

Presentation available at

<https://downloads.openmicroscopy.org/presentations/2017/Cambridge-Training/Image-Processing-OMERO>

Analysis workflows: client side


Description

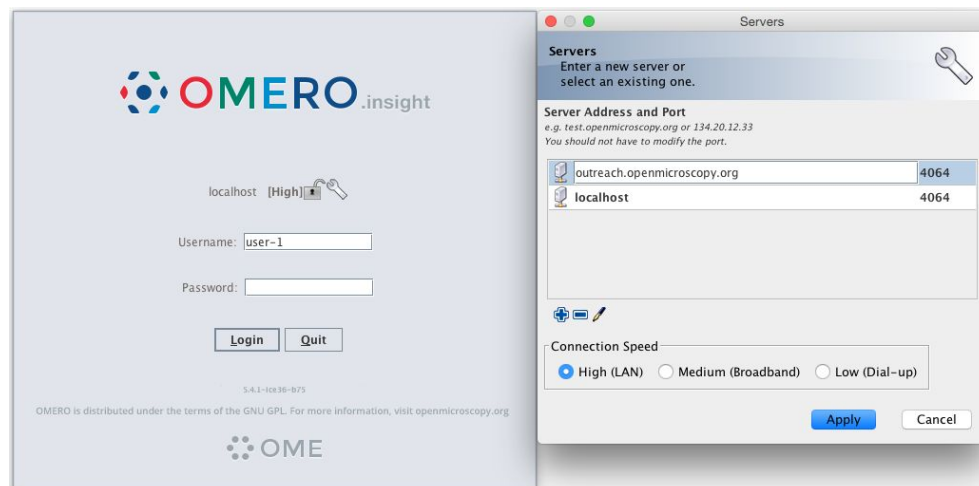
In the first part, we show how to manually analyse images stored in OMERO using existing plugins and in the second part, how to do the same analysis on the batch of images using the scripting facility in Fiji.

Setup

Fiji has been installed on the local machine with the OMERO.insight-ij plugin. The installation instructions can be found at <http://help.openmicroscopy.org/imagej.html#plugins> Installing the OMERO.plugin also adds the dependencies required to connect to OMERO using the script editor.

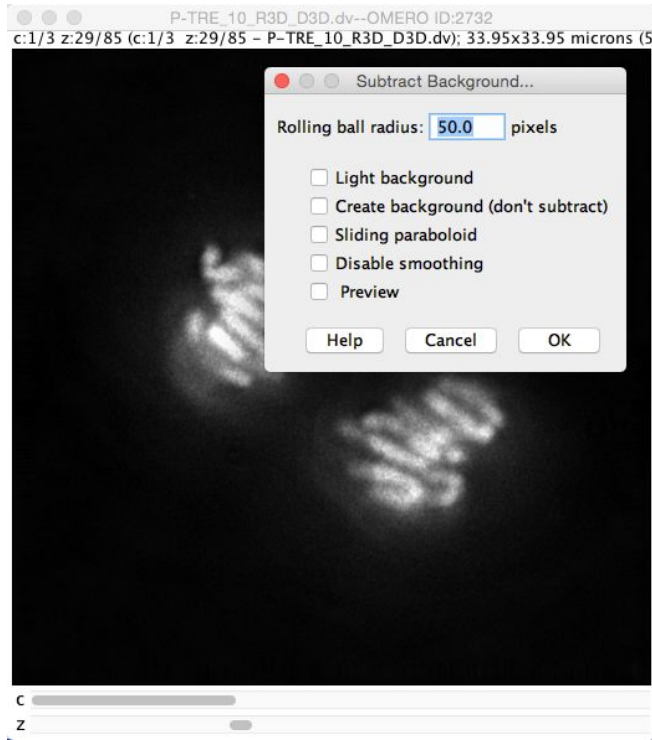
1. ImageJ/Fiji Manual workflow

- a. Launch Fiji
- b. To open an image from OMERO, Go to *Plugins > OMERO > Connect To OMERO*
- c. In the OMERO login dialog, click the wrench icon  and then add the server address in the dialog i.e. **outreach.openmicroscopy.org**. Click *Apply*.



- d.
- e. Enter your *Username* and *Password* and click *Login*.
- f. Browse to Dataset **PTRE** and double-click once on an Image to open it in Fiji
- g. N.B. Each plane will be transferred from the server to the client machine so this may take a few moments.

- h. Go to *Process > Enhance Contrast...* Use the default settings
- i. Go to *Process > Subtract Background...*
 - i. Set *Rolling ball radius: 50.0*.
 - ii. Do not select any checkboxes.
 - iii. Click *OK* and then *Yes*.



- j. *z*
- k. To save the generated image back to OMERO,
 - i. select *Plugins > OMERO > Save Image(s) to OMERO*.
 - ii. Create a new Dataset using the *New* button next to the dataset selection box.
 - iii. Enter a name you will recognise.
 - iv. Click *Add to Queue* then *Import*.
 - v. The image is saved as an OME-TIFF in the newly created dataset.

2. ImageJ/Fiji Scripting workflow

- a. Go to <http://outreach.openmicroscopy.org>
- b. Log in
- c. Select the Dataset ***Fiji-batch***.
- d. Note its ID on the right-hand panel, it will be used below



- e.
- f. Launch Fiji if not yet opened

- g. Go to *File > New > Script*
- h. A dialog pops up. In the *Language* menu select *Python*
- i. Copy the content of the local script **omero_bg_subtract_batch.py** into the text editor
- j. Edit the following parameters: dataset_id, USERNAME, PASSWORD.

```

62 # Setup
63 # -----
64
65 # OMERO Server details
66 HOST = "outreach.openmicroscopy.org"
67 PORT = 4064
68 group_id = "-1"
69 # parameters to edit
70 dataset_id = "1022"
71 USERNAME = "username"
72 PASSWORD = "passwd"
73
74 def open_image_plus(HOST, USERNAME, PASSWORD, PORT, group_id, image_id):
75     "Open the image using the Bio-Formats Importer"
76
77     options = ""
78     options += "location=[OMERO] open=[omero:server="
79     options += HOST
80     options += "\nuser="
81     options += USERNAME
82     options += "\nport="

```


Run Kill Show Errors Clear

Started New_.py at Thu Nov 30 11:28:33 GMT 2017

```

2732
P-TRE_10_R3D_D3D.dv.ome.tiff
uploading...
imported
2733
P-TRE_11_R3D_D3D.dv.ome.tiff
uploading...
imported
Done

```

- k.
 - l. Click *Run*
 - m. The script will process all images in the specified dataset, apply the same processing as in the manual workflow above and create new Images and add them to a new Dataset **script_editor_output_from_dataset_\$(dataset_id)** where **\$(dataset_id)** is the identifier of the dataset **Fiji-batch**
 - n. Return to the webclient and click the refresh button  above the left-hand panel to see the new Dataset.

Exercise 1: Time permitting, try to modify the script to add a tag annotation to the dataset using information on <https://docs.openmicroscopy.org/omero/5.4.1/developers/Java.html>

Exercise 2: Time permitting, modify the script to use another IJ macro e.g. crop a rectangular region so that a smaller image is imported into OMERO

Exercise 3: Using the example on <https://docs.openmicroscopy.org/omero/5.4.1/developers/Java.html>

Write a new script retrieving images contained in the Dataset **Fiji-batch**, load the thumbnail associated to each image and display the thumbnail in Fiji.

The thumbnail will have to be converted to grayscale in order to be viewable in Fiji.

Analysis workflow Server side

Description


In this section, we show how to analyse images using OMERO scripts.

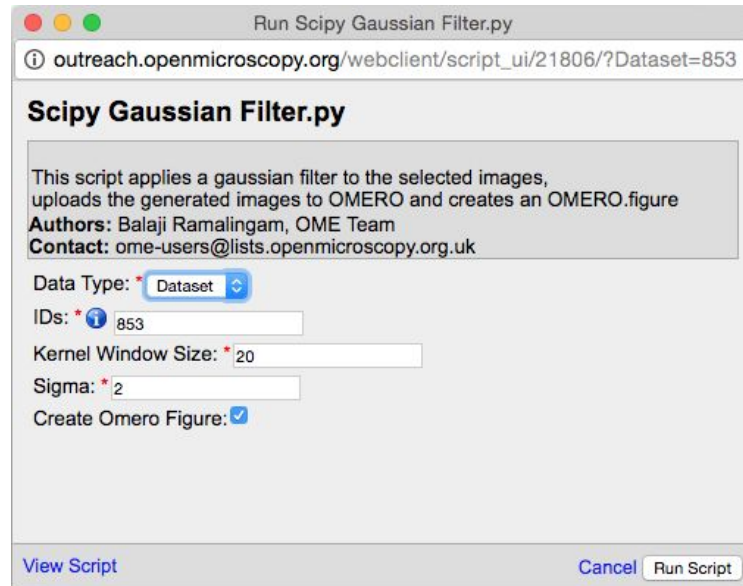
We show an example script generating some images and an OMERO.figure composed of the new images. The OMERO.figure will then be adjusted using the scripting facility in OMERO.figure.

Setup

An OMERO.script, *Scipy_Gaussian_Filter.py*, has been uploaded to the server.

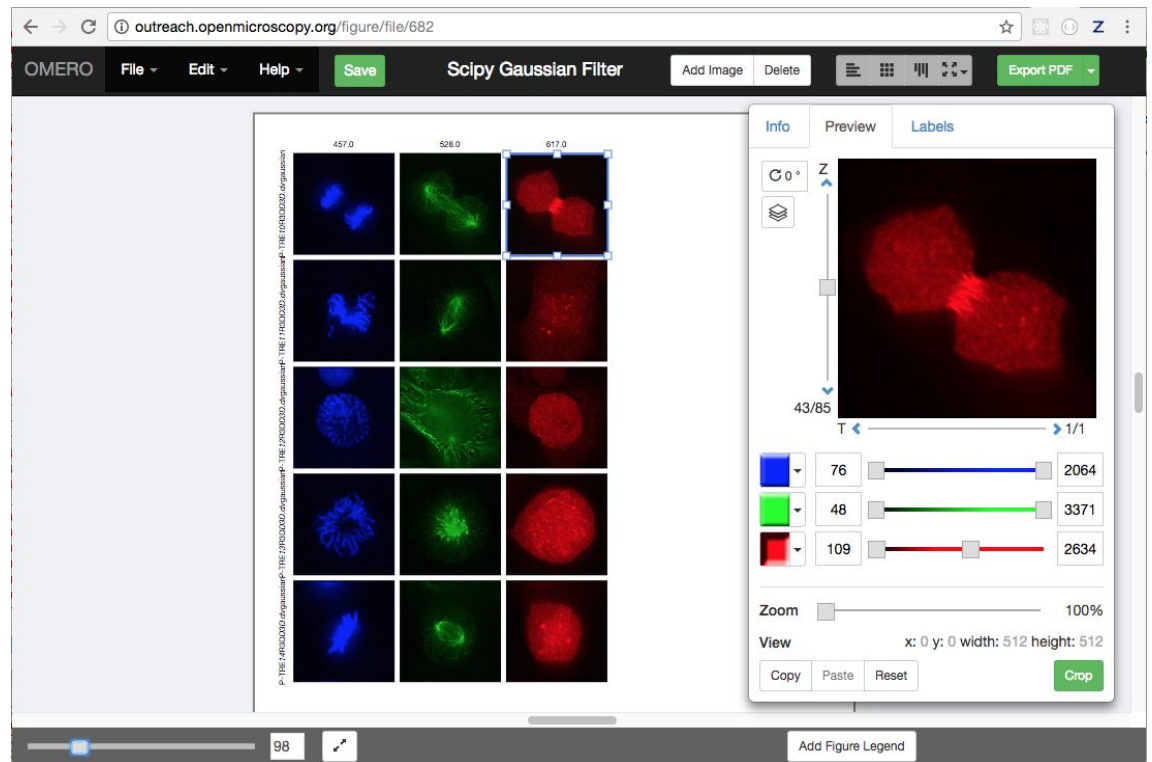
3. Process data using an OMERO.script

- a. Go to <http://outreach.openmicroscopy.org/> and login
- b. Select a Dataset, e.g. *Fiji-batch* or Image(s)
- c. Click the *Script* button in the top-right of the page 
- d. Click *cambridge_scripts* and then *Scipy Gaussian Filter....*



- e.
- f. Open the local copy of the *Scipy_Gaussian_Filter.py* script provided to follow the various steps.
- g. Scroll to the bottom of the script to see where the input parameters are defined, such as "Data_Type" and "IDs". Notice how these appear in the script dialog and are auto-populated with the currently-selected Dataset or Images.
- h. The script will process all the Images in the selected Dataset, create Images in a new Dataset and generate an OMERO.figure file that contains the new images.
- i. Click *Run Script*.

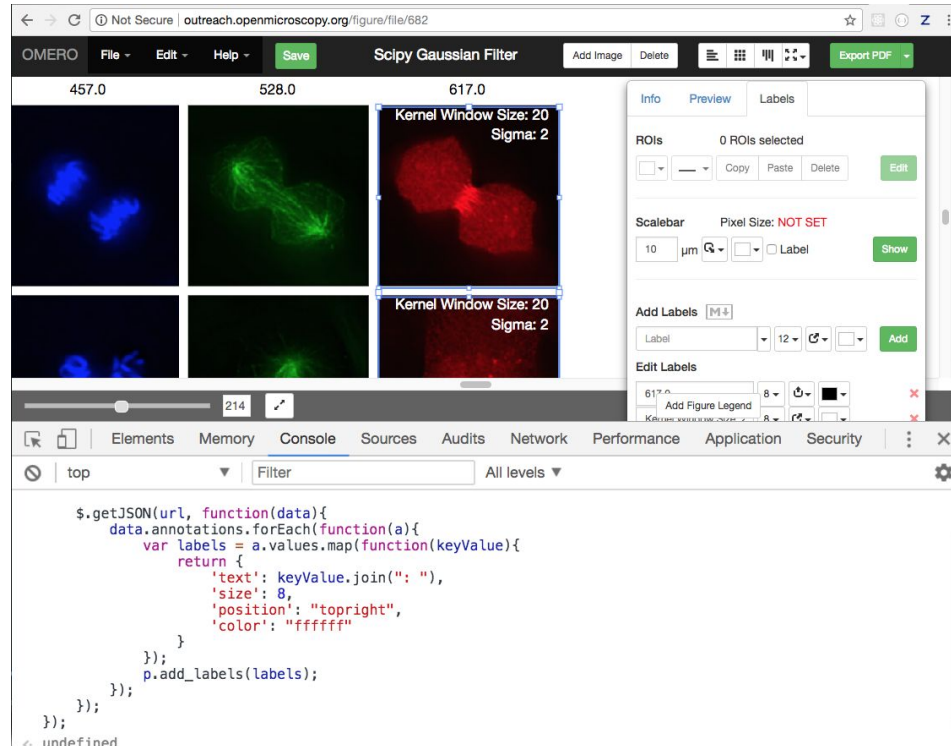
- j. When the script has completed, you will be able to browse to the new Dataset.
- k. To see the new Figure, click on the *Figure* link at the top of the page and choose *Open File*. Then pick the *Scipy Gaussian Filter* from the list.



l.

4. OMERO.Figure manipulation

- a. We can use the JavaScript console to script some figure updates, loading JSON data from OMERO via AJAX. We will create labels from Key-Value pairs (map annotations) that were added to the images by the *Scipy_Gaussian_Filter.py* script.
- b. Open the generated figure.
- c. Right-click on the page and choose *Inspect Element* to open devtools.
- d. Choose the *Console* tab.
- e. Copy the content of the local file *figure_labels_from_mapanns.js*
- f. Click or drag on the figure to select one or more image panels.
- g. Paste the code in the console. N.B. you can edit the *size*, *position* and *color* values in the code before running.



- h.
- i. Other examples are provided, named *figure_...js*, which can be used in the same way as above.
- j. NB: figureModel is a Backbone.js Model and the panels are a Backbone.js Collection. To inspect the current model, you can use File > Export as json.

Analysis using CellProfiler and R

Description

In this section, we will use Jupyter notebooks. See <http://jupyter.org/>

A Jupyter server has been set up to simplify the installation of the dependencies required by some of the scripts.

This will allow you to use the Python API and rOMERO-gateway and it will also show how 3rd party applications can be integrated.

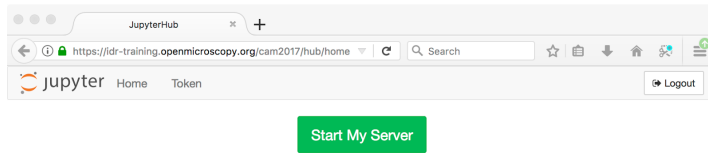
Setup

A Jupyter server has been set up. You will first need to log-in in order to access the notebooks

5. Jupyter connection

- a. To connect to the Jupyter server go to <https://idr-training.openmicroscopy.org/cam2017>
- b. Enter the username provided
- c. Enter the password provided

- d. When logged in, click on *Start My Server*



e.

6. OMERO hello world

1. In this section we will use examples from <https://docs.openmicroscopy.org/omero/5.4.1/developers/Python.html> and show how to manipulate data stored in an OMERO.server
2. Click on notebooks > python > OMERHelloWorldNotebook.ipynb
3. You need to edit the USERNAME and PASSWORD variables with your details.
4. In this example, we simply connect to OMERO and list the first 5 Projects and their Datasets.

5. Start at the top, select each step in turn and click Run:



6. When you see the Projects, Datasets and Images printed, add another step using the + button and use the docs link above to add code to:

- a. Get an Image using `conn.getObject()` and the image ID (choose from printed list above) and print the `sizeX`, `sizeY`, `sizeZ`, `sizeC`, `sizeT`.
- b. Get plane as numpy 2D array using `getPlane()`, print min & max values and display plane using `import matplotlib.pyplot as plt`, `%matplotlib inline`, and `plt.imshow(plane, cmap='gray')`

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.imshow(plane, cmap='gray')
```

7. Analyse data using CellProfiler notebook

- a. In this notebook, we will analyse data stored in OMERO in the dataset *CellProfiler images* using CellProfiler and store the analysis results back to OMERO. Python is used

8. Analyse output generated by CellProfiler using R notebook

- a. In this notebook, we will use the rOMERO-gateway to plot the results generated by CellProfiler as csv files above and store the general graph back to OMERO as a png.