

Presentations are available at

<https://downloads.openmicroscopy.org/presentations/2018/TOPIM-Crete/>

## TOPIM-2018 schedule

Thursday 12th July:

- 19:40 - 20:15 (35 mins) OMERO workshop introduction

Friday 13th July: Workshop, Track II

- 11:20 - 12:55 (1 hour 35 mins)
  - Introduction to OMERO. Data Management & Viewing images
  - CellProfiler analysis of HCS Data - quick viz in OMERO.parade
- 13:25 - 14:10 (45 mins)
  - Image analysis in Orbit, saving ROIs back to OMERO
  - Using the OMERO Java API in Orbit
- Lunch
- 18:00 - 18:45 (45 mins)
  - Fiji analysis -> ROIs
  - Python OMERO.script. ROIs -> OMERO.tables
  - R analysis of OMERO.table data
- 19:10 - 20:15 (1 hour 5 mins)
  - OMERO.parade
  - OMERO.figure

## Session 1: Introduction to OMERO & CellProfiler

### OMERO core concepts

#### Description:

We introduce some key features of the OMERO platform using the webclient.

#### Import Images (for information only)

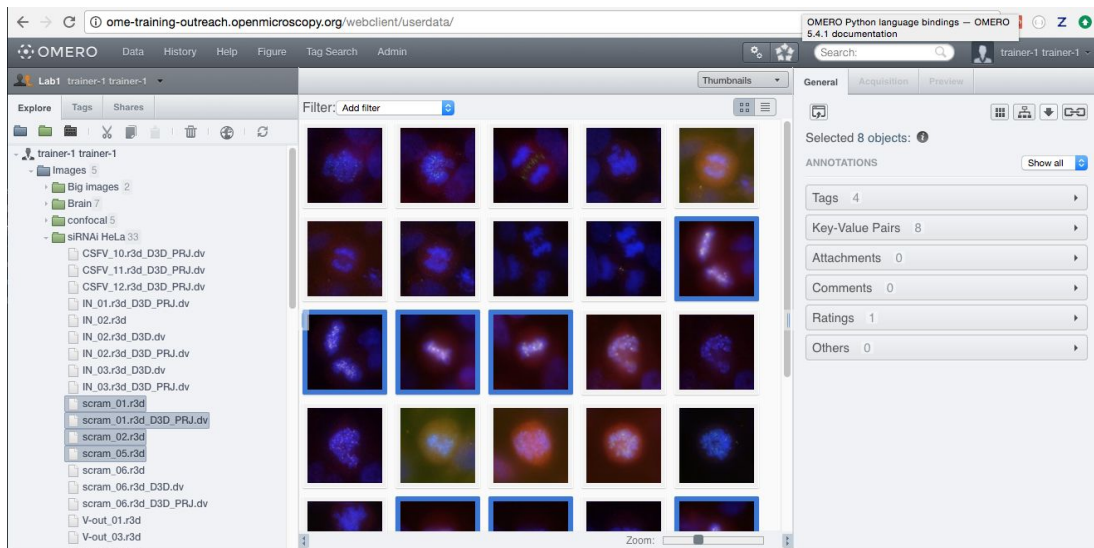
For this workshop, images have already been imported into OMERO.

1. Import using the Desktop Application OMERO.insight
  - a. you will need to have Java version 1.7 or higher installed.
  - b. Download OMERO.insight from  
<https://www.openmicroscopy.org/omero/downloads/>
  - c. The import instructions can be found at  
<https://help.openmicroscopy.org/importing-data-5.html>

2. Import using the Command Line tool
  - a. Installation instructions are available from <https://docs.openmicroscopy.org/latest/omero/users/cli/installation.html>
  - b. The import instructions can be found at <https://docs.openmicroscopy.org/latest/omero/users/cli/import.html>

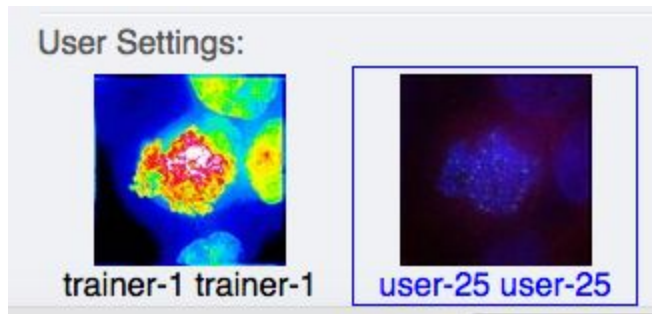
## Browse Images

1. Go to <http://outreach.openmicroscopy.org/>
2. Log in using the username and password provided.
3. You can browse 'folders' in the left-hand panel: *Image* folders are called *Datasets* and they are within *Projects*.
4. When a Dataset is selected, Image thumbnails are shown in the centre panel.



5. These represent imported Images: The original Images are stored on the server and generated thumbnails allow us to browse them.
6. Bio-Formats, <https://www.openmicroscopy.org/bio-formats/>, is used to read pixel-data and metadata from over 150 different image formats, including multi-z timelapse images with many channels, they are referenced as 5D Images. Large pathology and Medical images are also supported.
7. Select an Image. In the right-hand panel, metadata read by Bio-Formats and stored in a relational database is displayed:
  - a. core metadata in the General tab
  - b. additional metadata in the Acquisition tab
8. In the Preview tab in the right-hand panel, you can also view the Image.
9. For multi-plane images, sliders allow you to move through Z or Time dimensions.
10. Viewing Images **DOES NOT** download the whole Image to the client. Only the viewed Image plane is rendered from the original Image file on the server and sent back to the client.


11. You can adjust the rendering settings for each channel e.g. turn on/off the channels, adjust color settings, look-up tables, etc..
12. The rendering settings can be saved to the server. This **NEVER** changes the original Image data and can be reverted at any time.
13. The rendering settings can also be copied and pasted between Images or *Save to All* to apply the same settings to all Images in a given Dataset.
14. You can use settings which other users saved on your images and apply them for yourself. These settings are highlighted as thumbnails in the lower part of the Preview



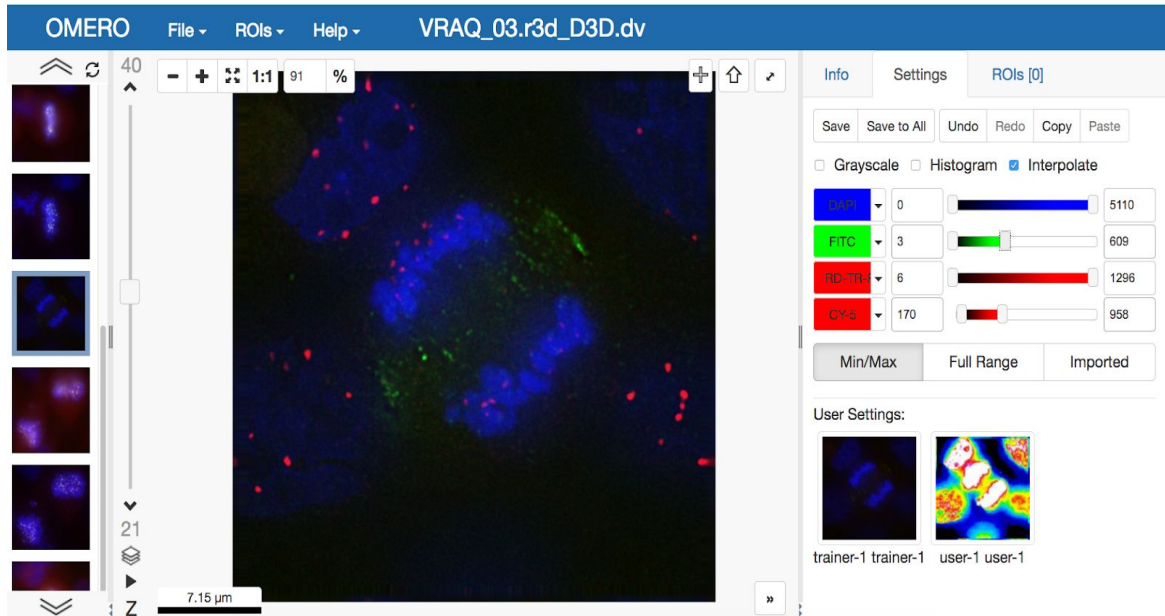
pane.


15. Your own settings are highlighted in blue.
16. You can revert to the original settings for an Image or Dataset. For example, using the context menu for a Dataset in the tree. *Rendering Settings > Set Imported and Save*.

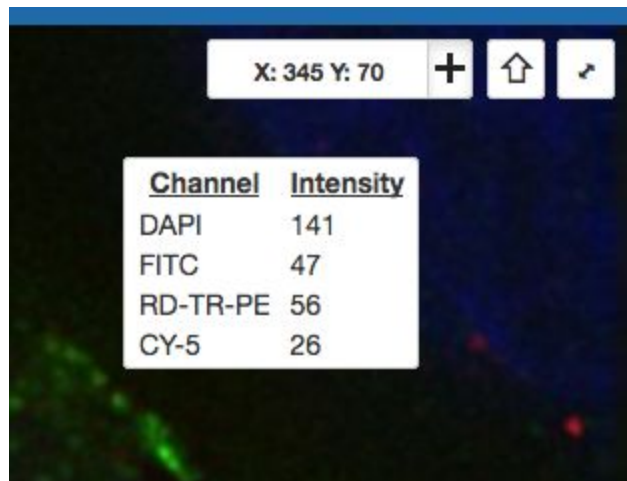
## View Images

1. Double-click on an Image thumbnail, or click the *Full Viewer*  button in the right-hand pane to open an Image in a larger viewer called OMERO.iviewer. It is a Web app developed and released independently from the webclient. The OMERO.web framework can be extended with multiple apps to view data in different ways. We will demonstrate some of them today.


- We can adjust the rendering settings and scroll through T and Z if it is a timelapse and/or a Z-stack image.




- Click the *Crosshairs* icon  at the top right of the viewer to enable pixel intensity display for the mouse pointer. Then mouse over the Image to see the pixel intensities



for the enabled channels.

- The ROIs tab  includes tools for viewing and drawing ROIs on the Image. These are saved back to the OMERO server. Select the *arrow tool* in the right-hand

pane  and draw an arrow on the Image, using Click-Move-Click (not drag),


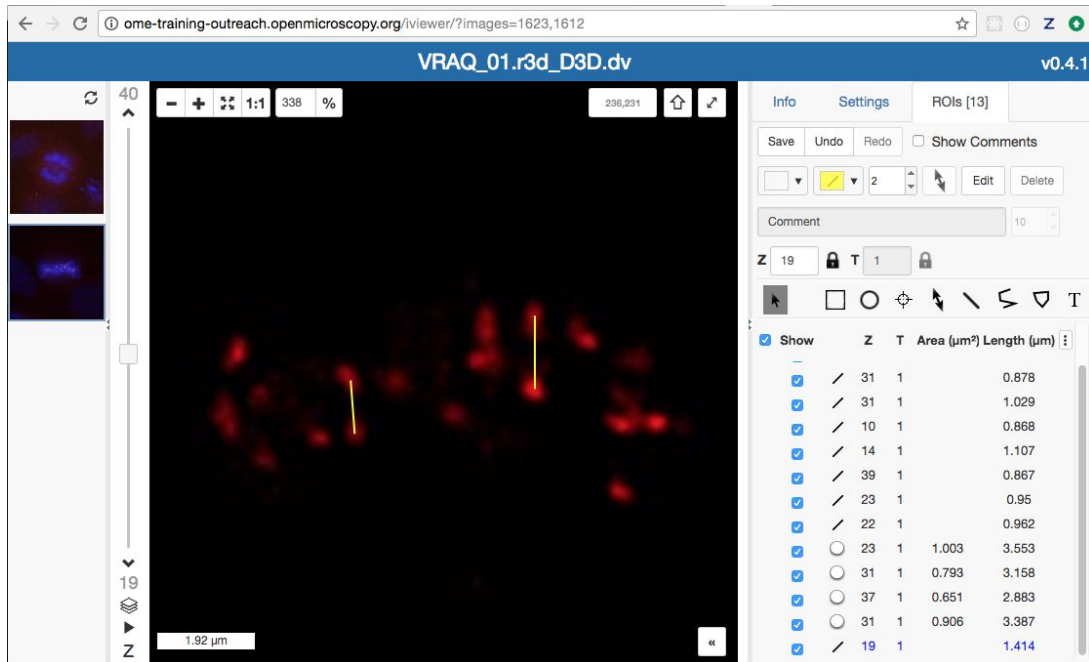
pointing to a feature in the Image and save it using the *Save button*  located in the upper part of the right-hand pane. Draw also a couple of other ROIs on different Z planes. Dragging on the Image is used for panning the image. Shift-drag rotates the

image (using selection tool) or draws “freehand” (polygon, polyline, rectangle & ellipse).

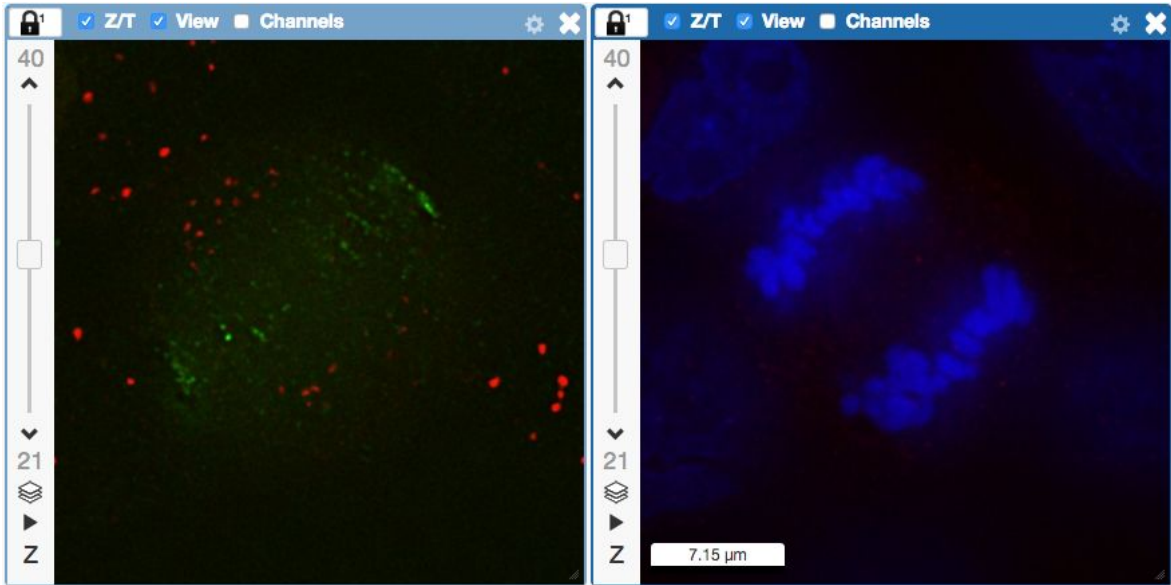



- Clicking onto the ROIs in the right-hand pane table brings the viewport to the position where the selected ROI is in the middle of it and navigates the viewport to the timepoint the ROI is associated with.

| <input checked="" type="checkbox"/> Show | Z | T  | C | Comment |
|------------------------------------------|---|----|---|---------|
| <input checked="" type="checkbox"/>      | 1 | 1  |   |         |
| <input checked="" type="checkbox"/>      | 1 | 1  |   |         |
| <input checked="" type="checkbox"/>      | 1 | 8  |   |         |
| <input checked="" type="checkbox"/>      | 1 | 11 |   |         |
| <input checked="" type="checkbox"/>      | 1 | 15 |   |         |

- Select several ROIs from different Z planes. You can select in the table (shift-click to select range) but also in the image itself using Cmd-drag to select multiple ROIs.
- Export the Intensities, areas and line lengths into Excel. Select, in the top-left corner *ROIs > Export (Excel)*. This will export the values from the selected ROIs.
- Save the viewport by selecting in the top-left corner *File > Save Viewport as PNG*
- Compare the two channels of one of the Images in multi-image view. Double-click on the thumbnail of the already opened Image in the left-hand pane of the viewer. This will open the Image again in a new window. Select one of the windows and in the right-hand pane switch the green channel off. On the other window, switch the blue

channel off.



10. We can synchronize the viewers by adding them both to the same sync Group. Click the *sync icon*  in the top-left corner and select the first option in the dropdown menu. Repeat for the other viewer. Now both Images have the *Z/T* and *View* (zoom and position of viewport) synchronized.



### Add Annotations, Filtering and Searching

1. Let's now add annotations to Images in the webclient.
2. We will add Tags to indicate Metaphase stages of these cells.
3. Select 1 or more images of cells which appear to be in metaphase to tag.
4. Choose the *Tag* option in the right-hand General tab and click [ + ] to launch the Tag dialog.
5. Choose the existing **Metaphase** from the list of Tags (type to filter above the list).
6. Click > to move it to the right column, then click OK.
7. Select an Image from the Dataset and in the right-hand *General* tab, click the harmonica *Key-Value Pairs* in the right-hand pane.



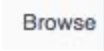
Added by: user-40 user-40

| Key         | Value        |
|-------------|--------------|
| mitomycin-A | 0mM          |
| PBS         | 10mM         |
| incubation  | 10min        |
| temperature | 37           |
| Organism    | Homo sapiens |

- 8.
9. The *Key-Value Pairs* allow you to add lab-book like additional metadata for the Image. These *Key-Value Pairs* are also specifically searchable.
10. It is also possible to add Comment, Attachment and Rating to selected Images.
11. Enter **mitomycin-A** into the search box in the top right of the webclient

and press Enter.

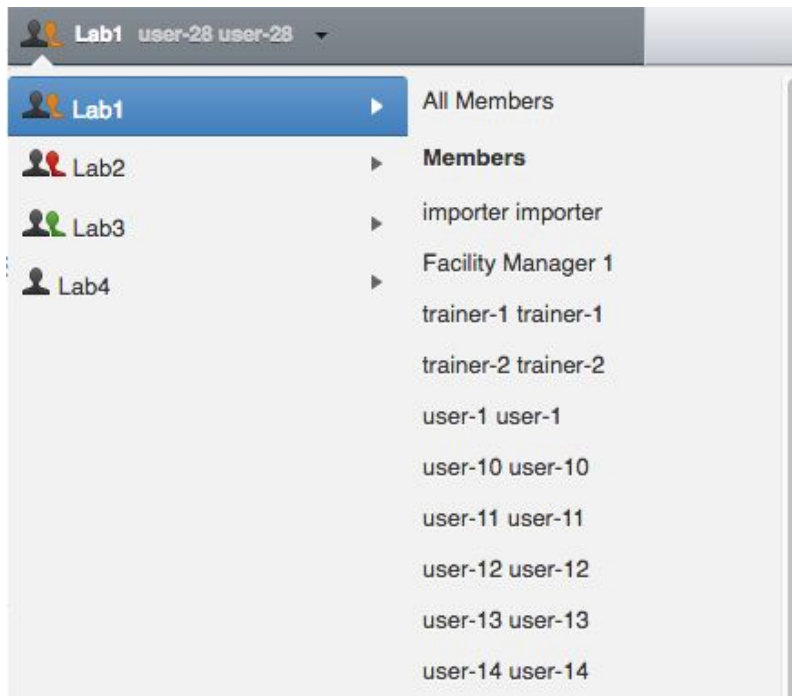
12. The search results will show any objects e.g. Images or Datasets which have anywhere the string **mitomycin-A**. Several images should be found, from 2 different datasets..
13. Refine the search now for only *Key-Value Pairs* which have the key **mitomycin-A** and value **0mM** by entering **mitomycin-A:0mM** into the search box and pressing Enter.

Now you should find only 21 images. Click on the *Browse* link  in one search result line of the last image (in the right-hand part of the centre pane) to navigate back to the main webclient.

14. Images can also be filtered by Tag or Rating in the centre pane, using the *Add filter* chooser above the thumbnails.
15. For example, choose Tag and then select Metaphase from the list of Tags to show the images we tagged earlier.
16. Review the filtered Images, choose a favourite Image and under the *Rating* section in the right-hand pane, click on the 5th star to add a rating of 5

17. Now we can remove the filtering by Tag and instead filter by *Rating of 5* to show only our favourite images.
18. OMERO offers various levels of permissions for groups and users. Depending on the permissions level of each group, a given user might be able to view, annotate or edit data belonging to other users. Permissions are managed by users with admin privileges. For this training session, we will work in a “Read-Annotate” group. This implies that users can view and annotate each other’s data but **cannot** delete other people’s data.

19. To see other people's data, click on the *Lab1* group name in the top left corner of the webclient. Then select the name of the user you wish to see the data of from the menu.



## Analysis with CellProfiler

### Description:

**CellProfiler** is a free open-source software for quantitative analysis of biological images.

For more details, go to <http://cellprofiler.org/>.

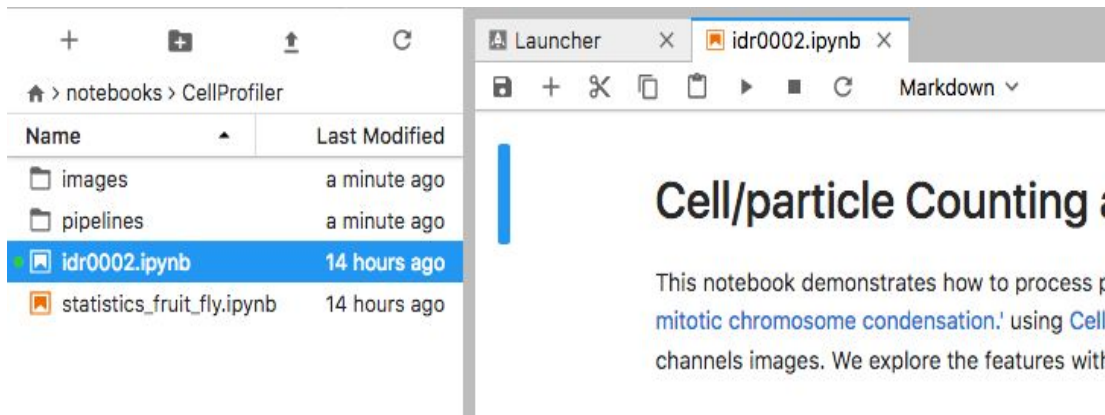
### Setup:

CellProfiler version 3.1.3 has been installed in a Jupyter notebook using docker as described at <https://github.com/ome/training-notebooks>.

1. We will use CellProfiler to analyse a Plate of Images in OMERO. We will run CellProfiler version 3.1.3 "headless" via a Jupyter Python notebook.
2. We will use the example pipeline from <http://cellprofiler.org/examples/#PercentPositive> to analyse RNAi screening data from IDR <http://idr.openmicroscopy.org/webclient/?show=screen-102>
3. First, open the webclient and find the Plate belonging to *trainer-1* named *plate1\_1\_013*.



- Go to <https://idr-analysis.openmicroscopy.org/training/hub/tmplogin> and look under Notebooks > CellProfiler for *idr0002.ipynb*.



- Select the first Step and click on the *Run* button to execute each step in turn.
- For the connection to OMERO, you will need to enter your login details before running the OMERO credentials step.

## OMERO Credentials

```
HOST = 'outreach.openmicroscopy.org'
# To be modified
USERNAME = 'username'
PASSWORD = 'password'
```

- 
- Select the plate in the webclient, find the Plate ID in the right-hand panel and copy this into the *plate\_id* variable in the next step of the notebook.

## Fetch OMERO Plate

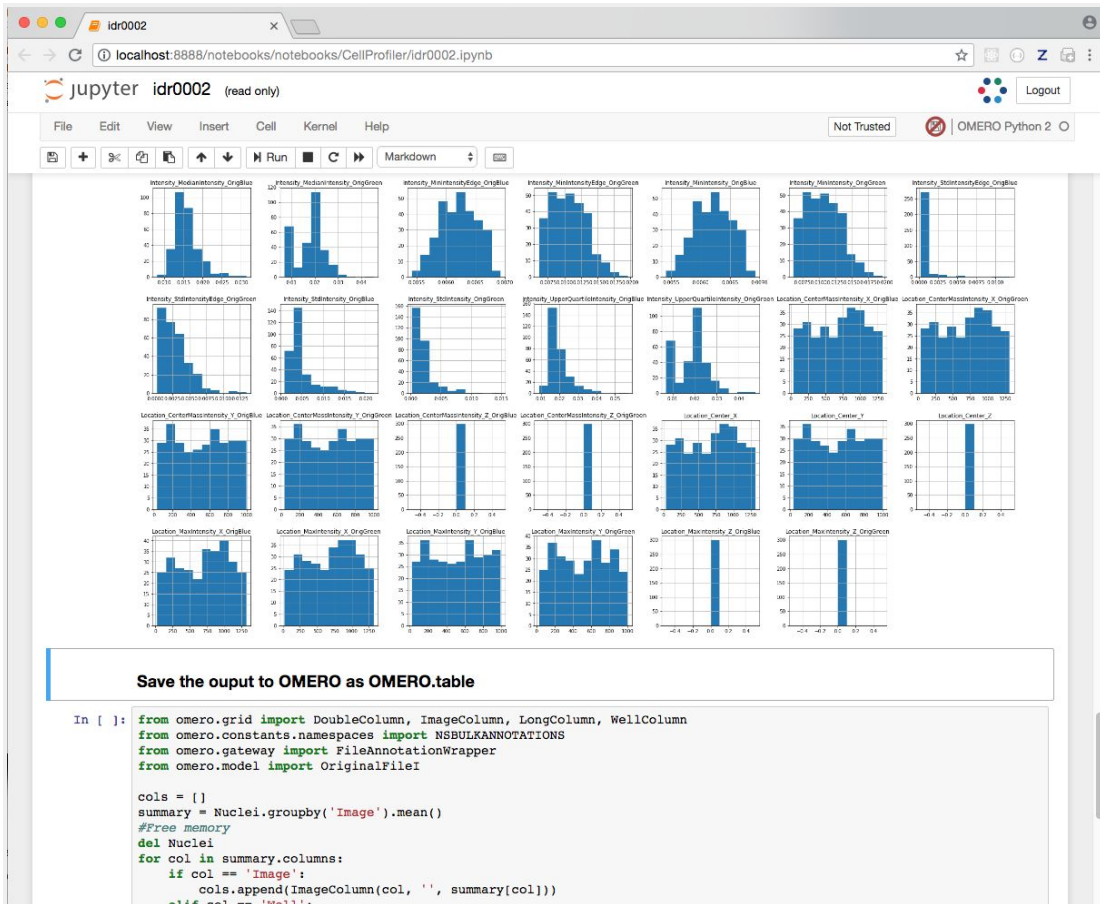
```
[ ]: # To be modified
plate_id = 422
```

- The following step loads the example pipeline and modifies it to remove the modules that are normally used for loading images from disk.
- These modules are replaced by InjectImage modules, using numpy planes loaded from OMERO Images.
- The pipeline is run on a 2-Channel image from each Well in the 96-well plate, generating a CSV file containing rows for different objects identified in the image and columns for various parameters measured.
- NB: To save time in the workshop, we can run on a subset of all Wells in the plate. To use the first 5 wells, add this line `wells = wells[0:5]` as shown:

```
# Create list from generator
wells = list(plate.listChildren())
wells = wells[0:5]
well_count = len(wells)
```

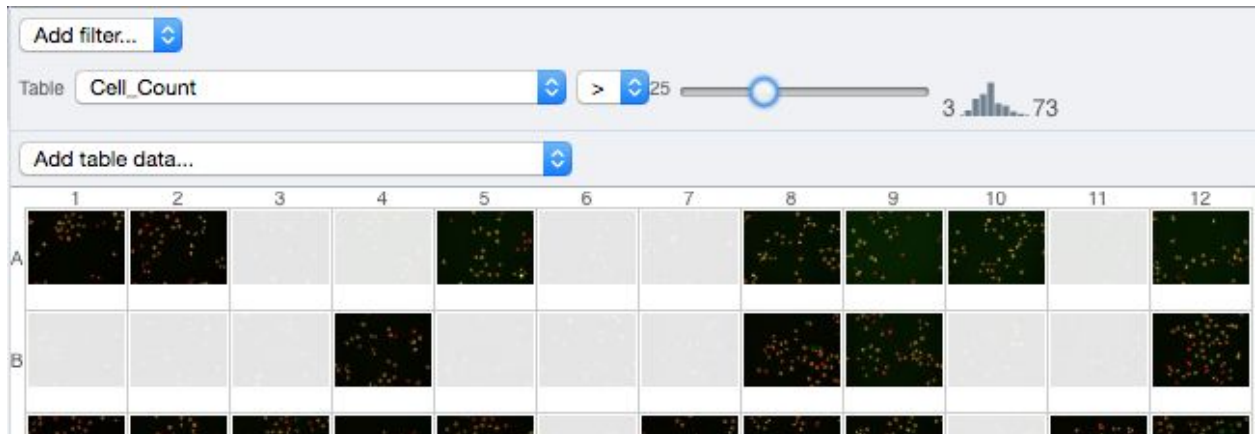
-

14. CSV files are read into a Dataframe for each image. We add the Image ID and Well ID, as well as the total number of Objects, Cell\_Count, to each Dataframe.
15. All the Dataframes are then concatenated into a single Dataframe.
16. We can visualise the data as histograms for each column with df.hist()



17. Finally, the Dataframe rows are grouped by Image to give an average value per Image of each parameter (column) in the table
18. This data is saved back to OMERO as an HDF5-based table attached to the Plate, which can be read by other clients.
19. Return to the webclient and select the Plate.
20. Select a Well and open the *Tables* pane in the General tab in the right-hand panel. This will show all the CellProfiler values for this Well.
21. In the *Thumbnails* chooser at the top-right of the centre panel, select the *Parade* plugin.
22. At the top-left of the centre panel choose *Add filter... -> Table* to filter Wells by the data from CellProfiler.
23. Change the filter from *ImageNumber* to *Cell\_Count* (at the bottom of the list).

24. Now you can use a slider to filter Wells by Cell Count.



## Session 2: Orbit as an OMERO client

### Description:

**Orbit Image Analysis** is a free open-source software with the focus to quantify big images like whole slide scans. It offers sophisticated image analysis algorithms. For more details, go to <http://www.orbit.bio/>.

### Setup

- Orbit has been installed on the local machine.

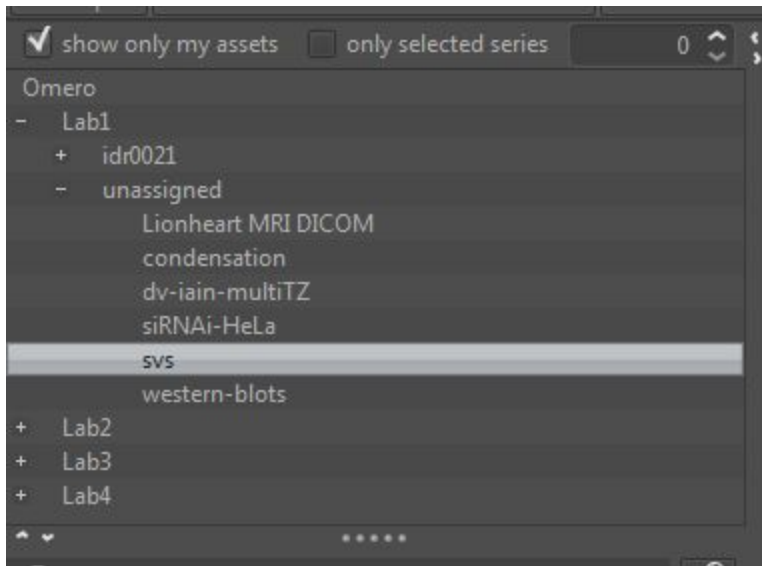
### Orbit workflow: Manual training and segmentation

1. Orbit uses a file named OrbitOmero.properties in user's home directory as described in <http://www.orbit.bio/omero-connection/>. This includes the server address we want to connect to. The port number is by default 4064 and thus does not need to be changed.
2. Configuring Orbit:
  - On Windows, you can use any good text editor e.g. notepad++ to open the OrbitOmero.properties.template. Make sure that the OrbitOmero.properties file which you save the template as after editing does not get an unwanted extension such as OrbitOmero.properties.txt. You can rename the file on the command line using the "rename" command in such case.
  - On Mac OS, use your favourite text editor and set the following value:
    - i. OmeroHost=outreach.openmicroscopy.org
    - ii. OmeroPort=4064

3. Launch Orbit and log into OMERO with the username and password provided:



4. Orbit will show data from OMERO in the left-hand panel. Click *show only my assets* to filter by data you own.
5. Select the "Lab1".
6. Datasets not within a Project are listed under *unassigned*:



7. Select the **svs** Dataset. Image thumbnails will be shown in the panel below.
8. Double-click the **77928.svs [Series 1]** thumbnail or drag and drop it into the centre panel and maximise the viewer window to fill the centre panel.



9. We want to train a model to recognize cell nuclei and use this for segmentation.

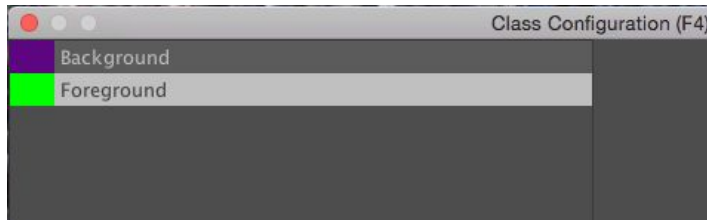


10. Click the Model tab and then the Classes button

11. In the dialog, remove the *Celltype 1* class by selecting it and click the *remove class* button.

12. Then select the *Celltype 2* class, and rename it by typing *Foreground* in the name field and clicking *rename class*.

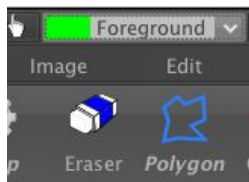
13. You should now have two classes named *Background* and *Foreground*:



14. Click *OK* to close the dialog.

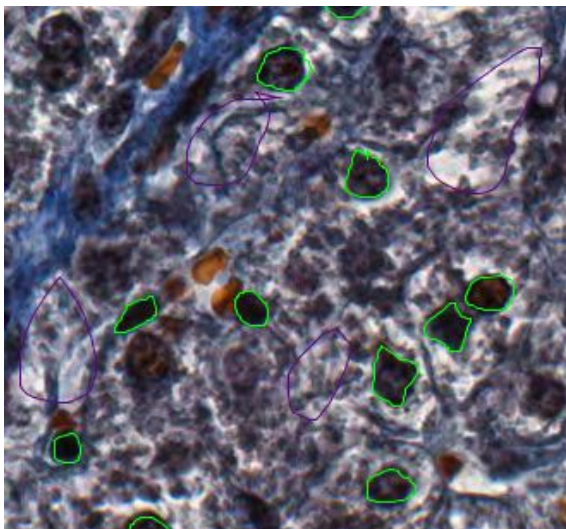
15. Now we will construct the model by defining regions of *Foreground* and *Background* on the Image.

16. Click the *Object Detection* tab, select the *Polygon* tool and choose the *Foreground* class from the chooser at the top-left of the screen.



17. You can now draw around a number of cell nuclei on the Image. The more accurately you draw and the higher number of objects you define will improve the performance of the segmentation, but about a dozen should be sufficient.

18. Now switch to the *Background* class and draw around several background regions:



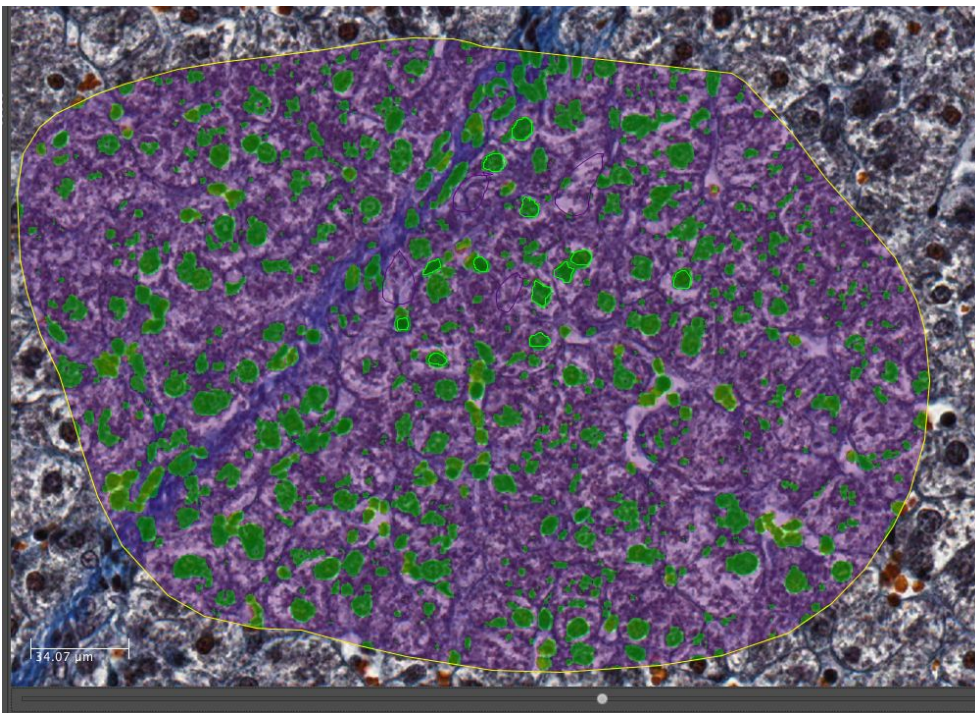


19. We can then train the model by clicking the *Train* button or press F7. You will see a progress bar in the right panel.

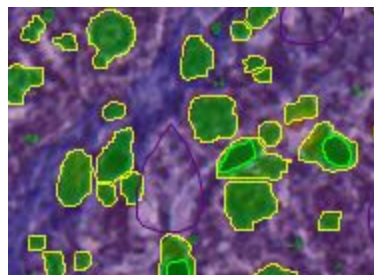
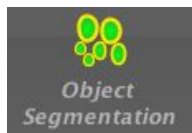
20. To see how this model classifies objects within a region, click the *Define ROI* button and draw around a region of the image. Then click *Classify*. If no ROI is drawn, Orbit will attempt to classify the whole Image which can be very time-consuming.



21. Once the classification is complete, a notification window pops up. Close it and view the results on the Image by dragging the slider below the Image to the right:



22. To segment the Image using this classification, click *Set Primary Segmentation Model* and then *Object Segmentation*.



23. Click the *Model* tab and *Save Model On Server*, enter a name to save the model to OMERO. Note that you can also use *Save Model as...* to save the model to your local drive.

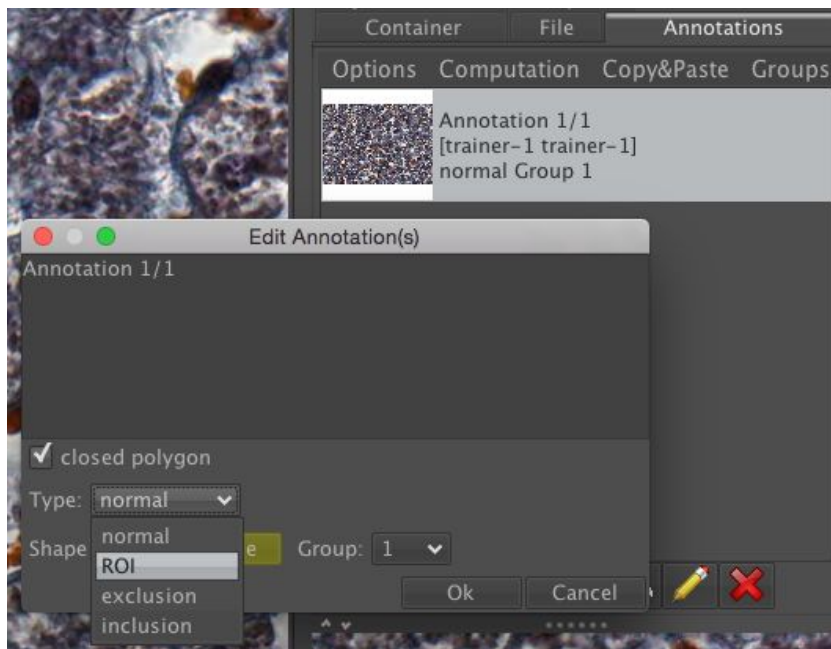
## Orbit workflow: Scripted segmentation and saving to OMERO

**Description:** We will use the model created in the last step above to repeat the segmentation, using a script which allows us to save the results back to OMERO. This will use a saved ROI Annotation instead of a temporary ROI as in the manual workflow.

1. Re-open the same image **77928.svs [Series 1]** to clear the ROIs and in the right panel select the *Annotations* tab.
2. Pan the Image to a region you wish to analyse, select the *Add Polygon* button and draw

around a region. 

3. Select this *Annotation* from the list in the right panel and click *Edit* (pencil icon).
4. In the dialog, set the *Type* to *ROI*.



5. Click *Ok*. This will save the ROI as an annotation on this image in OMERO.
6. Click on *Tools > Script Editor* to open a scripting window.
7. Copy the script from training-scripts: [orbit/segmentation.groovy](https://github.com/ome/omero/blob/master/scripts/orbit/segmentation.groovy) and replace the existing code in the script window.
8. Update username, password
9. The script will load the Orbit model and the ROI that we saved to OMERO, segment the image within the ROI and save the segmented shapes as Polygons to OMERO.
10. Click *Run*.
11. When complete, you can use *iviewer* to see the ROIs created in OMERO.

## Using the OMERO Java API to create Annotations

1. We can use the Orbit's script editor as a convenient way to explore the OMERO Java API.
2. Orbit interacts with OMERO (as above) using the Java class <https://github.com/mstritt/image-provider-omero/blob/master/src/main/java/com/actelion/research/orbit/imageprovider/ImageProviderOmero.java>
3. We will use the "Create a Map Annotation" example from <https://docs.openmicroscopy.org/omero/5.4.6/developers/Java.html> to add an annotation to the Image we've just analysed.
4. We will reuse the segmentation script that we have just run, but instead of saving Polygons we will create a Map Annotation to summarise some parameters of the analysis.
5. Find the "Create a Map Annotation" section on the OMERO Java documentation page above.
6. Copy this code and go to the Orbit script editor, with the segmentation script.
7. Edit the script to remove the for loop that saves Polygons `for (Shape shape: res.shapeList) {`, replacing it with the code we have just copied.
8. We can update the Keys and Values to e.g.:
  - o Analysis: Orbit
  - o ROIs: Roi count
  - o Model: Model name
9. We also need to link the Annotation to an Image instead of a Project, using the `omerolmageld`.
10. For example:

```
57
58 List<NamedValue> result = new ArrayList<NamedValue>();
59 result.add(new NamedValue("Analysis", "Orbit"));
60 result.add(new NamedValue("ROIs", "" + res.shapeList.size()));
61 result.add(new NamedValue("Model", model.getName()));
62 MapAnnotationData data = new MapAnnotationData();
63 data.setContent(result);
64 //Use the following namespace if you want the annotation to be editable
65 //in the webclient and insight
66 data.setNameSpace(MapAnnotationData.NS_CLIENT_CREATED);
67 DataManagerFacility fac = gateway.getFacility(DataManagerFacility.class);
68 fac.attachAnnotation(ctx, data, new ImageData(new ImageI(omeroImageId, false)));
69
```

11. Run the script again and then check in webclient for the Map Annotation on the image.

## Session 3: Analysis with Fiji and R

### Description



Fiji is a free open-source image processing package based on ImageJ.

We show how to analyze a batch of images stored in OMERO using the scripting facility in Fiji, the results of the analysis will be stored as an OMERO.table, then analyze the images using an OMERO script server-side, the results will this time be stored as a map annotation.


If analyzing other users data, the user analyzing the data needs to have admin privileges.

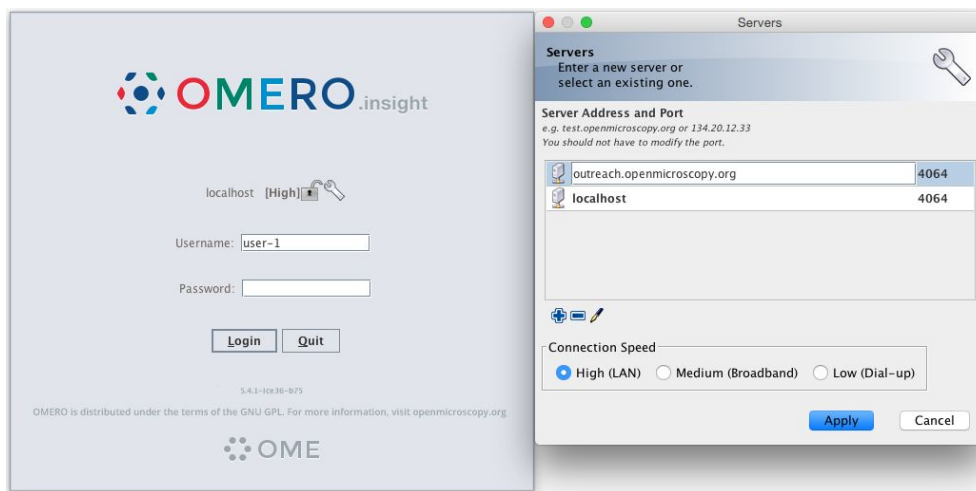
More information about restricted privileges can be found at <https://docs.openmicroscopy.org/latest/omero/sysadmins/restricted-admins.html>.

## 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 of Fiji.
- The OMERO.script [https://github.com/ome/training-scripts/blob/v0.1.0/practical/python/server/batch\\_roi\\_exp\\_ort\\_to\\_table.py](https://github.com/ome/training-scripts/blob/v0.1.0/practical/python/server/batch_roi_exp_ort_to_table.py) has been uploaded to the server.

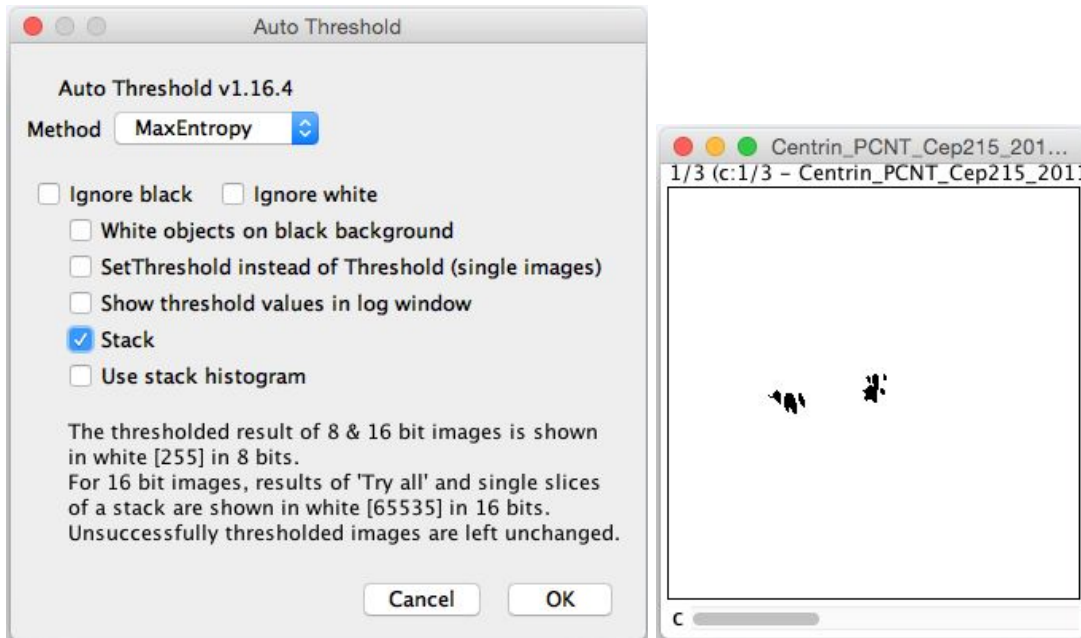
## Fiji Manual workflow

12. Launch Fiji.
13. To open an image stored in OMERO, Go to *Plugins > OMERO > Connect To OMERO*.
14. 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*.

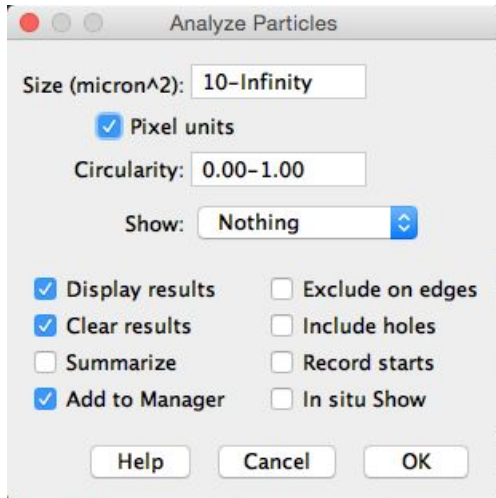


15. Enter your *Username* and *Password* and click *Login*.
16. Browse to Project **idr0021**, open any Dataset and double-click once on an Image to open it in Fiji. Bio-Formats is used to view the Image.

- Make sure to select *View stack with: Hyperstack* in the *Bio-Formats Import Options* dialog.
  - Note that each plane will be transferred from the server to the client machine so this may take a few moments.
17. To open the recorder, go *Plugins > Macros > Record...*, select *JavaScript* to record the actions. The steps will then be used in the Scripting workflow below.
18. Convert floating-point pixel-type to 8-bit using *Image > Type > 8-bit*.
19. Go to *Image > Adjust > Auto Threshold*, to open the *Auto Threshold* dialog:
- Select *MaxEntropy* for the *Method* parameter.
  - Check the checkbox *Stack*.



- 20.
21. Click OK.
22. Then open *Analyze > Analyze Particles...*
23. In the dialog
- Set *Size* to *10-Infinity* and check *Pixel units*.
  - Check the following checkboxes:
    - i. *Display results*
    - ii. *Clear results*
    - iii. *Add to Manager*
  - Click OK then Yes in the popup dialog indicating asking to *Process all 3 images?*.



24.

25. To save the thresholded image back to OMERO with the generated ROIs and the measurements:

- select *Plugins > OMERO > Save Image(s) to OMERO*.
- Create a New Dataset for the image
  - i. Click *New...* button next to the selection box on the Dataset row.
  - ii. In the dialog that pops up, enter a name and description (optional).
  - iii. Click *Create*.
- The newly created dataset will automatically be selected.
- Click *Add to the Queue* then *Import*.
- Go to OMERO.web and check that the measurements have been saved in a CSV file and attached to the images. The attachment can then be downloaded at any time.

## Fiji Scripting workflow

We will now repeat the manual analysis on a Dataset using the scripting facility available in Fiji.

1. Go to <http://outreach.openmicroscopy.org/>.
2. Log in
3. Select the Project **idr0021** and first Dataset **CDK5RAP2-C**.
4. Note its ID on the right-hand panel, it will be used below.
5. Launch Fiji. (relaunch if opened).
6. Go to *File > New > Script...*
7. A dialog pops up. In the *Language* menu select *Python*.
8. Copy the content of the script [https://github.com/ome/training-scripts/blob/v0.1.0/practical/jython/analyse\\_particles\\_for\\_another\\_user.jy](https://github.com/ome/training-scripts/blob/v0.1.0/practical/jython/analyse_particles_for_another_user.jy) into the text editor.
9. 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

```

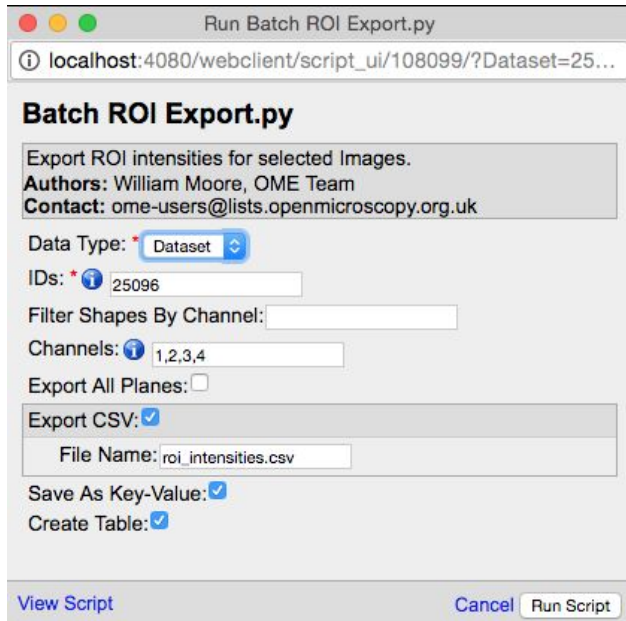
a.


10. Click *Run*.
11. The script will process all Images in the specified Dataset, applying threshold, analyzing particles and saving ROIs back to the Images in OMERO i.e. we reproduce in a script the manual steps recorded above.
12. Return to OMERO.web and open an Image from this Dataset in OMERO.iviewer.
13. Click the *ROIs* tab to see the added ROIs. Note that the ROIs have been assigned a Channel index to indicate which Channel they were derived from.
14. In the *Settings* tab, turning channels on/off will also show/hide ROIs assigned to those channels.

## Analysis workflow Server side

1. We will now analyse ROIs created on the Images above, getting the pixel intensity and area (points) of ROIs and creating a summary of the ROIs on each image.
2. Go to <http://outreach.openmicroscopy.org/> and login.
3. Select Project **idr0021** and first Dataset **CDK5RAP2-C**.
4. Click the *Script* button in the top-right of the page .
5. Under omero > export\_scripts > Batch\_ROI\_Export... we can see a new script released with OMERO 5.4.6 to export ROIs from multiple images.
6. We will use a customised version of this script.
7. In the script menu, go back... to the top of the menu, then select *users\_meeting > batch roi export to table....*
8. In the dialog that pops up, click on *View Script* to view the Python code.

9. Search for “idr” to find the code block that selects the filter\_channel based on Dataset and Channel names.
10. 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 Datasets or Images.



11. [View Script](#)
12. The script will process all the Images in the selected Dataset and data can be:
  - a. Exported as CSV file, with one table row per Shape/Channel.
  - b. Added as Key-Value pairs (Map Annotations) to Images. ROIs summarised as min/max/mean per Image.
  - c. Added as an OMERO table to the parent Project, with one row per Image.
13. Uncheck the *Create Table* option, since we have previously created a Table by running this script on ALL the Datasets in the **idr0021** Project.
14. Click *Run Script*.
15. The status of the processing is displayed in the Activities dialog .
16. When the script has completed, it will show in the the Activities dialog and allow you to download the CSV file. Download this and open in e.g. Excel to see the output data for all the shapes.
17. You will also be able to view the Key-Value Pairs (Map Annotations) in the right-hand panel of each Image that summarise all the ROIs on each image.

## Notebook R

For this section, we will use a notebook since the setup is a bit more complex. It is necessary to install several dependencies. We use JupyterLab, a dedicated computational environment. In the following example, we will generate a plot in R and some basic statistics from the output of the *batch roi export to table* script. The plot will be uploaded back to OMERO.

1. Go to <https://idr-analysis.openmicroscopy.org/training/hub/tmplogin>

2. In the Files > notebooks > R folder, select the notebook *idr0021\_rois.ipynb*.
3. Set the 'username' and 'password' in the first code cell.
4. Use the play button to move through the cells (for code cells wait until the \* changes to a number, i.e. the execution finished).
5. Run the first code cell to connect to the server.
6. The first step is to find the idr0021 project, the attached table and load the image and max\_points data directly as R dataframe. (Right-click -> Enable Scrolling for output, to collapse the long output)
7. Prepare the R dataframe: Reformat the image column to only contain the image ID. Iterate through the datasets to get the Image <-> Protein (dataset name) relation. Estimate the toroid parameter from the area (max\_points).
8. Then plot the data. First the datasets (Proteins) are ordered from lowest to highest mean diameter (to match the figure in the paper).
9. Check if there is a significant difference between the Proteins with respect to the toroid diameter. This is done as one-way analysis of variance. The value of  $2e-16$  indicates that there is a difference. One-way analysis of variance is questionable because it assumes normal distributed data and similar variances between the groups (this is not given, see plot). Therefore also do a 'Mann-Whitney' test to check the differences between each possible pairwise combination of Proteins. -> There is a significant difference between most of the groups, not for some of the groups (e. g. CDK5RAP2-C vs CEP192-M vs TUBG1-N , which can be seen on the plot as well).  
The results show a similar picture as the paper, but the details are different, because we used a simpler approach to estimate the toroid diameter, whereas the authors of the paper used a more accurate method to measure the toroid diameter.
10. Finally the plot and results are saved back to OMERO and attached to the project.

## Session 4: OMERO.parade and OMERO.figure

### Description

We show how to use OMERO.mapr: an OMERO.web app that enables browsing of data through attributes linked to images in the form of Map Annotations.

For more information about OMERO.mapr <https://github.com/ome/omero-mapr>

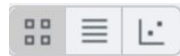
### Analyze metadata using OMERO.parade

1. Select the Project **idr0021**.
2. Choose the *Parade* option in the centre panel dropdown menu.
3. Expand all Datasets by clicking on the *Open All* button.
  - a. All the datasets will be expanded in the left-hand tree.
  - b. The Thumbnails will be loaded in the centre panel. This allows to browse a full project.

- c. Note that if you collapse a Dataset in the tree, the Thumbnails will be removed from the centre panel.
4. In the *Add filter...* selection box, select the *Key\_Value* item.
  - a. When the Map Annotations are loaded, pick the *Key Gene Symbol* and enter the Value *CEP* to show all *CEP* genes and then *CEP120* to show only images with that gene.
  - b. To remove this last filter, hover over the filter and click the X button that shows on hover.
5. In the *Add filter...* selection box, select the *ROI\_Count* item.
  - a. Enter a Value  $> 20$ . When you hover over the area used to enter the value, the range is indicated in the tooltip.
  - b. Then enter  $< 3$  or 4.
6. Remove all filters by clicking the X button showing on hover.
7. In the *Add filter...* selection box, select the *Table* item so we can find using the analytical results generated previously:
  - a. Choose the *max\_points* item and drag the slider to filter the Images. Note that *PCNT* has the largest number of Images with large ROIs.



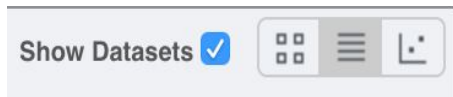
- b.
  - c. Adjust the controls to select  $< 50$ .
  - d. Note that in that case *CPAP* and *CENT2* have the most of the Images as expected.



8. Switch to Table layout (middle button)
9. In the selection box *Add table data...*, select
  - a. *Table\_max\_points*
  - b. *Table\_mean\_points*
  - c. *ROI\_count*

Note that it is currently not possible to remove a column.



10. Click on the name of a column to sort it.
11. Uncheck *Show Datasets* to sort all Images together e.g. by ROI count.



- 12.
13. Check the checkbox in each column to show the *Heatmap*. Note the corresponding pattern in the Heatmap.
14. Switch now to the Plot Layout (third button)
15. It takes the table data loaded and plot the values.
16. Filters can be added to plot the relevant results.
17. Try plotting by different Axis values.
18. Closing a Dataset in the left-hand tree removes the values from the plot.
19. Drag to select several outliers.
20. Note that you can use the selected images in right panel to annotate or *Open with...*

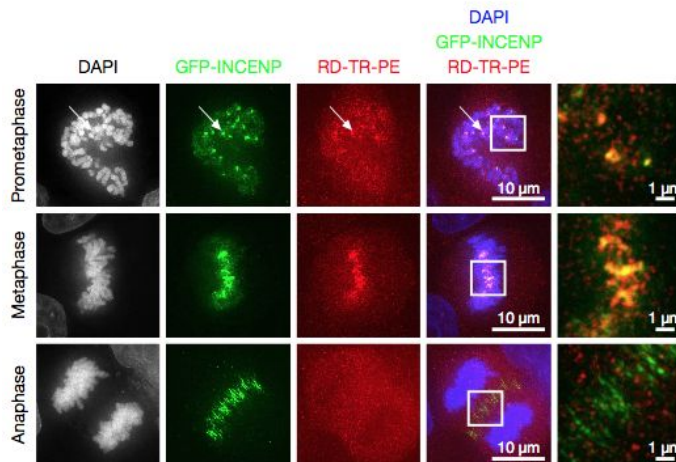
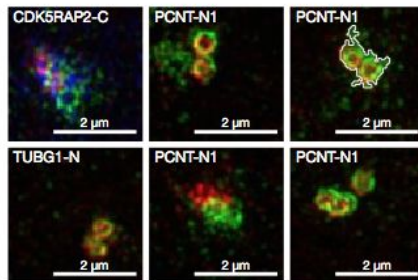
21. Choose *Open with Figure...*

## OMERO.figure: Use metadata in figure creation

1. Drag to arrange the Images into two rows, select all and click the *snap to grid* button  at the top of the page.
2. Select all Images and Zoom in around ~300%
3. Go to the *Labels* tab, select all Images and add a Scalebar of 1  $\mu\text{m}$  and adjust the size of the Label to 12.
4. Add label from Dataset name: *color=white location=top-left*
5. Select one image. In the *Labels* tab, click the *Edit* button for ROIs.
6. Load ROIs from OMERO and mouse-over the list to pick the largest ones.
7. Click to add it to the image and click *OK*.
8. Return to webclient tab, go to *siRNAi-HeLa* Dataset and filter by Rating of 5 stars.
9. Select 3 images and in right panel click link icon  then copy the link.
10. Return to figure tab, click *Add Image* button and paste the link into dialog. Click *OK*.
11. Arrange the 3 images into vertical column, select all and click *snap to grid* button.
12. Copy the 3 images and paste (keyboard shortcut or *Edit > Copy/Paste*) 3 times to create 3 more columns.
13. Select the first column panels and adjust rendering settings: Only first channel on, white.
14. Repeat for next 2 columns: 1 channel on for each column, adjusting levels if desired, leaving 4th column as 'merged' with multiple channels on.
15. Select all panels and zoom a little. Then select a row and drag image in the *Preview* tab to pan.
16. Copy and paste 'merged' column again to create 5th column. Zoom in to approx 500%.
17. Select the *Labels* tab, select the 'merged' and zoomed columns and click *Show Scalebar* button.
18. Click the *Label* checkbox to add label to scalebar. Select only the zoomed-in panels and change the scalebar to 1 micron.
19. Select the top-left panel and enter label text in the *Add Labels* form.
20. Choose label size, position (left vertical) and colour and click *Add* to create label.
21. Now undo this (Ctrl-Z or *Edit > Undo*) and instead create labels from Tags:
22. Select first column of panels and choose *Tags* from the label text-field drop-down options. Click *Add*.
23. Select the first row of panels and create labels in 'top' position using the *Channels* option.
24. Edit the created labels (bottom of *Info* tab) to rename green labels to GFP-INCENP.
25. Select just the first 'merged' image and click the ROIs *Edit* button in *Info* tab.
26. Draw Arrows or other shapes on the image, or load from OMERO. Click *OK*.
27. Click *Copy ROIs* in *Info* tab, select other panels in the same row and click *Paste* to add ROIs to these panels.



28. To create a Rectangle ROI indicating the region of the zoomed-in image, select the zoomed-in image on the first row and click *Copy* of the cropped region at the bottom of the Preview tab.
29. Now select the zoomed-out 'merged' panel, and paste this region as an ROI by clicking *Paste* under ROIs section of the *Labels* tab.
30. Repeat for other rows of the figure.
- 31.



- 32.
33. Go to *File > Paper Setup...* and choose Pages: 2 in the dialog. Click OK.
34. Finally return to webclient, select 'Big' images from the svs Dataset, copy the link to them and paste into the Add Image dialog in figure.
35. Move these big images to the 2nd page.
36. In the header, click on the Save button to save the Figure as "Figure 1".
37. To open other saved files, go *File > Open...*
38. We can view figures from our collaborators here and filter by name or Owner.
39. Open "Aurora-B figure 2" from trainer-2.
40. Select a panel and click on the *Webclient* link In the Info tab to show image in webclient.
41. Back in OMERO.figure, go to *File > Open* to choose the "Figure 1" file we saved above.
42. then click on *Export PDF* to export it as PDF.
43. Download the PDF and open it. If opened in a suitable application e.g. Illustrator, the elements on the page will still be editable.