Date: 20210614
Author: george

Requirements:

- Linux
- Slurm already configured
    - CPUs configured
    - Network established between client computer (work stations) and the computing servers (CPU)
- Matlab Code
    - Link
    - “/nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Code/currentDeskew_20210622.m”

Introduction:

Note that when images are collected from the lattice light sheet microscope (LLSM), they are not in a conventional XYZ orientation. Hence (deskew) will transform to the new coordinate system.

This process a subset of the preprocessing protocol. The list of the preprocessing that will be applied to the raw image are as follows:

1. Illumination correction
2. Deskew
3. Chromatic offset

For running the code, please navigate down to the “How to Run > Protocol” section.

Connecting to workstations

1. Checkout the protocol here or go to /nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/Connect_workstations_Remote.pdf

Setting up Slurm on Matlab

1. Checkout the protocol here or go to /nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/Cluster_Preprocessing.pdf

Detailed flowchart of the Code:

1. Add the code repository to the your instance of Matlab
    a. Some custom made codes (ones that we developed) are not within Matlab’s code library. Add the directory of where the code rests to tell matlab where it is
2. Initialize the source and the sink directory
    a. The source is always the folder of acquisition in the Lattice D-drive
    b. There are two sinks:

   i. The first is almost always the biologist’s scratch directory
   ii. The second is datasync3

3. Get the datasync3 sink folders
    a. Raw
    b. Preprocessed
4. Transfer the calibration file from the source to all the sink
5. Load the computed calibration variables into memory
    a. Averaged illumination correction image
    b. Averaged background image
    c. Chromatic offset Z-position
6. Insert a keyword and apply the preprocessing to the folder with the given keyword. Examples:
    a. Keyword = ‘Ex01_’
    b. Keyword = ‘Ex02_’
7. For each keyword, do the following
    a. Find the directory of the acquisition in the source folder (LLSM-d drive) using the keyword
    b. Transfer the acquisition folder to the sink
    c. Transfer the acquisition folder to datasync3_raw
        i. Add the flag
    d. Change the tif image name in the sink if scripting was used
        i. From ‘ex00_iter1_’ to ‘ex00_iter0001_’
            1. The output tif from labview creates mismatch because the fields re organized(iter1, iter,10, iter100, iter2, iter20, iter200,....) and they need to be iter0001, iter0002, iter0010, iter0020, iter0100, iter0200
    e. Organize the tif images into channels
    f. Organize the tif files into a data structure
    g. Preprocess
        i. This procedure is only for sample scan
            1. Illumination correction
            2. Deskew
            3. If more than one channel, then do
                a. Chromatic offset
    h. Transfer the preprocessed images from scratch to datasync3_preprocessed

How to Run

Intro:

The goal is to reduce the number of user inputs. Hence, the only change that needs to be done to the code is to change lines 32 (dirSource), 33 (dirSink), and 81 (for ii = 9:15). This setup applies to a given set up of the microscope used for the data collection (e.g. power, exposure, etc).

- dirSource is the source directory (LLSM D-drive)

- dirSink is scratch folder directory for the biologist
- ii=9:15 is a loop to determine which Ex folder to preprocess. In this case, ii starts from 9 and ends with 15. Hence Ex09, Ex10,...,Ex15 will be preprocessed
    - If only one folder is to be processed, say Ex41, set ii=41

<u>Assumption:</u>

- We assume that the code will be run after all the calibrations are completed. We also assume that there is nothing wrong with calibration acquisitions. Make sure that:
    - Images were collected from the desired camera
        - For example, if using 3-color dichorics, the signal will be emitted to CameraA if you use 488nm channel. Hence if using 488, save to cameraA. For 560 in this dichoic, save to cameraB and for 642, save to cameraA
    - Only one bead in the FOV for chromatic offset
    - Good enough SNR for each of the beads collectd in each of the channels
    - N-bessel is dithered
- The user will need to follow a specific folder structure detailed here. This includes both populating the Calibration files and the experimental acquisition folders
    - /nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/LLSM_Folder_Structure.pdf
- Make sure you have
    - Illumination correction for all colors used for the acquisition, in the desired camera
    - Chromatic offset for each color, from the desired camera
    - Background images for each camera obtained without samples and without illumination

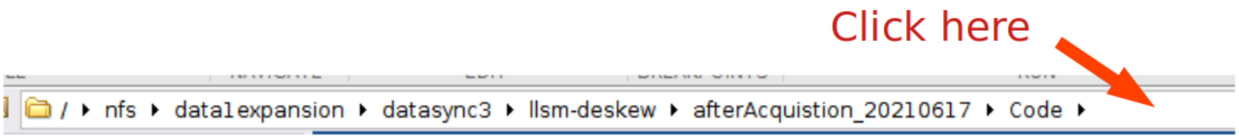
<u>Downloading the code:</u>

1. Using a Linux workstation, navigate yourself to /nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Code/, open currentDeskew_20210622.m and save it to your scratch folder
2. Open matlab. Load the above code

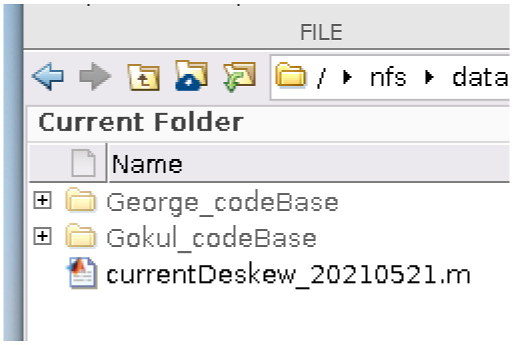
<u>Protocol</u>

1. Find a workstation or connect to it remotely via ssh (access the protocol from a linux computer)
    a. /nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/Connect_workstations_Remote.pdf
2. Check that Slurm is configured with Matlab (access the protocol from a linux computer)
    a. /nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/Cluster_Preprocessing.pdf
3. Open Matlab and then download the code to your scratch
    a. If matlab is not open, open terminal, connect to the remote computer if necessary and then execute the command “matlab”.

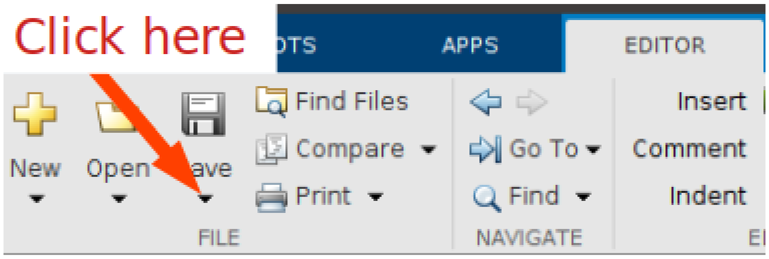
b. Once the Matlab GUI opens, copy and paste the following to matlab’s file explorer. Click on the whitespace after the last arrow. Highlight the directory and then copy and paste:
/nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Code/


Click here
/ ▸ nfs ▸ data1expansion ▸ datasync3 ▸ llsm-deskew ▸ afterAcquistion_20210617 ▸ Code ▸

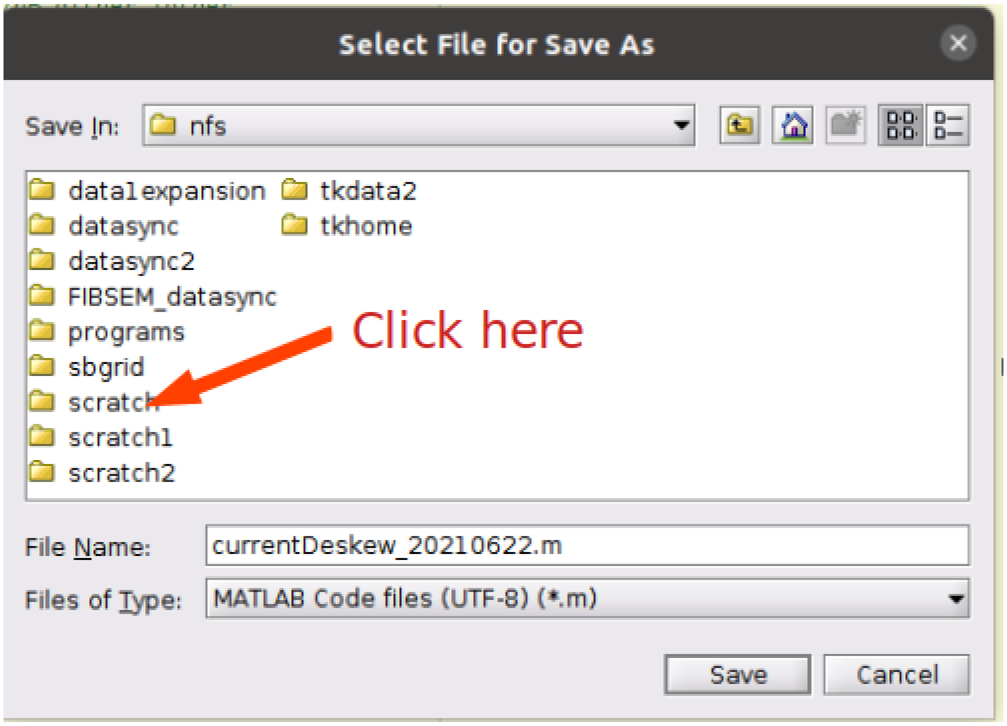
c. Find currentDeskew_20210521.m, right click on the file and select open.



d. Under EDITOR > Click the downward arrow under save, and select “Save As”.



e. Save the code to your scratch. Click on ( ) to go back to the parent directory. Keep clicking here until you are in /nfs. Then click on scratch, then your scratch folder



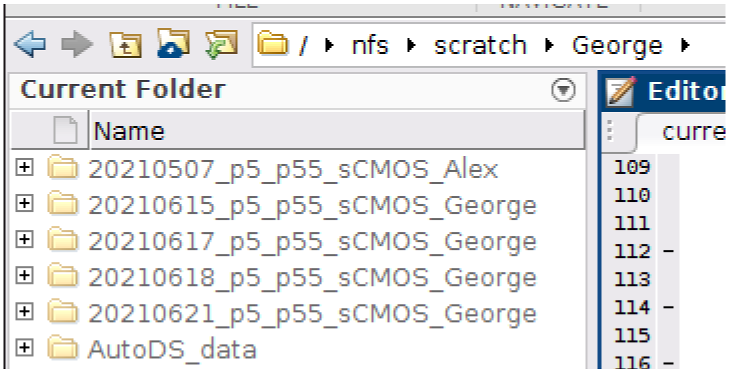
Select File for Save As

Save In: nfs

File Name: currentDeskew_20210622.m

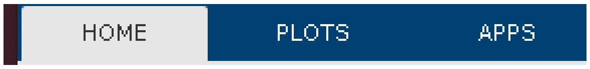
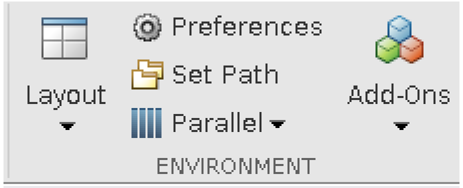
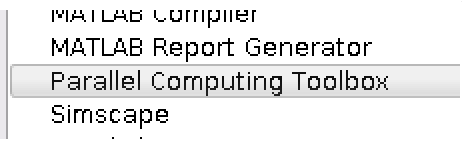
Files of Type: MATLAB Code files (UTF-8) (*.m)

f. In the file explorer, make sure you are in your scratch directory. Copy and paste "/nfs/scratch/" and then click on your scratch directory



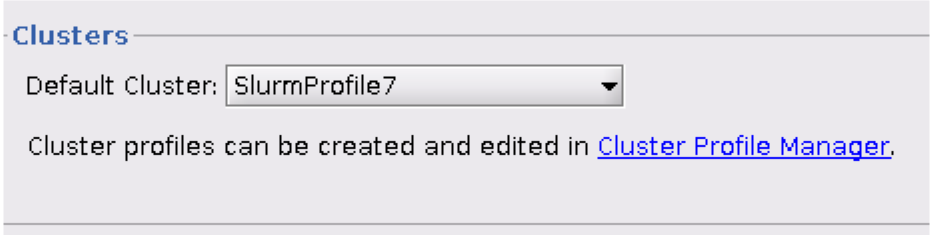
/ ▸ nfs ▸ scratch ▸ George ▸

Current Folder
Name
20210507_p5_p55_sCMOS_Alex
20210615_p5_p55_sCMOS_George
20210617_p5_p55_sCMOS_George
20210618_p5_p55_sCMOS_George
20210621_p5_p55_sCMOS_George
AutoDS_data
Editor
curre
109
110
111
112
113
114
115
116

4. Go to Home () > Preferences () > Parallel Computing Toolbox (). Check that your "Default Cluster" is set to

"SlurmProfileN", where N is a natural number (ex. SlurmProfile1, SlurmProfile2,...). This should not be local. If you see local, then click on "local" and select "SlurmProfileN". If you do not see "SlurmProfile", please follow the following protocol to set up slurm

   a. /nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/Cluster_Preprocessing.pdf

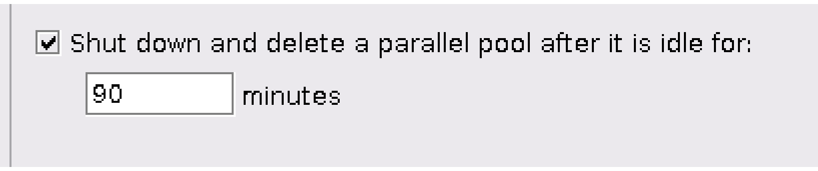


Clusters

Default Cluster: SlurmProfile7

Cluster profiles can be created and edited in Cluster Profile Manager.

5. By the default SlurmProfile configuration, once the user has reserved some CPUs, these will be reserved to you until idle time of 90 minutes (don't' use matlab at all for 90min). The timer resets every execution the user carries out in matlab.

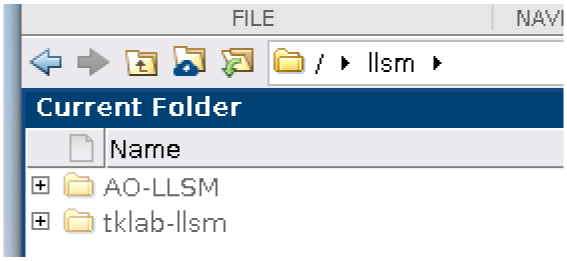


☑ Shut down and delete a parallel pool after it is idle for:

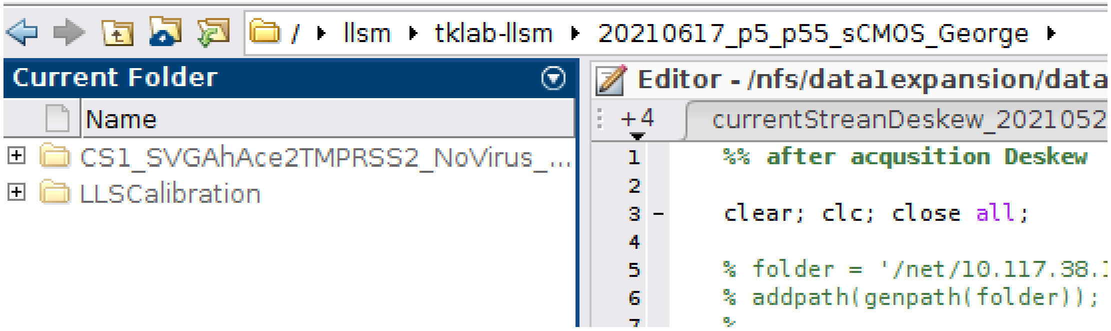
90 minutes

Code:

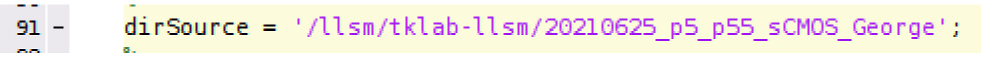
1. The code is broken up into three sections. In the first section, you will need to manually change three lines: 91 (dirSource), 101 (dirSink) and 111 (N_CPU), which takes about 3~4 minutes to execute. In the second section, only change one line: 193 (ExNumbers). The third section will be run when all experiments are completed (when the user is ready to clean up)
   a. Go to line 91 and enter the parental acquisition folder that contains all the acquisitions for the experiment (Ex. dirSource = '/llsm/tklab-llsm/20210621_p5_p55_sCMOS_George';
      i. Go to Matlab's file directory and type in /llsm



      ii. Choose either /tklab-llsm (standard LLSM) or /ao-llsm (AO llsm). You will be in the D-drive where the acquisitions will be saved. Navigate to today's parent experimental folder. Copy this directory and paste as the argument for dirSource

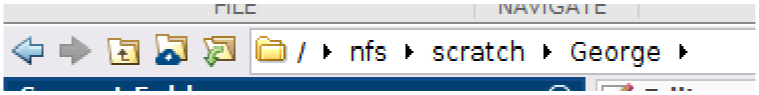

/ ▸ llsm ▸ tklab-llsm ▸ 20210617_p5_p55_sCMOS_George ▸
Current Folder
Name
CS1_SVGAhAce2TMPRSS2_NoVirus_...
LLSCalibration
Editor - /nfs/datalexpansion/data
+4
currentStreanDeskew_2021052

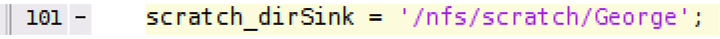
```
1      %% after acqusition Deskew
2
3 -    clear; clc; close all;
4
5      % folder = '/net/10.117.38.
6      % addpath(genpath(folder));
```



```
91 -     dirSource = '/llsm/tklab-llsm/20210625_p5_p55_sCMOS_George';
```

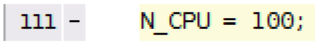
b. Follow the steps above (1a) to navigate to the biologists's scratch directory. Type in /scratch into Matlab's file explorer, and select the biologist's folder. Copy and paste as the argument to scratch_dirSink (line 101)





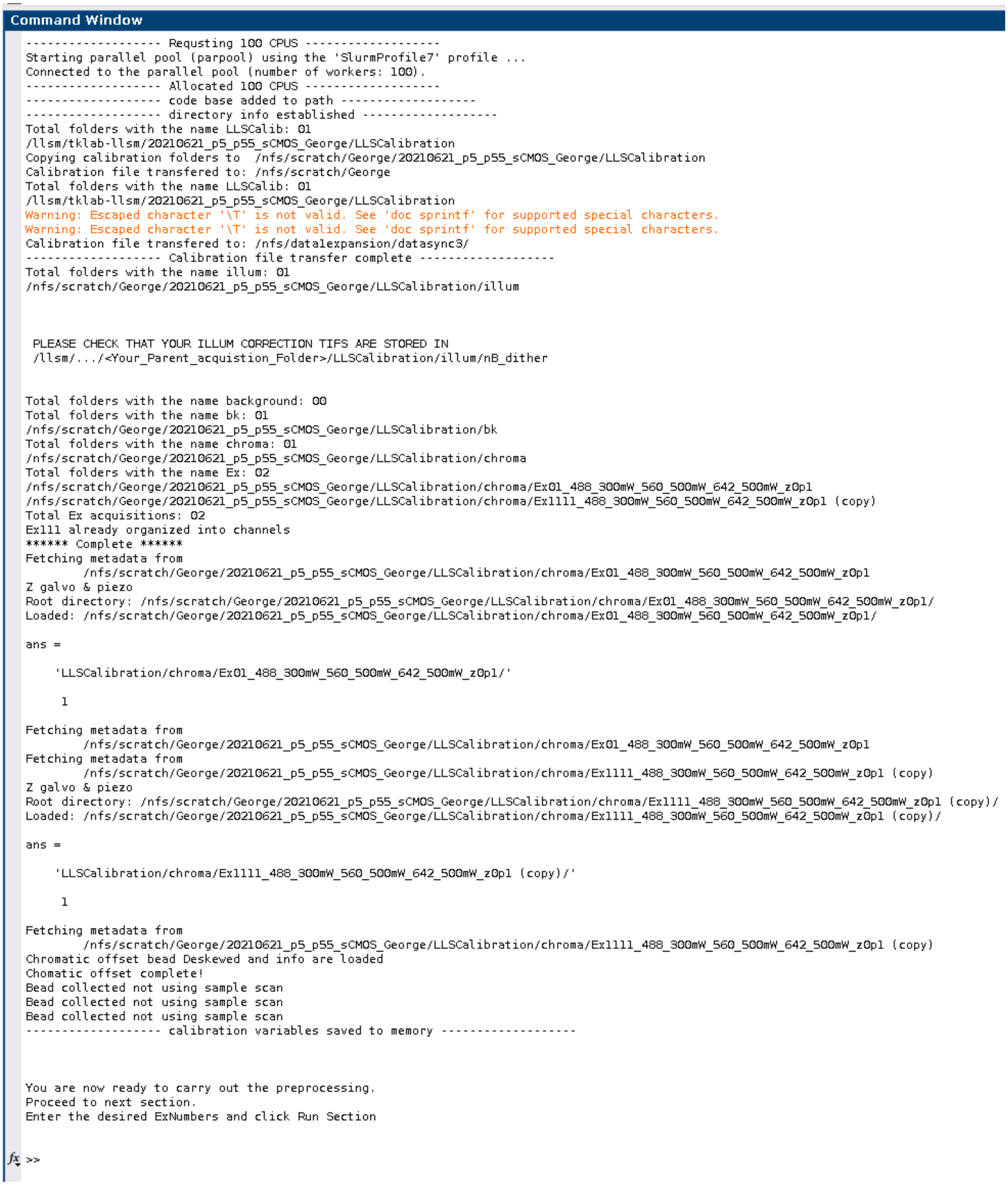
```
101 -     scratch_dirSink = '/nfs/scratch/George';
```

c. Go to line 111 and enter the number of CPUs you would like to use. A general thumb is that for one timepoint, allocate one CPU independent of the laser lines used for collection (eg. 3 channels, 100 timepoints, then recommended to set N_CPU =100). By default, the user will have access to these nodes until 90 minutes of inactivity.



```
111 -    N_CPU = 100;
```

2. Once the above is complete, please click on the first section (should be highlighted in yellow from lines 1 to 183) and then click Run Section (Run Section). Once executed, all codes should run smoothly (no red fonts -- means error) with the following output

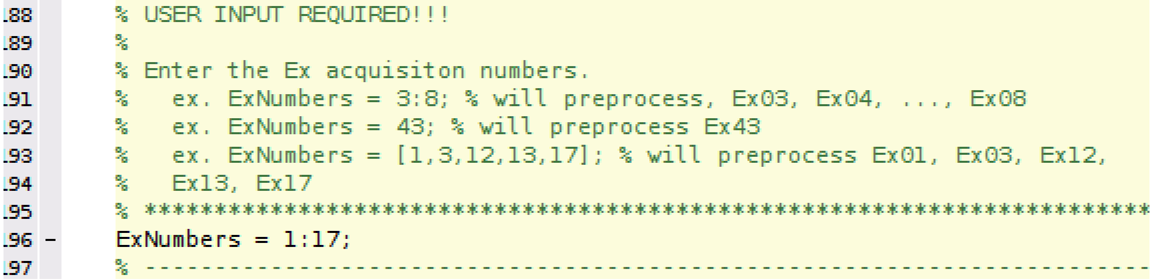


```
------------------- Requsting 100 CPUS -------------------
Starting parallel pool (parpool) using the 'SlurmProfile7' profile ...
Connected to the parallel pool (number of workers: 100).
------------------- Allocated 100 CPUS -------------------
------------------- code base added to path -------------------
------------------- directory info established -------------------
Total folders with the name LLSCalib: 01
/llsm/tklab-llsm/20210621_p5_p55_sCMOS_George/LLSCalibration
Copying calibration folders to  /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration
Calibration file transfered to: /nfs/scratch/George
Total folders with the name LLSCalib: 01
/llsm/tklab-llsm/20210621_p5_p55_sCMOS_George/LLSCalibration
Warning: Escaped character '\T' is not valid. See 'doc sprintf' for supported special characters.
Warning: Escaped character '\T' is not valid. See 'doc sprintf' for supported special characters.
Calibration file transfered to: /nfs/datalexpansion/datasync3/
------------------- Calibration file transfer complete -------------------
Total folders with the name illum: 01
/nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/illum



 PLEASE CHECK THAT YOUR ILLUM CORRECTION TIFS ARE STORED IN
 /llsm/.../<Your_Parent_acquistion_Folder>/LLSCalibration/illum/nB_dither


Total folders with the name background: 00
Total folders with the name bk: 01
/nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/bk
Total folders with the name chroma: 01
/nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma
Total folders with the name Ex: 02
/nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex01_488_300mW_560_500mW_642_500mW_z0p1
/nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex1111_488_300mW_560_500mW_642_500mW_z0p1 (copy)
Total Ex acquisitions: 02
Ex111 already organized into channels
****** Complete ******
Fetching metadata from
        /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex01_488_300mW_560_500mW_642_500mW_z0p1
Z galvo & piezo
Root directory: /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex01_488_300mW_560_500mW_642_500mW_z0p1/
Loaded: /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex01_488_300mW_560_500mW_642_500mW_z0p1/

ans =

    'LLSCalibration/chroma/Ex01_488_300mW_560_500mW_642_500mW_z0p1/'

     1

Fetching metadata from
        /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex01_488_300mW_560_500mW_642_500mW_z0p1
Fetching metadata from
        /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex1111_488_300mW_560_500mW_642_500mW_z0p1 (copy)
Z galvo & piezo
Root directory: /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex1111_488_300mW_560_500mW_642_500mW_z0p1 (copy)/
Loaded: /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex1111_488_300mW_560_500mW_642_500mW_z0p1 (copy)/

ans =

    'LLSCalibration/chroma/Ex1111_488_300mW_560_500mW_642_500mW_z0p1 (copy)/'

     1

Fetching metadata from
        /nfs/scratch/George/20210621_p5_p55_sCMOS_George/LLSCalibration/chroma/Ex1111_488_300mW_560_500mW_642_500mW_z0p1 (copy)
Chromatic offset bead Deskewed and info are loaded
Chomatic offset complete!
Bead collected not using sample scan
Bead collected not using sample scan
Bead collected not using sample scan
------------------- calibration variables saved to memory -------------------



You are now ready to carry out the preprocessing.
Proceed to next section.
Enter the desired ExNumbers and click Run Section


fx >>
```

3. In the second section, you will only need to enter the acquired experimental folder numbers on line 196. Currently, line 196 is set to `ExNumbers = 1:17;`, which means it will preprocess all folders with the name Ex01 , Ex02, …. up to Ex17. If you want to

process a single folder, say Ex01, then change the code to `for ii = 1`. Examples are shown below:

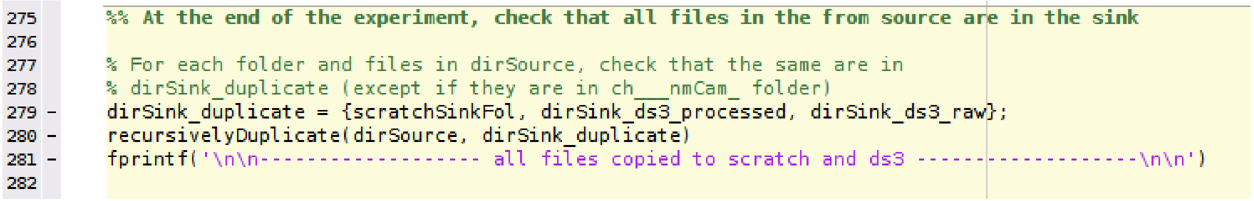
a. ExNumbers = 3:8; % will preprocess, Ex03, Ex04, ..., Ex08
b. ExNumbers = 43; % will preprocess Ex43
c. ExNumbers = [1,3,12,13,17]; % will preprocess Ex01, Ex03, Ex12, Ex13, Ex17



```
.88      % USER INPUT REQUIRED!!!
.89      %
.90      % Enter the Ex acquisiton numbers.
.91      %   ex. ExNumbers = 3:8; % will preprocess, Ex03, Ex04, ..., Ex08
.92      %   ex. ExNumbers = 43; % will preprocess Ex43
.93      %   ex. ExNumbers = [1,3,12,13,17]; % will preprocess Ex01, Ex03, Ex12,
.94      %   Ex13, Ex17
.95      % *************************************************************************
.96  -   ExNumbers = 1:17;
.97      % -------------------------------------------------------------------------
```

4. Once the above is complete, make sure that the second section is highlighted in yellow (from lines 184 to 274) and click run section.

5. If in a later time, more acquisitions were collected, please repeat step 3.
6. When all the acquisitions are completed and you are ready to wrap up your experiment or, changing microscope conditions (changing filters, dichroics, cameraSave, etc…) then run the third section of the code on from line 275 to line 283. This code will transfer all files in from the source directory (dirSource) to the sink directories (scratch, ds3) without any duplicates, and checks if some files are in the ch___nmCam_ folder.



```
275      %% At the end of the experiment, check that all files in the from source are in the sink
276
277      % For each folder and files in dirSource, check that the same are in
278      % dirSink_duplicate (except if they are in ch___nmCam_ folder)
279  -   dirSink_duplicate = {scratchSinkFol, dirSink_ds3_processed, dirSink_ds3_raw};
280  -   recursivelyDuplicate(dirSource, dirSink_duplicate)
281  -   fprintf('\n\n------------------- all files copied to scratch and ds3 -------------------\n\n')
282
```

Code:

```
%% After Acquistion Deskew
%
========================================================================================
=
% Description:
%
========================================================================================
=
%   The detailed protocol for this code is ready and stored:
%
/nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/After-Collectio
n-deskew.pdf
%       or
%
https://docs.google.com/document/d/12wCAudD5LOY8dr9uM5kSONCDbDZjeoQ7PNIYUeII1O
o/edit?usp=sharing
%
%   This code is divided in two sections:
%       1. Computing the calibration variables
%       2. Carrying out the data preprocessing pipeline, which includes:
%           i.   Illumination Correction
%           ii.  Deskew
%           iii. Chromatic offset (if more than one channel used)
%
%   Run 1. after all the calibration acquistions are complete (illum, bk,
%   chroma)
%   Run 2. after cell/sample acquisiton for each Ex (Ex01, Ex02,...)
%
%
========================================================================================
=
% Assumptions:
%
========================================================================================
=
%   1. Please run the first section of the code (1 above) AFTER the
%   microscopist have acquired the following verified calibration
%   files. Please make sure that the acquisitons are good (check with
%   imageJ)
%       1. Illumination correction in illum/nB_dither folder
%           i. Dithered, collected with the correct Camera
%       2. Chromatic offset in /chroma/Ex%02d_..._z0p%d
%       3. background (dark current) in /bk
%
%   2. Please follow the following folder structure:
%
/nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/LLSM_Folder
_Structure.pdf
%       or
%
https://docs.google.com/document/d/1BIs-AwhvWZcoKA77HgQDHzT-NPRCguaA41FyH1VW0
OE/edit?usp=sharing
%
%   3. We assume that all experiments (Ex01, Ex02,...) have been collected
%   using a calibrated microscope, and its calibration variables
%   (illumination correction, chromatic offset) are constant.
%
%   4. Everytime the user changes any microscope variable (except AOTF and
%   laser source power), the user will need to run this code from the
%   beggining with the new /LLSCalibrations. Examples of this include:
%       i. Changing the filters
%       ii. Changing the dichorics (make sure you change the CamSave also)
%       iii. Changing the annulus NA
%       iv. Changing to a different media
%
%
===========================================================================
=
% Protocol
%
===========================================================================
=
%   1. Acquire calibration acquisitions
%       i.   illumination correction
%       ii.  background (dark current)
%       iii. chromatic offset (only have one bead in the FOV)
%   2. Navigate to your scratch directory
%       ex. /nfs/scratch/George
%   2. Please change the following in the code:
%       i.  dirSource to the D-drive of the lattice computer
%           ex. dirSource = '/llsm/tklab-llsm/20210621_p5_p55_sCMOS_George';
%       ii. scratch_dirSink to the biologist's scratch
%           ex. scratch_dirSink = '/nfs/scratch/George';
%   3. Run the first section of the code
%       until disp('------------------- calibration variables saved to memory -------------------')
%   4. Save cell/sample acquisition to dirSource in LabView. Save Ex%02d
%      ex. Ex01, Ex02, Ex03
%   5. Change ExNumbers to the Ex number
%      ex. ExNumbers = 1:3 (this means Ex01, Ex02, Ex03)
%   6. Run the second section of the code
%   7. Repeat 5 until no more acquistions
%
%
=============================================================================
=
% Code
%
=============================================================================
=

% clear all variables
clear; clc; close all;

% -----------------------------------------------------------------------------
% *****************************************************************************
% USER INPUT REQUIRED!!!
%
% dirSource is the d-drive of the lattice, where the acquisitons will be
% saved.
%   ex. dirSource = '/llsm/tklab-llsm/20210621_p5_p55_sCMOS_George';
% *****************************************************************************
dirSource = '/llsm/tklab-llsm/20210621_p5_p55_sCMOS_George';
% -----------------------------------------------------------------------------

% -----------------------------------------------------------------------------
% *****************************************************************************
% USER INPUT REQUIRED!!!
%
% The biologist's scratch directory
%   ex. scratch_dirSink = '/nfs/scratch/George';
% *****************************************************************************
scratch_dirSink = '/nfs/scratch/George';
% -----------------------------------------------------------------------------

% -----------------------------------------------------------------------------
% *****************************************************************************
% USER INPUT REQUIRED!!!
%
% Number of CPUs you would like to allocate. A rule of thumb is that if you
% have 100 timepoints (100 tifs per channel), then N_CPU = 100.
% **********************************************************************
N_CPU = 100;
% ----------------------------------------------------------------------

% delete the previous reservation of the CPUs
delete(gcp('nocreate'))

fprintf('------------------- Requsting %d CPUS -------------------\n', N_CPU)
% request the N_CPU from SLURM
parpool(N_CPU)

% message log to notify user
fprintf('------------------- Allocated %d CPUS -------------------\n', N_CPU)

% add the code base to Matlab's path
folder = '/scratch/llsm-deskew/afterAcquistion_20210617/Code';
addpath(genpath(folder));

% message log to notify user

disp('------------------- code base added to path -------------------')

% datasync3 directory
% For example, the current experiment will be saved in datasync3 in
% /nfs/data1expansion/datasync3/tklab-llsm/20210621_p5_p55_sCMOS_George
datasync_dirSink = '/nfs/data1expansion/datasync3/';

% sink directory for scratch (combines the parent acquistion folder
%   ex. /nfs/scratch/George/20210618_p5_p55_sCMOS_George
scratchSinkFol = getSinkFol(dirSource, scratch_dirSink);

% get dirSink_DS3 directories
[dirSink_ds3_raw, dirSink_ds3_processed] = getDS3Directories(dirSource, datasync_dirSink);

disp('------------------- directory info established -------------------')

% Transfer calibration folder and compute calibration variables

% transfer dirSource to scratch
transferCalibrationFiles2(dirSource, scratch_dirSink);

% transfer dirSource to datasync3
transferCalibrationFiles2_DS3(dirSource, datasync_dirSink);
```

```
disp('------------------ Calibration file transfer complete ------------------');

% Get the illumination correction directories for each channel
I_directories = getIllum(scratchSinkFol);

% Compute illumination correction variable
I = getIllumImages(I_directories);

% Get the background (darkcurrent) directories for each channel
B_directories = getBk(scratchSinkFol, I.doIllum);

% compute the background (darkcurrent) variable
B = getBkImages(B_directories);

% Get the chromatic offset directories for each channel
C_directories = getChroma2(scratchSinkFol);
try % if you do have chroma folder (aka you did collect the chromatic offset)

   % compute the chromatic offset variable
   C = getChromaOffsetValues(C_directories);
catch % if you didnt collect chromatic offset beads (eg. one channel experiment)

   % compute the chromatic offset variable
   C.doIllum = false;
end

disp('------------------ calibration variables saved to memory ------------------')

fprintf('\n\n\nYou are now ready to carry out the preprocessing.\nProceed to next section.
\nEnter the desired ExNumbers and click Run Section\n\n\n')

%% Reserve nodes, Find the acquisiton folder, preprocess and transfer files

% ---------------------------------------------------------------------
% ************************************************************************
% USER INPUT REQUIRED!!!
%
% Enter the Ex acquisiton numbers.
%   ex. ExNumbers = 3:8; % will preprocess, Ex03, Ex04, ..., Ex08
%   ex. ExNumbers = 43; % will preprocess Ex43
%   ex. ExNumbers = [1,3,12,13,17]; % will preprocess Ex01, Ex03, Ex12,
%   Ex13, Ex17
% ************************************************************************
ExNumbers = 2:3;
```

```
% ------------------------------------------------------------------------

% counter for the loop
counter = 1;
% initialize empty cell
keywords = cell(1,length(ExNumbers));

% change the Ex number here. Default set to Ex02 and Ex03. If just one
% folder, set ii=1 (Ex01)
for ii = ExNumbers
   % fill in the empty cell
   keywords{counter} = sprintf('Ex%02d',ii);

   % update the counter
   counter = counter + 1;
end

% show the keywords for checks
disp(keywords)


% set dirSink as scratchSinkFol
dirSink = scratchSinkFol;

% iterate the loop for each keyword
for qq = 1:length(keywords)

   % parse the keyword from type cell to type char
   keyword = char(keywords{qq});

   % run the search algorithm (depth first search) and find all folders
   % with the same keyword in all subfolders in dirSource. dirSource is
   % lattice's D-drive
   Exs =  findKeywordDFS(dirSource, keyword);

   % iterate over all folders (usually should just be one -- since all
   % folders should have unique Ex folder extension)
   for ii=1:length(Exs)
      % Ex acquisiton folder
      Ex = Exs{ii};

      % transfer from local to scratch
      sinkEx = transferLocalToSink(dirSink, Ex);
```

```
    % transfer from local to datasync3/raw
    transferLocalToSink(dirSink_ds3_raw, Ex);

    % change name for scripting mode
    checkIter(Ex)

    % organize scartch folders into channels
    addChannels(sinkEx)

    % load the data
    data = AutoGU_loadConditionData3D(sinkEx);

    % start the timer
    t = tic;

    % do the preprocessing
    %   a. Illumination correction
    %   b. Deskew
    %   c. Chromatic offset
    preprocess3(data, sinkEx, I, B, C)

    % stop the timer and check how long it took
    toc(t);

    % transfer the preprocessed data into datasync3
    transferLocalFolderToSink(dirSink_ds3_processed, sinkEx)
  end
end


fprintf('\n\n\n\nPreprocessing completed!!\n\n\n\n');
disp('Change ExNumbers if you have acquired more data')
```