**Instructions for the MATLAB script package “ClnColorAnalysis”**

*Juwell Wu, Last Revision 2015-11-10*

This package includes custom MATLAB scripts for performing single clonal color analysis and clonal assignment of polyclonal populations as described in Wu et al. “Defining Clonal Color in Fluorescent Multi-Clonal Tracking” (2015). The scripts were created in MATLAB R2011b and last revised on MATLAB R2014b, using a 2.6GHz Intel Core i7 MacBookPro with 8Gb memory. This package also includes sample color data for running a demo and a copy of the DEMO output.

OVERVIEW OF SOFTWARE PACKAGE

MATLAB scripts in “ClnColorAnalysis” will create a DEMO directory “ClnColorDEMO” with the following sub-directories. These sub-directories will contain clonal color information of (mono)clonal populations and clonal assignment of polyclonal populations.

|  |  |
| --- | --- |
| **Sub-directory** | **Generated from custom code** |
| T0.01P0.01Step20tfmgrid | tfmgridcreate.m |
| T0.2P0.2Step1tfmgrid | tfmgridcreate.m |
| T5.0P5.0Step1tfmgrid | tfmgridcreate.m |
| xAF output | xAF.m |
| RCB output | relcellbright\_batch.m |
| relclnbright output | relclnbright\_batch.m |
| cmodspd output | cmodspd\_batch.m, chromspdanalysis\_batch.m |
| chromtfmplot output | chromtfmplot\_batch.m |
| bUnwarpJ output | <from bUnwarpJ plugin, Fiji> |
| cloneassign output | cloneassign\_batch.m |
| chromsphplot output | chromsphplot\_batch.m |

|  |  |
| --- | --- |
| **Sub-directory** | **Information** |
| T0.01P0.01Step20tfmgrid | Properties of transformed THETA-PHI grid |
| T0.2P0.2Step1tfmgrid | Properties of transformed THETA-PHI grid |
| T5.0P5.0Step1tfmgrid | Properties of transformed THETA-PHI grid |
| xAF output | Properties of 1xAF |
| RCB output | Relative cell brightness |
| relclnbright output | Relative clonal brightness |
| cmodspd output | Chromatic mode and spread information for monoclonal populations, chromatic stability |
| chromtfmplot output | Histogram of polyclonal populations in partial, transformed THETA-PHI grid |
| bUnwarpJ output | Correction parameters for non-ideal chromatic stability |
| cloneassign output | Clonal assignment of each cell in polyclonal populations |
| chromsphplot output | Spherical scatter plot of clonally assigned polyclonal population. Spherical histogram. |

DEMO color data files should be placed in “ClnColorAnalysis/ColorData”. They contain FlowJo exports of singlet cell color data from different cell populations:

|  |  |
| --- | --- |
| **Color data file** | **Cell population** |
| mcln01 | Monoclonal population 01, analyzed in a single FACS session. Used for clonal assignment. |
| mcln11-1 | A subset of data points from the file “mcln01”; for chromatic stability demonstration. |
| mcln11-2 | A subset of data points from “mcln01” that are not in “mcln11-1”; for chromatic stability demonstration. Chromatic stability is near-ideal (0) for “mcln11” as “mcln11-1”, “mcln11-2” contain color data from the same monoclonal population analyzed at the same time. |
| mcln02 | Monoclonal population 02, analyzed in a single FACS session. Used for clonal assignment. |
| mcln22-1 | Identical to “mcln02”; for demonstrating chromatic (in)stability and bUnwarpJ registration. |
| mcln22-2 | Same monoclonal population as “mcln02”, but analyzed at a different day / FACS session; for demonstrating chromatic (in)stability and bUnwarpJ registration. Chromatic stability is non-ideal (0) for “mcln22”. DEMO for bUnwarpJ chromatic instability correction shifts the chromatic position of “mcln22-1” to match that of “mcln22-2”, which is needed for proper clonal assignment of “pcln02”. |
| pcln01 | A random mix of data points from “mcln01” and “mcln02” (=”mcln22-1”). Source of each data point is listed in column 4 (“mcln”). Clonal assignment is accurate without bUnwarpJ chromatic instability correction as participant clonal color metrics were defined with mcln01 and mcln02 (=mcln22-1). |
| pcln02 | A random mix of data points from “mcln01” and “mcln22-2”. Source of each data point is listed in column 4 (“mcln”). Clonal assignment requires bUnwarpJ chromatic instability correction as participant clonal color metrics were calculated from “mcln01” and “mcln02” (=”mcln22-1”). |
| NoFP | Un-transduced population. Used for defining 1xAF. |

The folder “SAMPLE ClnColorDEMO” contains a copy of the expected outcome from running the demo (the sub-directories “T0.01P0.01Step20tfmgrid”, T0.2P0.2Step1tfmgrid” and “T5.0P5.0Step1tfmgrid” are excluded to limit the size of the software package). Screenshots are included in the folder to illustrate the drawing of 1xAF polygon gate and the instructions for running bUnwarpJ.

RUNNING THE DEMO

A. GITHub

1. Download the software package at this URL:

<https://github.com/juwellwwu/ClnColorAnalysis.git>

1. There are two .zip files in the package, “ColorData.zip” and “SAMPLEClnColorDEMO.zip”, which are stored with Git LFS. If they do not unzip properly, visit:

<https://github.com/juwellwwu/ClnColorAnalysis.git>

, click on the .zip file name, then click on “View Raw”. The zip file downloaded should open properly.

*B. MATLAB*

1. Rename the unzipped package and rename it “ClnColorAnalysis”. Copy the directory “ColorData” into “ClnColorAnalysis”.
2. Move the directory “ClnColorAnalysis” into MATLAB’s initial working folder (= startup folder).
3. Download codes from MATLAB Central File Exchange (<http://www.mathworks.com/matlabcentral/fileexchange/>) and unzip them in the directory “ClnColorAnalysis”.
   1. Inhull (<http://www.mathworks.com/matlabcentral/fileexchange/10226-inhull>)
   2. PATCH\_3DArray (<http://www.mathworks.com/matlabcentral/fileexchange/28497-plot-a-3d-array-using-patch>)
   3. export\_fig (<http://www.mathworks.com/matlabcentral/fileexchange/23629-export-fig>)
4. Add “ClnColorAnalysis” and all its sub-directories to MATLAB’s path.
5. Create the m.file “sphere\_1stoct.m”. In MATLAB’s command window, type:
   * + - 1. open(‘sphere’)
   1. This will open the source code for the MATLAB function sphere.m. Replace the values for the variables “theta”,”phi”,”cosphi” and “sintheta” to as follows:
      * + 1. theta = (-n:2:n)/n\*pi/4+pi/4;
          2. phi = (-n:2:n)'/n\*pi/4+pi/4;
          3. cosphi = cos(phi);
          4. sintheta = sin(theta);
   2. Save file as “sphere\_1stoct.m” in “ClnColorAnalysis” directory.
6. Run the script “tfmgridcreate.m” three times. For the prompt “Choose THETA-PHI grid step”, choose
   1. 1
   2. 2
   3. 3

respectively. Note: running this script will take >5.3GB of space.

1. Run the script “xAF.m”. Follow on-screen instructions to draw the 1xAF polygon gates.
2. Run the script “relcellbright\_batch.m” twice. In the section “USER INPUTS”, enter
   1. “ClnColorAnalysis/ColorData/Monoclone”
   2. “ClnColorAnalysis/ColorData/Polyclone”

for the variable “BatchInputFolder” respectively.

1. Run the script “relclnbright\_batch.m”. Relative clonal brightness data are generated in this script.
2. Run the script “cmodspd\_batch.m” six times. In the section “USER INPUTS”, enter values
   1. 0.5
   2. 0.25
   3. 0.1
   4. 0.05
   5. 0.02
   6. 0.01

for the variable “IsovalFrac” (= %nmax chromatic spreads, in decimals) respectively. Chromatic mode and chromatic spread data are generated by this script.

1. Run the script “chromspdanalysis\_batch.m” six times. In the section “USER INPUTS”, enter values
   1. 'ClnColorDEMO/cmodspd output/Isofrac0.50'
   2. 'ClnColorDEMO/cmodspd output/Isofrac0.25'
   3. 'ClnColorDEMO/cmodspd output/Isofrac0.10'
   4. 'ClnColorDEMO/cmodspd output/Isofrac0.05'
   5. 'ClnColorDEMO/cmodspd output/Isofrac0.02'
   6. 'ClnColorDEMO/cmodspd output/Isofrac0.01'

for the variable “BatchCvxHullInputFolder" respectively. Chromatic stability data are generated by this script.

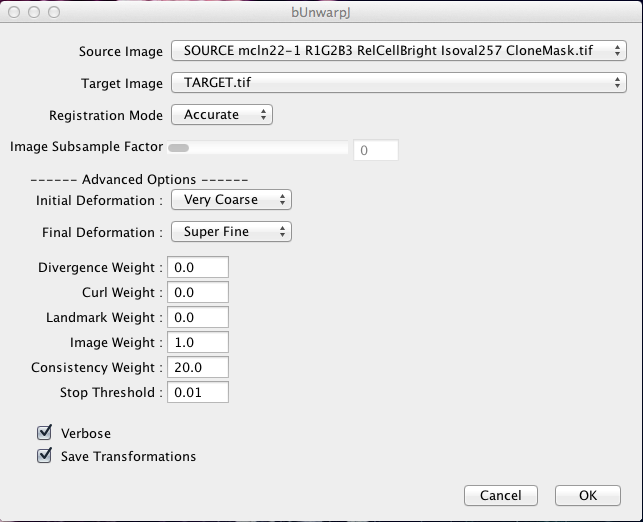
1. If want to correct for non-ideal chromatic stability, run the script “chromtfmplot\_batch.m” and perform the instructions for bUnwarpJ. Otherwise, skip to next step.
2. Run the script “cloneassign\_batch.m”. If not correcting for non-ideal chromatic stability, in the section “USER INPUTS”, enter value ‘n’ for the variable “RegShiftMaskQuery.” If correcting, enter ‘y’. Clonal assignments of polyclonal populations are performed by this script.
3. Run the script “chromspdanalysis\_batch.m”. The current setup plots the spherical scatter plot of polyclonal populations with each cell painted by its assigned clonal ID.

*C. bUnwarpJ in Fiji*

“bUnwarpJ” is a plug-in in Fiji. For the demo, the 50% chromatic spread of “mcln22-1” (SOURCE) is to be matched to the histogram peak of “mcln22-2” (TARGET). The “warping” is then used for proper clonal assignment of the polyclonal population “pcln02”.

1. Create directory “ClnColorDEMO>bUnwarpJ output”
2. Prepare SOURCE image: the SOURCE image for registration is the sum of chromatic spreads (50% isosurface) of all participant clones (“cmodspd output>Isofrac0.50>CloneMasks>\*CloneMask.tif”). For the demo, the SOURCE image is “SOURCE mcln22-1 R1G2B3 RelCellBright Isoval257 CloneMask.tif”. Note: \* CloneMask.tif created in MATLAB may open with 255 value for (black) background and 0 value for (white) chromatic spread area. Invert LUT and invert image so that the black blackground = 0 and white chromatic spread area = 255. Specify white chromatic spread areas as ROIs (“Analyze>Analyze Particles…”). Save ROI properties as SOURCE.roi (1 ROI) or SOURCE.zip (for multiple ROIs).
3. Open the transformed THETA-PHI histogram image “ClnColorDEMO>chromtfmplot output>\*LOG10 pTfmHistogram.tif”. Run “Process>Filters>Mean…”, enter “2.0” pixels for Radius. Change LUT to 3-3-2 RGB (“Image>Lookup Tables>3-3-2 RGB”). Open SOURCE.roi (or SOURCE.zip), move each ROI to match the histogram peak position of its corresponding clone. Update all ROIs and save their properties as TARGET.roi (1 ROI) or TARGET.zip (multiple ROIs).
4. Prepare TARGET image. Create new image (“File>New>Image…”): 8-bit, black background and of the same size as SOURCE image (3240x810pixels). Open TARGET.roi (or TARGET.zip). Fill in the registered ROIs in the new image and save the image as TARGET.tif. The background of TARGET.tif should also be black with value 0 and the chromatic spread areas should be white with value 255.
5. Registration with bUnwarpJ

* Open both SOURCE and TARGET files. Start bUnwarpJ (“Plug-ins> Registration>bUnwarpJ”).
* In the pop-up window “bUnwarpJ”, enter the following parameters:



* After registration, save all files. Create a copy of the file “[TARGET img name]\_inverse\_transf.txt” and rename it “[TARGET img name]\_inverse\_transf RAW.txt”. Leave SOURCE and TARGET images open.
* Start bUnwarpJ (“Plug-ins> Registration>bUnwarpJ”) again. When the pop-up window “bUnwarpJ” appears, click on the “File” icon on Fiji’s menu bar. The pop-up window “I/O Menu” appears. Click on “Convert Transformations to RAW.”
* In the pop-up window “Load elastic transformation file”, open the file “TARGET img name]\_inverse\_transf.txt”.
* In the pop-up window “Saving in raw – select raw transformation file”, open the file “[TARGET img name]\_inverse\_transf RAW.txt”. “[TARGET img name]\_inverse\_transf RAW.txt” file should start with the lines “Width=3240” and “Height=810” for the demo.
* Copy the file “[TARGET img name]\_inverse\_transf RAW.txt to the directory “ClnColorDEMO>bUnwarpJ output>”, if necessary.