The walkthrough is available at
https://downloads.openmicroscopy.org/presentations/2018/GBI-Sydney/

Description

We will demonstrate a number of features of the OMERO platform using an OMERO server based in Dundee, U.K. You can connect to the same server using login details provided to try the software for yourselves. For this workshop, images have already been imported into OMERO. The import instructions can be found at http://help.openmicroscopy.org/importing-data-5.html

Desktop client Import (not presented, for information only)

This section describes one way of importing data in an OMERO server using the Desktop GUI client OMERO.insight.

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

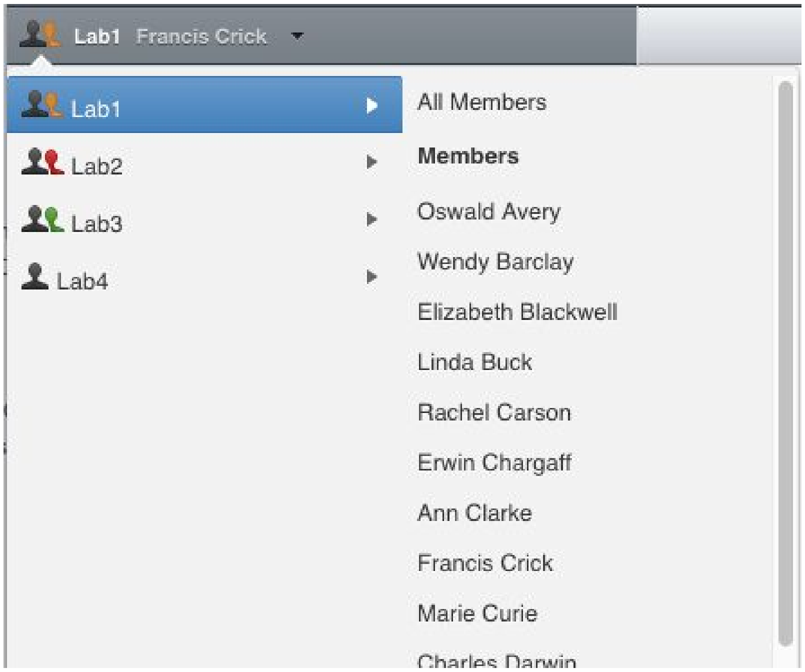
OMERO.server only, and can be finished while the next image(s) is/are being uploaded to the server.

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

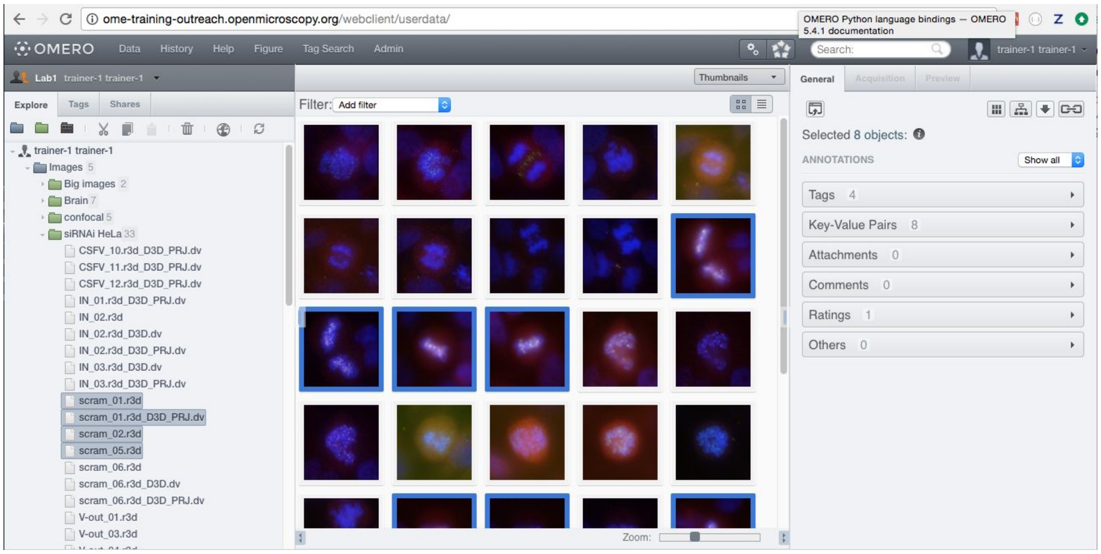
OMERO offers various levels of permissions for groups and users. Depending on the permissions level of each group, a given user might be able to view, annotate or edit data belonging to other users. Permissions are managed by users with admin privileges. For this training session, we will work in a "Read-Annotate" group. This implies that users can view and annotate each other's data but **cannot** delete other people's data.

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



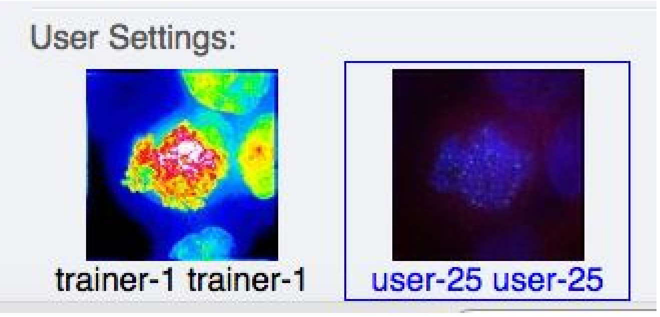
4. You can browse 'folders' in the left-hand pane: *Image* folders are called *Datasets* and they are within *Projects*.

5. When a Dataset is selected, Image thumbnails are shown in the centre panel.



6. These represent imported Images. The original Images are stored on the server and the generated thumbnails allow us to browse them.
7. 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.
8. 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.
9. In the *Preview* tab in the right-hand panel, you can also view the Image.
10. For multi-plane images, sliders allow you to move through Z or Time dimensions.
11. 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.
12. You can adjust the rendering settings for each channel e.g. turn on/off the channels, adjust color settings, look-up tables, etc..
13. The rendering settings can be saved to the server. This **NEVER** changes the original Image data and can be reverted at any time.
14. 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.
15. 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

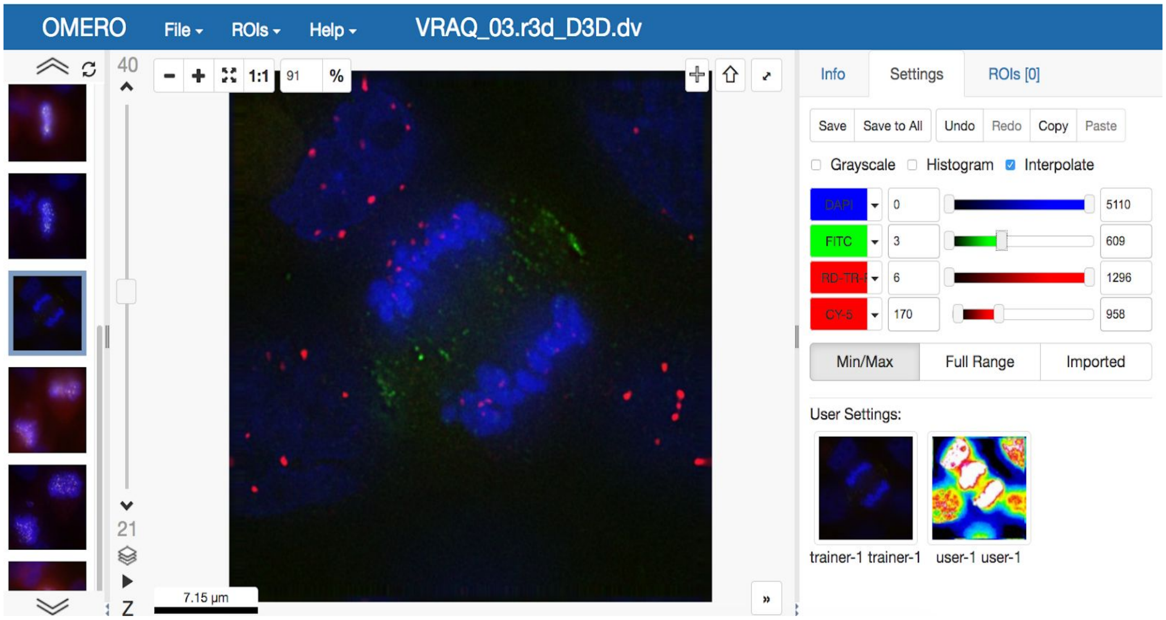
Preview pane. .



16. Your own settings are highlighted in blue.
17. 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.

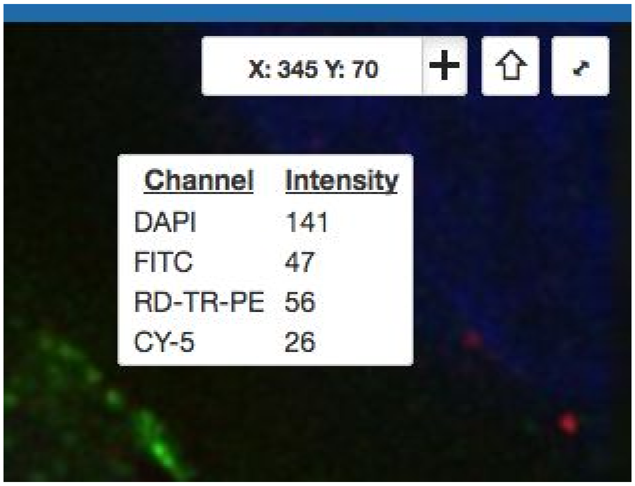


□ Grayscale □ Histogram ☑ Interpolate

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

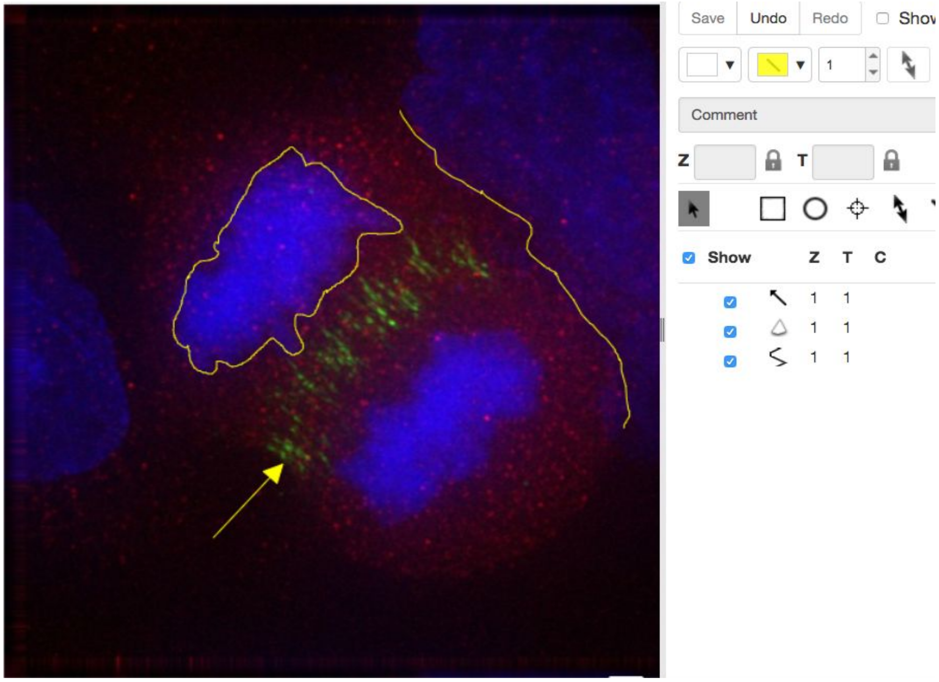
3. Click the *Crosshairs* icon ✛ at the top-right of the viewer to enable the pixel intensity display for the mouse pointer. Then mouse over the Image to see the pixel intensities

for the channels turned on.



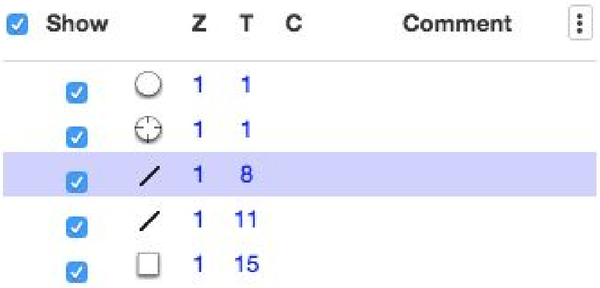
Channel	Intensity
DAPI	141
FITC	47
RD-TR-PE	56
CY-5	26

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).



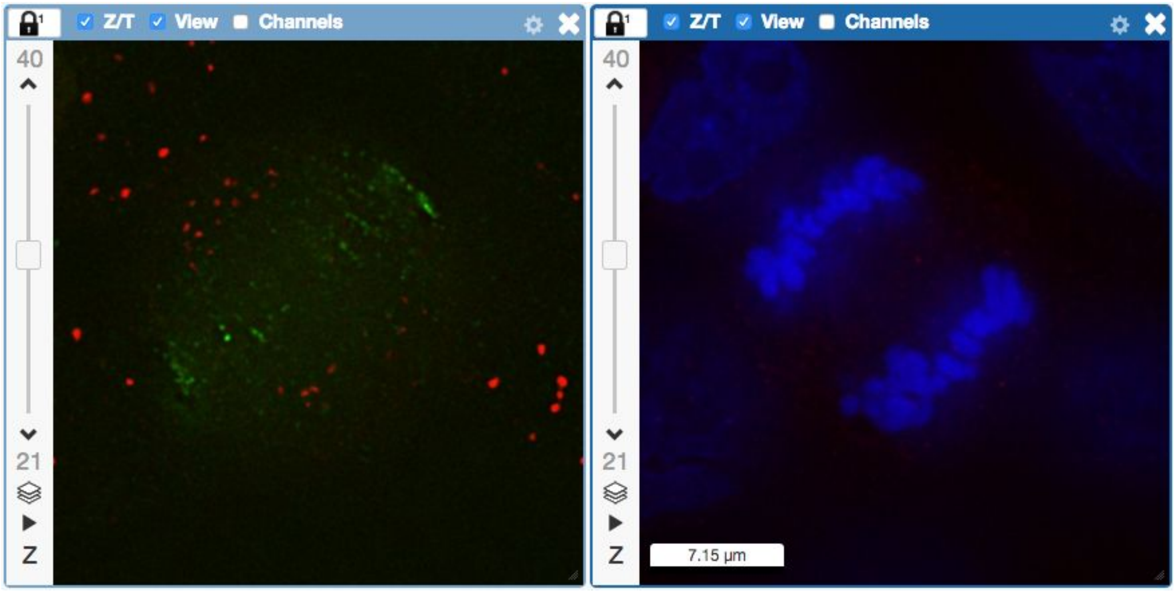
☑ Show		Z	T	C
☑	↖	1	1	
☑	△	1	1	
☑	≤	1	1	

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.



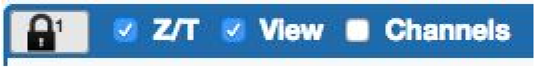
☑ Show		Z	T	C	Comment
☑	○	1	1		
☑	⊕	1	1		
☑	/	1	8		
☑	/	1	11		
☑	□	1	15		

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. .

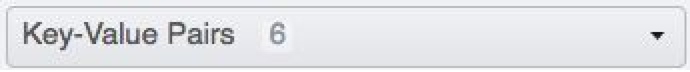


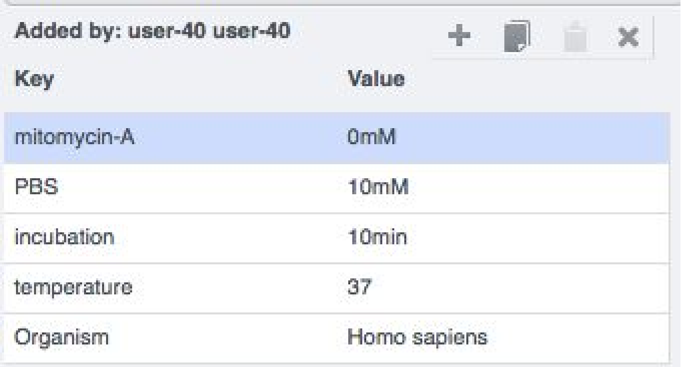
11. Click on *Info* tab in iviewer now, and find, in the right-hand pane, the link to *Dataset:* . Click on that link. This will bring you back to the webclient.

Dataset: siRNAi-HeLa

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*.

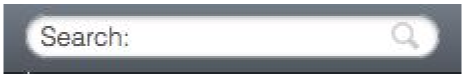




Added by: user-40 user-40

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

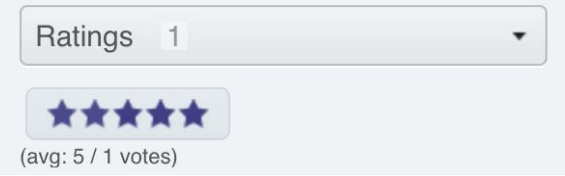
7. The *Key-Value Pairs* allow you to add lab-book like additional metadata for the Image. These *Key-Value Pairs* are also specifically searchable.
8. It is also possible to add Comment, Attachment and Rating to selected Images.
9. Enter ***mitomycin-A*** into the search box in the top right corner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



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

★★★★★

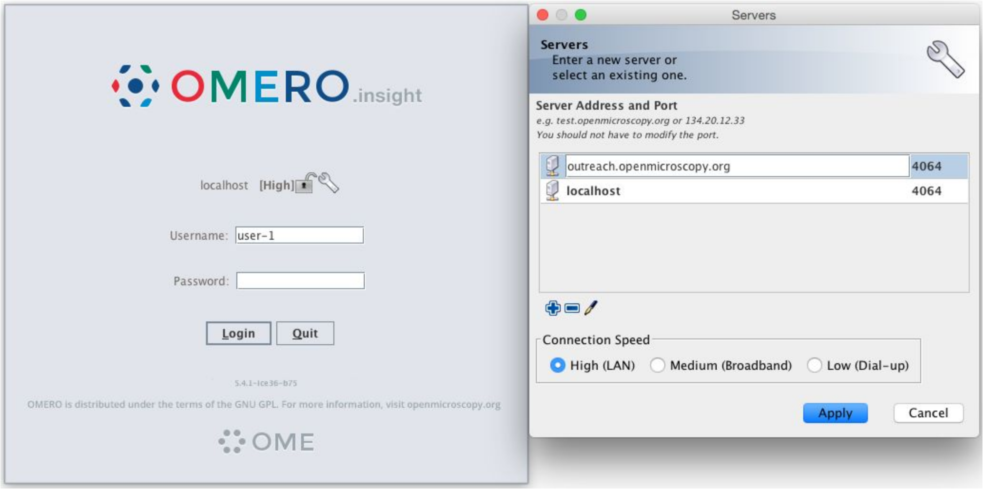
(avg: 5 / 1 votes)

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

Analyze with ImageJ/Fiji: Manual workflow

In this manual workflow section, we use ImageJ/Fiji and use the OMERO ImageJ plugin to analyze Images stored in OMERO. Instructions on how to install the OMERO plugin for ImageJ can be found at http://help.openmicroscopy.org/imagej.html.

1. Go to *Plugins* > *OMERO* > *Connect To OMERO.* This will show a login screen where you can enter the name of the server to connect to, the username and password. The OMERO plugin will allow you to browse your data in a similar manner to the webclient



Username: user-1

Password:

5.4.1-ice36-b75

OMERO is distributed under the terms of the GNU GPL. For more information, visit openmicroscopy.org

Servers

Enter a new server or select an existing one.

Server Address and Port

e.g. test.openmicroscopy.org or 134.20.12.33

You should not have to modify the port.

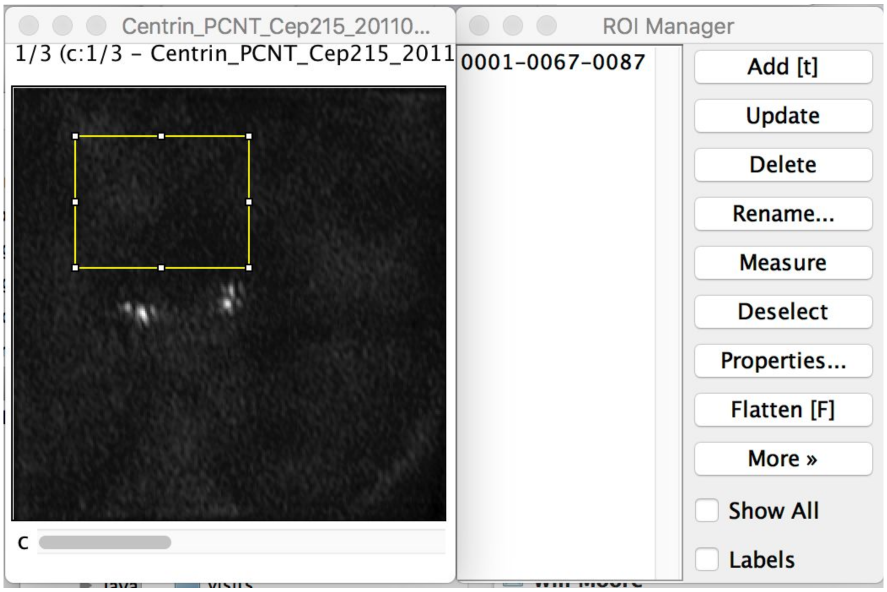
outreach.openmicroscopy.org 4064

localhost 4064

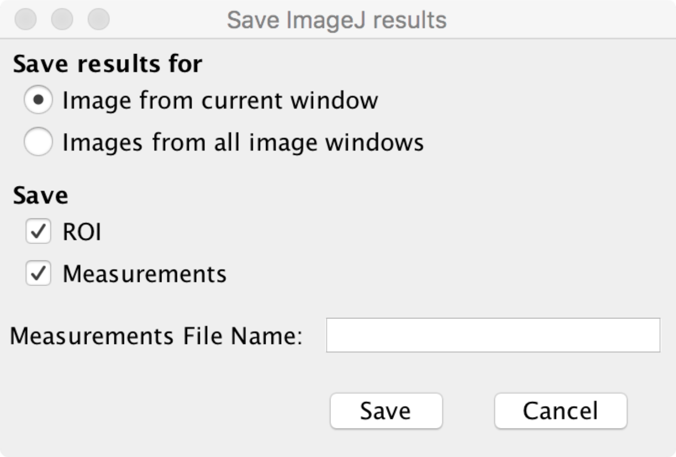
Connection Speed

● High (LAN) ○ Medium (Broadband) ○ Low (Dial-up)

2. Select the **A-Fiji-dataset** Dataset.
3. Double-click on a thumbnail or on an Image in the left-hand tree to open an Image in ImageJ.
4. Go to *Analyze > Tools > ROI Manager…*
5. Draw a shape using for example the Freehand selection tool
6. In the ROI manager, click the button *Add [t]* to add the shape to the Manager.



7. Move to another channel, using the *c* slider
8. Draw other shapes if desired. Add drawing a shape. Click *Add [t]* to add it to the Manager.
9. When done with the drawing, click the button *Measure* in the ROI Manager
10. A dialog with measurements for each shape pops up.
11. To save the ROI and the measurement back to OMERO, go to *Plugins* > *OMERO* > *Save ROIs To OMERO*
12. In the dialog popping up, under the *Save* section select *ROI* and *Measurements.*
13. The measurements are saved as a CSV file.



Save ImageJ results

Save results for

● Image from current window

○ Images from all image windows

Save

☑ ROI

☑ Measurements

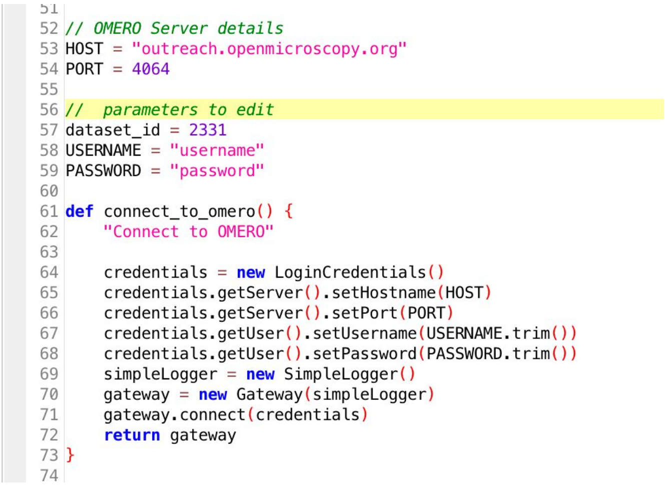
Measurements File Name:

14. Go back to the webclient, in the right-hand panel, check that there is a new CSV file under the *Attachments* harmonica
15. Open the image in iviewer to the see the ROIs and make sure that you can interact with them.

Analyze with ImageJ/Fiji: Scripting workflow

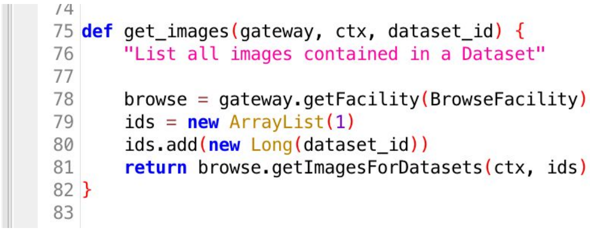
Analysing data manually can quickly become time consuming. In this section we show how we can use the OMERO API and the scripting facility available in Fiji to programmatically analyse data and save the ROIs and result back to the OMERO server.

1. Log in to the webclient using the credentials provided
2. Select the Dataset **A-Fiji-dataset**.
3. Note its ID on the right-hand panel, it will be used below.
4. Launch Fiji. (relaunch if opened).
5. Go to *File* > *New* > *Script*….
6. A dialog pops up. In the *Language* menu select *Groovy.*
7. Copy the content of the script https://raw.githubusercontent.com/ome/training-scripts/v0.3.0/practical/groovy/analyse_dataset_save_rois_and_summary_table.groovy into the text editor.
8. Let’s study the script:
    a. First connect to OMERO



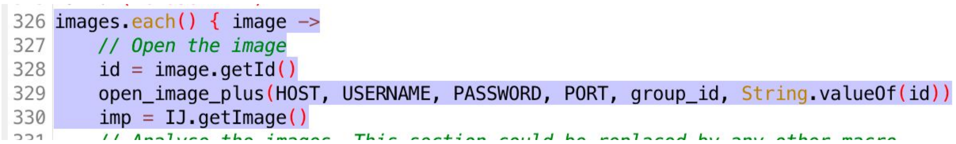
```
// OMERO Server details
HOST = "outreach.openmicroscopy.org"
PORT = 4064

//  parameters to edit
dataset_id = 2331
USERNAME = "username"
PASSWORD = "password"

def connect_to_omero() {
    "Connect to OMERO"

    credentials = new LoginCredentials()
    credentials.getServer().setHostname(HOST)
    credentials.getServer().setPort(PORT)
    credentials.getUser().setUsername(USERNAME.trim())
    credentials.getUser().setPassword(PASSWORD.trim())
    simpleLogger = new SimpleLogger()
    gateway = new Gateway(simpleLogger)
    gateway.connect(credentials)
    return gateway
}

```

b. Load all the images in the specified Dataset

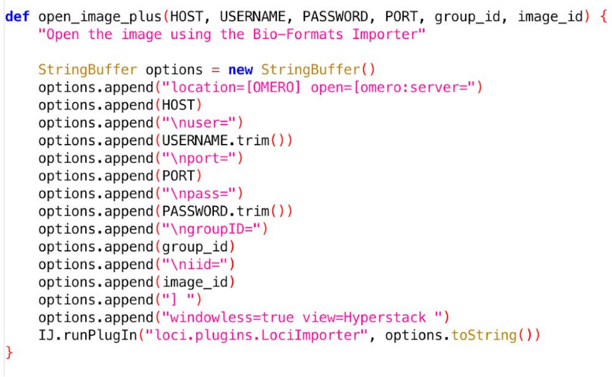


```

def get_images(gateway, ctx, dataset_id) {
    "List all images contained in a Dataset"

    browse = gateway.getFacility(BrowseFacility)
    ids = new ArrayList(1)
    ids.add(new Long(dataset_id))
    return browse.getImagesForDatasets(ctx, ids)
}

```

c. The script loops through all the Images and opens them using Bio-Formats



```
images.each() { image ->
    // Open the image
    id = image.getId()
    open_image_plus(HOST, USERNAME, PASSWORD, PORT, group_id, String.valueOf(id))
    imp = IJ.getImage()
    // Analyse the images. This section could be replaced by any other macro
```



```
def open_image_plus(HOST, USERNAME, PASSWORD, PORT, group_id, image_id) {
    "Open the image using the Bio-Formats Importer"

    StringBuffer options = new StringBuffer()
    options.append("location=[OMERO] open=[omero:server=")
    options.append(HOST)
    options.append("\nuser=")
    options.append(USERNAME.trim())
    options.append("\nport=")
    options.append(PORT)
    options.append("\npass=")
    options.append(PASSWORD.trim())
    options.append("\ngroupID=")
    options.append(group_id)
    options.append("\niid=")
    options.append(image_id)
    options.append("] ")
    options.append("windowless=true view=Hyperstack ")
    IJ.runPlugIn("loci.plugins.LociImporter", options.toString())
}
```

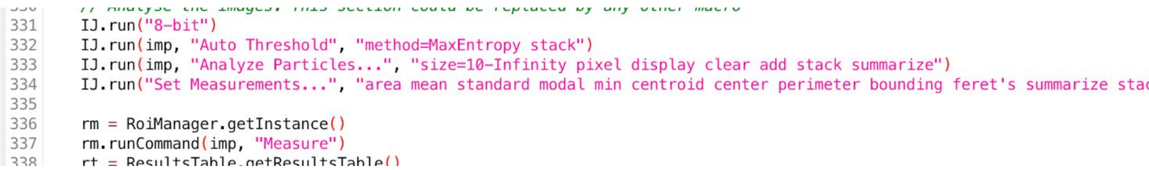
d. Each image is analyzed. The following steps described the analysis performed but this could be adjusted to your specific analysis:

i. Convert floating-point pixel-type to 8-bit (I*mage* > *Type* > *8-bit*)

ii. Apply Threshold with *MaxEntropy* and *Stack* selected (*Image* > *Adjust* > *Auto Threshold)*

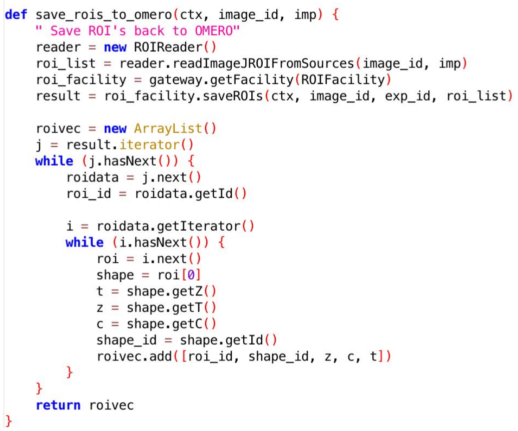
iii. Analyze particles with *Size* set to *10-Infinity* and *Pixel units, Display results, Clear results, Add to Manager s*elected

iv. Add the measurements to the *Results* table



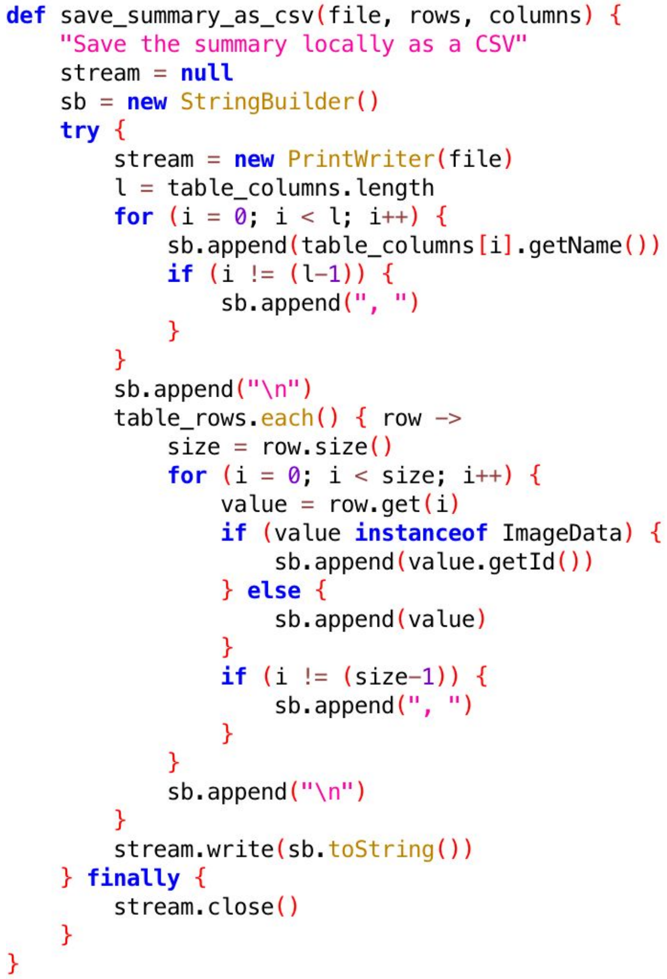
```
    // Analyse the images. This section could be replaced by any other macro
    IJ.run("8-bit")
    IJ.run(imp, "Auto Threshold", "method=MaxEntropy stack")
    IJ.run(imp, "Analyze Particles...", "size=10-Infinity pixel display clear add stack summarize")
    IJ.run("Set Measurements...", "area mean standard modal min centroid center perimeter bounding feret's summarize stac

    rm = RoiManager.getInstance()
    rm.runCommand(imp, "Measure")
    rt = ResultsTable.getResultsTable()
```

e. For each Image, the generated ROIs are saved back to the OMERO server



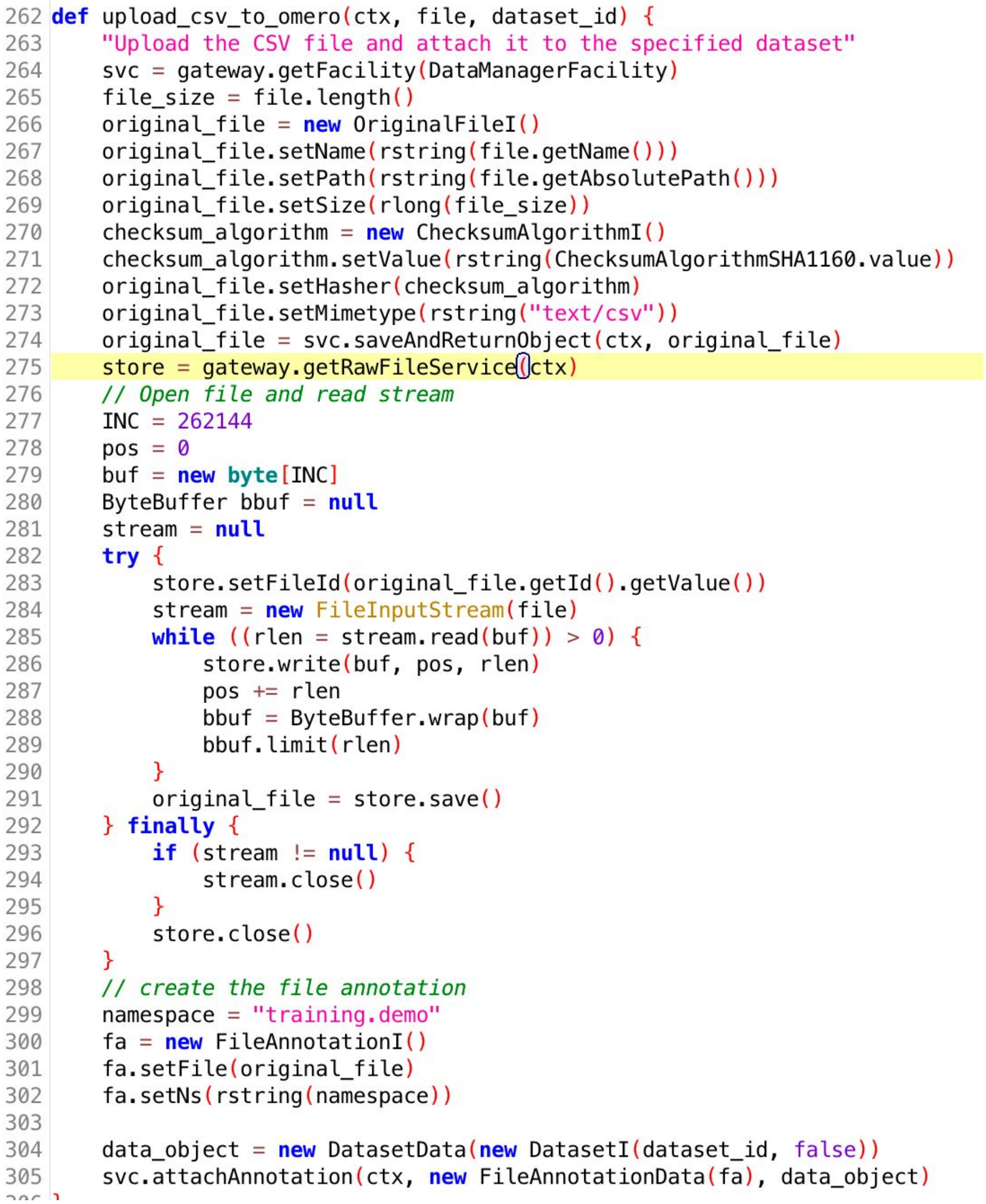
```
def save_rois_to_omero(ctx, image_id, imp) {
    " Save ROI's back to OMERO"
    reader = new ROIReader()
    roi_list = reader.readImageJROIFromSources(image_id, imp)
    roi_facility = gateway.getFacility(ROIFacility)
    result = roi_facility.saveROIs(ctx, image_id, exp_id, roi_list)

    roivec = new ArrayList()
    j = result.iterator()
    while (j.hasNext()) {
        roidata = j.next()
        roi_id = roidata.getId()

        i = roidata.getIterator()
        while (i.hasNext()) {
            roi = i.next()
            shape = roi[0]
            t = shape.getZ()
            z = shape.getT()
            c = shape.getC()
            shape_id = shape.getId()
            roivec.add([roi_id, shape_id, z, c, t])
        }
    }
    return roivec
}
```

f. A summary of all the measurements is generated



```
def save_summary_as_csv(file, rows, columns) {
    "Save the summary locally as a CSV"
    stream = null
    sb = new StringBuilder()
    try {
        stream = new PrintWriter(file)
        l = table_columns.length
        for (i = 0; i < l; i++) {
            sb.append(table_columns[i].getName())
            if (i != (l-1)) {
                sb.append(", ")
            }
        }
        sb.append("\n")
        table_rows.each() { row ->
            size = row.size()
            for (i = 0; i < size; i++) {
                value = row.get(i)
                if (value instanceof ImageData) {
                    sb.append(value.getId())
                } else {
                    sb.append(value)
                }
                if (i != (size-1)) {
                    sb.append(", ")
                }
            }
            sb.append("\n")
        }
        stream.write(sb.toString())
    } finally {
        stream.close()
    }
}
```

g. The CSV file is then attached to the Dataset



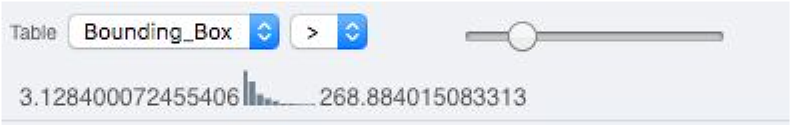
```
def upload_csv_to_omero(ctx, file, dataset_id) {
    "Upload the CSV file and attach it to the specified dataset"
    svc = gateway.getFacility(DataManagerFacility)
    file_size = file.length()
    original_file = new OriginalFileI()
    original_file.setName(rstring(file.getName()))
    original_file.setPath(rstring(file.getAbsolutePath()))
    original_file.setSize(rlong(file_size))
    checksum_algorithm = new ChecksumAlgorithmI()
    checksum_algorithm.setValue(rstring(ChecksumAlgorithmSHA1160.value))
    original_file.setHasher(checksum_algorithm)
    original_file.setMimetype(rstring("text/csv"))
    original_file = svc.saveAndReturnObject(ctx, original_file)
    store = gateway.getRawFileService(ctx)
    // Open file and read stream
    INC = 262144
    pos = 0
    buf = new byte[INC]
    ByteBuffer bbuf = null
    stream = null
    try {
        store.setFileId(original_file.getId().getValue())
        stream = new FileInputStream(file)
        while ((rlen = stream.read(buf)) > 0) {
            store.write(buf, pos, rlen)
            pos += rlen
            bbuf = ByteBuffer.wrap(buf)
            bbuf.limit(rlen)
        }
        original_file = store.save()
    } finally {
        if (stream != null) {
            stream.close()
        }
        store.close()
    }
    // create the file annotation
    namespace = "training.demo"
    fa = new FileAnnotationI()
    fa.setFile(original_file)
    fa.setNs(rstring(namespace))

    data_object = new DatasetData(new DatasetI(dataset_id, false))
    svc.attachAnnotation(ctx, new FileAnnotationData(fa), data_object)
```

9. Click *Run.*
10. Go to the webclient, navigate to one of the images which were analyzed
11. Double-click to open the image in OMERO.iviewer.
12. Demonstrate that the Image has now the polygon ROIs exported from Fiji and these are channel-specific.

Analyze metadata using OMERO.parade

First let’s look at the analytical results generated using the *Analyze particles* plugin in Fiji

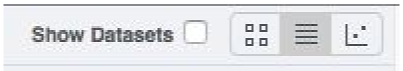
1. Navigate to the data of trainer-1. Select the Project **idr0021**.
2. Choose the *Parade* option in the centre panel dropdown menu.
3. Expand all Datasets by clicking on the *Open All* button.
    a. All the Datasets will be expanded in the left-hand tree.
    b. The Thumbnails will be loaded in the 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 *Bounding_Box* 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_Bounding_Box*
    b. Table_Total_Area
    c. Table_Image

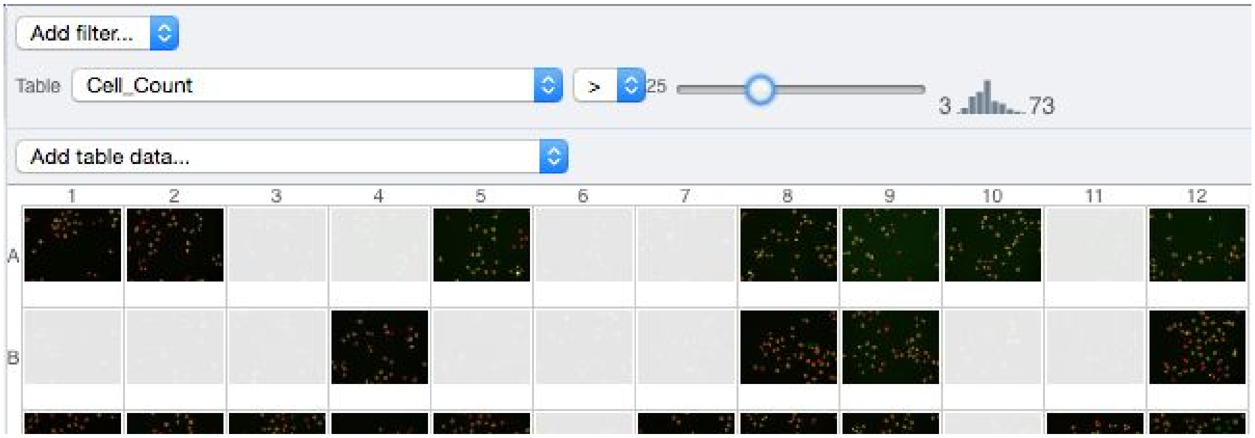
    Note that it is currently not possible to remove a column.
10. Click on the name of a column to sort it.
11. Uncheck *Show Datasets* to sort all Images together e.g. by ROI count.

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…*

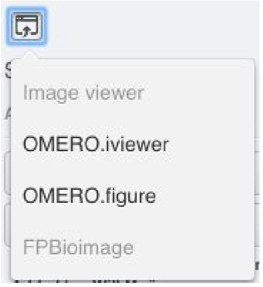
Let’s now look at the results generated by another 3rd party software, CellProfiler

22. Return to the webclient and select the Plate named **plate1_1_013_previously_analysed**
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.



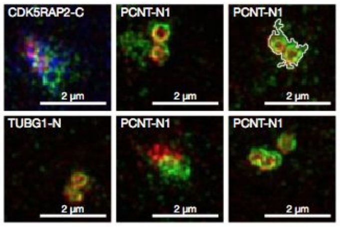
OMERO.figure: Use metadata in figure creation

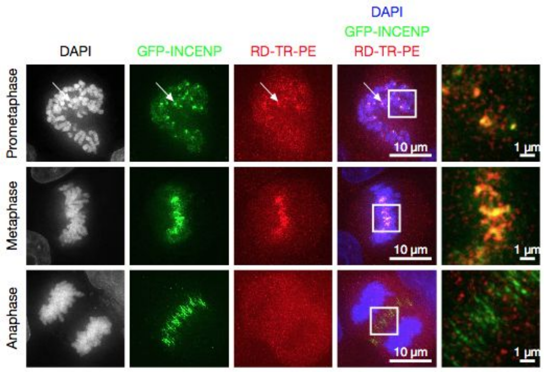
1. In the webclient, select the **idr0021** Project, filter by e.g. Bounding Box and select several filtered images.
2. In the right-hand panel, click the *Open with...* button and choose *OMERO.figure*:



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.