Image processing steps of the LongTracker image pre-processing app

Below is a description of the image processing in the LongTracker image pre-processing app. When sample MATLAB code is provided in the descriptions, function names in bold face type can be looked up in the MATLAB Help Center (www.mathworks.com/help/) for a more thorough description. **MATLAB file: Lgtker_ImageDataPreProcessing.mlapp**

- **STEP 1. CONVERT IMAGES TO STANDARD BIT DEPTH**: When raw data images are read in from their respective proprietary file format it is beneficial to convert them all to a standard 8-bit format. After reading in the raw image data, the 8-bit standardization starts with the image pixel values being converted from integer to a double precision format then normalizing them to a range of 0-1 by dividing with the raw data bit depth. The normalized pixels are then multiplied by 255 and converted to an 8-bit integer format. This standardization process also means all exported image files will be in an 8-bit image format. If images were recorded in an 8-bit format during data collection this step will be redundant. If the image data shows signs of irregular artifacts after being loaded in try changing the raw data bit depth. This conversion process is automatically applied when an image data file is selected and does not require action by the user other than changing the raw data bit depth if necessary.

  sample MATLAB code: Image8bit = **uint8**(255*(**double**(Raw2Dimage)/BitDepth))

- **STEP 2. APPLY IMAGE PROCESSING FUNCTIONS:** After standardizing the image bit depth, 2-D and 3-D image processing functions can be applied. The image data channel assigned to the RED channel has 4 functions available and the GREEN channel has 5. Image processing functions are applied in the order they are numbered, and functions are only applied if the box next to the function is checked. Here is a description of the image processing functions:

  1. **3-D Gaussian smoothing:** 3-D Gaussian smoothing is a standard low-pass filter applied at each time point by the convolution of a 3-D Gaussian kernel and image volume (z-stack). The xy-sigma parameter translates to the 2-D Standard Deviation (SD) of the Gaussian kernel in the X-Y plane, and the z-sigma parameter is the SD of the kernel along the z-axis. The actual pixel width/height/depth of the filter can be calculated with the formula: pixels = 2*ceil(2*sigma)+1. If the pixel size and z-step of the data set are dp and dz respectively, then it is recommended that the z-sigma parameter be set relative to the xy-sigma parameter such that: **z-sigma = xy-sigma*(dp/dz)**.
     In-depth description: *https://en.wikipedia.org/wiki/Gaussian_blur*

     sample MATLAB code: FilteredVolume = **imgaussfilt3**(ImageVolume, [xy-sigma, xy-sigma, z-sigma])

  2. **2-D Gaussian smoothing:** 2-D Gaussian smoothing is a standard low-pass filter applied at each time point by the convolution of a 2-D Gaussian kernel and z-slice image. The xy-sigma parameter translates to the 2-D width and height of the filter in the X-Y plane. The actual pixel width/height of the filter can be calculated with the formula: pixels = 2*ceil(2*sigma)+1.
     In-depth description: *https://en.wikipedia.org/wiki/Gaussian_blur*

     sample MATLAB code: FilteredImage = **imgaussfilt**(Image, xy-sigma)

  3. **Contrast-Limited Adaptive Histogram Equalization:** A regional image histogram equalization (regional contrast enhancement) is applied to each z-slice.
     In-depth description: *https://en.wikipedia.org/wiki/Adaptive_histogram_equalization*
     Reference: *Zuiderveld, Karel. "Contrast Limited Adaptive Histograph Equalization."* Graphic Gems IV*. San Diego: Academic Press Professional, 1994. 474–485.*

     sample MATLAB code: EnhancedImage = **adapthisteq**(Image, 'NumTiles', [16,16], 'Distribution', 'rayleigh')

  4. **Image Saturation:** An overall contrast enhancement is applied to each z-slice by setting a percentage, p, of the lowest and highest valued pixels to 0 and 255 respectively then all other pixels linearly adjusted to the new dynamic range.

     sample MATLAB code: SaturatedImage = **imadjust**(Image, **stretchlim**(Image, [p/100, (100-p)/100]))

  5. **Subtract Red image from Green image:** In order to improve cell nuclei segmentation in the green channel, the entire post-processed 3-D volume of the red channel (the red channel contains the cell membrane image data) is subtracted from the post-processed green channel at each time point.

- **STEP 3. PROCESSED IMAGES ARE EXPORTED:** After all image processing options and parameters are set, clicking on the START button will begin the exporting of the processed image data into 8-bit RGB z-stack files for each time point. These z-stacks are files that can be loaded into LongTracker. In the directory the processed data is being written, a subfolder is created called "RAW_DATA_8bitformat" that contains 8-bit z-stacks of the raw data for each channel and time point.

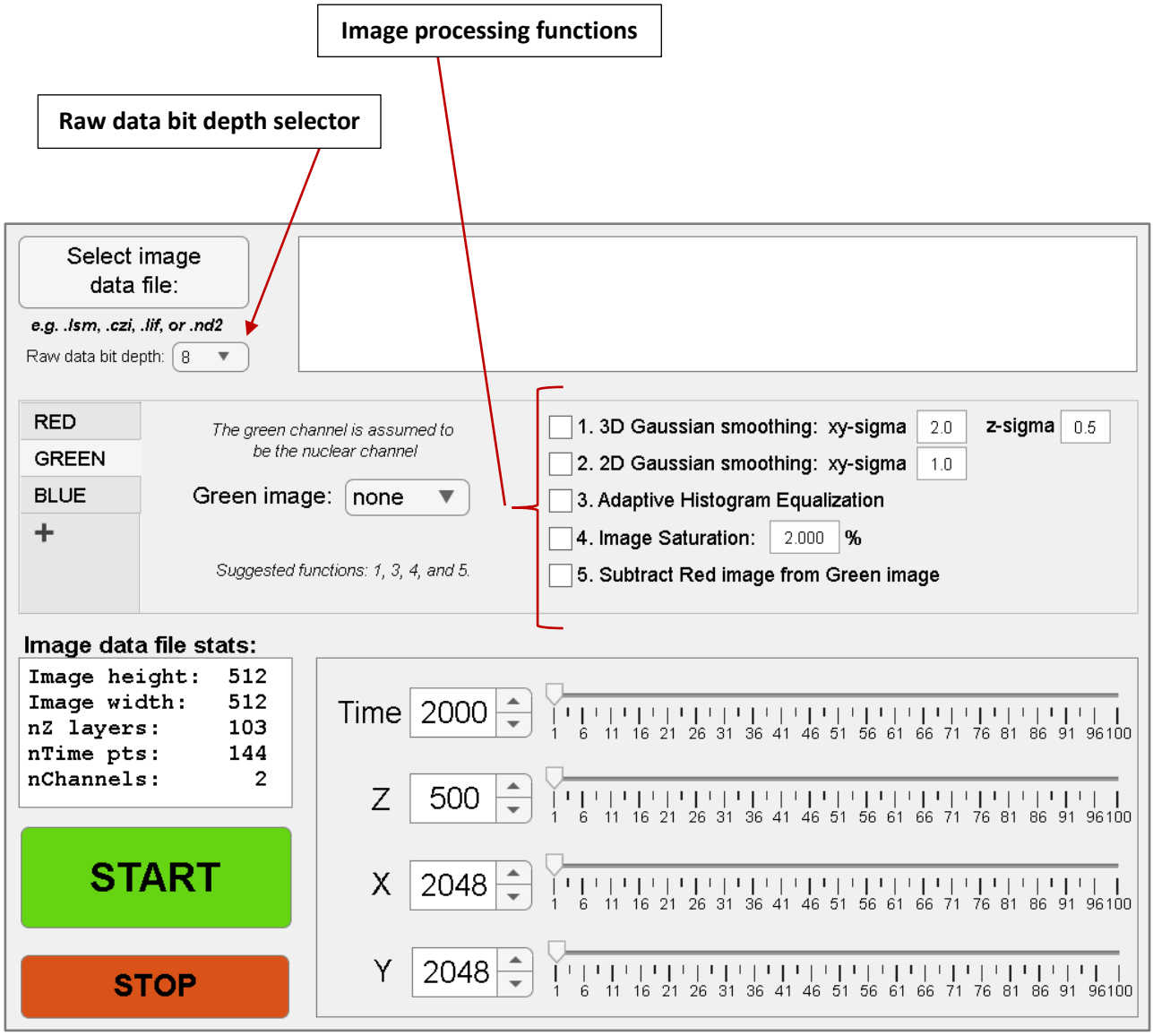
Image processing functions
Raw data bit depth selector
Select image data file:
e.g. .lsm, .czi, .lif, or .nd2
Raw data bit depth: 8
RED
GREEN
BLUE
+
The green channel is assumed to be the nuclear channel
Green image: none
Suggested functions: 1, 3, 4, and 5.
1. 3D Gaussian smoothing: xy-sigma 2.0 z-sigma 0.5
2. 2D Gaussian smoothing: xy-sigma 1.0
3. Adaptive Histogram Equalization
4. Image Saturation: 2.000 %
5. Subtract Red image from Green image
Image data file stats:
Image height: 512
Image width: 512
nZ layers: 103
nTime pts: 144
nChannels: 2
START
STOP
Time 2000
Z 500
X 2048
Y 2048
1 6 11 16 21 26 31 36 41 46 51 56 61 66 71 76 81 86 91 96100