Metamorph Files Folder To Hyperstacks
ImageJ macro, Marcel Boeglin Jul 2018 – Jan 2020
User guide

Features and introduction:

MetamorphFilesFolderToHyperstacks_ ImageJ macro
Author : Marcel Boeglin, July 2018 - January 2020
e-mail: boeglin@igbmc.fr

¤ Opens Metamorph multi-position time-series z-stacks of up to 7 channels from input folder
   and saves them as hyperstacks to output folder.

¤ Only **TIFF** and **STK** files are processed.

¤ Image files for which no series name can be found are skipped.

¤ Single file series (including output files of this macro the user manually copied to input folder) are
   skipped because output files would be identical to input files,

¤ Series having same channel sequence are grouped in channel groups. All series from a given
   channel group are processed using same parameters.

¤ Channel colors are determined automatically from filenames by use of red, green, blue, cyan,
   magenta, yellow and gray determinants but can be changed by the the user. Only these 7 colors
   are possible.
   For instance, the red determinants set is: **{"543", "555", "561", "594", "CY3", "Y3",
   "DsRed", "mCherry", "N21", "RFP", "TX2"}**.
   Channels containing a red, green, blue or gray determinant in their names are assumed to be
   respectively red, green, blue or gray.
   In future versions, it's planned to get channel colors from metadata in case Metamorph has been
   configured to not add the illumination settings to the filenames (which in that case contain just
   _w1, _w2, ..., _wN).

   **If none of the color determinants works with your illumination settings, just add
   the missing determinant(s) to the appropriate set in the macro (advanced users).**

¤ Dual camera channels handling:
   It's Assumed channels are saved in separate images. As in single camera acquisitions, colors are
   derived from filenames.
   Dual camera filenames are assumed to be of the type:
   seriesName_w1Lambda1 Lambda2...
   seriesName_w2Lambda1 Lambda2...
   One could expect that w1 corresponds to lambda1 and w2 to lambda2, but w1 may correspond to
   lambda2 and w2 to lambda1, depending on the configuration of Metamorph.

   The dual channel separator (a space character in the example) may be different, for instance a "-",
   depending on how dual camera illumination settings have been named in Metamorph.

Dual channel order and separator are managed by the macro but are assumed to be the same for all series in a given folder.
If the dual channel colors attribution fails, arbitrary (probably unwanted) colors are assigned to the channels. In such a case, the user can choose the color of each channel for each channel group.

¤ Output images are X, Y, Z and T calibrated if calibration data are available. If not, a unique xy-z-t calibration, applied to **all series**, can be set by the user to avoid manual calibration of output files.

¤ Allows control of z-range, position-range (s1, s2, s3, ...) and time-range of processed files.

¤ Does optional resizing or cropping of input files.

¤ Does optional maximum, average, minimum, median or sum z-projection of input files and color balance of output files.

¤ In case of heterogeneous z-dimensions between channels and 'Do z-projection' is unchecked, single-section channels are transformed into z-stacks having same depth as channels acquired as z-stacks by duplicating the single-section.

¤ Series handling dialog is not displayed if larger than screen (happens if the folder contains more series than can be added as checkboxes in a dialog). In this case all series that have passed the including and excluding file-filters are processed.

Installation and short instructions:

- copy "MetamorphFilesFolderToHyperstacks_43u11.ijm" or later to the "macros" folder of your ImageJ or Fiji installation.
- run "Plugins > Macros > Install...": choose the file of the macro to be installed (for now MetamorphFilesFolderToHyperstacks_43u11.ijm).

In the "Plugins > Macros" menue, you will have 4 new commands (requires 43u11):
- **Metamorph Files to Hyperstacks**
  does nothing
- **EasyMode_Zproj [F1]**
  processes all series from input folder, z-projected output files;
- **EasyMode_noZproj [F2]**
  processes all series from input folder, non z-projected output files;
- **Full_Mode [F3]**
 gives access to all features (up to 10 dialogs).

Easy modes just need you choose the input and output folders and the macro will run with default parameters.

Full mode takes less than a minute to answer the dialogs but may save tens of hours spent to fight with your computer to open, merge and save hundreds of images. Moreover, the region extraction procedure allows processing of images that you never would be able to process directly with ImageJ (when the size exceeds the ImageJ's capacity (about 2 G pixels for the XYCZT product of image's all dimensions)).
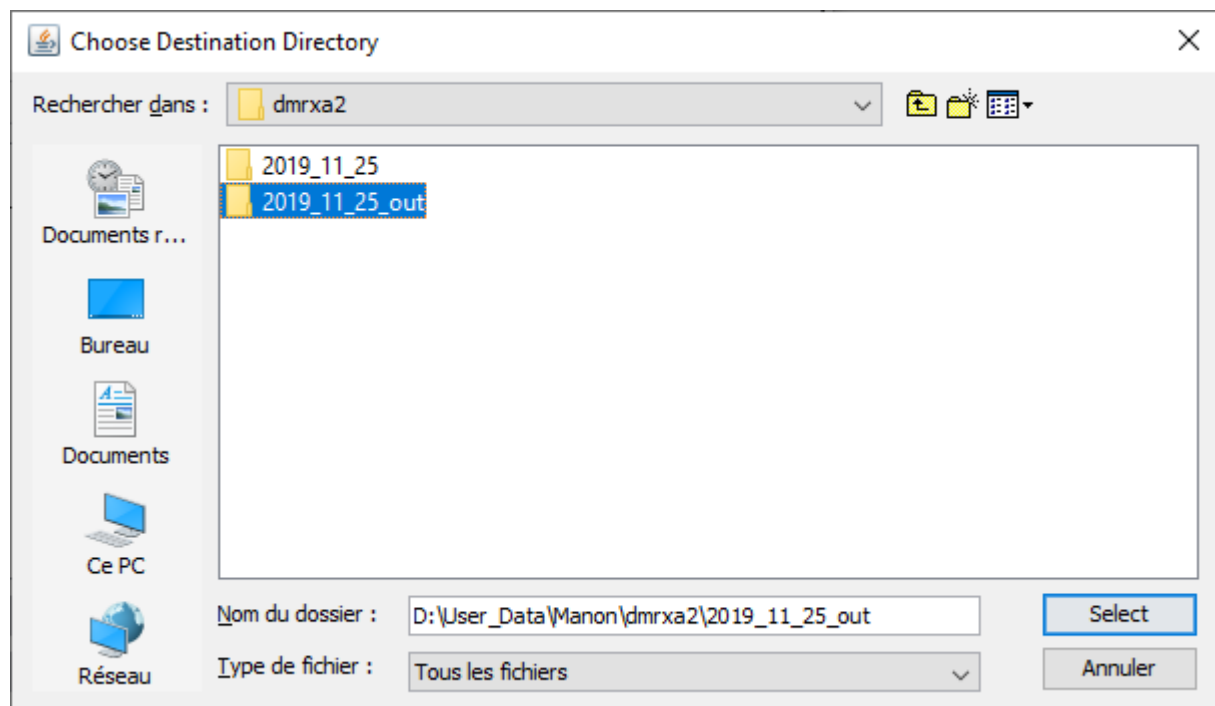
Full mode is described in details in next pages.

Detailed Procedure and description of dialog boxes:

1. Choose input folder:
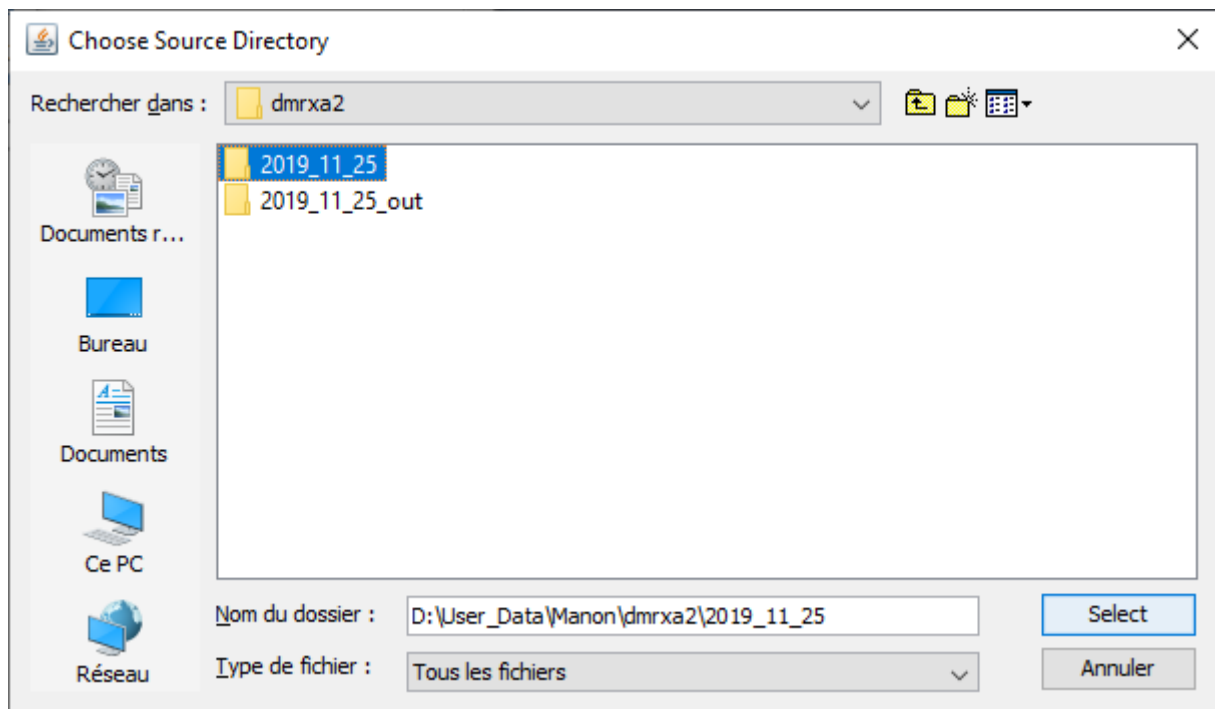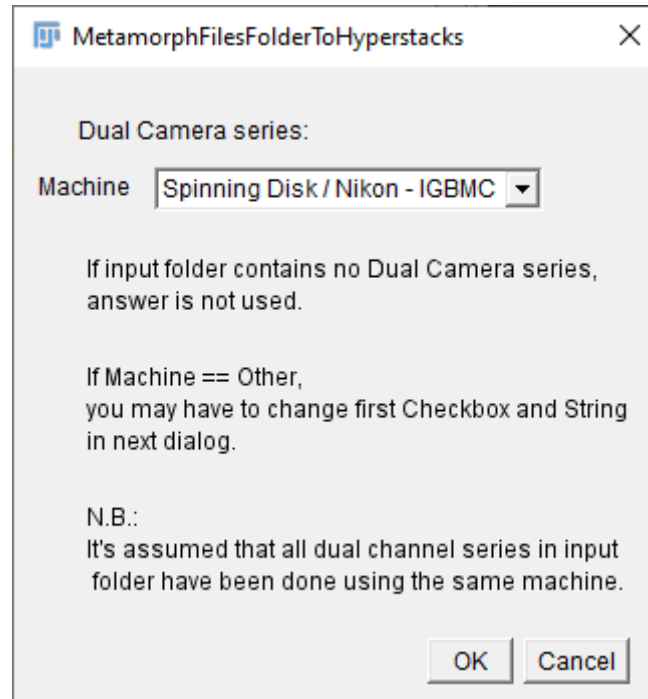

2. Choose output folder: shouldn't (and can't) be identical to input folder
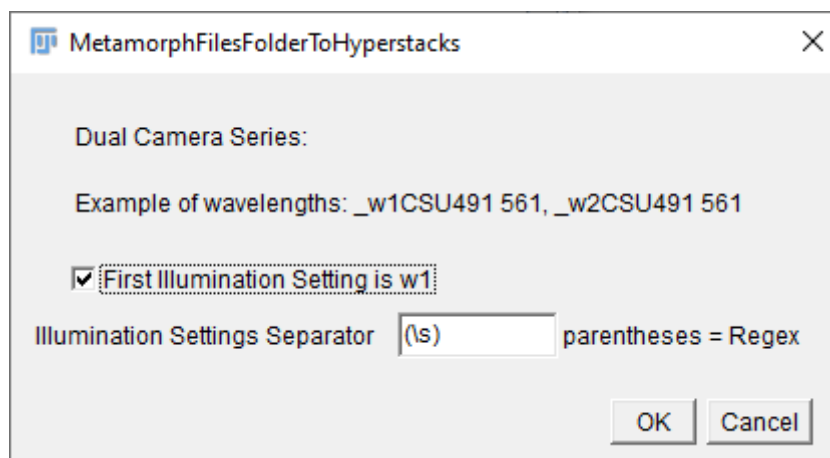
3. Dual Camera series, dialog 1:

- **If you didn't do dual-camera acquisitions, skip this dialog by clicking the OK button or typing "Enter" without changing anything;**
- **Else select the machine used for dual-camera acquisitions.**

3. Dual Camera series, dialog 2:

- **Defaults in this dialog are set automatically if you have selected a spinning disk from IGBMC's Imaging Center in dialog1,**

so as you can validate it by pressing the OK button or the "Enter" key.

4. Main Dialog:

> **File filters:** limit processing to filenames containing the including string but not the excluding string. Allows to process only a part of input folder ; reduces also delay to display next dialogs and the number of components they contain.

> **Missing files infos:** if some images have been lost, the macro doesn't stop but replaces any missing image with a black one where is written the path of the expected file. The path can be written in the output image's overlay (recommanded) or grabbed in its pixels.

> **Channel-dependent binning:** the macro is designed to manage such situations but this feature has not been tested.

> **Pixel-size from Metadata correction:** to be used only if Metamorph has been configured for pixel-size calibration. If the used objective is the declared one, let as is.

> **If no xy_z calibration data found** in metadata, a unique calibration can be used **for all series** (avoids having to do it manually afterwards by opening and saving again the output images).

> **z-Projection BY DEFAULT FOR ALL SERIES:** if checked, channel groups handling dialog defaults will be do z-projection for all channel groups, but you can choose to do z-projections or not for each channel-group individually.

> **Display Data Reduction Dialog:** select if you want to resize or crop images at import or if you want to limit processing to given positions, z or time ranges; you will get an extra dialog to indicate what you want to do.

> **If no frame interval or time unit is found** in metadata : you can set a unique time-calibration that will be used **for all output series** (avoids to do it manually afterwards).

> **Create an output file for each time point:** in case of huge time-lapse, you can merge only the channels for each time point to limit the size of output files. The result will be a multi-channel image for each time point.

> **Ignore single time point channels:** avoids processing images that are not parts of a time-lapse.

5. Data Reduction:

> **Resize at import:** use for instance 0.5 or less if output images are too big for installed memory or for ImageJ's capabilities.

> **Crop at import:** does extraction of rectangular regions from input images in multi-channel and / or time series. For more details, refer to **"Extracting regions from input images"**. Crop at import also saves memory.

> **Z-series:** if firstSlice > lastSlice, output stack will be reversed.
¤ Process range around median slice:
useful if series have multiple stack sizes and you want to limit output images stack sizes.

> **Multi-positions series:** to process for instance only positions s9, s10 and s11, set "firstPosition" to 9 and "lastPosition" to 11.

> **Time-series:**
to process for instance from t12 to end of timelapse, enter:
"firstTimePoint": 12
"lastTimePoint": -1
¤ Process range from t1:
if range is for instance 25 % and input folder contains a timelapse with 400 time points, only the 100 first time points will be processed. If the same folder contains another time-lapse of 100 time points, only those from 1 to 25 will be processed.

6. Process series:

This dialog lists all series found in input folder after file filters have been applied.
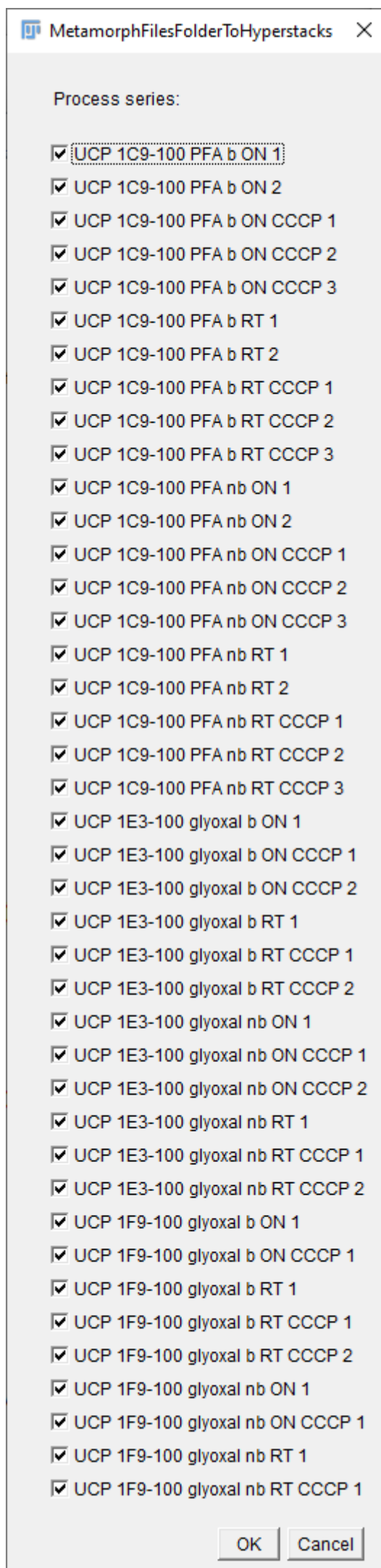
To skip a series, just uncheck it.

The list may be very long if not reduced by use of file filters in main dialog.
Remember that file filters allow to reduce processed files to a few series.

For instance, if the including filter (the upper one) in the Main Dialog is equal to a series name, only this series will be processed and the "Process series" dialog will even not be displayed.

If you chose ¤ **Crop at import** and don't want to process all series in the **"ImagesWithRois"** folder, just uncheck the series you don't want to process.
The chapter "Extract regions from input images" shows another way to limit the processing to given series.

MetamorphFilesFolderToHyperstacks ✕

Process series:

- ☑ UCP 1C9-100 PFA b ON 1
- ☑ UCP 1C9-100 PFA b ON 2
- ☑ UCP 1C9-100 PFA b ON CCCP 1
- ☑ UCP 1C9-100 PFA b ON CCCP 2
- ☑ UCP 1C9-100 PFA b ON CCCP 3
- ☑ UCP 1C9-100 PFA b RT 1
- ☑ UCP 1C9-100 PFA b RT 2
- ☑ UCP 1C9-100 PFA b RT CCCP 1
- ☑ UCP 1C9-100 PFA b RT CCCP 2
- ☑ UCP 1C9-100 PFA b RT CCCP 3
- ☑ UCP 1C9-100 PFA nb ON 1
- ☑ UCP 1C9-100 PFA nb ON 2
- ☑ UCP 1C9-100 PFA nb ON CCCP 1
- ☑ UCP 1C9-100 PFA nb ON CCCP 2
- ☑ UCP 1C9-100 PFA nb ON CCCP 3
- ☑ UCP 1C9-100 PFA nb RT 1
- ☑ UCP 1C9-100 PFA nb RT 2
- ☑ UCP 1C9-100 PFA nb RT CCCP 1
- ☑ UCP 1C9-100 PFA nb RT CCCP 2
- ☑ UCP 1C9-100 PFA nb RT CCCP 3
- ☑ UCP 1E3-100 glyoxal b ON 1
- ☑ UCP 1E3-100 glyoxal b ON CCCP 1
- ☑ UCP 1E3-100 glyoxal b ON CCCP 2
- ☑ UCP 1E3-100 glyoxal b RT 1
- ☑ UCP 1E3-100 glyoxal b RT CCCP 1
- ☑ UCP 1E3-100 glyoxal b RT CCCP 2
- ☑ UCP 1E3-100 glyoxal nb ON 1
- ☑ UCP 1E3-100 glyoxal nb ON CCCP 1
- ☑ UCP 1E3-100 glyoxal nb ON CCCP 2
- ☑ UCP 1E3-100 glyoxal nb RT 1
- ☑ UCP 1E3-100 glyoxal nb RT CCCP 1
- ☑ UCP 1E3-100 glyoxal nb RT CCCP 2
- ☑ UCP 1F9-100 glyoxal b ON 1
- ☑ UCP 1F9-100 glyoxal b ON CCCP 1
- ☑ UCP 1F9-100 glyoxal b RT 1
- ☑ UCP 1F9-100 glyoxal b RT CCCP 1
- ☑ UCP 1F9-100 glyoxal b RT CCCP 2
- ☑ UCP 1F9-100 glyoxal nb ON 1
- ☑ UCP 1F9-100 glyoxal nb ON CCCP 1
- ☑ UCP 1F9-100 glyoxal nb RT 1
- ☑ UCP 1F9-100 glyoxal nb RT CCCP 1

OK | Cancel

7. Channels handling:

The macro groups the series in sets having identical channel sequences where the order of channels is taken in account to avoid mixing files from different series.
For instance, sequence (w1Dapi, w2CY3) is different from (w1CY3, W2Dapi).

If the input folder contains multiple channel sequences, the "Channels handling" dialog will contain a components group for each of them, allowing to control each channel sequence individually.

Channel group n: contains: This combo box is for information only; it lists the series names concerned by nth Channel group.

All choices done here will be applied to all series from current channel group.

¤ **Do Z-projections:** this checkbox allows you to do or not z-projection for this channel group.

Following is just an example, the situation may be different for you:

¤ **w1Dapi** ¤ **w2CY3** ¤ **w3GFP:** unchecked channels will not be added to output images.

w1DAPI Max Intensity**:**
**w2CY3** Max Intensity**:**
w3GFP Max Intensity**:**
These combo boxes are for choice of projection type of first, second, third, … channel if
¤ **Do Z-projections is checked.**
The available projection types are: Average Intensity, Max Intensity, Min Intensity, Median and Sum Slices.
If any channel is projected using Sum Slices, output images type is 32-bit for all series in the channel group.

8. Channels colors and saturations:
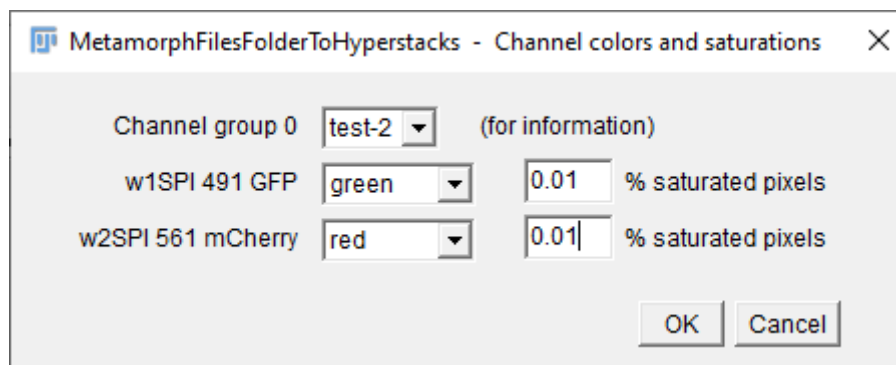
The macro determines automatically the default colors, but you can change output colors if you prefer different ones.

Caution: the macro uses ImageJ's color merge command. It fails if two channels have the same color. In this case, the macro tries to replace one of the duplicate colors by a different one, but you have no control on the procedure. If it fails due to an unknown bug, the execution will stop.

Channel group n: this combo box is for information only; it lists the series names concerned by nth Channel group.

The choices for a given channel group stand for all series it contains.

Following is just an example:

w1SPI 491 GFP green 0.01 % saturated pixels:
the macro proposes **"green"** because the channel name contains **"GFP"**.
w2SPI 561 mCherry red 0.01 % saturated pixels:
the macro proposes **"red"** because the channel name contains **"mCherry"**.

Saturated pixels %: the macro uses these values to set the maximum displayed values of each channel in output images, calculated in the median slice for z-stacks and median time point for time-lapse images.

Note: the macro never uses the "Apply" function from the Image > Adjust > Color Balance… or Brightness/Contrast… window.
This means that even if the output images appear saturated, their pixel values are **not** changed and are suitable for intensity measurements.
A simple click on "Reset" in the "B&C" or "Color" window will desaturate the active channel.

9. Choose positions to be processed:
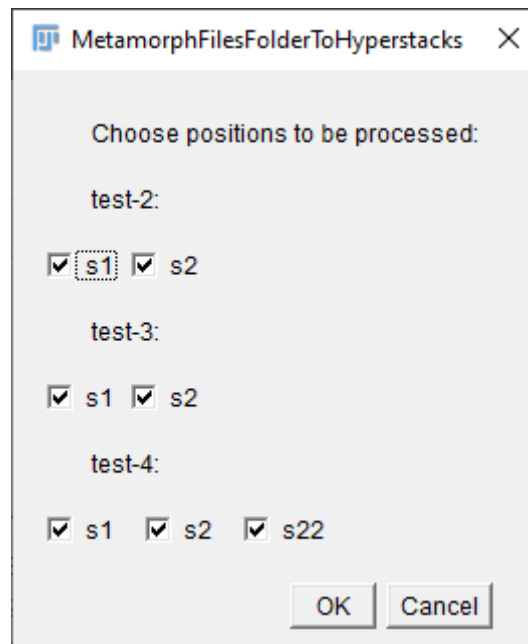
For each selected series:

the dialog displays a checkbox for each detected position in given series.
If you have filtered the positions, only those that have passed the filter will appear here.

To skip a position, just uncheck it.

This applies also if you chose ¤ **Crop at import**. If you don't want to process some positions of some of the series contained in the **"ImagesWithRois"** folder, just uncheck them in this dialog.

Note: Dialogs with component groups as several checkboxes on a same row cannot be validated by pressing "Enter", you must click on "OK".

10. Extract regions from input images:

 - **Create folder "ImagesWithRois" in input folder**

- **For each series or position from which you want to extract
  regions of interest:**
    - Open an input image (any channel or time-point)
    - If it's a stack, do a z-projection (Average, Max, Min etc.)
    - Draw a rectangle for each region to be processed
    - Add the rectangle to the overlay (press b or Ctrl b)
    - Save the image with rois to folder "ImagesWithRois". Don't rename it, if its name doesn't
      contain the series name and the position name, it will not be processed.
    - Close the projection-image.

 - **Launch MetamorphFilesFolderToHyperstacks:**
    - In the "Main Dialog", check "Display Data Reduction Dialog"
    - In the "Data Reduction" dialog check "¤ **Crop at import**"
   This will result in limiting process to:
    - series and positions for which an image is found in the "ImagesWithRois" folder.
    - regions drawn in these images.

¤ Output files:
  For each region drawn in images with rois:
   - Cropped image of merged channels and time-points of corresponding series.
   - The output image-names contain the coordinates of the upper-left edge of the extracted region.

¤ Tips:
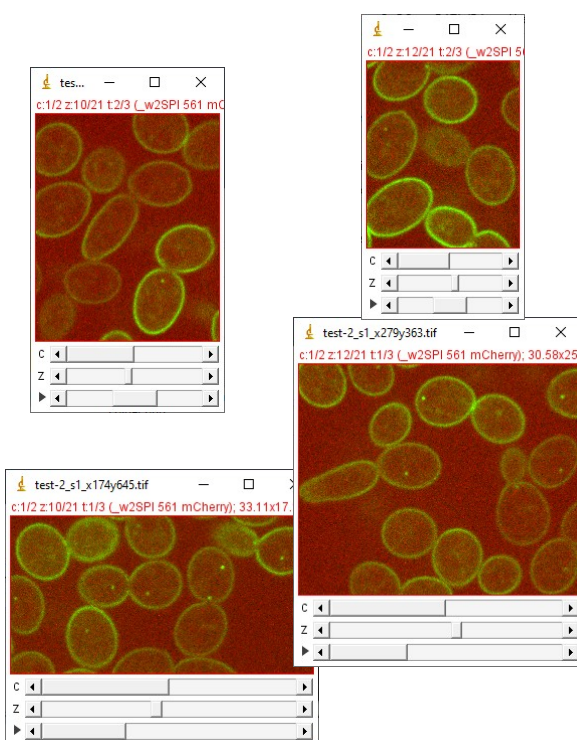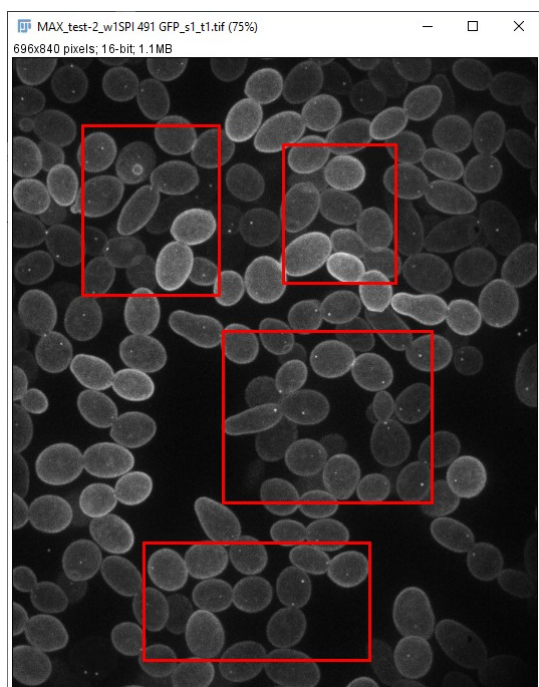 **In case of a timelapse and the sample moved during acquisition:**
  - Open an input image of the first timepoint to be processed, do z-projection if it's a stack.
  - Repeat this for the last time point to be processed.
  - Synchronize the two projections
  - Draw the rois in the first time point projection:
    during the drawing process, the roi will be visible also in the last timepoint projection, so that
    you can make it large enough to contain the sample at the begin and the end of the time-lapse.
  - Save the first timepoint projection to the "ImagesWithRois" folder.
  - If you want to keep the last timepoint projection, you can save it also in the "ImagesWithRois"
    folder, it will not be taken in account if you check the   ¤ Crop at input   checkbox.

  To find easier which output image corresponds to a given region:
  - Open the image with rois
  - launch Image > Overlay > List Elements
  - In the Overlay Elements table, double-click on a row activates corresponding roi;
    The X and Y coordinates of the roi can be compared with those of the cropped output image
    name which ends for instance with "_x92y90.tif", where x and y are the coordinates of the
    upper-left corner of the extracted region.

Examples of images with rois for regions extraction and output images:

test-2_s1

test-2_s2

Examples of images with rois for regions extraction and output images (continued):

test-3_s1

test-3_s2

Content of input folder in the regions-extraction example:

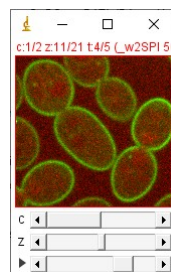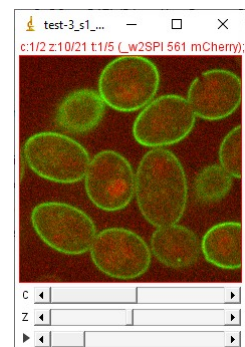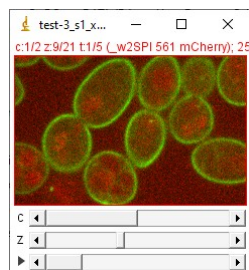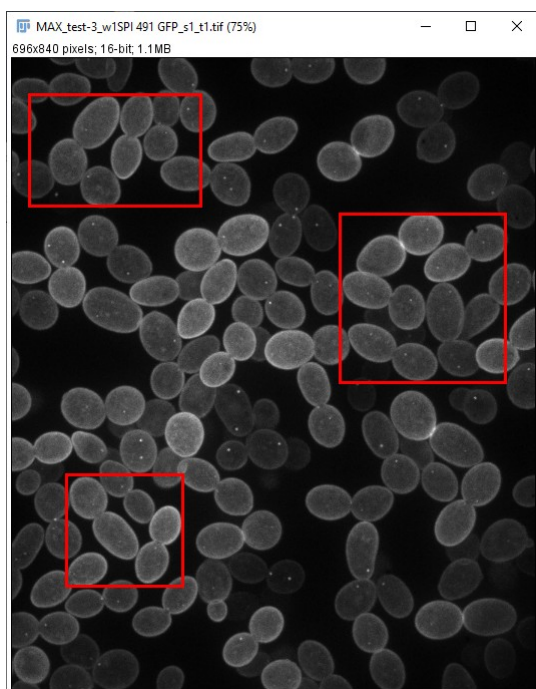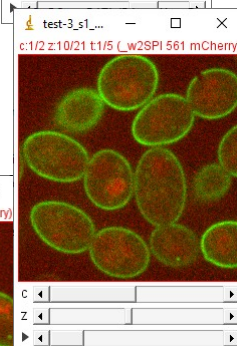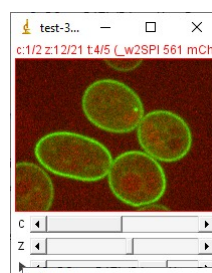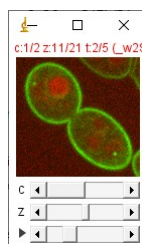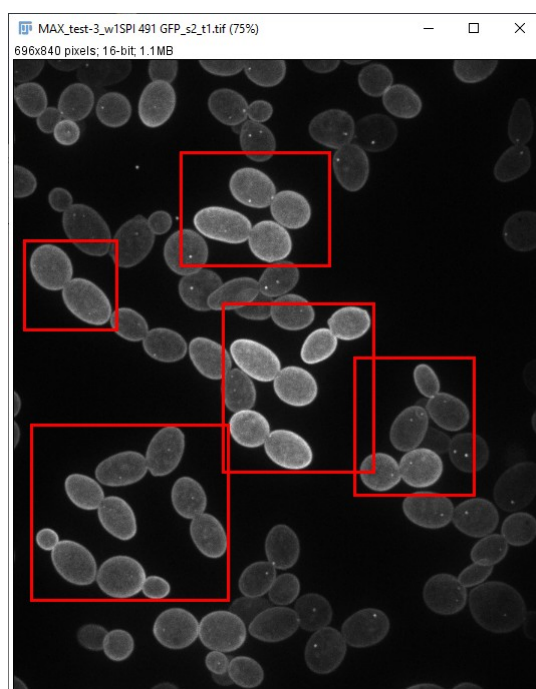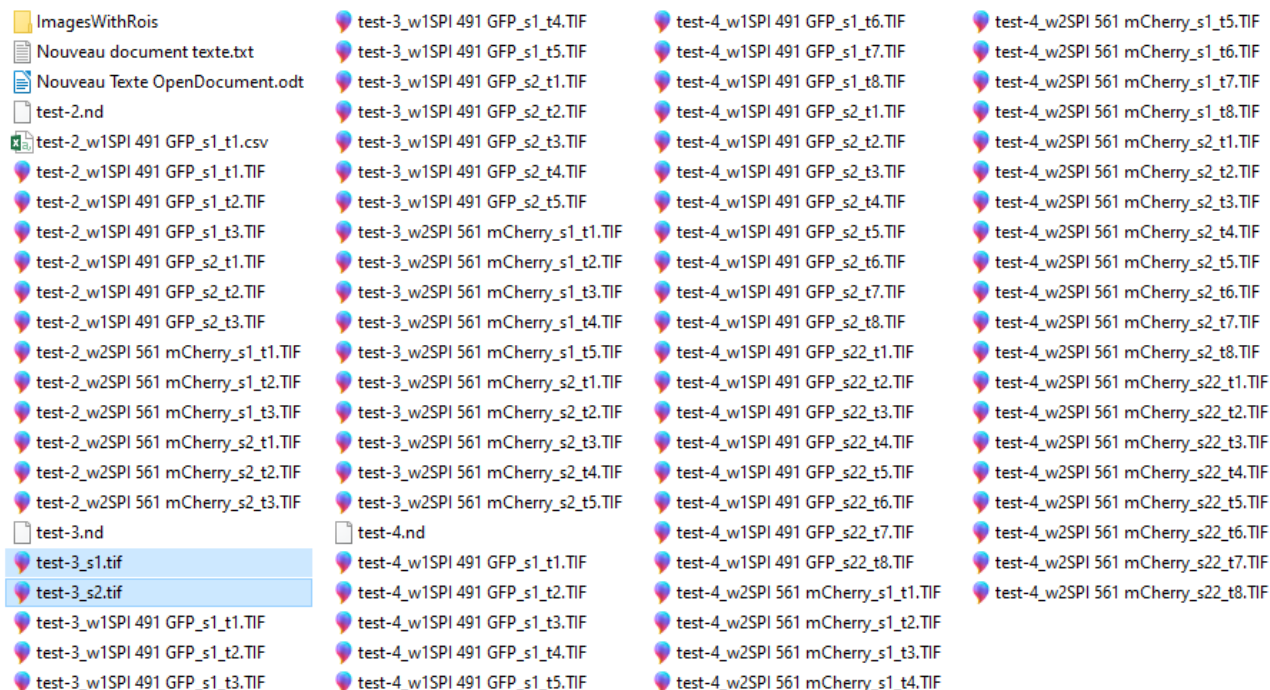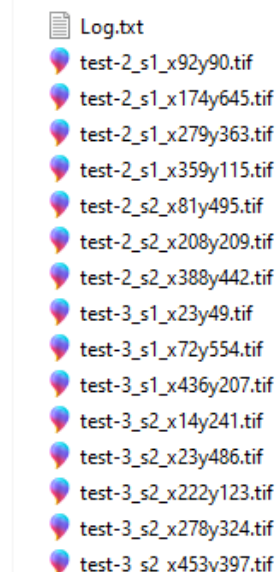|			
ImagesWithRois	test-3_w1SPI 491 GFP_s1_t4.TIF	test-4_w1SPI 491 GFP_s1_t6.TIF	test-4_w2SPI 561 mCherry_s1_t5.TIF
Nouveau document texte.txt	test-3_w1SPI 491 GFP_s1_t5.TIF	test-4_w1SPI 491 GFP_s1_t7.TIF	test-4_w2SPI 561 mCherry_s1_t6.TIF
Nouveau Texte OpenDocument.odt	test-3_w1SPI 491 GFP_s2_t1.TIF	test-4_w1SPI 491 GFP_s1_t8.TIF	test-4_w2SPI 561 mCherry_s1_t7.TIF
test-2.nd	test-3_w1SPI 491 GFP_s2_t2.TIF	test-4_w1SPI 491 GFP_s2_t1.TIF	test-4_w2SPI 561 mCherry_s1_t8.TIF
test-2_w1SPI 491 GFP_s1_t1.csv	test-3_w1SPI 491 GFP_s2_t3.TIF	test-4_w1SPI 491 GFP_s2_t2.TIF	test-4_w2SPI 561 mCherry_s2_t1.TIF
test-2_w1SPI 491 GFP_s1_t1.TIF	test-3_w1SPI 491 GFP_s2_t4.TIF	test-4_w1SPI 491 GFP_s2_t3.TIF	test-4_w2SPI 561 mCherry_s2_t2.TIF
test-2_w1SPI 491 GFP_s1_t2.TIF	test-3_w1SPI 491 GFP_s2_t5.TIF	test-4_w1SPI 491 GFP_s2_t4.TIF	test-4_w2SPI 561 mCherry_s2_t3.TIF
test-2_w1SPI 491 GFP_s1_t3.TIF	test-3_w2SPI 561 mCherry_s1_t1.TIF	test-4_w1SPI 491 GFP_s2_t5.TIF	test-4_w2SPI 561 mCherry_s2_t4.TIF
test-2_w1SPI 491 GFP_s2_t1.TIF	test-3_w2SPI 561 mCherry_s1_t2.TIF	test-4_w1SPI 491 GFP_s2_t6.TIF	test-4_w2SPI 561 mCherry_s2_t5.TIF
test-2_w1SPI 491 GFP_s2_t2.TIF	test-3_w2SPI 561 mCherry_s1_t3.TIF	test-4_w1SPI 491 GFP_s2_t7.TIF	test-4_w2SPI 561 mCherry_s2_t6.TIF
test-2_w1SPI 491 GFP_s2_t3.TIF	test-3_w2SPI 561 mCherry_s1_t4.TIF	test-4_w1SPI 491 GFP_s2_t8.TIF	test-4_w2SPI 561 mCherry_s2_t7.TIF
test-2_w2SPI 561 mCherry_s1_t1.TIF	test-3_w2SPI 561 mCherry_s1_t5.TIF	test-4_w1SPI 491 GFP_s22_t1.TIF	test-4_w2SPI 561 mCherry_s2_t8.TIF
test-2_w2SPI 561 mCherry_s1_t2.TIF	test-3_w2SPI 561 mCherry_s2_t1.TIF	test-4_w1SPI 491 GFP_s22_t2.TIF	test-4_w2SPI 561 mCherry_s22_t1.TIF
test-2_w2SPI 561 mCherry_s1_t3.TIF	test-3_w2SPI 561 mCherry_s2_t2.TIF	test-4_w1SPI 491 GFP_s22_t3.TIF	test-4_w2SPI 561 mCherry_s22_t2.TIF
test-2_w2SPI 561 mCherry_s2_t1.TIF	test-3_w2SPI 561 mCherry_s2_t3.TIF	test-4_w1SPI 491 GFP_s22_t4.TIF	test-4_w2SPI 561 mCherry_s22_t3.TIF
test-2_w2SPI 561 mCherry_s2_t2.TIF	test-3_w2SPI 561 mCherry_s2_t4.TIF	test-4_w1SPI 491 GFP_s22_t5.TIF	test-4_w2SPI 561 mCherry_s22_t4.TIF
test-2_w2SPI 561 mCherry_s2_t3.TIF	test-3_w2SPI 561 mCherry_s2_t5.TIF	test-4_w1SPI 491 GFP_s22_t6.TIF	test-4_w2SPI 561 mCherry_s22_t5.TIF
test-3.nd	test-4.nd	test-4_w1SPI 491 GFP_s22_t7.TIF	test-4_w2SPI 561 mCherry_s22_t6.TIF
test-3_s1.tif	test-4_w1SPI 491 GFP_s1_t1.TIF	test-4_w1SPI 491 GFP_s22_t8.TIF	test-4_w2SPI 561 mCherry_s22_t7.TIF
test-3_s2.tif	test-4_w1SPI 491 GFP_s1_t2.TIF	test-4_w2SPI 561 mCherry_s1_t1.TIF	test-4_w2SPI 561 mCherry_s22_t8.TIF
test-3_w1SPI 491 GFP_s1_t1.TIF	test-4_w1SPI 491 GFP_s1_t3.TIF	test-4_w2SPI 561 mCherry_s1_t2.TIF	
test-3_w1SPI 491 GFP_s1_t2.TIF	test-4_w1SPI 491 GFP_s1_t4.TIF	test-4_w2SPI 561 mCherry_s1_t3.TIF	
test-3_w1SPI 491 GFP_s1_t3.TIF	test-4_w1SPI 491 GFP_s1_t5.TIF	test-4_w2SPI 561 mCherry_s1_t4.TIF	

In the example, macro may works properly despite input folder contains output images. But it's recommanded to avoid such situations which may cause an unexpected issue.

Content of Input folder / ImagesWithRois:

MAX_test-2_w1SPI 491 GFP_s1_t1.tif
MAX_test-2_w1SPI 491 GFP_s2_t1.tif
MAX_test-3_w1SPI 491 GFP_s1_t1.tif
MAX_test-3_w1SPI 491 GFP_s2_t1.tif

Content of output folder:

Log.txt
test-2_s1_x92y90.tif
test-2_s1_x174y645.tif
test-2_s1_x279y363.tif
test-2_s1_x359y115.tif
test-2_s2_x81y495.tif
test-2_s2_x208y209.tif
test-2_s2_x388y442.tif
test-3_s1_x23y49.tif
test-3_s1_x72y554.tif
test-3_s1_x436y207.tif
test-3_s2_x14y241.tif
test-3_s2_x23y486.tif
test-3_s2_x222y123.tif
test-3_s2_x278y324.tif
test-3_s2_x453y397.tif

Tip in case of regions extraction (current example): you can force the macro to limit processing to given series/positions by removing corresponding images with rois from ImagesWithRois folder. This can be done by creating a subfloder "ImagesWithRois/Done" and moving them to that folder.

End notes:

The MetamorphFilesFolderToHyperstacks_ macro doesn't use the .nd file. As a consequence, it doesn't rename the positions "s1", "s2", "s3", … according to the names you gave them in the Multiposition tab in Metamorph. Do this would complicate a litle bit more the macro which is already quite enormous (more than 4000 program lines, about 200 variables, more than 90 functions.

Renaming the output files positions according to their names in the multiposition window of Metamorph will be the object of a future macro if the users demand is real.

Due to the fact that the ImageJ macro language is not object-oriented, it's difficult and tricky to store a lot of intermediate images. For this reason, if you want to extract N regions from a series or position, each necessary input file will be opened N times, resulting in a loss of time. But since the macro works in batch mode, it will just increase the computers process time but not your working time.

A more efficient implementation of the tasks done by the macro would need to rewrite it Java which is much more efficient and versatile than the ImageJ macro language. Java is strongly object-oriented and allows to store intermediate images and other objects in multidimensional arrays or collections and to recall them when necessary, simplifying drastically the code.

Translating the macro in Java would take as long as needed to design the macro because it would require a complete refactoring.