

isMap-napari manual

Version 0.1.0 — February 2026

1 General information

1.1 License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

1.2 Requirements

The isMap napari plugin requires:

- Python ≥ 3.10
- Napari + Qt (via conda)
- Cellpose + PyTorch
- Scientific Python stack (numpy, pandas, scikit-image, etc.)

All dependencies are resolved automatically when installing via conda + pip.

1.3 Installation

For installation, run the commands in an integrated terminal in a code editor or in a separate terminal window. The following installation guide is shown for Visual Studio Code (VS Code). The installation shown here requires Git. However, it is also possible to download the repository as a zip file directly into your preferred folder. In that case, open the folder where the .zip file was extracted in step 2 and skip step 4.

1. Open VS Code.
2. Open the folder where the repository will be cloned: File → Open Folder...
3. Open a new terminal: Terminal → New terminal
4. Clone the repository (from the terminal window).

```
git clone https://github.com/aklab-Tcell-signal-integration/napari-isMap.git
```

5. Move into the isMap-napari directory.

```
cd isMap-napari
```

6. Create and activate a virtual environment.

Conda is strongly recommended.

```
conda create -n venv-ismap -c conda-forge python=3.11
```

i The plugin is tested on Python 3.11.

7. Activate the environment.

```
conda activate venv-ismap
```

8. Install the plugin. This can take a while.

```
pip install ".[gui]"
```

9. Validate the installation.

```
npe2 validate ismap-napari
```

You should see:

✓ Manifest for 'immunological synapse Map analysis program' valid!

2 isMap - napari

The napari based isMap module bundles image processing, feature extraction, radial averaging, colocalization, and data visualization into one easy-to-use graphical user interface. Users can assess segmentation quality directly within the viewer, adjust filtering parameters accordingly and batch-process large datasets without any coding. All measurements and visualizations can be exported.

2.1 Data organization

The isMap napari plugin requires the data to be organized as shown below for automatic parsing:

- Images from the same condition should be stored in the same directory.
 - The images should be stored as individual files with a file format supported by BioFormats.
 - A list of compatible image formats can be found [here](#).

- Condition folders should be stored in an input directory that the user will be prompted to name when running the plugin.

```
Input_Directory/  
  Condition_A/  
    Image_1.vsi  
    Image_2.vsi  
    ...  
  Condition_B/  
    ...
```

2.2 Using the napari isMap plugin

1. Start napari from the terminal.

```
napari
```

2. Open the plugin: Plugins → isMap (immunological synapse map analysis program)

3. In the docked widget (Figure 1):

- Input Folder → folder with compatible image files
- Output Folder → where results are saved
- Choose the segmentation parameters:
 - Model → the Cellpose model used for segmentation
 - Diameter → estimated diameter of the synapses used for Cellpose segmentation
 - Diameter units → either in pixels or microns (pixels per micron is then extracted from image metadata)
 - Scale (x) → image rescaling factor used in Cellpose
- Run Segmentation → select the condition to be used for setting segmentation parameters, the channels to be analyzed, and the segmentation channel

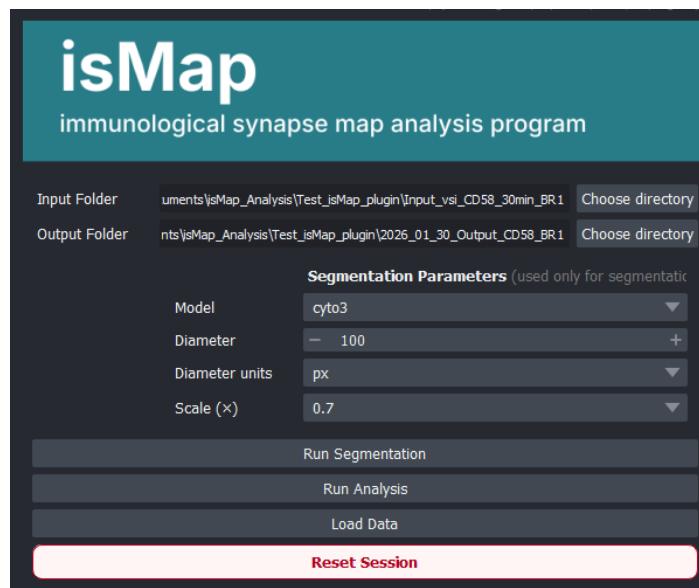


Figure 1: The isMap widget in napari.

4. After segmentation, select the circularity and diameter cutoffs for the full analysis (Figure 2 - left).
 - Distribution of cells are shown in the widget.
 - Toggle to see all the segmented images.
 - Only segmented cells assigned an ID are included in the analysis. Any changes to the cutoffs are reflected in the segmented images (Figure 2 - right).

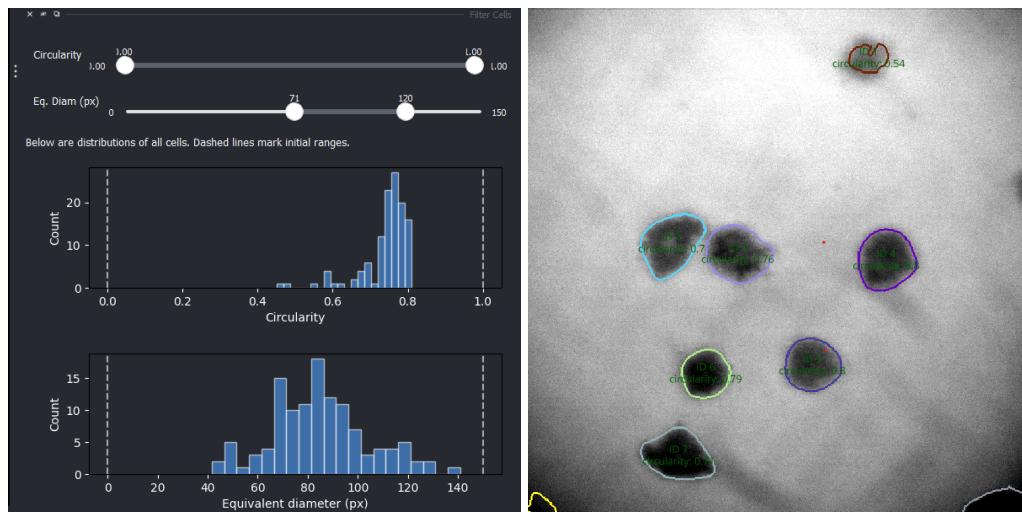


Figure 2: Adjusting circularity and diameter cutoffs for analysis. (Left) Cell distribution and cutoff sliders in the isMap widget. (Right) Segmented cells, with included cells labeled by ID.

5. Run the full analysis.

💡 Tip: If you have already run the analysis and wish to reload the results in napari, open the output folder using **Load Data** (Figure 1).

2.3 Output

After you click Run Analysis, isMap processes each input file and then shows the results inside the same napari window. A file called **filtered_points_export.csv** is saved to the output folder and contains all the measured metrics for each individual ROI. A **run.json** text file that records what you ran and with which parameters, so the results are reproducible.

2.3.1 Mean Fluorescence Intensity (MFI)

In the MFI tab of the isMap widget (Figure 3), MFI is displayed for each condition. Only one channel is shown at a time, and the channel can be selected using the **Metric** dropdown. Results can be viewed as raw (measured) values or as normalized values by enabling **Show normalized**. When normalized, MFI is normalized to the background signal within each condition.

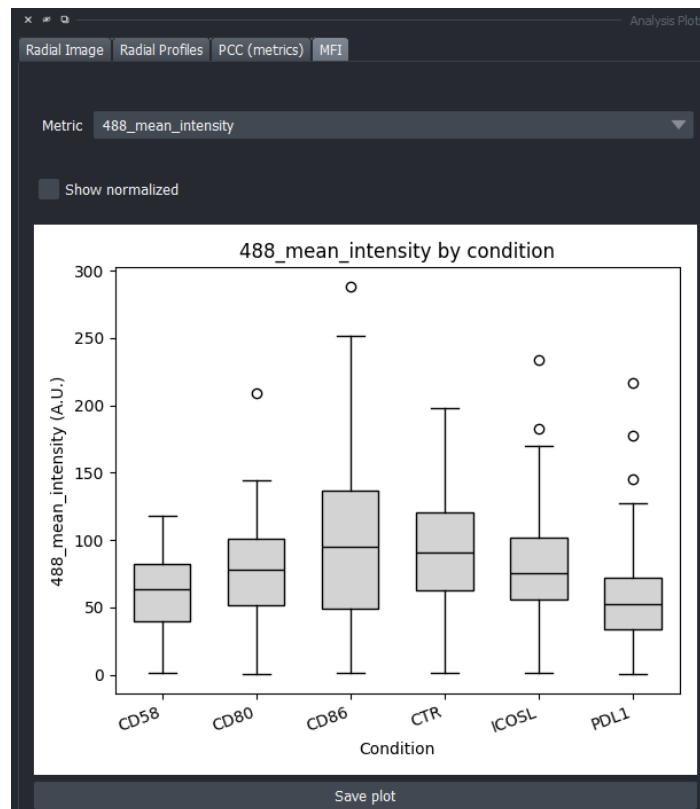


Figure 3: MFI tab in the isMap widget.

2.3.2 Pearson's correlation coefficient (PCC)

In the PCC (metrics) tab of the isMap widget (Figure 4), the PCC values are summarized as box plots for a selected target channel. The PCC is a measure of how linearly related the two channels are across pixels:

- R = 1: high intensity in channel A perfectly correlates with high intensity in channel B
- R = 0: No linear relationship.
- R = -1: Anti-correlation.

The condition and target channel can be chosen by the dropdowns **Condition** and **Target channel (Y-axis)**, respectively. Display options allow overlaying individual data points (**Show points**) and/or boxplots (**Show boxes**).

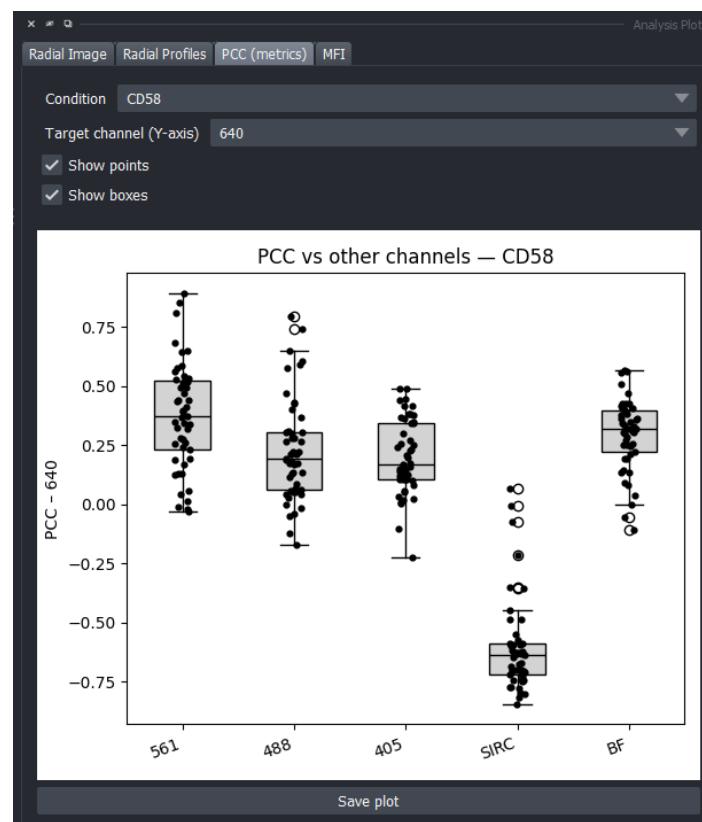


Figure 4: PCC tab in the isMap widget.

2.3.3 Radial images

The Radial Image tab displays a 2D radial image representation of the selected channel and condition (Figure 5). The Channel and Show dropdown controls which summary is displayed (e.g., Total Average), while Scaled (resize to target) rescales the radial image to match a target size for consistent comparisons across conditions. The Colormap dropdown sets the display colormap, such as viridis (Figure 5 - right). The color bar indicates intensity values used to render the radial image.

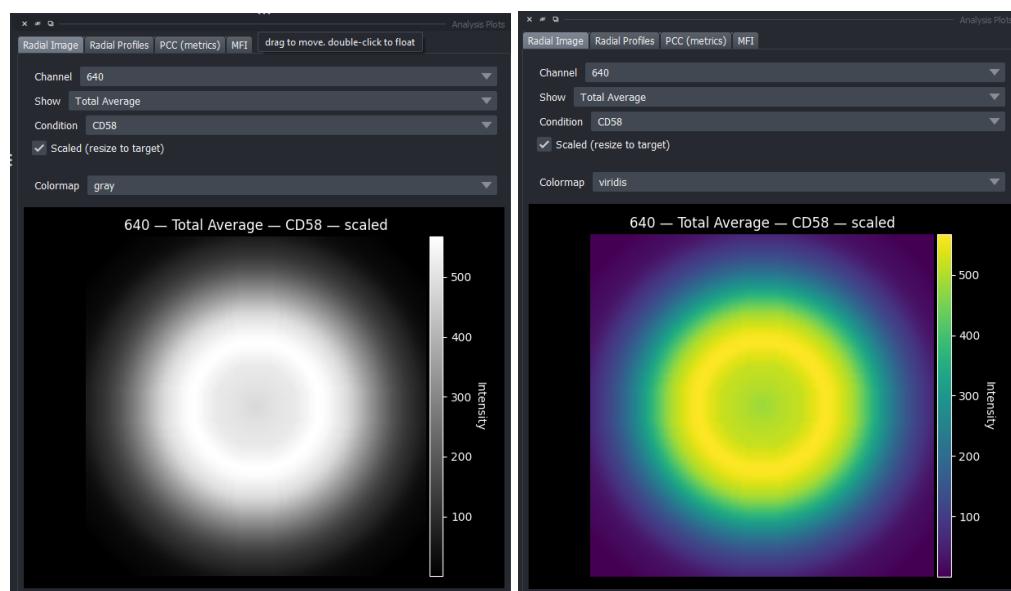


Figure 5: Radial image tab in the isMap widget. (Left) Grey scale and (Right) viridis colormap.

2.3.4 Radial intensity profiles

The Radial Profile tab plots the intensity of the radial image along the diameter of the radial image (Figure 6). The condition that is plotted can be selected from the **Condition** dropdown and the x-axis can either be normalized or shown in μm (**X-axis**). The image can also be rescaled to a set size, enabling direct comparison. Rescaling is performed using bilinear interpolation. Normalized MFI values measured along the diameter are plotted on the y-axis, with each channel scaled to the maximum MFI.



Figure 6: Radial profile tab in the isMap widget.

2.4 Example data set

An example image data set can be found [here](#).

The data set contains raw TIRFM images of primary human CD8+ T cells activated on supported lipid bilayers functionalized with ICAM-1, UCYT-1, and CD58.

TIRFM imaging was performed with an Olympus IX83 inverted microscope equipped with 405 nm, 488 nm, 561 nm and 640 nm laser lines, an Andor sCMOS Prime 95B camera and a 100x 1.45 NA oil immersion objective.

To analyze the data set, run each BR as individual inputs. The imaging channels are as follows:

- 405: ICAM-1
- 488: UCYT-1
- 568: Phalloidin/Actin
- 647: CD58
- SIRC
- Brightfield (BF)

We recommend using either phalloidin or brightfield (BF) as the mask channel.
