

Vision ~~Large~~ Language Models for Bio-Image Data Science

Robert Haase

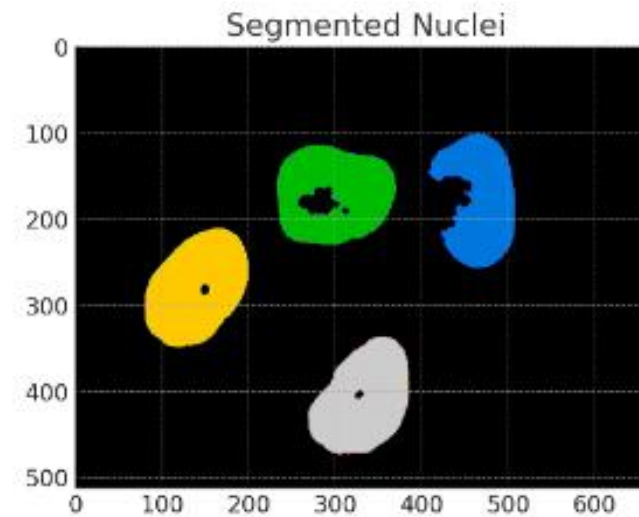
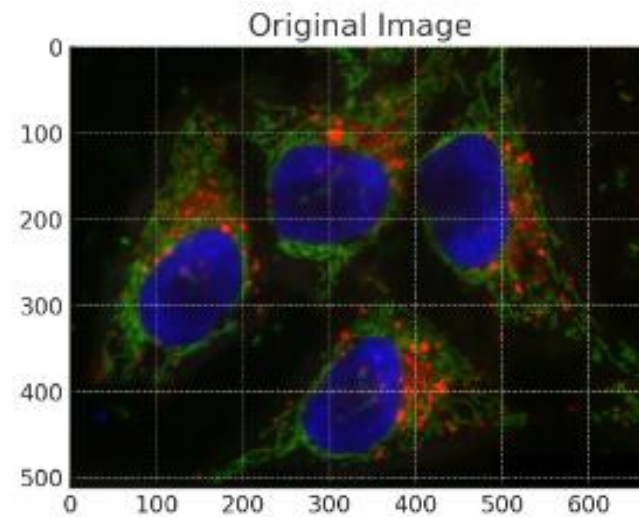


<https://doi.org/10.5281/zenodo.15735577>

These slides can be reused under the terms of the [CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/) license.

Bio-image Analysis

- My job ...

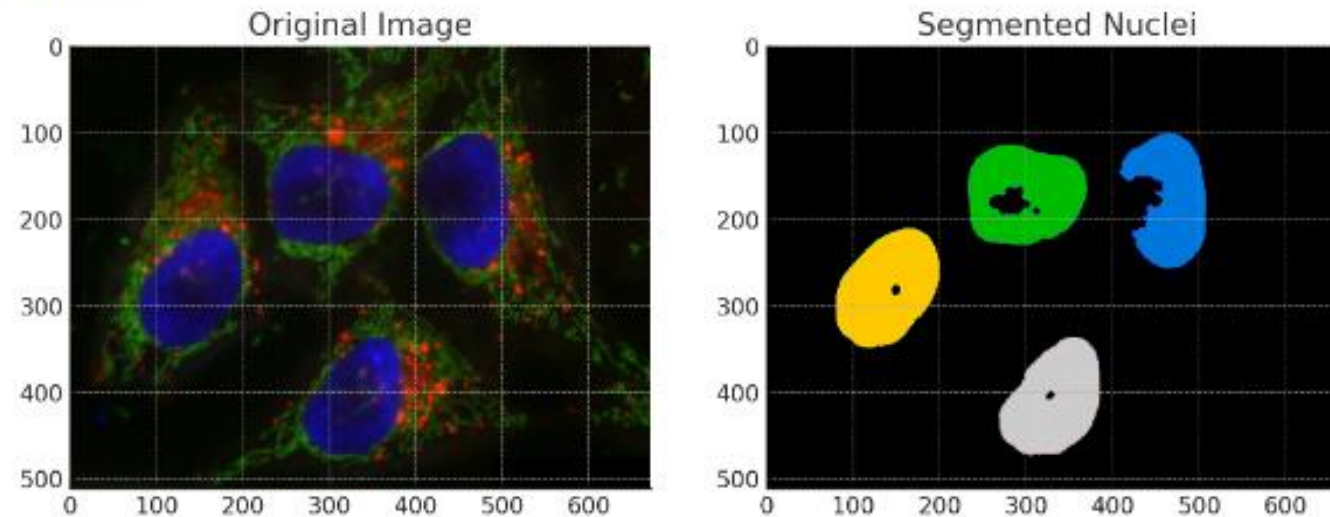


Bio-image Analysis

- My job is changing, since we have ChatGPT

Prompt:
“Please segment the
nuclei in the blue
channel of the
image”

ChatGPT

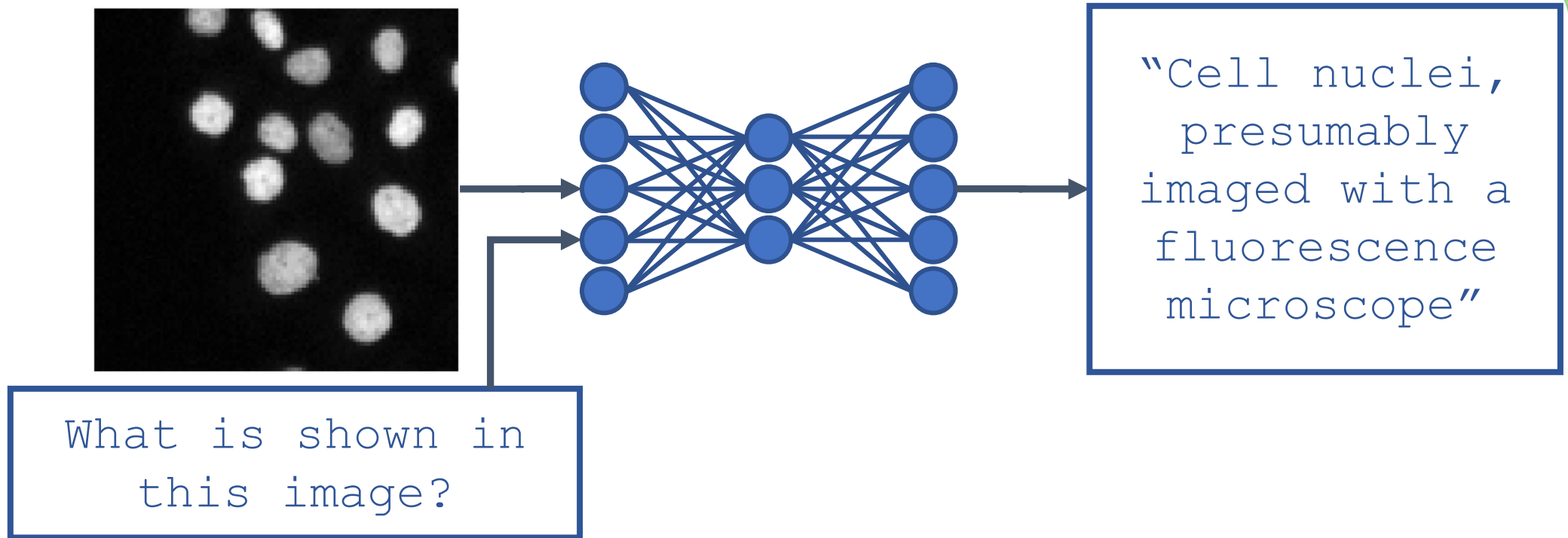


The segmentation of the nuclei has been performed on the blue channel of the image. The right side of the figure shows the segmented nuclei in different colors, each corresponding to a different nucleus in the field of view. Each color represents a unique label assigned to the individual nuclei detected in the image. [↗]



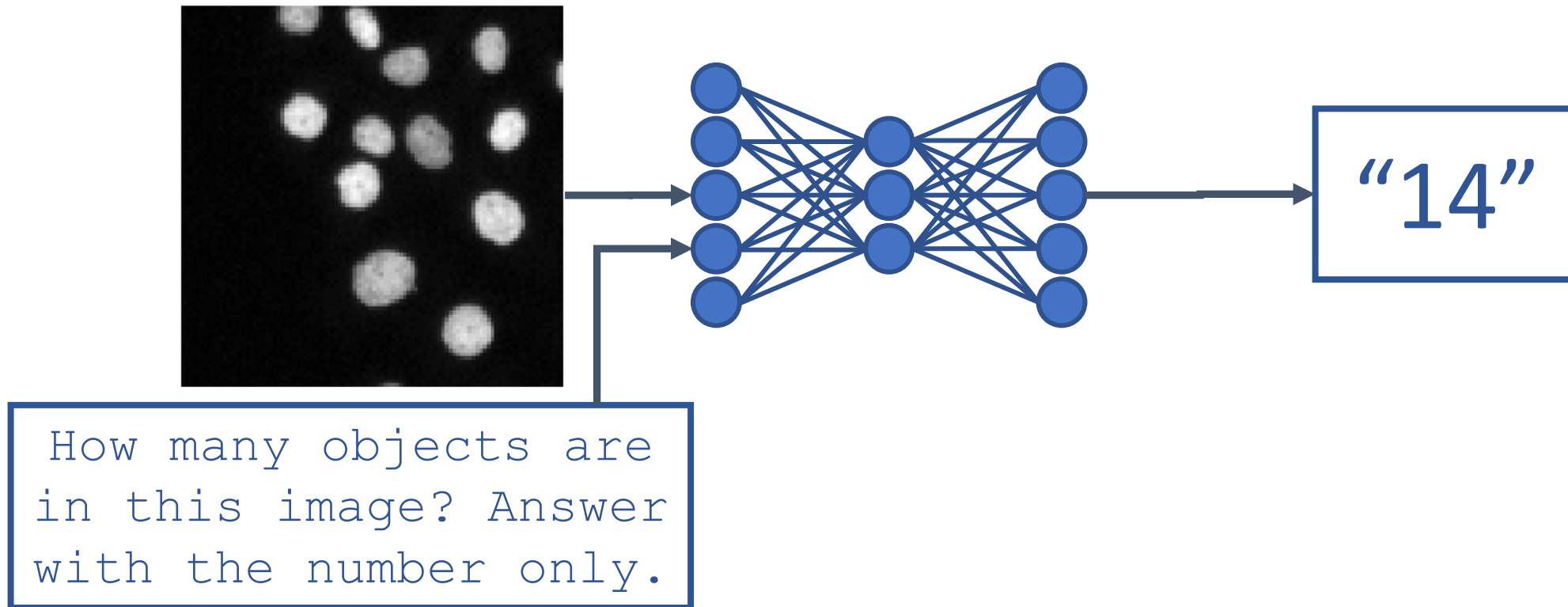
Vision language models

- Combine image + text to generate text



Vision language models for counting objects

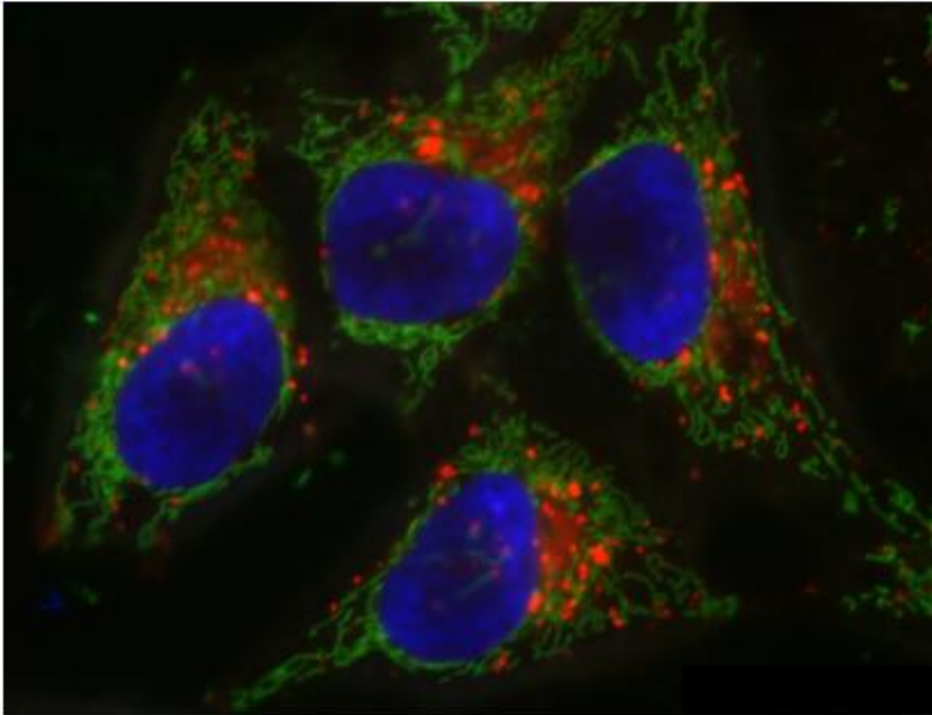
- Combine image + text to generate text



Vision language models for counting objects

VLMs were not exactly built for this purpose...

You



How many blue nuclei are in this image?



ChatGPT

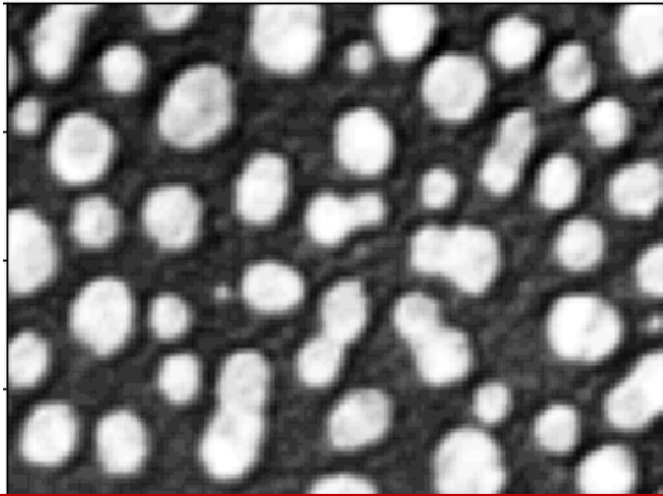
There are three blue nuclei visible in this image.



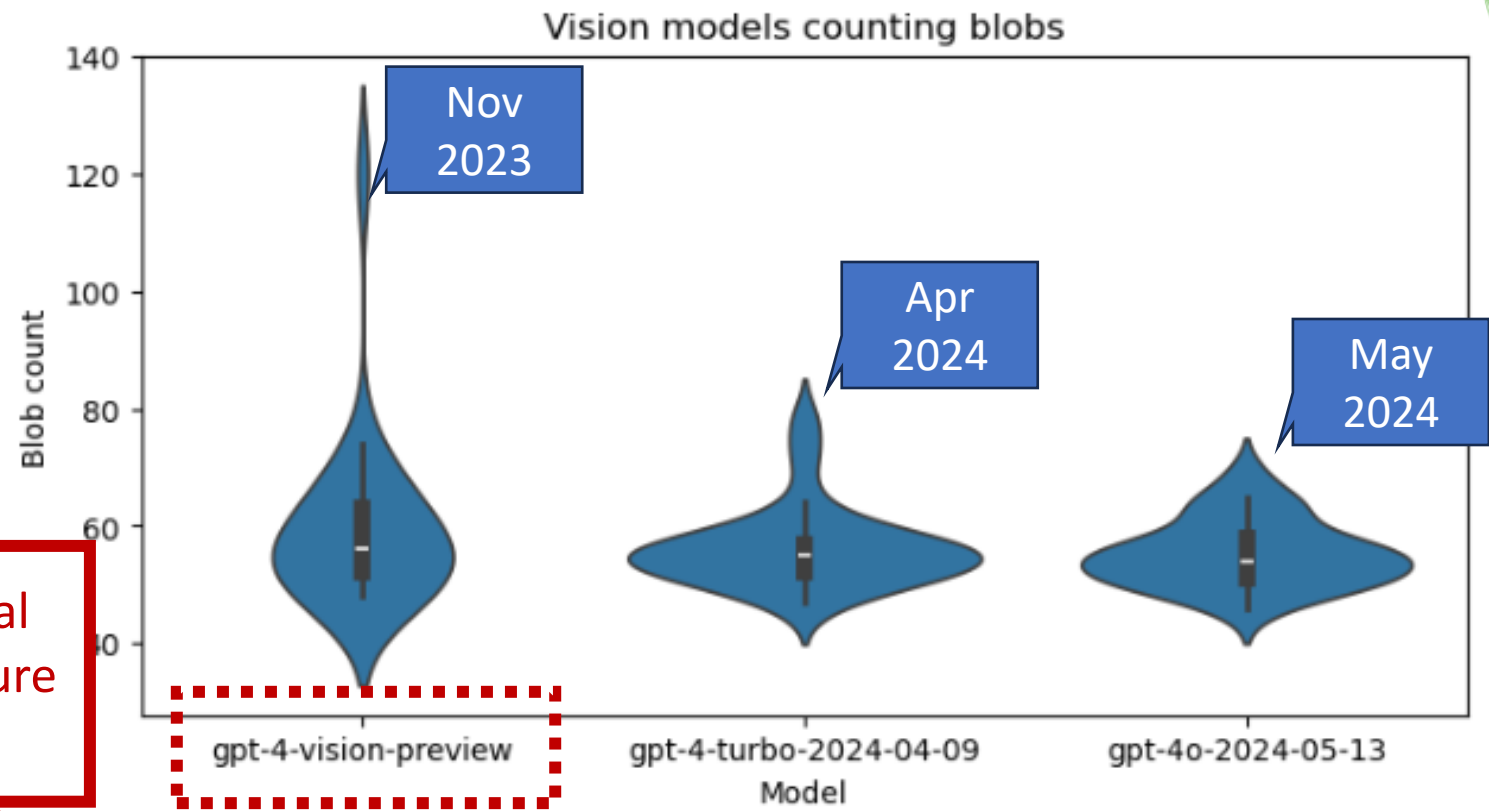
$n=1$

Vision language models for counting objects

- Prompt: „Analyse the following image by counting the bright blobs. Respond with the number only.“ (n=25)



Lesson learned: We need physical control of models and infrastructure to secure reproducibility!

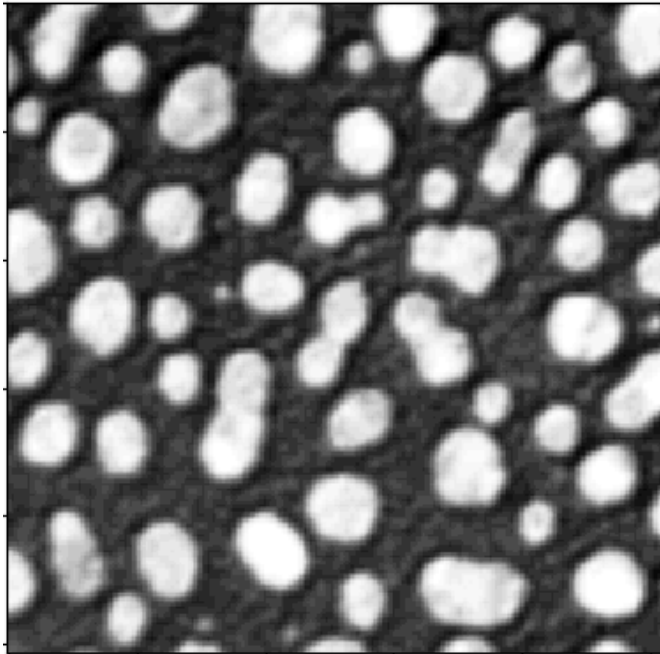


Model no longer available (Apr. 2025)

https://scads.github.io/generative-ai-notebooks/80_benchmarking_llms/20_vision_models.html

Vision language models for counting objects

- Prompt-engineering also works with VLMS



Baseline: Otsu-threshold

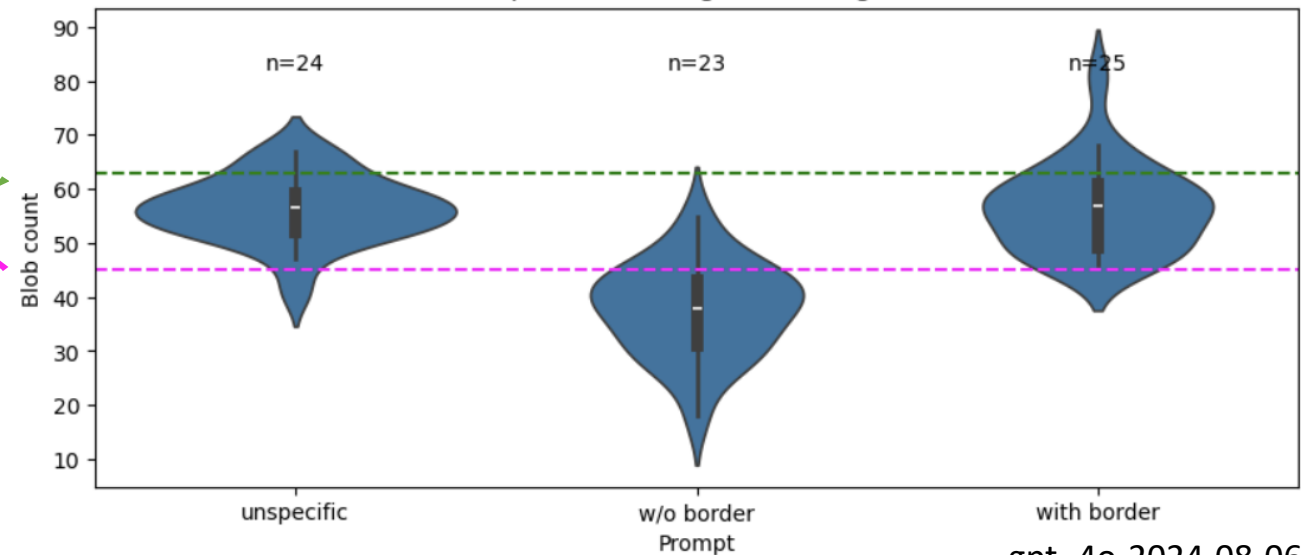
Baseline: Otsu-threshold, excl. borders

“Analyse the following image by counting the bright blobs. Respond ONLY the number.”

“Analyse the following image by counting the bright blobs. **Ignore the objects touching the image border.** Respond ONLY the number.”

“Analyse the following image by counting the bright blobs, **including the objects touching the image border.** Respond ONLY the number.”

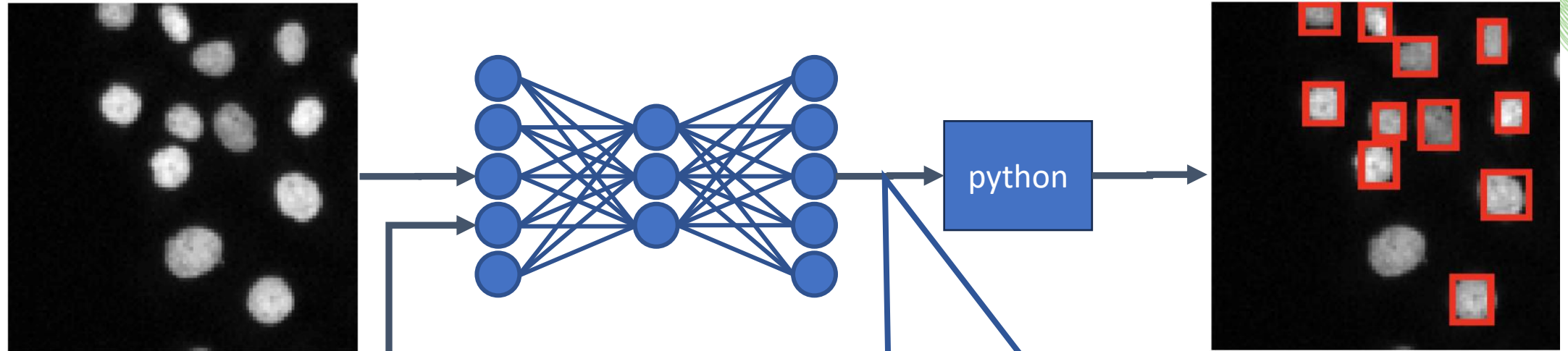
Prompts for counting blobs using VLMS



gpt -4o-2024-08-06

VLMs for bounding box segmentation

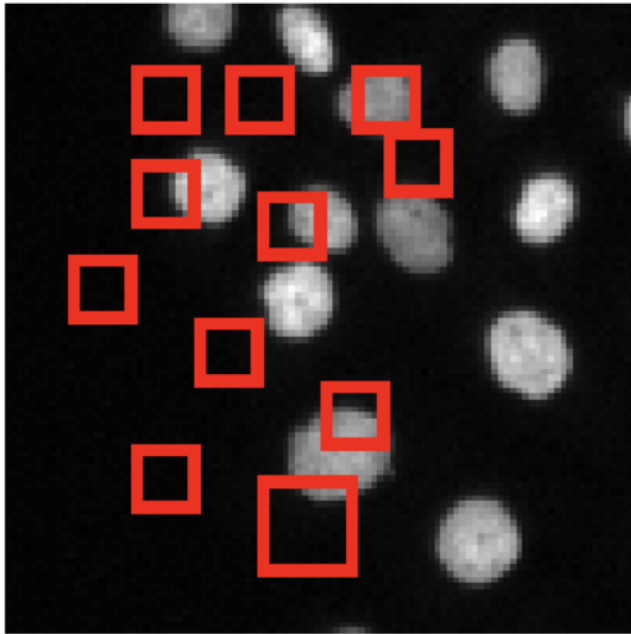
- Recap: VLMs combine images + text to produce text



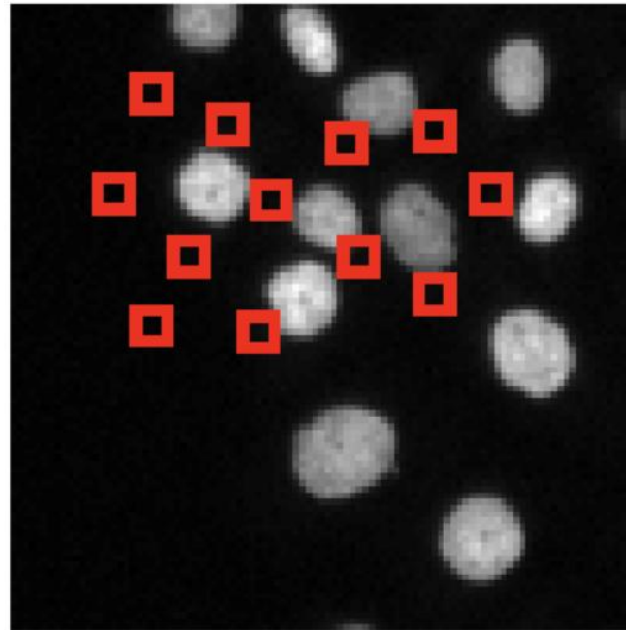
Give me a json object
of bounding boxes
around ALL bright
blobs in this image...

```
[{"x": 0.191, "y": 0.111,...},  
{"x": 0.313, "y": 0.161,...},...]
```

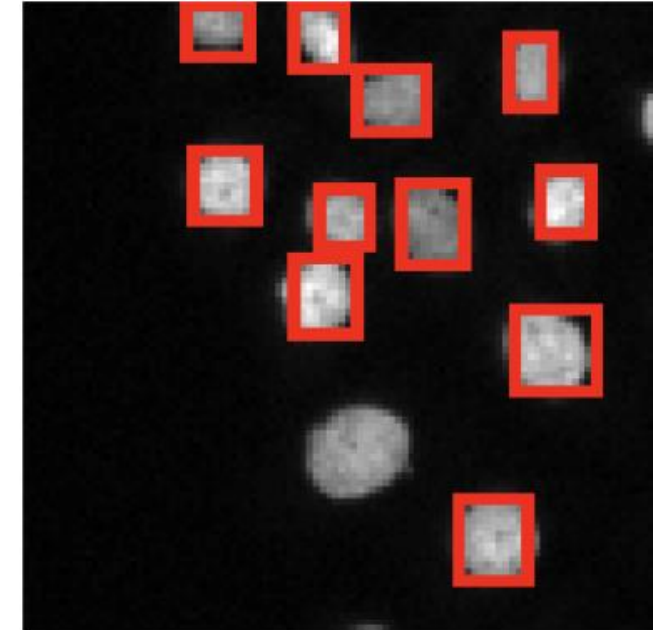
VLMs for bounding box segmentation



GPT-4o



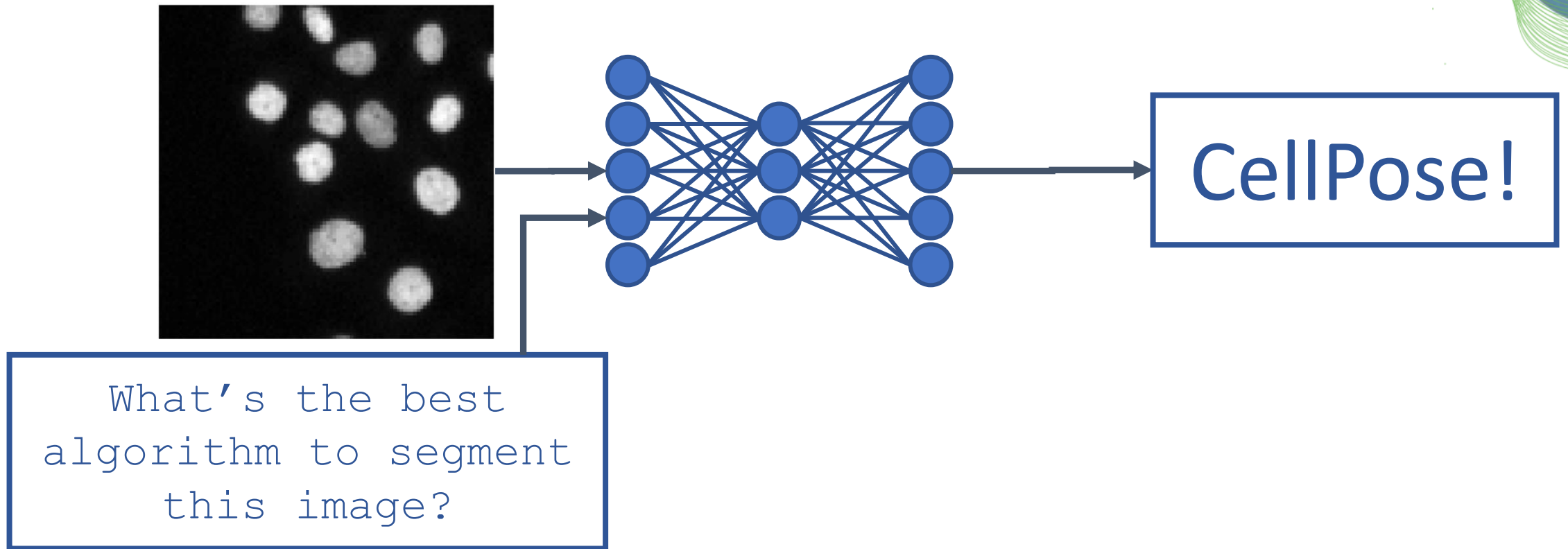
Claude 3.7 Sonnet



Moondream 2B

VLMs *guessing* segmentation algorithms

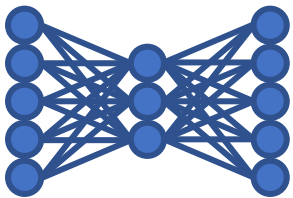
- Combine image + text to generate text



LLMs *guessing* segmentation algorithms

```
prompt = """You are a bioimage-analysis expert.  
What is the best image processing algorithm to segment microscopy images?  
Answer the algorithm name only. No explanations.  
"""
```

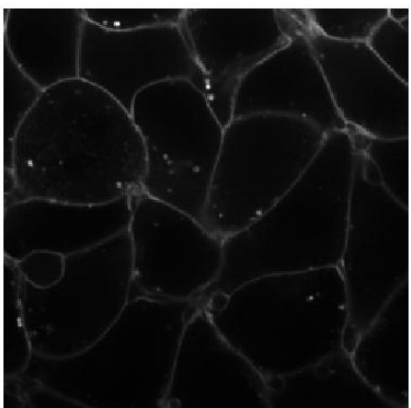
+ no image



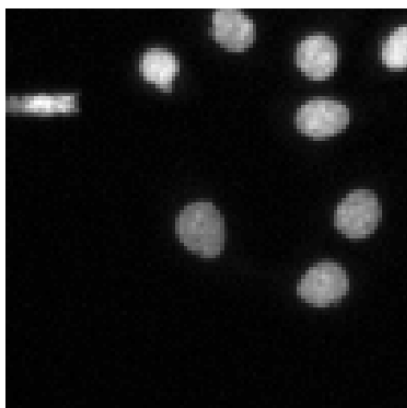
learning
cellpose
deep g
deeplabunet
e

VLMs guessing segmentation algorithms

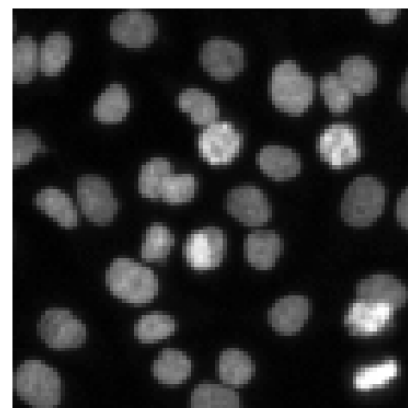
```
prompt = """You are a bioimage-analysis expert.  
What is the best image processing algorithm to segment this microscopy image?  
Answer the algorithm name only. No explanations.  
"""
```



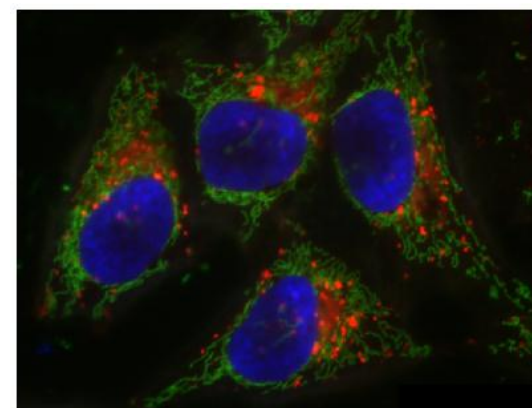
watershed



otsu thresholding
watershed



watershed
otsu thresholding



otsu thresholding
learning deep u
cellpose
watershed
g

VLMs guessing segmentation algorithms

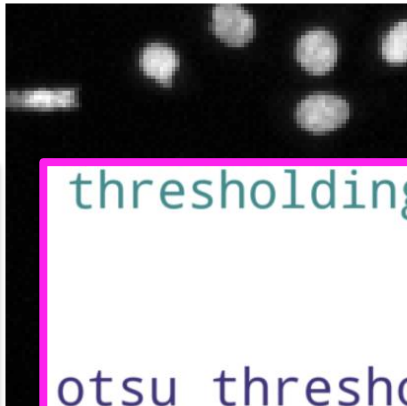
```
prompt = """You are a bioimage-analysis expert. You have a rule-book what algorithms to use for specific images.
```

```
## Rules
```

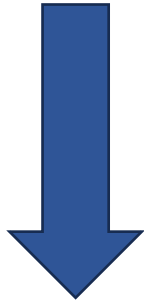
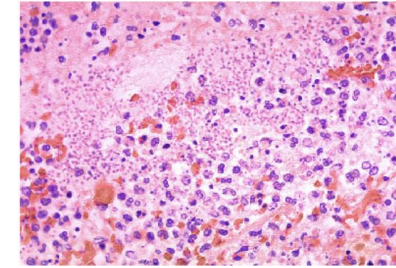
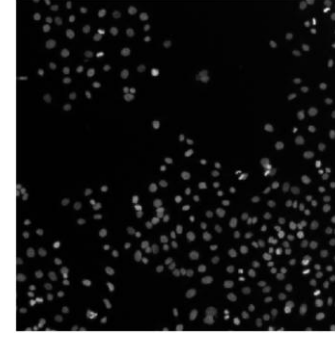
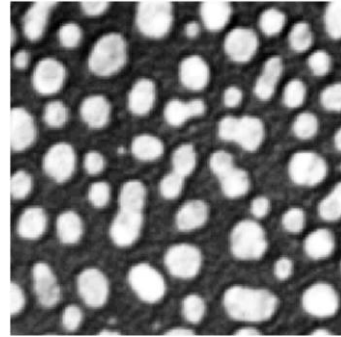
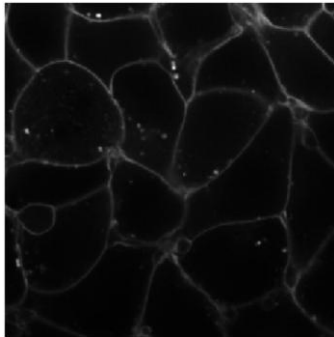
- * If an image shows sparse objects such as nuclei, use Otsu-thresholding for segmenting them.
- * If an image shows dense, partially overlapping objects such as nuclei, use StarDist.
- * If an image shows large cell-like structures with bright membranes, use the Watershed algorithm.
- * In case of doubt, use CellPose.

```
## The task
```

```
What is the best image processing algorithm to segment this microscopy image?  
Answer the algorithm name only. No explanations.  
"""
```

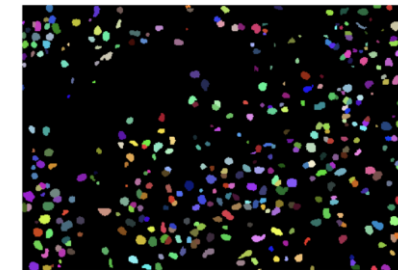
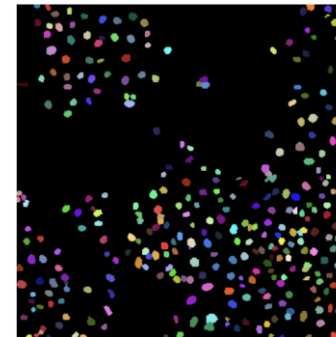
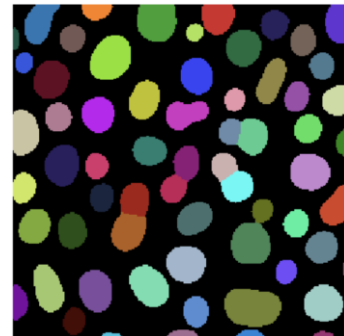


Speaking of Cellpose[-SAM]



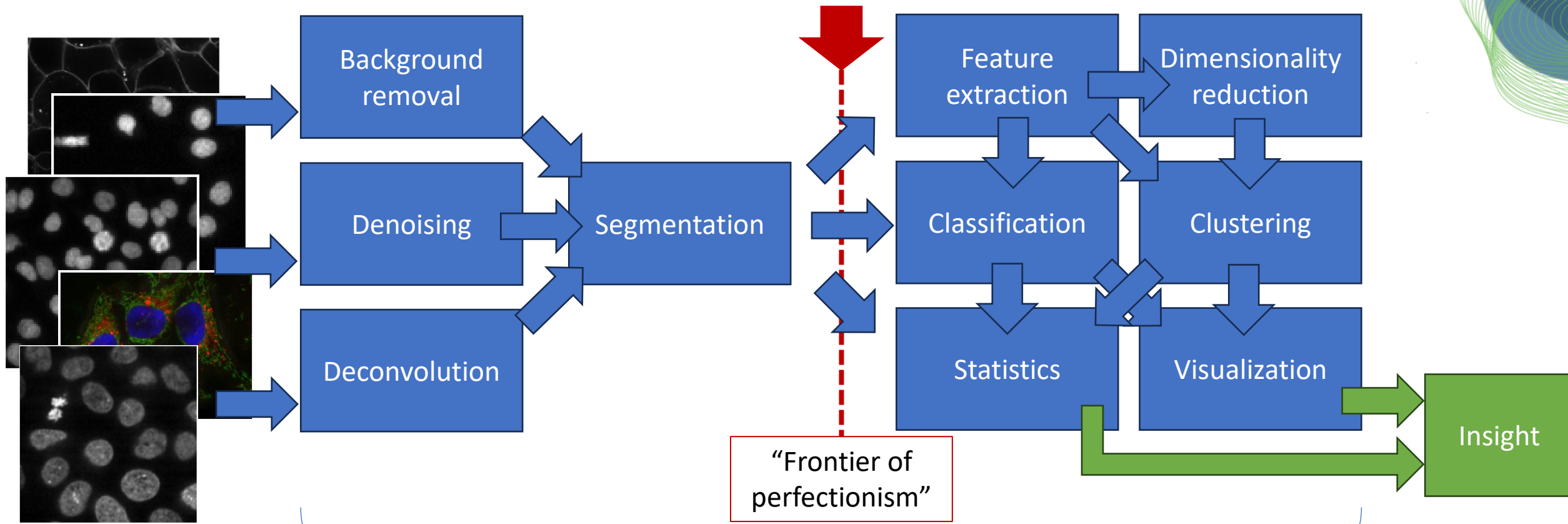
```
masks, flows, styles = model.eval(image,  
    batch_size=32,  
    flow_threshold=0.4,  
    cellprob_threshold=0.0,  
    normalize={"tile_norm_blocksize": 0})  
  
stackview.insight(masks.astype(np.uint32))
```

“Foundation
model”



Bio-image analysis beyond segmentation

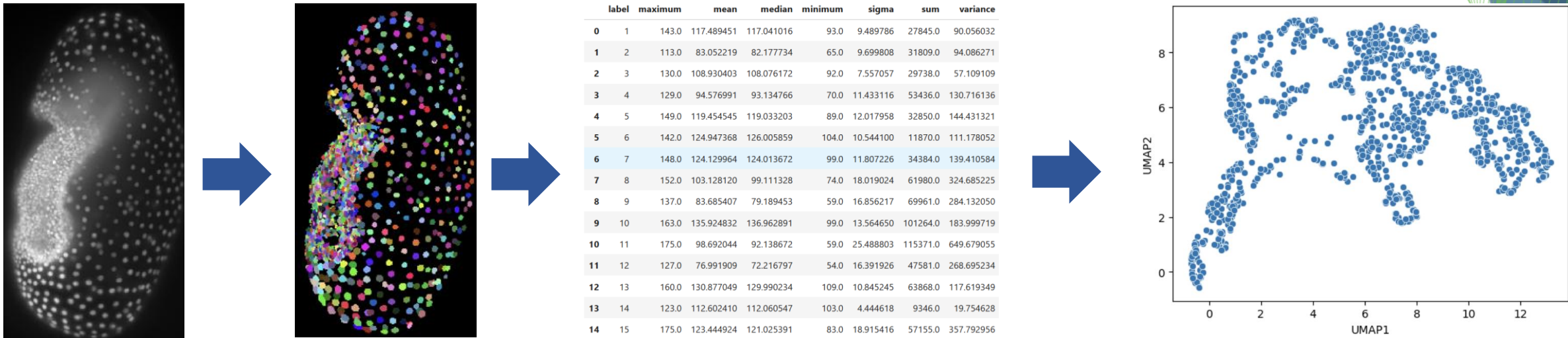
- Crossing this border seems challenging



Wide variety of [computational] skills required to achieve all this

Benchmarking LLMs for Bio-image Analysis

- Use-case: segment an image, extract features and create a UMAP.



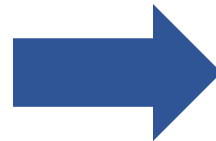
Unit-test pass-rate (n=10):

	reference	gpt-4-turbo-2024-04-09	Claude-3-opus-20240229	gpt-4-1106-preview	gpt-3.5-turbo-1106	gemini-pro	codellama
workflow_segment_measure_umap	1.0	0.8	0.7	0.9	0.1	0.2	0.0

Benchmarking LLMs for Bio-image Analysis

- Use-case: compute the correlation matrix

	label	maximum	mean	median	minimum	sigma	sum	variance	elongation	feret_diameter
0	1	143.0	117.489451	117.041016	93.0	9.489786	27845.0	90.056032	1.228690	8.774964
1	2	113.0	83.052219	82.177734	65.0	9.699808	31809.0	94.086271	1.325096	13.152946
2	3	130.0	108.930403	108.076172	92.0	7.557057	29738.0	57.109109	1.565911	12.884099
3	4	129.0	94.576991	93.134766	70.0	11.433116	53436.0	130.716136	1.227027	14.352700
4	5	149.0	119.454545	119.033203	89.0	12.017958	32850.0	144.431321	1.429829	10.723805
5	6	142.0	124.947368	126.005859	104.0	10.544100	11870.0	111.178052	1.196911	9.273618
6	7	148.0	124.129964	124.013672	99.0	11.807226	34384.0	139.410584	1.137575	13.114877
7	8	152.0	103.128120	99.111328	74.0	18.019024	61980.0	324.685225	1.373404	14.866069
8	9	137.0	83.685407	79.189453	59.0	16.856217	69961.0	284.132050	1.205095	14.456832
9	10	163.0	135.924832	136.962891	99.0	13.564650	101264.0	183.999719	1.169756	14.730920
10	11	175.0	98.692044	92.138672	59.0	25.488803	115371.0	649.679055	1.201427	16.911535
11	12	127.0	76.991909	72.216797	54.0	16.391926	47581.0	268.695234	1.272889	15.066519
12	13	160.0	130.877049	129.990234	109.0	10.845245	63868.0	117.619349	1.143453	13.152946
13	14	123.0	112.602410	112.060547	103.0	4.444618	9346.0	19.754628	2.025953	9.110434
14	15	175.0	123.444924	121.025391	83.0	18.915416	57155.0	357.792956	1.339038	13.892444



	label	maximum	mean	median	minimum	sigma	sum	variance	elongation	feret_diameter
label	1.000000	-0.605035	-0.651268	-0.637158	-0.581233	-0.136779	-0.170934	-0.134539	-0.014857	0.105859
maximum	-0.605035	1.000000	0.824653	0.787135	0.577706	0.576118	0.522637	0.563160	-0.028076	0.144944
mean	-0.651268	0.824653	1.000000	0.997051	0.918750	0.040456	0.157387	0.052848	0.122359	-0.173872
median	-0.637158	0.787135	0.997051	1.000000	0.928168	-0.013701	0.111119	-0.002710	0.133479	-0.205403
minimum	-0.581233	0.577706	0.918750	0.928168	1.000000	-0.317573	-0.057844	-0.273489	0.217240	-0.311868
sigma	-0.136779	0.576118	0.040456	-0.013701	-0.317573	1.000000	0.633356	0.959845	-0.250655	0.436449
sum	-0.170934	0.522637	0.157387	0.111119	-0.057844	0.633356	1.000000	0.602559	-0.074555	0.775197
variance	-0.134539	0.563160	0.052848	-0.002710	-0.273489	0.959845	0.602559	1.000000	-0.191963	0.370870
elongation	-0.014857	-0.028076	0.122359	0.133479	0.217240	-0.250655	-0.074555	-0.191963	1.000000	0.184445
feret_diameter	0.105859	0.144944	-0.173872	-0.205403	-0.311868	0.436449	0.775197	0.370870	0.184445	1.000000

Unit-test pass-rate (n=10):

	reference	gpt-4-turbo-2024-04-09	Claude-3-opus-20240229	gpt-4-1106-preview	gpt-3.5-turbo-1106	gemini-pro	codellama
pair_wise_correlation_matrix	1.0	1.0	1.0	0.9	1.0	0.5	0.1

Benchmarking LLMs for Bio-image Analysis

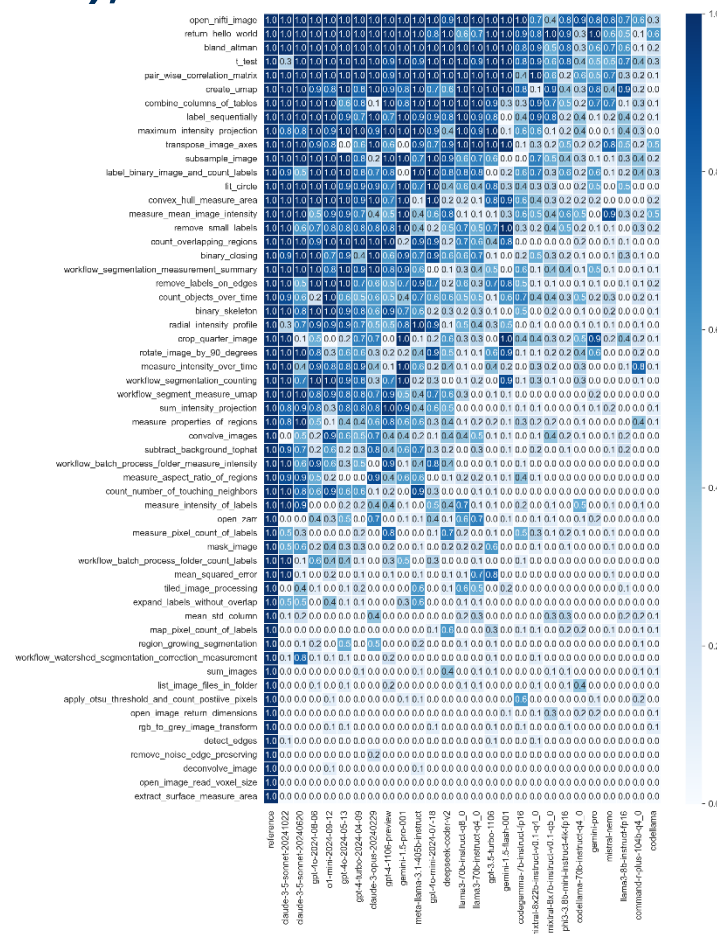
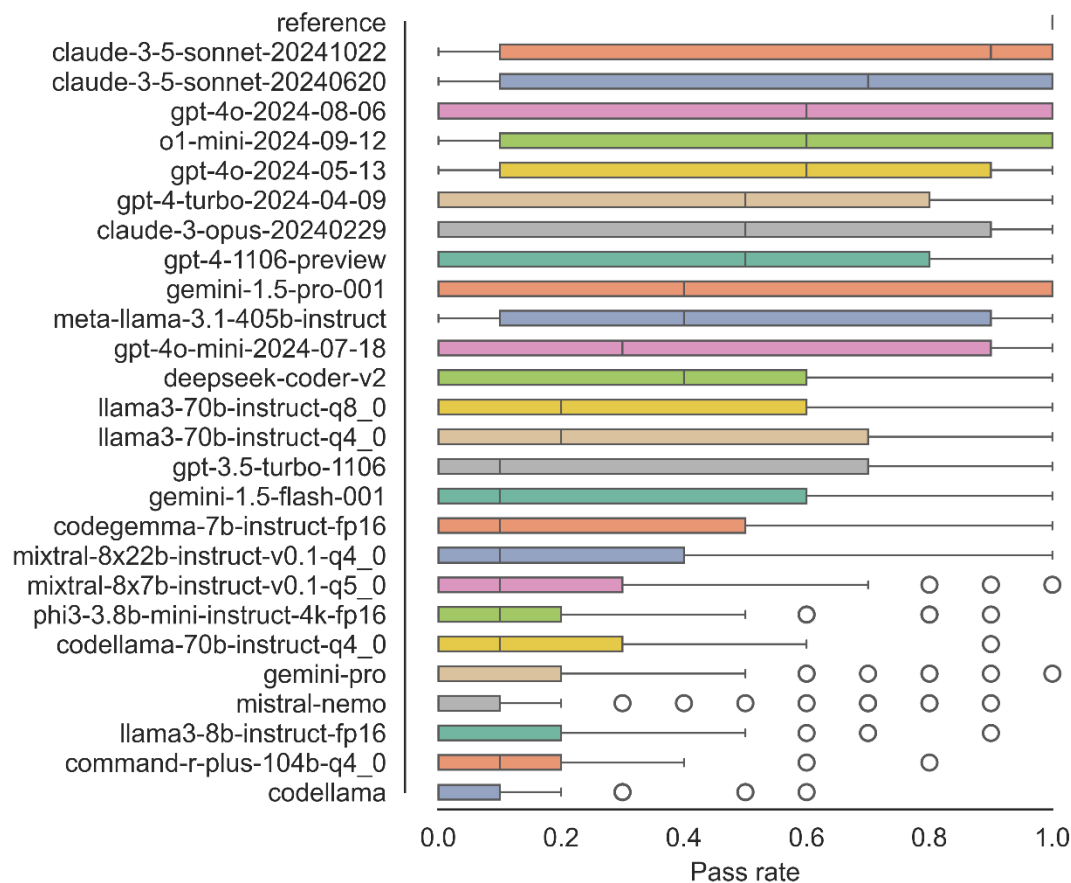
- Use case: Open a zarr file



Unit-test pass-rate (n=10):

	reference	gpt-4-turbo-2024-04-09	Claude-3-opus-20240229	gpt-4-1106-preview	gpt-3.5-turbo-1106	gemini-pro	codellama
open_zarr	1.0	0.0	0.7	0.0	0.0	0.2	0.0

- Summary: 57 use-cases (yet), 26 LLMs (yet), n=10



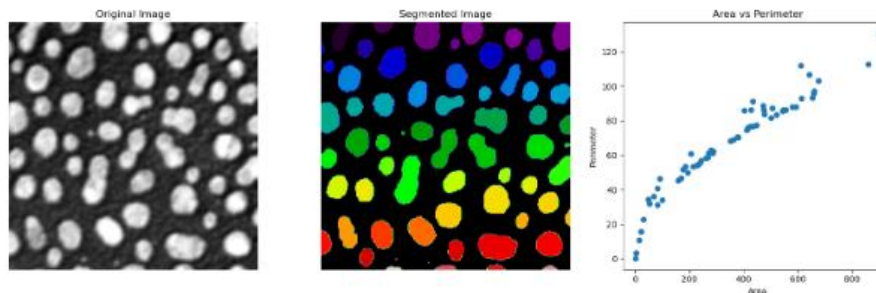
AI-assisted bio-image analysis in Jupyter

- **bia-bob**: long-context prompting for image-analysis code.

```
[1]: import bia_bob
```

```
[2]: %%bob
* Load blobs.tif from disk
* Segment the bright objects on the dark background
* Measure area and perimeter of the objects
* Show the original image, the segmented image and an
  area versus perimeter plot side-by-side using subfigures.
```

This notebook may contain text, code and images generated by artificial intelligence. Used model: claude-3-5-sonnet-20240620, vision model: claude-3-5-sonnet-20240620, endpoint: None, bia-bob version: 0.21.0.. Do not enter sensitive or private information and verify generated contents according to good scientific practice. Read more: <https://github.com/haesleinhuepf/bia-bob#disclaimer>



	label	area	perimeter
0	1	433.0	91.254834
1	2	185.0	53.556349
2	3	658.0	95.698485
3	4	434.0	76.870058
4	5	477.0	83.798990

```
import numpy as np
from skimage import io, filters, measure, segmentation
import matplotlib.pyplot as plt
import pandas as pd
import pyclesperanto_prototype as cle
import stackview

# Load the image
image = io.imread('blobs.tif')

# Segment the bright objects
thresh = filters.threshold_otsu(image)
binary = image > thresh

# Label the objects
labels = measure.label(binary)

# Measure properties
properties = ['label', 'area', 'perimeter']
df = pd.DataFrame(measure.regionprops_table(labels, properties=properties))

# Create the figure with subfigures
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15, 5))

# Original image
ax1.imshow(image, cmap='gray')
ax1.set_title('Original Image')
ax1.axis('off')

# Segmented image
ax2.imshow(labels, cmap='nipy_spectral')
ax2.set_title('Segmented Image')
ax2.axis('off')

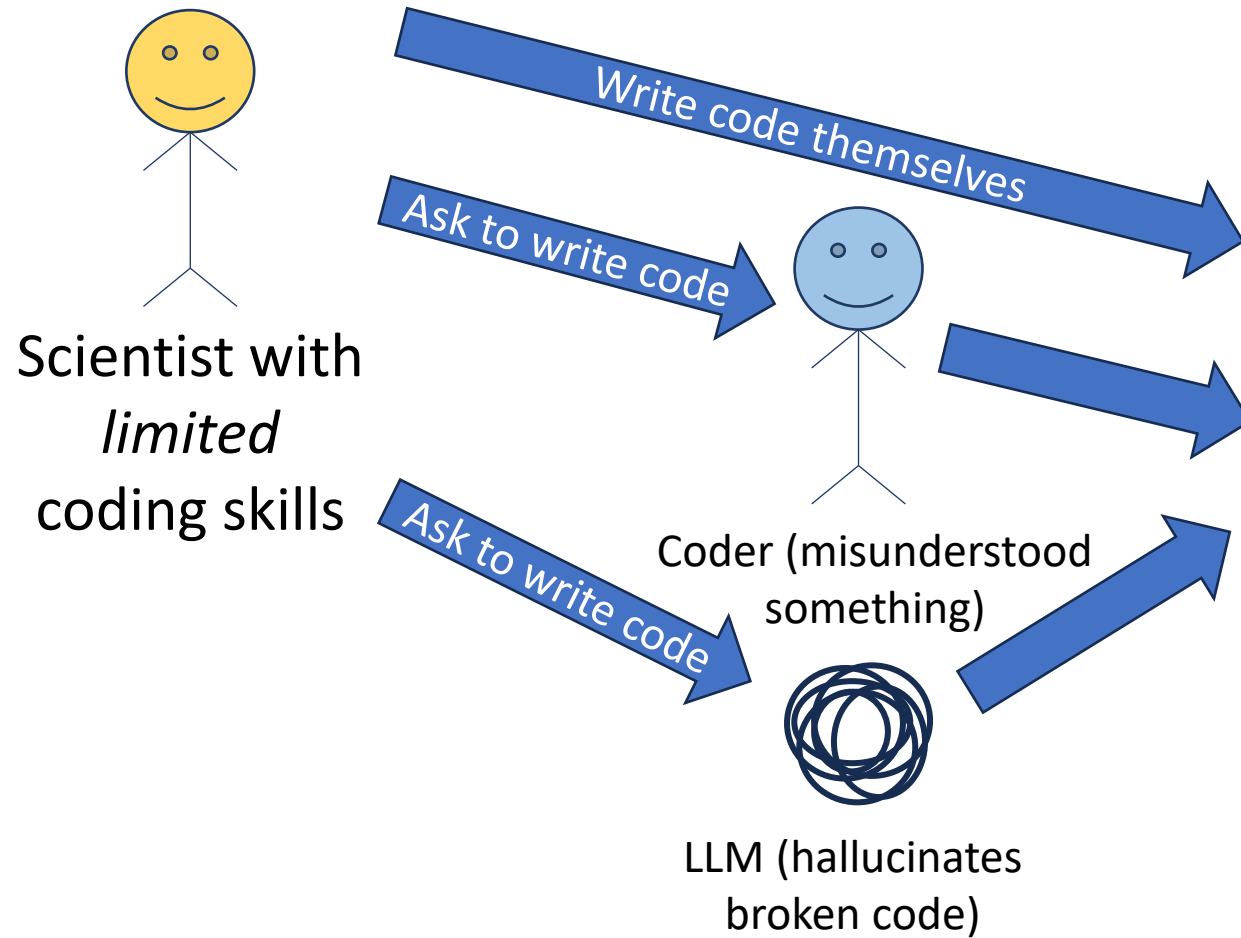
# Scatter plot
ax3.scatter(df['area'], df['perimeter'])
ax3.set_xlabel('Area')
ax3.set_ylabel('Perimeter')
ax3.set_title('Area vs Perimeter')

plt.tight_layout()
plt.show()

# Display the dataframe
print(df.head())
```



Unprompted feedback



```
[1]: import unprompted
```

Cheap, fast

Hi, this is *unprompted* 0.1.1 using gpt-4.1-nano under the hood. Following code cells and related output will be interpreted by AI to provide feedback and suggest improvements. If you want to keep code and/or its output private, do not use this tool or configure it to use a local LLM. Check the documentation for details. Also *unprompted* does mistakes. Treat its suggestions carefully.

```
[2]: distance = 5
      time = 3

      speed = distance * time
```

► unprompted feedback: Change to `speed = distance / time` to reflect the correct physical relationship.

Unprompted feedback may facilitate identifying issues in code early

Unprompted feedback from an AI-assistant cannot replace human peer-review

Can LLMs solve real-world GitHub issues?

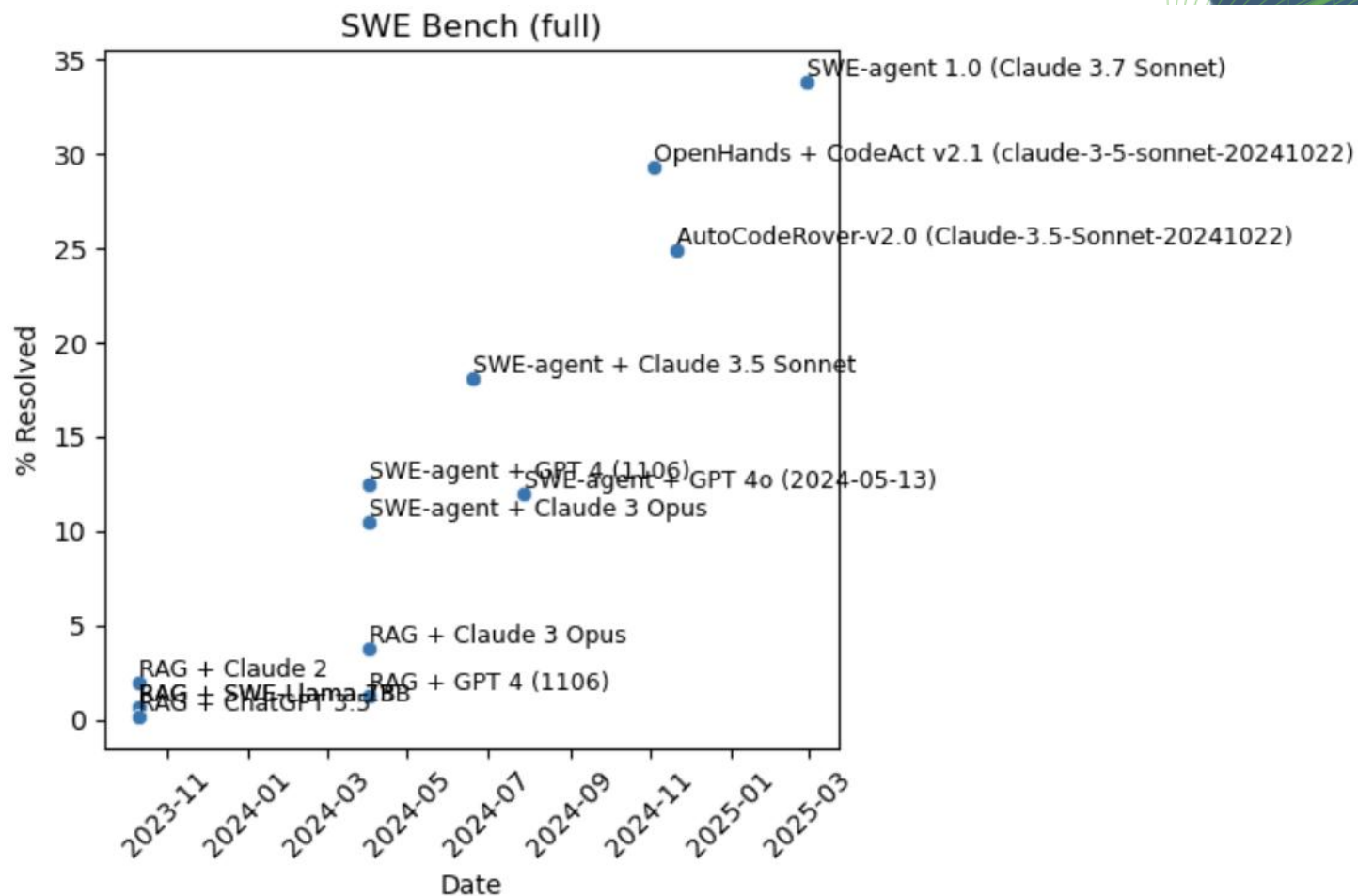
Published as a conference paper at ICLR 2024

SWE-BENCH: CAN LANGUAGE MODELS RESOLVE REAL-WORLD GITHUB ISSUES?

Carlos E. Jimenez^{*1,2} John Yang^{*1,2} Alexander Wettig^{1,2}
Shunyu Yao^{1,2} Kexin Pei³ Ofir Press^{1,2} Karthik Narasimhan^{1,2}

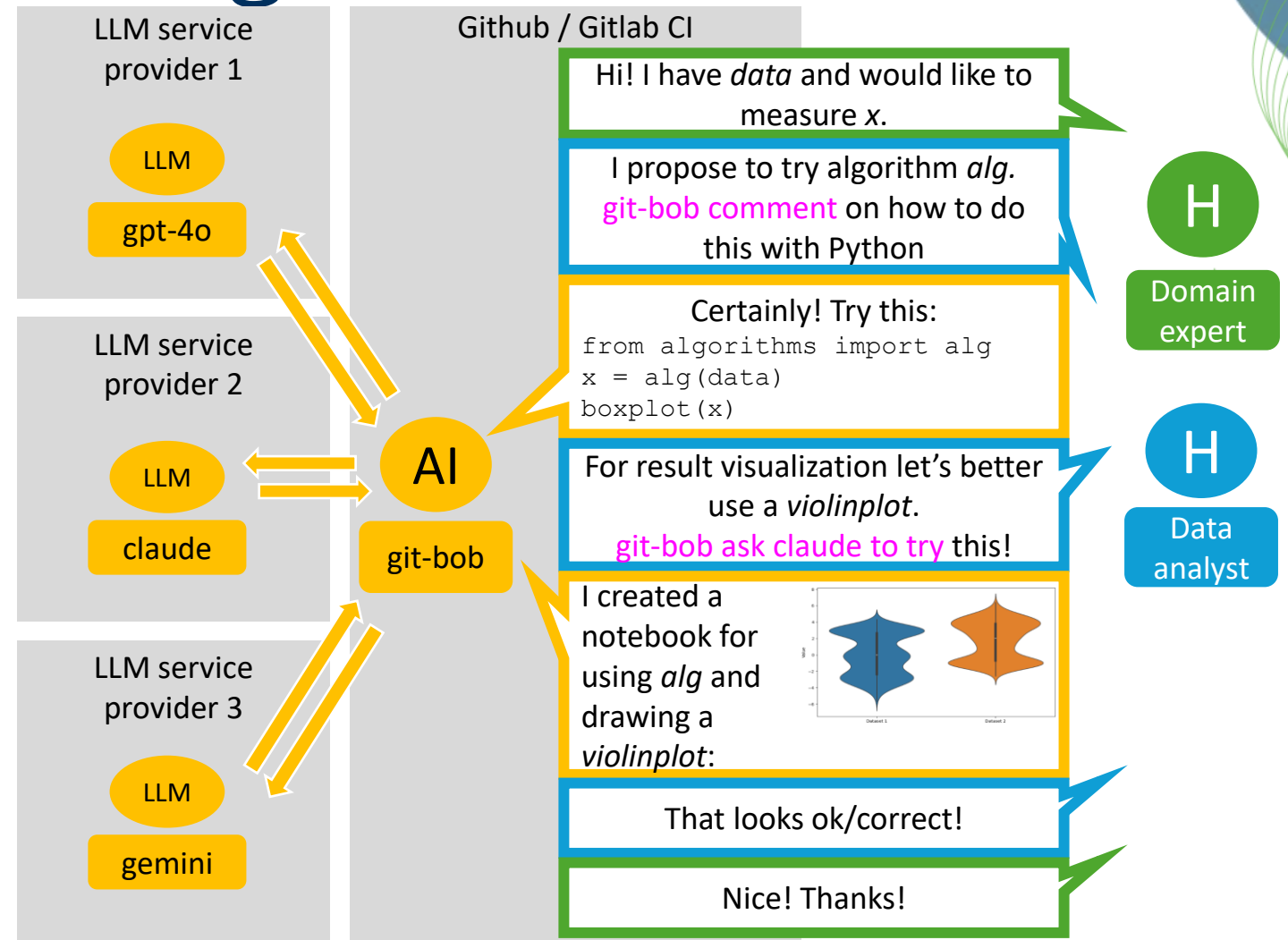
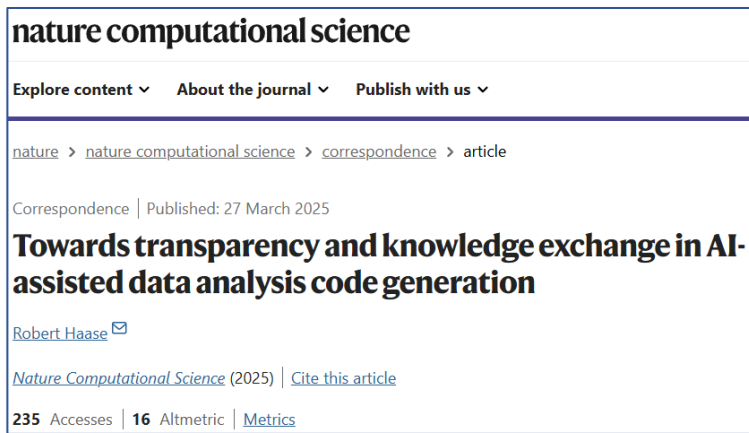
¹Princeton University ²Princeton Language and Intelligence ³University of Chicago

Model	SWE-bench	
	% Resolved	% Apply
Claude 3 Opus	3.79	46.56
Claude 2	1.97	43.07
ChatGPT-3.5	0.17	26.33
GPT-4-turbo	1.31	26.90
SWE-Llama 7b	0.70	51.74
SWE-Llama 13b	0.70	53.62



Collaborative working with AI assistants

- If technical solutions do not exist, socio-cultural solutions may help.
- Human-human-AI interaction
- Transparency
- Knowledge exchange



Chatting with LLMs on GitHub/Lab

- ... about data analysis tasks



haesleinhuepf opened on Oct 12, 2024

Member ...

In the folder "download_statistics", there are multiple files named for example "20240628.csv". The filename without ending is obviously a date, the creation date of the data. Write a jupyter notebook that opens all these files using pandas and plots over time how many records are in these files. Save the notebook in the notebooks directory and make sure the relative link the download_statistics folder is correct.

git-bob try again

Create sub-issue

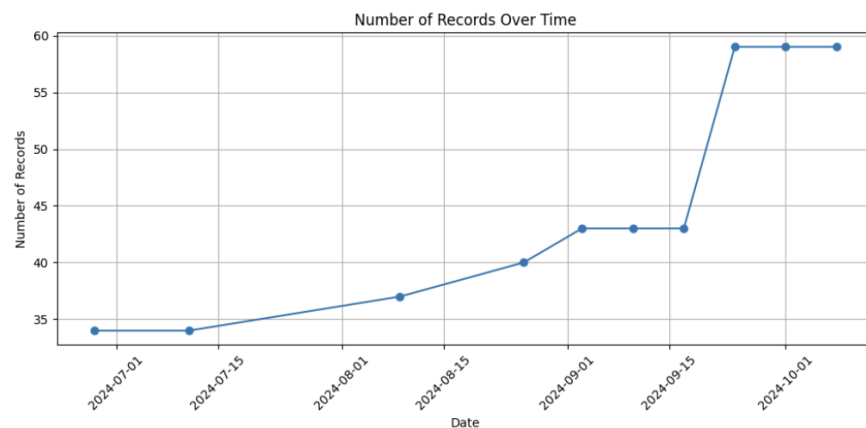


github-actions bot on Oct 12, 2024

Contributor ...

This message was generated by [git-bob](#) (version: 0.8.2, model: github_models:gpt-4o, [log](#)), an experimental AI-based assistant. It can make mistakes and has [limitations](#). Check its messages carefully.

In this update, a Jupyter notebook was created to read CSV files and plot the number of records over time, resulting in the creation of [notebooks/plot_zenodo_links_over_time.ipynb](#). Additionally, an image file was generated:



marabuuu on Oct 14, 2024

Cool, good job git-bob :D



Chatting with LLMs on GitHub/Lab

- ... to receive support on python libraries



haesleinhuepf opened on Oct 7, 2024

Owner ...

I have a 2d+t timelapse dataset as numpy array. I would like to save it as animated gif to disk. Can stackview do this? If yes, how?

git-bob comment

Create sub-issue



1



github-actions bot on Oct 7, 2024

Contributor ...

This message was generated by [git-bob](#) (version: 0.6.4, model: gpt-4o-2024-08-06, [Run details](#)), an experimental AI-based assistant. It can make mistakes and has [limitations](#). Check its messages carefully.

Certainly! Stackview can help you save your 2D+t timelapse dataset as an animated GIF to disk. You can use the `stackview.animate` function for this purpose. Here's a code snippet that demonstrates how to achieve this:

```
import stackview
import numpy as np

# Assuming 'timelapse_data' is your 2D+t timelapse dataset
timelapse_data = np.array(...) # Your numpy array data

# Save the animation as a GIF
stackview.animate(timelapse_data, filename="timelapse_animation.gif", frame_delay_ms=100)
```

This code will take your timelapse data and save it as an animated GIF named `timelapse_animation.gif` with a frame delay of 100ms between frames. Adjust the `frame_delay_ms` as needed to control the speed of the animation.



1

2

LLMs are everywhere

Bioimage-io ChatBot

BioImage.IO Chatbot BioImage Seeker (Melman)

Hi there! I'm Melman. I am help you navigate the bioimaging tools and provide information about bioimaging. How can I help you today?

How can I segment dense nuclei in a fluorescence microscopy image?

To segment dense nuclei in fluorescence microscopy images, you can use several tools and methods available in the bioimaging community. Here are some recommended approaches and tools based on your needs:

1. **Cellpose**: This tool is designed for 2D nuclei segmentation and can handle dense and overlapping nuclei. It outputs a binary mask showing where nuclei were detected. More information can be found [here](#).
2. **CellProfiler**: This software offers a workflow for nuclei segmentation in 2D images. It can detect and segment nuclei, outputting a binary mask. More details are available [here](#).
3. **ImageJ/Fiji**: The Watershed method in ImageJ/Fiji is effective for separating touching and overlapping nuclei. A detailed tutorial on using this method for DAPI-stained nuclei can be found [here](#).
4. **Ilastik**: This tool allows for interactive learning and segmentation. You can create label classes for different objects (e.g., background, nuclei, mitotic nuclei, cytoplasm) and use live prediction mode to refine the segmentation. More information can be found in the [ilastik documentation](#).
5. **Scikit-image**: This Python library provides various functions for image processing, including segmentation. An example of segmenting human cell nuclei using scikit-image can be found in the [scikit-image documentation](#).

These tools and methods should help you effectively segment dense nuclei in your fluorescence microscopy images.

[More Details](#)

Ready to chat? Type your message and press enter!

[Send](#) [Edit Profile](#) [Reset](#) [Feedback](#) [Options](#)

Remaining Message Quota: 19 (reset hourly)

Warning: BioImage.IO Chatbot can make mistakes. Consider checking important information.

Tips: Customize your profile to get personalized answers based on your background.

If you find our chatbot helpful for you, please consider cite us: [arXiv 2310.18351](#)



nature methods

[Explore content](#) [About the journal](#) [Publish with us](#)

[nature](#) > [nature methods](#) > [correspondence](#) > article

Correspondence | Published: 09 August 2024

BioImage.IO Chatbot: a community-driven AI assistant for integrative computational bioimaging

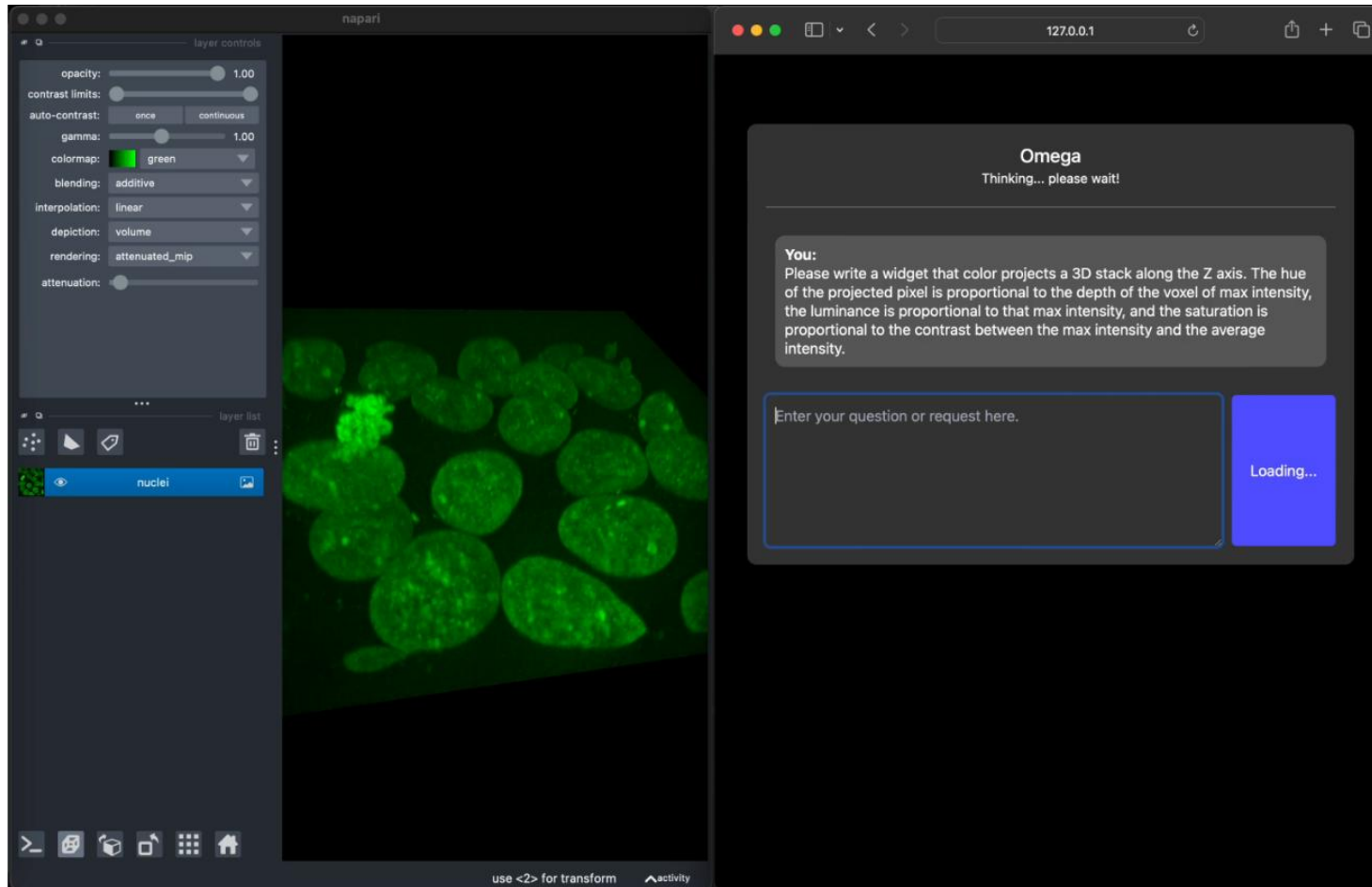
[Wanlu Lei](#), [Caterina Fuster-Barceló](#), [Gabriel Reder](#), [Arrate Muñoz-Barrutia](#) & [Wei Ouyang](#)

[Nature Methods](#) **21**, 1368–1370 (2024) | [Cite this article](#)

865 Accesses | **1** Altmetric | [Metrics](#)

LLMs are everywhere

Napari-chatGPT / Omega



nature methods

Explore content ▾ About the journal ▾ Publish with us ▾

[nature](#) > [nature methods](#) > [correspondence](#) > article

Correspondence | Published: 10 June 2024

Omega – harnessing the power of large language models for bioimage analysis

[Loïc A. Royer](#) ✉

[Nature Methods](#) (2024) | [Cite this article](#)

58 Altmetric | [Metrics](#)

Good Scientific Practice for [AI-assisted code generation]

Robert Haase

These slides can be reused under the terms of the [CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/) license.

Rules...

- „When making their results publicly available, researchers should, in the spirit of research integrity, disclose whether or not they have used generative models, and if so, which ones, for what purpose and to what extent.“

Statement by the Executive Committee
of the Deutsche Forschungsgemeinschaft (DFG,
German Research Foundation) on the Influence of
Generative Models of Text and Image Creation on
Science and the Humanities and on the DFG's
Funding Activities

September 2023

Rules...

5. Good practice when using and developing AI systems

Reviewing content: All AI-generated content should be carefully reviewed by users to mitigate the risks listed above as much as possible. Users are responsible for the content they use/distribute and may be held liable if the content violates the law. This applies in particular if the AI-generated results are used for decision-making or evaluation.

Transparency and reproducibility: Users should always be transparent and openly document when and to what extent AI was used (e.g. via a disclaimer, footnote, note, citation, etc.). Developers of AI systems should make sufficiently transparent to users and experts their basic operation, and the data used for training. In the spirit of Open Science, models and data should be made accessible on trusted infrastructures, following the motto “as open as possible as closed as necessary” in line with the Helmholtz Open Science Policy. In addition, users should always be informed when they are interacting with an AI system rather than a human.

Responsible (non-)use: AI services should not be used if there is a risk of sensitive data being disclosed, e.g., personal data, confidential business data, trade secrets, unpublished research results/data, research proposals, etc.

Use of certain services: Where possible, users should use AI services that provide transparent information about their data sources. For data protection reasons, users should also favor AI systems that are hosted in Europe.

HELMHOLTZ

Spitzenforschung für
große Herausforderungen.

Recommendations for the use of artificial intelligence

Version 1.0

(version approved by the Helmholtz General Assembly on September 18, 2024)

Rules...

4.1 Risk: Privacy violation and unintentional disclosure of information

(Generative) AI systems often store and/or use the information entered by users, for example to further train the AI. In this way, the information entered by users can reach both the service provider and other future users. Users of certain AI systems thus run the risk of passing on sensitive, confidential or personal data to others.

• • •

Recommended action for risk mitigation

- When using (external) AI systems, care should be taken to ensure that **no sensitive data is passed on to the service providers.** This includes
 - Personal data (see GDPR for details)
 - Trade secrets, confidential/strategic business information
 - Unpublished research results/data, scientific manuscripts, research and project proposals
- If AI systems are used nonetheless, care must be taken to ensure that the sensitive data is obfuscated so that it cannot be reconstructed or used by either the external service provider or other users.
- When sensitive data cannot be obfuscated because all information in the document is necessary for a meaningful evaluation, e.g., of scientific manuscripts or research and project

HELMHOLTZ

Spitzenforschung für
große Herausforderungen.

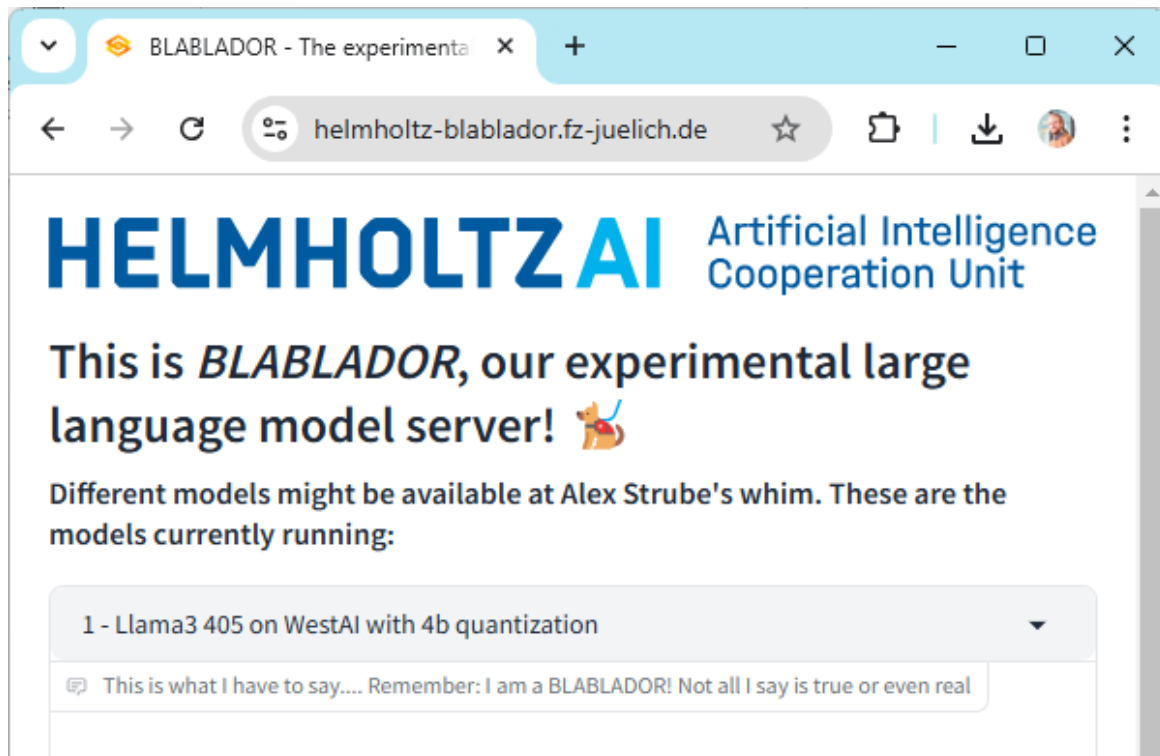
Recommendations for the use of artificial intelligence

Version 1.0

(version approved by the Helmholtz General Assembly on September 18, 2024)

Helmholtz Blablador

- Use your *own* LLM service provider!



```
[1]: from bia_bob import bob
      bob.initialize(endpoint='blablador',
                    model='alias-large')
```

This notebook may contain text, code and images generated by artificial intelligence. Used model: alias-large, vision model: None, endpoint: <https://helmholtz-blablador.fz-juelich.de:8000/v1>, bia-bob version: 0.25.3.. Do not enter sensitive or private information and verify generated contents according to good scientific practice. Read more: <https://github.com/haesleinhuepf/bia-bob#disclaimer>

```
[2]: %bob load blobs.tif and show it.
```

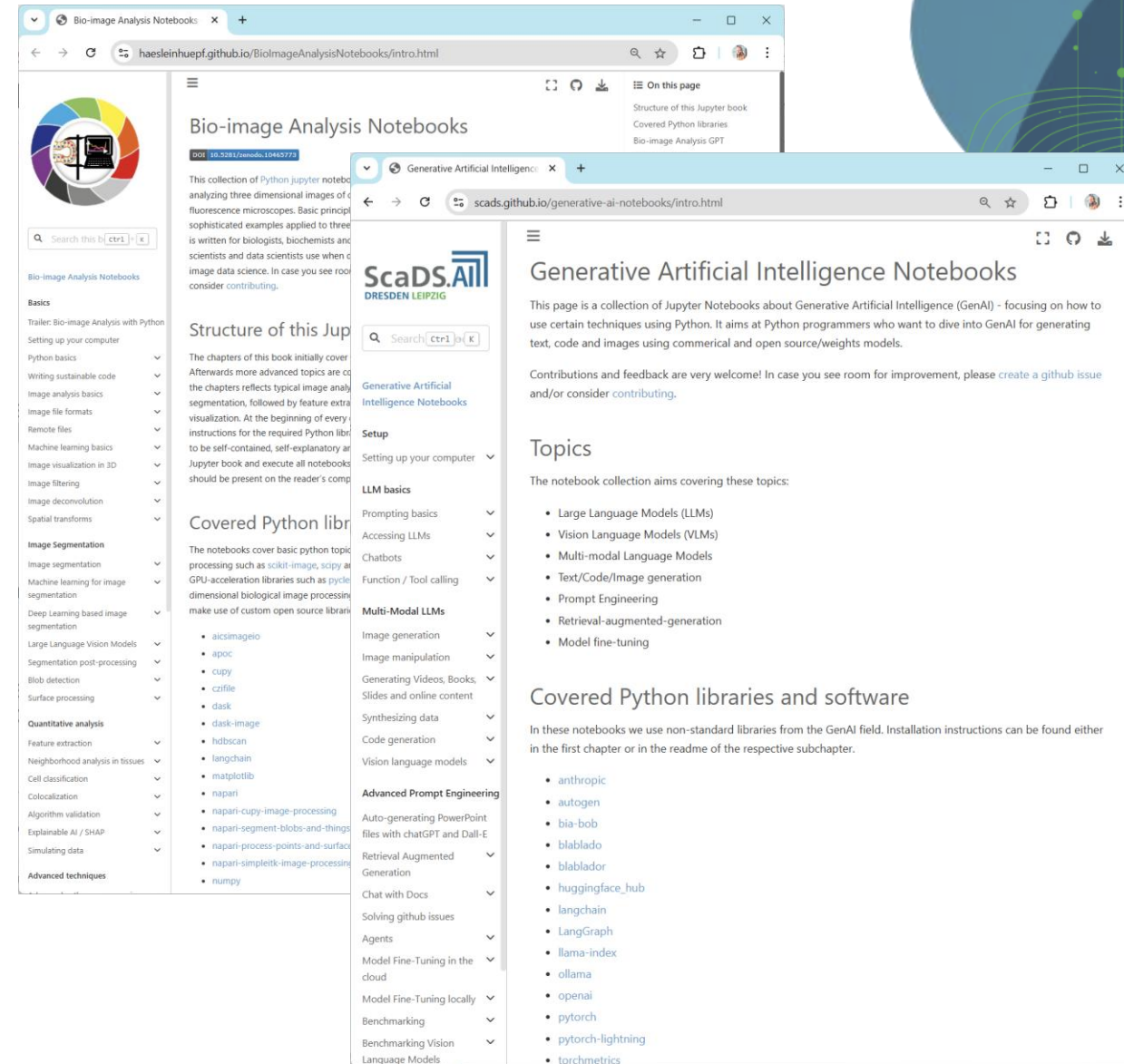
```
[ ]: from skimage.io import imread
      import stackview

      # Load the image
      image = imread("blobs.tif")

      # Display the image
      stackview.insight(image)
```


Summary

- Using AI / LLMs in [bioimage] data analysis seems more and more unavoidable
- Maintaining good scientific practice is challenging (esp. peer-review)
- Relying on institutional infrastructure exclusively is important but hard
- Learn how to use AI *responsibly* (-> train the trainers)
- We are on the edge towards AI-agents analyzing image data autonomously.



Acknowledgements

Collaborators & contributors

Christian Tischer (EMBL Heidelberg), Jean-Karim Hériché (EMBL Heidelberg), Nico Scherf (MPI CBS Leipzig), Loïc A. Royer (CZ Biohub), Elena Nicolay (UFZ Leipzig), Kevin Yamauchi (ETH Zurich / Basel), Seth Hinz, Teun A.P.M. Huijben, Jordão Bragantini (CZ Biohub San Francisco), Ilan Theodoro (CZ Biohub and the State University of Campinas, Brazil), Ian C. (University of British Columbia), Pradeep Rajasekhar (WEHI Australia)

Communities & platforms



NFDI 4
BIOIMAGE



image.sc

BIDS Team @ ScaDS.AI / UL



Lea Kabjesz, Lea Gihlein, Mara Lampert



<https://doi.org/10.5281/zenodo.15735577>

Funding



Bundesministerium
für Forschung, Technologie
und Raumfahrt



SACHSEN 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.



Deutsche
Forschungsgemeinschaft



LLMs Bio-Image Analysis
Helmholtz Imaging Conf.
Robert Haase
@haesleinhuepf
June 25th 2025

RH acknowledges the financial support by the Federal Ministry of Education and Research of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Center of Excellence for AI-research „Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig“, project identification number: ScaDS.AI



TECHNISCHE
UNIVERSITÄT
DRESDEN



UNIVERSITÄT
LEIPZIG