

---

# **OMERO.figure guide Documentation**

***Release 0.1.0***

**Open Microscopy Environment**

**May 27, 2020**



---

## Contents

---

<b>1</b>	<b>Create figures using OMERO.figure</b>	<b>3</b>
<b>2</b>	<b>OMERO.figure scripting</b>	<b>9</b>



OMERO.figure is a popular tool for creating figures from Images in OMERO. Image metadata can be used to facilitate figure creation. For more information, see <https://github.com/ome/omero-figure>.

Contents:



---

## Create figures using OMERO.figure

---

OMERO.figure is a web-based tool for creating figures from Images in OMERO. Image metadata can be used to facilitate figure creation.

### 1.1 Description:

This guide covers:

- Opening images in OMERO.figure to create a new figure
- How to add additional images to an existing figure
- Arranging panels in the desired figure layout
- How to synchronize the rendering settings between panels
- How to add scalebars, labels and ROIs to panels
- How to save and export figures as PDF or TIFF

### 1.2 Setup:

- Install the OMERO.figure web app as described at <https://pypi.org/project/omero-figure/>

### 1.3 Resources:

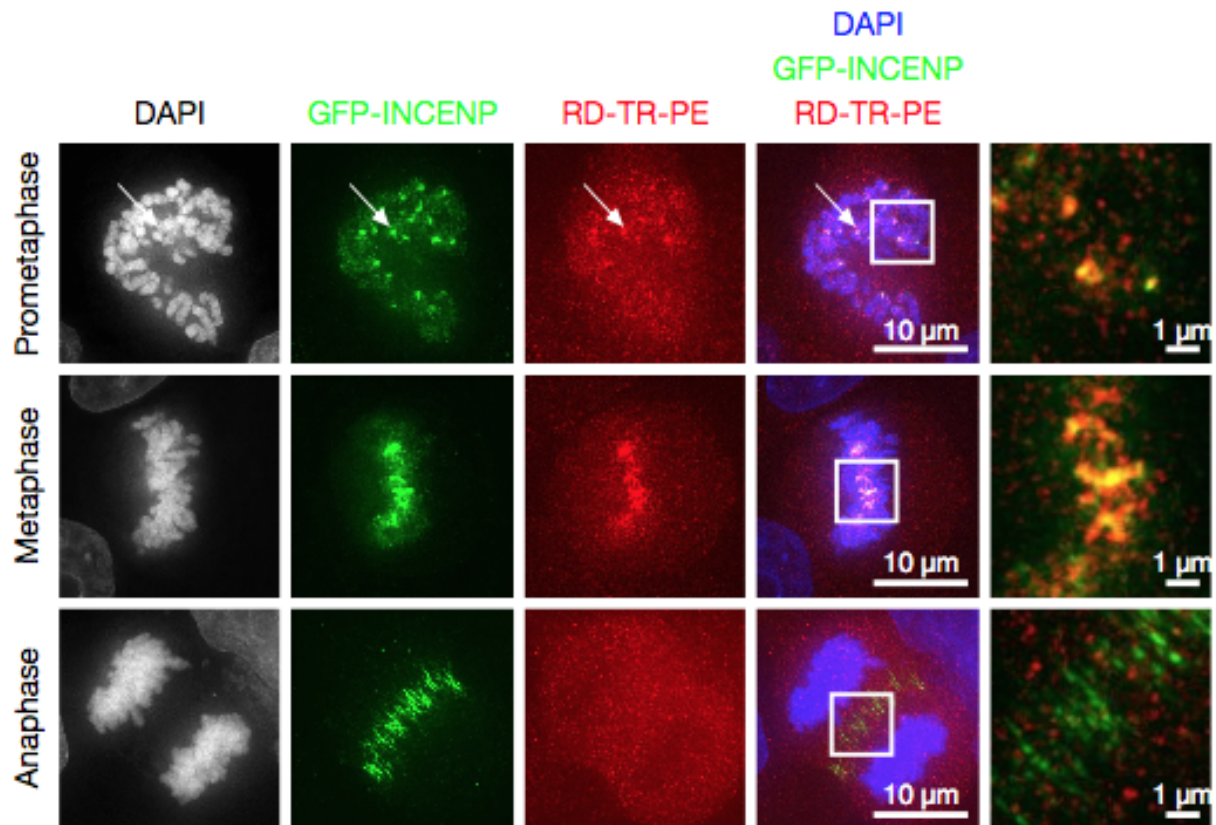
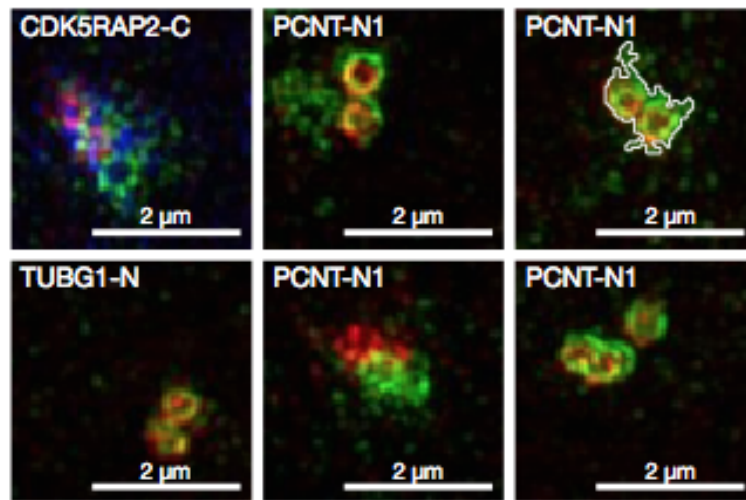
- Sample images from the Image Data Resource (IDR) <https://idr.openmicroscopy.org/webclient/?show=project-51>. See <https://github.com/ome/training-scripts/blob/master/maintenance/preparation/idr0021-data-prep.md> for download and import instructions.
- DV images from <https://downloads.openmicroscopy.org/images/DV/siRNAi-HeLa/>

- SVS ‘big’ pathology images from <https://downloads.openmicroscopy.org/images/SVS/>

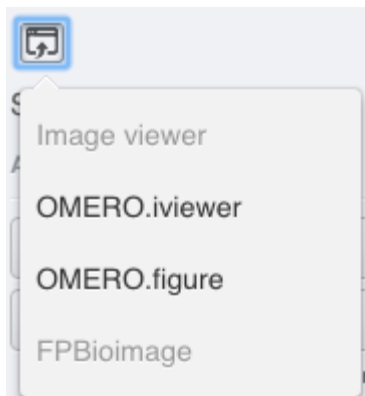
## 1.4 Step-by-Step:


Using the sample images above, we will create a figure like this, but you can use any multi-channel images.

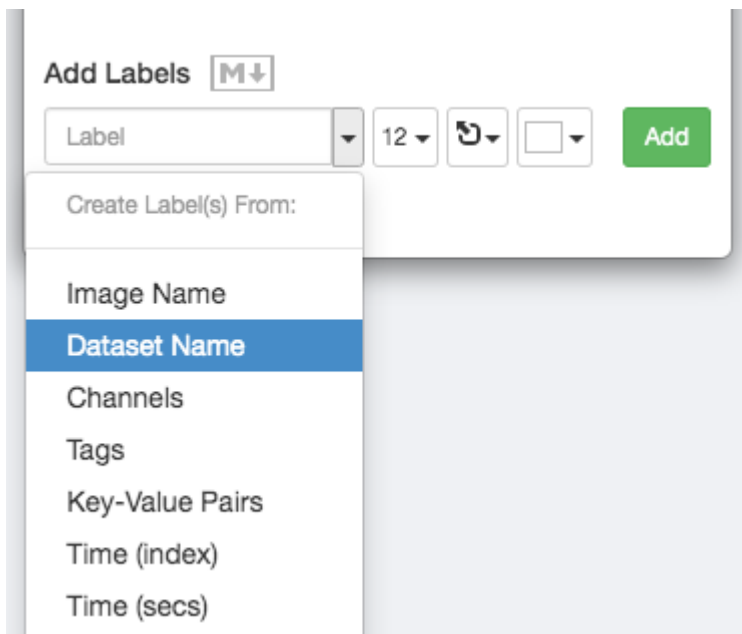




1. In the webclient, select 6 images from the **idr0021** Project.
2. In the right-hand panel, click the *Open with...* button and choose *OMERO.figure*:



3. This will open these images in OMERO.figure in a new browser tab.
4. Drag to arrange the Images approximately into two rows, select all (use Ctrl-A or drag to select) and click the *snap to grid* button  at the top of the page.
5. Select all Images and Zoom in around ~300%, using the Zoom slider in the right *Preview* tab.
6. Go to the *Labels* tab, select all Images and add a Scalebar: Click the *Show* button, choose a length of 2  $\mu\text{m}$ , click the *Label* checkbox and adjust the size of the Label to 12.
7. Add labels: choose *Dataset Name* in the label input dropdown list, choose *color white* and *position=top-left*.



8. Click *Add* to create the new label.
9. Add labels: choose *Key-Value Pairs* in the label input dropdown list, and in the following popup choose the *Key* which you want to add the value of from the dropdown menu.

## Labels from Key-Value Pairs ×

Choose Key (shows numbers of images):

Antibody Target (2)


☐ Include Key in Label

### Example Label

CENT2

Close

OK

10. Click *OK* in that dialog to create the new label.
11. Select one image. In the *Labels* tab, click the *Edit* button for ROIs.
12. If the image has ROIs in OMERO, click *Load ROIs*.
13. Mouse over the list to show each ROI on the Image and click to add it to the Image.
14. Click *OK* to close the dialog.
15. Return to the webclient tab, select the **siRNAi-HeLa** Dataset. N.B.: You may wish to filter the images when selecting those to add to your figure, e.g. Filter by Rating.
16. Select 3 images and in the right-hand panel click the link icon  then copy the link.
17. Return to the OMERO.figure tab, click *Add Image* button and paste the link into dialog. Click *OK*.
18. Arrange the 3 images into a vertical column, select all and click *snap to grid* button.
19. Copy the 3 images and paste (keyboard shortcut or *Edit > Copy/Paste*) 3 times to create 3 more columns.
20. Select the panels in the first column and adjust the rendering settings: Turn only the first channel on and set the color to white.
21. 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.
22. Select all panels and zoom a little. Then select all the panels from one row and drag the image in the *Preview* tab to pan the selected images to the same point.
23. Copy and paste the **merged** column again to create a 5th column. Zoom in to approx 500%.

24. Select the *Labels* tab, select the **merged** and **zoomed** columns and click *Show Scalebar* button.
25. Click the *Label* checkbox to add a label to the scalebar. Select only the zoomed-in panels and change the scalebar to 1 micron.
26. Select the top-left panel and enter a label text in the *Add Labels* form. “Prometaphase” in the example above.
27. Set the label size (14), position (left vertical) and color (black) and click *Add* to create a label.
28. If we have Tags on the images, we can use these to create labels:
29. Select the first column of panels and choose *Tags* from the label text-field drop-down options. Click *Add*.
30. Select the first row of panels and create the labels in the *top* position using the *Channels* option to add Labels for active channels in each image.
31. Edit the created labels located at the bottom of the *Info* tab to rename the green labels to **GFP-INCENP**.
32. Select just the first **merged** Image and click the ROIs *Edit* button in the *Labels* tab.
33. Draw arrows or other shapes on the Image, or load ROIs from OMERO. Click *OK* to close the dialog.
34. Click *Copy ROIs* in the *Labels* tab, select the other panels in the same row and click *Paste* to add ROIs to these panels.
35. 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.
36. Now select the zoomed-out ‘**merged**’ panel, and paste this region as an ROI by clicking *Paste* under ROIs section of the *Labels* tab.
37. Repeat for other rows of the figure. At this point we have created the figure in the screenshot above.

## 1.5 Saving and exporting figures

1. Go to *File > Paper Setup...* and in the dialog that pops up choose *Pages*: 2. Click *OK*.
2. Finally return to the webclient, select ‘Big’ images from the **svs** Dataset, copy the link to them and paste it into the *Add Image* dialog in OMERO.figure.
3. Move the big images to the 2nd page.
4. In the header, click on the *Save* button to save the Figure as “Figure 1”.
5. The URL will update. You can bookmark this URL or share with collaborators to view your figure.
6. To open other saved files, go *File > Open...*
7. We can view figures from our collaborators here and filter by name or Owner.
8. Choose a figure to Open. For example the **Aurora-B figure 2** from trainer-2.
9. Select a panel and click on the *Webclient* link in the *Info* tab to show the image in the webclient.
10. Back in OMERO.figure, go to *File > Open...* to choose the “Figure 1” file saved above.
11. Click on *Export PDF* to export it as PDF.
12. Download the PDF and open it. If opened in a suitable application e.g. Illustrator, the elements on the page will still be editable.

## 2.1 Description:

We can use JavaScript in the browser console to script changes to a figure. This is an experimental feature and not documented. In this example, we will automatically add labels based on analysis results.

## 2.2 Setup:

- Install the OME figure web app as described at <https://pypi.org/project/omero-figure/>
- Upload the script `Split_View_Figure.py` to the OME scripting service

## 2.3 Resources:


- Any multi-channel images, such as <https://downloads.openmicroscopy.org/images/DV/siRNAi-HeLa/>
- Any time-lapse images, such as <https://downloads.openmicroscopy.org/images/DV/will/FRAP/>

## 2.4 Step-by-Step:

### 2.4.1 Figure creation in Python

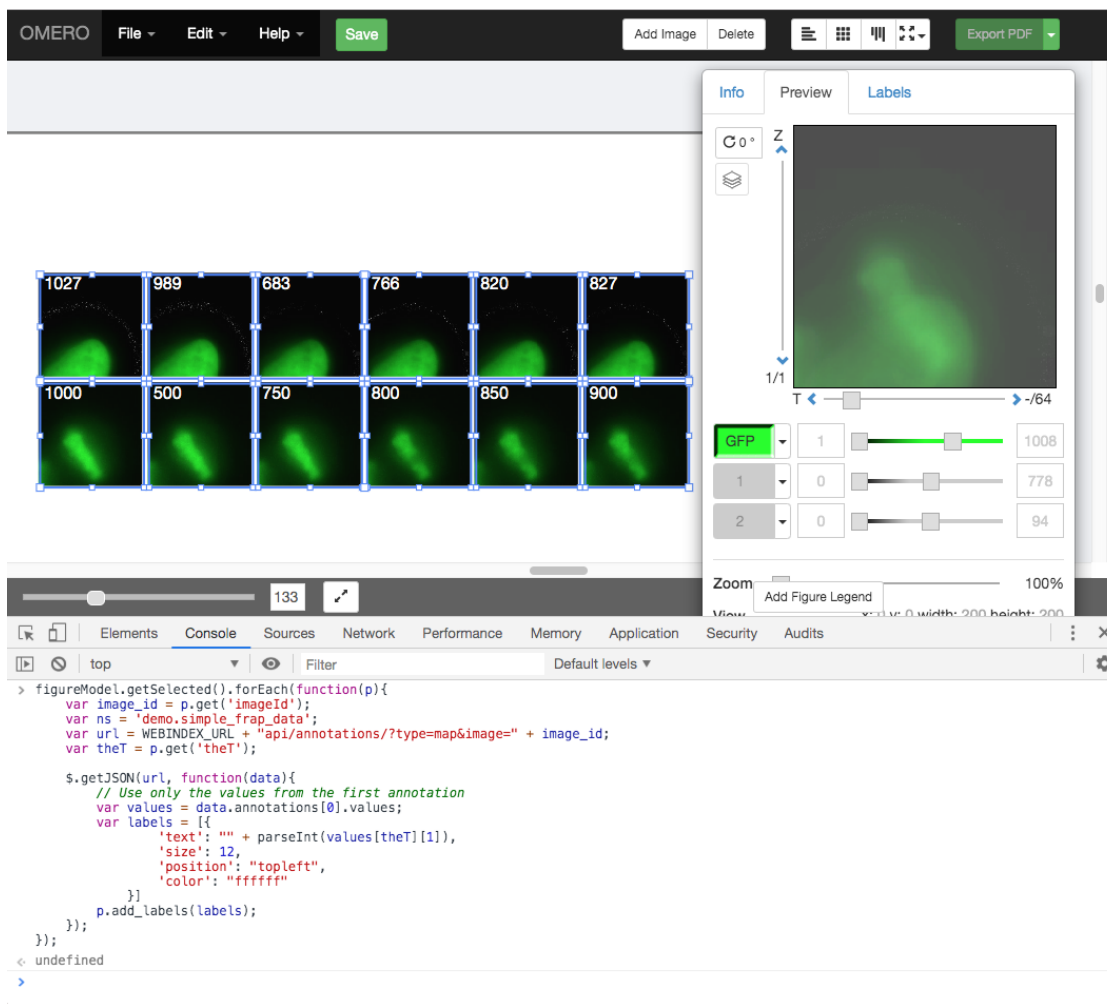
OMERO.figure files are simply JSON data, stored in OME File Annotations with a specific namespace of `omero.web.figure.json`. We can create these files using Python scripts, uploaded to the OME scripting service to make them available to all OME users.

The format of the JSON is described at [https://github.com/ome/omero-figure/blob/master/docs/figure\\_file\\_format.rst](https://github.com/ome/omero-figure/blob/master/docs/figure_file_format.rst). We will use the example `Split_View_Figure.py` script.

1. Select a few images in the webclient.
2. Click on the Script button  in the top-right of the webclient and choose the `Split_View_Figure.py` script (e.g. under Workshop Scripts).
3. Run the script. When complete, open the OMERO.figure app and File > Open.
4. Choose the most recent figure, called “Split View Figure”.

## 2.4.2 Figure editing in JavaScript

We will use the time-lapse images listed above to create a FRAP figure but you can use any time-lapse images.



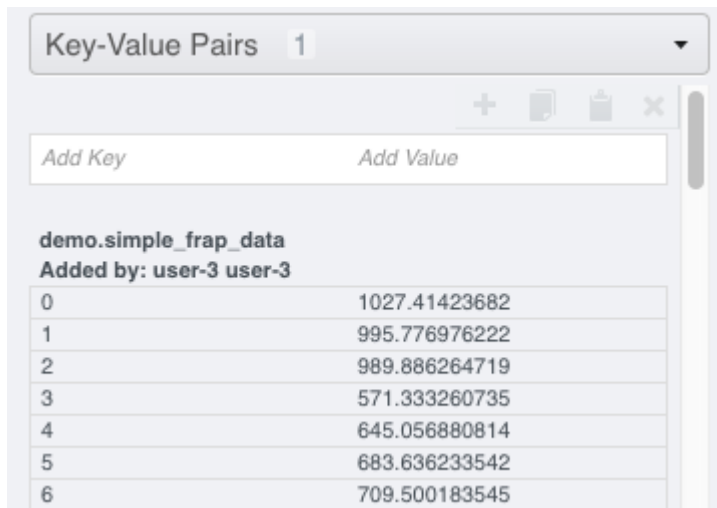
1. To see the data model for any current file in OMERO.figure, go to *File > Export as JSON...*
2. You will see that the `panels` list defines the panels and each panel has attributes. For example, a panel with a single white label might include the following attributes:

```
"name": "image1.tiff",
"labels": [{"text": "label text", "size": 12, "position": "topleft", "color": "FFFFFF"}],
"x": 200, "y": 200, "width": 100, "height": 100,
...many other attributes not shown...
```

3. The `figureModel` variable is accessible in the console. We can use `figureModel.getSelected()` to get selected panels and for each panel we can call `p.set()` to change an attribute.
4. For example, to set the height of each selected panel to 200, we can do:

```
figureModel.getSelected().forEach(function(p) {
  p.set('height', 200)
});
```

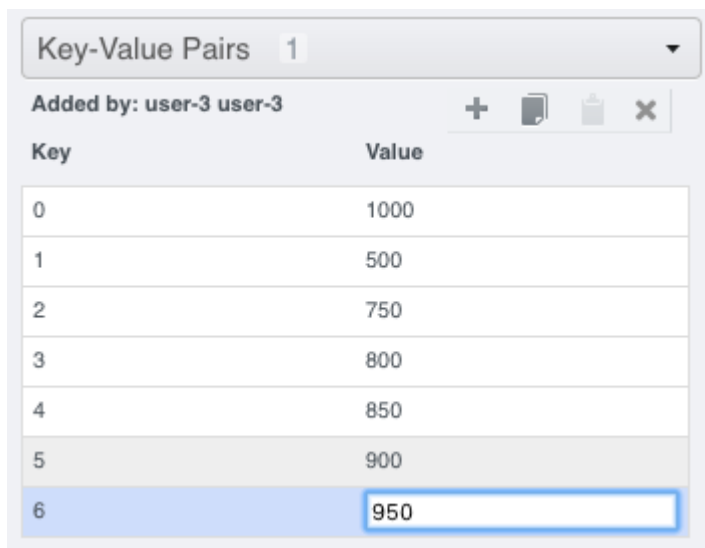
5. We can use AJAX to load JSON data and we will use `p.add_labels()` to create labels.
6. In this example we will load the FRAP intensities from the Map Annotations on these images.
7. Select 2 FRAP images that have previously been analysed to create a Map Annotation with the namespace `demo.simple_frap_data`.



The screenshot shows a web interface titled 'Key-Value Pairs' with a dropdown menu set to '1'. Below the title, there are buttons for '+', a document icon, a trash icon, and a close 'X' icon. There are also input fields for 'Add Key' and 'Add Value'. The main content area displays a table with the following data:

demo.simple_frap_data	
Added by: user-3 user-3	
0	1027.41423682
1	995.776976222
2	989.886264719
3	571.333260735
4	645.056880814
5	683.636233542
6	709.500183545

8. Alternatively, you can add your own Map Annotation with each *Key* being a *T-index* (start at 0), and the *Value* will be a FRAP intensity (number).



The screenshot shows the same 'Key-Value Pairs' interface. The dropdown menu is still '1'. The table now shows data added by 'user-3 user-3'. The values are different from the previous screenshot, and the row for key '6' is highlighted in blue with a text input field containing '950'.

Key	Value
0	1000
1	500
2	750
3	800
4	850
5	900
6	950

9. Create a Figure with 2 images.
10. Copy and paste each image several times and increment T-index in the Preview panel to show multiple time-points per image.

11. Open the browser console by *right-click > Inspect Element (Firefox)* or *right-click > Inspect (Chrome)* and click on the *Console* tab.
12. Copy the code from [https://raw.githubusercontent.com/ome/training-scripts/v0.6.0/practical/javascript/figure\\_frap\\_mapannotation\\_label.js](https://raw.githubusercontent.com/ome/training-scripts/v0.6.0/practical/javascript/figure_frap_mapannotation_label.js)
13. Drag to select the FRAP movie images in the figure.
14. Paste the code into the console. **Do not hit enter yet.**
15. Inspect the code. It will iterate through each of the **selected** panels, an AJAX call is made to load the Map Annotations with the namespace that we created from FRAP values above.
16. NB: If you manually created your own Map Annotation above, you can remove the line `url += '&ns=' + ns;` to avoid filtering by namespace.
17. The FRAP values are a list of `[key, value]` pairs and we can get the value for the current T index of the panel with `values[theT][1]` and use this to create a label.
18. Hit Enter to run the code on selected panels.
19. The labels should be added. Note that you can undo and redo these changes in the UI as normal.
20. Try out other examples in <https://github.com/ome/training-scripts/tree/v0.6.0/practical/javascript>