OMERO guide Fiji Documentation

Release 0.2.1

Open Microscopy Environment

May 07, 2021

Contents

Fiji is a popular free open-source image processing package based on ImageJ, https://imagej.net/Fiji. We demonstrate how to use Fiji User Interface and OMERO and we also demonstrate how to analyze data in OMERO using Fiji scripting facility.

Contents:

How to install the OMERO plugin for Fiji/ImageJ

1.1 Description

Fiji is a free open-source image processing package based on ImageJ. The following workflow shows how to install the OMERO plugin for Fiji and ImageJ.

The OMERO plugin does not have an update site yet.

1.2 Setup step-by-step

We assume that you have already Fiji/ImageJ installed locally.

In this section, we will cover the steps required to install the OMERO plugin for Fiji. If you wish to install it for ImageJ, an additional step is needed.

We first describe the common installation steps for ImageJ and Fiji. We then describe how to install the *Bio-Formats Package* for ImageJ.

Installing the OMERO plugin in Fiji also adds the dependencies required to connect to OMERO using the Script Editor of Fiji.

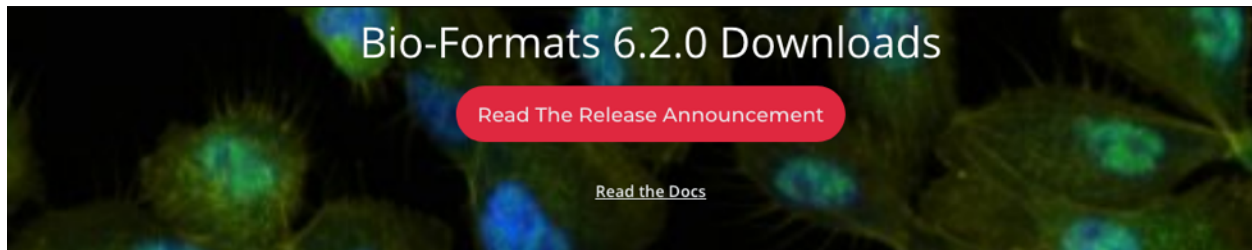Installation of the OMERO plugin for Fiji and ImageJ, the common steps:

- Find the Plugins folder of your Fiji application and check if it contains any old omero_ij-5.x.x-all.jar file(s) or OMERO.imagej-5.x.x folder(s). Remove any such jar files or folders from the Plugins folder.

- Download from https://www.openmicroscopy.org/omero/downloads the latest 5.x.x version of ImageJ/Fiji plugin for OMERO
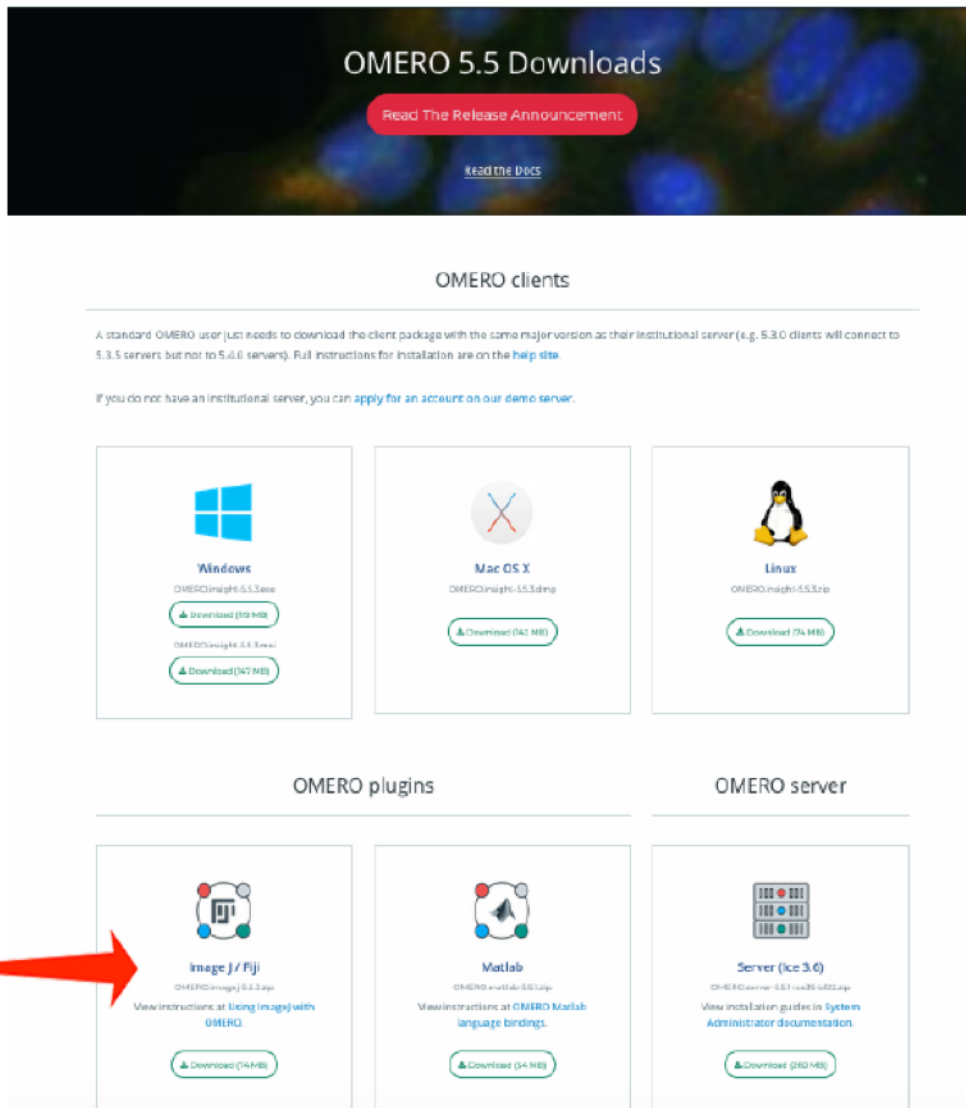
- For recent plugin versions (5.5.7 and higher): Note where you downloaded the omero_ij-5.x.x-all.jar file. Copy that file into the *plugins* folder of Fiji.

- For plugin versions lower than 5.5.7: Extract the downloaded .zip archive. Remember where you extracted it to.

- For plugin versions lower than 5.5.7: Copy the extracted folder and paste it to the *plugins* folder of Fiji.

- **Note:** For plugin versions lower than 5.5.7: Some Windows unzip apps create a double folder enclosing the plugin. If that is the case, copy the inner OMERO.imagej-5.x.x folder into *Fiji.app > plugins* folder.

- Now, restart Fiji. If you are using ImageJ, follow with the additional step below.

Installation of Bio-Formats Package, ImageJ only:

- Download the latest version of the *Bio-Formats Package* from: https://www.openmicroscopy.org/bio-formats/downloads

- Move the downloaded file into the *ImageJ > plugins* folder.

- Restart ImageJ.

Analyze manually OMERO data using the OMERO plugin for Fiji/ImageJ

2.1 Description

The following workflows should work both with ImageJ and Fiji, after these have been correctly set up with the OMERO plugin for Fiji/ImageJ.

Using the User Interface of the OMERO plugin, we will show:

- How to connect to OMERO using the OMERO plugin for ImageJ/Fiji.

- How to open an image from OMERO.server into Fiji/ImageJ.

- How to manually save ROIs and measurements as CSV back to the original image in OMERO.server.

- How to import a newly created image from Fiji/ImageJ into OMERO.

2.2 Setup

- Install Fiji on the local machine with the OMERO.insight-ij plugin. The installation instructions can be found at *How to install the OMERO plugin for Fiji/ImageJ*.

2.3 Resources

- Samples images from the Image Data Resource (IDR) idr0021.

2.4 Step-by-step

2.4.1 Example 1: ROIs and measurement creation

In this first example we show how to open an OMERO image in Fiji, draw ROIs, measure those ROIs and show how to save the ROIs and the measurement back to OMERO.

1. Launch Fiji/ImageJ.

2. 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 OMERO.web.

3. In the OMERO login dialog, click the wrench icon and then add the server address in the dialog. 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.

4. Click Apply.
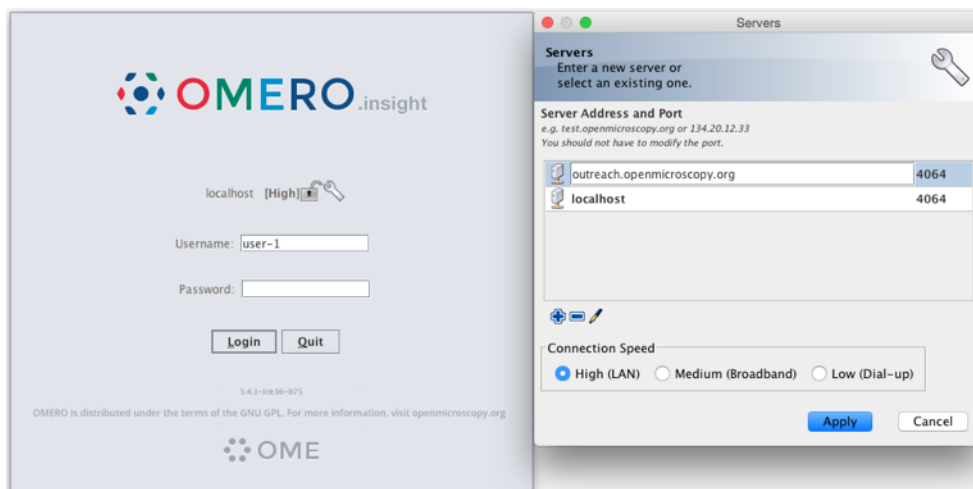
5. Enter your credentials and click *Login*.

6. Select the **A-Fiji-dataset** Dataset.

7. Double-click on a thumbnail or on an Image in the left-hand tree to open an Image in ImageJ.

8. Go to *Analyze > Tools > ROI Manager. . .*

9. Draw a shape using for example the Freehand selection tool.

10. In the ROI manager, click the button *Add [t]* to add the shape to the ROI Manager.
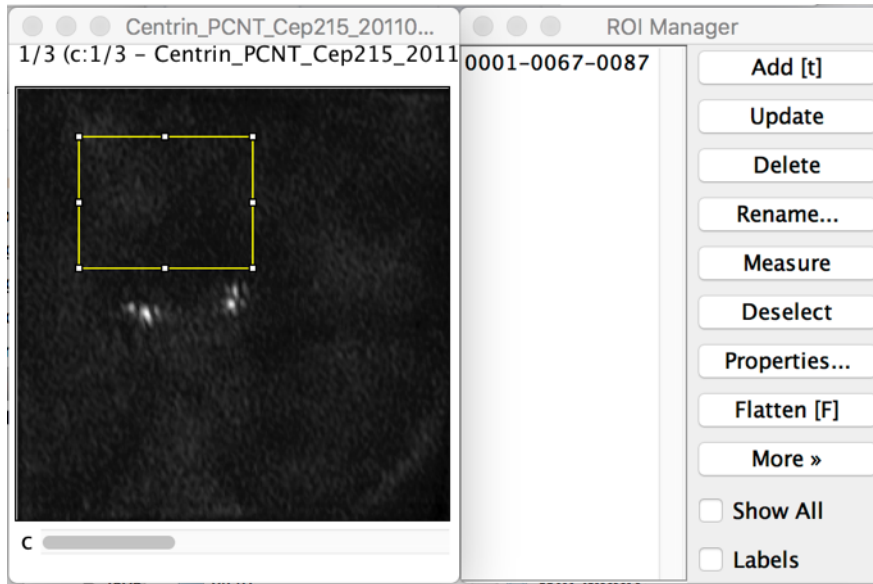
11. Move to another channel, using the *c slider*.

12. Draw other shapes if desired. Click *Add [t]* to add them to the ROI Manager.

13. When done with the drawing, click the button *Measure* in the ROI Manager.

14. A dialog with measurements for each shape pops up.

15. To save the ROI and the measurement back to OMERO, go to *Plugins > OMERO > Save ROIs To OMERO*.

16. In the dialog popping up, under the *Save* section select *ROI* and *Measurements*.

17. The measurements are saved back to OMERO as a CSV file and linked to the Image.
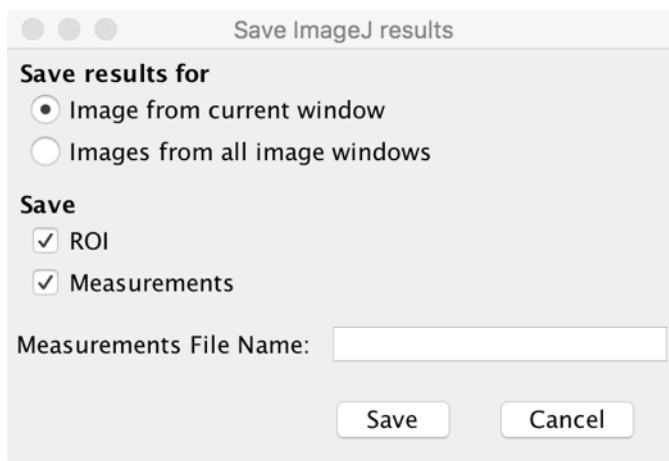
18. Go to OMERO.web and log in.

19. Select the image opened in Fiji/ImageJ.

20. Check that there is a new CSV file under the *Attachments* harmonica.

21. Open the image in OMERO.iviewer to the see the ROIs and make sure that you can interact with them.

2.4.2 Example 2: Cropping and import

In this example, we show how to open an OMERO image, crop the image and import the cropped image back to OMERO as OME-TIFF.

In this example we assume that the user is already logged in. If this is not the case, go back to the first step of Example 1.

1. Select a dataset, for example the *A-Fiji-dataset* Dataset.

2. Double-click on a thumbnail or on an Image in the left-hand tree to open an Image in Fiji/ImageJ.

3. Draw a Rectangle on the Image.

4. Select the option *Image > Crop*.

5. A new Image will be displayed in a Fiji/ImageJ window.

6. Go to *Plugins > OMERO > Save Image(s) To OMERO*.

7. An Import dialog will pop up.
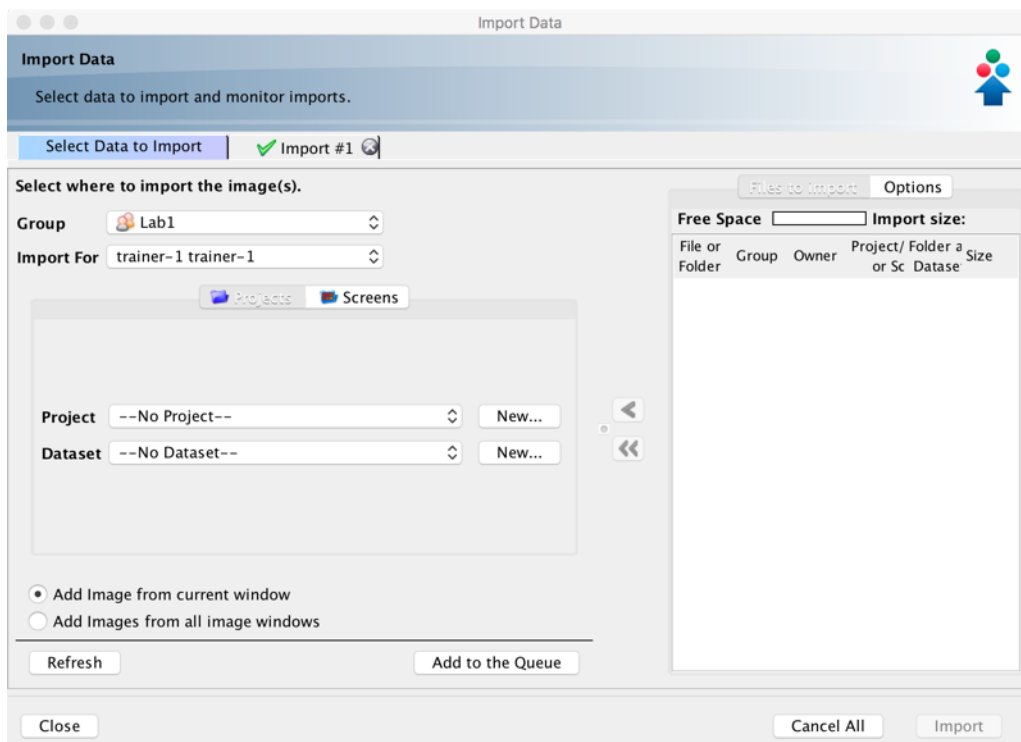
8. Check that the option *Add Image from current window* is selected.

9. Select where to import the cropped Image, for example an existing Dataset, e.g. *A-Fiji-dataset*. You can also select *New From Folder* option which will create a new Dataset named with the name of the image you opened from OMERO to Fiji. In case you select *No Dataset* option, the new image will be displayed in the *Orphaned Images* folder in OMERO.

10. Click *Add to the Queue* button.

11. Then click *Import*. The import will start.

12. When the import is done, go back to the Tree view in the Fiji plugin or OMERO.web. Refresh. Check the new Image.

CHAPTER 3

Segment manually OMERO data using Fiji/ImageJ

3.1 Description

The following workflows should work both with ImageJ and Fiji, after these have been correctly set up with the OMERO plugin for Fiji/ImageJ.

Using the User Interface of the OMERO plugin, we will show:

- How to connect to OMERO using the OMERO plugin for Fiji/ImageJ.

- How to open an image from OMERO.server into Fiji/ImageJ.

- How to manually segment an image opened from OMERO in Fiji/ImageJ using the plugin *Thresholding and Analyze particles* in Fiji/ImageJ.

- How to record this workflow in Fiji/ImageJ, using the *Recorder* plugin, for later use.

3.2 Setup

- Fiji has been installed on the local machine with the OMERO.insight-ij plugin. The installation instructions can be found at *How to install the OMERO plugin for Fiji/ImageJ*.

3.3 Resources

- Samples images from the Image Data Resource (IDR) idr0021.

3.4 Step-by-step

1. Launch Fiji/ImageJ.

2. 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 OMERO.web.

3. In the OMERO login dialog, click the wrench icon and then add the server address in the dialog. 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.

4. Click Apply.
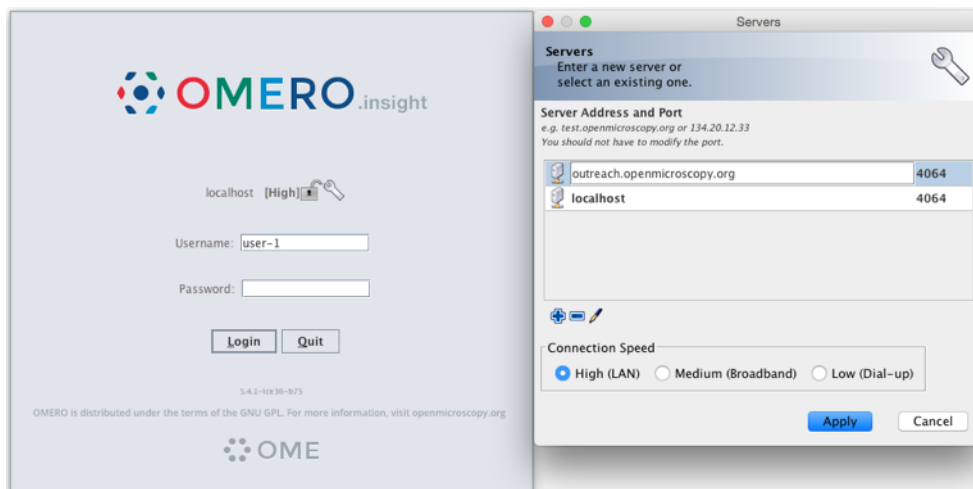
5. Enter your credentials and click *Login*.

6. Browse to the Project **idr0021**, open any Dataset and double-click once on an Image to open it in Fiji. Bio-Formats is used to view the Image.

   • Make sure to select *View stack with: Hyperstack* in the *Bio-Formats Import Options* dialog.

   • Note that each plane will be transferred from the server to the client machine so this may take a few moments.

7. To open the Recorder, go *Plugins > Macros > Record. . .* , select Macros to record the actions. The steps will then be used in the Scripting workflow.

8. Convert floating-point pixel-type to 8-bit using *Image > Type > 8-bit*.

9. Go to *Image > Adjust > Auto Threshold*, to open the *Auto Threshold* dialog:

   • Select *MaxEntropy* for the *Method* parameter.

   • Check the checkbox *Stack*.
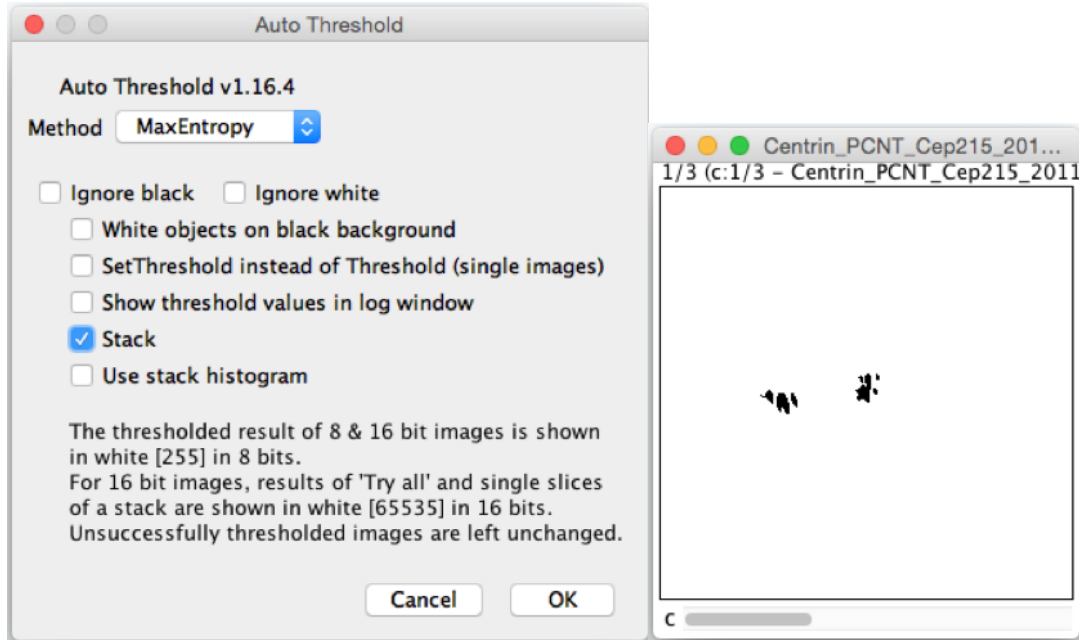
10. Click *OK*.

11. Then open *Analyze > Analyze Particles. . . .*

12. In the dialog

    • Set *Size* to *10-Infinity* and check *Pixel units*.

    • Check the following checkboxes:

        i. *Display results*

        ii. *Clear results*

        iii. *Add to Manager*

    • **Click *OK* then *Yes* in the popup dialog indicating asking to** *Process all X images?*.
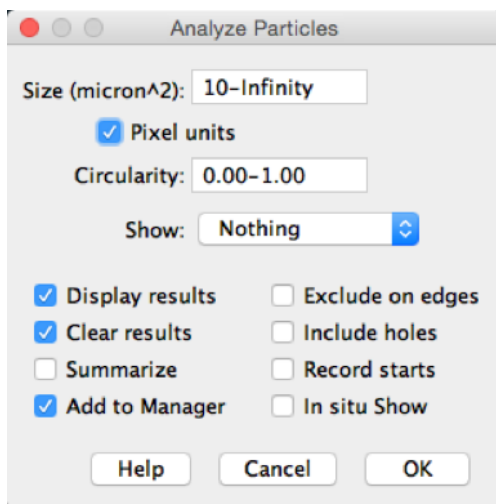
13. To save the thresholded Image back to OMERO with the generated ROIs and the measurements:

    • Select *Plugins > OMERO > Save Image(s) to OMERO*.

- Create a New Dataset for the image

    i. Click the *New. . .* button next to the selection box on the Dataset row.

    ii. In the dialog that pops up, enter a name and a description (optional).

    iii. Click Create.

- The newly created Dataset will automatically be selected.

- Click *Add to the Queue* then *Import*.

- Go to OMERO.web and check that the measurements have been saved in a CSV file and attached to the Images. The attachment can then be downloaded at any time.

14. **To save the recorded macro to OMERO.server:**

    - In Fiji, find the Recorder window and click "Create". A new window will pop up with the macro you just recorded. Verify the macro on a new image. Once the verification succeeds, just close the macro window and in the dialog which pops up select "Yes" to save the macro. Save the macro locally e.g. "your-macro-name.ijm".

    - In OMERO.web, select the dataset you would like the macro to be run on and attach the "your-macro-name.ijm" macro you just recorded as file attachment to that dataset. The attachment can then be downloaded at any time. Also, the attached macro can be used by Fiji scripts - see example on threshold_scripting.html.

Segment OMERO data using the Scripting editor in Fiji

4.1 Description

The following workflows should work in Fiji, after these have been correctly set up with the OMERO plugin for Fiji/ImageJ.

Using the Scripting editor of Fiji, we will show:

- How to connect to OMERO using the JAVA API.

- How to load all the images within a given Dataset.

- How to load the images in ImageJ using Bio-Formats.

- How to analyze the images using *Thresholding and Analyze Particles* plugin in ImageJ within a script.

- How to save the segmented ROIs as polygons in OMERO.

- How to collect the measurement associated to the ROIs and

    - Save them as an OMERO.table and link the table to the Dataset.

    - Save them as a CSV file and link it to the Dataset.

4.2 Setup

- Install Fiji on the local machine with the OMERO.insight-ij plugin. The installation instructions can be found at *How to install the OMERO plugin for Fiji/ImageJ*.

4.3 Resources

- Data: Samples images from the Image Data Resource (IDR) idr0021.

- Java API documentation.

- Script: Groovy script for automatic segmentation of images from OMERO using Fiji - `analyse_dataset_save_rois_and_summary_table.groovy`.

4.4 Step-by-Step

We will now repeat the manual analysis *Segment manually OMERO data using Fiji/ImageJ* on a Dataset using the scripting facility available in Fiji.

Let's go over the script to understand the logic and see how it matches the UI steps.

This script explores the JAVA API using Groovy.

It will process all the Images in the specified Dataset, applying threshold, analyzing particles and saving ROIs back in OMERO i.e. we reproduce in a script the manual steps recorded. Further, it will create a CSV and OMERO.table to be attached to that Dataset in OMERO.

1. In your browser, go to the server address provided.

2. Log in using the credentials provided.

3. Make sure you are selecting your own data. Select the Dataset **A-Fiji-dataset**.

4. Launch Fiji.

5. Go to *File > New > Script. . . .*

6. A dialog pops up. In the *Language* menu, select *Groovy*.

7. Copy, into the text script editor of Fiji, `analyse_dataset_save_rois_and_summary_table.groovy`.

8. You will be asked to enter your login credentials when you run the script.

9. Click *Run*.

10. Return to OMERO.web and open an Image from this Dataset in OMERO.iviewer.

11. Click the *ROIs* tab to see the added ROIs. Note that the ROIs have been assigned a Channel index to indicate which Channel they were derived from.

12. In the *Settings* tab, turning channels on/off will also show/hide ROIs assigned to those channels.

13. Open the image in OMERO.figure for a quick publication by going to Info tab in iviewer and clicking on

    OMERO.figure in the Open with line.

14. The script `idr0021.groovy` was used previously to run on the whole **idr0021** Project and produce the resulting CSV file. We can now download this file from OMERO and open it in Microsoft Excel.

15. In Excel, select the Dataset column and the column bounding_box and then click on *Insert > X Y Scatter > Scatter* . Depending on the version of Excel you are using, this option might not be available.
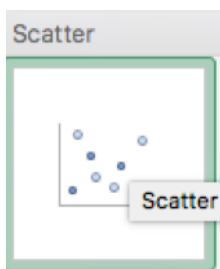
17. This will create a scatter plot. Right-click into the Chart now and select *Change chart type > Statistical > Box and Whisker*.

Segment OMERO data using Fiji run headlessly in notebook

5.1 Description

The following examples follow up on the script shown in *Segment OMERO data using the Scripting editor in Fiji*. But, unlike in *Segment OMERO data using the Scripting editor in Fiji*, here, Fiji is included into a Docker image and run in headless mode. The scripts are embedded into a Jupyter Notebook. The Groovy language is used, which is very similar to Java and can be used in Fiji scripting. The scripts show:

- How to run a Fiji macro attached to an OMERO Dataset, save ROIs, Results and Images (as OME-TIFF) to OMERO
- How to crop an image in OMERO using Fiji cropping functionality.

5.2 Setup

Fiji has been installed in a Docker image using repo2docker.

5.3 Resources

- Data: Samples images from the Image Data Resource (IDR) idr0021.
- Macro: `fiji-macro-segment.ijm`

5.4 Step-by-Step

1. Create a Fiji macro, for example by recording it and save it locally. You can for example use adjusted macro created in *Segment manually OMERO data using Fiji/ImageJ*. See `fiji-macro-segment.ijm`.

Note: Not all commands which can be run and recorded in the Fiji user interface can be also run in the Fiji in headless mode.

2. In your browser, go to the server address provided.

3. Log in using the credentials provided.

4. Make sure you are selecting your own data. Select the Dataset **A-Fiji-dataset**.

5. Open the *Attachments* harmonica in the right-hand panel and click on the plus icon. Browse and attach the `fiji-macro-segment.ijm` to this Dataset as File Annotation.

6. Run the notebook run_attached_macro.ipynb.

7. Select the first Step and click on the Run button to execute each step in turn.

8. For the connection to OMERO, you will be asked to enter your login details when running the first cell under *Description*. Note that the connection itself happens only in the next cell, highlighted as *Connect to OMERO*.

9. In the following cell, you will be asked to enter the ID of the Dataset onto which you attached the Macro file above.

10. The code in the next cell is fetching the attachments on the Dataset and offering a dropdown menu for you to select a suitable attachment i.e. your attached macro.

11. The next cell is asking for input about the types of data which should be saved back to OMERO (ROIs, Results or Images). This is of course dependent on the type of results your Macro is producing in Fiji. For this example, we select ROIs and Results.

12. The script in the next cell will process all the Images in the specified Dataset, applying threshold, analyzing particles, i.e. steps which are captured inside your macro file. Further, it will save ROIs back in OMERO and create a CSV to be attached to each image in that Dataset in OMERO.

13. Return to OMERO.web and open an Image from this Dataset in OMERO.iviewer.

14. Click the *ROIs* tab to see the added ROIs.

15. In the *Settings* tab, turning channels on/off will also show/hide ROIs assigned to those channels.

16. Open the image in OMERO.figure for a quick publication by going to the *Info* tab in OMERO.iviewer and

    Open With: OMERO.figure |

    clicking on OMERO.figure in the Open with line.

17. Click the *ROIs* tab to see the added ROIs. Note that the ROIs have been assigned a Channel index to indicate which Channel they were derived from.

18. In the *Settings* tab, turning channels on/off will also show/hide ROIs assigned to those channels.

19. Run the notebook crop_image.ipynb.

20. Run through the notebook, logging in as previously and indicating the Image ID of the image you want to crop.

21. In OMERO.web, refresh and note that the cropped image has been imported into a newly created Dataset *Cropped image*.

Use ImageJ in Python

6.1 Description:

This section shows how to use ImageJ as a Python library to analyze data in OMERO.

Using the Python API allows us to easily load the 2D-plane we need to see or analyze. This is much easier than using the Java API and Bio-Formats plugin.

We will show in the examples:

- How to start ImageJ in Python.

- How to load data from OMERO.

- How to run ImageJ macro from Python.

6.2 Setup

For using the Python examples and notebooks of this guide we recommend using Conda (Option 1). Conda manages programming environments in a manner similar to virtualenv.

Alternatively you can use `repo2docker` to build and run a Docker image locally (Option 2). This Docker image will provide the Conda environment and Jupyter notebooks with some image analysis workflows.

6.2.1 *Option 1*

Install omero-py and pyimagej via Conda:

- Install Miniconda if necessary.

- Create a programming environment using Conda and activate it:

```
$ conda create -n imagej_python python=3.6

$ conda activate imagej_python
```

- Install omero-py and pyimagej:

```
$ conda install -c conda-forge pyimagej

$ conda install -c ome omero-py
```

Note: we have noticed problems using the Option 1 approach when running the scripts on some operating systems due to issues with the ImageJ1-ImageJ2 bridge.
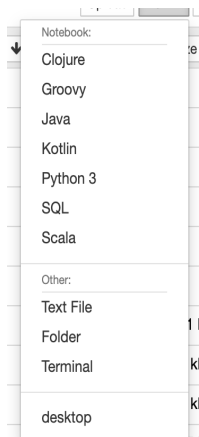
6.2.2 Option 2

Create a local Docker Image using `repo2docker`, see `README.md`:

```
$ pip install jupyter-repo2docker
$ git clone https://github.com/ome/omero-guide-fiji.git
$ cd omero-guide-fiji
$ repo2docker .
```

When the Image is ready:

- Copy the URL displayed in the terminal in your favorite browser

- Click the `New` button on the right-hand side of the window

- Select `Terminal`

- A Terminal will open in a new Tab

- A Conda environment has already been created when the Docker Image was built

- To list all the Conda environments, run:

```
$ conda env list
```

- The environment with the OMERO Python bindings and a few other libraries is named `notebook`, activate it:

```
$ conda activate notebook
```

6.3 Step-by-Step

The script used in this document is `run_macro_python.py`. One of the advantages of this approach is that we can load only the 2D-planes we wish to analyze.

The script used in this document contains an ImageJ1 macro that needs graphical user interface (GUI) elements, and thus it requires using ImageJ in GUI mode. In this GUI mode, the resulting windows content is handled. If you are running the example in the Docker container, you will also need to start UI environment if it is not already up. If you have used the Option 2 above, select `desktop` to start the UI environment.
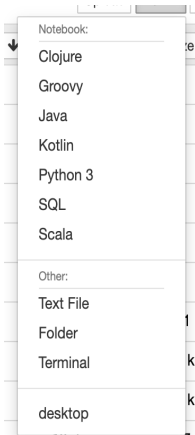
If you are using the Option 1 above, you will first need to update the script to point to your local installation of Fiji or use one of the options described in ImageJ Tutorials. You can now run the script. To run the script, go to the folder `scripts/python` and run:

```
$ python run_macro_python.py
```

If you do not use any ImageJ1 features e.g. macro, you do **not** need the UI environment.

Below we explain the various methods in the scripts: how to start Fiji, how to load the planes to analyze and how to run an ImageJ1 macro.

In this example, Fiji has been installed locally.

6.3.1 Script's description

Import modules needed:

```
import imagej
from omero.gateway import BlitzGateway
```

Load Fiji. If you run the script locally, point to your local installation of Fiji or load Fiji "on the fly". Note that the parameter *headless* has been set to *False* since we need the graphical user interface to run the ImageJ1 macro:

```
def start_fiji():
    ij = imagej.init('/srv/conda/vnc/Fiji.app', headless=False)
    ij.getVersion()
    return ij
```

Connect to the server. It is also important to close the connection again to clear up potential resources held on the server. This is done in the `disconnect` method:

```python
def connect(hostname, username, password):
    """
    Connect to an OMERO server
    :param hostname: Host name
    :param username: User
    :param password: Password
    :return: Connected BlitzGateway
    """
    conn = BlitzGateway(username, password,
                        host=hostname, secure=True)
    conn.connect()
    conn.c.enableKeepAlive(60)
    return conn


def disconnect(conn):
    """
    Disconnect from an OMERO server
    :param conn: The BlitzGateway
    """
    conn.close()
```

Load an image from IDR:

```python
image = conn.getObject("Image", image_id)
```

Load the binary plane as numpy array:

```python
def load_plane(image):
    """
    Load a 2D-plane as a numpy array
    :param image: The image
    """
    pixels = image.getPrimaryPixels()
    return pixels.getPlane(0, 0, 0)
```

To be used in ImageJ, the numpy array will be converted into ImageJ types using the *to_java()* method. In order the use the methods implemented above in a proper standalone script: **Wrap it all up** in an `analyse` method and call it from `main`:

```python
def analyse(ij, conn, image_id):
    # Step 3 - Load image
    image = conn.getObject("Image", image_id)
    # -
    plane = load_plane(image)
    from jnius import autoclass
    autoclass('ij.WindowManager')
    ij.ui().show('Image', ij.py.to_java(plane))
    macro = """run("8-bit")"""
    ij.py.run_macro(macro)
```

```python
def main():
    try:
        hostname = input("Host [wss://idr.openmicroscopy.org/omero-ws]: \
                          ") or "wss://idr.openmicroscopy.org/omero-ws"
        username = "public"
        password = "public"
        image_id = int(input("Image ID [1884807]: ") or 1884807)
        print("initializing fiji...")
        ij = start_fiji()
        print("connecting to IDR...")
        conn = connect(hostname, username, password)
        print("running the macro...")
        analyse(ij, conn, image_id)
    finally:
        if conn:
            disconnect(conn)


if __name__ == "__main__":
    main()
```

CHAPTER 7

Contribute

Changes to the documentation or the materials should be made directly in the omero-guide-fiji repository.