BioImage-IT

Integration of data-management with analysis

Sylvain Prigent, Ludovic Leconte,
César Augusto Valades Cruz, Léo Maury, Charles Kervrann, Jean Salamero

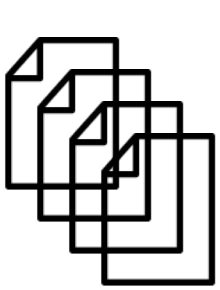
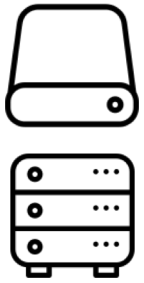
STED/Serpico team (Curie/Inria)

FRANCE-BIOIMAGING

Why BioImage-IT ?

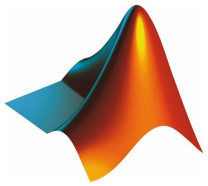
Data management and analysis is tedious…

Problem 1: Data management
- Hard drives: storage without metadata
- Servers: storage without metadata
- data traceability ?

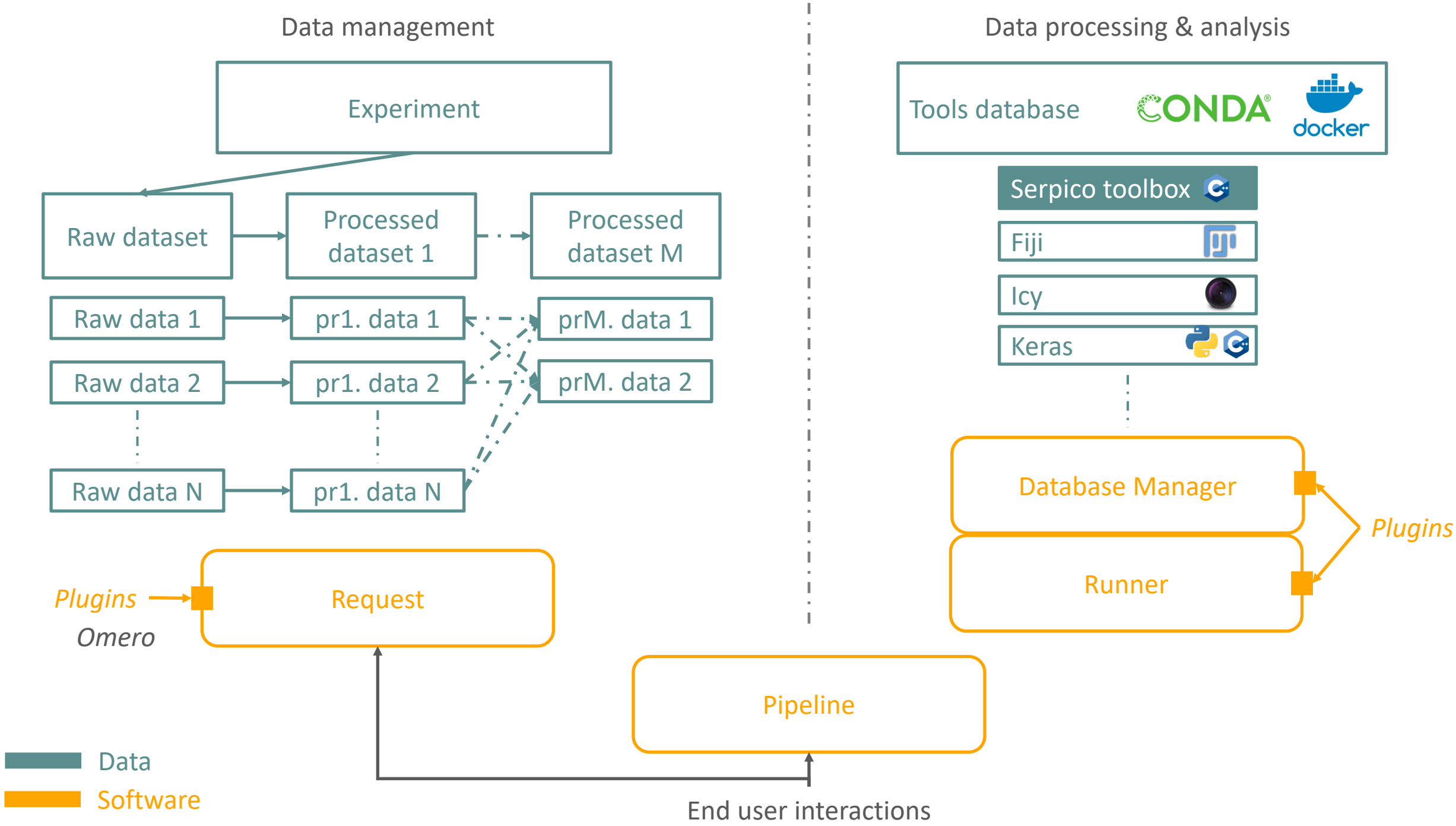

Problem 2: Processing & visualization tools
- Many languages
- For different systems (Linux, Apple, Microsoft)
- Documentations & test dataset ?



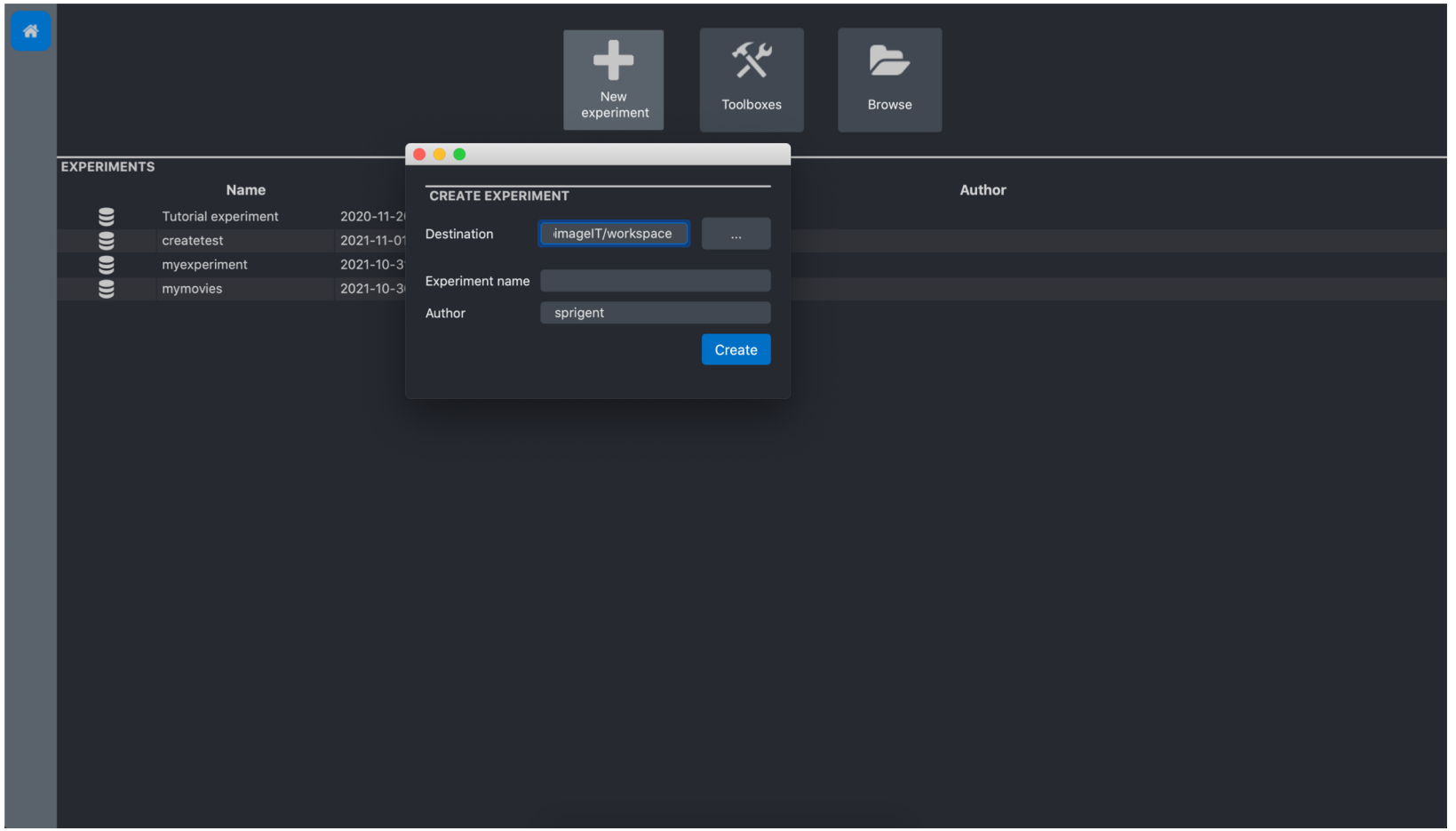
Problem 3: link data to processing & visualization
- Manual scripting
- Systems configurations (ex: send data to cluster)

What is BioImage-IT ?


Data management
Data processing & analysis
Experiment
Tools database
CONDA
docker
Serpico toolbox
Fiji
Icy
Keras
Raw dataset
Processed dataset 1
Processed dataset M
Raw data 1
pr1. data 1
prM. data 1
Raw data 2
pr1. data 2
prM. data 2
Raw data N
pr1. data N
Database Manager
Runner
Plugins
Plugins
Omero
Request
Pipeline
Data
Software
End user interactions

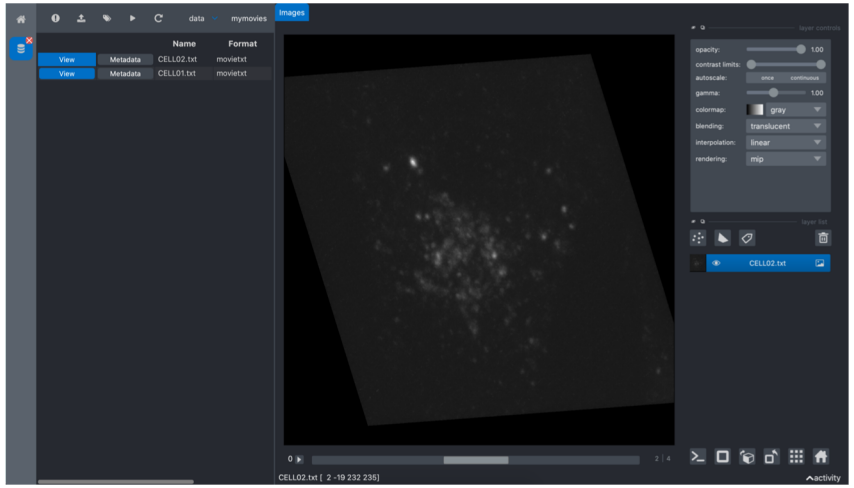
User interface

Home

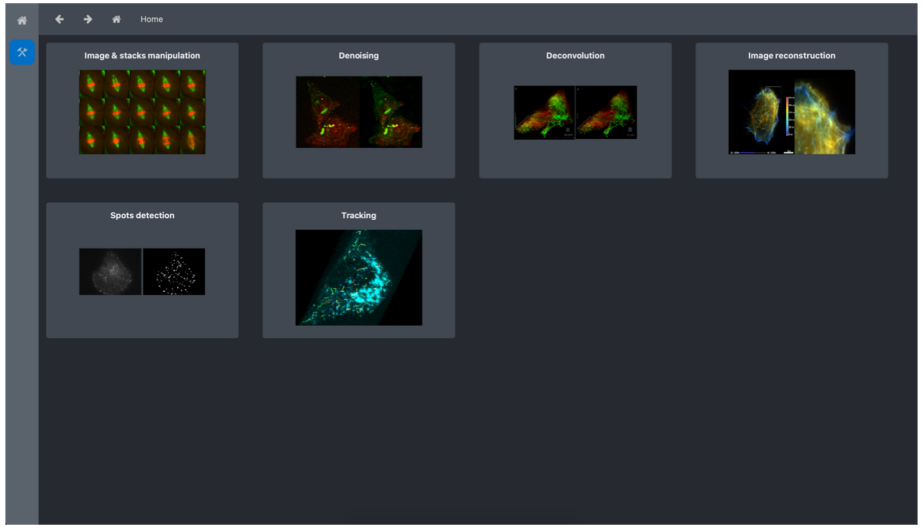


User interface

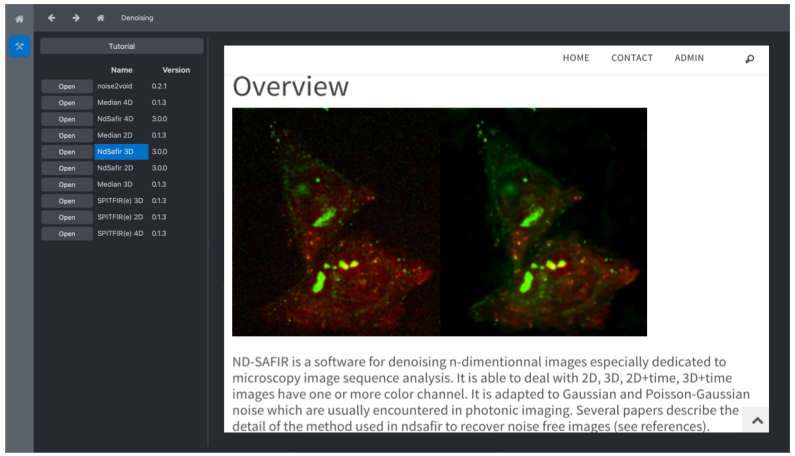
Experiment editor



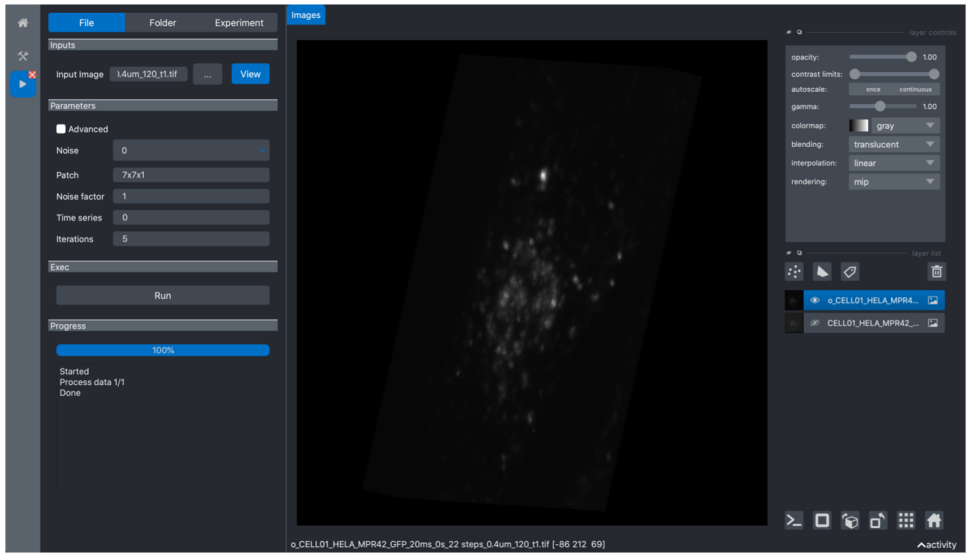
Toolboxes finder



Tool documentation



Tool runner



Developers tools

Interact with the python API

```python
from bioimagepy.experiment import Experiment

## create an experiment
my_experiment = Experiment()
my_experiment.create(name="myexperiment",
                     author="Sylvain Prigent",
                     uri="./")

print("experiment created")
```

```python
# import the data
my_experiment.import_dir(dir_uri='./synthetic_data/data/',
                         filter="\.tif$",
                         author='Sylvain Prigent',
                         format='tif',
                         date='2019-03-17',
                         copy_data=True)

# show the experiment directory
list_files('./myexperiment')
```

Pipeline

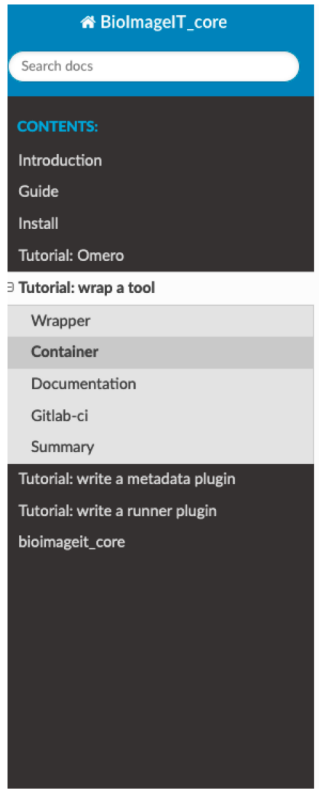
```python
from bioimagepy.experiment import Experiment
from bioimagepy.pipeline import Pipeline

ConfigAccess('../bioimagepy/config_sample.json')
my_experiment = Experiment('../userdata/myexperiment')
pipeline = Pipeline(my_experiment)
```

```python
process1 = pipeline.add_process(ProcessAccess().get('svdeconv2d_v0.1.0'))
process1.set_parameters('sigma', 2, 'regularization', 2, 'weighting', 0.1, 'method', 'SV')
process1.set_dataset_name('deconv')
process1.add_input('i', 'data')
process1.run()
```

```python
process2 = pipeline.add_process(threshold_particles)
process2.set_parameters('threshold', 'Default dark')
process2.set_dataset_name('particles')
process2.add_input('input', 'deconv')
process2.run()
```

```python
process3 = pipeline.add_process(ProcessAccess().get('Wilcoxon_v1.0.0'))
process3.set_dataset_name('wilcoxon')
process3.add_input('x', 'particles', 'Population=population1', 'count')
process3.add_input('y', 'particles', 'Population=population2', 'count')
process3.run()
```

Integrate your own tools



The tool wrapper is in the file called `sampletool.xml`. Its content is shown bellow. It is a classical Galaxy Project wrapper. The command is `meanfilter` and take three arguments, the input image `-i`, the output image `-o` and one parameter `-r` the radius of the filter. The two important information in this wrapper are the `<requirements>` section that contains the address of the *Docker* image container that we will be generated with the *gitlab-ci*, and the address of the documentation in `<help>` that will be generated by the *gitlab-ci* pages.

```xml
<tool id="sampletool" name="SampleTool" version="0.1.0" python_template_version="3.5">
    <requirements>
        <container type="docker">registry.gitlab.inria.fr/
        bioimage-it/sampletool:766efe8b8397fef0115e20f...
        </container>
    </requirements>
    <command detect_errors="exit_code"><![CDATA[
        meanfilter -i ${i} -o ${o} -r ${r}
    ]]></command>
    <inputs>
        <param type="data" name="i" format="tiff" label="Input Image" help="2D image" />
        <param argument="-r" type="integer" value="3" label="Radius" help="Filter radius (in p
    </inputs>
    <outputs>
        <data name="o" format="tiff" label="Output image" />
    </outputs>
    <tests>
        <test>
            <param name="i" value="celegans.tif" />
            <param name="r" value="3" />
            <output name="o" file="celegans_result.tif"
            compare="sim_size" />
        </test>
    </tests>
    <help><![CDATA[
        https://bioimage-it.gitlabpages.inria.fr/sampletool/
    ]]></help>
    <citations>
    </citations>
</tool>
```

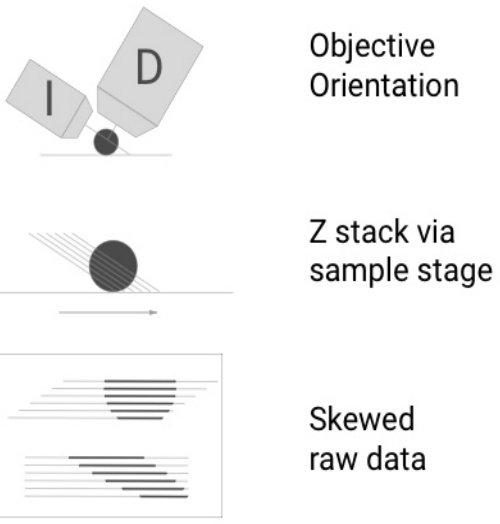
Use case:
Lattice light sheet image analysis pipeline

LLSM original pipeline

Application: 3D+t tracking of endosomes using Lattice Lightsheet microscopy

Pipeline: needs at least 3 different software

LLSM acquisition



Deskew pycudadecon → Deconvolution pycudadecon → Tracking Trackmate → Visualization Napari

https://llspy.readthedocs.io/en/latest/main.html

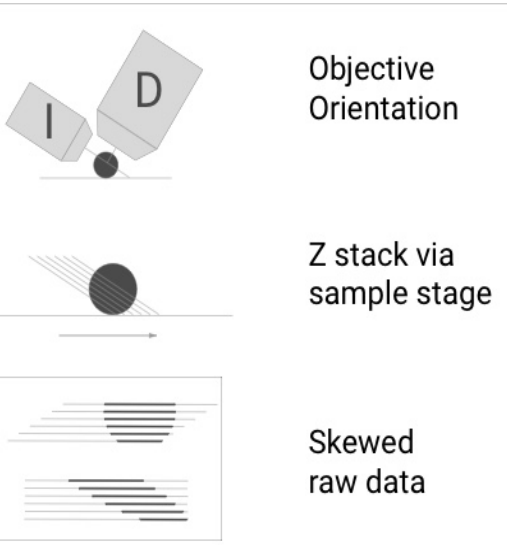
*Tinevez, JY.; Perry, N. & Schindelin, J. et al. (2017), "**TrackMate: An open and extensible platform for single-particle tracking.**", Methods **115**: 80-90, PMID 27713081*

*napari contributors (2019). **napari: a multi-dimensional image viewer for python**. doi:10.5281/zenodo.3555620*

LLSM BioImage-IT pipeline

Pipeline: with BioImage-IT -> interact with a single interface

LLSM acquisition

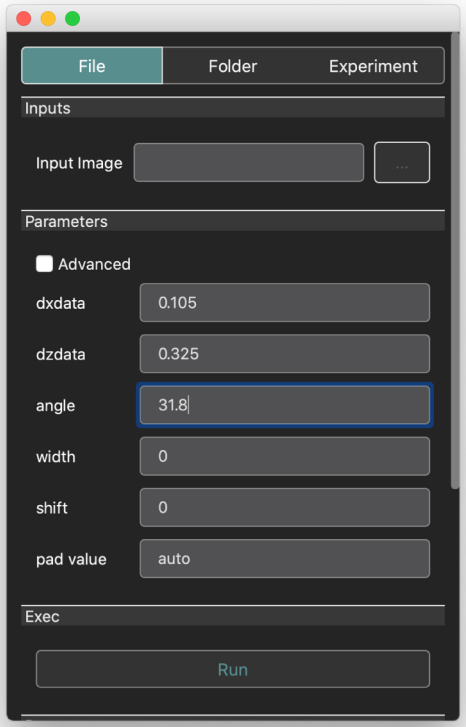


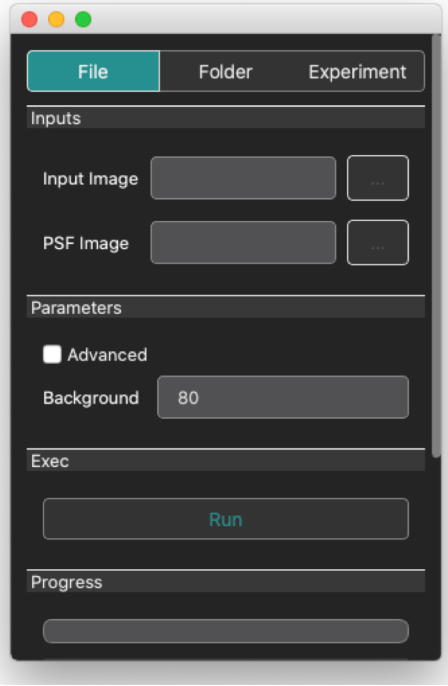
Deskew
pycudadecon

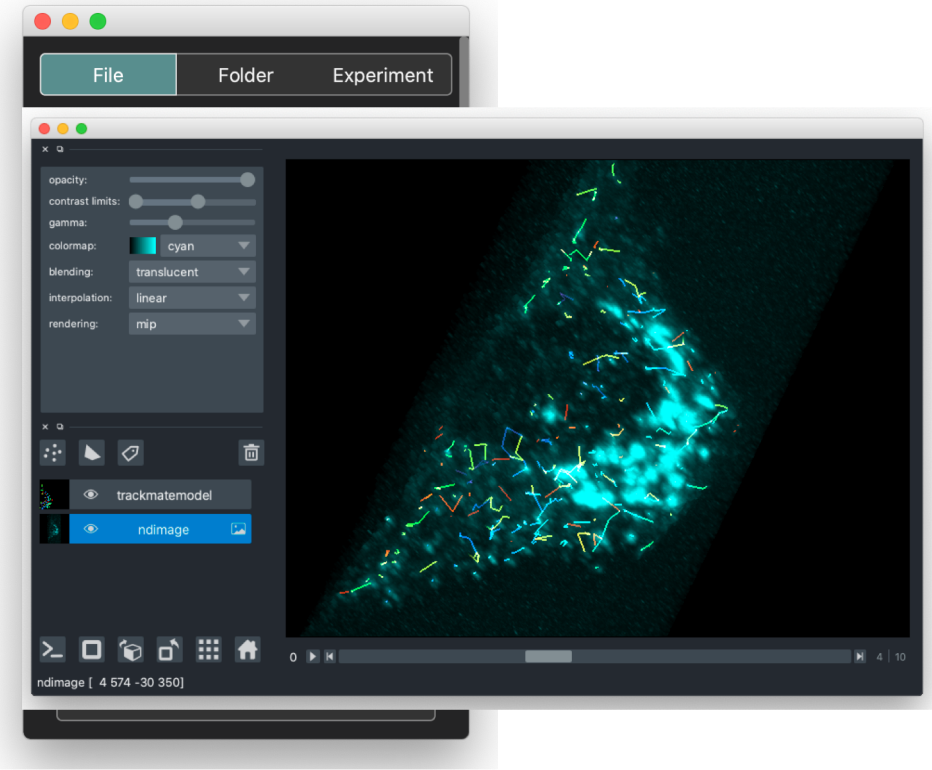
Deconvolution
pycudadecon

Tracking
Trackmate

Visualization
Napari





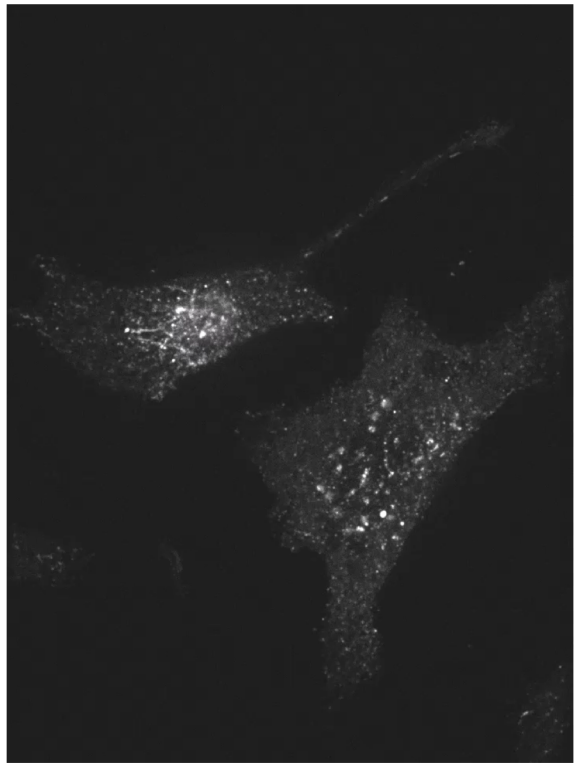


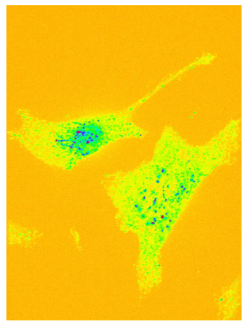
Sted/Serpico toolbox

Understood the complexity of the spots

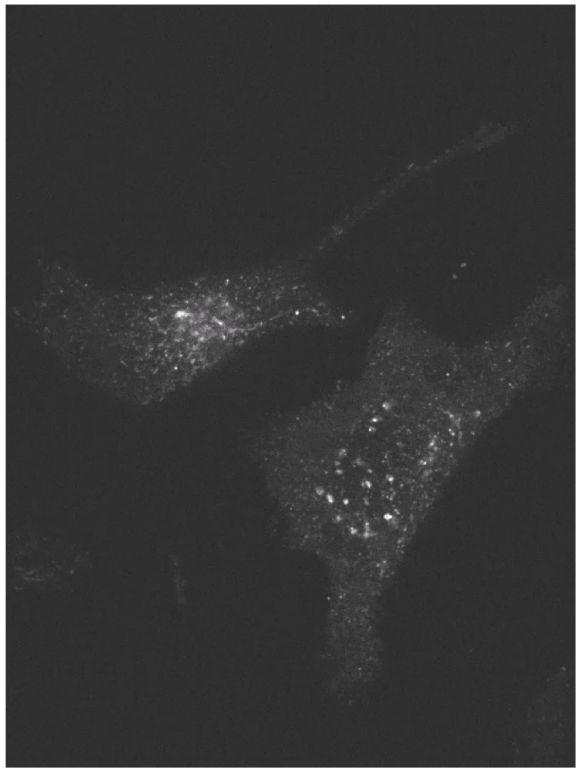
J. Boulanger, et al. **Patch-based non-local functional for denoising fluorescence microscopy image sequences**. IEEE Trans. on Medical Imaging, 29(2): 442-454, 2010

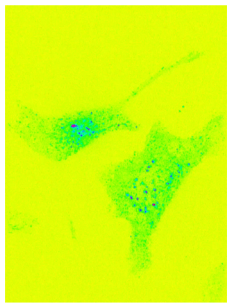
Normal power laser



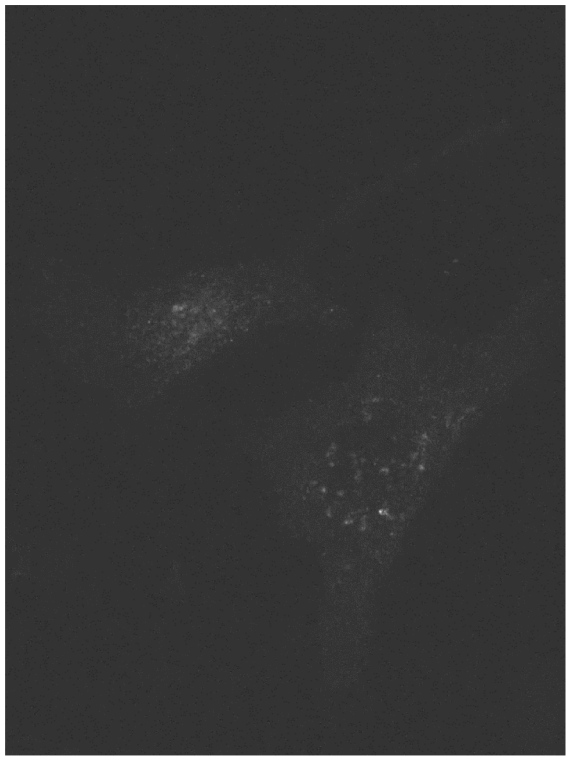


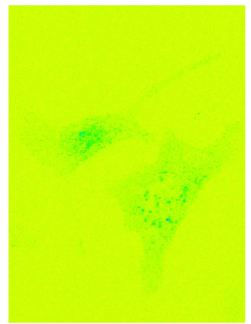
Laser divised by 2



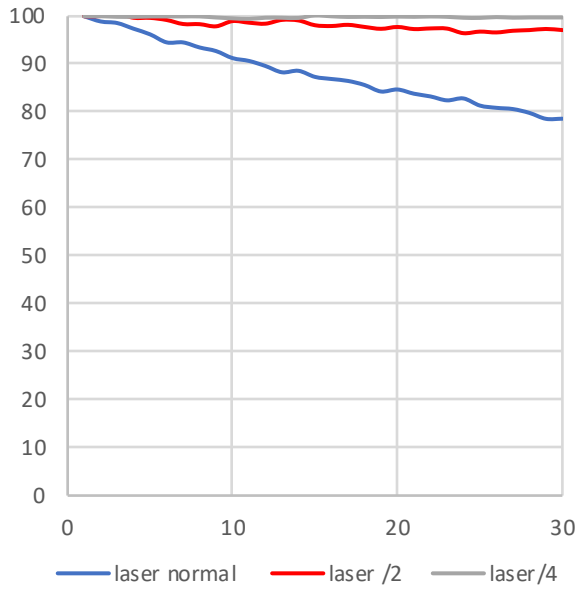


Laser divised by 4





Photobleaching comparison

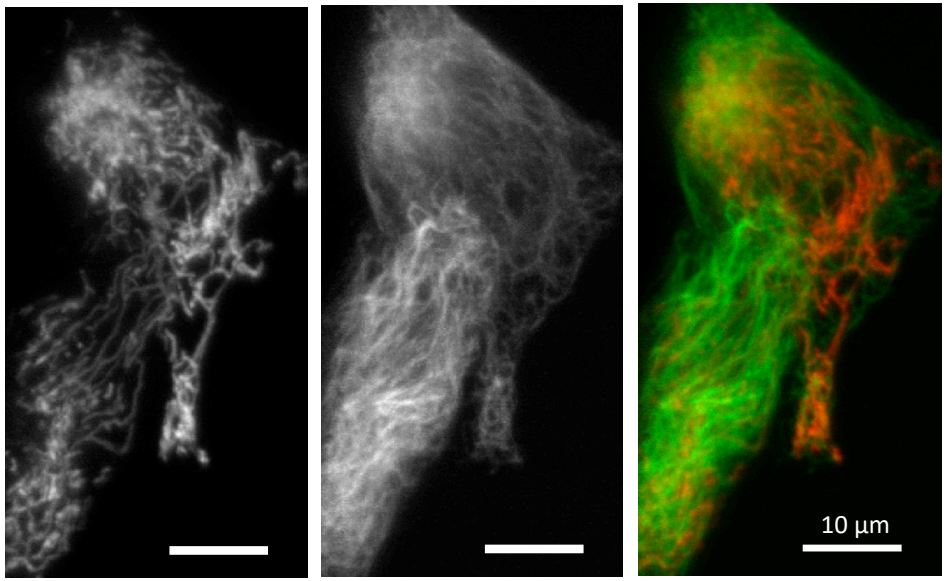


information:
Spinning3 HELA M6PR eGFP
Steps : 26 @0.3µm
30 times points
@10ms

SPITFIR(e): deconvolution and denoising 2D, 3D, 3D+t

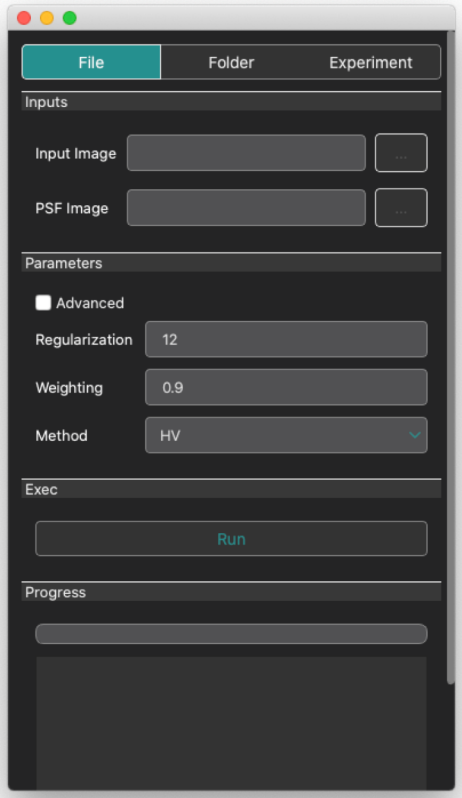
Smooth and sparse model dedicated to light microscopy: $$\mathrm{HSV}_\rho(u) = \int_\Omega \underbrace{\sqrt{\rho^2 \|\mathcal{H}u(\mathbf{x})\|_F^2 + (1-\rho)^2 u(\mathbf{x})^2}}_{\|D_{2,\rho}u(\mathbf{x})\|_2} \, d\mathbf{x}$$

LLSM acquisition
+ Deskew

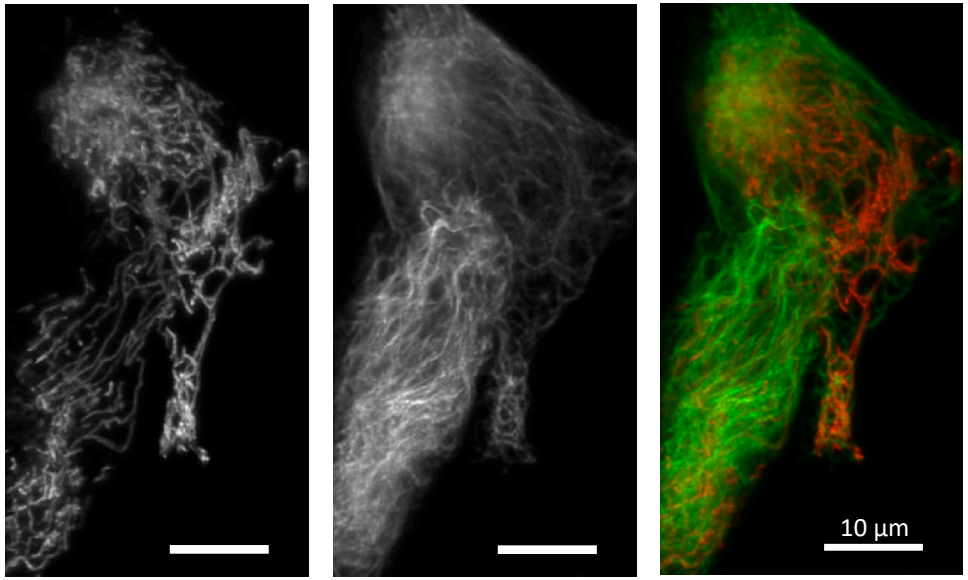


Microtubules (green) vs Mitochondria (red)
2 s/volume
RPE1 Cells

BioImage IT
SPITFIR(e)



Result



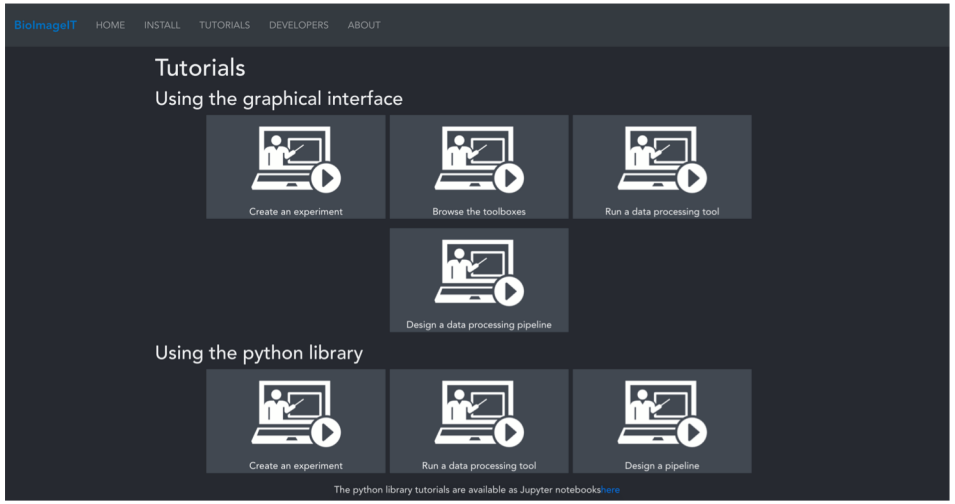
Deskew+
4D denoising+
3D Deconvolution

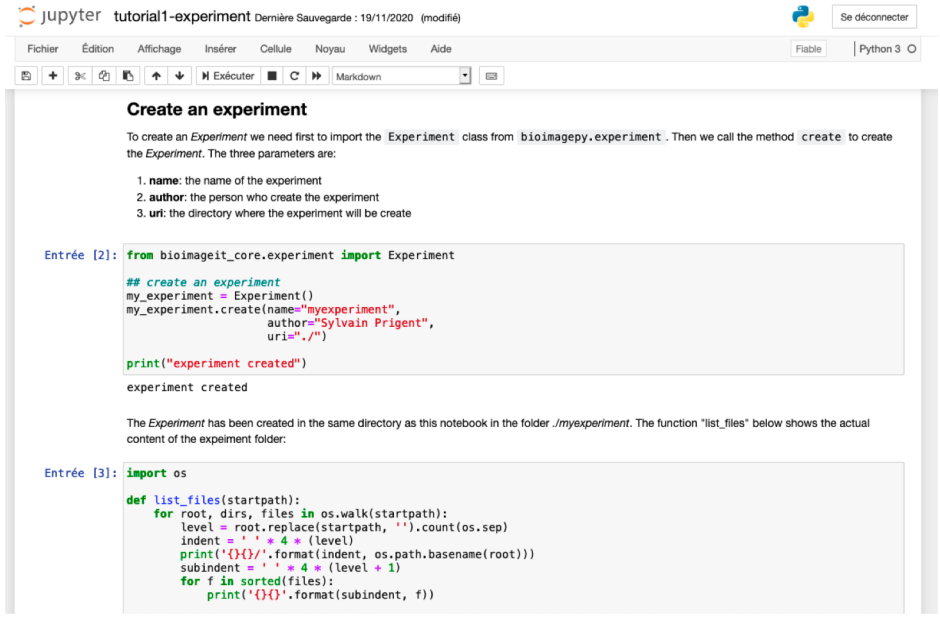
Manuscript in preparation

Resources

Website

- bioimageit.github.io





Codes repositories

- github.com/bioimageit
- https://gitlab.inria.fr/serpico

Hands-on

2D, 3D, 3D+t Image denoising

Acknowledgments

Fundings

- France-BioImaging
- EIT-Digital
- Airbus - BPI

Sted/Serpico team

- Jean Salamero
- Charles Kervrann
- Ludovic Leconte
- César Augusto Valades Cruz
- Léo Maury

Collaborators

Inria Mosaic team – ENS Lyon

- Guillaume Cerutti
- Jonathan Legrand

Institut Curie - *eLabFTW*

- Nicolas Carpi

Institut Curie – *DSI (ongoing)*

- *Alexandre Fargeot*