Neubias Workshop 18/10/2018

The presentation and a PDF version of the workshop are available at https://downloads.openmicroscopy.org/presentations/2018/Neubias-Edinburgh/

Software versions used for this workshop:

- OMERO: 5.4.9
- OMERO.iviewer: 0.5.0
- OMERO.figure: 4.0.1
- OMERO.parade: 0.1.1
- OMERO.FPBioimage: 0.3.0
- OMERO training scripts: 0.4.1
- Bio-Formats: 5.9.1

Program

1. Introductory talk about OMERO
2. OMERO core concepts
3. Analysis workflow
4. Analyze metadata using OMERO.parade
5. OMERO.figure

OMERO core concepts

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

Data setup and ownership, usernames

All images for this workshop have been pre-imported for you into the OMERO.server. For training purposes, we prepared 40 users on the OMERO.server. **Each** of these 40 users has **their own set of images.** These sets consist of i**mages of the same name, size, shape, form and quality for each user.** Thus, the data of each user appears the same, but, in actual fact, are different and totally independent sets.Thus, small discrepancies and differences between the users are completely possible. Further, if one user deletes their own data in OMERO (**please do not delete anything** on our server), this will not have any bearing on the other 39 sets of images belonging to the 39 other users.

The **login names** of the 40 users are “user-x” where x goes from 1 to 40, We have given to each of the 40 users in OMERO a unique first name and surname which we picked from a list of 40 famous scientists e.g. Ada Lovelace or Francis Crick. This is the name (“your” name) which you will see in the top-right corner of the OMERO webclient after you log in with your loginname. In the OMERO webclient the default view shows only your own images. The beginning of the **Browse images** section below explains how to navigate to other users’ data.

Import setup (for information only)

For this workshop, images have already been imported into OMERO. To demonstrate the import process, we will import a few more images using the Java Desktop client: OMERO.insight.

1. Import using OMERO.insight
   a. Requirement: Java version 1.7 or higher must be 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 (for your information only)
   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

Import images (for information only)

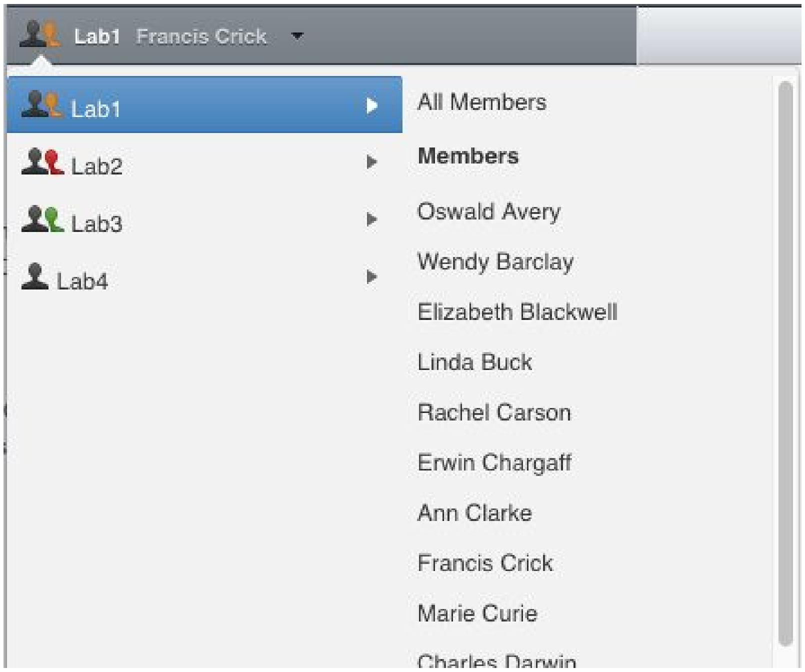
1. Open OMERO.insight and in the login dialog, click onto the spanner icon 🔧. This will open a list of servers to which you can connect to. By default, only “localhost” is listed. Click on the *plus* icon to add a new line to the list and type into the line the server address outreach.openmicroscopy.org
2. When done, click *Apply.*
3. Log in using the username and password provided.
4. In OMERO.insight, click on the Importer Icon in the toolbar.
5. Browse your local hierarchy in the left-hand pane of the importer, select single images or whole folders and add these to the Queue by clicking on the *arrow* ➤ icon.
6. In the *Import Location* window, select the target Project and Dataset (existing or create a new one) to import to.
7. Note: If no Dataset is selected or created, a new Dataset will be automatically created and named after the folder containing the images to be imported.
8. Optional: Go to the *Options* tab
   a. Click on Add tag to images ⊞ : to bring the Tag selection dialog.

    b. Select the tag(s) on the left-hand side or create a new one.
    c. Click > to move the tag(s) to the right-hand side.
    d. Click *Save.*

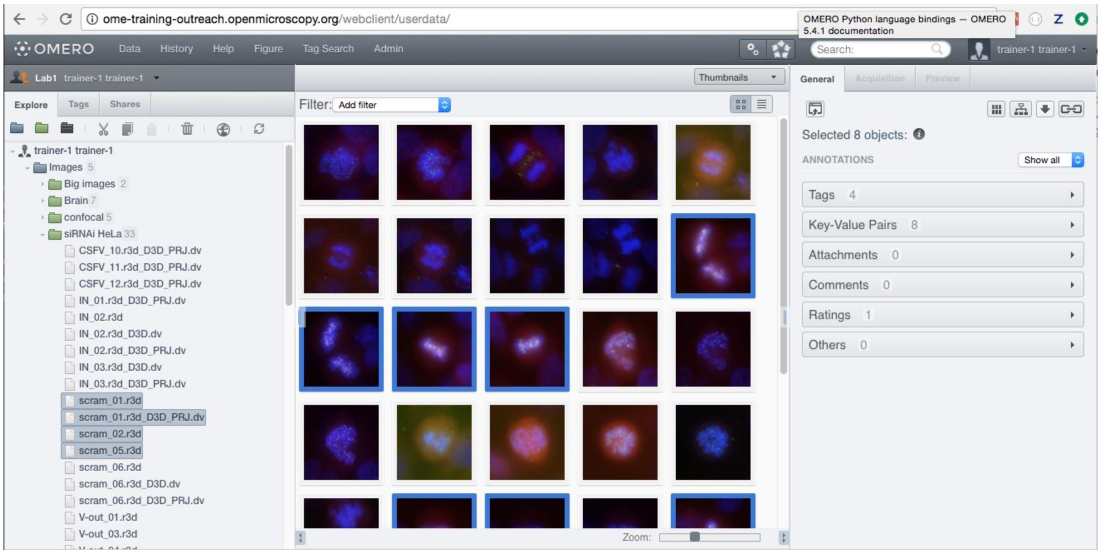
9. Click on the *Import* button in the bottom-right corner of the Importer window. You should see two progress bars for every image imported, *Upload* and *Processing*.
10. Note: The import of the next image in the queue starts immediately after the *Upload* of the previous one is finished. The *Processing* phase of the import is done on the OMERO.server only, and can be finished while the next image(s) is/are being uploaded to the server.
11. Once imports are finished, go back to the OMERO.insight main window and click the *Refresh* button 🔄 above the right-hand pane. This will display the imported images inside the Dataset and/or Project you specified previously in the *Import Location* window.
12. (demo only step) Now, the demonstrator will log out from OMERO.insight and login again, this time as some other user and will show the import process again, this time importing a different set of images. After this import, the two sets of images (belonging to two different users) will be shown in the webclient.

Browse Images

1. In your web browser, go to https://outreach.openmicroscopy.org/
2. Log in using the username and password provided.
3. 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.
4. 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 from the menu.



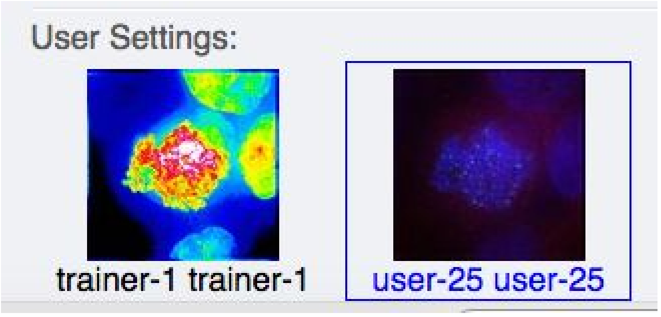
5. You can browse ‘folders’ in the left-hand pane: *Image* folders are called *Datasets* and they are within *Projects*.
6. When a Dataset is selected, Image thumbnails are shown in the centre panel.



7. These represent imported Images. The original Images are stored on the server and the generated thumbnails allow us to browse them.
8. Bio-Formats, https://www.openmicroscopy.org/bio-formats/, is used to read the 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.
9. Select an Image. In the right-hand pane, 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. All the metadata read by Bio-Formats can be downloaded at any time.

10. In the *Preview* tab in the right-hand panel, you can also view the Image.
11. For multi-plane images, sliders allow you to move through Z or Time dimensions.
12. 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.
13. You can adjust the rendering settings for each channel e.g. turn on/off the channels, adjust color settings, look-up tables, etc..
14. The rendering settings can be saved to the server. This **NEVER** changes the original Image data and can be reverted at any time.
15. The rendering settings can also be copied and pasted between Images. To modify the rendering settings in batch, click on the *Save to All* button to apply the same settings to, for example, all Images in a given Dataset.
16. You can use the settings which other users saved on your Images and apply them for your own Image. These settings are highlighted as thumbnails in the lower part of the *Preview* pane. .

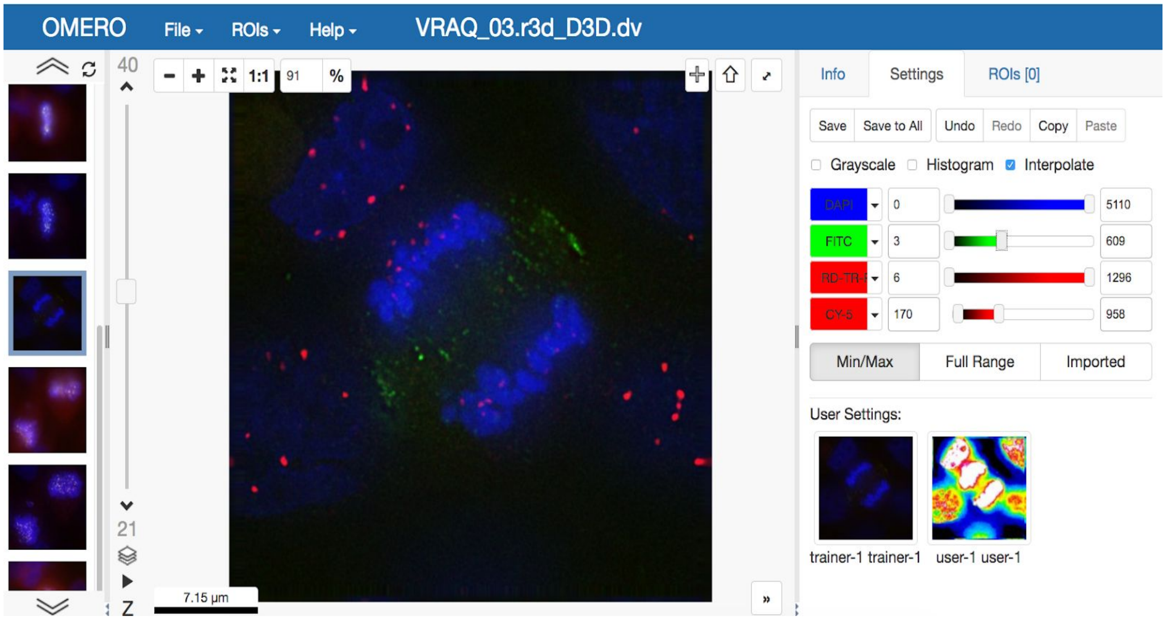


17. Your own settings are highlighted in blue.
18. You can revert to the original settings for an Image or Dataset. For example, using the context menu for a Dataset in the tree, select *Rendering Settings* > *Set Imported and Save*.

View Images

1. Double-click on an Image thumbnail, or click the *Full Viewer* 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.

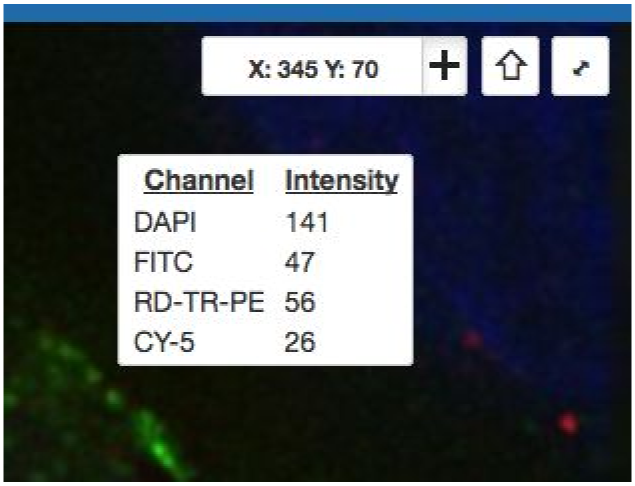
2. We can adjust the rendering settings and scroll through Z or T.



☐ Grayscale ☐ Histogram ☑ Interpolate

DAPI	0	5110
FITC	3	609
RD-TR-PE	6	1296
CY-5	170	958

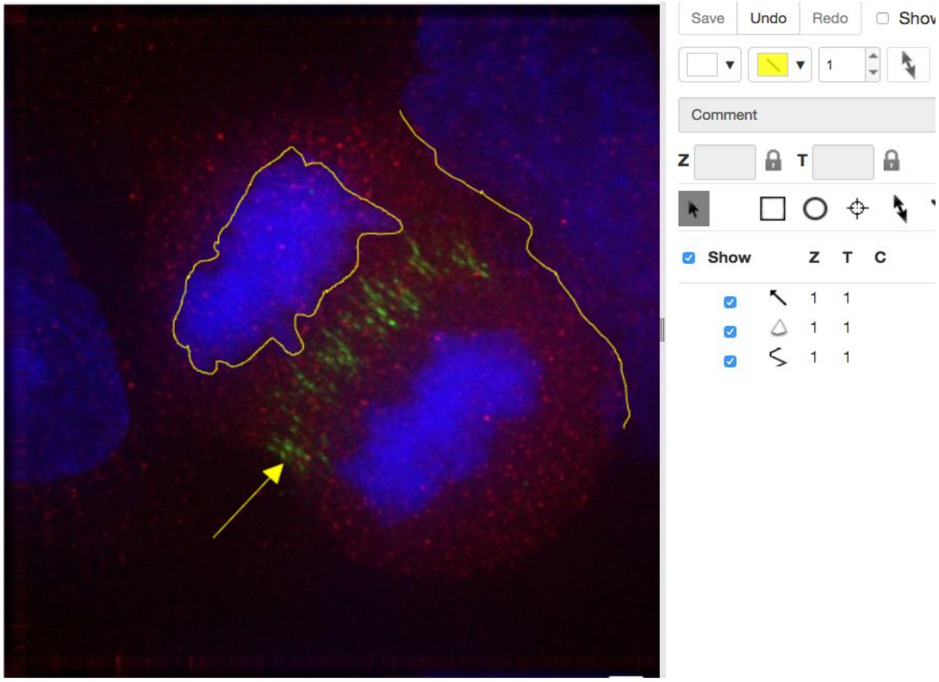
3. Click the *Crosshairs* icon ✛ at the top-right of the viewer to enable the pixel intensity display for the mouse pointer. Then mouse over the Image to see the pixel intensities for the channels turned on.


X: 345 Y: 70

Channel	Intensity
DAPI	141
FITC	47
RD-TR-PE	56
CY-5	26

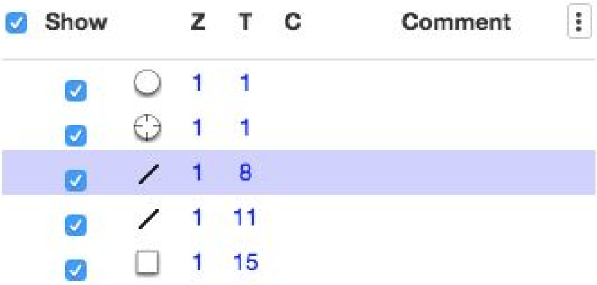
4. The ROIs tab ROIs [0] includes tools for viewing and drawing ROIs on the Image. These are saved back to the OMERO server. Select the *arrow tool* in the right-hand pane ↘ and draw an arrow on the Image, using Click-Move-Click (not drag), pointing to a feature in the Image and save it using the *Save button* Save located in the upper part of the right-hand pane. Draw also a couple of other ROIs on different Z planes.

    a. Dragging on the Image is used for panning the image.
    b. *Shift-drag* rotates the image (using selection tool) or draws “freehand” (polygon, polyline, rectangle and ellipse).



☑ Show		Z	T	C
☑	↖	1	1	
☑	△	1	1	
☑	⟨	1	1	

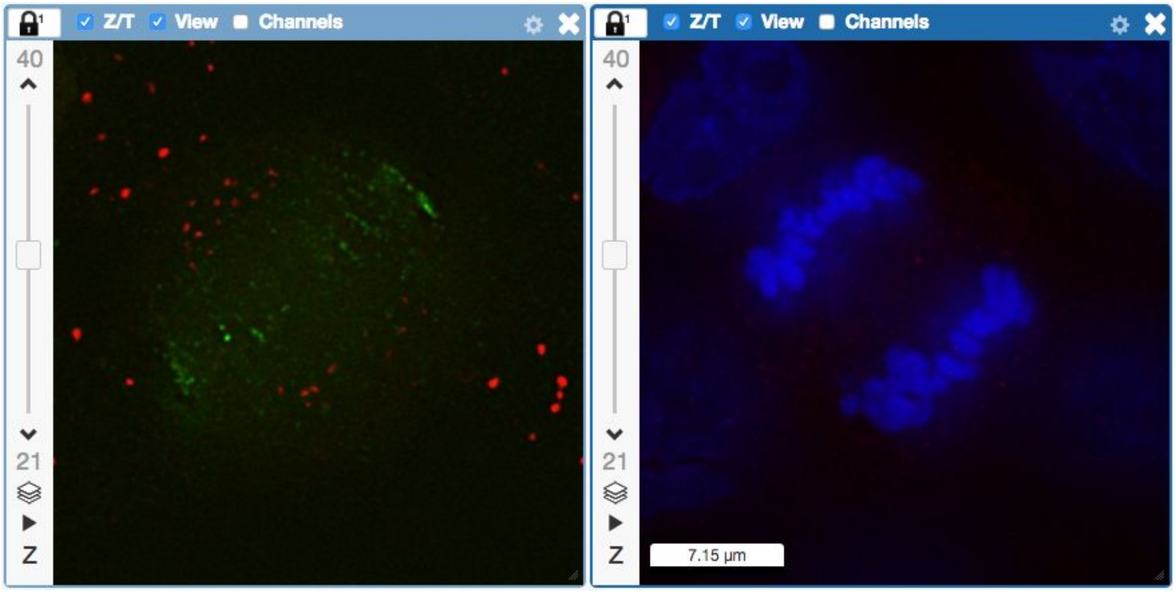
5. Clicking onto the ROIs in the right-hand pane table brings the viewport to the position where the selected ROI is in the middle of it and navigates the viewport to the timepoint the ROI is associated with.



☑ Show		Z	T	C	Comment
☑	○	1	1		
☑	⌖	1	1		
☑	/	1	8		
☑	/	1	11		
☑	□	1	15		

6. Select several ROIs from different Z planes. You can select in the table (*Shift-click* to select a range) but also in the image itself using *Cmd-drag* to select multiple ROIs.
7. 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.
8. Save the viewport by selecting in the top-left corner *File* > *Save Viewport as PNG.*
9. Compare the two channels of one of the Images in multi-image view:
    a. 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.
    b. Select one of the windows and in the right-hand pane switch the green channel off.

c. On the other window, switch the blue channel off.



10. We can synchronize the viewers by adding them both to the same sync Group.
    a. Click the *sync icon* in the top-left corner and select the first option in the dropdown menu.
    b. Repeat for the other viewer.
    c. Now both Images have the *Z/T* and *View* (zoom and position of viewport) synchronized. .
11. Click on *Info* tab in iviewer now, and find, in the right-hand pane, the link to *Dataset:*

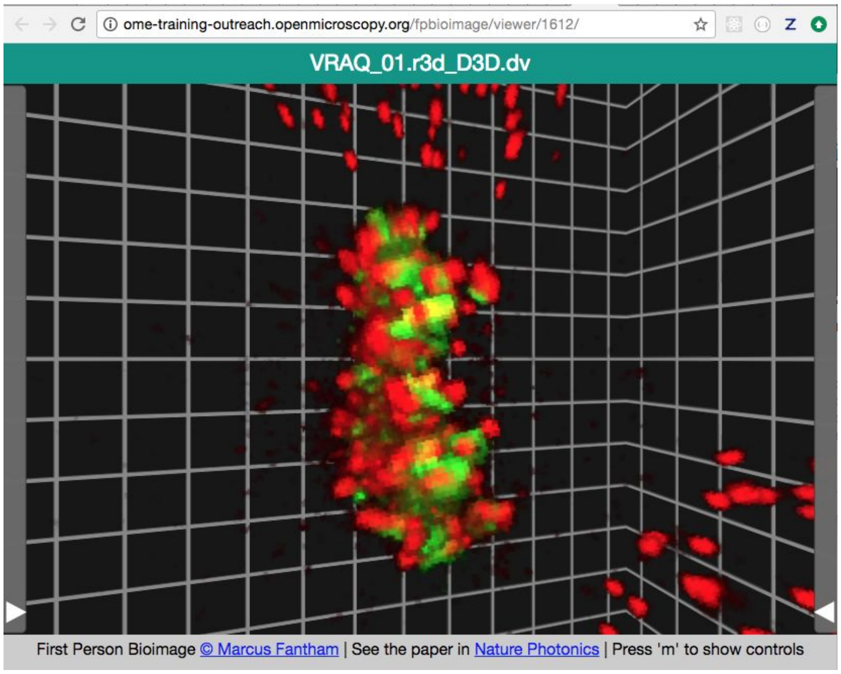
Dataset: siRNAi-HeLa

. Click on that link. This will bring you back to the webclient.

12. Adjust the channel names:
    a. Select any image inside that Dataset and click on the *pencil* icon in the right-hand pane next to Channels.
    b. Input **"DAPI"** instead of channel **"457"** and **"FITC"**, **"GFP-INCENP"** and **"CY-6"** for the other channels.
    c. Click the *Apply to All* button Apply to All and confirm by clicking *Continue*.
13. Open an image from the Dataset **siRNA-HeLa** with multiple Z-sections e.g. **VRAQ_01.r3d_D3D.dv** in a 3D viewer: OMERO.FPBioimage.
    a. First select the Image.
    b. In the *Preview* tab, switch off all channels except FITC and the GFP-INCENP channel.
    c. Save the new rendering settings.

d. Right-click on the image, or use the *Open with...* icon on top of the right-hand pane to open the image with FPBioimage.
e. Click *Start* in the new viewer window.
f. We can see that the centromeres are well aligned on the metaphase plate on the selected Image, whereas the centromeres are located in and around the spheroid on the **IN_02.r3d** Image for example.

14.



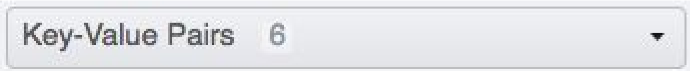
ome-training-outreach.openmicroscopy.org/fpbioimage/viewer/1612/

VRAQ_01.r3d_D3D.dv

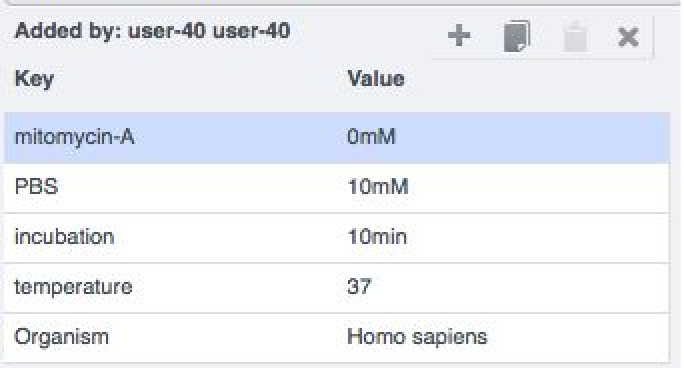
First Person Bioimage © Marcus Fantham | See the paper in Nature Photonics | Press 'm' to show controls

Add Annotations, Filtering and Searching

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



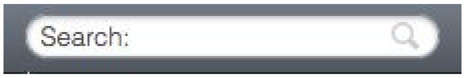
Key-Value Pairs 6



Added by: user-40 user-40

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

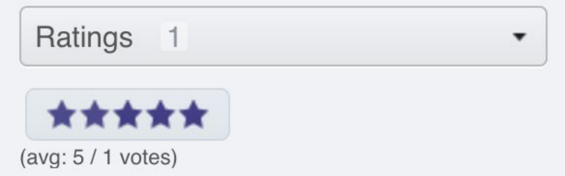
7. The *Key-Value Pairs* allow you to add lab-book like additional metadata for the Image. These *Key-Value Pairs* are also specifically searchable.
8. It is also possible to add Comment, Attachment and Rating to selected Images.
9. Enter ***mitomycin-A*** into the search box in the top right corner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



Search:

10. 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..
11. 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 Browse 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.
12. Images can also be filtered by Tag or Rating in the centre pane, using the *Add filter* chooser above the thumbnails.
13. For example, choose Tag and then select **Metaphase** from the list of Tags to show the images we tagged earlier.
14. 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



Ratings 1

★★★★★

(avg: 5 / 1 votes)

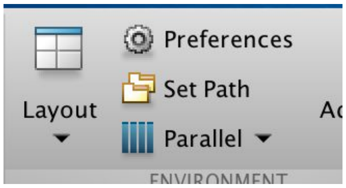
15. Now we can remove the filtering by Tag and instead filter by *Rating of 5* to show only our favourite images.

Analyze data using MATLAB

Here we demonstrate how to analyze a batch of images associated to the paper [Subdiffraction imaging of centrosomes reveals higher-order organizational features of pericentriolar material](). This will show how to connect to OMERO, load data, analyse and save the output back to the server. We will use [https://docs.openmicroscopy.org/latest/omero/developers/Matlab.html]() as a reference.

Setup

1. The OMERO.matlab toolbox and the Image Processing toolbox have been installed.
2. Make sure that the OMERO.matlab toolbox is on the MATLAB path. To add it to the path, you can
    a. Launch MATLAB
    b. Under the *HOME* tab, click on *Set Path* (middle of the top task bar).



.

    c. A Set Path dialog pops up.
    d. Click on the button *Add with Subfolders...*
    e. Select the OMERO.matlab toolbox, Click *Open*
    f. Close the Set Path dialog, you can either save the path for future use or not.

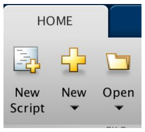
For the exercises below, we will use code examples from the following page [https://docs.openmicroscopy.org/latest/omero/developers/Matlab.html]()

- The parameter `servername` is outreach.openmicroscopy.org
- The parameter `user` is the username provided i.e. user-x
- The parameter `password` is the password provided

Exercise 1

The aim of this exercise is to show how to connect to OMERO

1. Launch MATLAB
2. Create a new Script, name it for example `analyse_dataset.m.`



3. When saved, click on the *EDITOR* tab. Note: If you do not have an EDITOR tab in your Matlab, you are probably in a new window after you clicked on *New* button in the taskbar (step above). In such case, you can click on the small blue downward arrow on top right of that window and select *Dock* to get the EDITOR tab into your workspace. Once done, you can create as many short script snippets as you like in this environment by clicking on a plus to open a new tab in your EDITOR.
4. Create a connection.
5. Print out the group identifier the user is in.

Exercise 2

The aim of this exercise is to show how to load data from OMERO. Continue in the script started in the exercise above.

1. Go to https://outreach.openmicroscopy.org
2. Log in using the credentials provided.
3. Select the Dataset **CDK5RAP2-C** in the left-hand tree. Note its ID displayed in the right-hand panel.
4. In the same script, load the dataset with all its images using the `getDatasets` function
5. Print out the name of the dataset
6. Iterate through the Images in the dataset. A Java list is returned, you may want to convert it to a Matlab list.
7. For each Image
    a. Retrieve its name and its ID.

Exercise 3

The aim of this exercise is to determine the channels to analyse. All the images have multiple channels but only the channel whose name is contained in the dataset name must be analyzed. Index starts at 0.

Continue in the same script

1. Now for each Image:
    a. Load the channels
    b. Iterate through the channels
    c. Retrieve the index of the channel whose name is contained in the dataset name using the `contains` function.
    d. Print out the index

Exercise 4

The aim of this exercise is to load the plane corresponding to the channel to analyse. Analyse it using either a modified online example or using an algorithm more suitable for the type of images. The generated masks will be converted into OMERO ROI and saved back to the server. For each Image, we will also keep track of the shape with the highest area so it can then be stored in a CSV file (see **Exercise 5**).
All the images have only one z-section and one timepoint. Continue in the same script.

For each Image:

1. Load the plane corresponding to the channel using the `getPlane` function.
2. Use the example available at https://uk.mathworks.com/help/images/detecting-a-cell-using-image-segmentation.html.
    a. In our case, the Image `I` is the plane loaded above.
    b. Use Step 2, Step 3, Step 5, Step 6. Since we skip Step 4, the Binary object in Step 5 to use is BWsdil and not BWdfill. For each step, expect Step6, remove the line starting with `figure ….` For step 6, use f = figure…. Do not forget to close the figure using `close(f);`
    c. Use "*roberts*" instead of "*sobel*"
3. **OR** (this option will give more accurate results)

    ```
    threshNstd = 6;
    minPixelsPerCentriole = 20;   % minimum size of objects of
    interest
    vals = reshape(plane, [numel(plane), 1]);   % reshape to 1 column
    mean1 = mean(vals);
    std1 = std(vals);
    % images are mostly background, so estimate threshold using basic
    stats
    thresh1 = mean1 + threshNstd * std1;
    bwRaw = imbinarize(plane, thresh1);
    BWfinal = bwareaopen(bwRaw, minPixelsPerCentriole);  % remove
    small objects
    fig = figure; imshow(BWfinal), title('segmented image');
    ```

4. Convert the masks generated into OMERO polygon using the function `bwboundaries`.
    a. Create an OMERO ROI
    b. Trace the region boundaries using the `bwboundaries` function.
    c. To store the `imageId/MaxArea`, initialize an array of zeros with 2 columns and n rows when n is the number of images using the function `zeros`. See https://uk.mathworks.com/help/matlab/learn_matlab/matrices-and-arrays.html
    d. For each boundary:
        i. Convert the boundary into an OMERO polygon. See https://uk.mathworks.com/help/images/ref/bwboundaries.html

ii. Link the polygons to the analysed plane

iii. Determine the polygon with biggest area using the `polyarea` function, see https://uk.mathworks.com/help/matlab/ref/polyarea.html. Keep track of the `imageId` and `max area` in the matrix created above.

iv. Add the polygons to the ROI

v. i.e.

```
[B,L] = bwboundaries(BWfinal, 'noholes');
max_area = 0;
for k = 1 : length(B)
     boundary = B{k};
     x_coordinates = boundary(:,2);
     y_coordinates = boundary(:,1);
     % Create an OMERO polygon, link in the to the plane
     % determine the shape with the biggest area
     % add polygons to the ROI
end
```

vi.

e. Link the ROI to the Image and save the ROI

5. Go back the https://outreach.openmicroscopy.org
6. Select an Image in the Dataset
7. Open it in OMERO.iviewer and check that the ROIs are there and that you can interact with them

Exercise 5

The aim of this exercise is to create a CSV file: each row contains the Dataset Name, the Image ID and Max area. The content of the CSV file will be plotted in Matlab, then, the CSV will be uploaded to OMERO.

1. Create a temporary CSV i.e. `.cvs` .See the first example at https://uk.mathworks.com/help/matlab/ref/tempname.html
2. Open the file in write mode `'w'` using the function `fopen.` See https://uk.mathworks.com/help/matlab/ref/fopen.html
3. Write, in the file, the string `'Dataset_name,ImageID,Area\n'` using the function `fprintf.` See https://uk.mathworks.com/help/matlab/ref/fprintf.html. This will be header of the file.
4. Write the values stored previously into the file using `fprintf`
   a. Concatenate the values per row using the `strcat` function, see https://uk.mathworks.com/help/matlab/ref/strcat.html. The numerical values need to be turned into string using the num2str function, see https://uk.mathworks.com/help/matlab/ref/num2str.html.
   b. Use `fprintf(fileID, '%s\n', concatenated_value)` to add one row per image.

5. Close the file.
6. Plot the content of the file
    a. Read the newly created CSV file using https://uk.mathworks.com/help/matlab/ref/csvread.html. Read the second and third columns i.e. ImageID and Area. Ignore the first row since it corresponds to the header.
    b. The data array read will now content 2 columns
        i. Read column 1 (ImageID) and column 2 (area)
    c. Use a scatter plot to plot column 1 vs column 2. Name the axis as ImageID and Area. See https://uk.mathworks.com/help/matlab/ref/scatter.html
7. Create a file annotation with the CSV file and link it to the Dataset. Use as `mimetype` `text/csv` and as `namespace` `training.demo`
8. Go back the https://outreach.openmicroscopy.org
9. Select the dataset
10. Check that it has an attachment

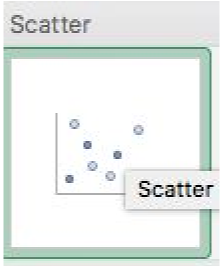
Exercise 6 (optional, more advanced)

We analyse all the images contained in the Project **idr0021**. We follow the steps from Exercises 1 to 5 but we need to keep track of the max area and the number of points on the boundary for each image in each dataset contained in the **idr0021** Project.
Images with the extension `.tif` must be excluded.
We will attach the CSV file and the OMERO.table to the Project itself (not to single datasets).
We will study the CSV further using Excel and we will use the OMERO.table for OMERO.parade in the further course of the workshop. You can follow the OMERO.parade part (see below) even if you do not manage to execute this exercise here, because we prepared a user with the OMERO.table prior to the workshop.

1. Go to https://outreach.openmicroscopy.org
2. Log in using the credentials provided.
3. Select the Project **idr0021** in the left-hand tree. Note its ID displayed in the right-hand panel.
4. Create a new Script, name it for idr0021.m
5. Create a connection.
6. Load the Datasets and Images contained in the Project **idr0021** using the `getProjects` function.
7. Iterate through the Datasets
8. For each Dataset
    a. Retrieve its name
    b. Iterate through the Images
        i. For each image
            1. Load the channels

            2. Select the channel whose name is contained in the dataset name. Only that channel should be analysed.
            3. Load the plane corresponding to the first z-section, first timepoint and the channel selected above (index starts at 0)
            4. Analyse the loaded plane, see **Exercise 4**
            5. Convert the boundaries of the segmented objects into an OMERO polygon
            6. Determine the polygon with the biggest area. Store the value for each image.
            7. Create a ROI and add the polygons to it.
            8. Link the ROI and the Image and save back to OMERO

8. Save the biggest areas to a CSV file
    a. A row should contains
        i. The name of the Dataset
        ii. The ID of the Image
        iii. The value of the biggest area
        iv. The number of points on the boundary
    b. Upload the CSV and link the file to the Project **idr0021**
9. Now Save the results as an OMERO.table
    a. A row should contain:
        i. The ID of the Image
        ii. The value of the biggest area
        iii. The number of points on the boundary
10. Go back to OMERO.web. Check that a CSV file is attached to the Project **idr0021** e.g. results.csv
11. We can now download this file from OMERO and open it in Microsoft Excel.
12. In Excel, select the *Dataset* and *Area* columns and then click on *Insert > X Y Scatter* > *Scatter .* Depending on the version of Excel you are using, this option might not be available.



.

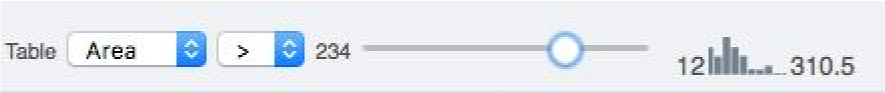
13. This will create a scatter plot. Right-click into the Chart now and select *Change chart type > Statistical > Box and Whisker.*

OMERO.parade and OMERO.figure

Analyze metadata using OMERO.parade

First let's look at the analytical results generated using the Matlab script above

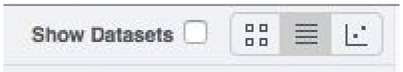
1. Select the Project **idr0021**.
2. Choose the *Parade* option in the centre panel dropdown menu.
3. Expand all Datasets by clicking on the *Open All* button.
    a. All the Datasets will be expanded in the left-hand tree.
    b. The Thumbnails will be loaded in the centre panel. This allows to browse 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. When you hover over the area used to enter the value, the range is indicated in the tooltip.
    b. Then enter < 2. Then = 0 (some images have no ROIs on them)
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 *Area* item and drag the slider to filter the Images. Note that *PCNT* has the largest number of Images with large ROIs.



8. Switch to Table layout (middle button) .

9. In the selection box *Add table data*…, select
    a. *Table_Image*
    b. Table_Area
    c. Table_Points

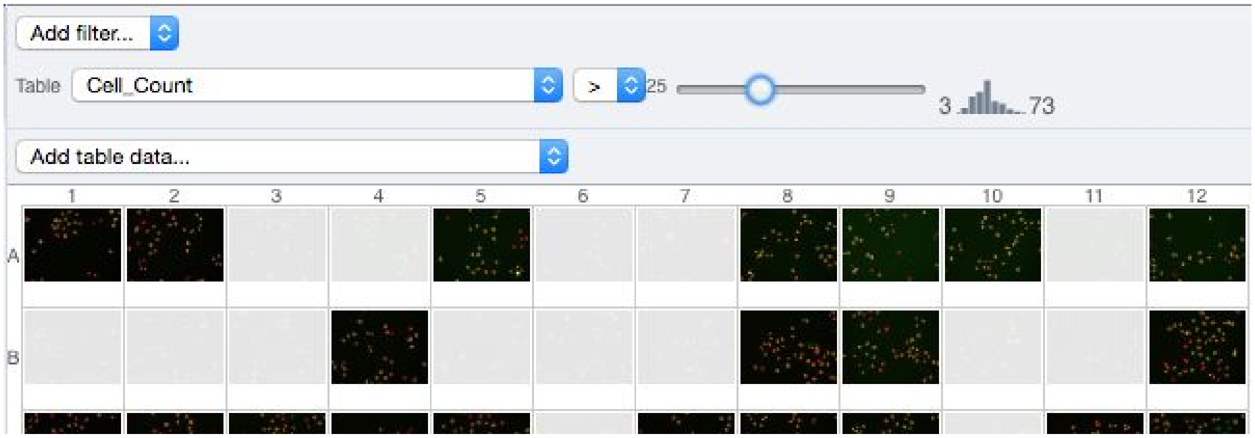
    Note that it is currently not possible to remove a column.
10. Click on the name of a column to sort it.
11. Uncheck *Show Datasets* to sort all Images together e.g. by ROI count.

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…*

Let's now look at the results generated by CellProfiler

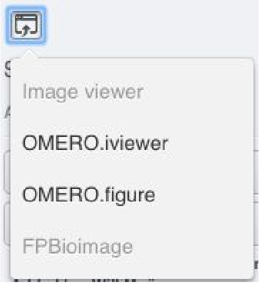
22. Return to the webclient and select the Plate named **plate1_1_013_previously_analyzed** (you have to navigate to trainer-1's data)..
23. Select a Well and open the *Tables* pane in the General tab in the right-hand panel. This will show all the CellProfiler values for this Well.
24. In the *Thumbnails* chooser at the top-right of the centre panel, select the *Parade* plugin.
25. At the top-left of the centre panel choose *Add filter…* -> *Table* to filter Wells by the data from CellProfiler.
26. Change the filter from *ImageNumber* to *Cell_Count* (at the bottom of the list).
27. Now you can use a slider to filter Wells by Cell Count.



OMERO.figure: Use metadata in figure creation

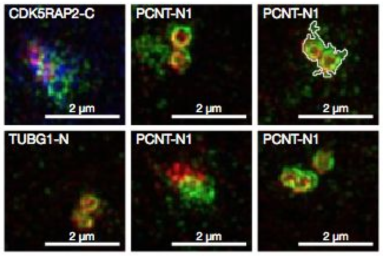
1. In the webclient, select the **idr0021** Project, filter by e.g. *Area* and select several filtered images.

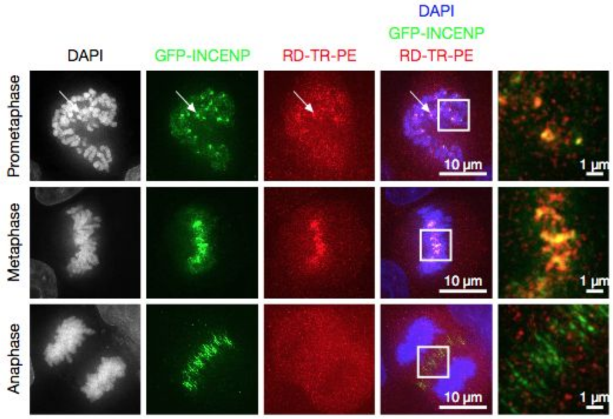
2. In the right-hand panel, click the *Open with...* button and choose *OMERO.figure*:



3. This will open these images in OMERO.figure in a new browser Tab.
4. Drag to arrange the Images into two rows, select all and click the *snap to grid* button at the top of the page.
5. Select all Images and Zoom in around ~300%
6. Go to the *Labels* tab, select all Images and add a Scalebar of 1 µm and adjust the size of the Label to 12.
7. Add label from Dataset name: *color=white location=top-left*
8. Select one image. In the *Labels* tab, click the *Edit* button for ROIs.
9. Load ROIs from OMERO and mouse-over the list to pick the largest ones.
10. Click to add it to the image and click *OK*.
11. Return to the webclient tab, select the **siRNAi-HeLa** Dataset and filter by Rating of 5 stars.
12. Select 3 images and in the right-hand panel click the link icon then copy the link.
13. Return to the OMERO.figure tab, click *Add Image* button and paste the link into dialog. Click OK.
14. Arrange the 3 images into a vertical column, select all and click *snap to grid* button.
15. Copy the 3 images and paste (keyboard shortcut or *Edit* > *Copy/Paste*) 3 times to create 3 more columns.
16. Select the panels in the first column and adjust the rendering settings: Turn only the first channel on and set the color to white.
17. Repeat for the next 2 columns: 1 channel turned on for each column, adjusting the levels if desired, leaving the 4th column as ‘merged’ with multiple channels on.
18. Select all panels and zoom a little. Then select a row and drag the image in the *Preview* tab to pan.
19. Copy and paste the **‘merged’** column again to create a 5th column. Zoom in to approx 500%.
20. Select the *Labels* tab, select the ‘**merged’** and **zoomed** columns and click *Show* Scalebar button.
21. Click the *Label* checkbox to add label to the scalebar. Select only the zoomed-in panels and change the scalebar to 1 micron.
22. Select the top-left panel and enter a label text in the *Add Labels* form.
23. Choose the label size, position (left vertical) and color and click *Add* to create label.
24. Now undo this (*Ctrl-Z* or *Edit* > *Undo*) and instead create labels from Tags:

25. Select the first column of panels and choose *Tags* from the label text-field drop-down options. Click Add.
26. Select the first row of panels and create the labels in the ‘top’ position using the *Channels* option.
27. Edit the created labels located at the bottom of the *Info* tab to rename the green labels to **GFP-INCENP**.
28. Select just the first ‘**merged**’ image and click the ROIs *Edit* button in the *Info* tab.
29. Draw arrows or other shapes on the image, or load ROIs from OMERO. Click *OK*.
30. Click *Copy* ROIs in the *Info* tab, select the other panels in the same row and click *Paste* to add ROIs to these panels.
31. 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.
32. Now select the zoomed-out **‘merged’** panel, and paste this region as an ROI by clicking *Paste* under ROIs section of the *Labels* tab.
33. Repeat for other rows of the figure.





34. Go to *File* > *Paper Setup…* and in the dialog that pop up choose *Pages: 2.* Click *OK*.
35. Finally return to the webclient, select ‘Big’ images from the **svs** Dataset, copy the link to them and paste into the *Add Image* dialog in OMERO.figure.
36. Move the big images to the 2nd page.
37. In the header, click on the *Save* button to save the Figure as “Figure 1”.
38. To open other saved files, go *File* > *Open…*
39. We can view figures from our collaborators here and filter by name or Owner.

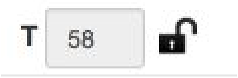
40. Open the **Aurora-B figure 2** from trainer-2.
41. Select a panel and click on the *Webclient* link in the *Info* tab to show the image in the webclient.
42. Back in OMERO.figure, go to *File* > *Open...* to choose the “Figure 1” file saved above.
43. Click on *Export PDF* to export it as PDF.
44. Download the PDF and open it. If opened in a suitable application e.g. Illustrator, the elements on the page will still be editable.

Exercise 7

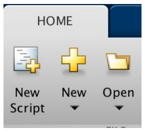
Once again we will use examples from
https://docs.openmicroscopy.org/latest/omero/developers/Matlab.html

The aim of this exercise is to write a simple FRAP script.

1. Go to https://outreach.openmicroscopy.org
2. Log in using the credentials provided.
3. Find the Dataset **FRAP.**
4. Note the ID of the first image in this Dataset.
5. Double-click on the first Image from this Dataset to open it in OMERO.iviewer.
6. Play the timelapse in OMERO.iviewer.
7. Draw in OMERO.iviewer an ellipse ROI on the spot which was bleached.
8. We will use an Ellipse saved on the Image in OMERO.iviewer **without T index set** (spans all T).
9. Click on the lock next to the T Box to unlock .



10. Save the ROI.
11. We are now ready to start the script
12. Launch Matlab
13. Create a new Script, name it for example frap`.m.`



14. Create a connection.
15. Load the dataset **FRAP** with the images.
16. Iterate through the images
17. For each image:
    a. Find the number of timepoints
    b. Load the ROIs linked to the Image.
    c. Iterate through the shapes to only keep the ellipses.Only keep the Ellipses
    d. Analyse the ellipses all the timepoints and `z=0` and `c=0`

   e. Create three arrays: `keys`, `values` and `means`. Keys will store the timepoint as a string, values the mean as a string, and means the mean value for each analyzed timepoint.
   f. Create a map annotation using the keys and values arrays and link it to the image.
   g. Create a CSV file with headers `Image_name,ImageID,Timepoint,Mean`
   h. Create a file annotation for the CSV and link it to the image
   i. Plot the results using the means array
   j. Save the plot as a PNG.
   k. Upload the PNG as a file annotation. Link it to the image

18. Go to https://outreach.openmicroscopy.org
19. Select an Image with ROI in the dataset **FRAP**
20. Check on the right-hand side that the map annotation has been created and that the CSV and PNG files have been attached

Solutions

Exercise 1

```
host='outreach.openmicroscopy.org';
% To be modified
user='USERNAME';
password='PASSWORD';
client = loadOmero(host);
client.enableKeepAlive(60);
% Create an OMERO session
session = client.createSession(user, password);
% Retrieve the group's identifier
eventContext = session.getAdminService().getEventContext();
groupId = eventContext.groupId;
disp(groupId);
```

Exercise 2

```
host='outreach.openmicroscopy.org';
% To be modified
user='USERNAME';
password='PASSWORD';
client = loadOmero(host);
client.enableKeepAlive(60);
% Create an OMERO session
session = client.createSession(user, password);
```

```
% Load the Dataset/Images
dataset = getDatasets(session, datasetId, true);
datasetName = dataset.getName().getValue();
images = toMatlabList(dataset.linkedImageList);
% Iterate through the images
for i = 1 : numel(images)
      image = images(i);
      imageId = image.getId().getValue();
      disp(image.getName().getValue());
      disp(imageId);
end
```

Exercise 3

```
host='outreach.openmicroscopy.org';
% To be modified
user='USERNAME';
password='PASSWORD';
client = loadOmero(host);
client.enableKeepAlive(60);
% Create an OMERO session
session = client.createSession(user, password);

% Load the Dataset/Images
dataset = getDatasets(session, datasetId, true);
datasetName = dataset.getName().getValue();
images = toMatlabList(dataset.linkedImageList);
% Iterate through the images
for i = 1 : numel(images)
      image = images(i);
      imageId = image.getId().getValue();
      % Load the channels information to determine the channel to analyze
      channels = loadChannels(session, image);
      channelIndex = 0;
      for j = 1 : numel(channels)
          channel = channels(j);
          channelId = channel.getId().getValue();
          channelName = channel.getLogicalChannel().getName().getValue();
          % Determine the index of the channel to analyze
          if contains(char(datasetName), char(channelName))
```

```
                channelIndex = j-1; % OMERO index starts at 0
                break
          end
      end
      disp(channelIndex);
end
```

Exercise 4

```
host='outreach.openmicroscopy.org';
% To be modified
user='USERNAME';
password='PASSWORD';
datasetId = 23953;
online_algo = false;

client = loadOmero(host);
client.enableKeepAlive(60);
% Create an OMERO session
session = client.createSession(user, password);
% Initialize the service used to save the Regions of Interest (ROI)
iUpdate = session.getUpdateService();
% Load the Dataset/Images
dataset = getDatasets(session, datasetId, true);
datasetName = dataset.getName().getValue();
images = toMatlabList(dataset.linkedImageList);
% Iterate through the images
values = zeros(numel(images), 2);
for i = 1 : numel(images)
      image = images(i);
      imageId = image.getId().getValue();
      % Load the channels information to determine the channel to analyze
      channels = loadChannels(session, image);
      channelIndex = 0;
      for j = 1 : numel(channels)
          channel = channels(j);
          channelId = channel.getId().getValue();
          channelName = channel.getLogicalChannel().getName().getValue();
          % Determine the index of the channel to analyze
          if contains(char(datasetName), char(channelName))
              channelIndex = j-1; % OMERO index starts at 0
              break
          end
      end
```

```
z = 0;
t = 0;
% Load the plane, OMERO index starts at 0. sizeZ and SizeT = 1
plane = getPlane(session, image, z, channelIndex, t);
if ~online_algo
    threshNstd = 6;
    minPixelsPerCentriole = 20;   % minimum size of objects of interest
    vals = reshape(plane, [numel(plane), 1]);   % reshape to 1 column
    mean1 = mean(vals);
    std1 = std(vals);
    % images are mostly background,
    % so estimate threshold using basic stats
    thresh1 = mean1 + threshNstd * std1;
    bwRaw = imbinarize(plane, thresh1);
    % remove small objects
    BWfinal = bwareaopen(bwRaw, minPixelsPerCentriole);
    fig = figure; imshow(BWfinal), title('segmented image');
 else
    method = 'roberts';
    factor = 1;
    [~, threshold] = edge(plane, method);
    BWs = edge(plane, method, threshold*factor);
    se90 = strel('line', 3, 90);
    se0 = strel('line', 3, 0);
    BWsdil = imdilate(BWs, [se90 se0]);
    BWnobord = imclearborder(BWsdil, 4);
    seD = strel('diamond',1);
    BWfinal = imerode(BWnobord,seD);
    BWfinal = imerode(BWfinal,seD);
    fig = figure; imshow(BWfinal), title('segmented image');
 end

 [B,L] = bwboundaries(BWfinal, 'noholes');
 roi = omero.model.RoiI;
 max_area = 0;
 for k = 1 : length(B)
     boundary = B{k};
     x_coordinates = boundary(:,2);
     y_coordinates = boundary(:,1);
     shape = createPolygon(x_coordinates, y_coordinates);
     setShapeCoordinates(shape, z, channelIndex, t);
     roi.addShape(shape);
     area = polyarea(x_coordinates, y_coordinates);
```

```
           max_area = max(max_area, area);
       end
       values(i, 1) = imageId;
       values(i, 2) = max_area;
       % Link the roi and the image
       roi.setImage(omero.model.ImageI(imageId, false));
       if ~isempty(B)
           roi = iUpdate.saveAndReturnObject(roi);
       end
       close(fig);
end
```

Exercise 5

```
host='outreach.openmicroscopy.org';
% To be modified
user='USERNAME';
password='PASSWORD';
datasetId = 23953;
online_algo = false;

client = loadOmero(host);
client.enableKeepAlive(60);
% Create an OMERO session
session = client.createSession(user, password);
% Initialize the service used to save the Regions of Interest (ROI)
iUpdate = session.getUpdateService();
% Load the Dataset/Images
dataset = getDatasets(session, datasetId, true);
datasetName = dataset.getName().getValue();
images = toMatlabList(dataset.linkedImageList);
% Iterate through the images
values = zeros(numel(images), 2);
for i = 1 : numel(images)
      image = images(i);
      imageId = image.getId().getValue();
      % Load the channels information to determine the channel to analyze
      channels = loadChannels(session, image);
      channelIndex = 0;
      for j = 1 : numel(channels)
          channel = channels(j);
          channelId = channel.getId().getValue();
          channelName = channel.getLogicalChannel().getName().getValue();
```

```
    % Determine the index of the channel to analyze
    if contains(char(datasetName), char(channelName))
        channelIndex = j-1; % OMERO index starts at 0
        break
    end
end
z = 0;
t = 0;
% Load the plane, OMERO index starts at 0. sizeZ and SizeT = 1
plane = getPlane(session, image, z, channelIndex, t);
if ~online_algo
    threshNstd = 6;
    minPixelsPerCentriole = 20;   % minimum size of objects of interest
    vals = reshape(plane, [numel(plane), 1]);   % reshape to 1 column
    mean1 = mean(vals);
    std1 = std(vals);
    % images are mostly background
    % so estimate threshold using basic stats
    thresh1 = mean1 + threshNstd * std1;
    bwRaw = imbinarize(plane, thresh1);
    % remove small objects
    BWfinal = bwareaopen(bwRaw, minPixelsPerCentriole);
    fig = figure; imshow(BWfinal), title('segmented image');
else
    method = 'roberts';
    factor = 1;
    [~, threshold] = edge(plane, method);
    BWs = edge(plane, method, threshold*factor);
    se90 = strel('line', 3, 90);
    se0 = strel('line', 3, 0);
    BWsdil = imdilate(BWs, [se90 se0]);
    BWnobord = imclearborder(BWsdil, 4);
    seD = strel('diamond',1);
    BWfinal = imerode(BWnobord,seD);
    BWfinal = imerode(BWfinal,seD);
    fig = figure; imshow(BWfinal), title('segmented image');
end

[B,L] = bwboundaries(BWfinal, 'noholes');
roi = omero.model.RoiI;
max_area = 0;
for k = 1 : length(B)
    boundary = B{k};
```

```
          x_coordinates = boundary(:,2);
          y_coordinates = boundary(:,1);
          shape = createPolygon(x_coordinates, y_coordinates);
          setShapeCoordinates(shape, z, channelIndex, t);
          roi.addShape(shape);
          area = polyarea(x_coordinates, y_coordinates);
          max_area = max(max_area, area);
      end
      values(i, 1) = imageId;
      values(i, 2) = max_area;
      % Link the roi and the image
      roi.setImage(omero.model.ImageI(imageId, false));
      if ~isempty(B)
           roi = iUpdate.saveAndReturnObject(roi);
      end
    close(fig);
end

% create a CSV
headers = 'Dataset_name,ImageID,Area\n';
f = [tempname,'.csv'];
fileID = fopen(f,'w');
fprintf(fileID, headers);
for i = 1 : numel(images)
      row = strcat(char(datasetName), ',', num2str(values(i, 1)), ',',
num2str(values(i, 2)));
      fprintf(fileID,'%s\n',row);
end
fclose(fileID);

% Read the CSV and plot ImageID vs Area
data = csvread(f, 1, 1);
col1 = data(:, 1);
col2 = data(:, 2);
scatter(col1, col2)
xlabel('ImageID')
ylabel('Area')

% Create a file annotation and link it to the Dataset
fileAnnotation = writeFileAnnotation(session, f, 'mimetype', 'text/csv',
'namespace', 'training.demo');
linkAnnotation(session, fileAnnotation, 'dataset', datasetId);
```

Link to the script
https://github.com/ome/training-scripts/blob/v0.4.1/practical/matlab/analyze_dataset_save_rois_and_attachments.m

Exercise 6: https://github.com/ome/training-scripts/blob/v0.4.1/practical/matlab/idr0021.m
Exercise 7: https://github.com/ome/training-scripts/blob/v0.4.1/practical/matlab/simple_frap.m