Fractal:
An open-source framework for reproducible bioimage analysis at scale using OME-Zarrs

29/11/2024

Joel Lüthi (BioVisionCenter, University of Zurich)

Goals of today's seminar

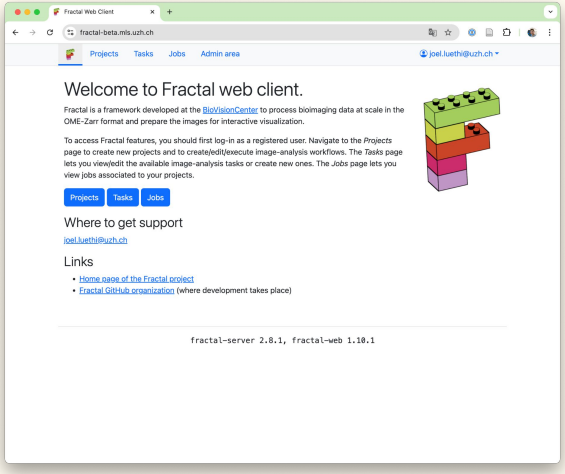
1. What is an OME-Zarr?

2. Introduction to Fractal

3. Fractal Demo

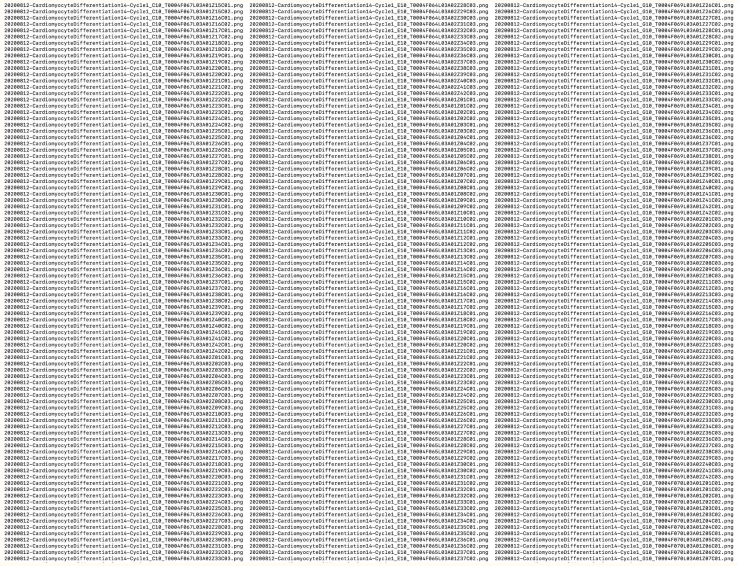
OME Zarr

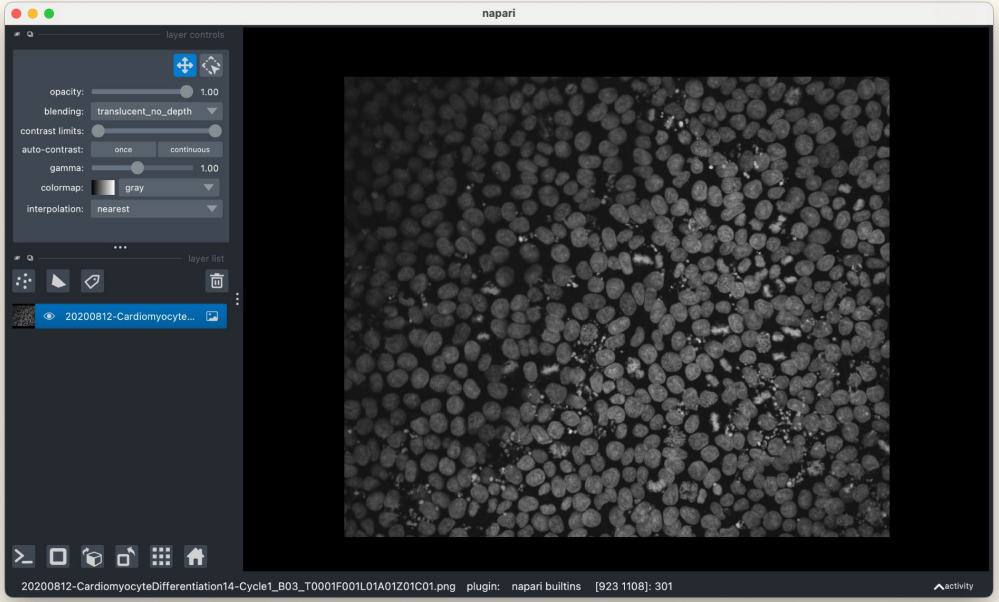
Fractal



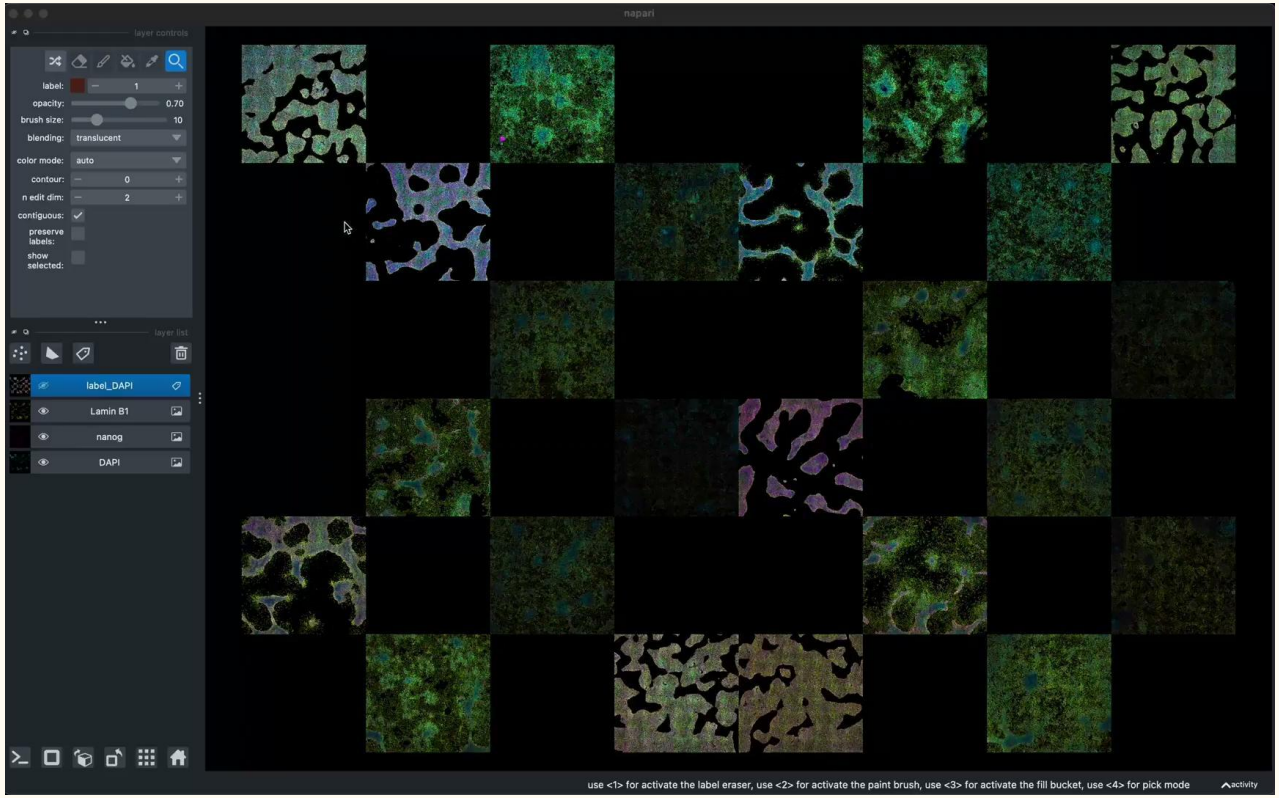
Modern microscopes produce a lot of data!

What do you do with 163’518 images?





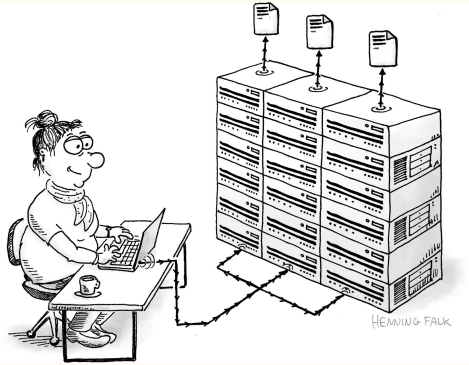
We need ways to interactively explore large datasets



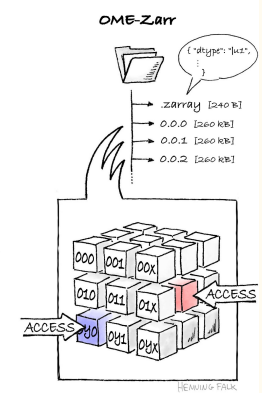
How should we store large bioimage data?

Multi-resolution data

Cloud & HPC compatible



Support for 5D

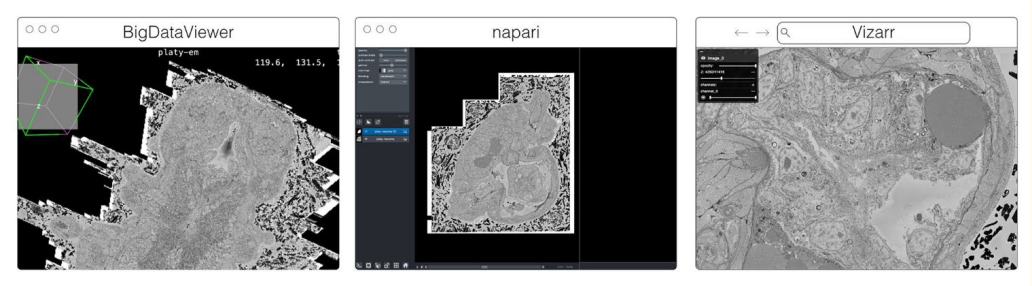


=> Let's design our new custom file format!

Let's **NOT** design our new custom file format!
=> Let's use OME-Zarr

OME

Zarr



OME-Zarr: a cloud-optimized bioimaging file format with international community support

Josh Moore, Daniela Basurto-Lozada, Sébastien Besson, John Bogovic, Jordão Bragantini, Eva M. Brown, Jean-Marie Burel, Xavier Casas Moreno, Gustavo de Medeiros, Erin E. Diel, David Gault, Satrajit S. Ghosh, Ilan Gold, Yaroslav O. Halchenko, Matthew Hartley, Dave Horsfall, Mark S. Keller, Mark Kittisopikul, Gabor Kovacs, Aybüke Küpcü Yoldaş, Koji Kyoda, Albane le Tournoulx de la Villegeorges, Tong Li, Prisca Liberali, Dominik Lindner, Melissa Linkert, Joel Lüthi, Jeremy Maitin-Shepard, Trevor Manz, Luca Marconato, Matthew McCormick, Merlin Lange, Khaled Mohamed, William Moore, Nils Norlin, Wei Ouyang, Bugra Özdemir, Giovanni Palla, Constantin Pape, Lucas Pelkmans, Tobias Pietzsch, Stephan Preibisch, Martin Prete, Norman Rzepka, Sameeul Samee, Nicholas Schaub, Hythem Sidky, Ahmet Can Solak, David R. Stirling, Jonathan Striebel, Christian Tischer, Daniel Toloudis, Isaac Virshup, Petr Walczysko, Alan M. Watson, Erin Weisbart, Frances Wong, Kevin A. Yamauchi, Omer Bayraktar, Beth A. Cimini, Nils Gehlenborg, Muzlifah Haniffa, Nathan Hotaling, Shuichi Onami, Loic A. Royer, Stephan Saalfeld, Oliver Stegle, Fabian J. Theis, Jason R. Swedlow

doi: https://doi.org/10.1101/2023.02.17.528834

Visualization tool	Use	Language/framework
AGAVE	Linux, MacOS, Windows	C++, OpenGL
ITKWidgets	Web (Jupyter)	Python, WASM
MoBIE/BigDataViewer	Linux, MacOS, Windows	Java
napari	Desktop	Python
Neuroglancer	Web	WebGL
Validator	Web	Svelte
Viv	Web	React, deck.gl
webKnossos	Web	React, WebGL
website-3d-cell-viewer	Web	React, TypeScript, WebGL

An up-to-date version of the table is maintained at https://ngff.openmicroscopy.org/tools and contributions are welcome

Moore et al., 2023

OME-Zarr: The next generation file format for bioimage data => compatible with large scale data & cloud storage

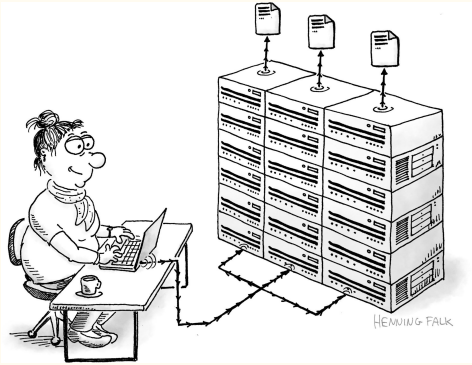
OME-Zarr: a cloud-optimized bioimaging file format with international community support

Josh Moore, Daniela Basurto-Lozada, Sébastien Besson, John Bogovic, Jordão Bragantini, Eva M. Brown, Jean-Marie Burel, Xavier Casas Moreno, Gustavo de Medeiros, Erin E. Diel, David Gault, Satrajit S. Ghosh, Ilan Gold, Yaroslav O. Halchenko, Matthew Hartley, Dave Horsfall, Mark S. Keller, Mark Kittisopikul, Gabor Kovacs, Aybüke Küpcü Yoldaş, Koji Kyoda, Albane le Tournoulx de la Villegeorges, Tong Li, Prisca Liberali, Dominik Lindner, Melissa Linkert, Joel Lüthi, Jeremy Maitin-Shepard, Trevor Manz, Luca Marconato, Matthew McCormick, Merlin Lange, Khaled Mohamed, William Moore, Nils Norlin, Wei Ouyang, Bugra Özdemir, Giovanni Palla, Constantin Pape, Lucas Pelkmans, Tobias Pietzsch, Stephan Preibisch, Martin Prete, Norman Rzepka, Sameeul Samee, Nicholas Schaub, Hythem Sidky, Ahmet Can Solak, David R. Stirling, Jonathan Striebel, Christian Tischer, Daniel Toloudis, Isaac Virshup, Petr Walczysko, Alan M. Watson, Erin Weisbart, Frances Wong, Kevin A. Yamauchi, Omer Bayraktar, Beth A. Cimini, Nils Gehlenborg, Muzlifah Haniffa, Nathan Hotaling, Shuichi Onami, Loic A. Royer, Stephan Saalfeld, Oliver Stegle, Fabian J. Theis, Jason R. Swedlow

doi: https://doi.org/10.1101/2023.02.17.528834

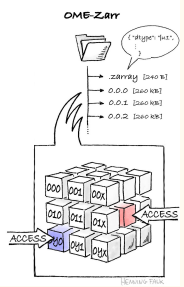
Large community effort

Multi-resolution data



Cloud & HPC compatible

Chunked format with 5D support



How do we run
reproducible image analysis
at scale
using **OME-Zarrs**?

Fractal offers a framework for FAIR image analysis using OME-Zarrs

Fractal

The Fractal Framework

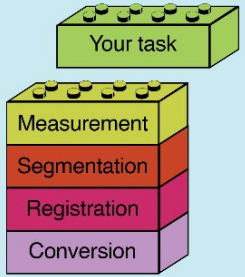
https://fractal-analytics-platform.github.io

Process TBs of images as OME-Zarrs

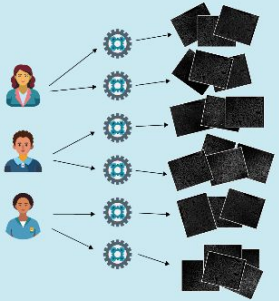
OME Zarr

- Next-generation file format for bioimage data
- Cloud-ready community standard for 5D images

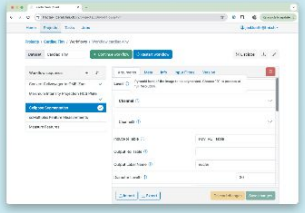
Extensible & user-definable workflows



Scalable workflow execution



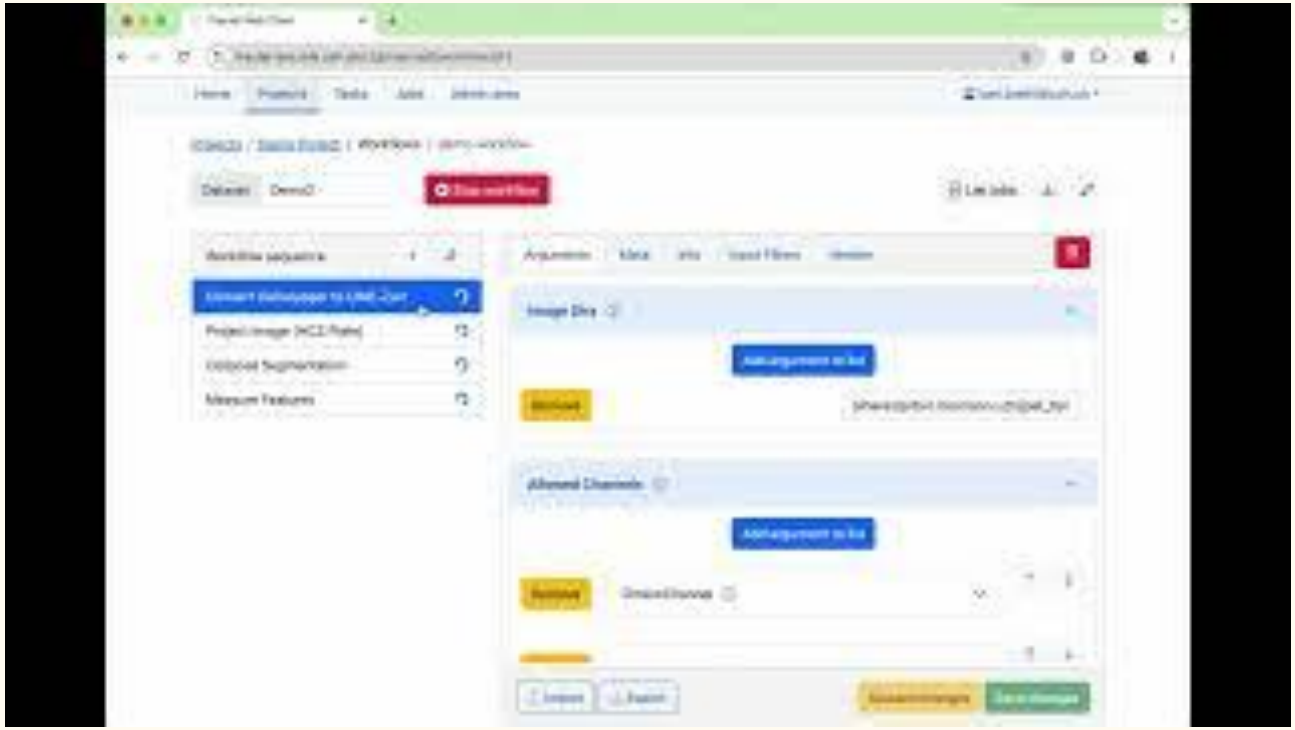
Web-based workflow management



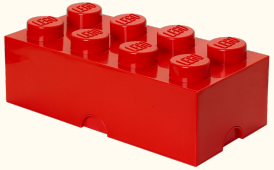
Interactive visualization

VIZARR

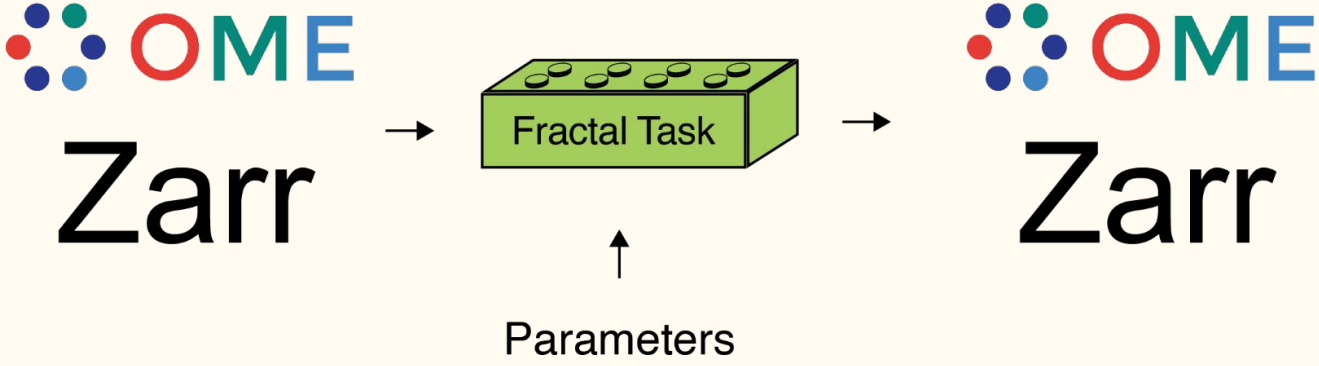
Control your workflows via a web interface



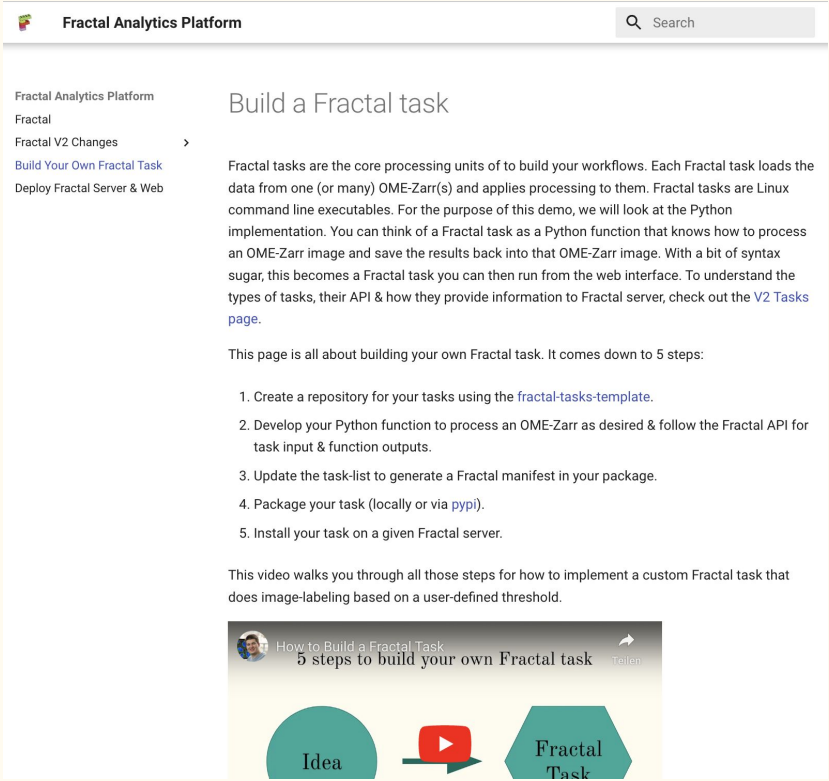
Fractal: Build modular, user-designed workflows



Your task?



How to build your own Fractal task?



Fractal Analytics Platform

Search

Fractal Analytics Platform
Fractal
Fractal V2 Changes
Build Your Own Fractal Task
Deploy Fractal Server & Web

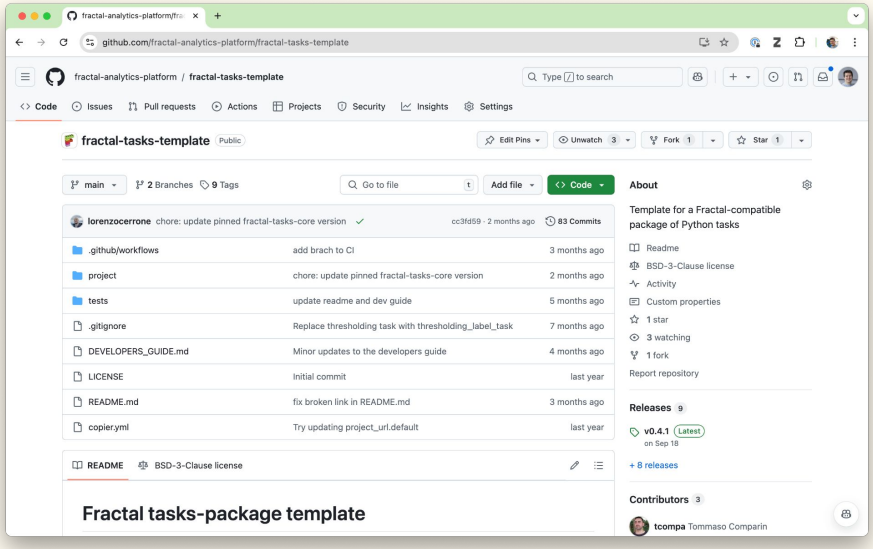
Build a Fractal task

Fractal tasks are the core processing units of to build your workflows. Each Fractal task loads the data from one (or many) OME-Zarr(s) and applies processing to them. Fractal tasks are Linux command line executables. For the purpose of this demo, we will look at the Python implementation. You can think of a Fractal task as a Python function that knows how to process an OME-Zarr image and save the results back into that OME-Zarr image. With a bit of syntax sugar, this becomes a Fractal task you can then run from the web interface. To understand the types of tasks, their API & how they provide information to Fractal server, check out the V2 Tasks page.

This page is all about building your own Fractal task. It comes down to 5 steps:

1. Create a repository for your tasks using the fractal-tasks-template.
2. Develop your Python function to process an OME-Zarr as desired & follow the Fractal API for task input & function outputs.
3. Update the task-list to generate a Fractal manifest in your package.
4. Package your task (locally or via pypi).
5. Install your task on a given Fractal server.

This video walks you through all those steps for how to implement a custom Fractal task that does image-labeling based on a user-defined threshold.



github.com/fractal-analytics-platform/fractal-tasks-template

fractal-analytics-platform / fractal-tasks-template

Code | Issues | Pull requests | Actions | Projects | Security | Insights | Settings

fractal-tasks-template Public

main | 2 Branches | 9 Tags | Go to file | Add file | Code

lorenzocerrone chore: update pinned fractal-tasks-core version — cc3fd59 · 2 months ago — 83 Commits

.github/workflows	add brach to CI	3 months ago
project	chore: update pinned fractal-tasks-core version	2 months ago
tests	update readme and dev guide	5 months ago
.gitignore	Replace thresholding task with thresholding_label_task	7 months ago
DEVELOPERS_GUIDE.md	Minor updates to the developers guide	4 months ago
LICENSE	Initial commit	last year
README.md	fix broken link in README.md	3 months ago
copier.yml	Try updating project_url.default	last year

README | BSD-3-Clause license

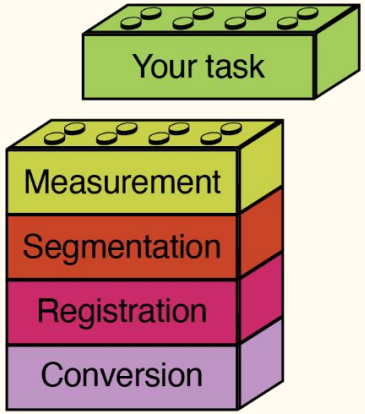
Fractal tasks-package template

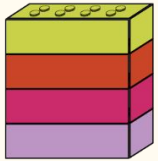
About
Template for a Fractal-compatible package of Python tasks
Readme
BSD-3-Clause license
Activity
Custom properties
1 star
3 watching
1 fork
Report repository

Releases 9
v0.4.1 Latest
on Sep 18
+ 8 releases

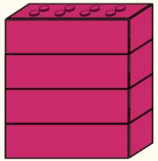
Contributors 3
tcompa Tommaso Comparin

Fractal has a growing community of task developers & task packages

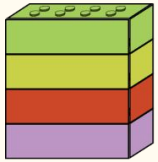




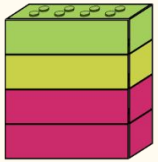
Fractal tasks core



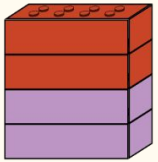
Fractal multi-view stitcher



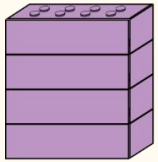
Operetta com-pose



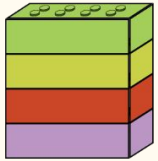
scMultipleX



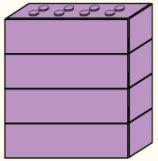
Fractal plant-seg tasks



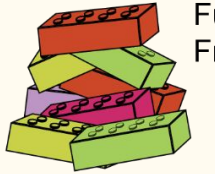
Fractal lif converters



Apricot task collection

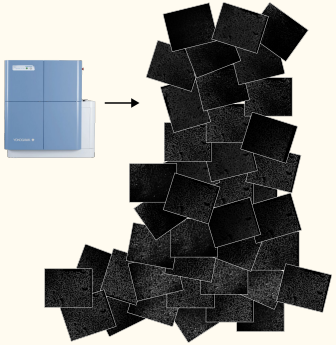


Fractal faim ipa



Full list of Fractal tasks

Fractal enables complex workflows



Yokogawa
Cellvoyager
3D Images

OME
Zarr

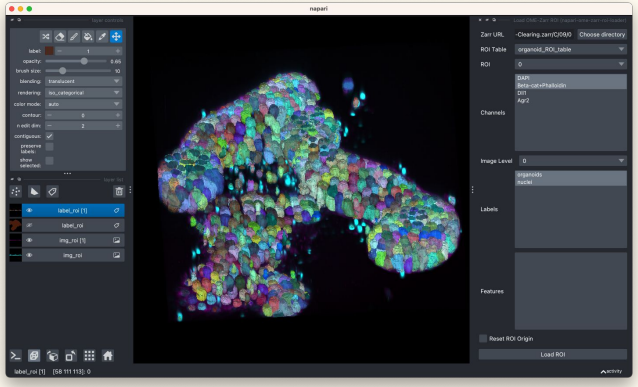
3D
OME-Zarr

MIP

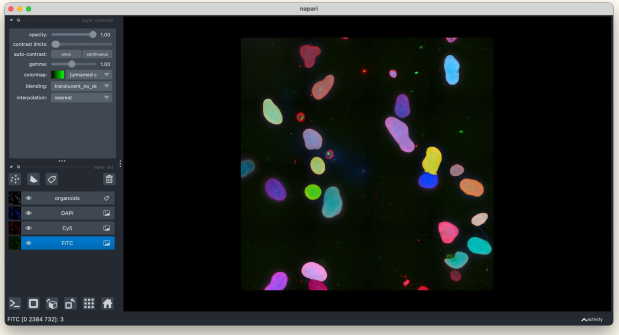
OME
Zarr

2D
OME-Zarr

3D nuclear segmentation



3D
measurements



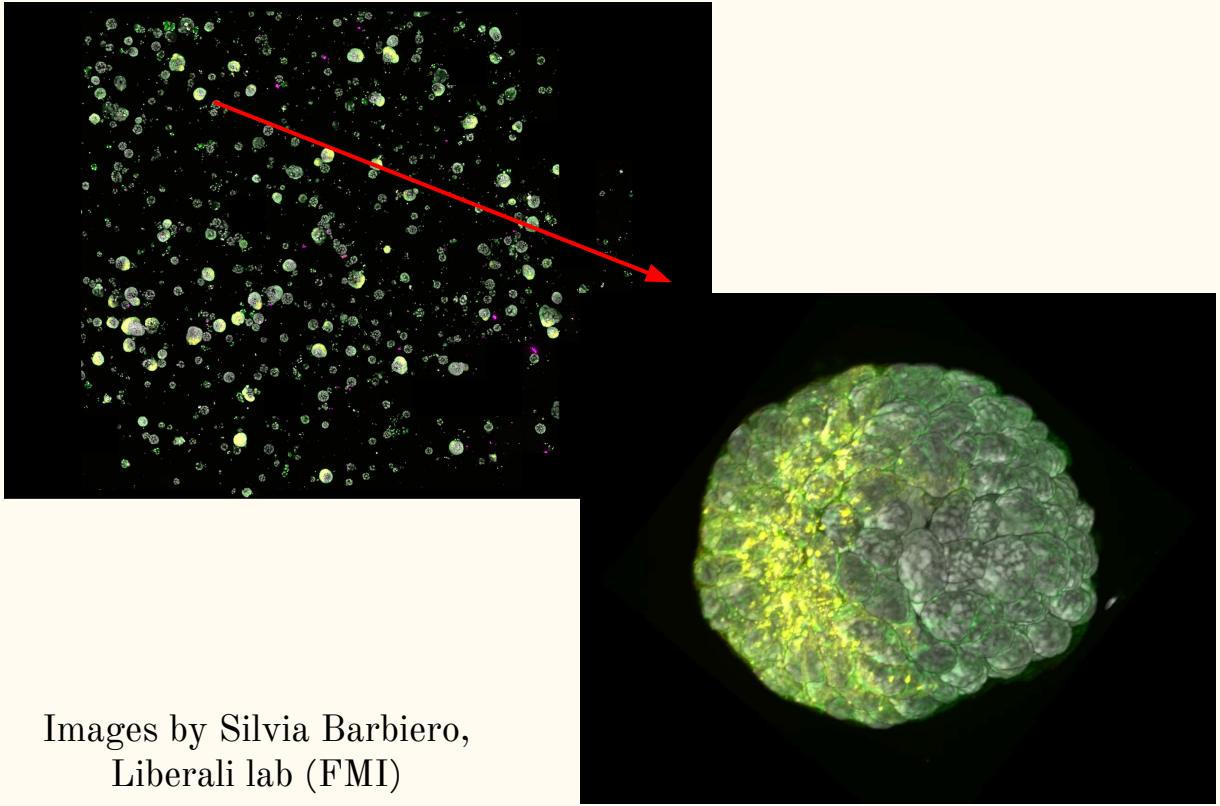
2D organoid segmentation

	area	mean_intensity	standard_deviation_intensity
label			
1	2400.0	264.030426	71.719330
2	528.0	251.848480	121.864410
3	1536.0	200.345047	62.951511
4	5856.0	305.467712	95.109253
5	2768.0	241.529617	68.545265
...	...	...	...

2D
measurements

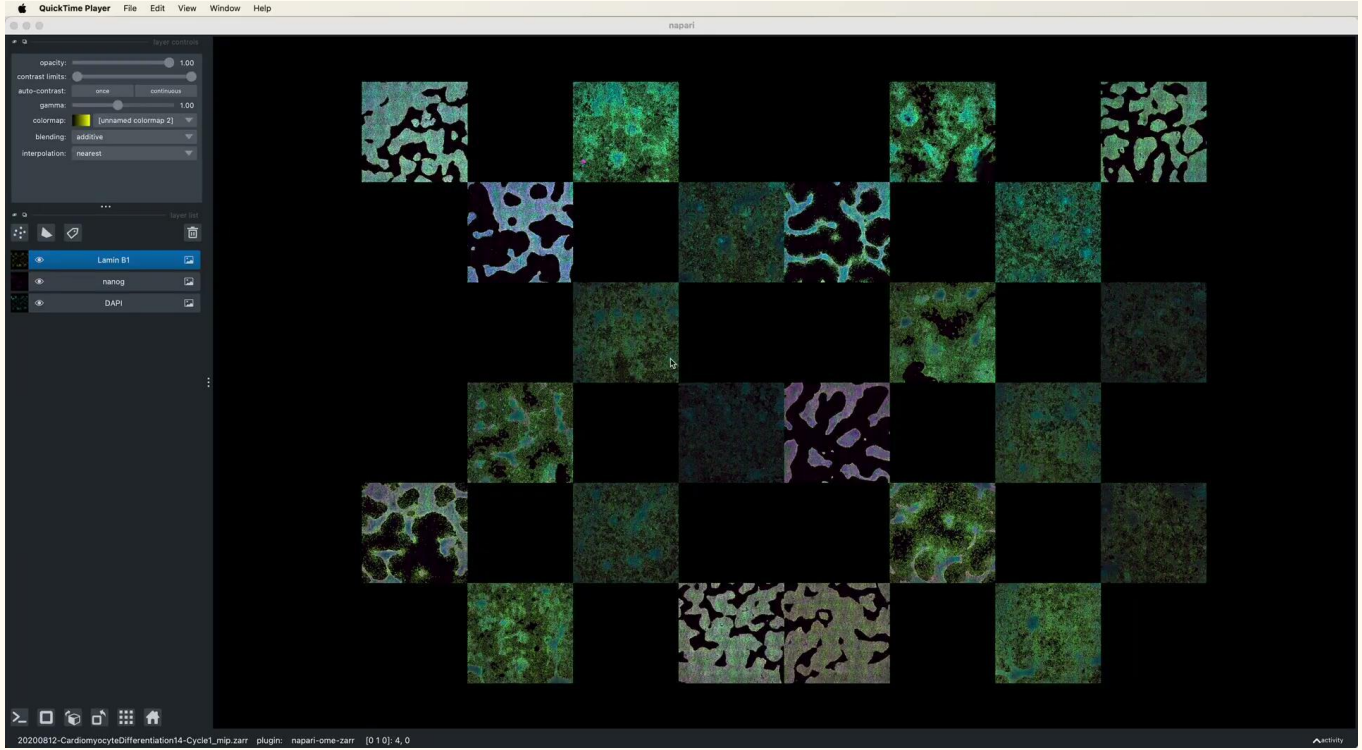
Benefits of processing OME-Zarrs

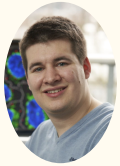
- Easily process images at optimized resolutions
- Iterate over arbitrary regions of huge 3D image datasets
- Use a standard format across microscopes => build interoperable processing units



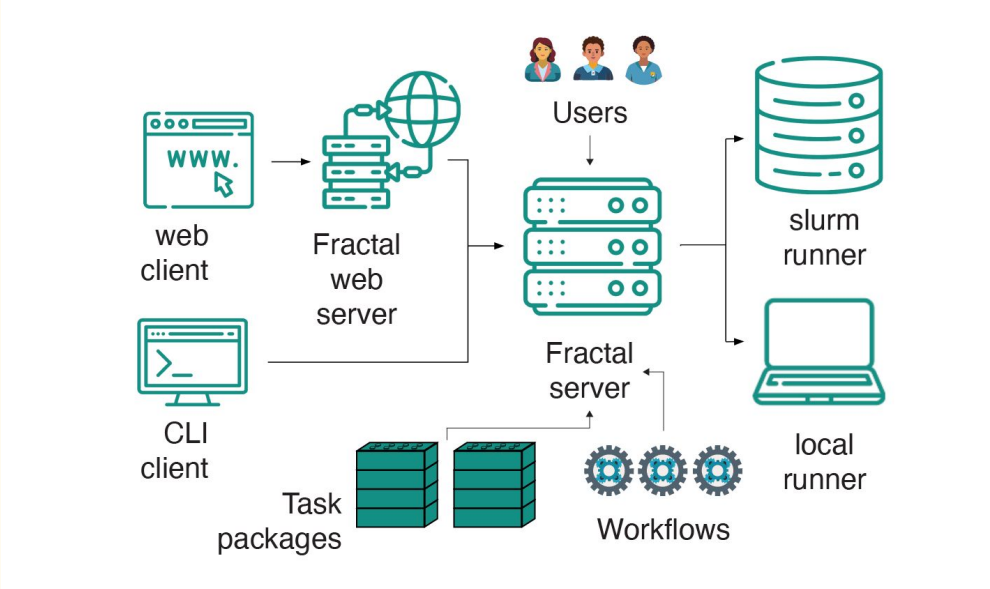
Images by Silvia Barbiero, Liberali lab (FMI)

Fractal-processed OME-Zarrs contain images, segmentation and measurements => allow for integrated analysis like classification

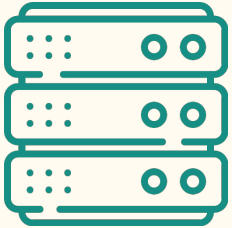




Fractal runs in federated deployments



UZH Fractal Server

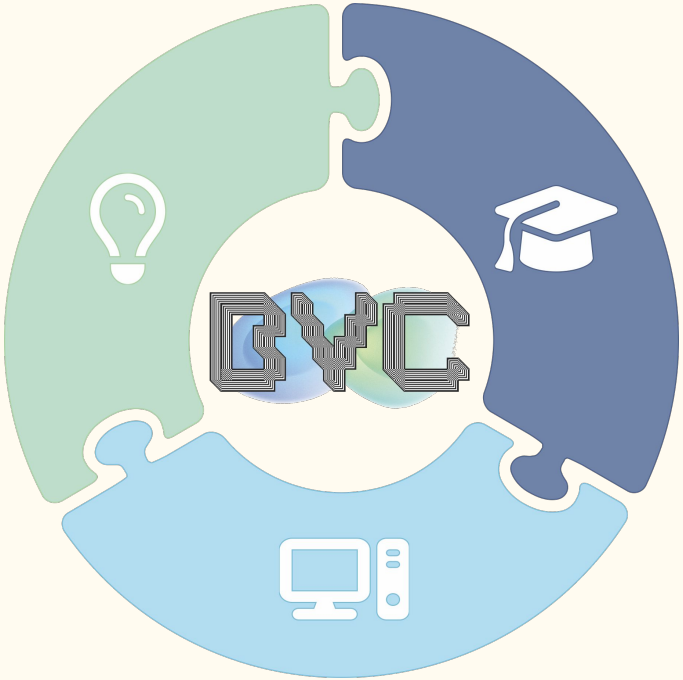


FMI Fractal Server

Your Fractal Server

Demo time:
The UZH Fractal beta server
https://fractal-beta.mls.uzh.ch/

Fractal is developed at the BioVisionCenter



www.biovisioncenter.uzh.ch

Our mission

Make state-of-the-art bioimage analysis at scale **accessible to all**

Champion standards for sharing and reproducing bioimage analysis pipelines

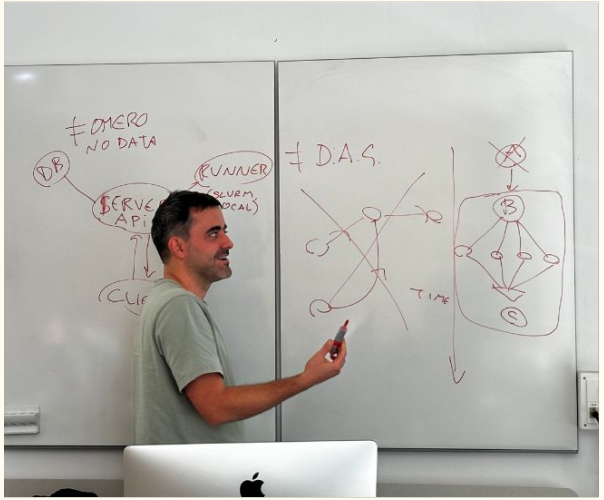
FMI Friedrich Miescher Institute for Biomedical Research

Universität Zürich[UZH]

The BioVisionCenter: a hub for bioimage analysis

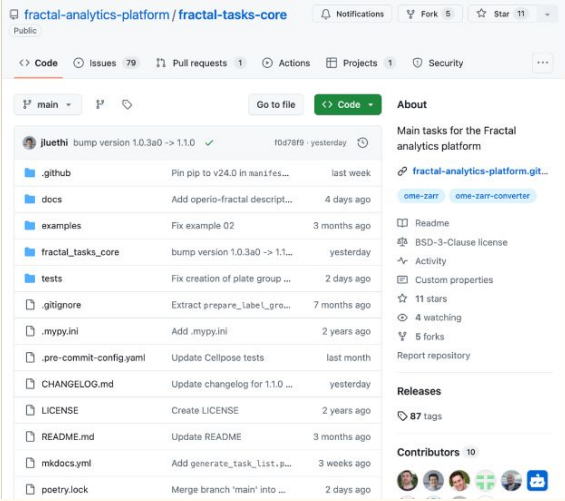
Cutting-edge expertise

We keep up with the fast-paced development of ML applied to computer vision



Excellence

We develop methods at the highest professional software standards



Partnership

We carry out advanced development work in specific areas of interest with our partners

Fractal is under active development

- We're developing Fractal in the open
- All our packages come with a permissive BSD3 open-source license
- Fractal runs in a federated fashion:
  - Every institution runs their own Fractal server
  - The BioVisionCenter partners with institutions to support their deployment & usage of Fractal

Component	GitHub Repository	Documentation	Package
server	fractal-server	fractal-server docs	fractal-server on PyPI
client	fractal-client	fractal-client docs	fractal-client on PyPI
web client	fractal-web	fractal-web docs	-
core tasks	fractal-tasks-core	fractal-tasks-core docs	fractal-tasks-core on PyPI

https://fractal-analytics-platform.github.io

Fractal timeline

Start of Fractal development

Fractal 1.0: Stable HCS processing

Fractal moves to the BioVisionCenter

2022

2023

Fractal 2.0: Beyond HCS & flexibility

Fractal UZH Beta server

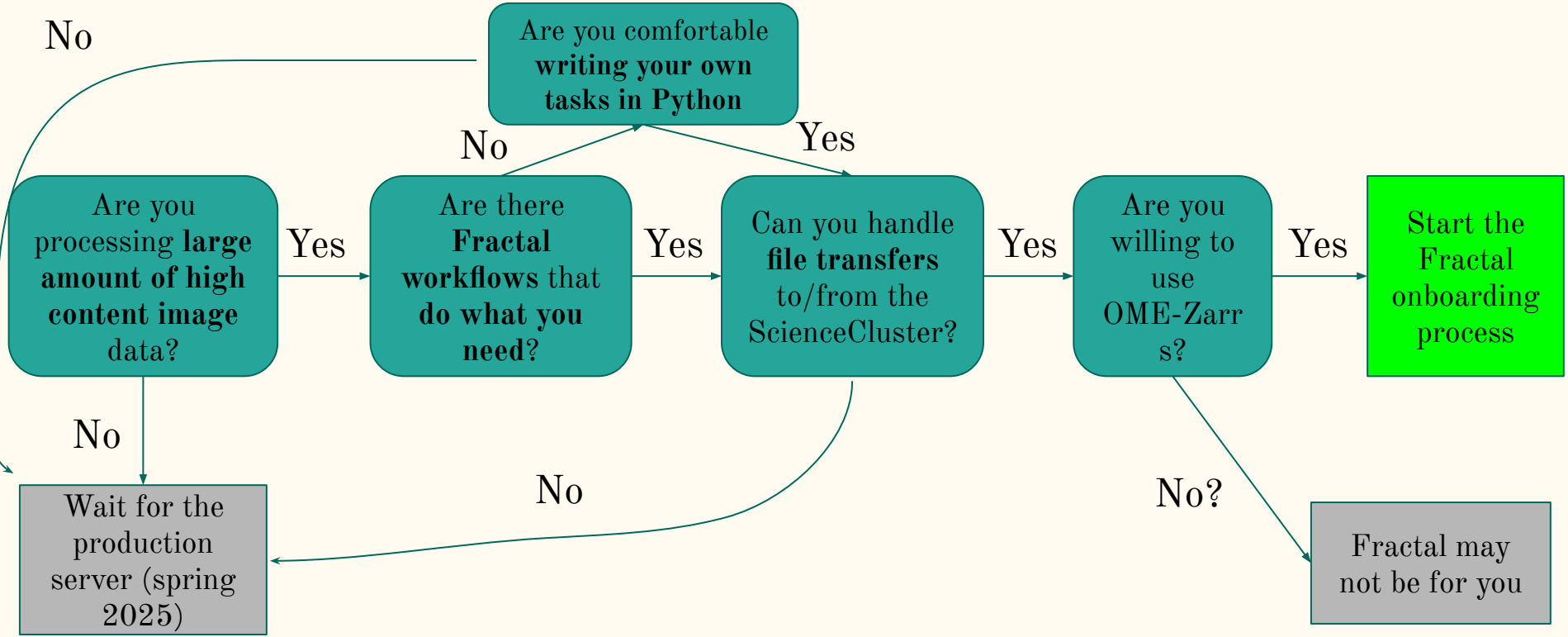
Fractal UZH production server

2024

2025

- Easier data transfers
- More converters for ZMB microscopes
- Viewer improvements
- Better task building tools (ngio)

Should **you** use the Fractal beta server?



Additions coming in the production mode in spring 2025

- Easier data transfers
- Viewer improvements
- More converters for ZMB microscopes & more processing tasks
- Better tooling to build your own tasks
- Collaborate with the BioVisionCenter to build tasks for/with you

Acknowledgements

https://fractal-analytics-platform.github.io

Universität Zürich[UZH]

FMI
Friedrich Miescher Institute for Biomedical Research

Development support: exact lab Trieste, Italy

Early Fractal support: liberalilab.org PELKMANSLAB CHAN ZUCKERBERG INITIATIVE