

---

# **OMERO.figure guide Documentation**

***Release 0.1.0***

**Open Microscopy Environment**

**May 10, 2023**



---

## 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>
<b>3</b>	<b>Move Figures between Groups</b>	<b>17</b>
<b>4</b>	<b>Contribute</b>	<b>21</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

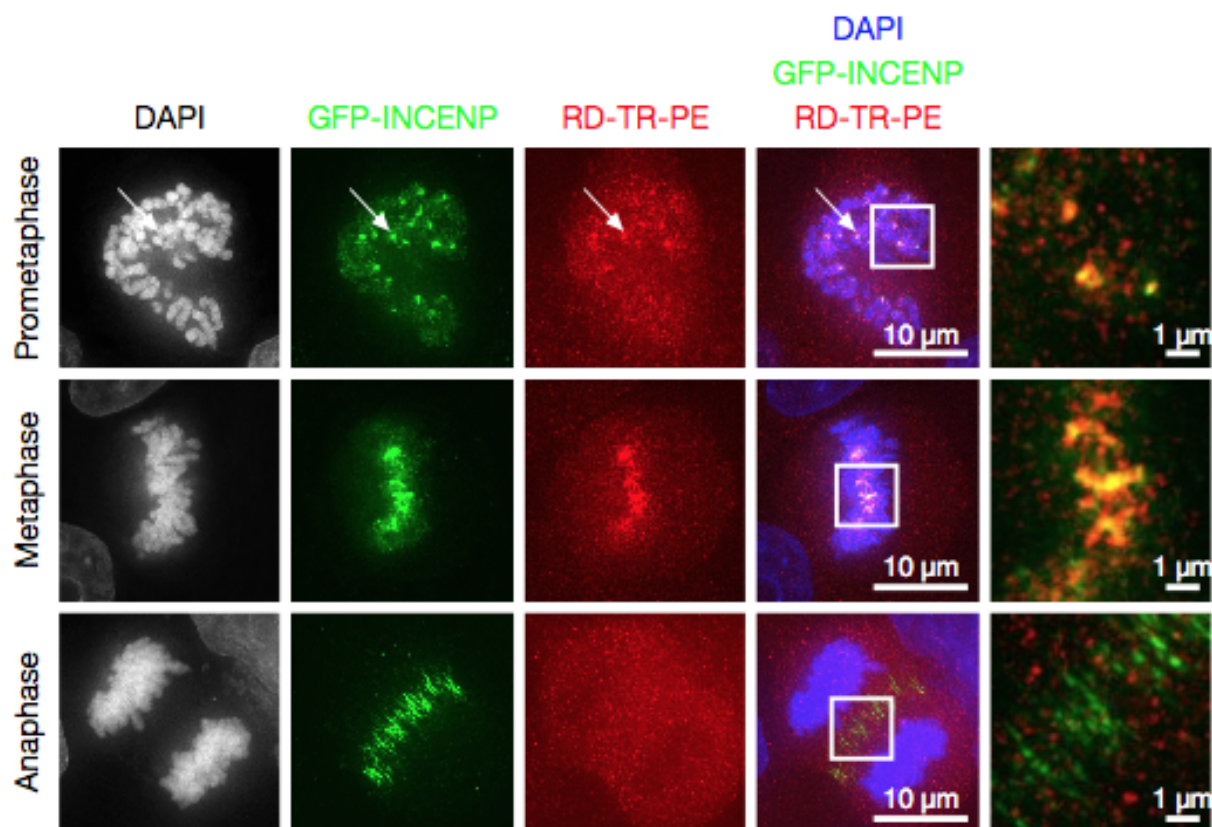
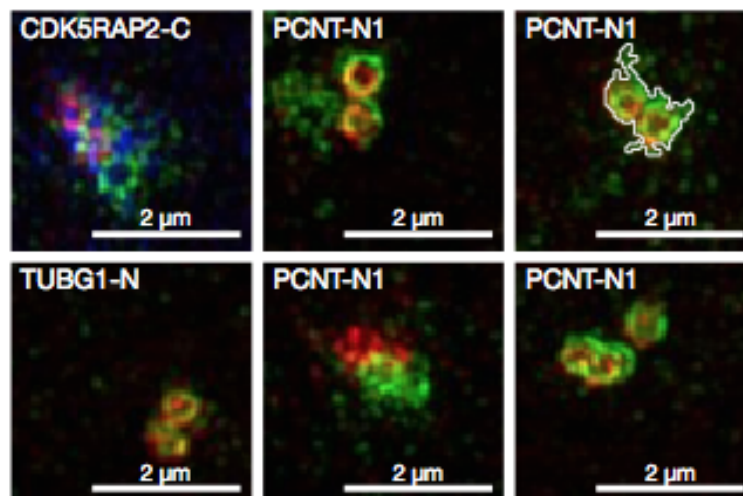
- Install the OMERO.figure web app as described at <https://pypi.org/project/omero-figure/>

### 1.3 Resources

- Sample images from the Image Data Resource (IDR) [idr0021](#). See [idr0021-data-prep.md](#) for download and import instructions.
- DV images from [siRNAi-HeLa](#).
- SVS ‘big’ pathology images from [SVS](#).

## 1.4 Step-by-Step

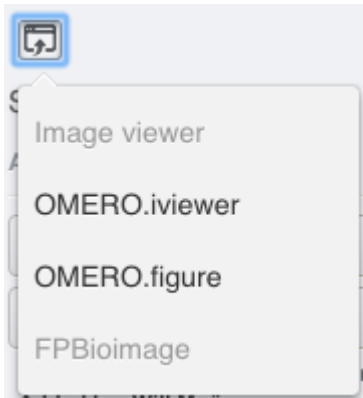
Using the sample images above, we will create a figure like this, but you can use any multi-channel images.




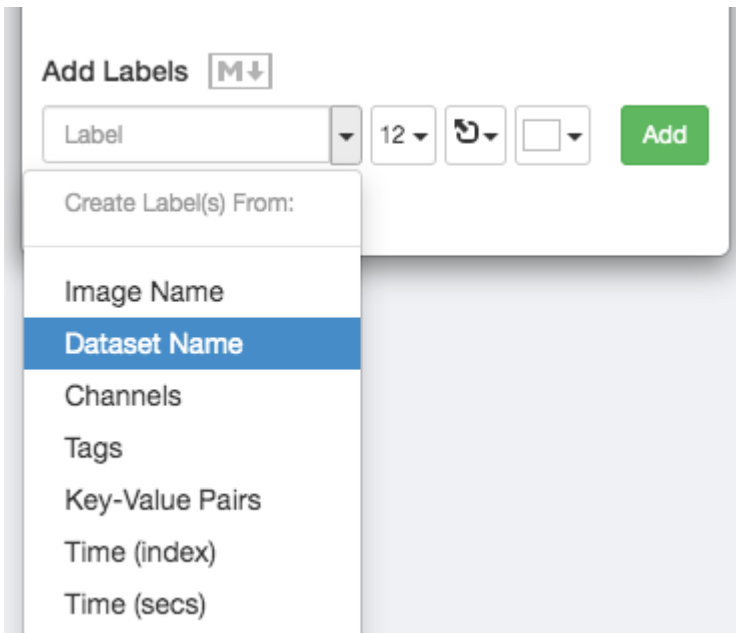
1. In the webclient, select 6 images from the **idr0021** Project.



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




- This will open these images in OMERO.figure in a new browser tab.
- 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.
- Select all Images and Zoom in around ~300%, using the Zoom slider in the right *Preview* tab.
- 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.
- Add labels: choose *Dataset Name* in the label input dropdown list, choose *color white* and *position=top-left*.



- Click *Add* to create the new label.
- 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 Python scripts on the OMERO.server or JavaScript in the browser Console to create or modify OMERO.figures. These are experimental features and not documented elsewhere.

### 2.2 Setup

- Install the OMERO.figure web app as described at <https://pypi.org/project/omero-figure/>
- Upload the script `Split_View_Figure.py` to the OMERO scripting service
- For the *Shapes heatmap from OMERO.table* example, you'll need to draw ROIs on an Image and create an OMERO.table on the Image with a Shape column as described in the example.


### 2.3 Resources

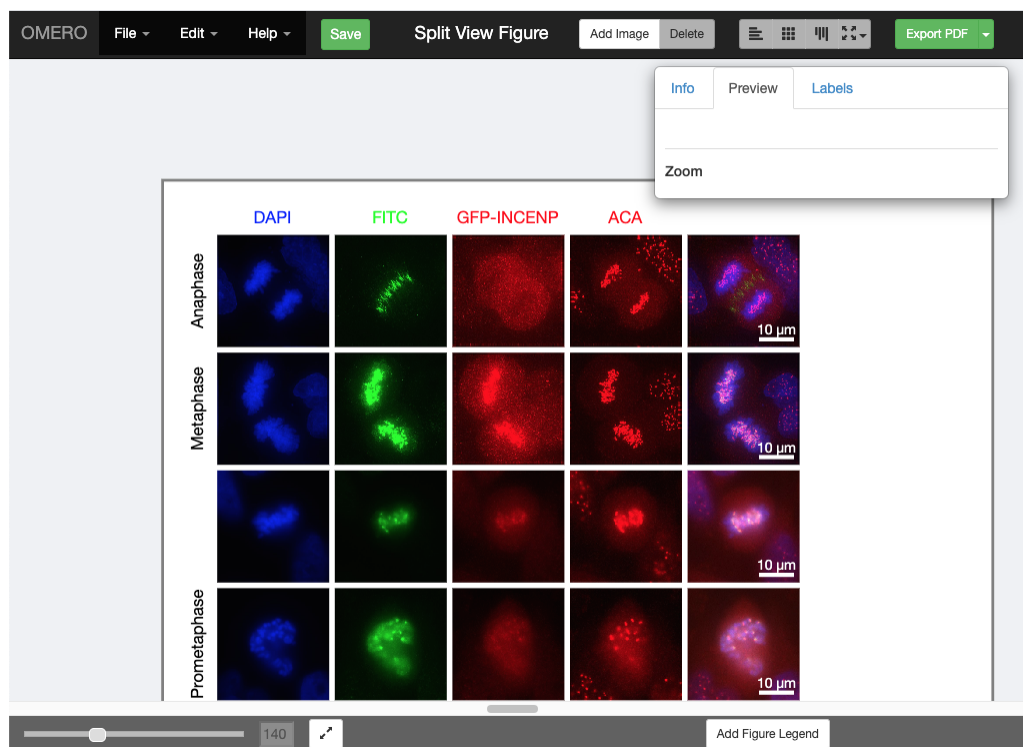
- For *Figure creation in Python*, any multi-channel images, e.g. from `siRNAi-HeLa`
- For the *Labels from Map Annotations* example, any time-lapse images, e.g. `FRAP`.
- For the other sections, any images can be used.

### 2.4 Figure creation in Python

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

The format of the JSON is described in the [Format](#) page. We will use the example `Split_View_Figure.py` script.

1. Select a few multi-channel 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. Choose a value for *Row Labels* input. The default is label each row with the *Image Name*, but you can also choose *Tags* as in the example below.
4. Run the script. When complete, open the OMERO.figure app and File > Open.
5. Choose the most recent figure, called “Split View Figure”.



## 2.5 Figure editing in JavaScript

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* of the browser *Developer Tools*. The easiest way to open the developer tools in most browsers is to open the context menu (right-click) anywhere on the page and choose *Inspect*.
4. We can use `figureModel.getSelected()` to get selected panels and for each panel we can call `p.set()` to change an attribute.

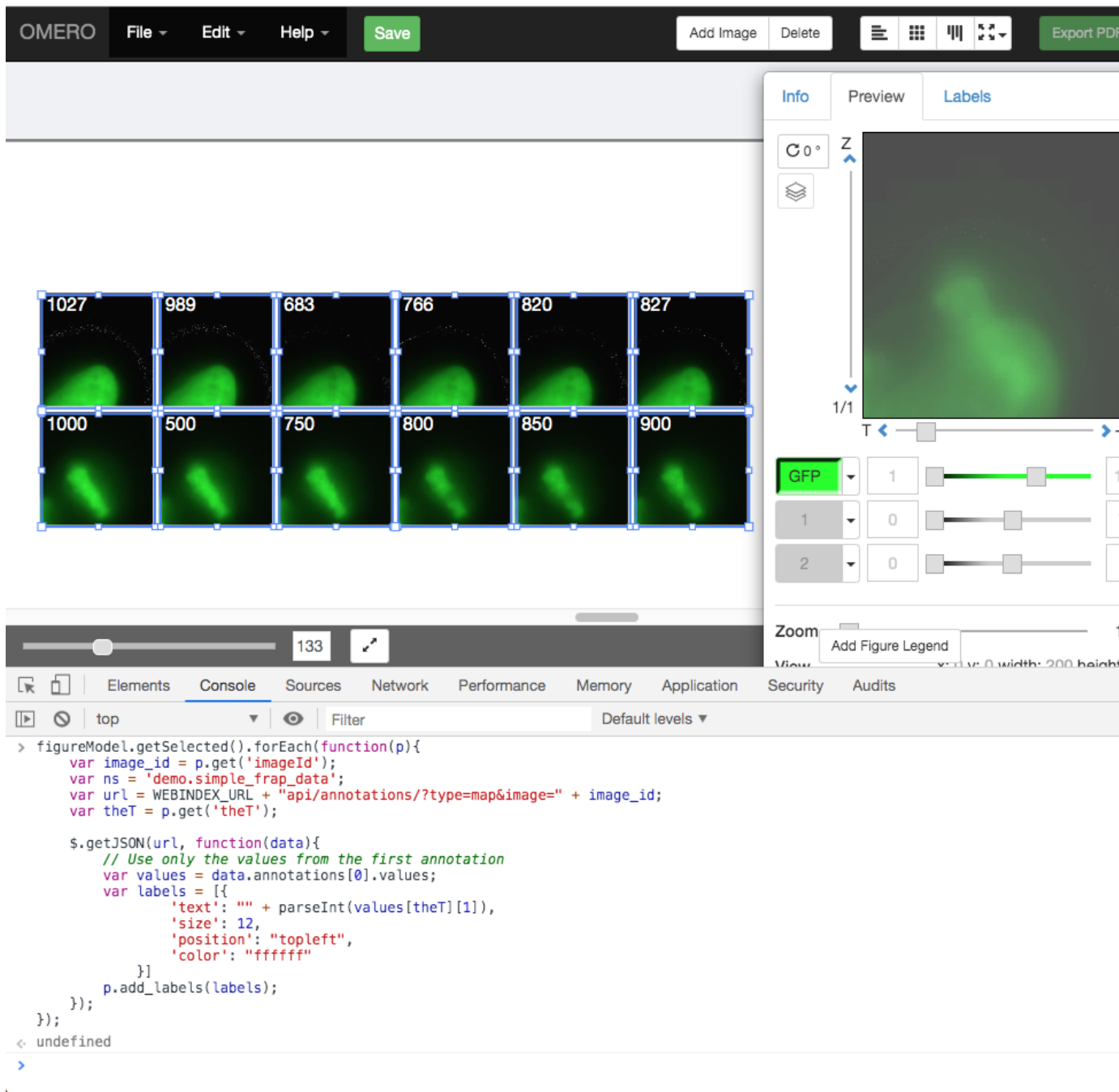
5. For example, to set the `height` of several panels to 200, we can select the panels in the figure UI and paste this code snippet in the *Console*:

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

6. There are several JavaScript examples in the [scripts](#) folder. Many of these are quite simple and self-explanatory. Below are some more complex examples that require specific set-up steps.

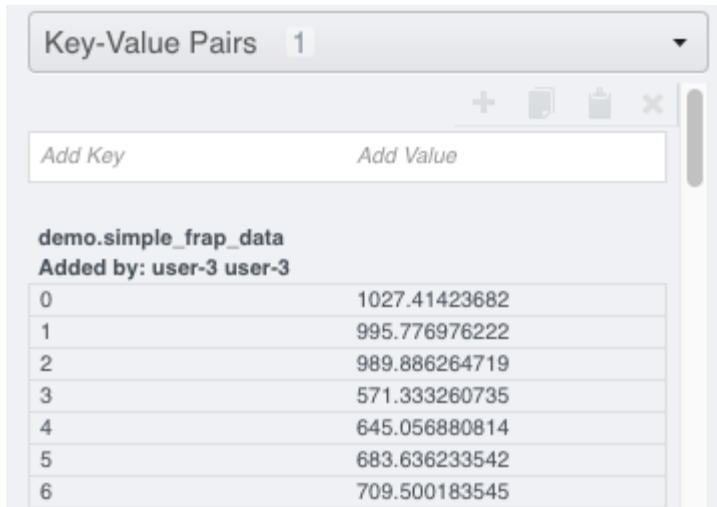
### 2.5.1 Example 1: Labels from Map Annotations

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



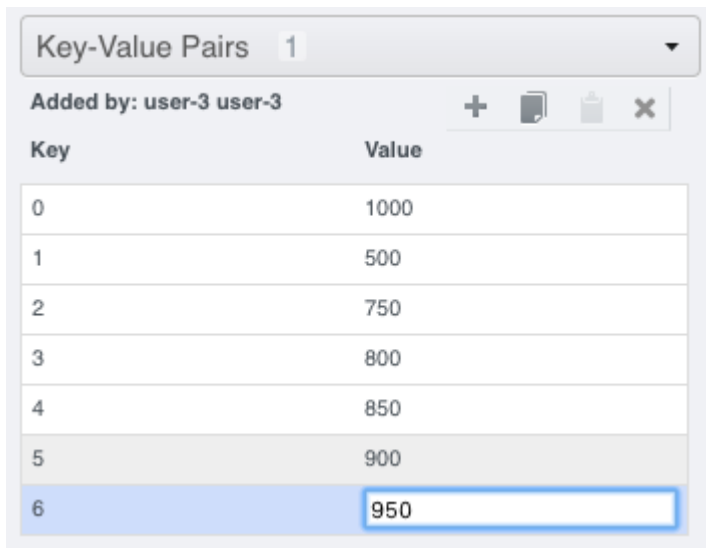
1. We can use AJAX to load JSON data and we will use `p.add_labels()` to create labels.
2. In this example we will load the FRAP intensities from the Map Annotations on these images.
3. Select 2 FRAP images that have previously been analysed to create a Map Annotation with the namespace `demo.simple_frap_data`.





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

- 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).



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

- Create a Figure with 2 images.
- Copy and paste each image several times and increment T-index in the Preview panel to show multiple time-points per image.
- Open the browser console by *right-click > Inspect Element (Firefox)* or *right-click > Inspect (Chrome)* and click on the *Console* tab.
- Copy the code from *figure\_frap\_mapannotation\_label.js* <[https://github.com/ome/omero-guide-figure/tree/master/scripts/figure\\_frap\\_mapannotation\\_label.js](https://github.com/ome/omero-guide-figure/tree/master/scripts/figure_frap_mapannotation_label.js)>.
- Drag to select the FRAP movie images in the figure.
- Paste the code into the console. **Do not hit enter yet.**
- 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.
- NB: If you manually created your own Map Annotation above, you can remove the line `url += '&ns=' + ns;` to avoid filtering by namespace.

13. 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.
14. Hit Enter to run the code on selected panels.
15. The labels should be added. Note that you can undo and redo these changes in the UI as normal.

## 2.5.2 Example 2: Shapes heatmap from OMERO.table

This example uses an OMERO.table linked to each Image to generate a heatmap of colors applied to Shapes on the figure panel.

### Setup:

1. If you wish to use Images and table data from *idr0079*, see the setup steps at [idr0079-data-prep](#).
2. Alternatively, perform the following steps:
3. For the Image you wish to use, add some ROIs to the Image. You can see the ROI and Shape IDs in the iviewer ROI table.
4. To setup the OMERO.table, create a CSV file with an `Roi` column and a `Shape` column containing the corresponding IDs and 1 or more number columns. The `#header` defines the column types: `l` (long) for *integers* and `d` (double) for *floats*. For example:

```
# header roi,l,d,d,l
Roi,Shape,Area,Sphericity,Pixels
1,10,34.5,0.5,110
2,11,18.2,0.6,55
2,12,44.1,0.9,210
```

5. Save the edited csv as `data.csv`.
6. With `omero-metadata` installed on the command-line, we can create an OMERO.table on the Image, using the Image ID:

```
$ omero metadata populate Image:123 --file data.csv
```

### OMERO.figure steps:

1. In OMERO.figure, add the Image to the figure, then in the ROIs dialog, load the Shapes from OMERO and add them to the panel. The JSON data for each Shape will have an `id` that corresponds to the Shape in OMERO.
2. View the JavaScript snippet at [figure\\_table\\_data\\_shapes.js](#). This uses the ID of each Shape of the panel to query the most recent OMERO.table on the Image using the endpoint: `/webgateway/table/Image/{imageId}/query/?query=Shape-{shapeId}`, which returns all table rows for that Shape ID. From the JSON returned, we find the column index for the data we want, e.g. `Sphericity`, and then get the value for that column. Once the values for all Shapes on the panel are loaded, the code calculates the range and generates a heatmap color for each value in that range. This is set as the color on each Shape.
3. Select the panel in the figure, then paste the JavaScript code into the browser *Console* and hit Enter
4. In the screenshot below, Shapes in the first panel are colored according to the `Centroids_RAW_X` column and Shapes on the lower panel are colored according to the `Sphericity` column. Images in this example are from *idr0079*.

The screenshot shows the OMERO.figure interface. The main workspace displays two panels of a cell image. The top panel is labeled 'Centroids RAW X' and the bottom panel is labeled 'Sphericity'. Both panels show a cell with segmented regions colored in green, yellow, and red. The right sidebar contains several controls: 'Info', 'Preview', and 'Labels' tabs; 'ROIs' section with '106 ROIs selected', a dropdown for '2 pt', and buttons for 'Copy', 'Paste', 'Delete', and 'Edit'; 'Scalebar' section with 'Pixel Size: 0.090 µm', a dropdown for '10', a unit selector 'µm', a 'Show' button, and a 'Label' checkbox; 'Add Labels' section with a dropdown for 'Label', a size dropdown for '12', a color selector, and an 'Add' button; and 'Edit Labels' section with a dropdown for 'Sphericity', a size dropdown for '21', a color selector, and a close button. The bottom console shows the following JavaScript code:

```
> async function shapeData(panel) {
  const shapeIds = panel.get("shapes").map(s => s.id).filter(id => id > 0);
  console.log('shapeIds', shapeIds);
  let vals_by_shape = {};
  for (let i = 0; i < shapeIds.length; i++) {
    // Load one at a time - more reliable
    let url = '/webgateway/table/Image/${panel.get('imageId')}/query/?query=Shape-${shapeIds[i]}';
    let r = await fetch(url).then(rsp => rsp.json());
    let colIndex = r.data?.columns?.indexOf("Sphericity");
    let shapeIndex = r.data?.columns?.indexOf("Shape");
    if (colIndex && shapeIndex && r.data?.rows.length > 0) {
      console.log("Value", r.data.rows[0][colIndex]);
      vals_by_shape[r.data.rows[0][shapeIndex]] = r.data.rows[0][colIndex];
    }
  };
  // Once all loaded, we can calculate range and assign colours to shapes
  const values = Object.values(vals_by_shape);
  let minVal = Math.min(...values);
  let valRange = Math.max(...values) - minVal;
  console.log('min, range', minVal, valRange);
  const new_shapes = panel.get("shapes").map(shape => {
    // hide any shapes we don't have data for
    if (!vals_by_shape[shape.id]) return { ...shape, strokeWidth: 0.01 };
    let value = (vals_by_shape[shape.id] - minVal) / valRange;
    let red = parseInt(value * 255).toString(16);
    let green = (255 - parseInt(value * 255)).toString(16);
    red = red.length == 1 ? '0' + red : red;
    green = green.length == 1 ? '0' + green : green;
    return { ...shape, strokeColor: `#${red}${red}${green}00`, strokeWidth: 5 };
  });
  panel.set('shapes', new_shapes);
}
figureModel.getSelected().forEach(shapeData);
```



---

### Move Figures between Groups

---

Moving Figures between Groups in OMERO might be necessary for example in cases where a scientist decides to move their data into another Group to enable cooperation and viewing by other colleagues or to make the data public.

As with all data in OMERO, Figures belong to a particular Group and are visible only to members of that Group. Figures can contain Images from the same Group as the Figure or from any other Group. Figures and the Images they contain can be independently moved from one Group to another but should ideally be kept in the same Group. This avoids the situation when a user can view a Figure, but not the Images within it.

This walkthrough offers two alternatives for getting Figures into another Group, either with or without duplicating the Figures and the Images contained in them.

### 3.1 Description

This guide covers:

- Moving of Figures and Images between Groups.
- Duplicating of Figures and Images into a new Group.

### 3.2 Setup

- Install the scripts `Figure_Images_To_Dataset.py` and `Dataset_Images_To_New_Figure.py` on your OMERO.server.
- We suggest in this workflow the installation under a folder `Figure_scripts` but you can install the script in any folder.
- See [server-side script guides](#) for further details.

## 3.3 Resources

- Any Images and Figures created from these Images.

## 3.4 Step-by-Step

### 3.4.1 Moving of Figures and Images between Groups


This workflow assumes that you have an OMERO.figure already created and want to move this Figure into another Group.

1. Log in to OMERO.web.
2. Click on `Figure` link above the central pane to get into OMERO.figure.
3. Inside OMERO.figure, click `File > Open` and open the Figure you wish to move to another Group. Copy the last part of the url from the address bar of your browser, which contains the Figure ID, e.g.:  
<https://your-omero-server.org/web/figure/file/79828>  
has the Figure ID 79828. The Figure ID will be needed later in case you also want to collect and move the Images contained in the Figure as described below.
4. Click `File > Move Figure to Group...` In the following dialog, select the Group you wish to move the Figure to and click OK.
5. Observe a dialog reporting a success of the move. Verify that when you now again click `File > Move Figure to Group...` the Figure is now reported to be in your intended Group.

---

**Note:** You might want to move the Images contained in the Figure to the other Group as well. This step is optional, if you want to use your Figures by yourself only, but you should consider it in case the users you want to share your Figure with do not have permissions to see your Images in the original Group (for example they are members of the target Group only).

---

1. In the following steps, we first collect all the Images in the Figure into a single Dataset to enable moving them to the target Group in a single step.
2. Go back to the tab with OMERO.web, create a new Dataset and select it.
3. Click on the Scripts icon  above the central pane of OMERO.web. Select `Figure scripts > Figure Images to Dataset`. Start the script and in the dialog, enter the Figure ID into the Figure IDs field. If you wish to work with multiple Figures, IDs can be separated with commas, e.g. 79828, 79830, 71228. Click Run.
4. When the script finishes, refresh the page and find the Images contained in your Figure linked to the Dataset. Note that the Images are linked to the chosen Dataset without removing them from any existing Dataset. They will be doubly-linked but not duplicated.
5. The above means that when you execute the Move into another Group, the Images in your Dataset will no longer be available in the original Group.
6. Move the Images into the Group you have moved the Figure to. Be sure to select the Images in OMERO.web when moving, not the Dataset. If the Dataset is selected, the Images are left in the original Group, and only the empty Dataset is moved. This is because the Images are linked to the original Dataset which is left in the

original Group. Follow [the Move workflow](#) to execute the Move. The Move will enable any member of the target Group to view both the Figure and the Images within it.

### 3.4.2 Duplicating of Figures and Images into a new Group

If you wish to keep your Figures and the contained Images in their original Group, while also making them available to users in another Group, you can duplicate them as described below.

In this workflow, we first duplicate the Images within one or more Figures. Then move these Images to the target Group.


Finally, using a script, we copy the Figure into the target Group, replacing the Images within it with the duplicated Images.

Note that this workflow involves usage of Command Line.

1. Collect the Images contained in the Figures into a single Dataset using the workflow [Moving of Figures...](#) For this, you can find out the Figure IDs either manually as described in the [Moving of Figures...](#) or by running hql queries on the command line, such as:

```
$ omero hql --all --limit 1000 --style plain --ids-only "select f.id from
↪FileAnnotation f where (f.details.group.name = 'Lab1' and f.details.owner.id =
↪454)" | sed -e 's/^.*,//g' | paste -s -d, -
```

which will retrieve all the Figure IDs of user with ID 454 in a Group Lab1 in a format which you can immediately copy and paste into the Figure Images to Dataset script.

2. Start your command line terminal and duplicate the Dataset with the Images contained in the Figures as described in [the Duplicate workflow](#).
3. Go to OMERO.web, select the duplicate Dataset and Move it to the target Group. For that, follow [the Move workflow](#).
4. Find the Dataset which you have just moved and select it.
5. Click on the Scripts icon  above the central pane of OMERO.web. Select Figure scripts > Dataset Images To New Figure.
6. Start the script and in the dialog, enter the Figure ID into the Figure IDs field. If you wish to work with multiple Figures, IDs can be separated with commas, e.g. 79828, 79830, 71228. Click Run. This will copy each specified Figure, update the Images within it to those in the duplicate Dataset (using the Image name to match the replacement Images) and save the Figure to the new Group.
7. Click on Figure link above the central pane to get into OMERO.figure.
8. Inside OMERO.figure, click File > Open. In the top-right corner of the new dialog, click on the Group dropdown and select your target Group name. Verify that the list contains the newly created Figures.





## CHAPTER 4

---

### Contribute

---

Changes to the documentation or the materials should be made directly in the [omero-guide-figure repository](#).