SINGAPORE IMMUNOLOGY NETWORK, A*STAR, SINGAPORE

HSB surface creation manual

5/21/2018

Table of Contents

Hue-Saturation-Brightness Surface Creation Scripts

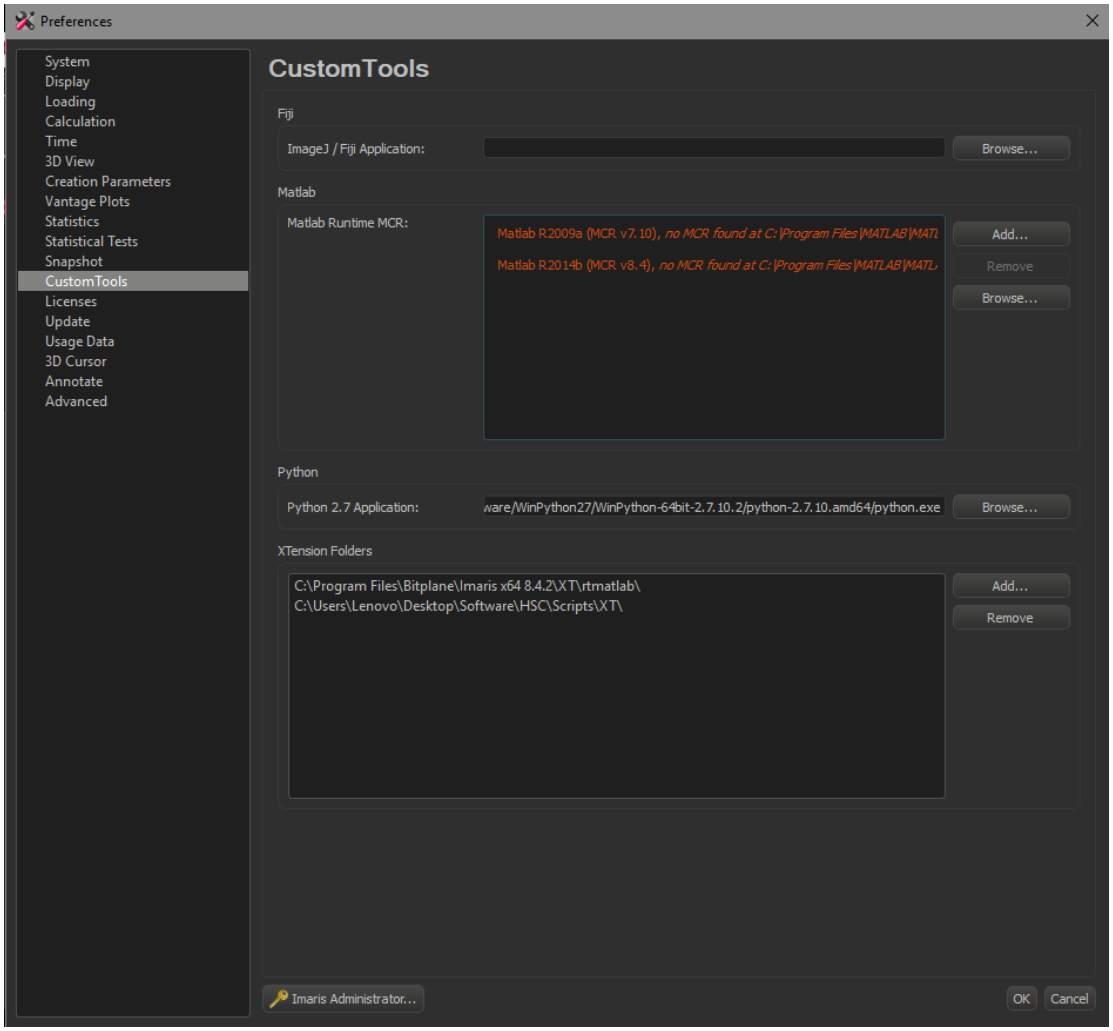
Getting started

In order to visualize the hue of surfaces and cells, a customized look up table for the hue channel has been created for Fiji and Imaris.

The time needed to install the hue-saturation-brightness surface creation scripts is less than a minute on a typical computer.

Setting up Python 2.7 for Imaris

To set up Python 2.7 for Imaris XT, go to **File** > **Preferences** > **CustomTools, click Browse under Python 2.7 Application and select the drive where WinPython 2.7 is installed. Select the file path for the python executable,** *<WinPython>/<WinPython-64bit-2.7.10.2>/<python.exe>.*



Location for installing Python 2.7 for Imaris XT

Installing Hue Look Up Tables for Imaris

Go to the drive where Imaris is installed, *Bitplane/<Imaris version> /colorTables* and paste the LUT labelled *HSB_Hue* in. Imaris will automatically detect the LUT the next time it starts up.

Importing Imaris XT script into Imaris

Go to the drive where Imaris is installed, *Bitplane/<Imaris version> /XT* and paste the XT script labelled *XTIsolateSurfacesWithCsvList.py* in. Imaris should display the option for selecting the XT script under **Image Processing** the next time it starts up.

Importing HSB surface creation Look Up Tables into FIJI

To load in the *HSB_hue.lut* into FIJI, go to *<FIJI folder>\FIJI.app\luts* and paste the *HSB_Hue* in. Do the same for the *Backgroundpicker.lut*. FIJI will automatically detect the LUT the next time it starts up.

Important!

Note: For the HSB scripts, it is important that the TIFF files of the image are in 32 bit, and not in 16 bit.

If you experience the following error message shown below, it is because the data is not in 32 bit.

```
Directory chosen: C:/Users/yrtan/Desktop/3R/P01_Renamed TIFs
...Started processing. Please wait until program says it is done.
"P01_Rename TIF files_log.txt" was ignored.
"P03_Merge channels_log.txt" was ignored.

>> Reading in data...
Traceback (most recent call last):
  File "C:\Users\yrtan\Desktop\HSB manuscript scripts\P03A_Merge channels with m
aximum intensity transform_2017 09 09.py", line 142, in <module>
    img = openAsArray(filename)
  File "C:\Users\yrtan\Desktop\HSB manuscript scripts\P03A_Merge channels with m
aximum intensity transform_2017 09 09.py", line 37, in openAsArray
    im = numpy.array(PIL.Image.open(filename).convert(mode)).astype(type)
  File "D:\Software\WinPython\WinPython-64bit-3.5.1.3\python-3.5.1.amd64\lib\sit
e-packages\PIL\Image.py", line 2309, in open
    % (filename if filename else fp))
OSError: cannot identify image file 'q_t0000_c00_x000_y000_z0000.tif'


...Done. Press 'Enter' to exit.
```

Error message if image data is not in 32 bit

In order to save out TIFF files in 32 bit from Imaris, go to the **Edit > Change data type> 16 bit to 32 bit** and save out the files. In order to save out TIFF files in 32 bit from Fiji, go to **Image > type> 32-bit** and save out the files.

Suggested Workflow

1) Open the image file in Fiji or Imaris, and save it out as a 32 bit image.
2) Rename the images so that the downstream scripts can read in the images.
3) (Optional) If there are more than 3 channels, merge the channels using the maximum intensity channel script.
4) Generate the Hue, Saturation and Brightness channels.
5) Load in the Hue, Saturation and Brightness channels into Imaris or any other program that can create 3-dimensional surfaces where metrics can be extracted.

6) Use the brightness/ maximum intensity channel for surface creation.
7) (Optional – if spectral compensation is required)
    a. Load in the single stains into Imaris or any other program that can create 3-dimensional surfaces where metrics can be extracted.
    b. Use the channel of the single stain for surface creation.
    c. Export the statistics of the surfaces in a CSV format.
    d. Load the statistics into FlowJo.
    e. Gate the single stains and derive the spillover coefficients.
    f. Key in the spillover coefficients into the compensation matrix and that the values are accurate.
    g. Key in the values from the compensation matrix into the Channel unmixing script and use it to generate the corrected channels.
    h. Load in the corrected channels into the Imaris file with surfaces.
8) Export statistics from the surfaces.
9) Aggregate the statistics into a single comma separated values (CSV) file.
10) Load in the CSV file into FlowJo or another software that can carry out gating (eg. XiT).
11) Gate the cells and export the populations in a CSV format.
12) (Optional) Visualise the gated surfaces in the image file (Imaris XT only) using the XTIsolateSurfacesWithCsvList script.

The time needed to merge and generate the hue, saturation and brightness channels is dependent on the size of the raw data, but for a typical four channel, 16-bit single tile image acquired on the confocal microscope, it should take less than a minute on a typical computer.

Naming Convention used in scripts

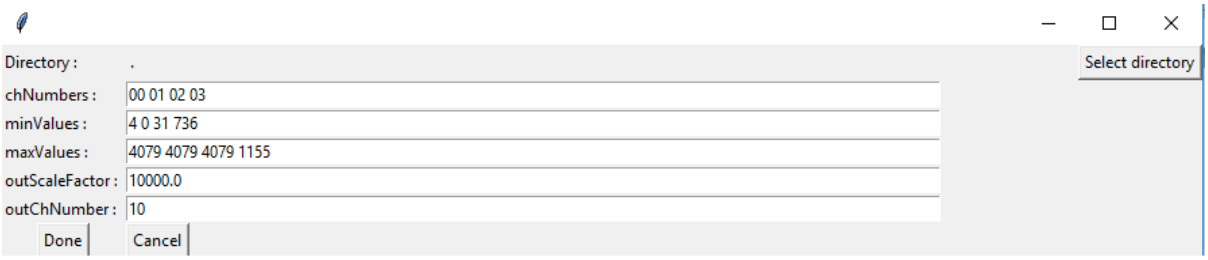
e.g. *q_t0001_c01_x000_y000_z0001.tif*

q: prefix
t: timepoint
c: channel number
x: value of X position
y: value of Y position
z: value of Z position

The naming convention is important as the scripts read in the files in a specific manner. When TIF files are saved out from Imaris, it is necessary to rename the files to fit the current naming convention.

How to input values into the graphical user interface

The default values for each script will pop up automatically each time the graphical user interface is launched. The user can use these default values, together with the options listed in this manual for each script as a reference on the type of values to input. Values in a list (eg. for channel numbers in the example shown below) can be separated by a space or by a comma.



Graphical user interface for the *P02 Calculate max intensity channel* ***script***

Renumbering files utility script

Script name: *S00_Renumber files utility script*

Aim: To renumber files in the event where either the prefix, timepoint, channel, x, y or z values are not in the desired sequence

Description: This utility script provides the user with the option of renumbering files, as there may be certain occasions where certain programs save out files beginning with 0, while others save out files beginning with 1.

Input: TIF files to be renumbered. The TIF files need to already have been renamed to the standard format (script 1).

Output: Renumbered TIF files will be duplicated and saved in the subfolder *S00_Renumbered TIFs*.

Parameter	Description
tForceAdjacent cForceAdjacent xForceAdjacent yForceAdjacent zForceAdjacent	ForceAdjacent forces the channel numbers to be sequential in order for the t, c, x, y, z axis. Possible values are (1) on or off (0).
tRestartNumbering cRestartNumbering xRestartNumbering yRestartNumbering zRestartNumbering	RestartNumbering forces the channel numbers to be changed based on the for the t, c, x, y, z axis. Possible values are (1) on or off (0).
tFlipSequence cFlipSequence xFlipSequence yFlipSequence zFlipSequence	FlipSequence forces the channel numbers to reverse in order for the t, c, x, y, z axis. Possible values are (1) on or off (0).
tRestartNumber cRestartNumber xRestartNumber yRestartNumber zRestartNumber	Starting number of its corresponding parameter if RestartNumbering is turned on
pOverrideExisting	Turn on(1) or off(0) to trigger if prefixes will be changed

pOverrideString	All prefixes found will be changed to this string if pOverrideExisting is turned on
swapAxis	Turn on(1) or off(0) to trigger swapping of axes
axesToSwap	Sets the axes for swapping if ' swapAxis ' is turned on. Axes for swapping allowed: 'tc', 'tx', 'ty', 'tz', 'cx', 'cy', 'cz', 'xy', 'xz', 'yz'

Example:

Modes	**New Channel Numbers**
cForceAdjacent = 0 cRestartNumbering = 0 cFlipSequence = 0	[1, 2, 7, 15, 16, 18] #Original sequence
cForceAdjacent = 1 cRestartNumbering = 0 cFlipSequence = 0	[1, 2, 3, 4, 5, 6] #Channel 16 is now 5 (4th number after 1st channel)
cForceAdjacent = 0 cRestartNumbering = 1 cFlipSequence = 0 cRestartNumber = 17	[17, 18, 23, 31, 32, 34] #Channel 16 is now 32 (was 15 larger than 1st channel, and 1st channel is now 17)
cForceAdjacent = 0 cRestartNumbering = 0 cFlipSequence = 1	[18, 16, 15, 7, 2, 1] #Channel 16 is now 2 (numbering is flipped around)
cForceAdjacent = 1 cRestartNumbering = 1 cFlipSequence = 0 cRestartNumber = 17	[17, 18, 19, 20, 21, 22] #Channel 16 is now 21 (1st channel is now 17, and channel 16 was 4th number after)
cForceAdjacent = 1 cRestartNumbering = 1 cFlipSequence = 1 cRestartNumber = 17	[22, 21, 20, 19, 18, 17] #Channel 16 is now 18 (= 5th example above + flip sequence)
cForceAdjacent = 1 cRestartNumbering = 0 cFlipSequence = 1	[6, 5, 4, 3, 2, 1] #Channel 16 is now 2 (= 2nd example + flip sequence)
cForceAdjacent = 0 cRestartNumbering = 1 cFlipSequence = 1 cRestartNumber = 17	[34, 32, 31, 23, 18, 17] #Channel 16 is now 18 (= 3rd example + flip sequence)

Rename TIF files to standard format

Script name: *P01_Rename tif files into standard format*

Aim: To rename the TIFF files to the standard naming format.

Description: As several different imaging programs utilize different naming conventions, it is necessary to standardize the file names for input into the HSB surface creation scripts. This script will attempt to match the file name format as entered by the user in the matchstring parameter according to re.match python scripting style. If matched, it attempts to assign XYTZC parameters to identified text, based on the numerical positions given in the script parameters. If the file name is not matched, it attempts a series of standard FIJI/Imaris style conventions. If still not matched, program does not rename it after making a copy into the folder. The log file indicates which naming convention is used to rename the individual files.

The naming conventions accepted by the script are listed below:

FIJI naming conventions:

a. TZC - *FIJI_t006_z007_c008.tif*
b. TZ - *FIJI_t009_z010.tif*
c. ZC - *FIJI_z011_c012.tif*
d. TC - *FIJI_t013_c014.tif*
e. C - *FIJI_c015.tif*
f. T - *FIJI_t016.tif*
g. Z - *FIJI0025.tif*

Imaris naming conventions:

h. TCZ - *Imaris_T17_C18_Z019.tif*
i. TZ or T only - *Imaris_T20_Z021.tif*
j. CZ or C only - *Imaris_C22_Z023.tif*
k. Z - *Imaris_Z024.tif*

Parameter	Description
matchstring	Custom file naming format input by the user.
ttt	Parameter for time; set to -1 if it does not exist in the file name.
ccc	Parameter for channel number; set the number of channels here.
xxx	Parameter for x; set the number of x; set to -1 if it does not exist in the file name.
yyy	Parameter for y; set the number of y; set to -1 if it does not exist in the file name.
zzz	Parameter for z; set the number of z here.
prefix	Set the prefix for the renamed images; the default is "q".

Input: All channels acquired during imaging

Output: Renamed image files in folder *P01_Renamed TIFs*

Reduce number of channels to 3 by merging channels

Merge channels with maximum intensity transform

Script name: *P02_Calculate max intensity channel*

Aim: To merge the maximum intensities of multiple channels into one channel.

Description: If the user has more than three channels, it is necessary to merge channels together so the information from both channels can be captured for cell surfacing. The current script allows for the merging of multiple channels together by performing a maximum intensity projection of the input channels.

Parameter	**Description**
inChNumbers	Channel numbers of input channels. Provide a list
minValues	Minimum values of input channels. Provide a list in the same sequence as the channel numbers.
maxValues	Maximum values of input channels. Provide a list in the same sequence as the channel numbers.
outScaleFactor	Scale factor to multiply the output channel with.
outChNumber	Channel number of single output channel

Input: Channels for merging

Output: Merged channel in folder *P02_Calculated Max Intensity Channel*

Generate Hue, Saturation and Brightness/Value channels

Script name: *P04_Generate HSB channels*

Aim: To generate the Hue, Saturation and Brightness/Value channel from the 3 channels for surfacing downstream.

Description: If the user does not need to carry out noise removal, the cells can be directly segmented using the Hue, Saturation and Brightness/Value channels. The Hue, Saturation and Brightness/Value channels will merge information from the 3 channels.

Parameter	Description
inChNumberA	Input channel number for Channel A which will be treated as Red in RGB
inChNumberB	Input channel number for Channel B which will be treated as Green in RGB
inChNumberC	Input channel number for Channel C which will be treated as Blue in RGB
sfA sfB sfC	Scale factor for channels A, B and C
minA maxA minB maxB minC maxC	Minimum and maximum values for channels A, B and C
outChNumberH	Channel number for Hue channel
outChNumberB	Channel number for Brightness/Value channel
outChNumberS	Channel number for Saturation channel

Input: Merged/ original 3 channels

Output: Hue and Brightness channels in folder *P04_ HSB*

Surface creation using the Brightness/ Maximum Intensity channel in Imaris.

Aim: To surface the cells using Imaris

Description: An important part of the HSB surface creation workflow is cell surfacing in order to extract out statistics. This function is performed in Imaris using the surfacing function.

Input: TIF files of the original channels and the transformed hue, saturation and brightness channels or maximum intensity channel.

Output: Cell surfaces in Imaris.

Note: For better cell segmentation, it is recommended that at least one of the channels should stain the cytoplasm.

Spectral Compensation

Script name: S01_Channel unmixing

Aim: To carry out spectral compensation for channel spillover.

Description: If channel spillover exists, it is necessary to carry out spectral compensation to generate corrected channels.

Parameter	Description
inChNumbers	Channel numbers of input channels. Provide a list
cMatrix	Spillover coefficients. Key in NxN spillover coefficients in same way as in FlowJo - row headers are Dyes, column headers are Detectors.

Input: All channels acquired during imaging, channels of single stains.

Output: Corrected channels in *S01_Unmixed*

Extract statistics from Imaris

Aim: To extract out the statistics from the surfaced cells.

Description: After cell surfacing, the statistics can now be extracted out from the relevant channels

Input: Cell surfaces n relevant channels for statistics extraction (eg. original imaging channels and normalized channels)

Output: CSV files of cell surface statistics.

Aggregate statistics into single CSV file

Script name: *P05_Aggregate Imaris statistics output*

Aim: To compile the different Imaris statistics into one CSV file for visualization in FlowJo.

Description: The Imaris statistics for the surfaces are saved as individual files, and it is necessary to aggregate them into a single excel file for import into FlowJo.

Input: Folder containing the cell surface statistics

Output: An excel file containing the cell surface statistics in *P05_Aggregated statistics*

Parameter	Description
prefix	Prefix to be attached to the output filename

Analyse and Gate surfaced populations

Aim: To analyse and gate the surfaced populations in FlowJo or any other program suitable for flow-cytometry-like analysis (eg. XiT) that accepts CSV files as input.

Description: The aggregated Imaris statistics can now be analysed in a histocytometry manner using FlowJo.

Input: CSV file containing the aggregated cell surface statistics

Output: CSV file containing the cell surface statistics

Visualise gated populations in Imaris (Imaris 8.4)

(Note: to use this function the user needs the license for ImarisXT and the XT script currently does not work for Imaris 9.0 and above)

Aim: To identify the isolated surfaces based on their IDs within the Imaris image for verifying the accuracy of the surface and for visualization.

Description: After analysis of the cells on FlowJo, the surface IDs of the populations that have been gated out can be visualized in the original Imaris file for quality checks.

Input: Surface IDs of the gated population; Imaris file with surfaces

Output: Visualisation of the gated population in the Imaris file.

Using the command line interface

Other than the graphical user interface, the HSB surface creation scripts can also be run using the command line interface, to allow for integration of the scripts into other workflows and pipelines. The command line interface exposes all the parameters configurable and this will allow the user to have full control over the functionalities provided by the script. However, this method of running the scripts is only recommended for advanced users familiar with programming.