

MÜNSTER IMAGING NETWORK WS 44



WWU

A large, three-dimensional white text sculpture spelling out 'WWU' sits on a stone surface. The letters are thick and have soft shadows, giving them a three-dimensional appearance. In the background, a blurred view of a historic town square with buildings and flowers can be seen under a blue sky.

WS44

**High throughput & automated data analysis and data
management workflow with Cellprofiler and OMERO**

Thomas Zobel & Sarah Weischer

Münster Imaging Network, Cells in Motion Interfaculty Centre



TEAM

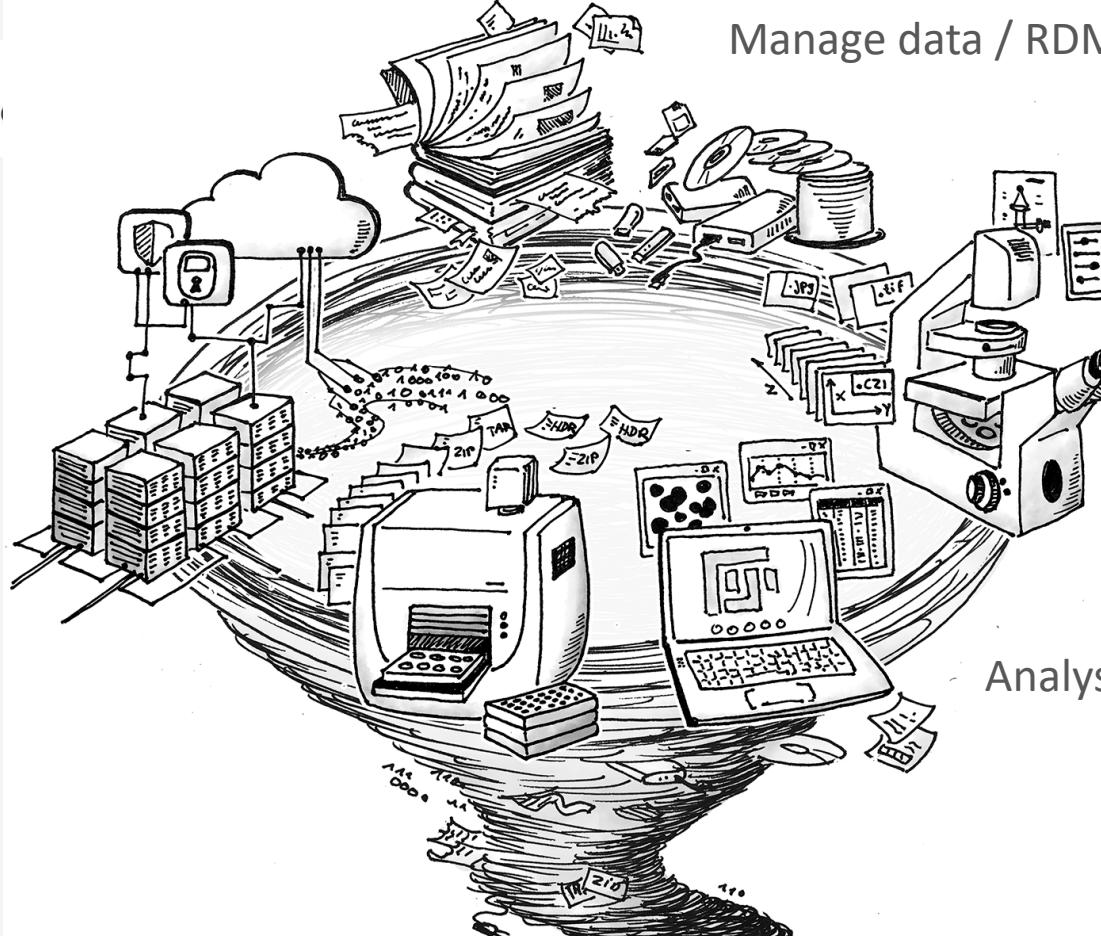


Dr. Thomas Zobel

Jens Wendt

Sarah Weischer

High throughput analysis



Acquire data / screens

Analyse data / macro or pipelines



High throughput & automated data analysis and data management workflow with Cellprofiler and OMERO

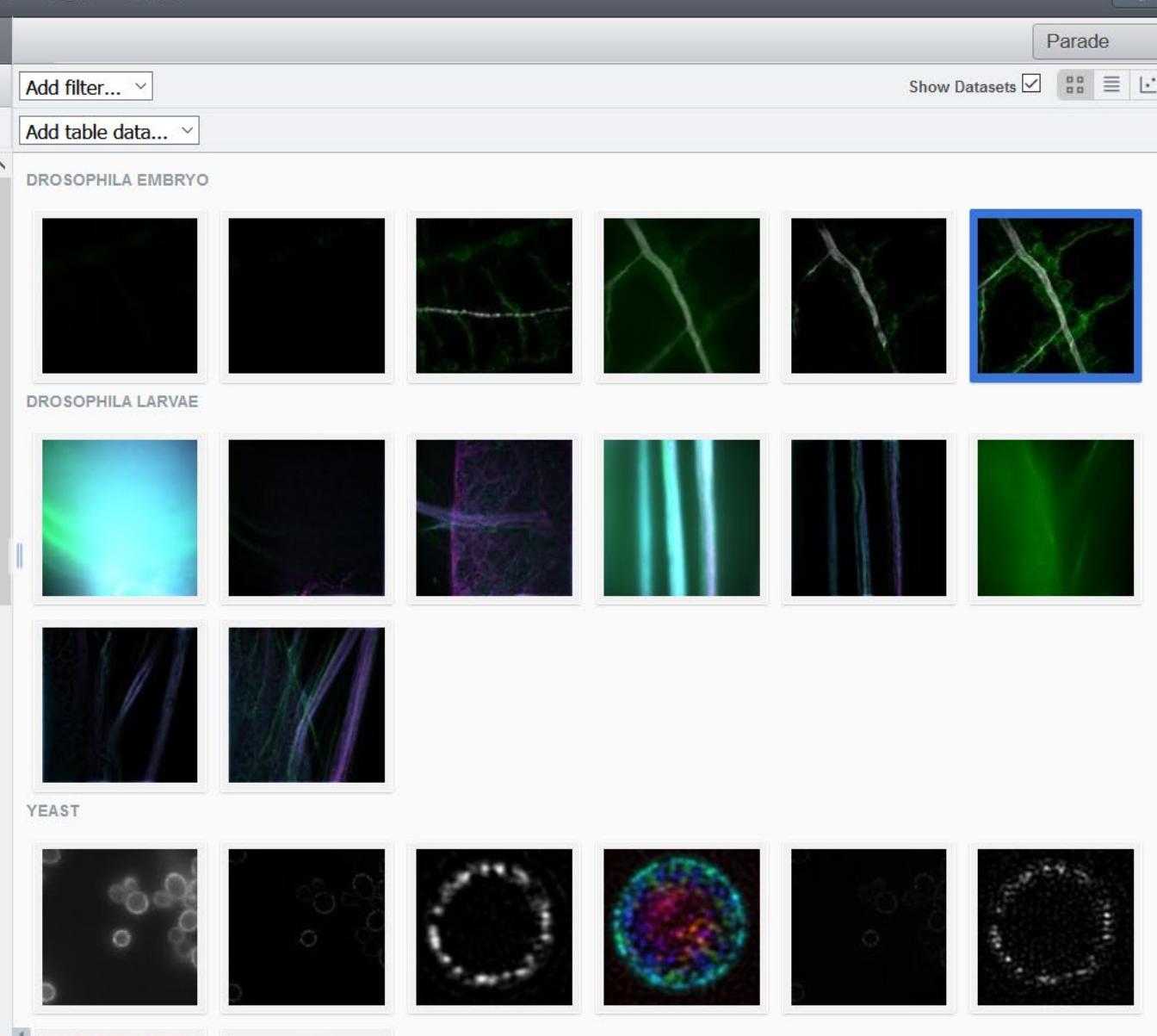


Public Example Data Thomas Zobel

Explore Tags Shares

Thomas Zobel

- DeepLearning 1
- SMLM 1
- Zeiss Elyra 7 Demo 4
 - Drosophila Embryo 6
 - ...Aptome tracks seq 488 561 simul.czi
 - ...ome tracks seq 488 561 simul_SIM.czi
 - ..._SIM_Maximum intensity projection.czi
 - ... 647 SIM tracks seq 488 561 simul.czi
 - ... SIM tracks seq 488 561 simul_SIM.czi
 - ..._SIM_Maximum intensity projection.czi
 - Drosophila Larvae 8
 - ... ZStack water obj seq ZStackMode.czi
 - ...ack water obj seq ZStackMode_SIM.czi
 - ..._SIM_Maximum intensity projection.czi
 - ... ZStack water obj seq ZStackMode.czi
 - ...ack water obj seq ZStackMode_SIM.czi
 - ...il obj ZStackMode FrameFast Dual.czi
 - ...bj ZStackMode FrameFast Dual_SIM.czi
 - ..._SIM_Maximum intensity projection.czi
 - Yeast 8
 - Image 23 777 _488 3D Time.czi
 - ...3 777 _488 3D Time_SIM_strong.czi
 - ...7 _488 3D Time_SIM_strong_crop.czi
 - ...rong_crop_Color-coded projection.czi
 - ... 777 _488 3D Time_SIMstrong12.czi
 - ...7 _488 3D Time_SIMstrong12crop.czi
 - ...ong12crop_Color-coded projection.czi
 - ...88 3D Time_SIMstrong12crop_proj.czi
 - Zeiss Live Cells 31
 - 24_LifeAcht tdTomato Apotome.czi



General Acquisition Preview

Import Date: 2020-03-02 10:31:45
 Dimensions (XY): 2048 x 2048
 Pixels Type: uint16
 Pixels Size (XYZ) (µm): 0.03 x 0.03 x 0.11
 Z-sections/Timepoints: 1 x 1
 Channels: TV1-T1-SR, TV1-T2-SR, TV2-T2-SR
 ROI Count: 0

Tags 6

VNC - Processed - MIP -
 Lattice SIM - ELYRA 7 - D. melanogaster -

Key-Value Pairs 1

Added by: Thomas Zobel

Key	Value
Organism	Drosophila melanogaster
Imaging Method	Lattice SIM
Organism Part	3rd instar Larvae
Sample Condition	Fillet
1 Antibody	HRP
2. Antibody	Alexa488
Fixation	4% PFA
Mounting	Mowiol

Tables

Attachments 1

Figure_2021-3-3_21-56-30.pdf (5.11 MB)

Comments 0

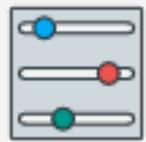
OMERO Features



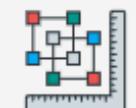
- Import over 140 image formats from different microscopes (Bio-Formats)
- Read the original metadata (pixel size → scale bar)



- View your image data over the internet from anywhere
- Browse thumbnails, image previews and view Z-projections



- Adjust the rendering settings for your images
- See the settings from other group members



- Draw ROIs and use basic analysis



- Add metadata like tags, key-value pairs, ratings or comments to multiple or single images



- Use OMERO.figure to quickly draw figures for presentations, meetings and publications
- Export figures as pdf, tif and vector graphics



- Share your data within your group including annotations, comments, adjustments and tags



- Open images directly in ImageJ/Fiji
- Save ROIs, overlays and results from Fiji into OMERO

OMERO Features – Screens

OMERO Data History Help Tag Search Figure Admin

Search: Thomas Zobel

MiN - Münster Imaging Network Jens Wendt

Explore Tags Shares

Index: Field#1

Thumbnails

A1

Well ID: 5815
Owner: Jens Wendt

Tags 0

Key-Value Pairs 0

Add Key Add Value

Tables

Attachments 0

Comments 0

Ratings 0

Others 0

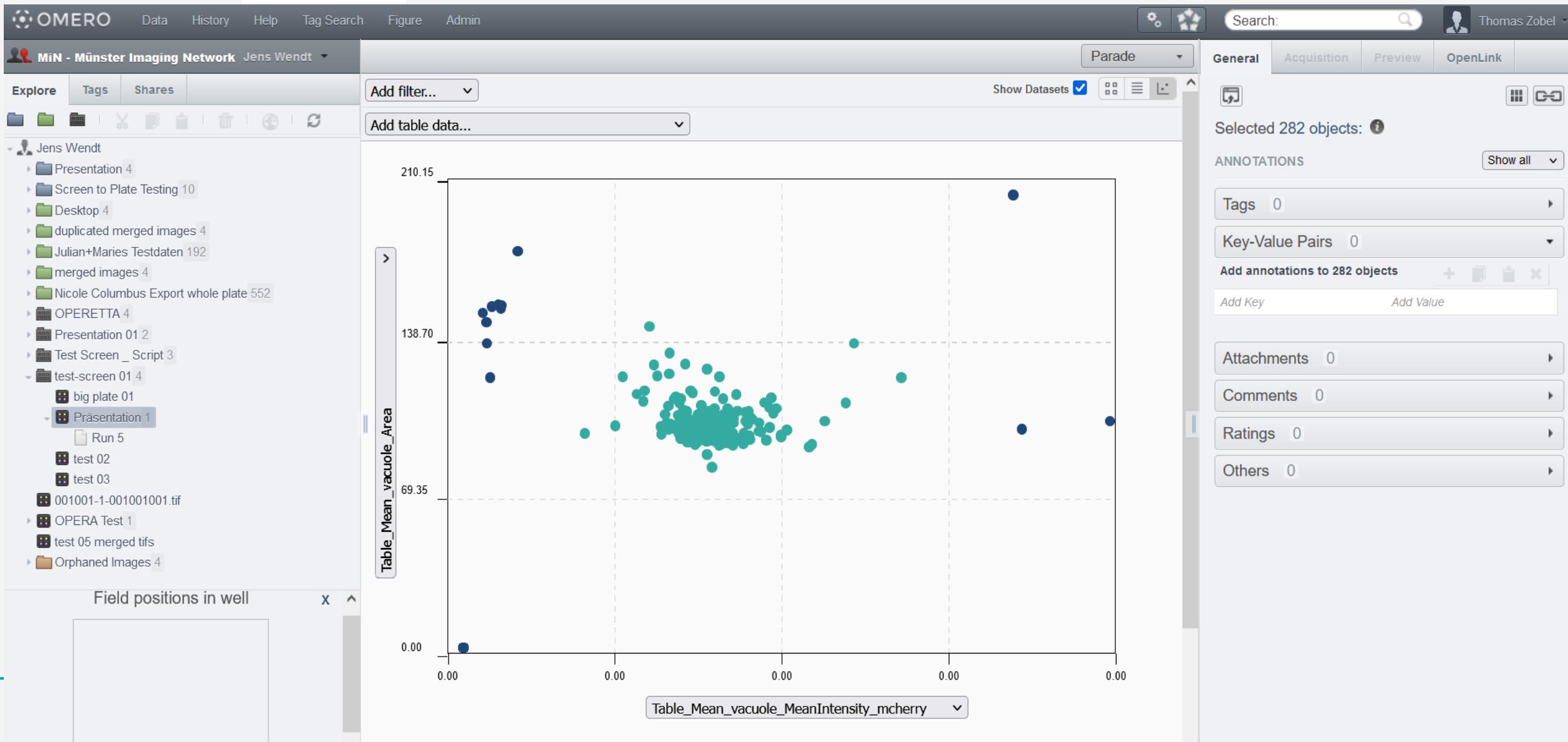
Field positions in well

wrap images per row Fields from well

A1

The OMERO interface displays a grid of microscopy images (A1-P1) arranged in 12 rows (A-P) and 21 columns (1-21). A specific image in row A, column 1 is highlighted. Below the grid, a 'Field positions in well' panel shows a zoomed-in view of fields A1 and P1. To the right, a detailed view of well A1 is shown, including its well ID (5815), owner (Jens Wendt), and various metadata sections like Tags, Key-Value Pairs, and Tables.

OMERO Features – Screens



OMERO Features – Screens

OMERO Data History Help Tag Search Figure Admin

MiN - Münster Imaging Network Jens Wendt

Parade

Add filter... Show Datasets

Add table data...

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A																								
B																								
C																								
D																								
E																								
F																								
G																								
H																								
I																								
J																								
K																								
L																								
M																								
N																								
O																								
P																								

Field positions in well X

Selected 282 objects: i

Annotations

Tags 0

Key-Value Pairs 0

Add annotations to 282 objects

Add Key Add Value

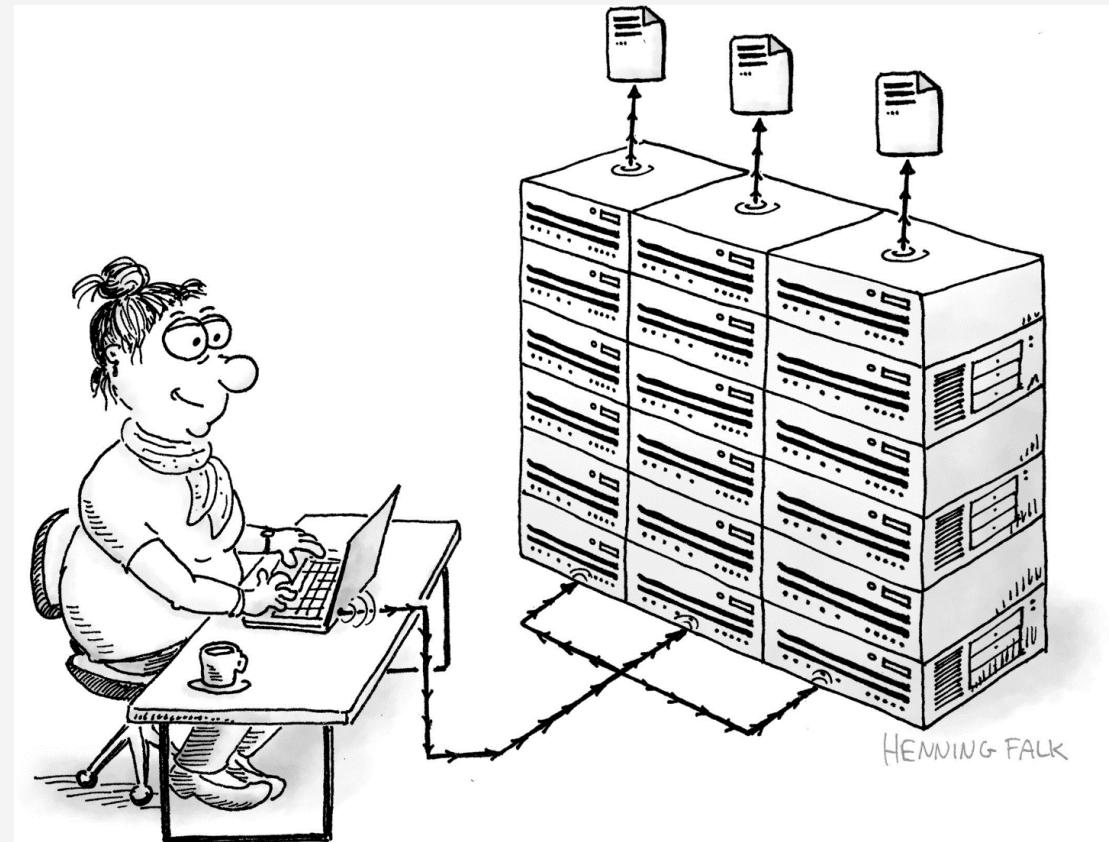
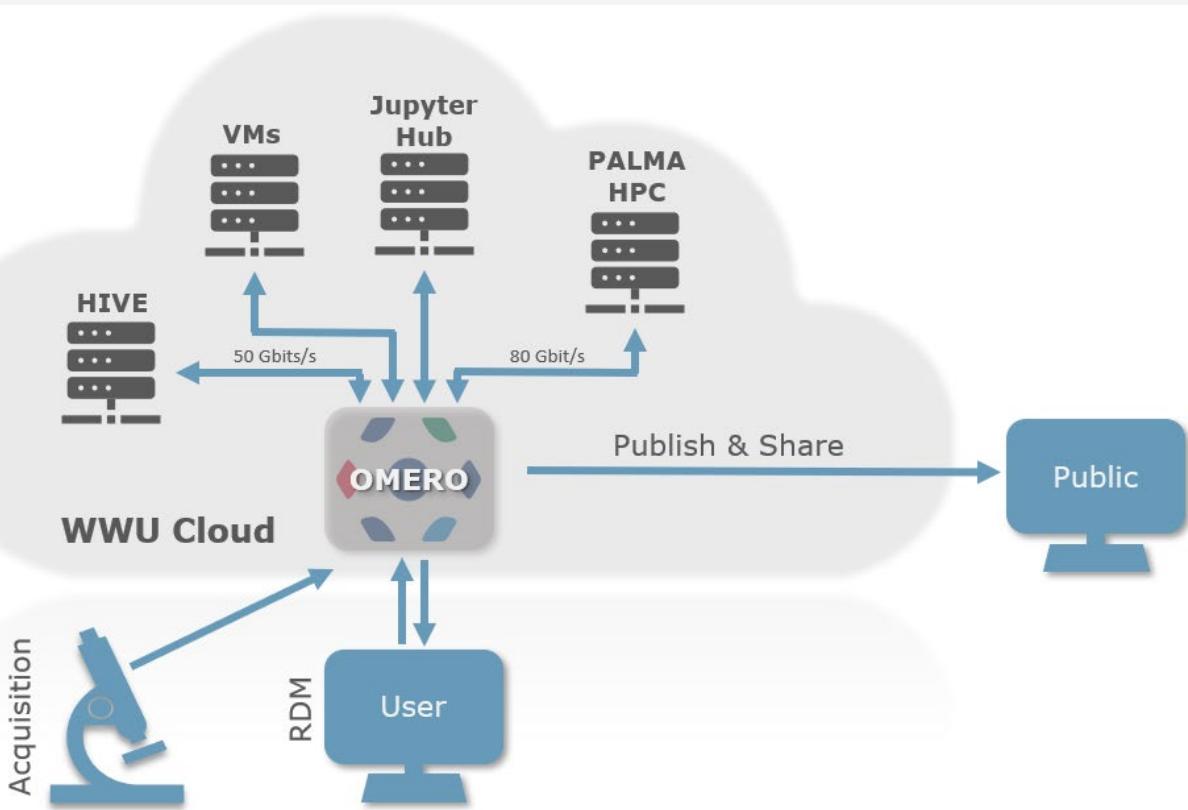
Attachments 0

Comments 0

Ratings 0

Others 0

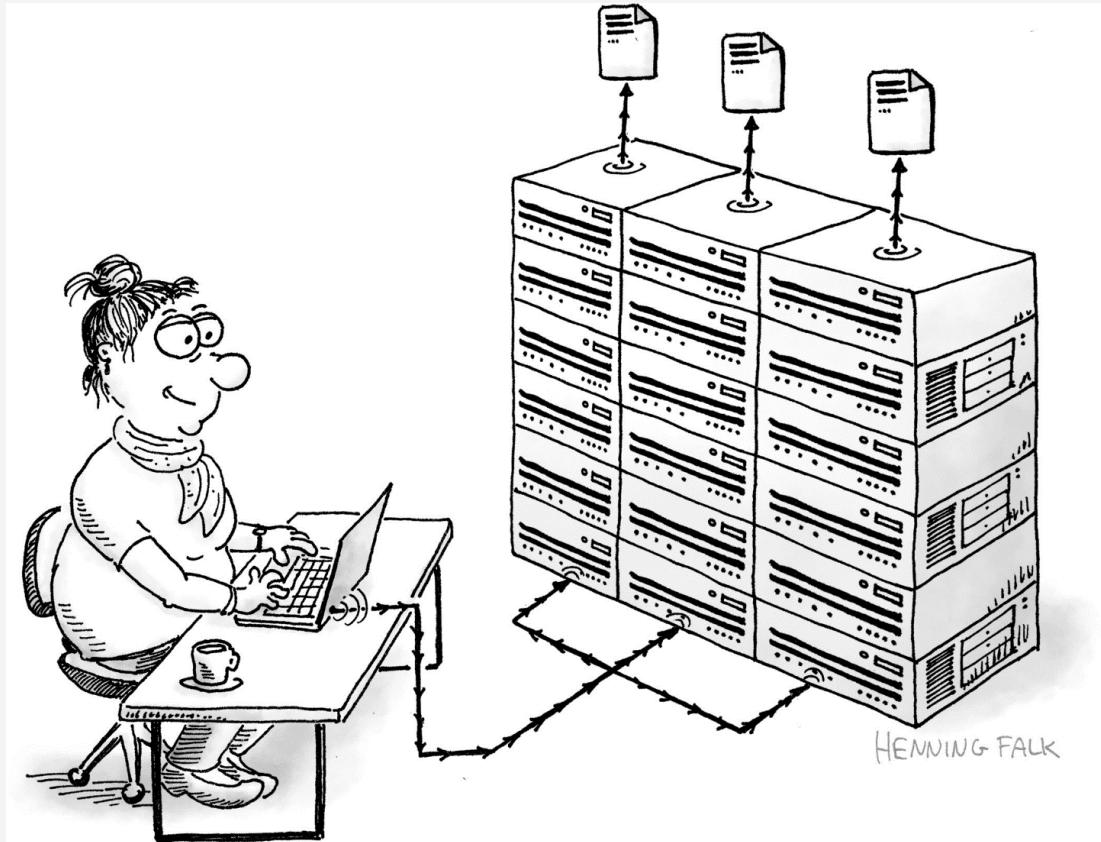
OMERO @MIN – IT-Infrastructure



Falk, H. zarr-developers/zarr-illustrations-falk-2022 | Zenodo, 2022. URL
<https://doi.org/10.5281/zenodo.7037367> (accessed 8.31.22).

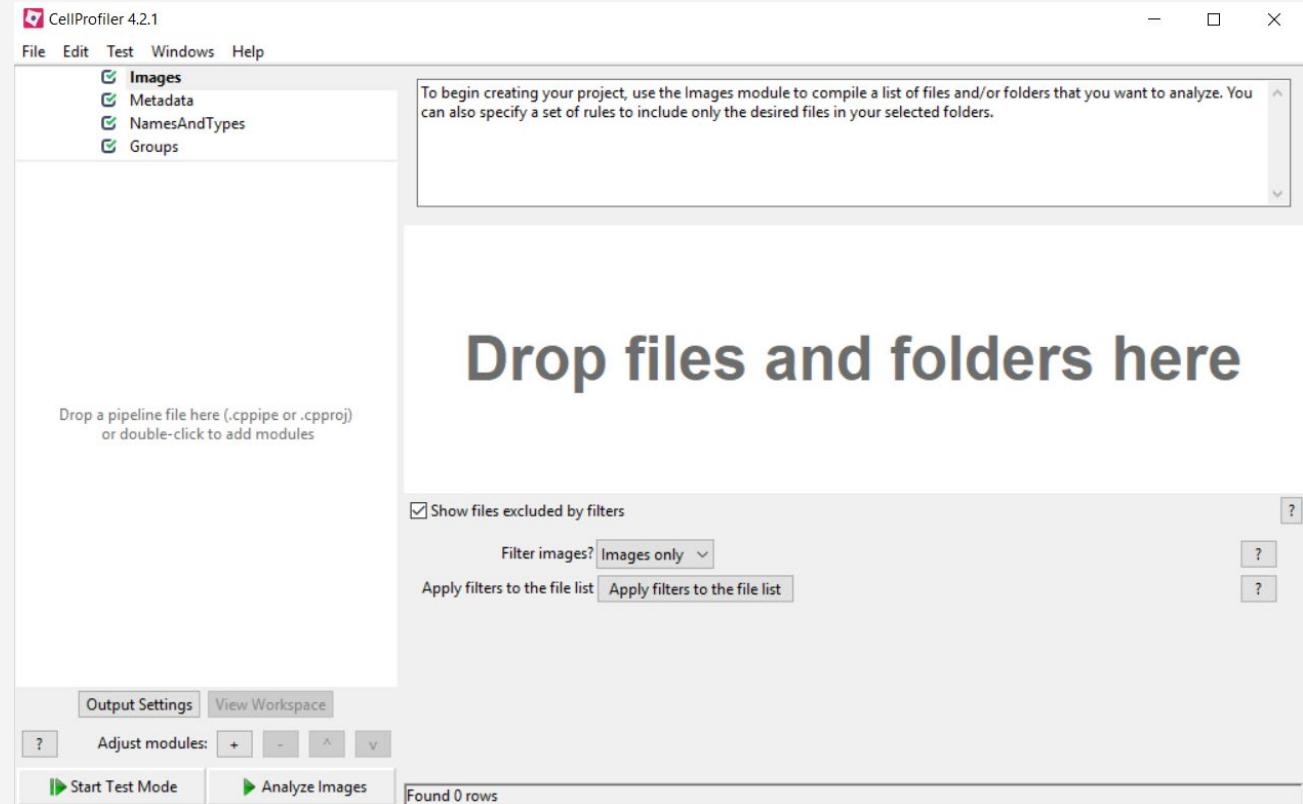
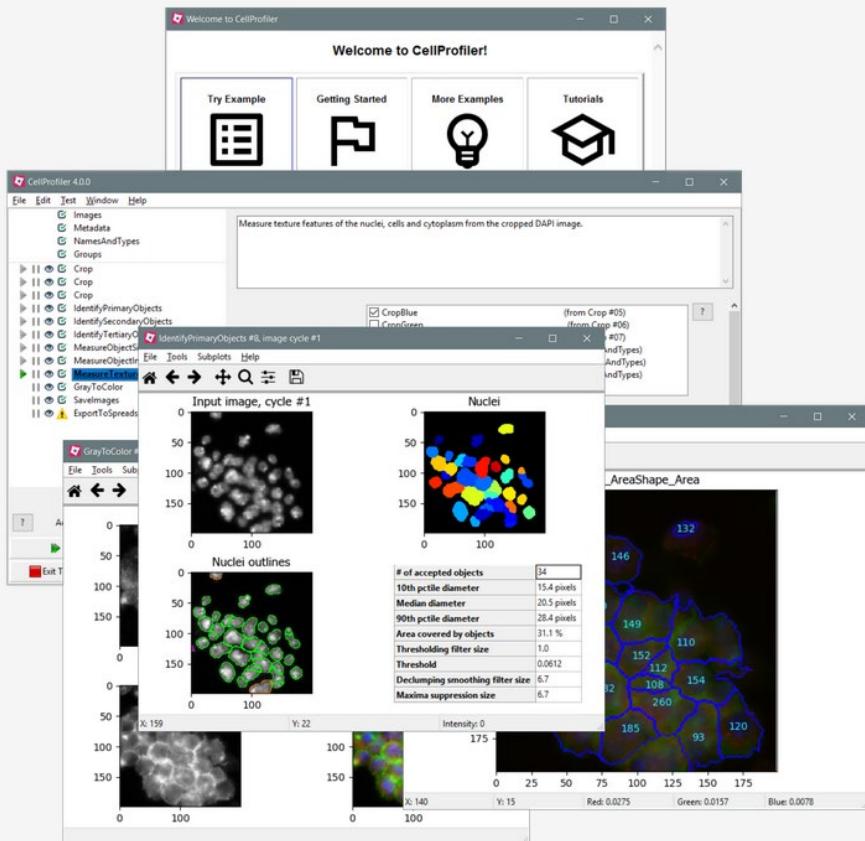
HIVE Specifications:

CPU: **256 Cores (AMD EPYC 7702)**
RAM: **1024 GB**
GPU: **Quattro RTX 8000 42GB**
Storage: **12 TB SSD RAID + 2 TB M2 SSD with 3200MB/s**
OS: **Windows Server 2019**
Net.: **50Gbit/s to WWU-Cloud**



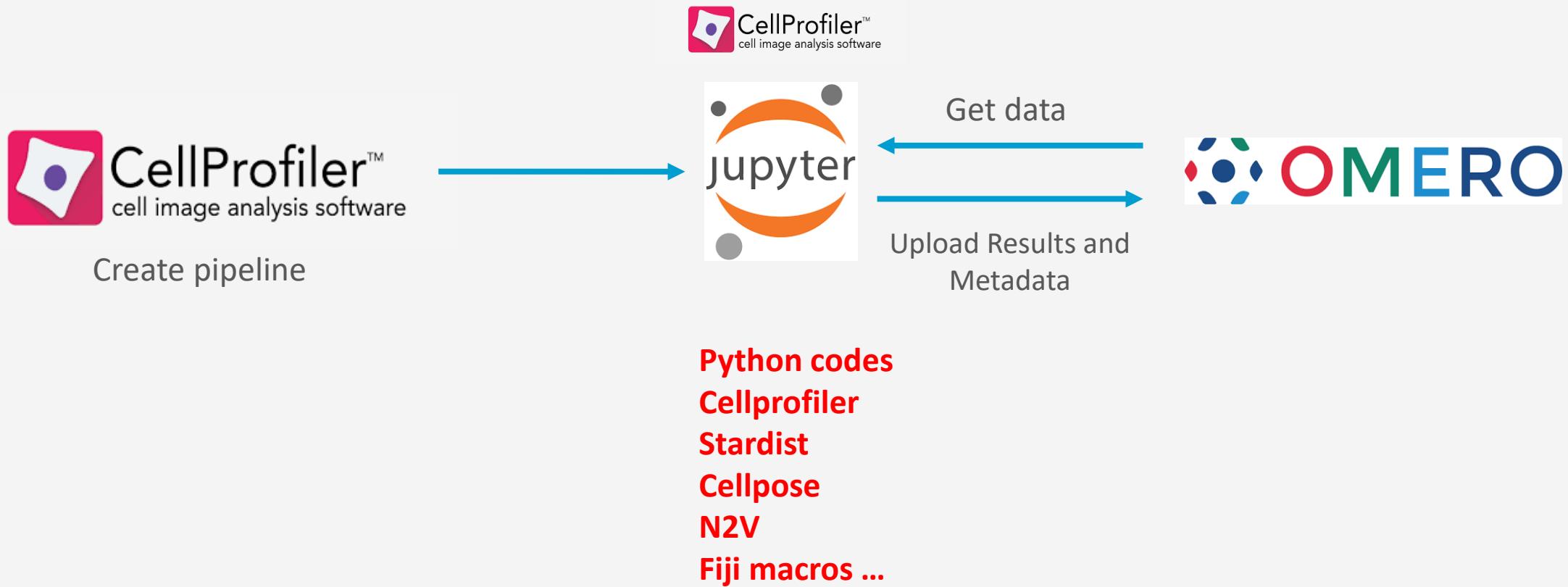
Falk, H. zarr-developers/zarr-illustrations-falk-2022 | Zenodo, 2022. URL <https://doi.org/10.5281/zenodo.7037367> (accessed 8.31.22).

Cellprofiler

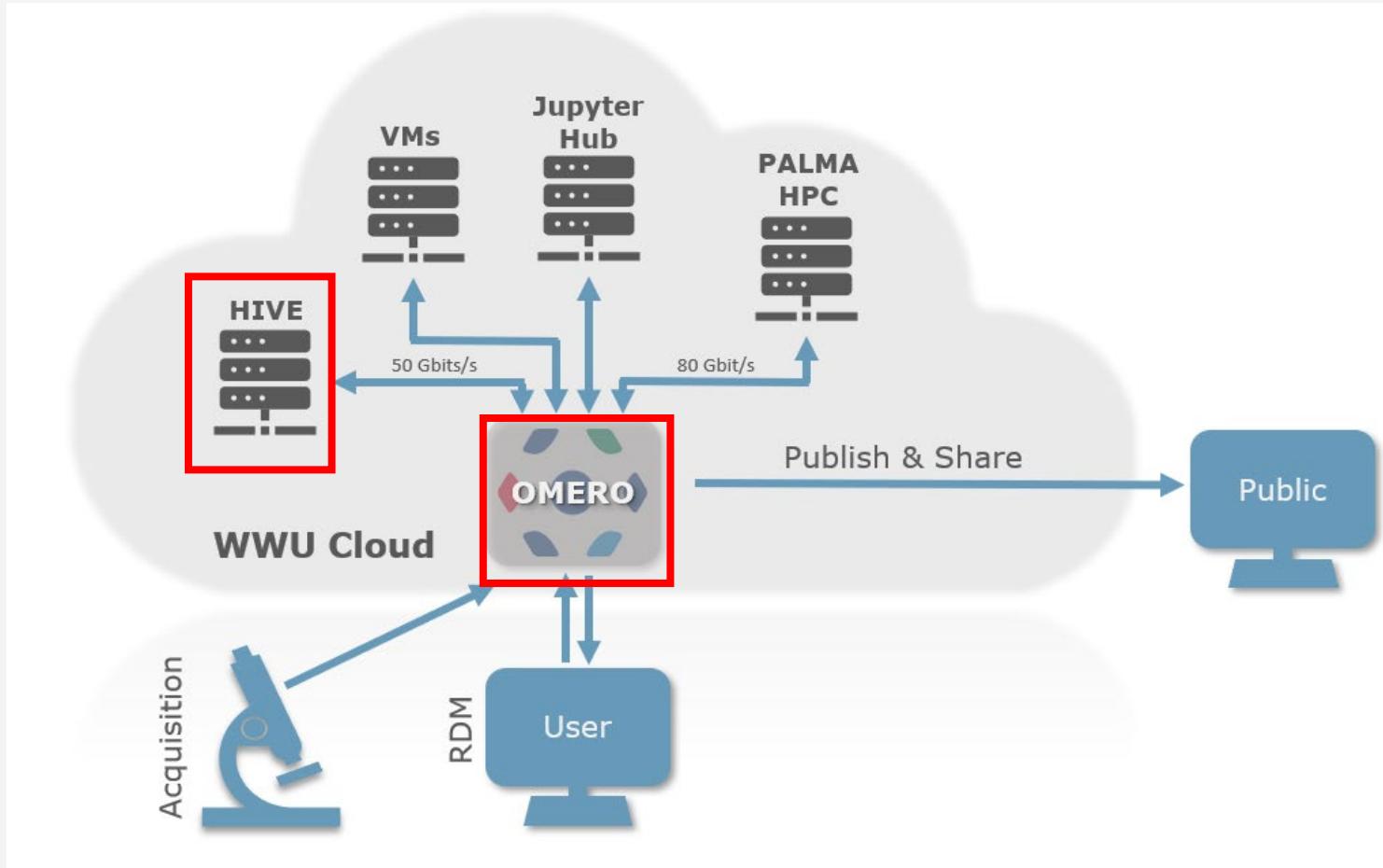


Why Cellprofiler?

For users: Easy adjustable image analysis pipeline

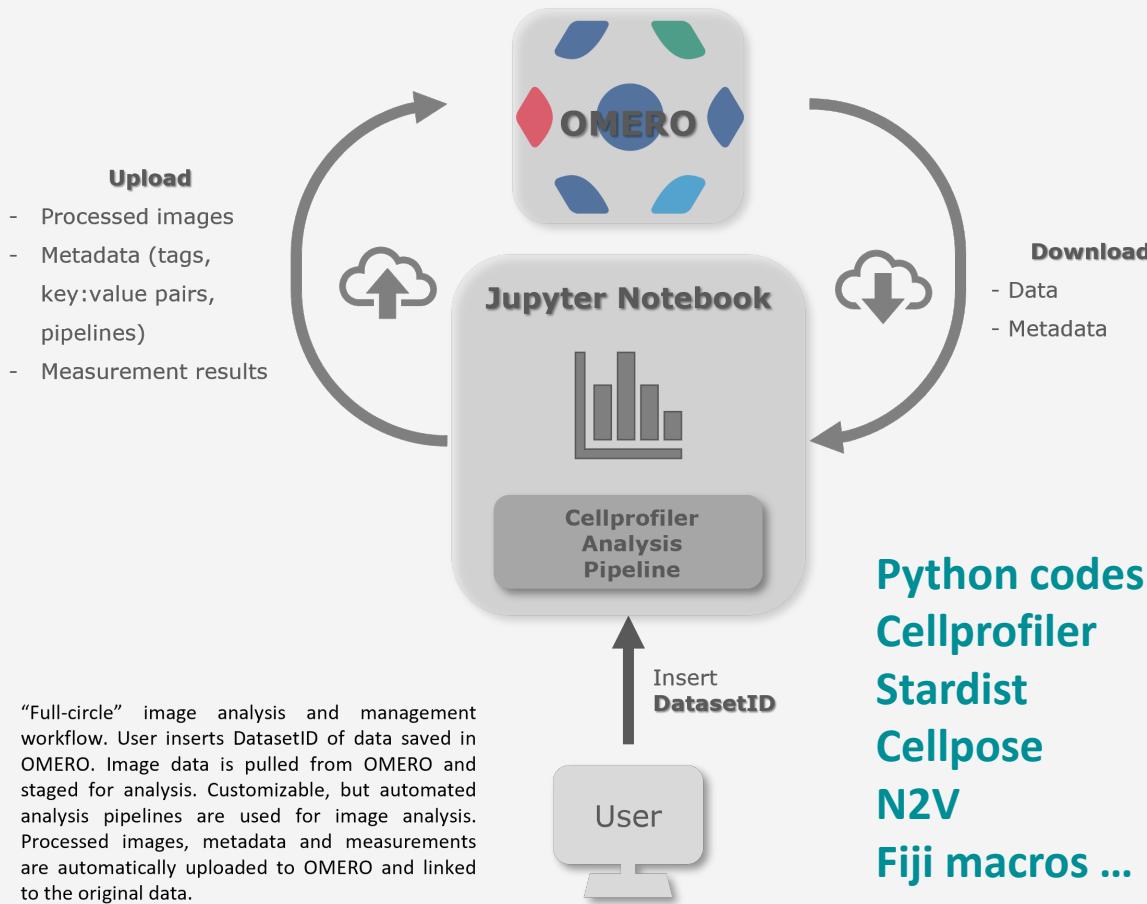


Data Management and Analysis Infrastructure MiN



- OMERO server fully integrated in WWU Cloud
- HIVE (Analysis server):
 - high speed access to OMERO server/data
 - dedicated image analysis software
 - custom python-environments for data analysis (cell profiler, deep learning etc)

High throughput & automated data analysis and data management workflow with Cellprofiler and OMERO



- Cellprofiler analysis pipeline for user-friendly analysis parameter tuning for own data (e.g., different cell types)
- Prevent manual down- and upload of data
- Automated upload of used pipeline as documentation
- User-guided tagging of resulting images for data organization

Tasks during the workshop

- 1 Data preparation for analysis
- 2 Data injection into analysis pipeline
- 3 Automated data analysis using image analysis pipelines (e.g., Cellprofiler)
 - Prepared pipeline for nuclei and cell segmentation using **cellpose***
 - Selected cell shape measurements*
- 4 Upload of the resulting images (including tags and metadata) and measurement results (omero.tables)
- 5 Explorative data analysis using omero.parade/omero.parade-crossfilter

Dataset and Code

Dataset

The data used in this workshop is derived from Pascual-Vargas et al., Sci Data, 2017 "RNAi screens for Rho GTPase regulators of cell shape and YAP/TAZ localisation in triple negative breast cancer" DOI: 10.1038/sdata.2017.18

The data is publicly available in the Image Data Resource (idr0028) <https://idr.openmicroscopy.org/webclient/?show=screen-1651>

The datasets contain RNAi screens of cancer cells that were stained with Hoechst (Nuclei), Tubulin, Actin and Yap/Taz.

Licenses & Code

The code presented here is partially based on the following scripts and resources:

- Omero Dataset_To_Plate.py script by Will Moore, OME Team, Copyright © 2006-2014 University of Dundee. All rights reserved. Source: https://github.com/ome/omero-scripts/blob/68c7505e62115e9c086a8e5a1d3edc1d4aff35f3/omero/util_scripts/Dataset_To_Plate.py
- InjectImage module for Cellprofiler. Copyright © 2020-2021 University of Dundee. All rights reserved. Source: <https://omero-guides.readthedocs.io/en/latest/cellprofiler/docs/index.html>; <https://github.com/ome/omero-guide-cellprofiler>
- General Omero-Python API documentation, Source: <https://omero-guides.readthedocs.io/en/latest/python/docs/gettingstarted.html>
- Cellprofiler Python API, Copyright © 2003 - 2021 Broad Institute, Inc. All rights reserved. Source: <https://github.com/CellProfiler/CellProfiler/wiki/CellProfiler-as-a-Python-package>
- ezomero (<https://github.com/TheJacksonLaboratory/ezomero>)

Structure Jupyter Notebook

Define Variables

Screen/PlateID, Pipeline, Channels

Prepare Images

Get image information from omero

Prepare Cellprofiler

Prepare JVM, adjust pipeline (import of images using “injectimages”, export)

Run Pipeline

Segmentation of cells, certain measurements on cell shape and protein expression

Upload results to OMERO

Upload image data, measurement results and pipeline, add tags and key-value pairs, link original images and result images

Practical Part - Preparations

- 1) Set-up your Conda Environment
 - See the manual here: https://github.com/MuensterImagingNetwork/TiM23_WS44_HTPImageAnalysis_Omero/tree/main/InstallationCondaEnv
- 2) Configure Cellprofiler
 - Set plugin-folder
 - Set max number of workers
 - Conserve system memory
- 3) Test the cellprofiler pipeline
 - Check the cellpose configuration
 - Run pipeline on test images
 - Wrong segmentation? Check your cellpose models
- 4) Start the Jupyter Notebook

2 Configure Cellprofiler

Start Cellprofiler

```
Start cellprofiler
1 conda activate cellprofiler
2 cellprofiler
```

Cellprofiler Settings

CellProfiler plugins directory

Display welcome text on startup

Warn if Java runtime environment not present

Show the "Analysis complete" message at the end of a run

Show the "Exiting test mode" message

Warn if images are different sizes

Show the sampling menu

Maximum number of workers

Temporary folder

Save pipeline and/or file list in addition to project

Conserve system memory

3 Test Cellprofiler Pipeline

The NamesAndTypes module allows you to assign a meaningful name to each image by which other modules will refer to it.

Assign a name to **Images matching rules**

Process as 3D? Yes No

Match **All** of the following rules

Select the rule criteria File Does Contain C1

Name to assign these images **Nuclei**

Select the image type Grayscale image

Set intensity range from Image metadata

Duplicate this image

Match **All** of the following rules

Select the rule criteria File Does Contain C2

Name to assign these images **Actin**

Select the image type Grayscale image

Set intensity range from Image metadata

Duplicate this image

Remove this image

Match **All** of the following rules

Select the rule criteria File Does Contain C3

Name to assign these images **Tubulin**

Select the image type Grayscale image

Set intensity range from Image metadata

Duplicate this image

Remove this image

Update	Actin	Nuclei	Tubulin	YapTaz
1	014022-28_C2.tiff	014022-28_C1.tiff	014022-28_C3.tiff	014022-28_C4.tiff

3 Test Cellprofiler Pipeline

<input checked="" type="checkbox"/> Images
<input checked="" type="checkbox"/> Metadata
<input checked="" type="checkbox"/> NamesAndTypes
<input checked="" type="checkbox"/> Groups
<input checked="" type="checkbox"/> GrayToColor
<input checked="" type="checkbox"/> RescaleIntensity
<input checked="" type="checkbox"/> RescaleIntensity
<input checked="" type="checkbox"/> RescaleIntensity
<input checked="" type="checkbox"/> RunCellpose
<input checked="" type="checkbox"/> RunCellpose
<input checked="" type="checkbox"/> FilterObjects
<input checked="" type="checkbox"/> RelateObjects
<input checked="" type="checkbox"/> OverlayOutlines
<input checked="" type="checkbox"/> MeasureObjectIntensity
<input checked="" type="checkbox"/> CalculateMath
<input checked="" type="checkbox"/> DisplayDataOnImage
<input checked="" type="checkbox"/> MeasureObjectSizeShape
<input checked="" type="checkbox"/> ConvertObjectsToImage
<input checked="" type="checkbox"/> ConvertObjectsToImage
<input checked="" type="checkbox"/> Savelimages
<input checked="" type="checkbox"/> ExportToSpreadsheet



Detection mode Cells

Select the input image Actin_Rescaled (from RescaleIntensity #08)

Supply nuclei image as well? Yes No

Expected object diameter 60

Name the output object Cytoplasm_Seg

Save probability image? Yes No

Use averaging Yes No

Use GPU Yes No

Detection mode Nuclei

Select the input image Nuclei_Rescaled (from RescaleIntensity #07)

Expected object diameter 30

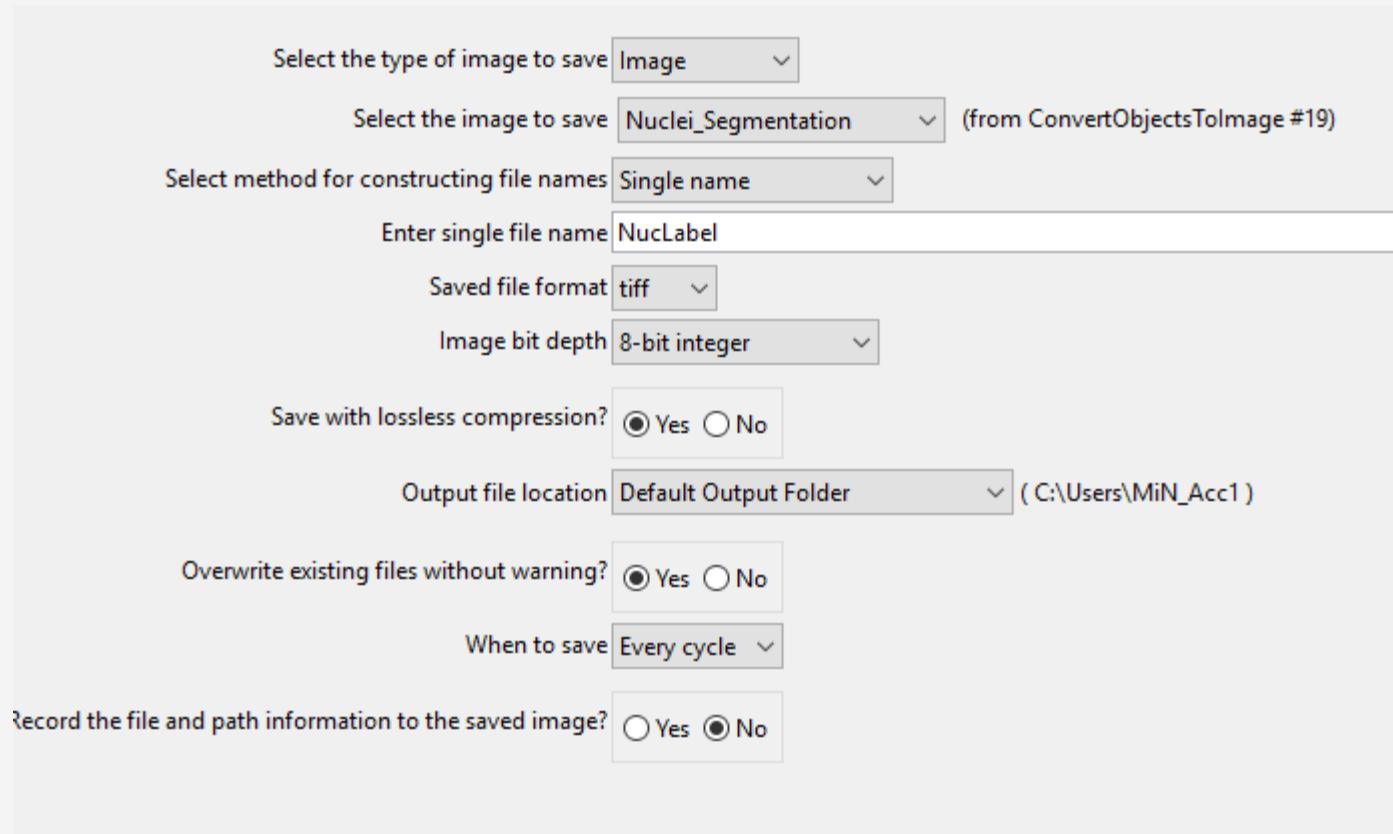
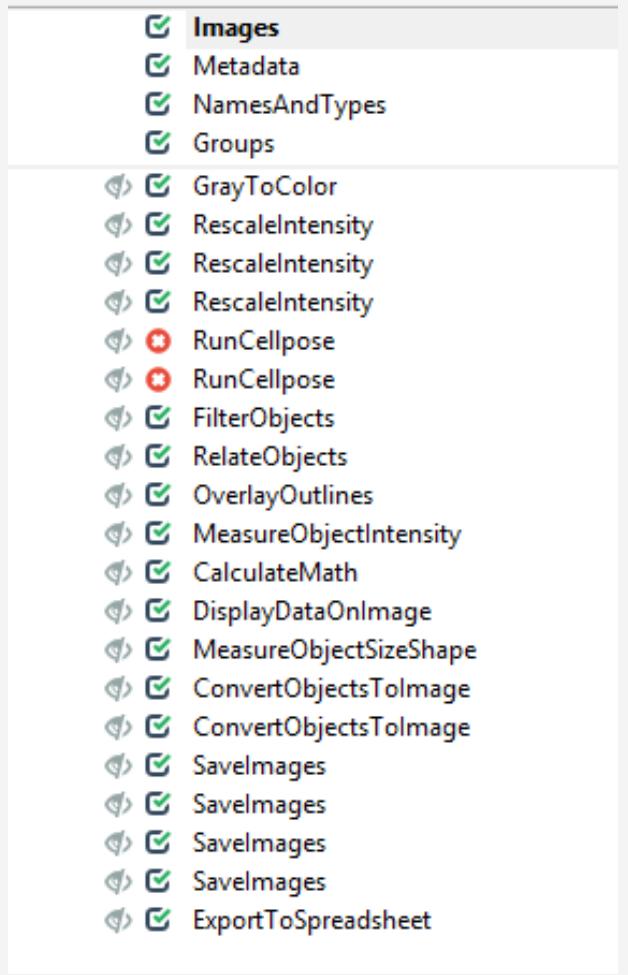
Name the output object Nuclei_Seg

Save probability image? Yes No

Use averaging Yes No

Use GPU Yes No

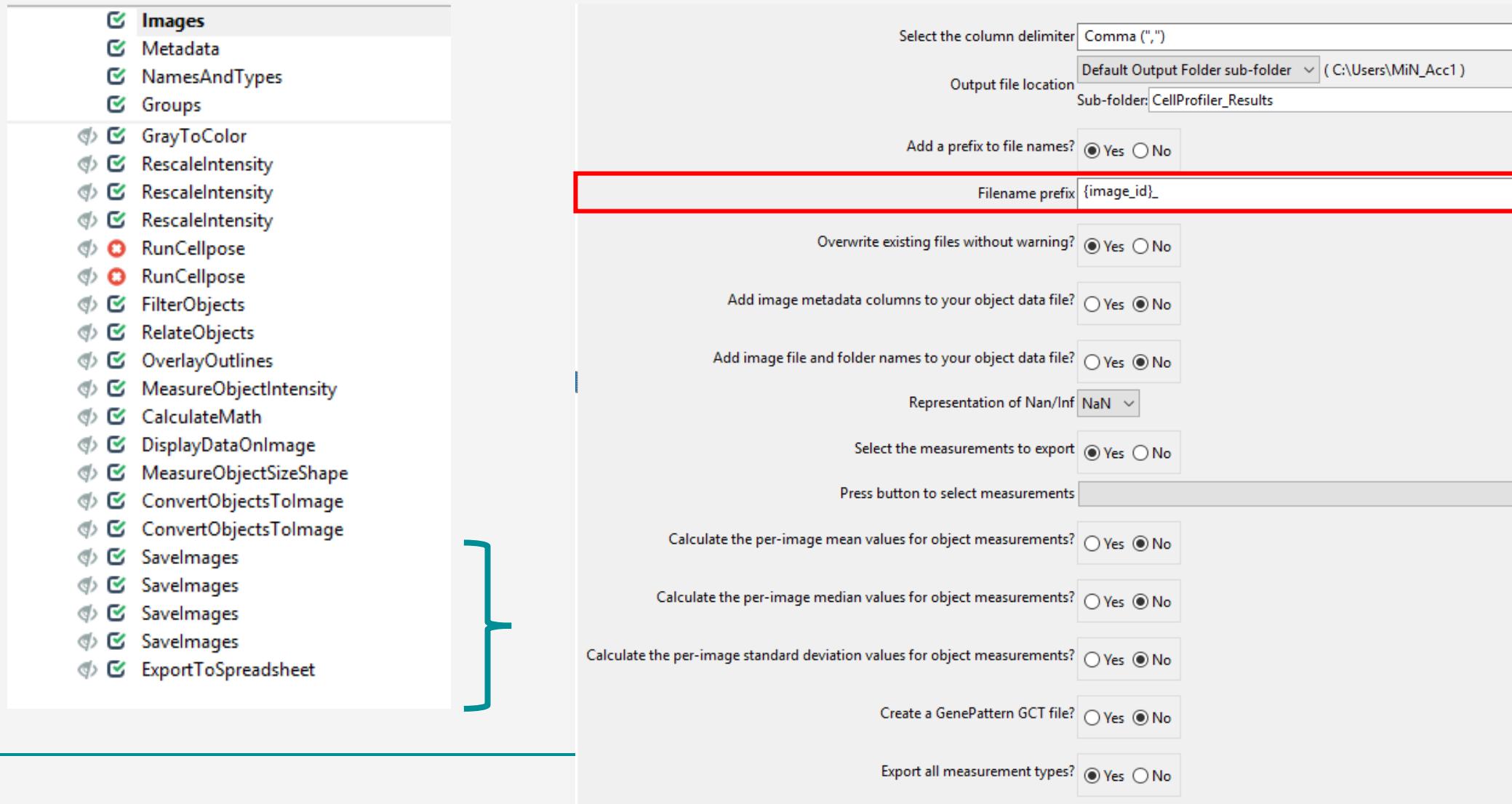
3 Test Cellprofiler Pipeline



A screenshot of the 'Save Image' dialog box. The dialog contains several configuration options:

- Select the type of image to save: Image
- Select the image to save: Nuclei_Segmentation (from ConvertObjectsToImage #19)
- Select method for constructing file names: Single name
- Enter single file name: NuLabel
- Saved file format: tiff
- Image bit depth: 8-bit integer
- Save with lossless compression?: Yes (radio button selected)
- Output file location: Default Output Folder (C:\Users\MiN_Acc1)
- Overwrite existing files without warning?: Yes (radio button selected)
- When to save: Every cycle
- Record the file and path information to the saved image?: No (radio button selected)

3 Test Cellprofiler Pipeline



The screenshot shows the CellProfiler Pipeline Editor interface. On the left, a list of processing steps is displayed, each with a checkbox and a small icon. A teal bracket is positioned to the right of the first few steps. The steps listed are:

- Images
- Metadata
- NamesAndTypes
- Groups
- GrayToColor
- RescaleIntensity
- RescaleIntensity
- RescaleIntensity
- RunCellpose
- RunCellpose
- FilterObjects
- RelateObjects
- OverlayOutlines
- MeasureObjectIntensity
- CalculateMath
- DisplayDataOnImage
- MeasureObjectSizeShape
- ConvertObjectsToImage
- ConvertObjectsToImage
- Savelimages
- Savelimages
- Savelimages
- Savelimages
- ExportToSpreadsheet

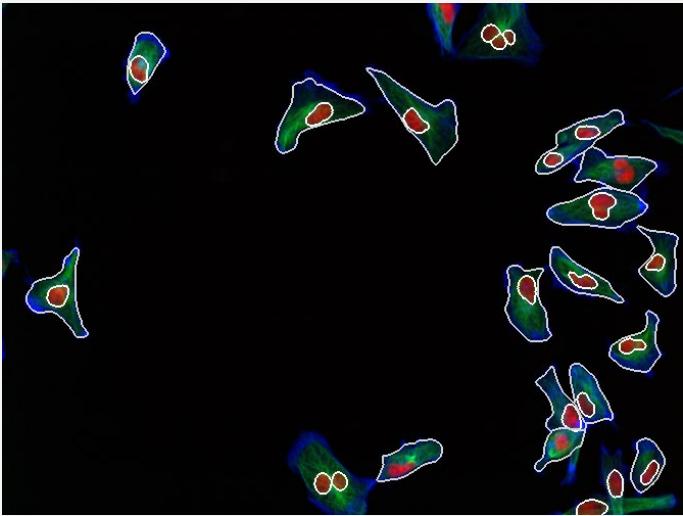
The main panel contains various export settings:

- Select the column delimiter: Comma (",")
- Output file location: Default Output Folder sub-folder (C:\Users\MiN_Acc1)
- Sub-folder: CellProfiler_Results
- Add a prefix to file names? Yes No
- Filename prefix: {image_id}_
- Overwrite existing files without warning? Yes No
- Add image metadata columns to your object data file? Yes No
- Add image file and folder names to your object data file? Yes No
- Representation of Nan/Inf: NaN
- Select the measurements to export Yes No
- Press button to select measurements
- Calculate the per-image mean values for object measurements? Yes No
- Calculate the per-image median values for object measurements? Yes No
- Calculate the per-image standard deviation values for object measurements? Yes No
- Create a GenePattern GCT file? Yes No
- Export all measurement types? Yes No

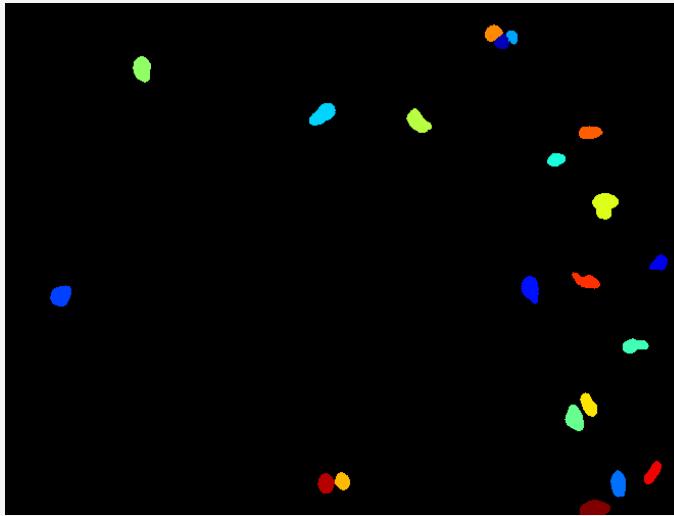
3 Test Cellprofiler Pipeline

Output of the pipeline

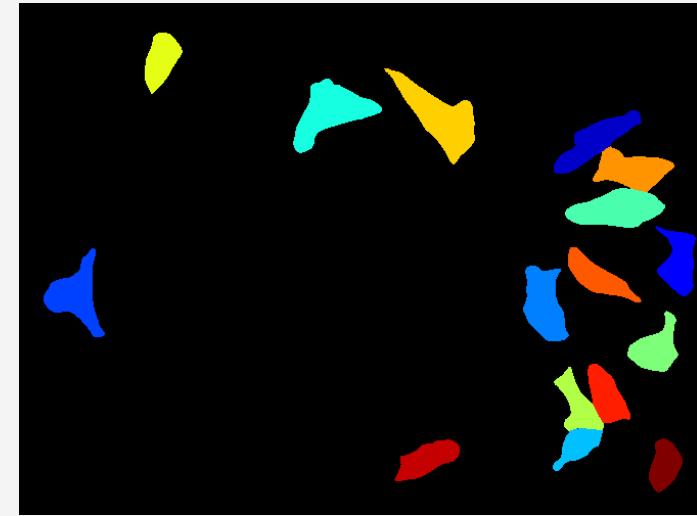
Overlay of segmentation



Nuclei Segmentation



Cytoplasmic Segmentation

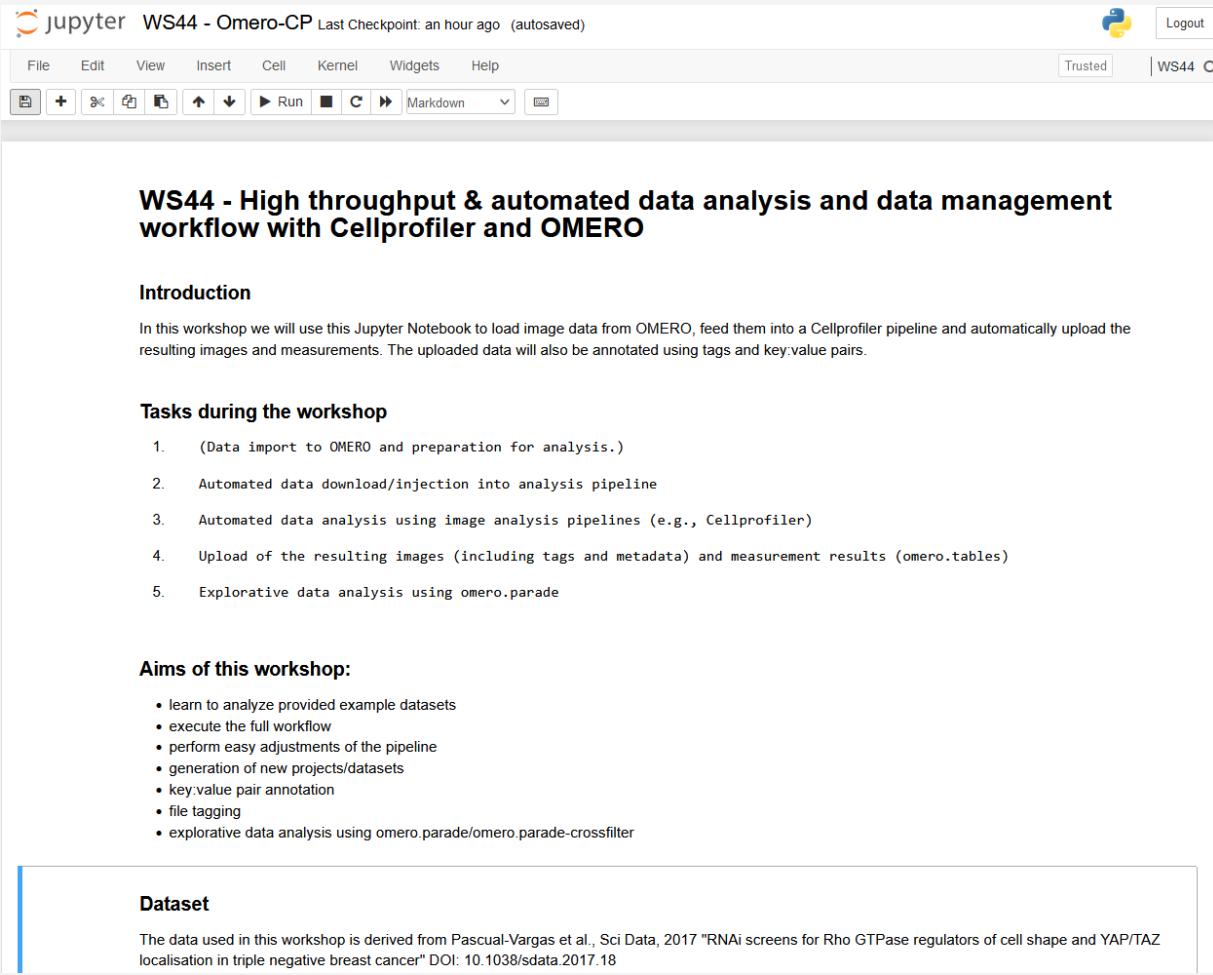


+ Measurement Results

4 Start the Jupyter Notebook

- Save the notebook under “C:\”

```
Start Jupyter
1 conda activate cellprofiler
2 jupyter notebook
```



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** Jupyter WS44 - Omero-CP Last Checkpoint: an hour ago (autosaved)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help.
- Header:** Trusted, WS44
- Main Content:**
 - Section Header:** WS44 - High throughput & automated data analysis and data management workflow with Cellprofiler and OMERO
 - Section:** Introduction

In this workshop we will use this Jupyter Notebook to load image data from OMERO, feed them into a Cellprofiler pipeline and automatically upload the resulting images and measurements. The uploaded data will also be annotated using tags and key:value pairs.
 - Section:** Tasks during the workshop
 - (Data import to OMERO and preparation for analysis.)
 - Automated data download/injection into analysis pipeline
 - Automated data analysis using image analysis pipelines (e.g., Cellprofiler)
 - Upload of the resulting images (including tags and metadata) and measurement results (omero.tables)
 - Explorative data analysis using omero.parade
 - Section:** Aims of this workshop:
 - learn to analyze provided example datasets
 - execute the full workflow
 - perform easy adjustments of the pipeline
 - generation of new projects/datasets
 - key:value pair annotation
 - file tagging
 - explorative data analysis using omero.parade/omero.parade-crossfilter
 - Section:** Dataset

The data used in this workshop is derived from Pascual-Vargas et al., Sci Data, 2017 "RNAi screens for Rho GTPase regulators of cell shape and YAP/TAZ localisation in triple negative breast cancer" DOI: 10.1038/sdata.2017.18

5 Insert OMERO login, OMERO-ids and file locations

& Run the notebook

Parameters

In the code block below, you will add specific analysis parameters, such as the screen and plate id, you would like to image, as well as filepaths and other settings.

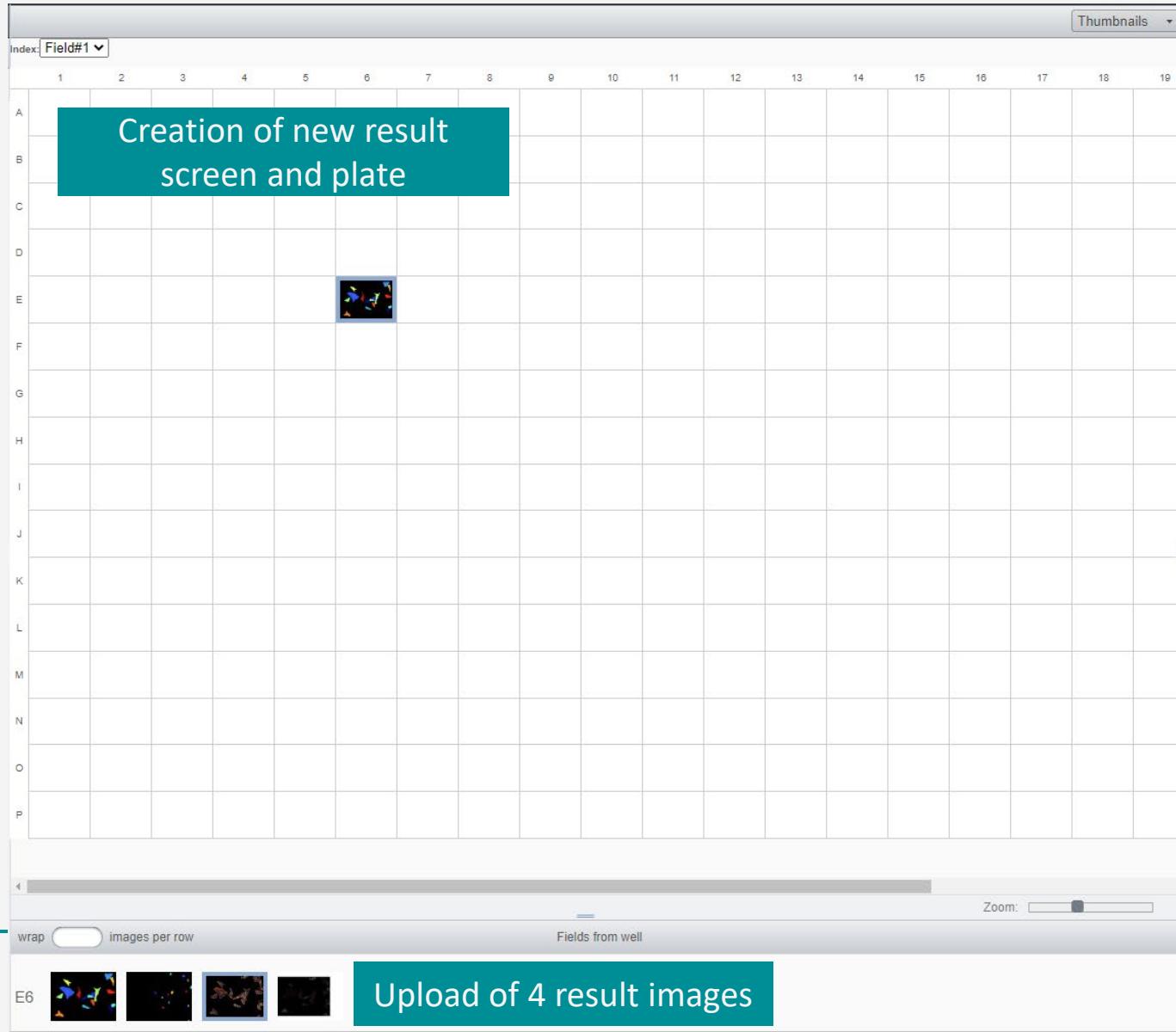
```
In [ ]:  
# Login to OMERO  
OMEROUSER = input(f"Enter username: \t")  
OMEROPASS = getpass.getpass(prompt = f"Enter password: \t")  
  
OMEROHOST = ''  
OMEROPORT = ''  
OMEROWEB = ''
```

```
In [ ]:  
# Connection Check:  
conn=ezomero.connect(OMEROUSER, OMEROPASS, "", host=OMEROHOST, port=OMEROPORT, secure=True)  
print(conn.isConnected())
```

```
In [ ]:  
# OMERO IDs  
screen_id = "#Insert ID of dataset that you want to analyse  
plate_id = "#Insert corresponding plate ID  
project_id = #Project ID for temp - dataset  
selected_well = "E6" # Insert well you want to analyse  
tag_owner_id = "# To keep the omero server clean, we will all use tags from 1 tag owner. Otherwise everyone would produce their own tags.  
  
# Pipeline  
pipe_dir = r"" #Insert directory of pipeline including name of pipeline  
  
# Input and saving directories:  
output_dir = "temp_dir"  
# if you want to use a temporary directory that is automatically created use: "output_dir = 'temp_dir'"  
  
# Cellprofiler-settings  
# (maybe remove)  
overwrite_results = 'Yes' # If yes, data present in the output folder will be overwritten  
output_file_format = 'tiff' # 'npy' for numpy array, 'tiff' for image (label images: 16-bit floating point)  
plugin_directory = ""  
  
# Name of the new dataset to which the label images will be uploaded  
new_plate_name = "Results_"  
append_original_plate_name = True # False  
  
# Specify the channels that should be used for segmentation and analysis  
# Same names as in CP pipeline!  
ch1 = "Nuclei" #Nuclei segmentation  
ch2 = "Actin" #Actin (cell body) segmentation  
ch3 = "Tubulin"  
ch4 = "YapTaz" #YapTaz for analysis  
# ... expand if you have more channel .. ch5 = xx  
  
channels = [ch1, ch2, ch3, ch4]
```

Output of the notebook:

Creation of new result screen and plate



Index: Field#1

Fields from well

E6 

Upload of 4 result images

Zoom:

wrap images per row

General Acquisition Preview OpenLink

Full viewer 

193062_Overlay.tif

Image ID: 290018
Owner: Sarah Weischer [Show all](#)

Image Details

Original Image: <https://omero-imaging.uni-muenster.de/webclient/?show=image-193062>

Import Date: 2023-03-13 12:01:21
Dimensions (XY): 667 x 501
Pixels Type: float
Pixels Size (XYZ) (µm): -
Z-sections/Timepoints: 1 x 1
Channels: 0, 1, 2
ROI Count: 0

Tags 1

Overlay

Key-Value Pairs 1

Add Key	Add Value
myns	
Added by:	Sarah Weischer
TIM23	WS44
Software	Cellprofiler 4.2.5
Segmentation Algorithm	Cellpose

Tables 2

- LM2_siGENOME_1A_CellprofilerResults_27feat (linked to Screen: 852)
- LM2_siGENOME_1A_CellprofilerResults_27feat_1ImagePWell (linked to Screen: 852)

Attachments 0

Comments 0

Ratings 0

Others 0

Link to original image

Tags based on image name

Key:value pairs

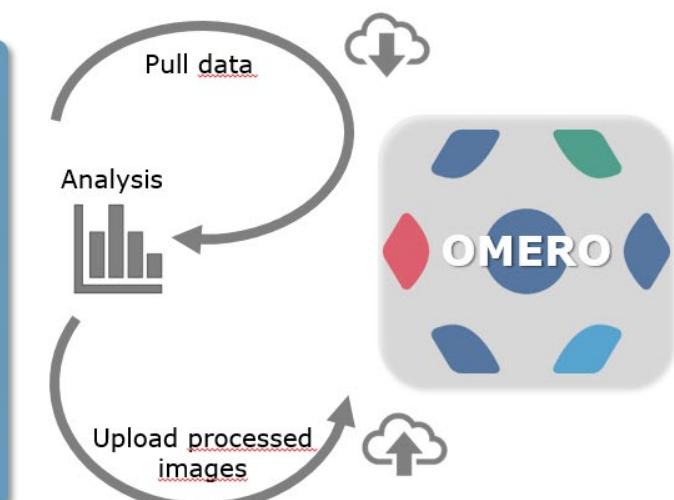
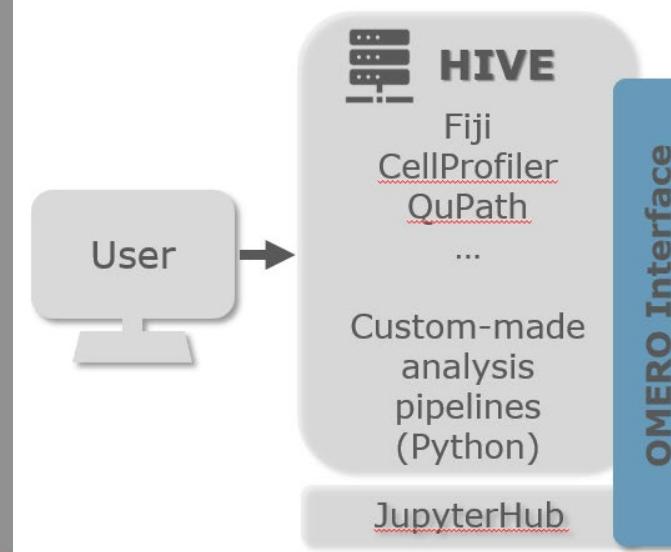
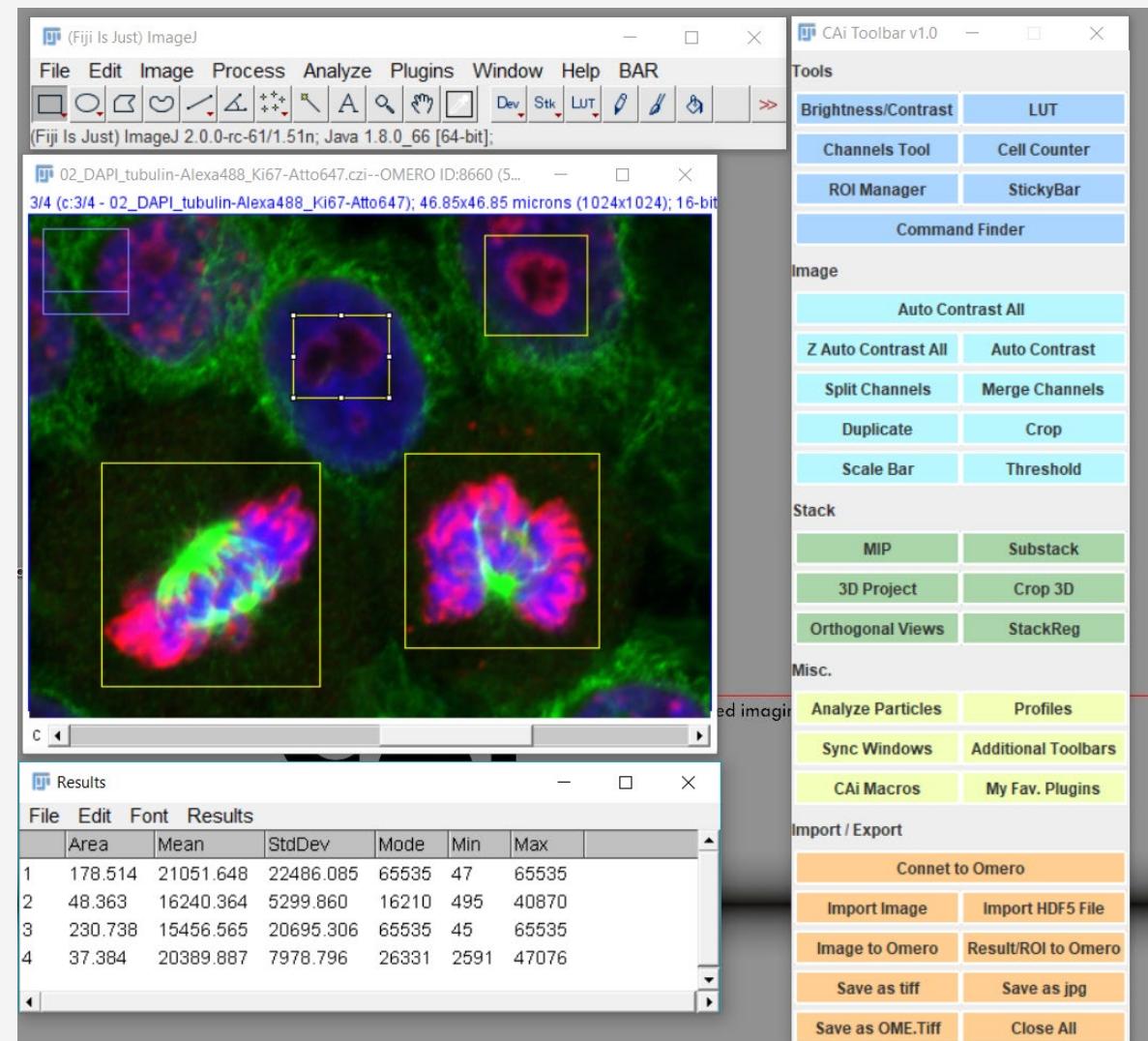
Analysis results as omero.table

Analysis pipeline uploaded as attachment to screen

https://github.com/MuensterImagingNetwork/TiM23_WS44_HTPImageAnalysis_Omero

OMERO – Fiji . Batch analysis plugin

OMERO Features – Analysis



Data Analysis with Fiji – Macros / Batch / Screens

OMERO Batch Plugin

Warning: all windows will be closed.

Connection
Connection status: **Disconnected**

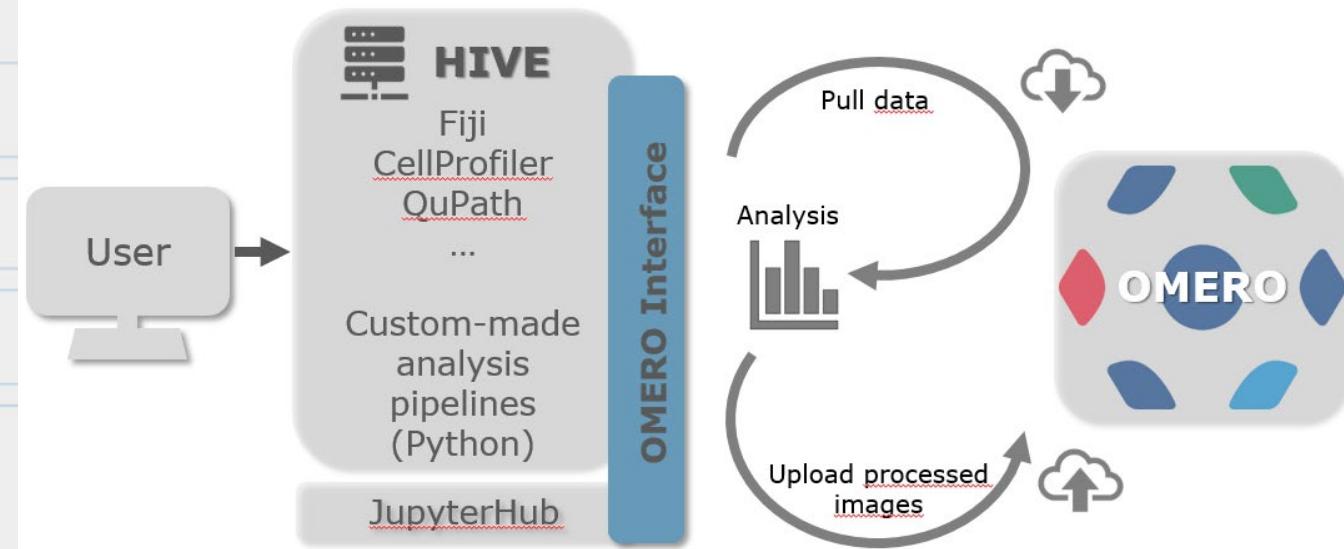
Source
Where to get images to analyse: OMERO Local

Input
Images folder: Recursive

Macro
Macro file:

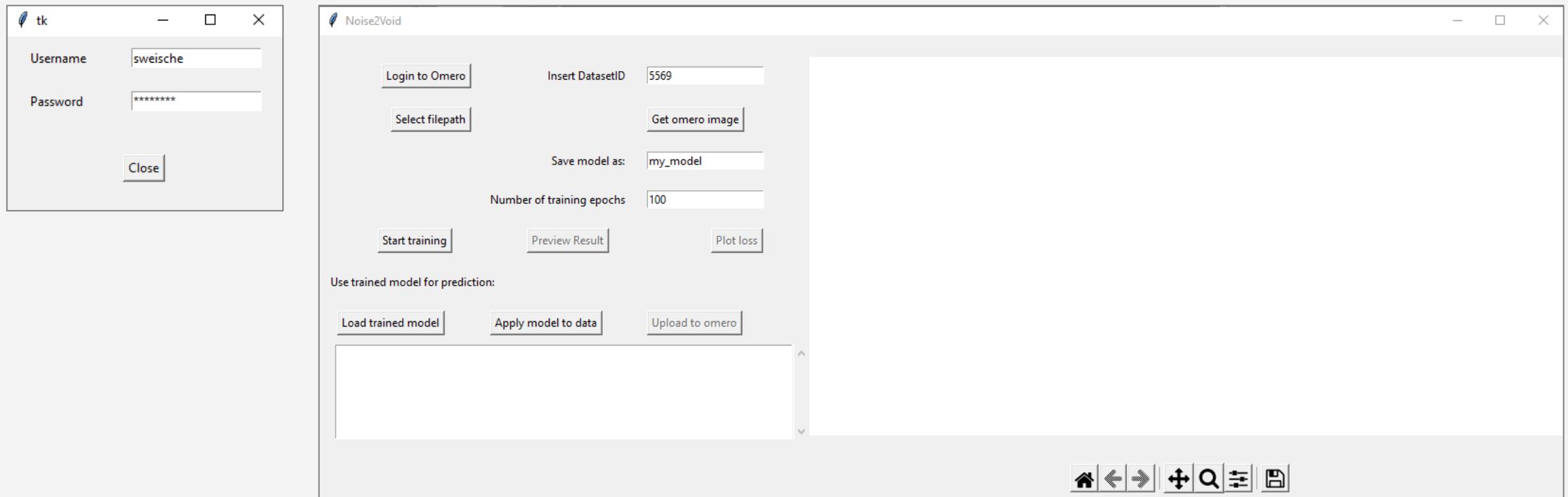
The macro returns:
 New image(s) Results table(s) ROIs Log file

Output
Where to save results: OMERO Local



Other Python scripts:

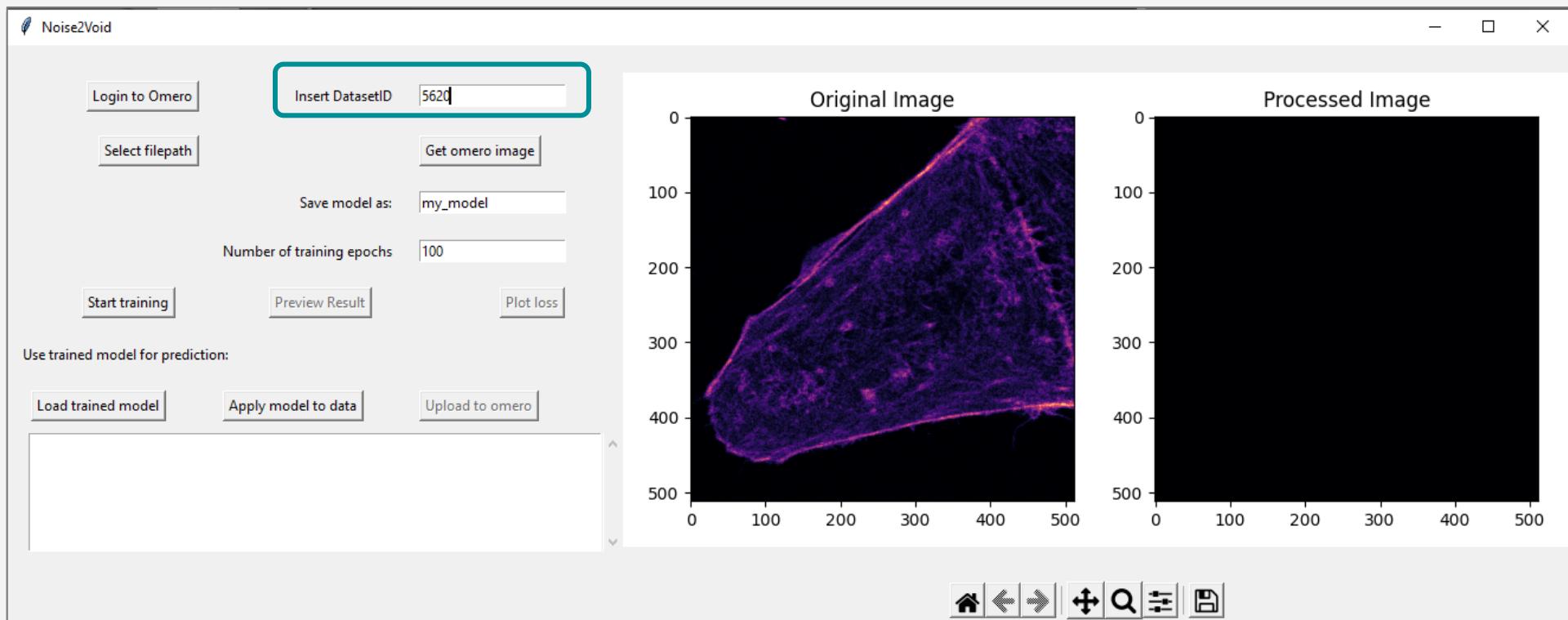
DeepLearning tool N2V using Python with automated OMERO download



runs remotely on Hive (analysis server)

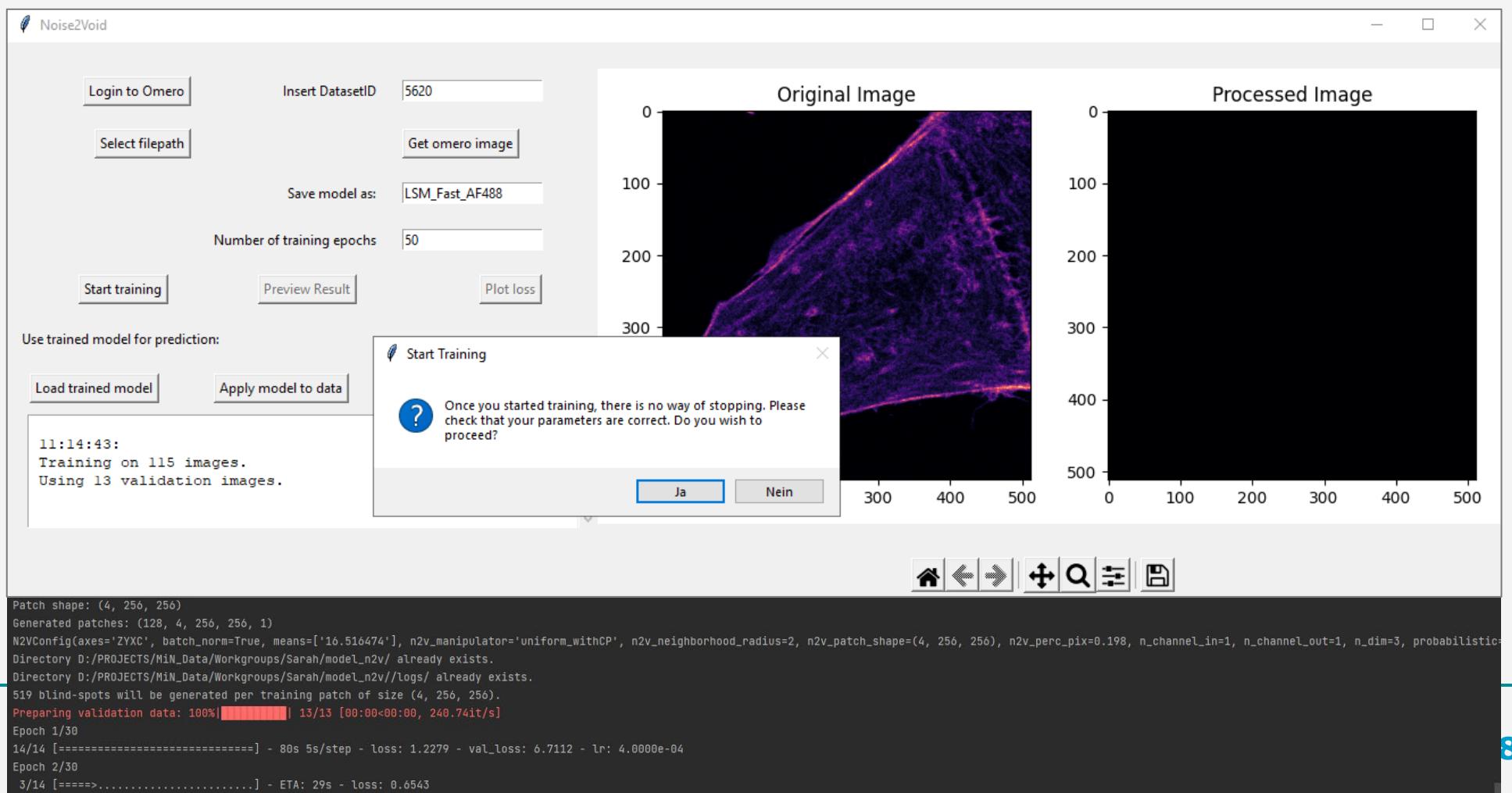
DeepLearning tool N2V using Python with automated OMERO download

- Insert DatasetID
- Retrieve images as np.array

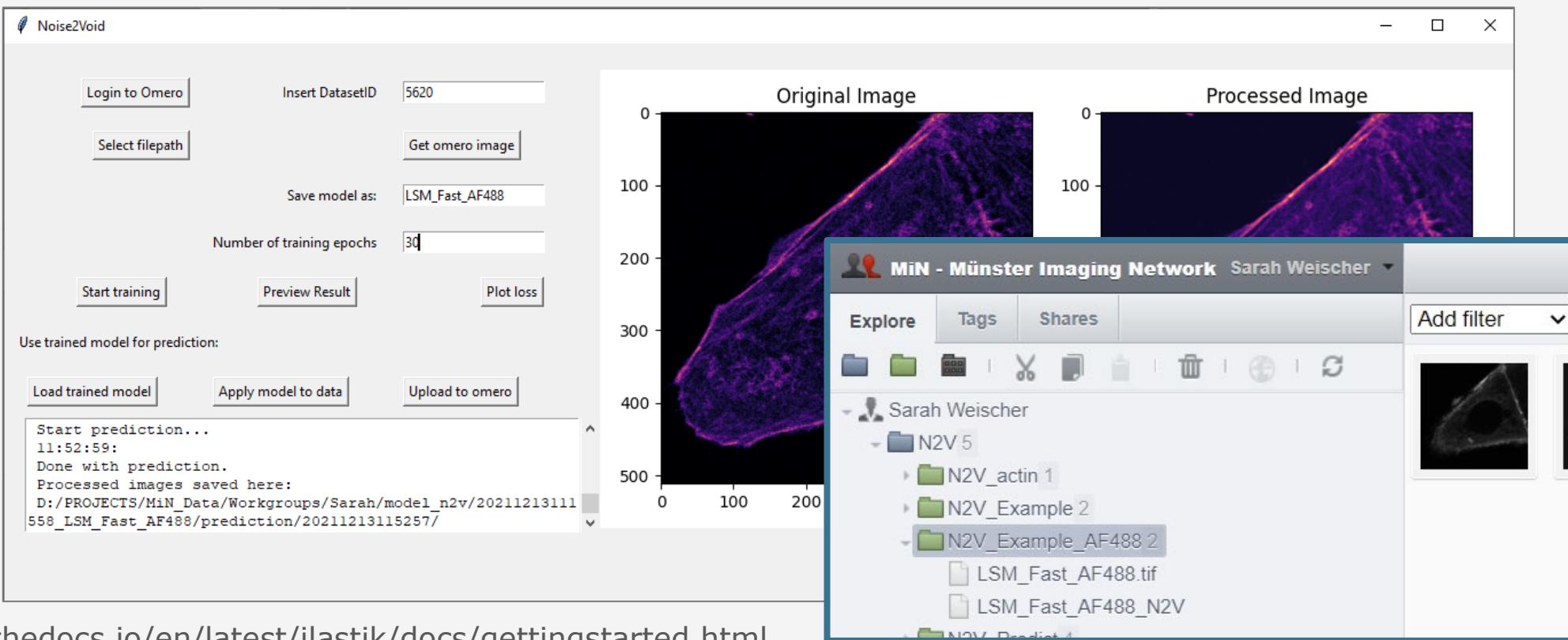


* Code adapted from:
<https://docs.openmicroscopy.org/omero/5.6.0/developers/Python.html#read-data>

DeepLearning tool N2V using Python with automated OMERO download



DeepLearning tool N2V using Python with automated OMERO download



Upload code adapted from:
<https://omero-guides.readthedocs.io/en/latest/ilastik/docs/gettingstarted.html>
`def save_results(...)`

Thanks

WS 44

Dr. Christian Schmidt
Coordinator NFDI4BIOIMAGE

Susanna Kunis
Uni-Osnabrück / CellNanOs

Dr. Markus Blank-Burian
WWU Münster / WWU-IT

Jens Wendt
Sarah Weischer

Frank Berkemeier / Holger Przybytzin
WWU Münster / ULB

Christian Döring
WWU Münster / CIM

OME-Team
University of Dundee

GerBi / RDM4Mic members
Tom Boissent HHU

