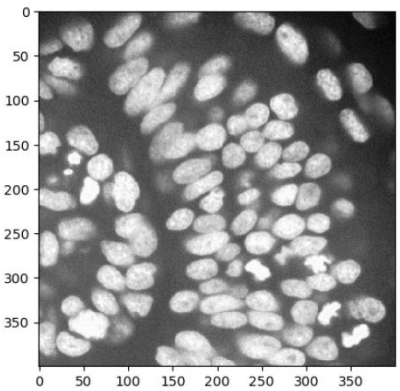
Bundesministerium
für Bildung
und Forschung



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**



	area	mean_intensity	major_axis_length	minor_axis_length
0	594.0	40572.809764	28.611591	26.537947
1	645.0	43764.872868	33.511511	24.566916
2	1105.0	51970.561991	45.232031	31.456308
3	718.0	47015.487465	31.023274	29.5
4	791.0	49132.515803	36.382253	27.7
...
105	238.0	30477.126050	20.252197	15.2
106	615.0	32886.154472	41.030760	19.8
107	110.0	33042.445455	14.366347	9.5
108	222.0	43304.180180	25.370549	11.6
109	167.0	43378.808383		

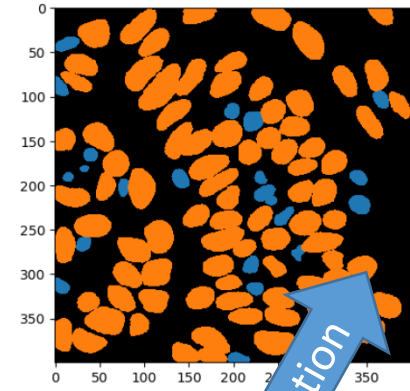
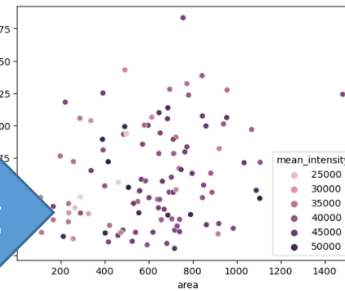


Image filtering

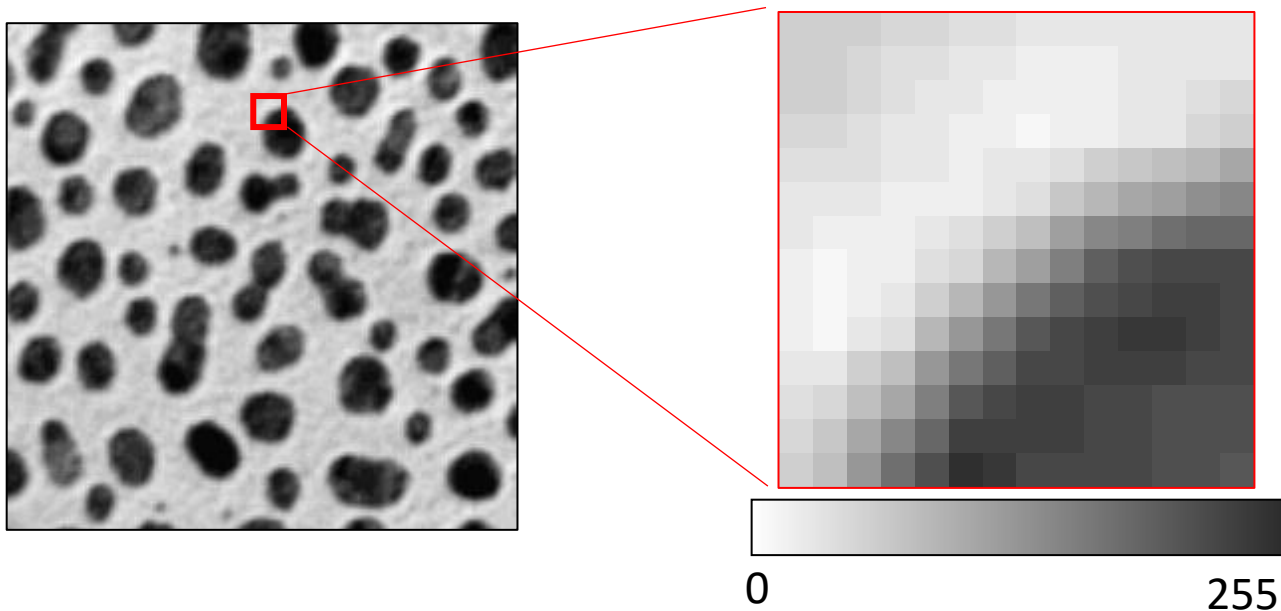
Image segmentation

Feature extraction

Object classification

Images and pixels

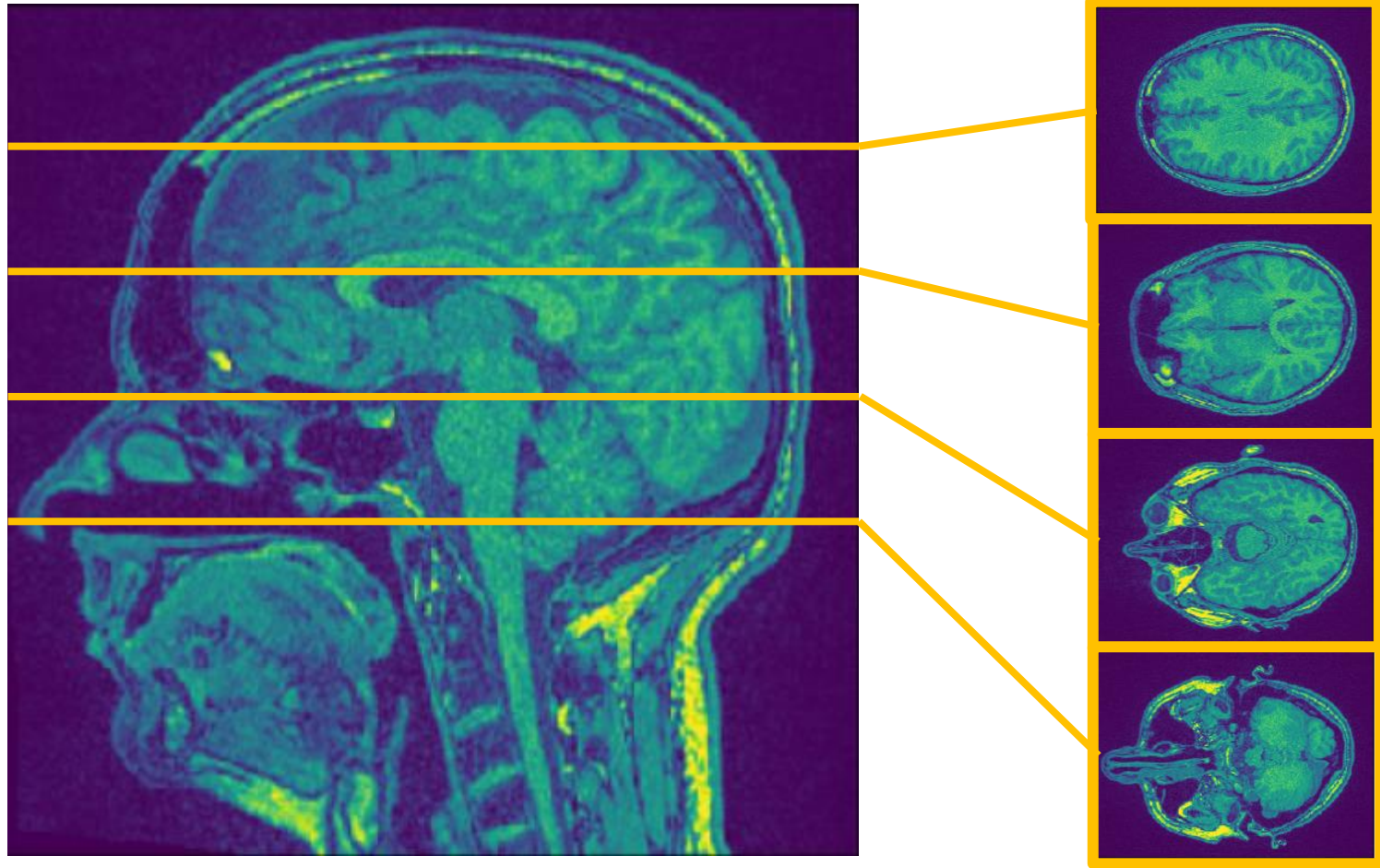
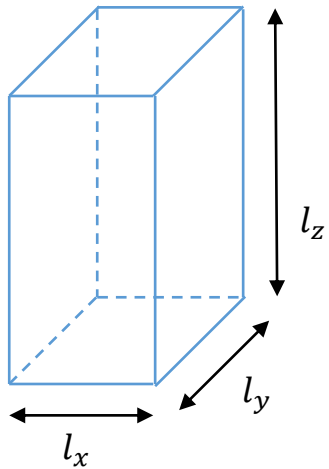
- An image is just a matrix of numbers: pixels: “picture element”
- The edges between pixels are an artefact of the imaging / digitization. They are not real!



48	48	48	40	40	32	32	24	24	24	24	24	24	24
48	48	40	32	32	24	24	16	16	16	24	24	24	24
48	48	40	32	24	24	16	16	16	16	24	24	32	40
40	40	32	24	24	16	16	8	16	16	24	24	40	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	184	176	176	168

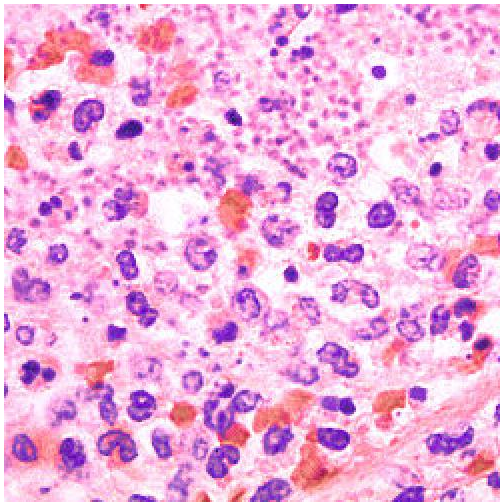
Image stacks and voxels

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



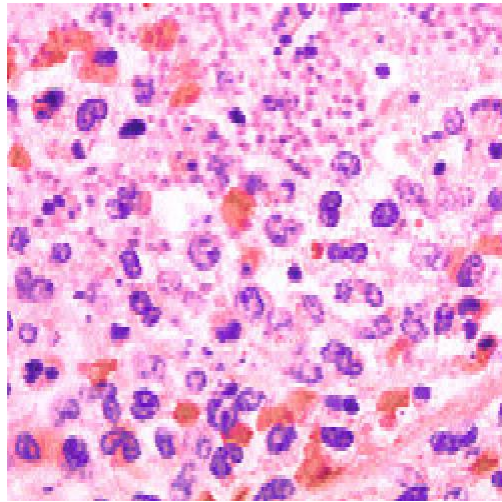
Anisotropy

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



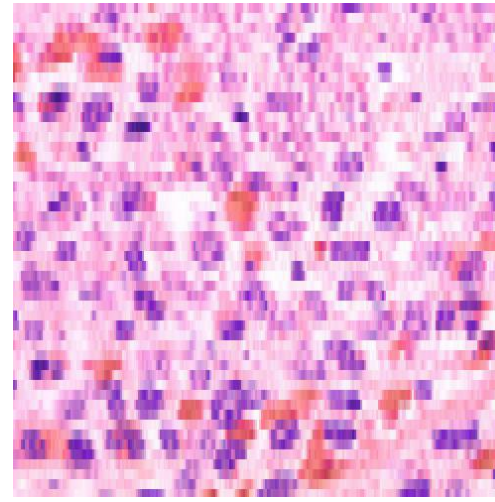
1:1

250 x 250 px



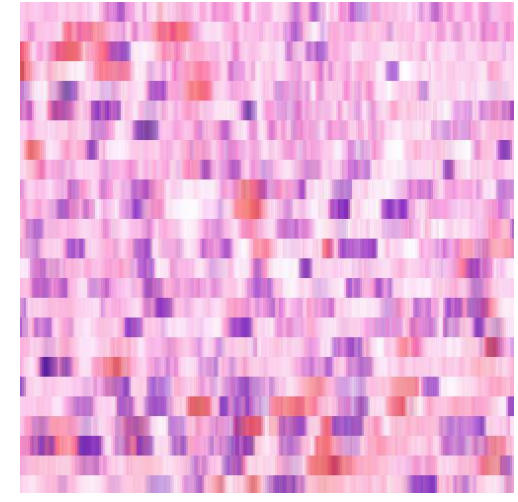
1:2

250 x 125 px



1:5

250 x 50 px

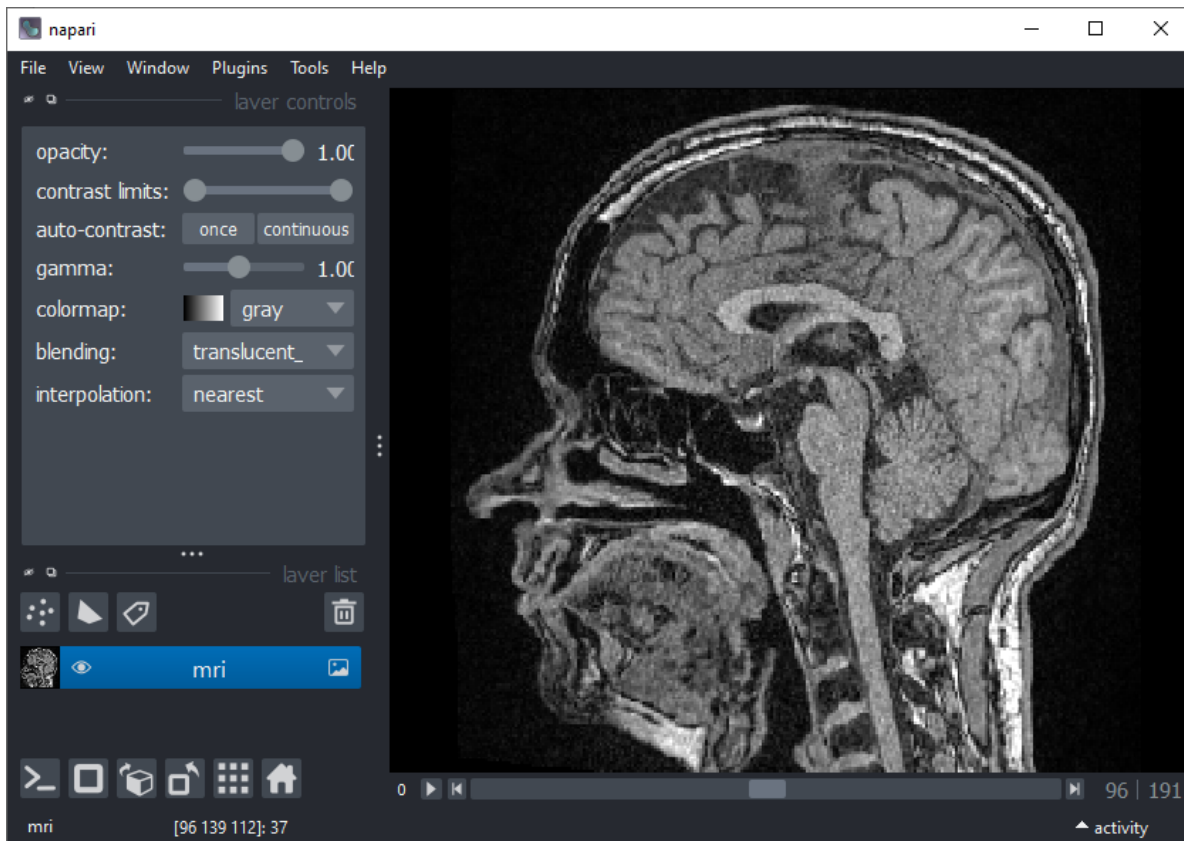


1:10

250 x 25 px

Image stacks and voxels

Interactive tools available



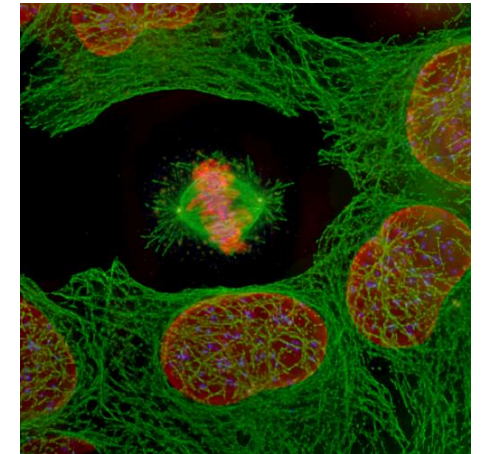
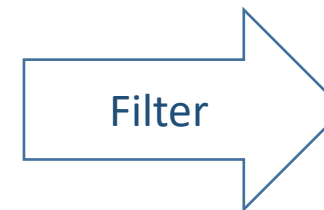
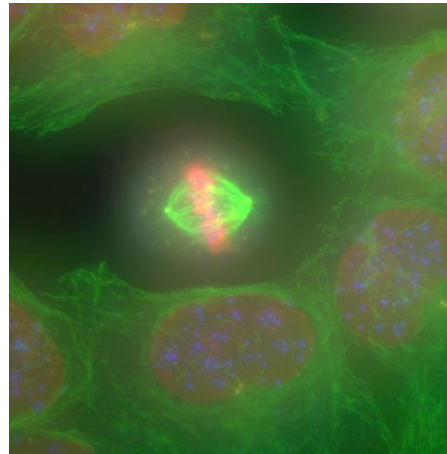
```
import stackview  
  
stackview.slice(mri_image)
```



Slice

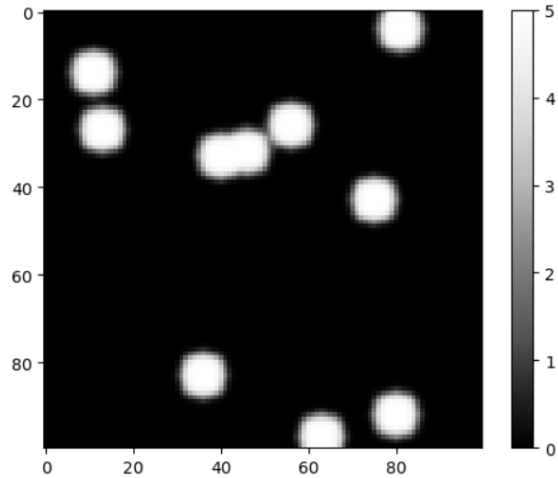
Filters

- 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



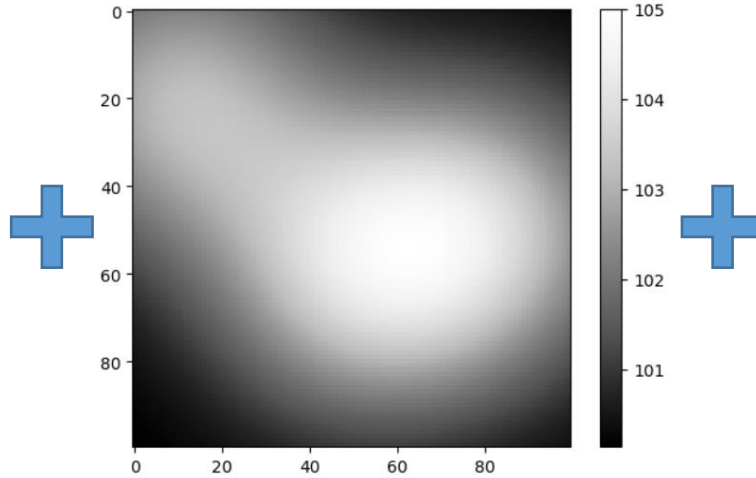
Effects harming image quality

“nuclei”



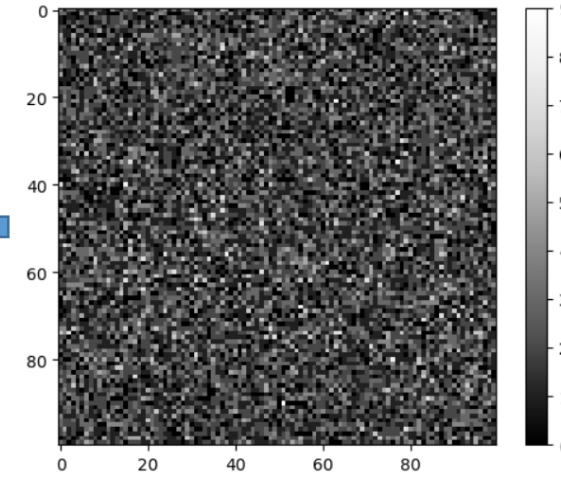
- Aberrations, defocus
- Motion blur

“background”



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

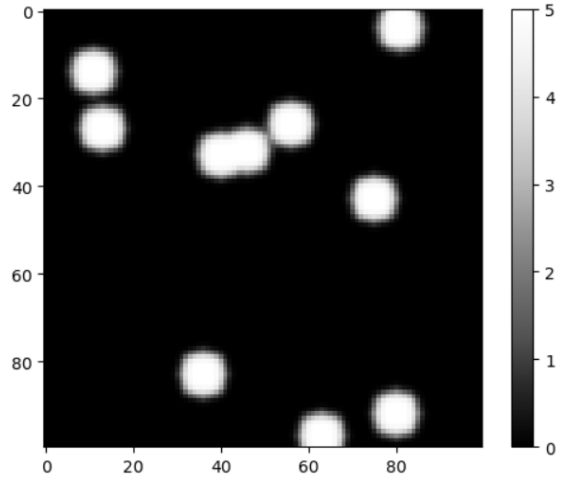
“noise”



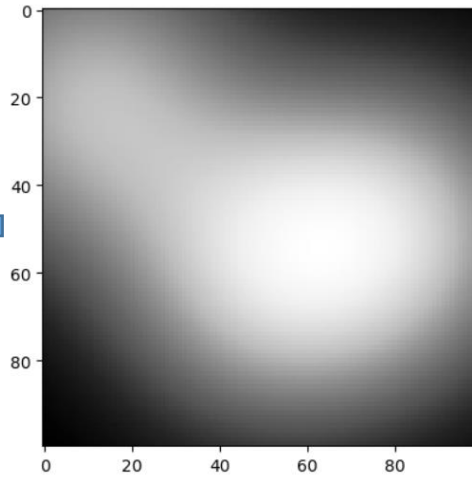
- Shot noise (arriving photons)
- Dark noise (electrons made from photons)
- Read-out-noise (electronics)

Effects harming image quality

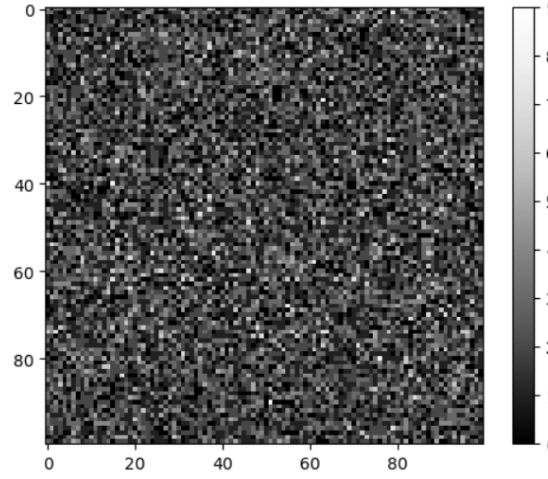
“nuclei”



“background”



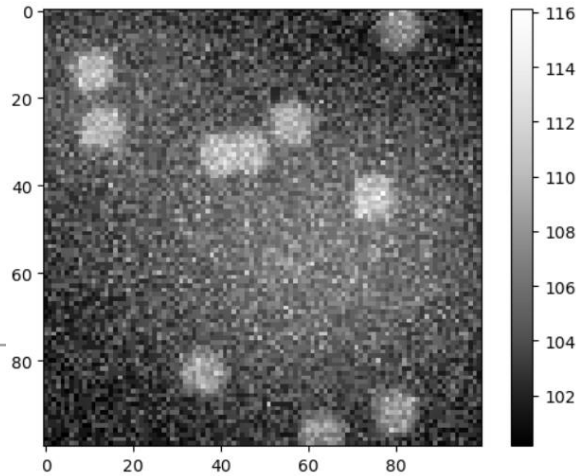
“noise”



+

+

=



Segmentation

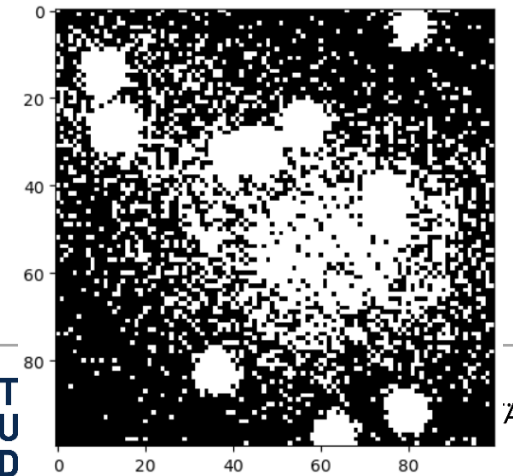
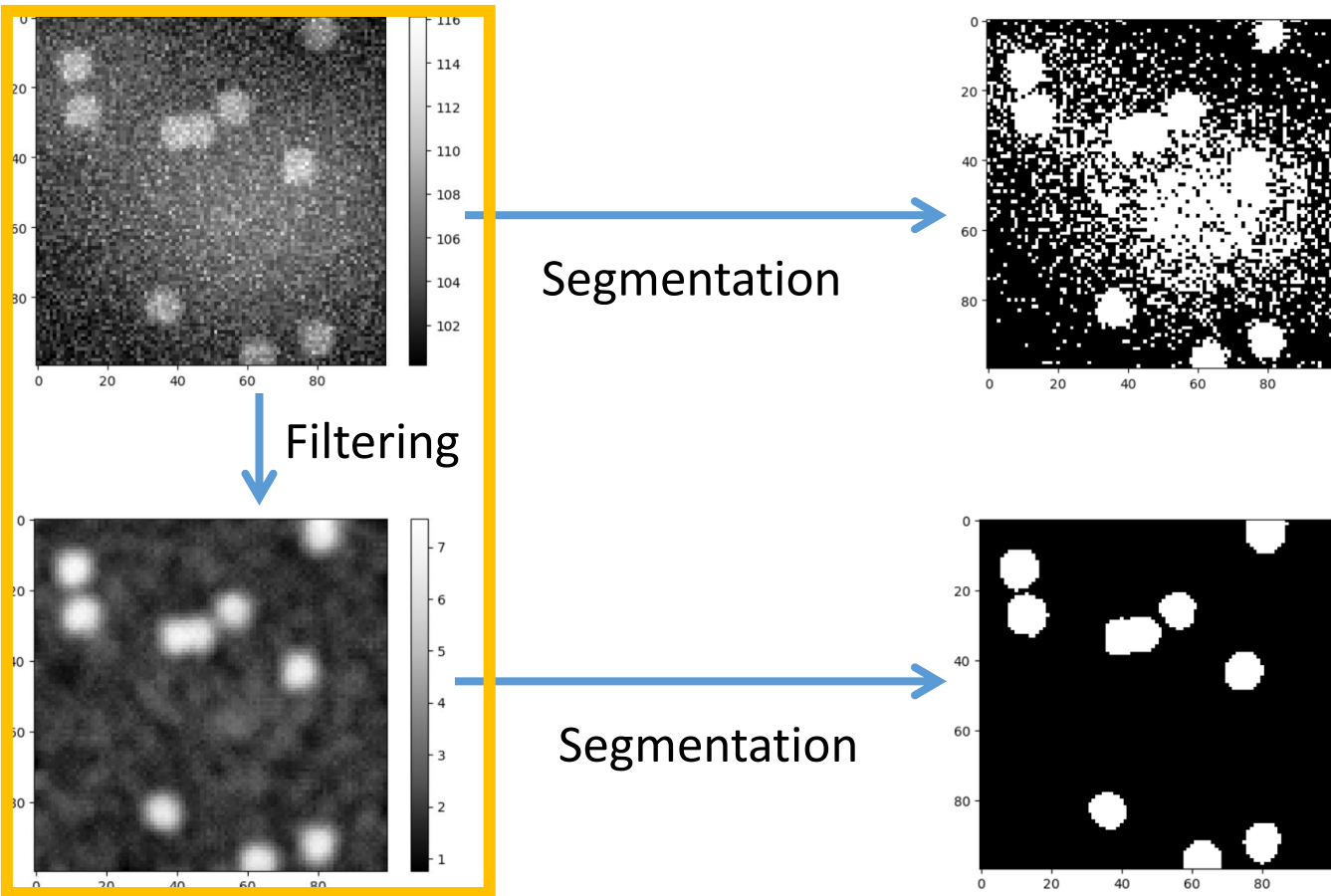


Image filtering

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



Oh no! I see thousands of tiny white objects!

Ok, it's just 9 objects.



Image filtering

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

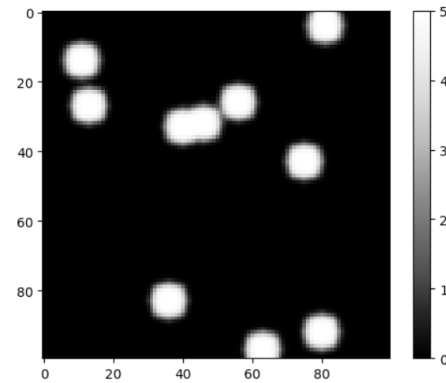
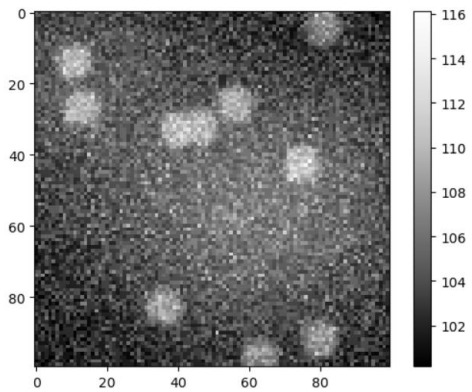
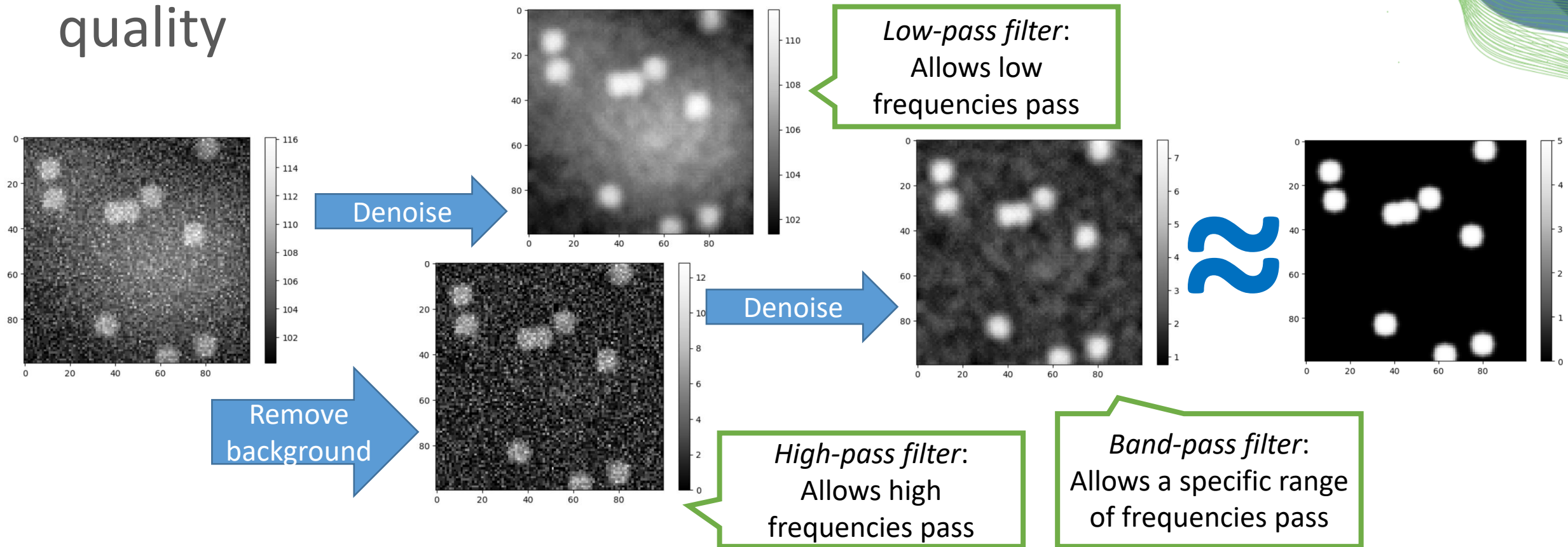


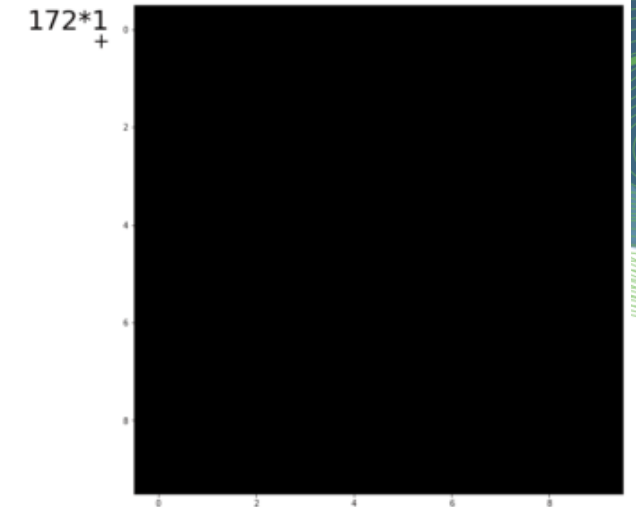
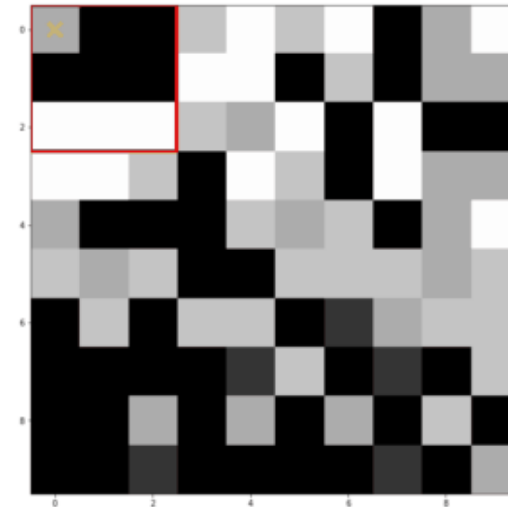
Image filtering

- 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

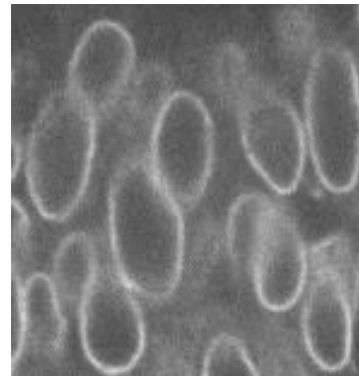
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Linear filters

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

- Examples

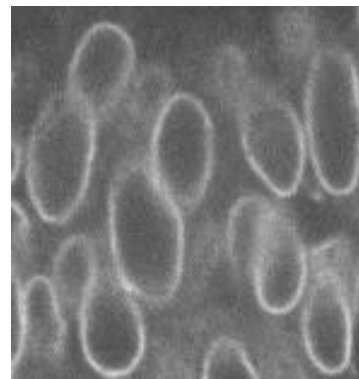
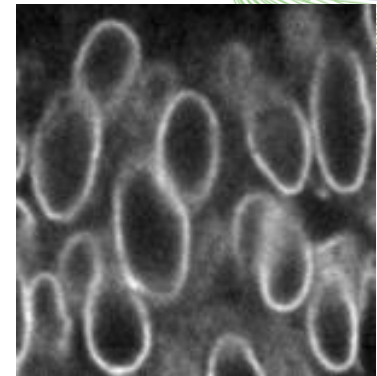
- Mean
- Gaussian blur
- Sobel-operator
- Laplace-filter



*

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

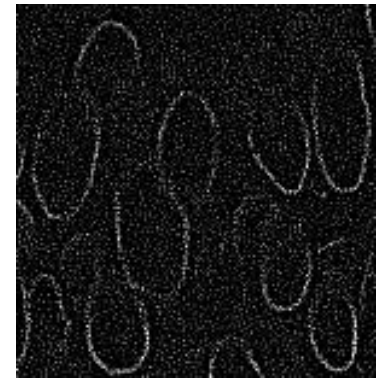
=



*

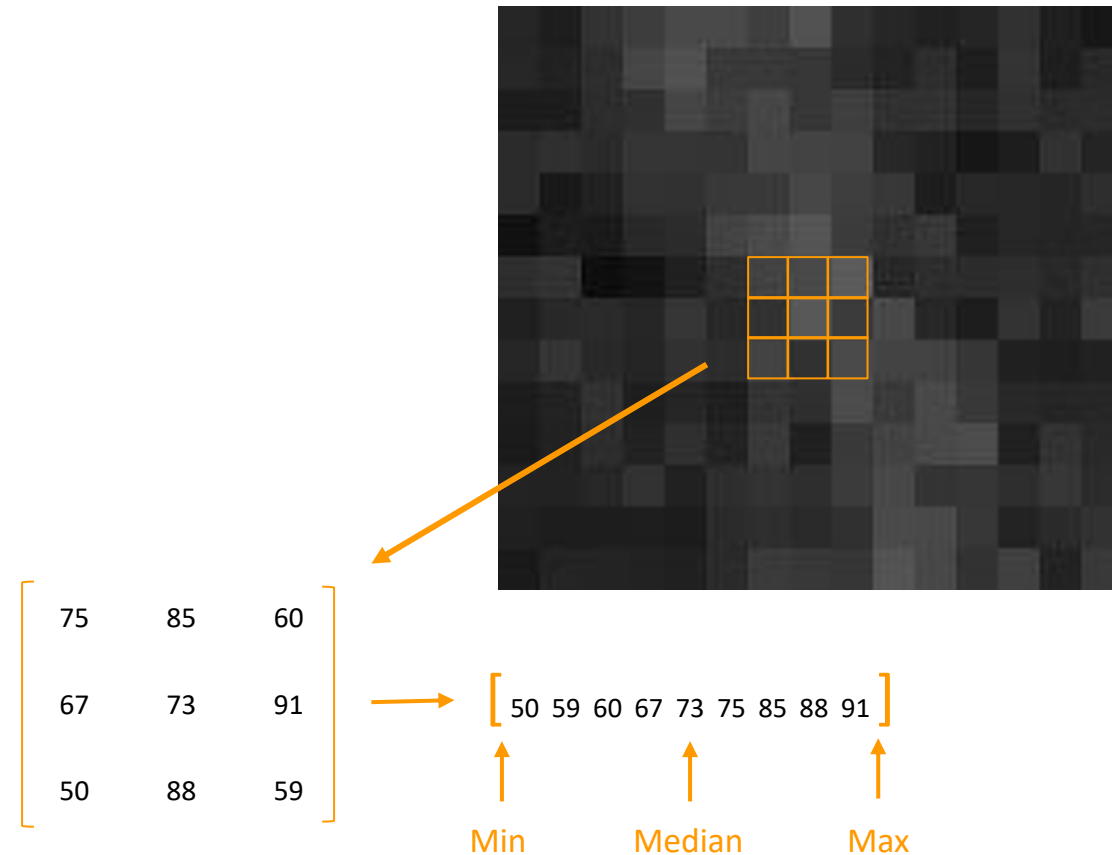
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

=



Non-linear Filters

- Non-linear filters also replace pixel value inside as 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)

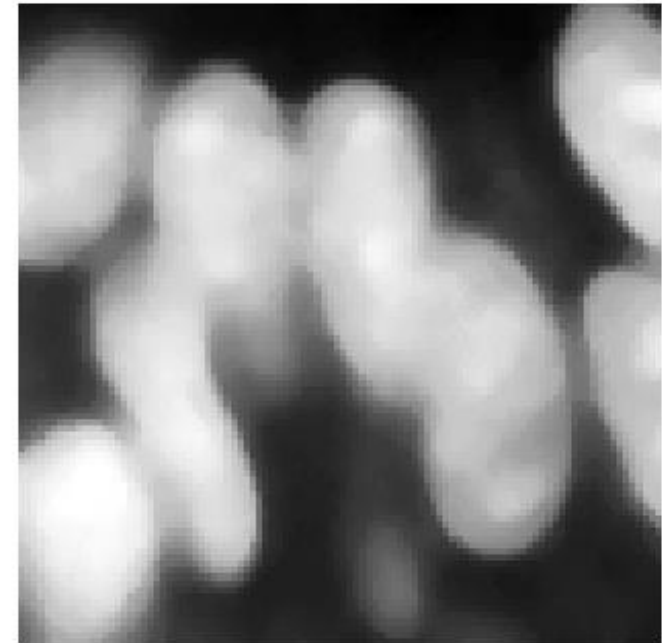
Original



Gaussian

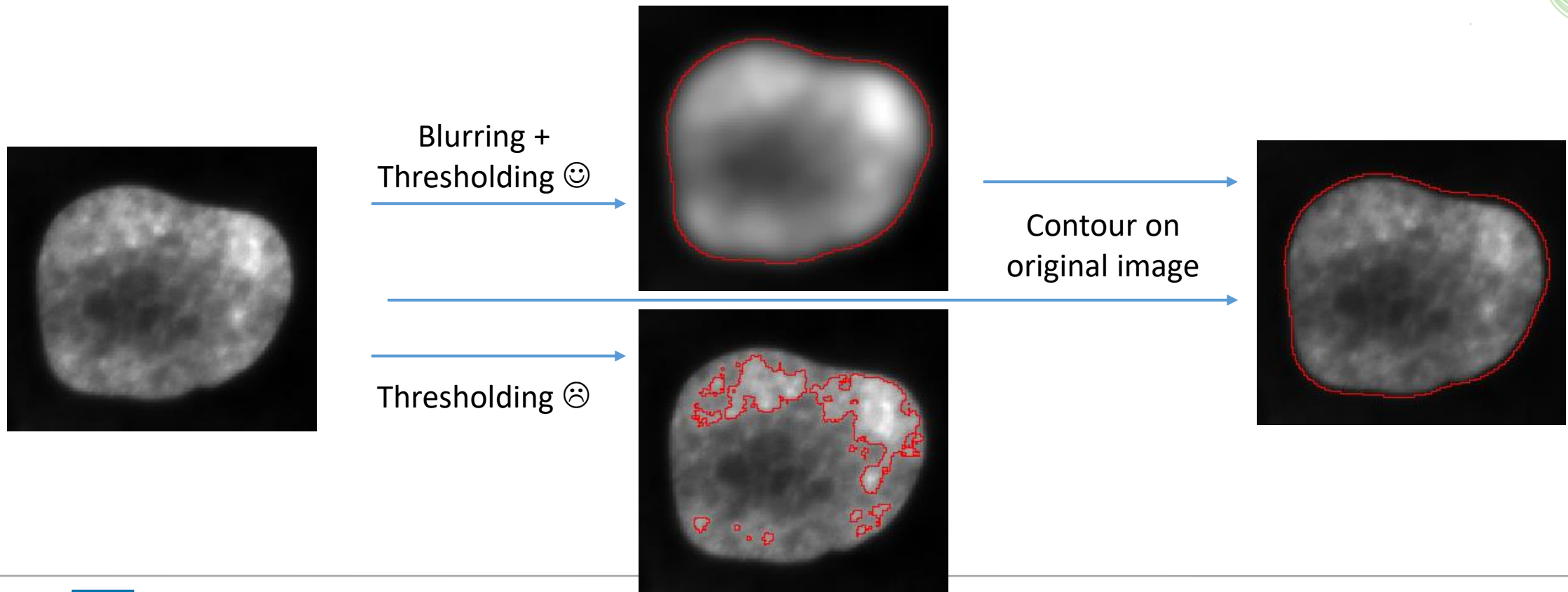


Median



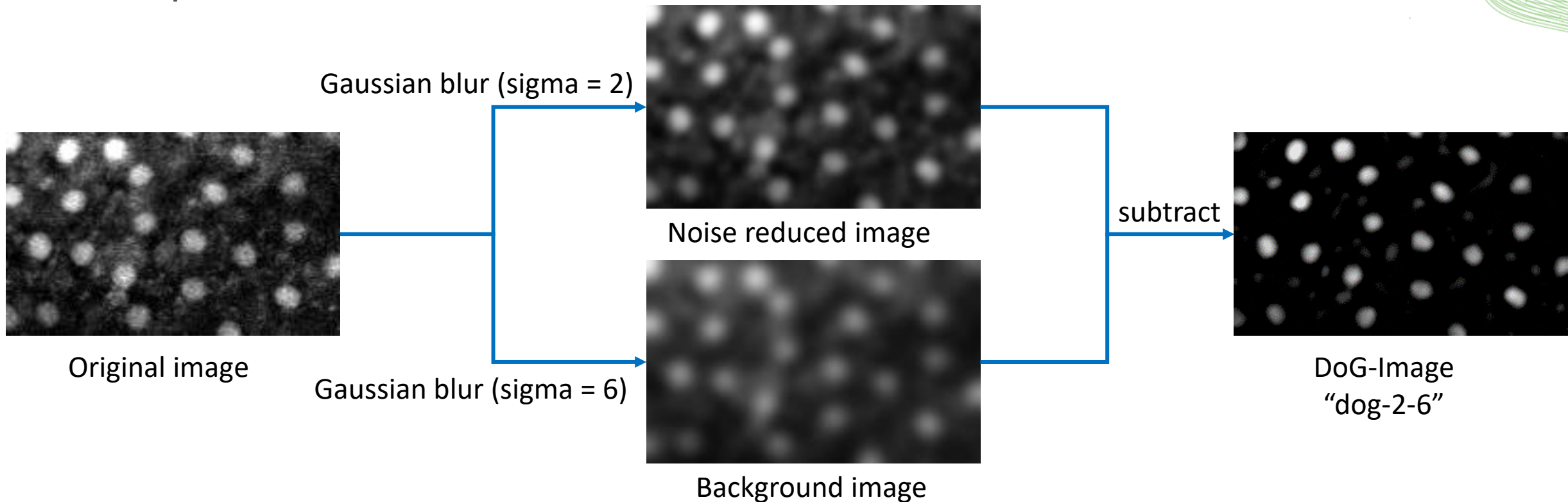
Filtering for improving thresholding results

- 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!



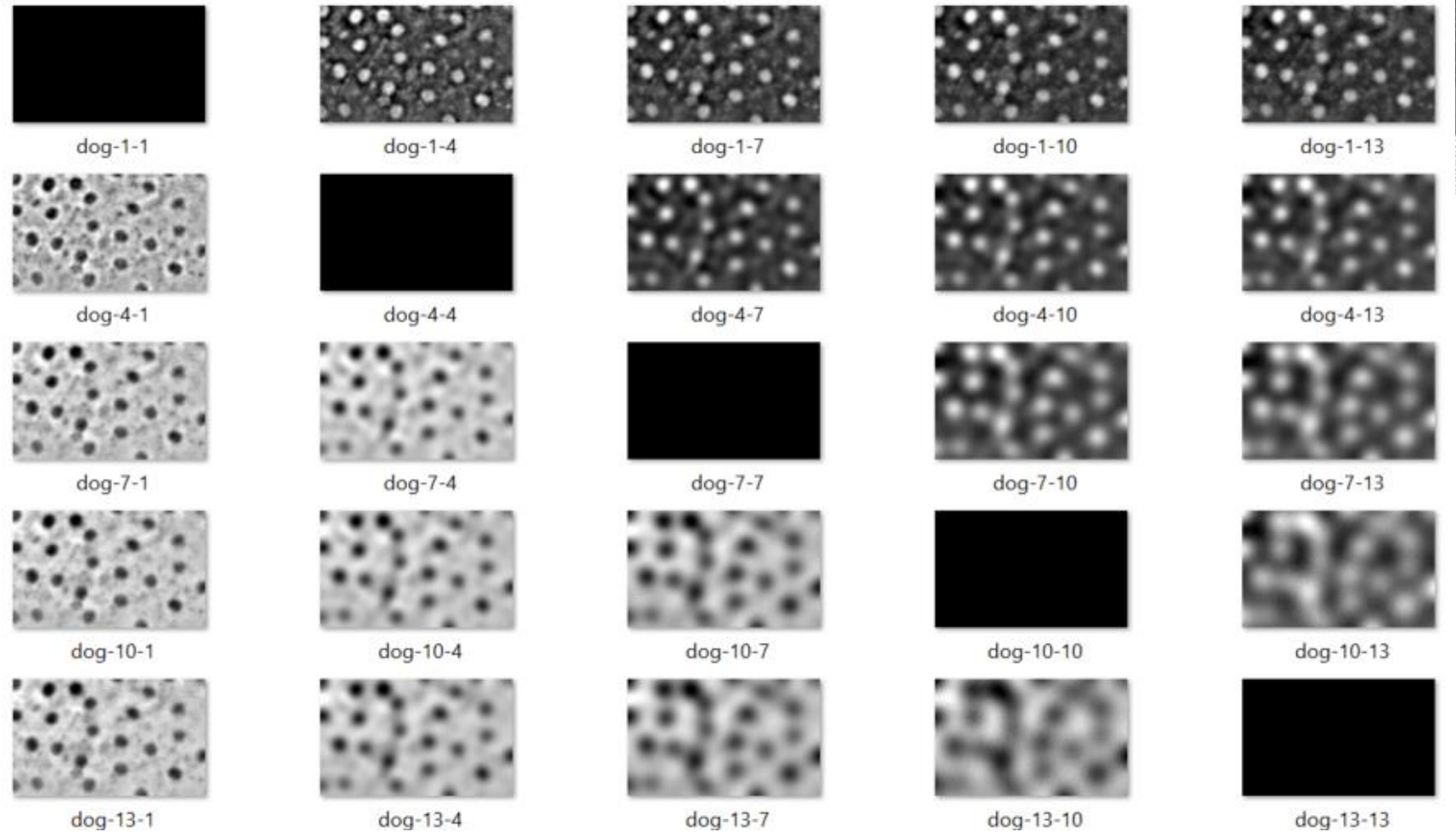
Difference-of-Gaussian (DoG)

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



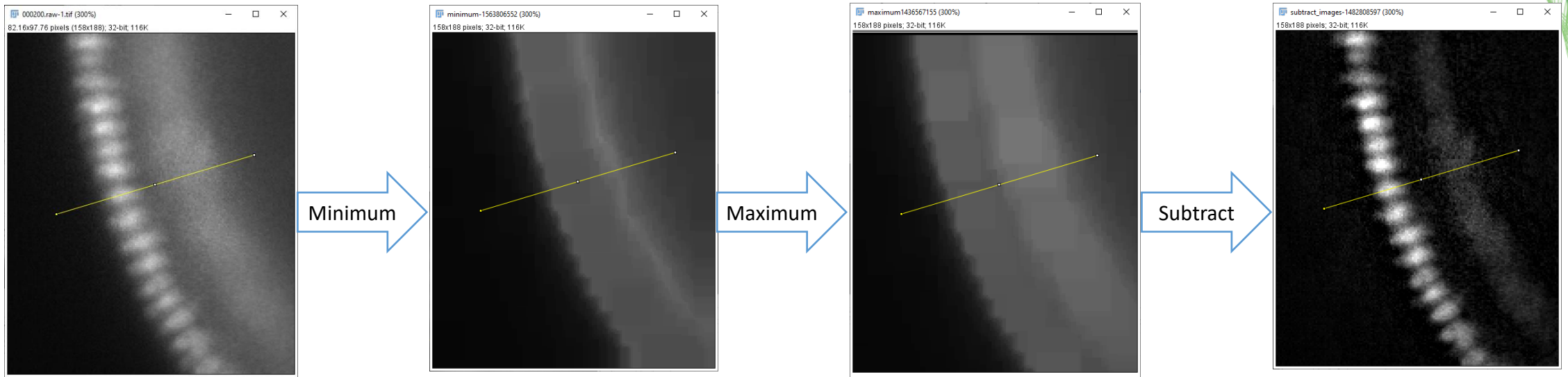
Difference-of-Gaussian (DoG)

- Example DoG images



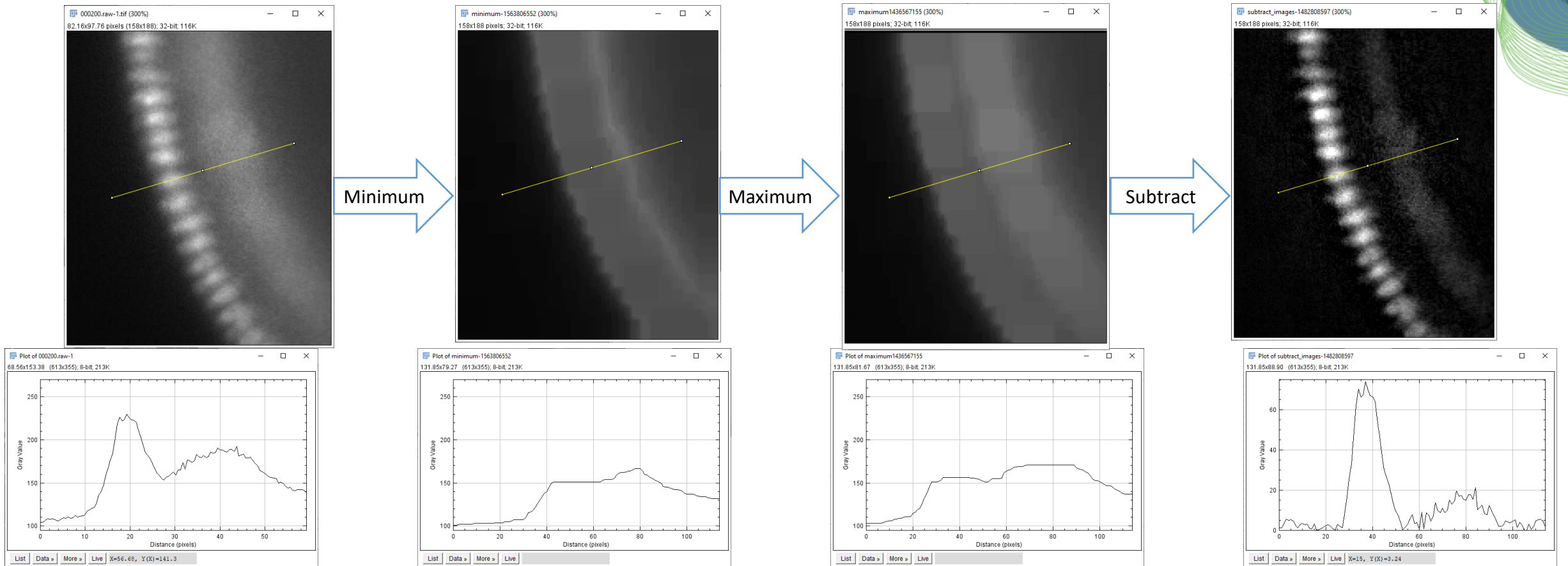
Top-hat filter

- Background subtraction



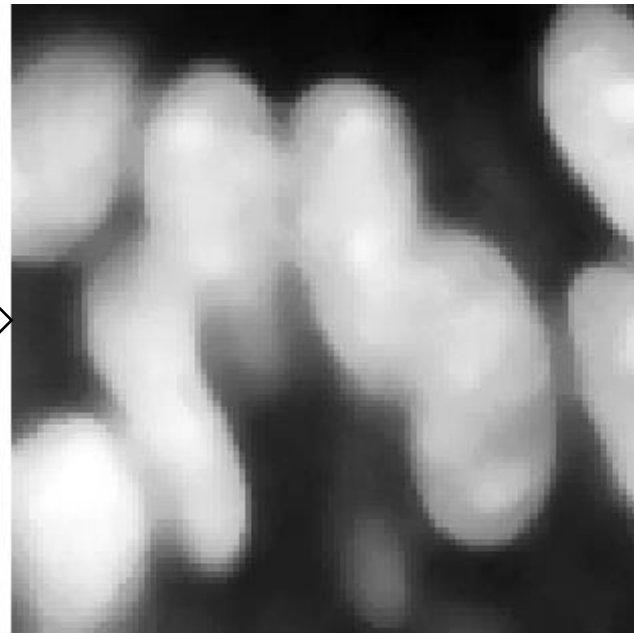
Top-hat filter

- Background subtraction



Quiz: Noise removal

- The median filter is a ...

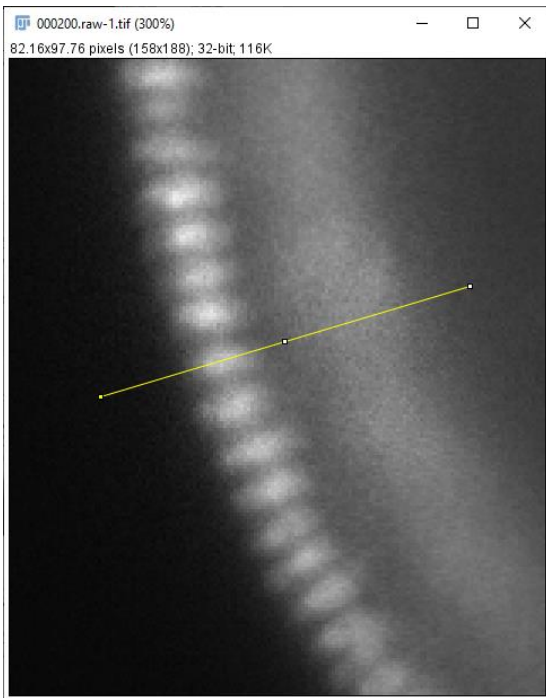


Linear filter

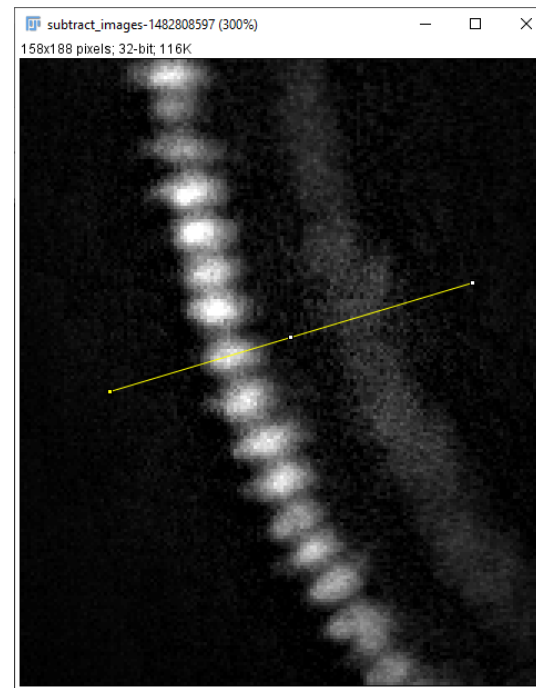
Non-linear filter

Background removal

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



Top-hat



Low-pass
filter

High-pass
filter



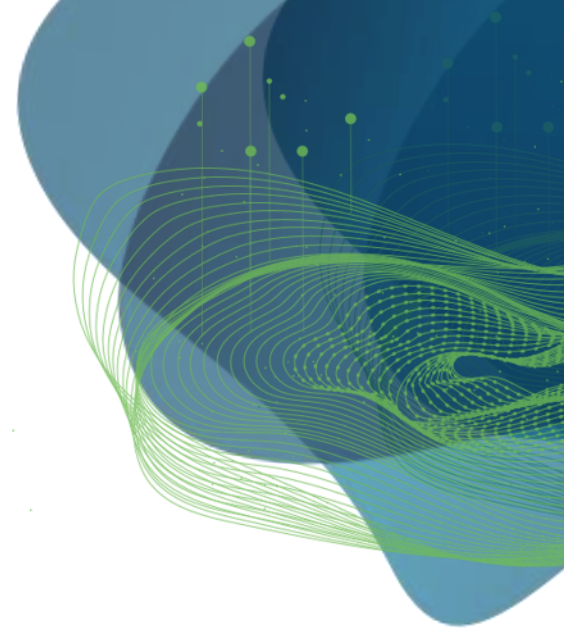
ScaDS.AI

DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Exercises

Robert Haase



GEFÖRDERT VOM



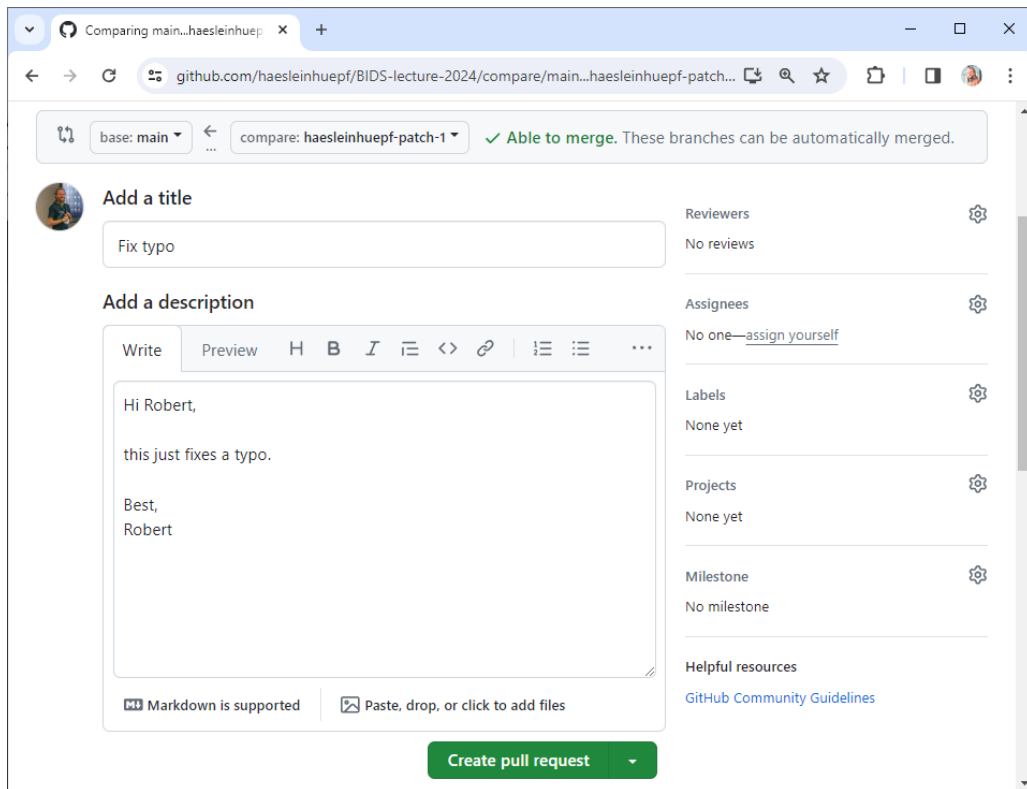
Bundesministerium
für Bildung
und Forschung



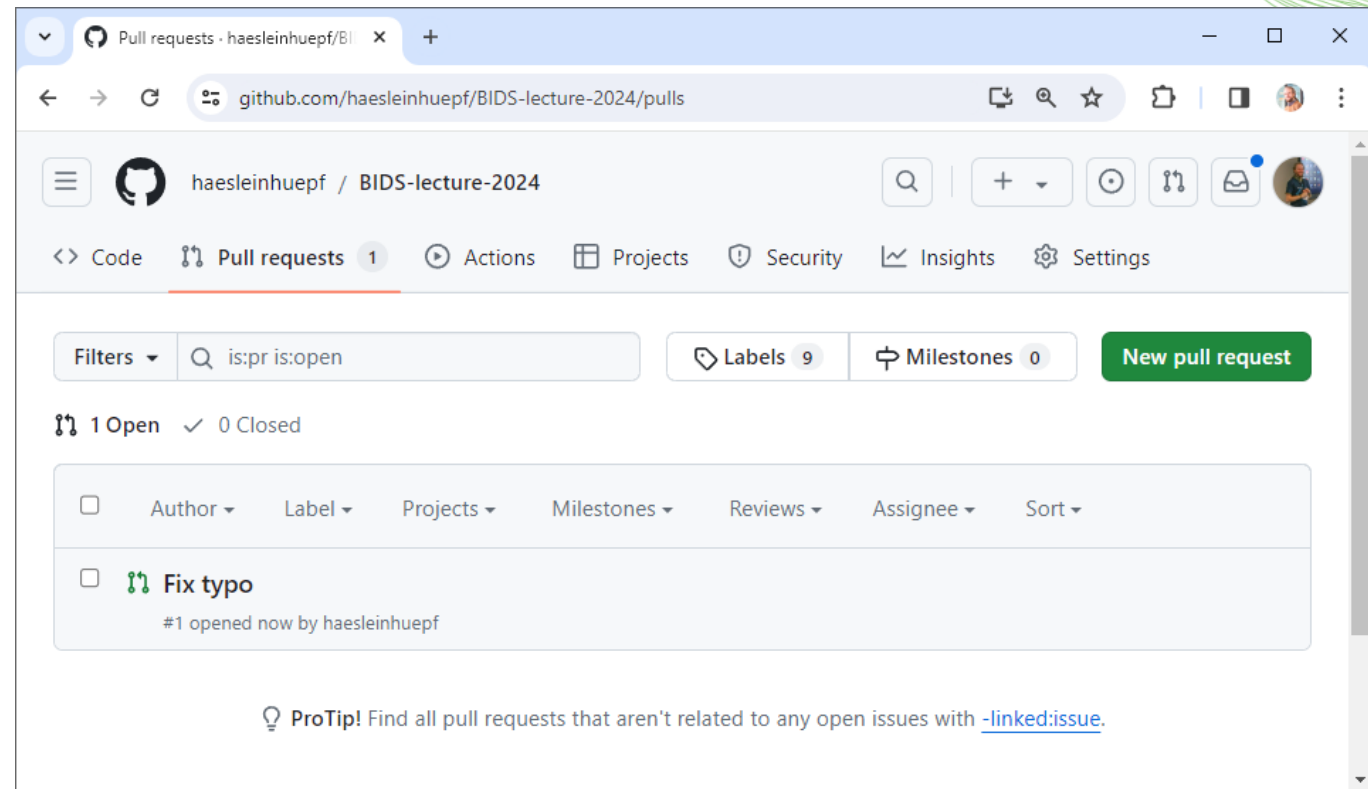
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Exercise: pull-request

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



The screenshot shows the GitHub interface for creating a pull request. The browser address bar shows the URL: `github.com/haesleinhuepf/BIDS-lecture-2024/compare/main...haesleinhuepf-patch-1`. The form includes a title field with the text "Fix typo", a description field with the text "Hi Robert, this just fixes a typo. Best, Robert", and a "Create pull request" button. The interface also shows the "base: main" and "compare: haesleinhuepf-patch-1" branches, and a message indicating that the branches are "Able to merge".



The screenshot shows the GitHub pull requests page for the repository "haesleinhuepf / BIDS-lecture-2024". The browser address bar shows the URL: `github.com/haesleinhuepf/BIDS-lecture-2024/pulls`. The page displays a list of pull requests with one open pull request titled "Fix typo" by the user "haesleinhuepf". The page also shows filters, labels, milestones, and a "New pull request" button.

Exercise: image processing

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

Exercise

Open the `banana0020.tif` data set from last week and visualize it in a yellowish colormap.

```
[ ]:
```

Exercise

Write a function that computes the Difference of Gaussian.

```
[ ]: def difference_of_gaussian(image, sigma1, sigma2):  
    # enter code here
```

Use a simple function call to try out the function.

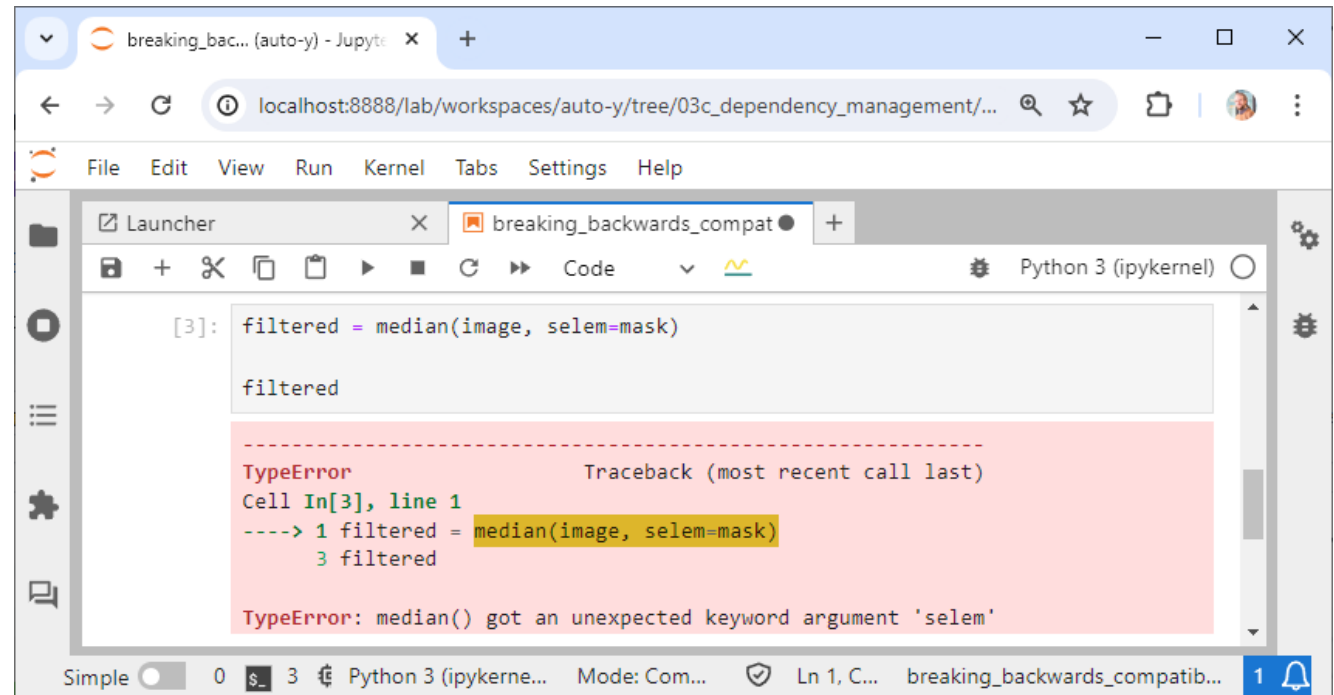
```
[ ]: dog_image = difference_of_gaussian(image3, 1, 5)  
imshow(dog_image)
```

Use the `stackview` library to play with it interactively.

```
[ ]: stackview.Interact(difference_of_gaussian, image3)
```

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



```
[3]: filtered = median(image, selem=mask)

filtered

-----
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'
```