

ImageCUBE User Guide v0.5.7

Last edited on January 20th, 2020

Masaaki Uematsu

Department of Lipid Signaling, National center for Global Health and Medicine, Tokyo,
Japan.

Department of Lipidomics, The University of Tokyo, Tokyo, Japan

CONTENTS

0. ImageCUBE Features.....	3
1. Get started	3
1.1. <i>System Requirements</i>	3
1.2. <i>Installation</i>	3
1.3. <i>Conversion of hyperspectral data to .spc file</i>	3
2. Main Window	4
2.1. <i>Main Buttons</i>	4
2.2. <i>Preferences</i>	5
2.3. <i>Menu-bar and Shortcuts</i>	5
3. Data with one spectrum.....	6
3.1. <i>Comparison of Spectra</i>	6
3.2. <i>Source Code for Spectrum Subtraction</i>	8
4. Preprocessing of Hyperspectral Data.....	9
4.1. <i>Size setting</i>	9
4.2. <i>Cosmic Ray Removal</i>	10
4.3. <i>Noise Reduction based on Principal Component Analysis</i>	10
5. Basic Analysis of Hyperspectral Data	12
5.1 <i>Basic Image Reconstruction</i>	12
5.2. <i>Contrast/Histogram</i>	13
5.3. <i>Comparison of Spectra</i>	15
6. Spectrum Unmixing.....	16
6.1. <i>Cell-free Position</i>	16
6.2. <i>Unmixing with added spectra</i>	16
6.3. <i>Create and Export Unmixing Methods</i>	17
6.4. <i>Unmixing by using the Developed Method</i>	17
7. Batch Processing.....	18
7.1. <i>How It Works</i>	18
7.2. <i>Actions Specific to Batch Processing</i>	18
8. Plugins	20
8.1. <i>How to execute</i>	20
8.2. <i>How to write</i>	20
9. Python version and Python packages used.....	20

0. ImageCUBE Features

ImageCUBE is a Python-based open source software for analyzing hyper spectral data, which consists of x , y , and spectrum dimensions. The aim of this ImageCUBE is to provide a platform to create, share, and execute user-written analysis methods for hyper spectral data. In this term, ImageCUBE was implemented with a batch processing function as well as a function to import plugins. We ask users to cite the following paper:

<Not published yet.>

Thank you for your support.

1. Get started

1.1. System Requirements

Mac OSX 10.14 “Mojave” or later.

Windows version will be released soon.

Linux version is yet to be decided.

1.2. Installation

ImageCUBE can be downloaded from the following URL (**currently not available**).

<http://www.ncgmlipidsp.jp>

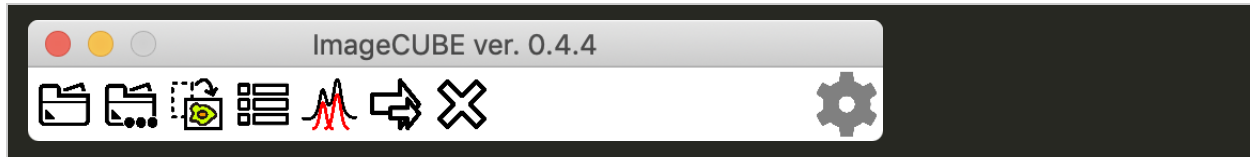
1.3. Conversion of hyperspectral data to .spc file

Currently, ImageCUBE (version 0.4.4) can only open .spc files of Thermo Galactic. Any acquired hyperspectral data should first be converted to .spc files by using a manufacturer-specific file converter. Conversion of data into .spc files from text files will soon be available on the ImageCUBE software. Details about the .spc file format of are available at

<https://www.yumpu.com/en/document/view/40416248/a-brief-guide-to-spc-file-format-and-using-gspcio>.

2. Main Window

2.1. Main Buttons



(open)

Open files with .spc extension. Multiple files can be selected. If the size (number of pixels in the x and y dimensions) of the data is not described in the .spc file, the size-setting popup will automatically appear (see [4.1. Size Setting](#)). The unmixing method with the extension .umx can also be opened (see [6.3. Create and Export Unmixing Methods](#)). You can also drag files from your file manager and drop them on the main window, and ImageCUBE will load the corresponding files.



(open recursively)

Files with .spc extension can be opened by searching all subdirectories of the selected directory.



(resize)

This button can be used to change the data size (number of pixels in x and y dimensions). Here, the size-setting popup will automatically appear (see [4.1. Size Setting](#)).



(map-spectrum table)

This button is used to open a window listing all of added map-images and spectra during the process of analysis (see [3.1. Comparing Spectra](#)).



(create unmixing method)

This button is used to open a window to create the unmixing method (see [6.3. Create and Export Unmixing Methods](#)).



(batch window)

Opens a batch processing window (see [7. Batch Processing](#)).



(close all)

Closes all opened windows.

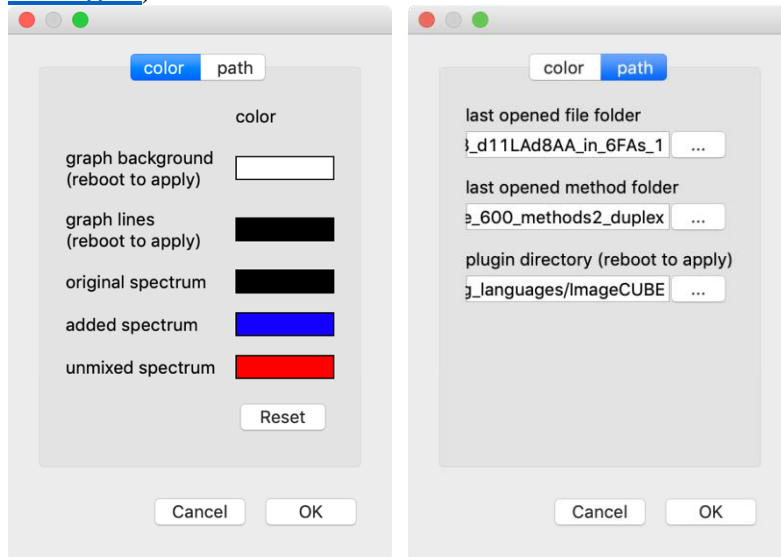


(preferences)

A window for general preferences will appear (see [2.2. Preferences](#)).

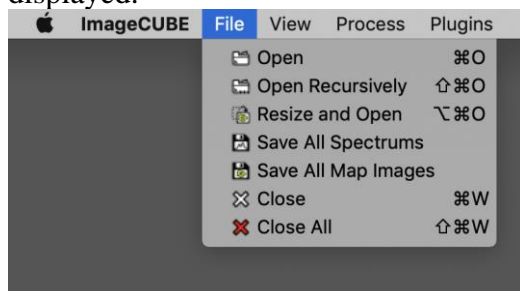
2.2. Preferences

Configuration of two basic preferences, the color of the spectrum and the settings of paths, can be optimized in this window. Setting plugin directory enables you to import plugins (see [8. Plugins](#)).



2.3. Menu-bar and Shortcuts


Some functions can be triggered via menu-bar, where all available shortcuts are also displayed.

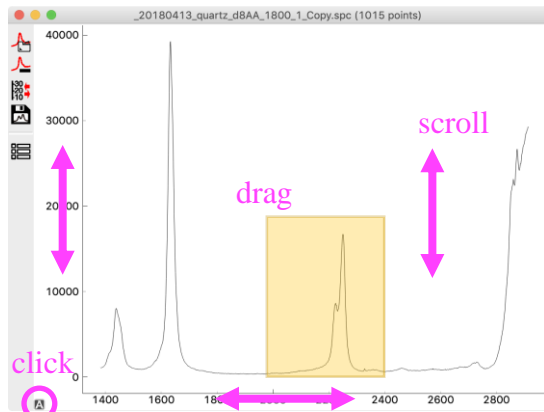


3. Data with one spectrum

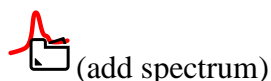
3.1. Comparison of Spectra

A window with the spectrum panel pops up when data with one spectrum is opened. To change the view, you can perform either of the following actions:

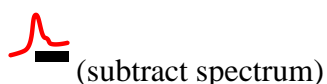
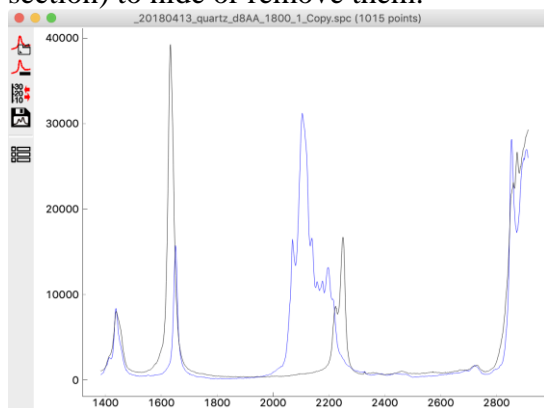
- Scroll around the horizontal or vertical axis, or inside the graph field.
- Drag inside the graph field to magnify a rectangular area.
- Click  at the bottom left corner to show all graphs.



The spectrum panel contains two virtual layers (view boxes 1 and 2) to display spectra. View box 1 is mainly used to display the original spectrum, and the added spectra during the process of analysis are mainly displayed on the view box 2. These two view boxes share the same horizontal axis, but have different vertical axes, thus enabling the comparison of spectra from different origins. The spectra in the view box 2, can be vertically enlarged by displaying the right axis (see the description of the “*right axis*” button in this section). In this window, the following functions are available.



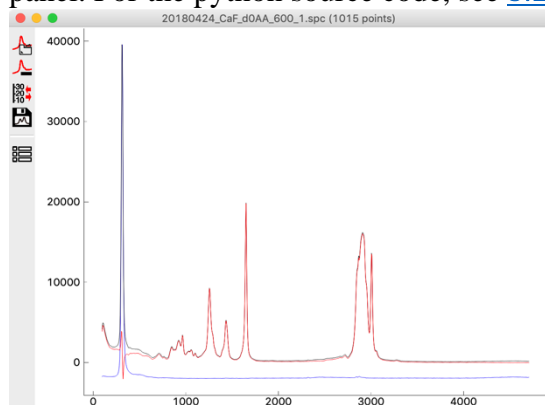
Spectra from other files can be added to the view box 2 in the spectrum panel. They can be used as spectra to subtract (see “*subtract spectrum*” button in this section) or as standard spectra for unmixing (see [6.2. Unmixing with added spectra](#)). Added spectra can be selected through the Map-Spectrum table (see the description of “*Map-Spectrum table*” button in this section) to hide or remove them.




This method is used to purify spectra by subtracting the background from the measured data. By clicking on this button, a popup appears to ask for the spectrum range (r_1 and r_2), and the added spectrum is subtracted from the original spectrum as follows:

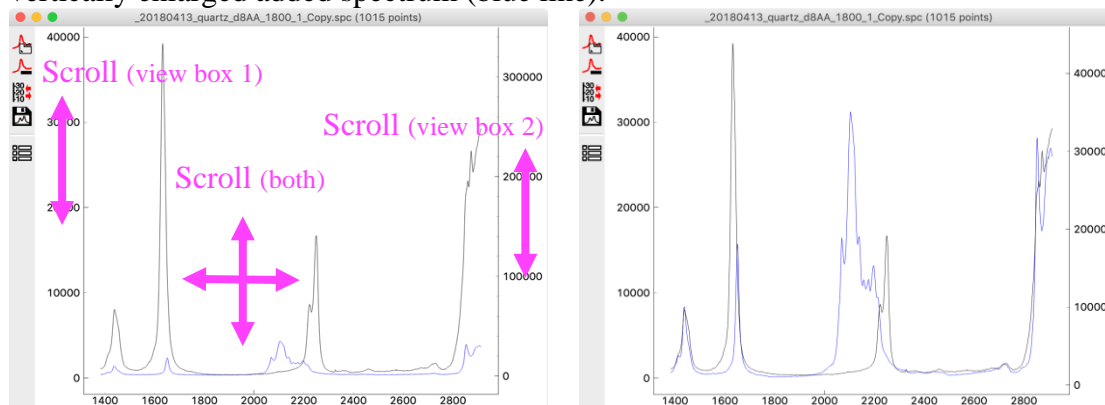
$$\mathbf{sub} = \mathbf{ori} - a \cdot \mathbf{add}$$


where **ori**, **add**, and **sub** represent the vectors of the original, added, and subtracted spectra. Weighting value a is calculated such that the region from r_1 to r_2 of **sub** is as linear as possible. In the following figure, black, blue and red lines indicate the **ori**, **add**, and **sub** spectra, respectively. The generated spectrum is added to the view box 2 in the spectrum panel. For the python source code, see [3.2. Source Code for Subtracting Spectra](#) section.



 (right axis)

This button shows the axis for the view box 2 (where mainly the added spectra are added to) at the right side of the spectrum panel. By scrolling on the right axis, only spectra added in the view box 2 are vertically enlarged, while scrolling on the left axis enlarges only spectra in the view box 1 (where the original spectrum is located). The following figures show the vertically enlarged added spectrum (blue line).

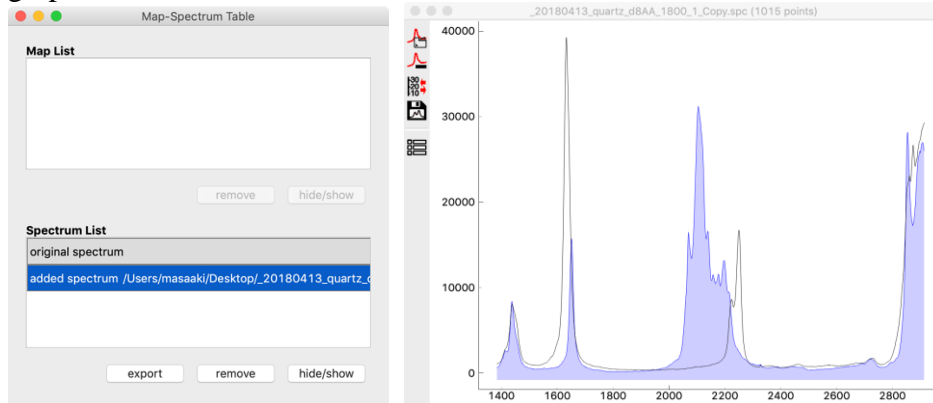


 (save spectrum)

This command saves the currently displayed spectrum view as a .svg file in the folder from which the original spectrum was obtained. No message dialog will appear.

 (Map-Spectrum table)

This button is used to opens the Map-Spectrum table, showing all items displayed in either the map panel or the spectrum panel. The selected items in the table are highlighted in the graph.



3.2. Source Code for Spectrum Subtraction

/src/main/python/Modules/draw.py

```
class SpectrumWidget(pg.PlotWidget):
    ...

    def spectrum_linear_subtraction(self, sRS, eRS):
        sRS_idx, eRS_idx = np.sort([self.spc_file.get_idx(sRS), self.spc_file.get_idx(eRS)])
        added_x_list = self.parent.spectrum_widget.vb2.addedItems[0].xData
        added_y_list = self.parent.spectrum_widget.vb2.addedItems[0].yData
        order = np.argsort(added_x_list)
        added_x_list = added_x_list[order]
        added_y_list = added_y_list[order]
        x_list = self.vb2.addedItems[0].xData
        x_min = self.spc_file.x.min()
        x_max = self.spc_file.x.max()
        x_min = max([x_min, min(x_list)])
        x_max = min([x_max, max(x_list)])
        common_x_region = (x_min <= self.spc_file.x) & (self.spc_file.x <= x_max)
        self.umx_x_list = self.spc_file.x[common_x_region]
        master_x_list, master_y_list = self.spc_file.get_data(sRS, eRS, sub_idx=0)
        added_regional_y_list = np.interp(master_x_list, added_x_list, added_y_list)

        def morm(n):
            diff_y_list = master_y_list - n * added_regional_y_list
            params, residuals, rank, s = np.linalg.lstsq(np.vstack([master_x_list, np.ones(len(master_x_list))]).T, diff_y_list)
            return residuals[0]

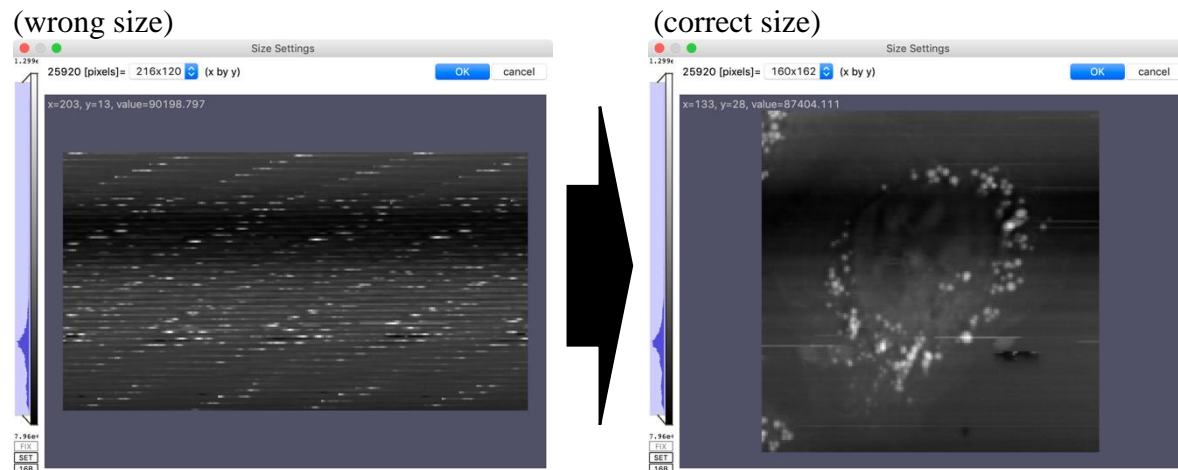
        optimization_results = minimize_scalar(rnorm, bounds=(0, np.inf))
        self.umx_height_matrix = np.array([np.append(-optimization_results.x, [1])])

        # [:, 0] and [:, 1] represents added and original spectrum, respectively
        self.umx_y_matrix = np.empty((len(self.umx_x_list), 2), dtype=float, order='F')
        self.umx_y_matrix[:, 0] = np.interp(self.umx_x_list, added_x_list, added_y_list)
        self.umx_y_matrix[:, 1] = self.spc_file.sub[0].y[common_x_region]
```

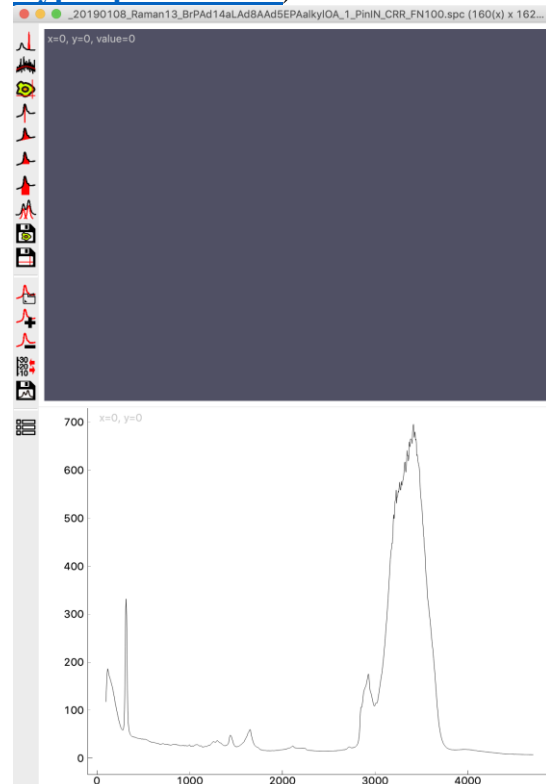

4. Preprocessing of Hyperspectral Data

4.1. Size setting

When opening a .spc file for the first time, the size-settings dialog (a dialog to set number of x and y pixels) appears. Once the size is decided, this information is automatically recorded in the original .spc file so that the no dialog will appear when the same file is opened for the next time. The displayed image is constructed based on the whole area under the curves of spectrum.

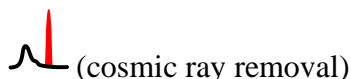


After the size is set, a window with the map image panel and the spectrum panel opens. The map image panel at the top displays the constructed map images (see [5. Basic Analysis of Hyperspectral Data](#)).



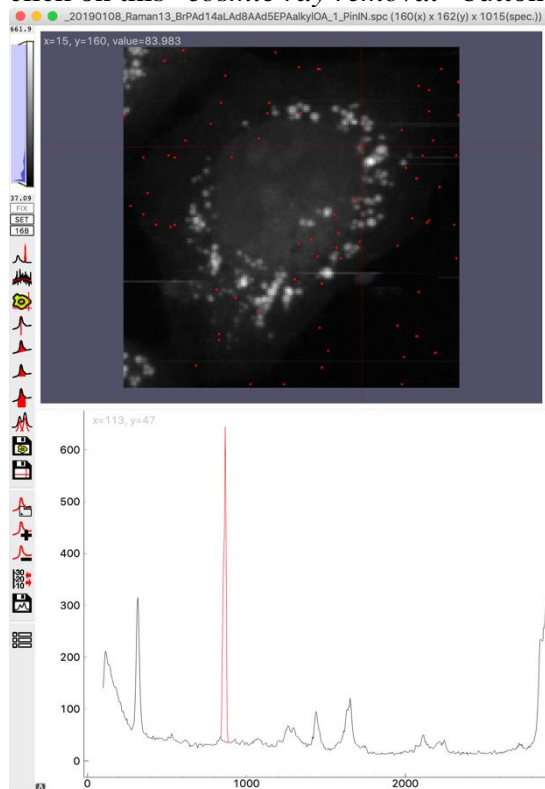
To change the data size again, click the “*resize*” button in the main window (see [2.1. Main Buttons](#)).

4.2. Cosmic Ray Removal



(cosmic ray removal)

The detection of cosmic rays takes time (approximately 5 seconds for 10,000 pixels), hence the information about the detected cosmic rays is saved to the original file once this process is executed, and is automatically displayed when it is opened next time. After the process, pixels with cosmic rays are colored in red. The detected and removed cosmic rays are represented as red lines in the spectrum panel. To hide or clear the detected cosmic rays, right click on this “*cosmic ray removal*” button.



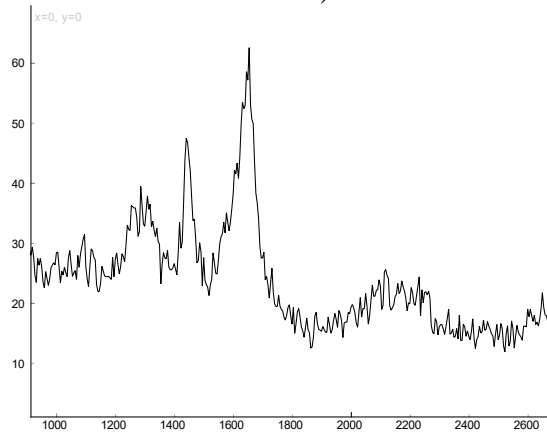
4.3. Noise Reduction based on Principal Component Analysis



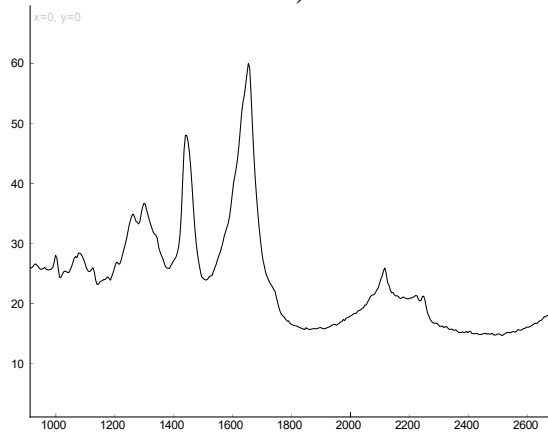
(PCA-based noise reduction)

This command reduces noise reduction based on principal component analysis (PCA). After executing this process, a window to enter the number of components to include pops up. The default value is 100. The results of the noise reduction are not saved to the original file, but cannot be undone on the displayed window. Thus, to redo the noise reduction, please re-open the file.

(before the noise reduction)



(after the noise reduction)



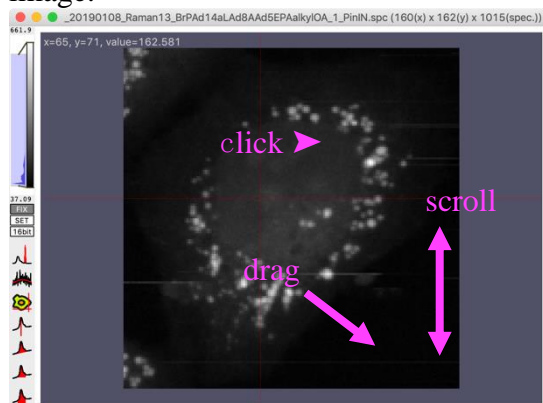
(set cell free position)

See [6.1. Cell free Position.](#)

5. Basic Analysis of Hyperspectral Data

5.1 Basic Image Reconstruction

Several methods to construct map images from hyperspectral images are prepared by default. Constructed images can be magnified and moved by scrolling and dragging in the map image panel. Spectra at different pixels can be displayed by clicking the pixel-of-interest in a map image.



(signal intensity)

This command allows the construction of map images based on the intensity at a certain spectrum point for every pixel. Constructed map images are added to the Map-Spectrum table (see “*Map-Spectrum table*” item in [3.1. Comparing Spectra](#)).



(signal to baseline)

This command allows the construction of map images based on the area between the curve and the baseline. The constructed map images are added to the Map-Spectrum table (see “*Map-Spectrum table*” item in [3.1. Comparing Spectra](#)).



(signal to horizontal baseline)

This command allows the construction of map images based on the area between the curve and the horizontal baseline. The constructed map images are added to the Map-Spectrum table (see “*Map-Spectrum table*” item in [3.1. Comparing Spectra](#)).



(signal to axis)

This command allows the construction of map images based on the area under the curve. The constructed map images are added to the Map-Spectrum table (see “*Map-Spectrum table*” item in [3.1. Comparing Spectra](#)).



(save all map images)

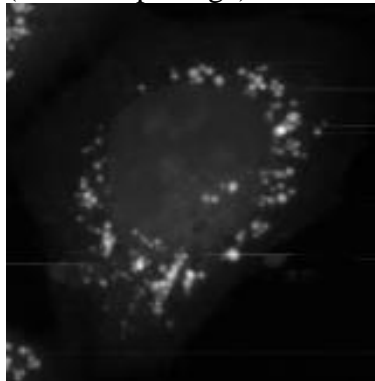
All the map images displayed in Map-Spectrum table are saved as 8-bit, 16-bit, or floating-point 32-bit gray-scale tiff files. The contrast condition can be changed when saving the files (see [5.2. Contrast/Histogram](#)). The saved files are named automatically based on how images are constructed.



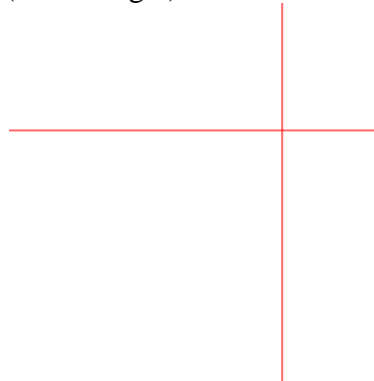
(save target)

The position of the currently selected pixel in the map image is saved as a .svg file with crosshairs; it can be used to display positions in map images by overlaying the output crosshairs onto the saved map images. The saved files are named automatically based on the x and y coordinates of the target position.

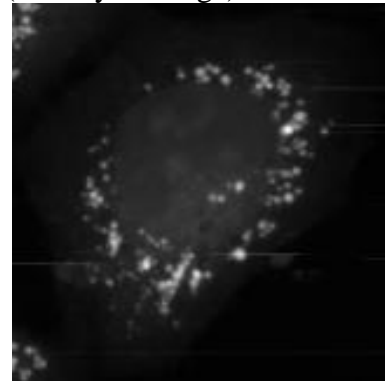
(saved map image)



(saved target)

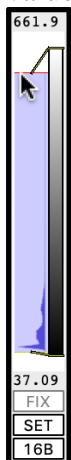


(overlayed image)

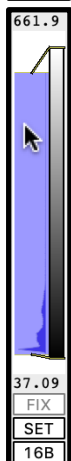


5.2. Contrast/Histogram

Contrast/Histogram bar appears after the construction of map images. The contrast can be changed by moving the top or bottom bars representing the maximum and the minimum values of the contrast, or dragging the blue contrast box sandwiched between these bars.

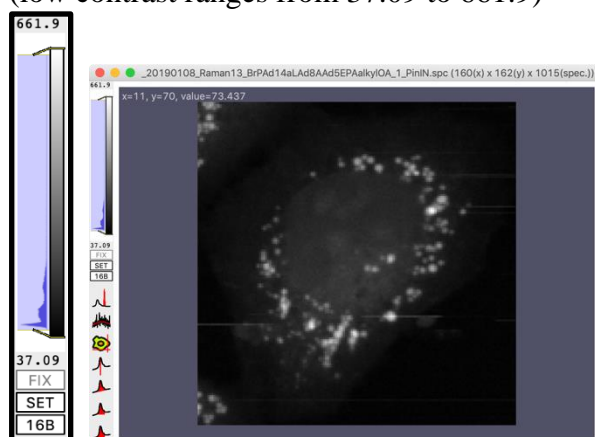


The bars turn to red when you hover the mouse pointer over them.

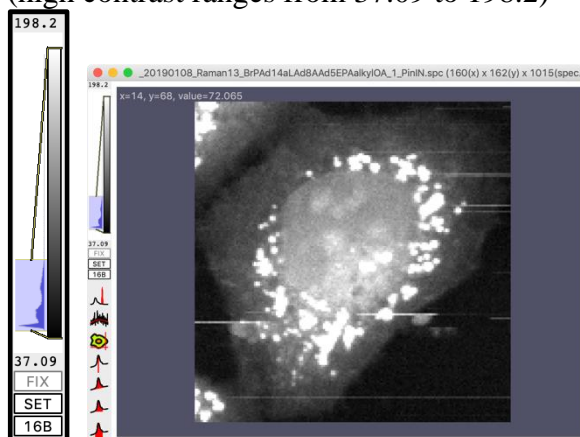


The box becomes dark when you hover the mouse pointer over it.

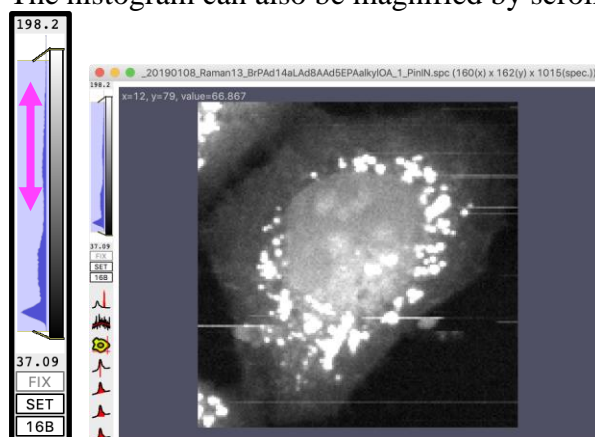
(low contrast ranges from 37.09 to 661.9)



(high contrast ranges from 37.09 to 198.2)



The histogram can also be magnified by scrolling.



When the button is pressed, it becomes into dark as . Under this condition, the contrast of the map images is fixed even when the displayed map images are changed through the “Map-Spectrum table” (see [3.1. Comparing Spectra](#)). The contrast can still be changed through “Contrast/Histogram bar”, by moving the top and bottom bars, or by dragging the contrast box. When the button is light as , the contrast is automatically changed when the displayed map image is changed, based on the maximum and the minimum values in a map image.

When saving map images (see [5.1. Basic Image Reconstruction](#)), if the button is pressed and appears as , all images will be saved with current contrast. Otherwise (when it appears as), the contrast is automatically changed image by image. Images are automatically named based on the method through which they were constructed and the contrast settings, and then saved.

By pressing the button, the range setting popup appears and the direct value of the contrast can be entered.

When the **16B** button is clicked, the button changes to the **8bit** or **32bit** button, and the images can be saved with the different bit depths. When images are saved in 32 bit, the contrast settings are ignored and the values of each pixel are saved as is, including negative values, although the file sizes are doubled compared to when the images is saved as 16 bit.

5.3. Comparison of Spectra



(add spectrum)

See [3.1. Comparison of Spectra](#).



(add current spectrum)

The current displayed original spectrum is added to the spectrum panel, and remains in the spectrum window even when the spectrum in other pixels are displayed by clicking on the pixel-of-interest in the map image.



(subtract spectrum)

The added spectrum is subtracted from the original spectrum for each pixel. See the description of “*subtract spectrum*” button in [3.1. Comparison of Spectra](#) for more details.



(save spectrum)

See [3.1. Comparison of Spectra](#).



(right axis)

See [3.1. Comparison of Spectra](#).



(Map-Spectrum table)

See [3.1. Comparison of Spectra](#).

6. Spectrum Unmixing

The basic workflow of spectrum unmixing is as follows:

1. Add standard spectra to the view box 2 in the spectrum panel.
2. (optional) Add spectrum at the cell-free position to the view box 2.
3. Initiate spectrum-unmixing procedure by clicking the “*unmixing*” button.
4. Set spectrum ranges to perform unmixing.

6.1. Cell-free Position



(set cell-free position)

Set a cell-free position and save the x and y coordinates into the original file. This command is useful when executing the spectrum unmixing method with the “*background option*” created by users (see [6.4. Unmixing by using a Method](#)). When the “*background option*” is turned on in the unmixing method, the spectra at the cell-free position are automatically used as one of the standard spectra for spectrum unmixing.

This procedure is different for step 2 of the basic workflow, and only setting the cell-free position does not add the spectrum at the cell-free position to view box 2. If you try to unmix without using the spectrum unmixing method, but want to use the spectrum at the cell-free position as one of the standard spectra, it should be directly added to view box 2 by clicking on the “*add current spectrum*” button (see [5.3. Comparison of Spectra](#)).

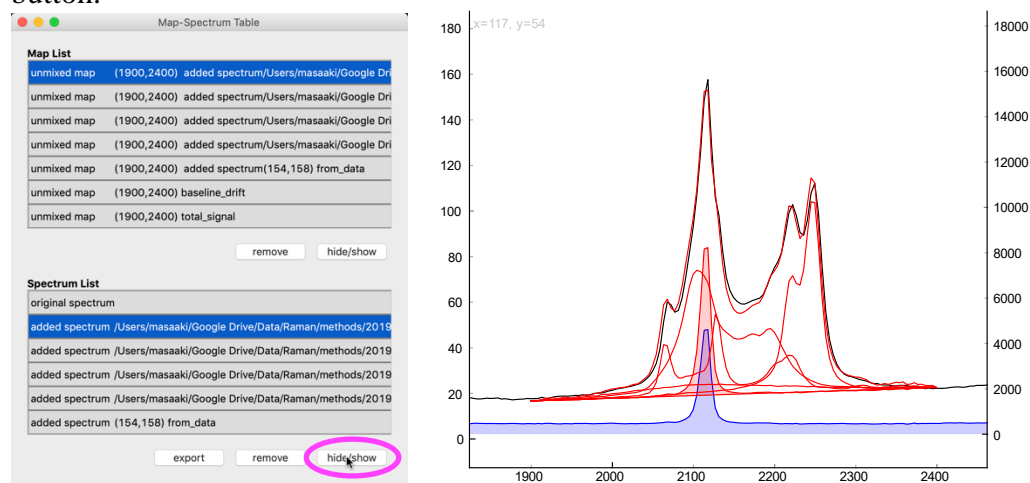
See also [6.3. Create and Export Unmixing Methods](#).

6.2. Unmixing with added spectra



(unmixing button in a data window)

By pressing this button, The range-setting window pops up. After clicking on the “ok” button, spectrum unmixing is executed based on the added spectra to view box 2. Map images constructed based on the result of spectrum unmixing are added to the “*Map-Spectrum table*”. The added spectra in view box 2 are automatically hidden and not displayed in the spectrum panel. To show the added spectra in view box 2, select the spectrum-of-interest from the spectrum list in the “*Map-Spectrum table*”, and click on the “*hide/show*” button.



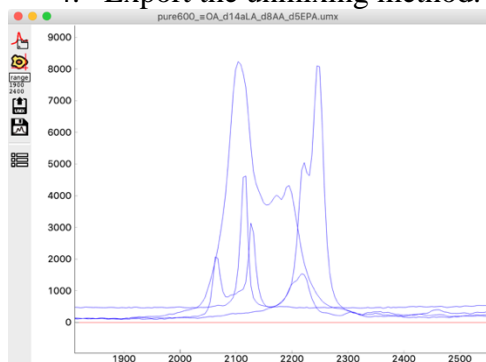
6.3. Create and Export Unmixing Methods



(unmixing button in the main window; see [2.1. Main Buttons](#))

By pressing this “unmixing” button in the main window, a window to create spectrum unmixing method pops up. The basic workflow for developing a spectrum unmixing method is as follows:

1. Add spectra used for spectrum unmixing.
2. (optional) Set whether or not to include a spectrum from the cell-free position as the standard spectra.
3. Set range for spectrum unmixing.
4. Export the unmixing method.



The following functions are available.



(add spectrum)

The spectra are added in the same way as written in [3.1. Comparison of Spectra](#), except that the spectra are not added to the view box 2, but to the view box 1. Hence, the button to show the right axis does not exist.



(include/exclude spectrum from a cell-free position)

Switch the option of whether to include a spectrum from cell-free position to the standard spectra. When it is included, a horizontal red line appears in the spectrum panel. See also [6.1. Cell free Position](#).



A button to set the range for spectrum unmixing.



(export unmixing method)

A button to export the unmixing method.

6.4. Unmixing by using the Developed Method



(unmixing by using the developed method; right click on the button in a data window)

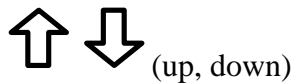
A window to select .umx file pops up. The unmixing method created in [6.3. Create and Export Unmixing Methods](#) can be executed.

7. Batch Processing

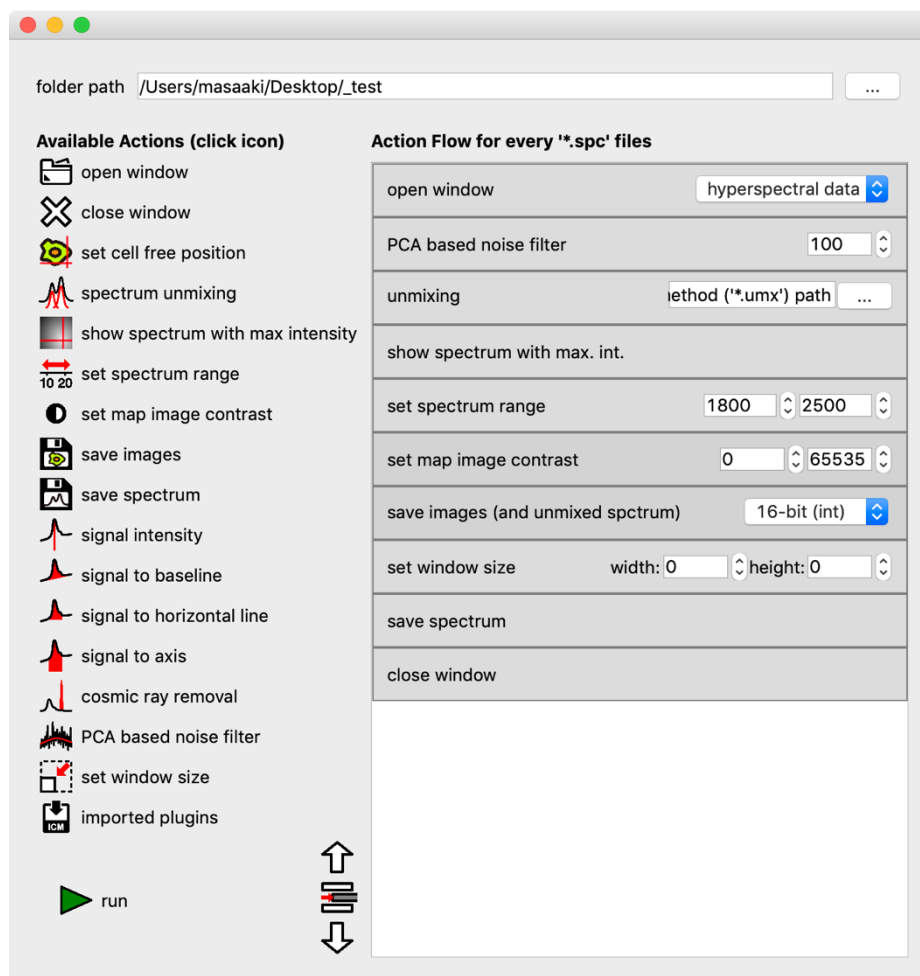
7.1. How It Works


By clicking the “*batch window*” button (see [2.1. Main Buttons](#)), a window to edit batch processing appears. The available actions are listed at the left side. By clicking the icons, corresponding actions are added to the action flow.

The order of actions can be changed by selecting the action block in the action flow and pressing either of the following buttons:



Added actions can be removed by pressing the remove button:



Enter the folder path at the top of the window, and press the “*run batch*” button:  to execute a batch. Files with .spc extensions in the folder are recursively collected and actions in the action flow are individually executed.

7.2. Actions Specific to Batch Processing





(show spectrum with max intensity)



(set window size)

Size zero indicates that the window size is not changed.



(set map image contrast)



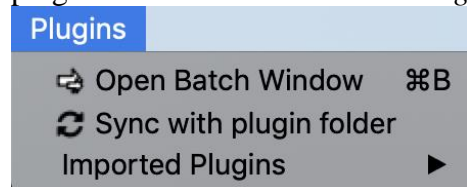
(imported plugins)

Plugins created and/or imported (see [8. Plugins](#)) can also be incorporated into the batch processing.

8. Plugins

8.1. How to execute

Files ending with the .icm extension in the plugin directory (see [2.2. Settings](#)) are automatically imported as plugins when ImageCUBE application is opened. The imported plugins can be executed from “*Plugins > Imported Plugins*”.



When a new plugin file with the .icm extension is added to the plugin folder or the content of the existing .icm file is edited, they can be updated and imported by pressing “*Plugins > Sync with plugin folder*”.

8.2. How to write

A plugin function can be written in Python syntax and is simply a text file with the .icm extension.

The main and the current focused windows can be accessed using the following code, from which you can access default functions to process hyperspectral data.

/plugin folder/sample_plugin.icm

```
self # main window
self.current_focused_window # current focused window
```

The source code of ImageCUBE is uploaded on GitHub (<https://github.com/MasaakiU/ImageCUBE>) to help you to develop custom plugins, which also comprises the sample plugin files.

9. Python version and Python packages used

Python	3.6.6
numpy	1.15.1
scipy	1.1.0
spc	0.4.0
PyQt5	5.12.2
pyqtgraph	0.10.0
Pillow	4.2.1
fbs	0.8.3
Cython	0.28.5