Project QuantiCellPhe-Overview

Ziqing Ye

Trcek Lab @ Johns Hopkins University

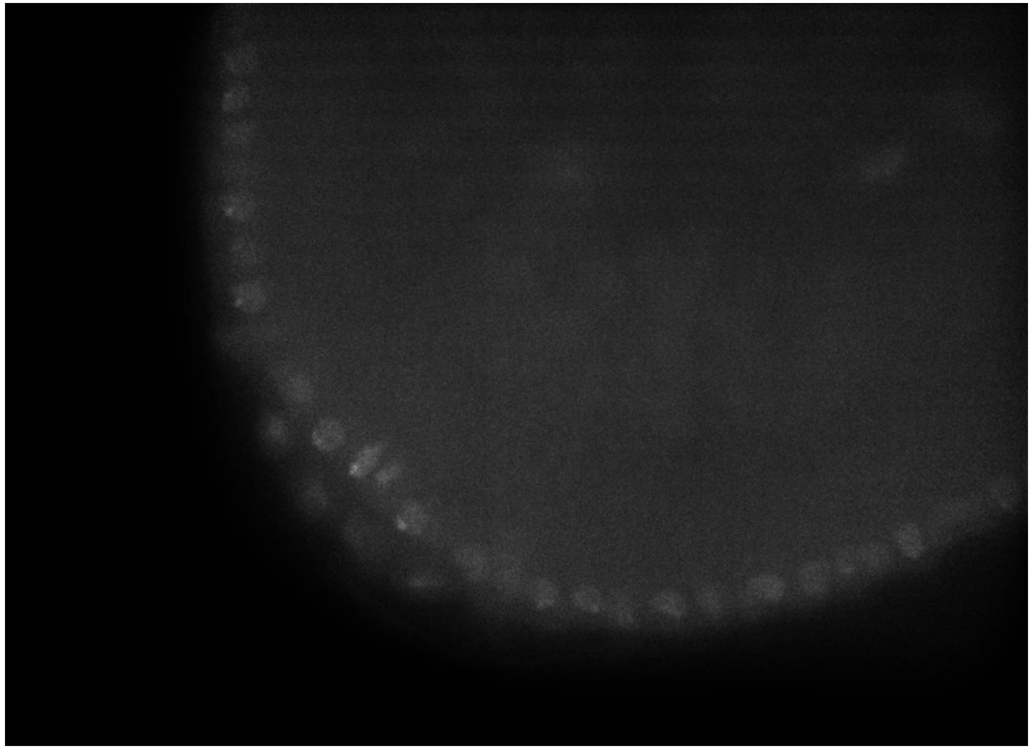
Overview

- Goal: quantify the number of pole cells that has sufficient granule expression (?)
- Mechanism:
  - Auto Gradient Normalization ©
  - Smart Thresholding ©
  - Connected component analysis & record nuclei coordinates
  - 3D Cell Count (and phenotype identification) ©

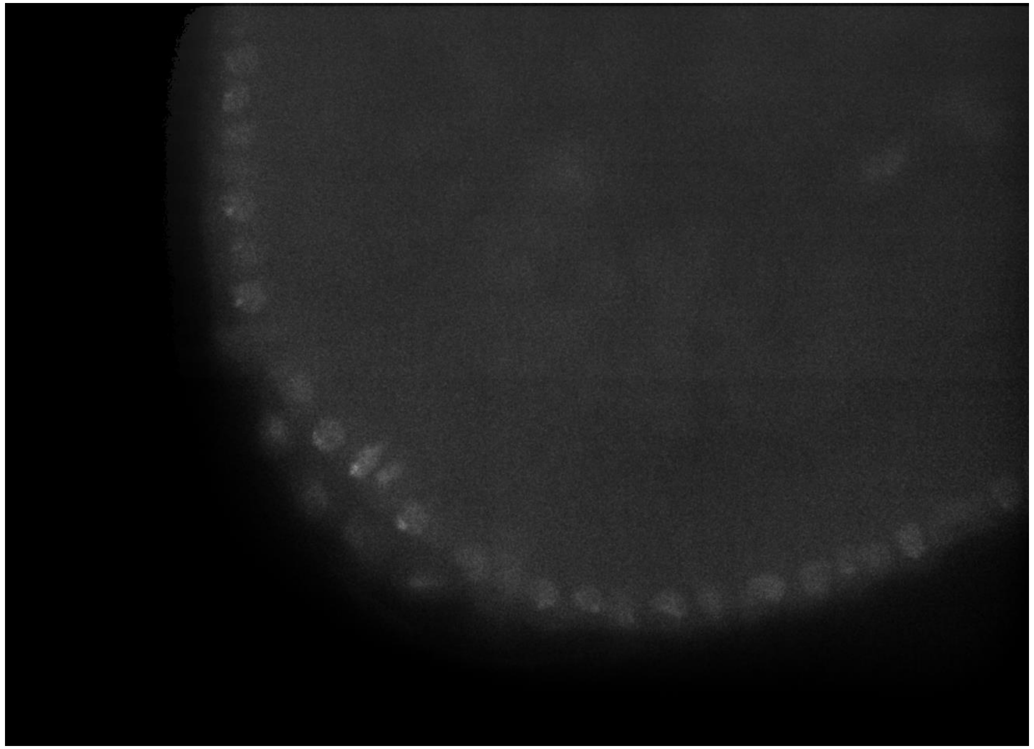
2D Nuclei identification

Auto Gradient Normalize

Sample image: Embryo_3_w2iSIM-405-DAPI_cmle-0034



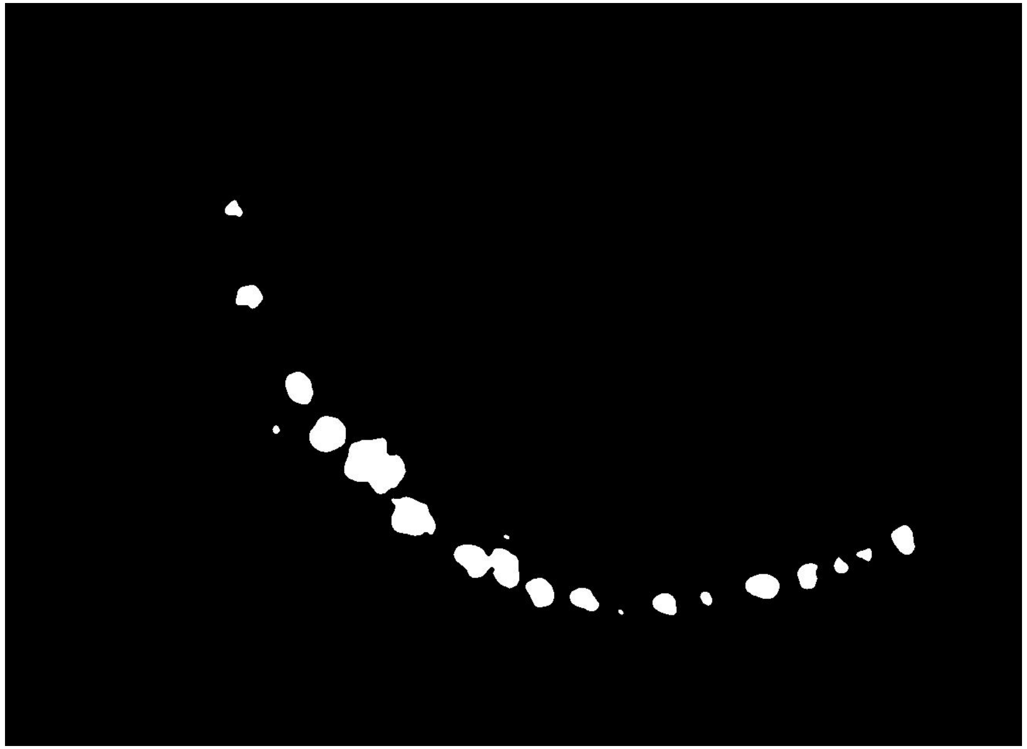
Original



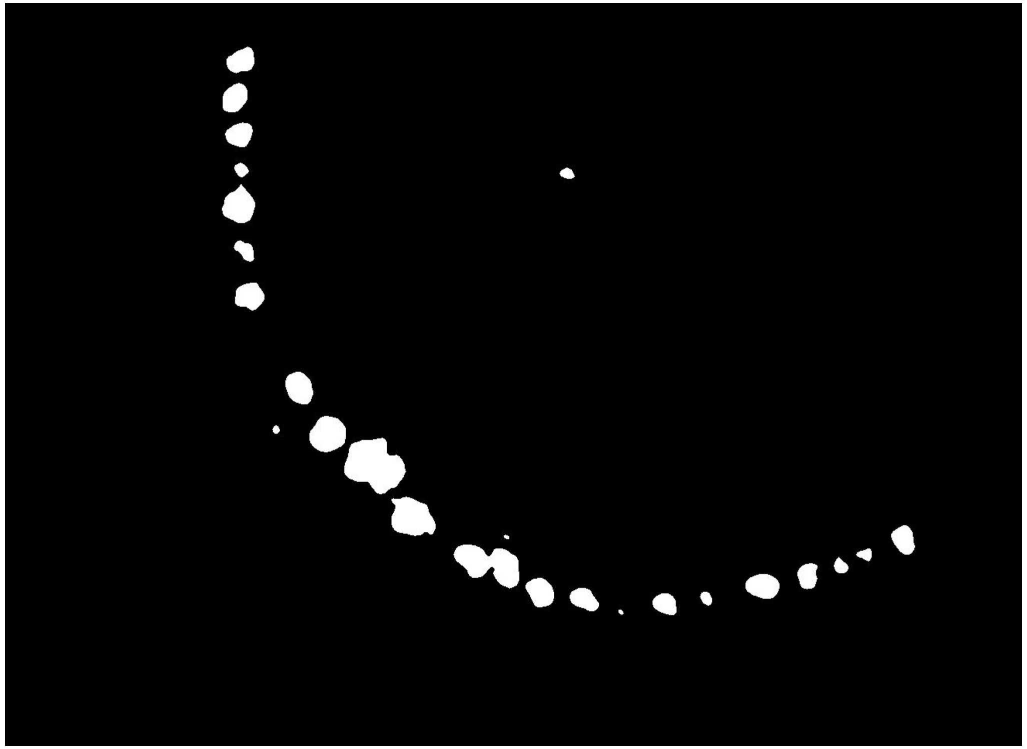
Gradient Normalized

Auto Gradient Normalize (Threshold result)

Sample image: Embryo_3_w2iSIM-405-DAPI_cmle-0034, threshold = 55



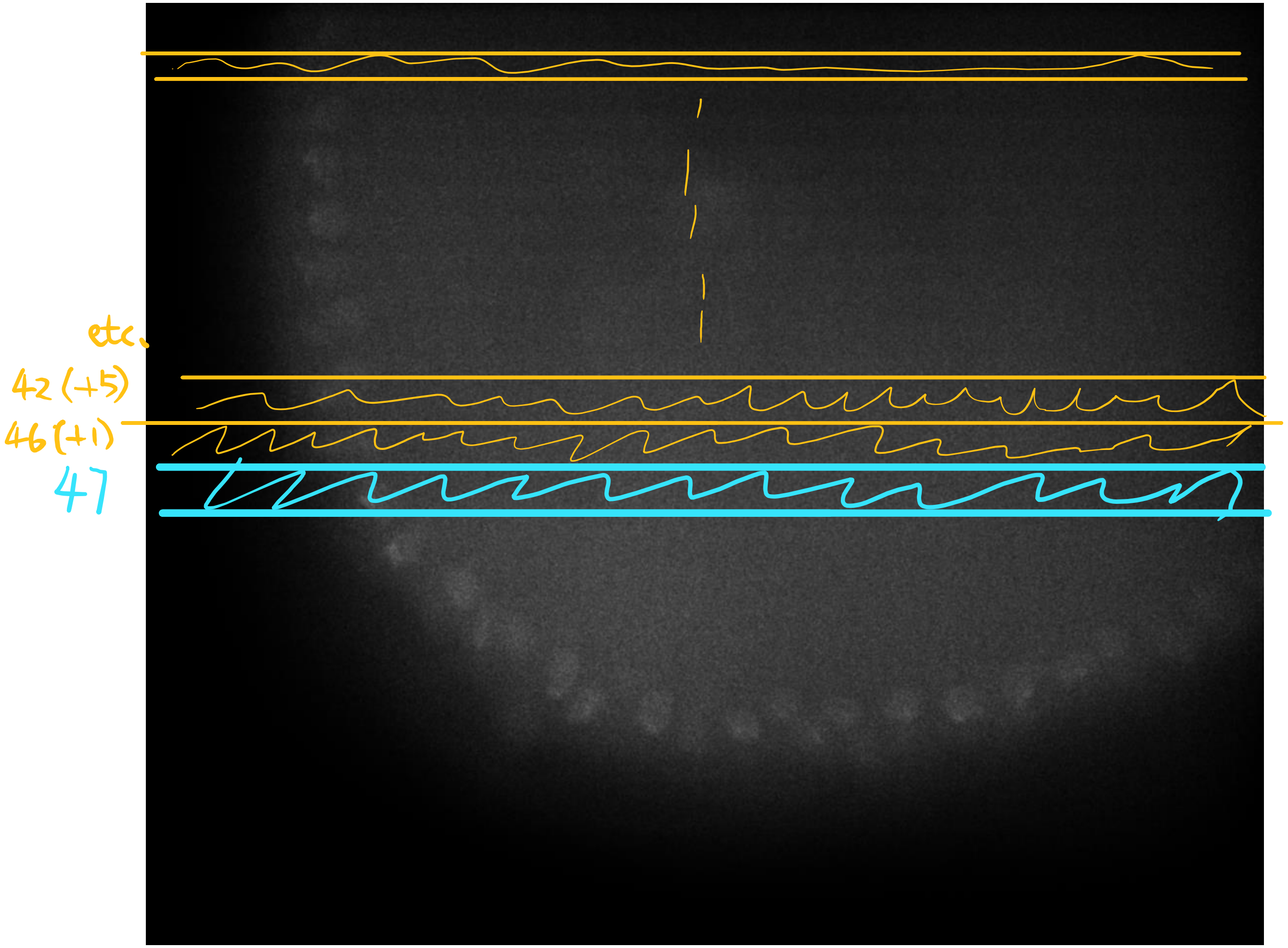
Original



Gradient Normalized

Auto Gradient Normalize – Mechanism

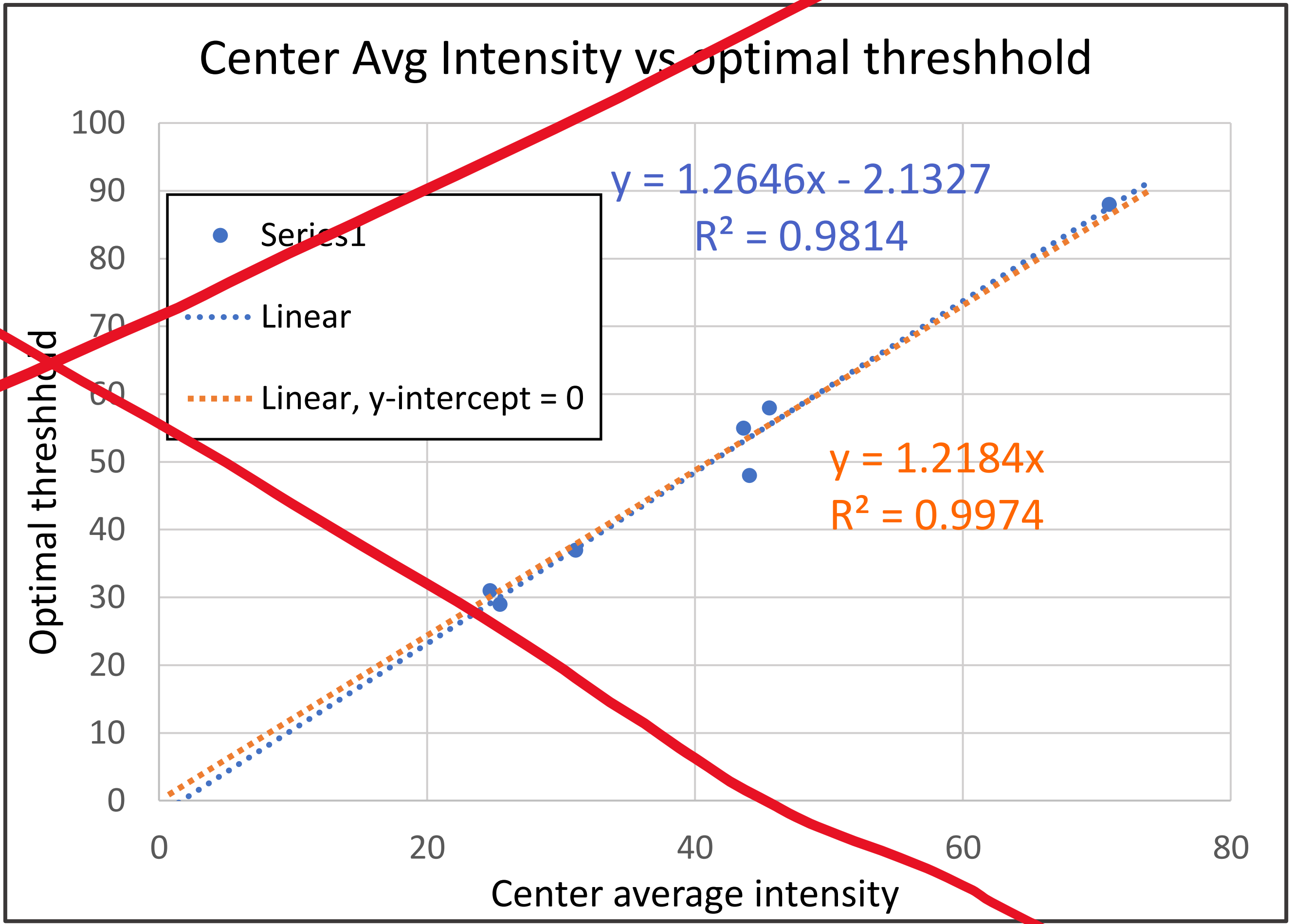
- Step 1: calculate the average pixel intensity of the center strip
- Step 2: adjust the intensity of each horizontal strip accordingly
- Same for vertical adjustments



I wish it was this simple
but nope

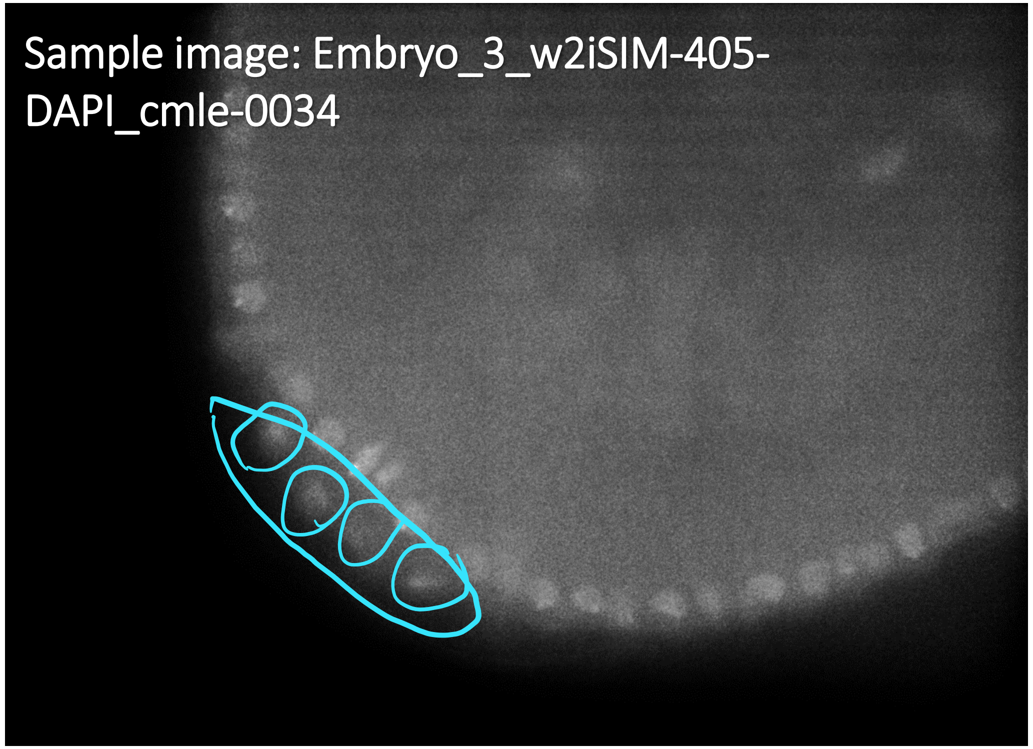
Determining the optimal threshold

- The optimal threshold for selected images across the spectrum of z values (slice 1, 34, 58, 84, 125, 128, 138) were determined through hand-tuning
- These, along with the “center average intensity” data calculated by the Auto Gradient Normalize algorithm, were plotted in Excel and underwent linear fit.
  - We got the optimal threshold as a function of center average intensity

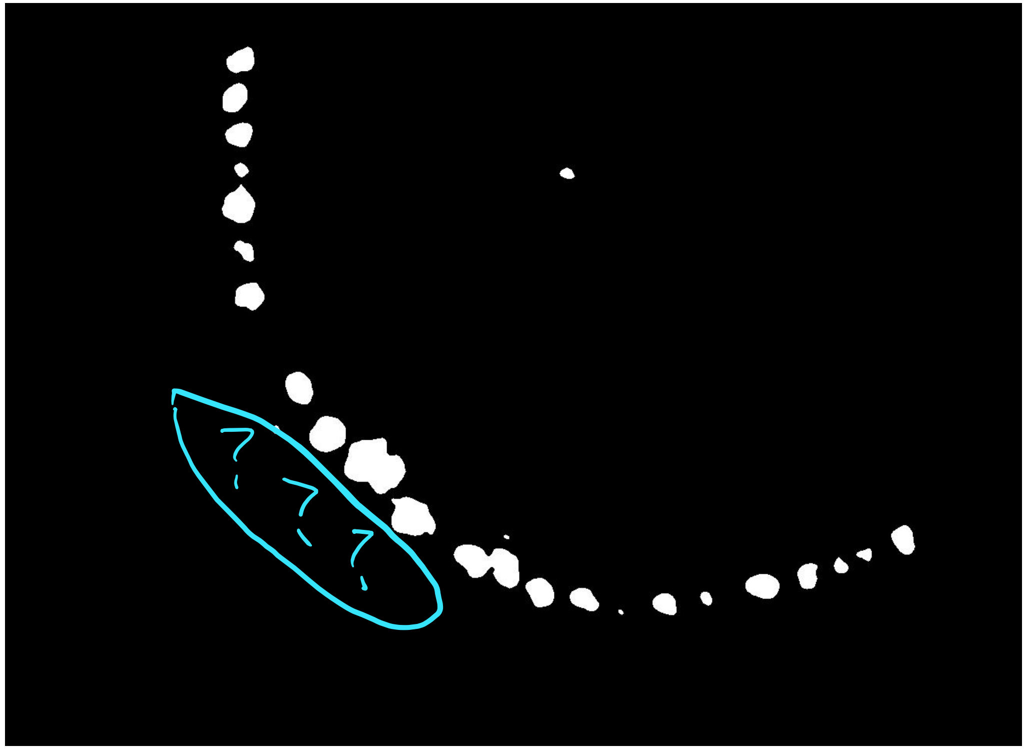


Smart Threshold – introduction

- Having a single threshold value for an entire slice did not work even with Gradient Normalize. (Surprise!!!!!)



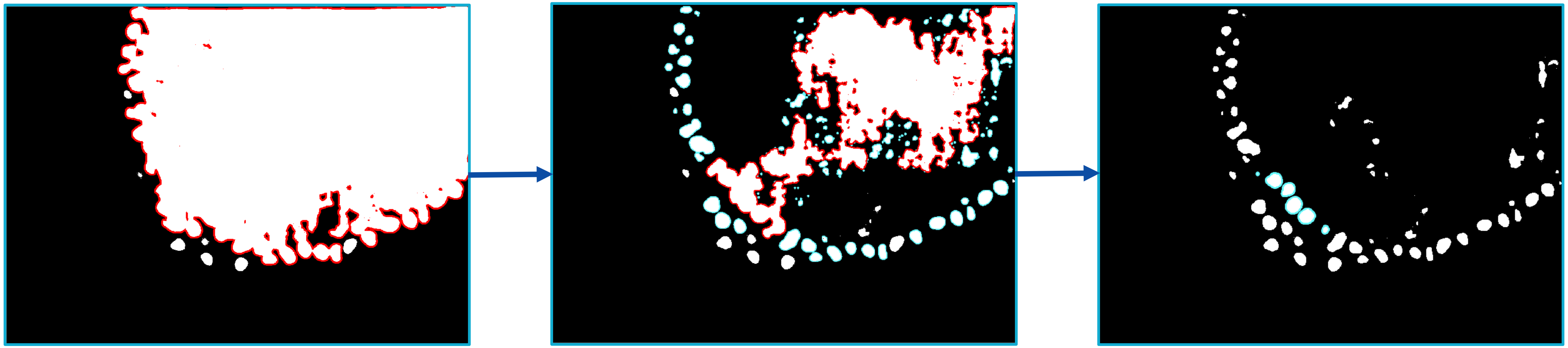
Original image



after threshold (single value, 55)

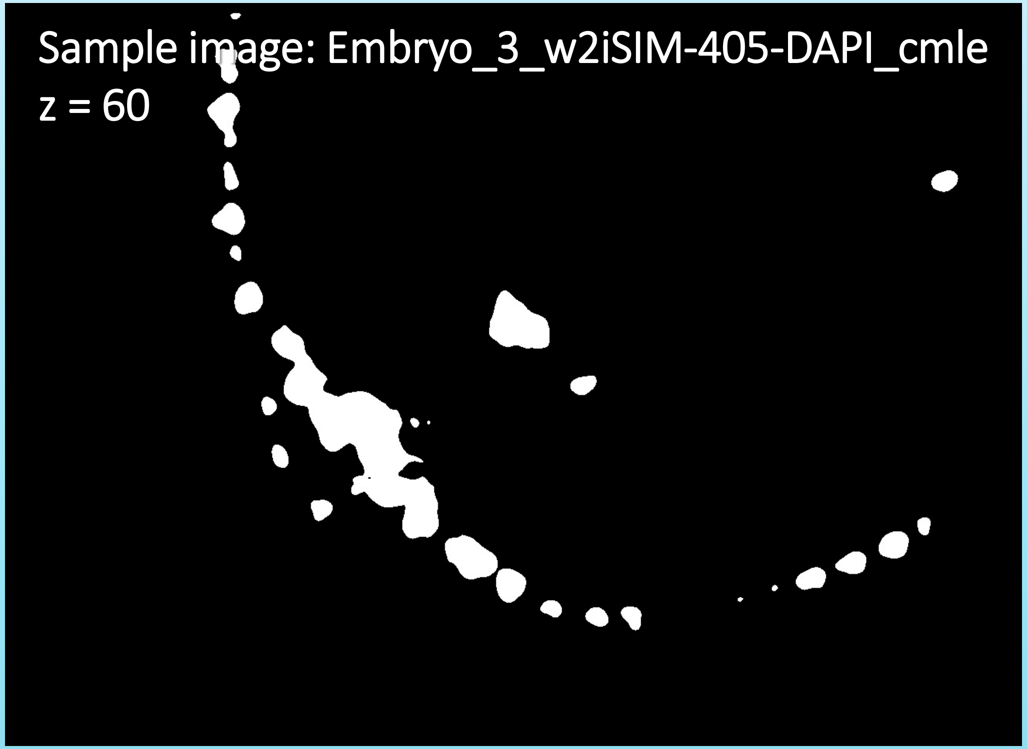
Smart Threshold – Mechanism

- Start from a lower threshold
- Identify continuous regions that are not properly segmented under the current threshold (based on the region size), then lower the pixel value of those regions, then re-iterate until all parts are properly segmented



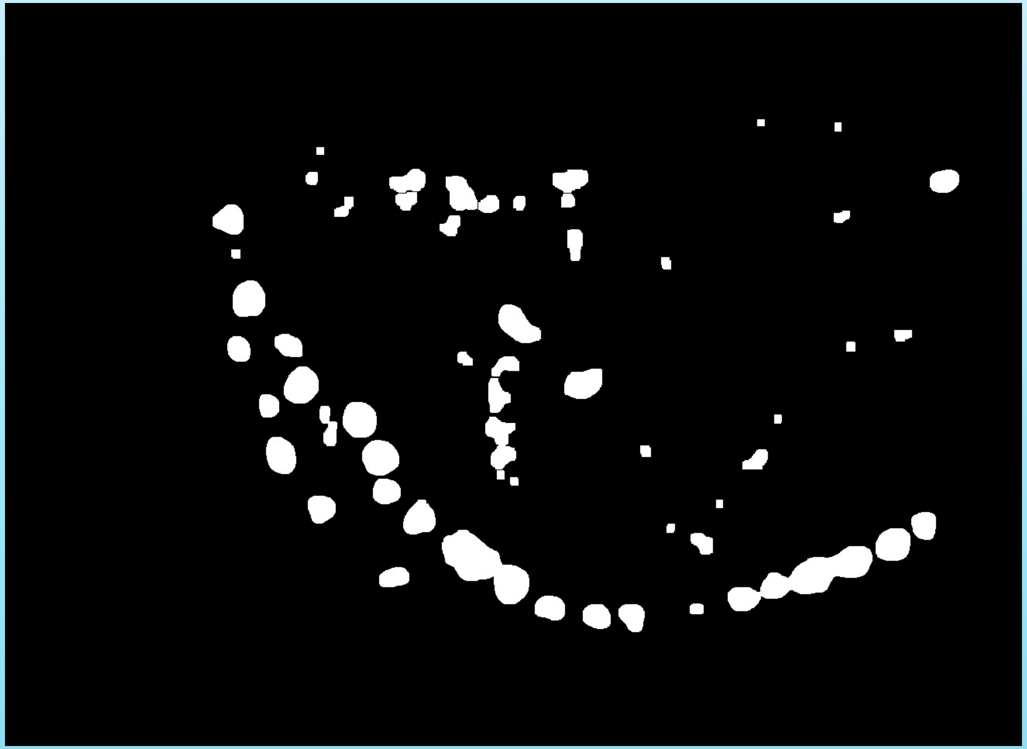
Smart Threshold – result

- It is not perfect, but it performs significantly better at the tip region where the phenotypic cells are located (compared to single value thresholding).



Sample image: Embryo_3_w2iSIM-405-DAPI_cmle
z = 60

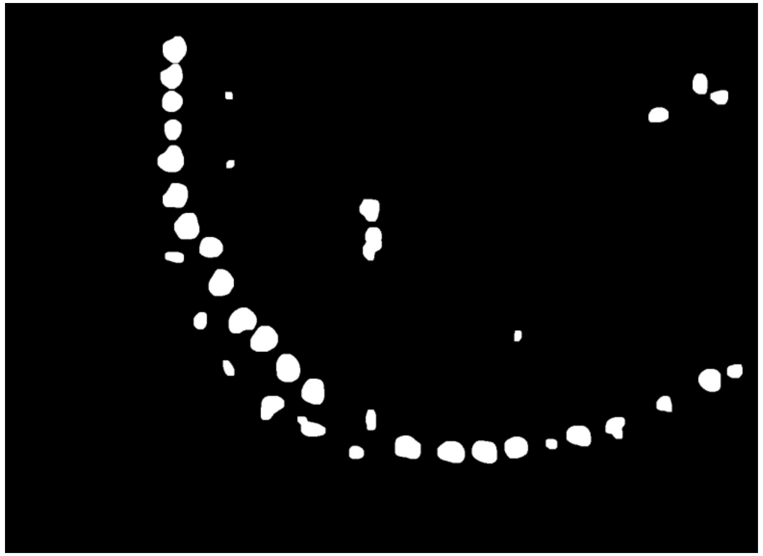
Original single-value threshold

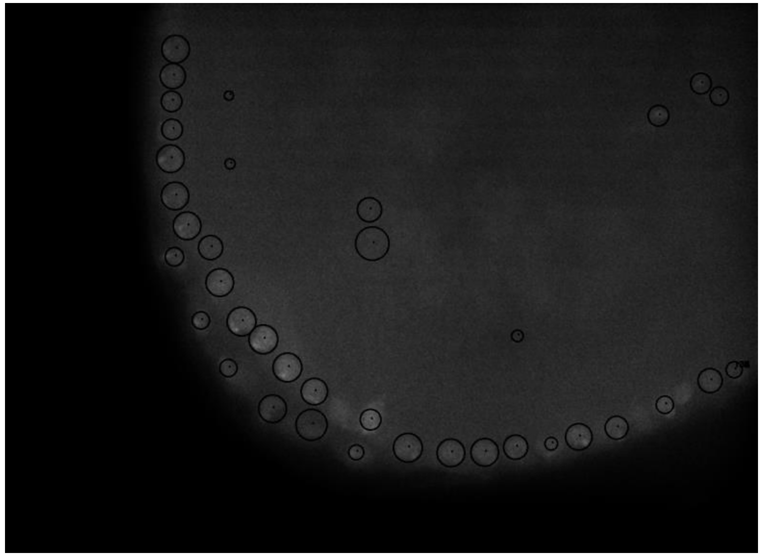


Smart Threshold (max iteration = 30)

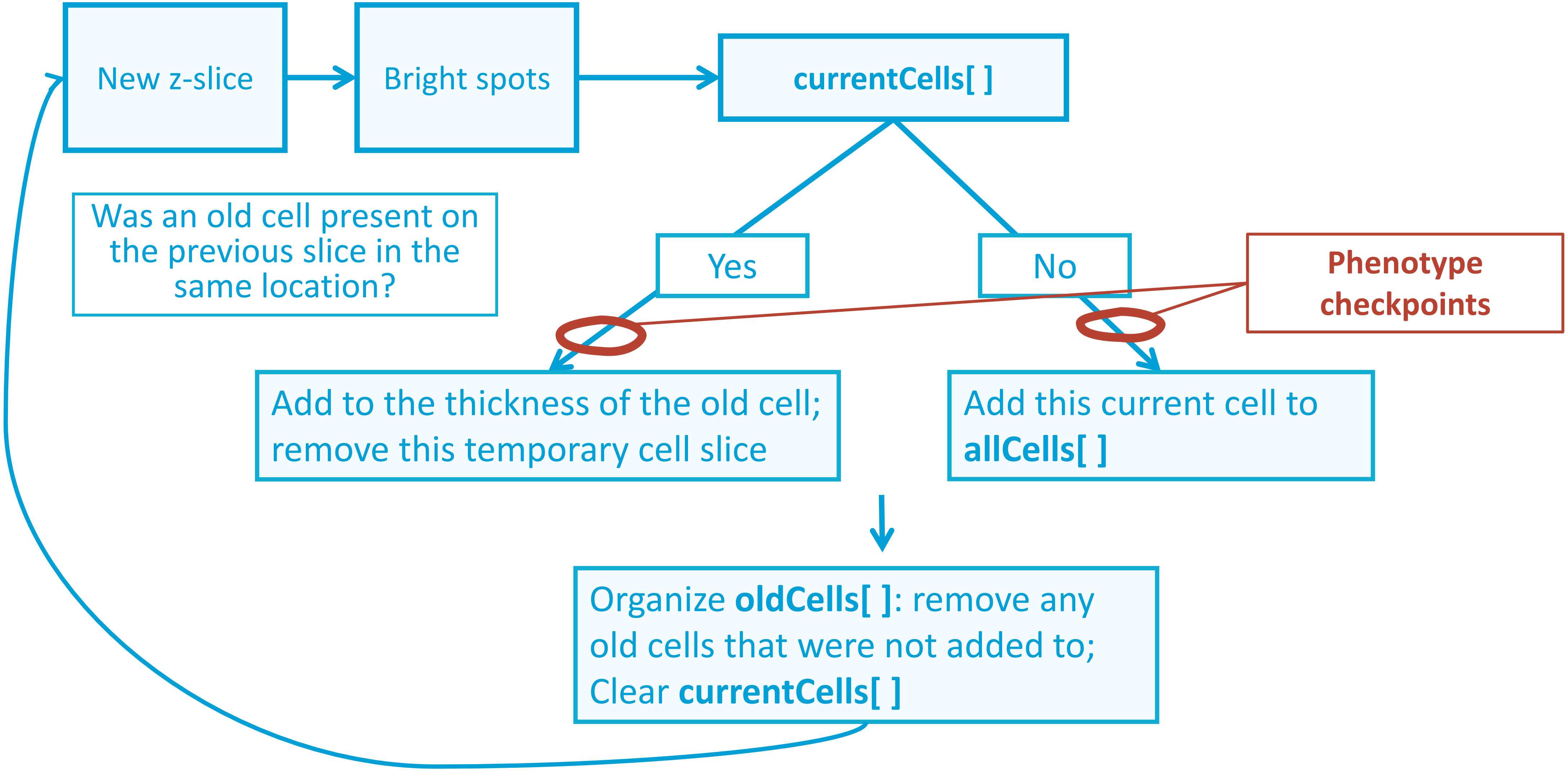
3D Nucleus Detection

- The stack is segmented one z-slice at a time
- Each distinct bright spot on the segmented image is identified as a temporary "cell": an object from the custom Cell class, which has x coordinate, y coordinate, z coordinate, radius, thickness, and phenotype parameters
- Multiple lists were created to store different kinds of Cell objects.
  - currentCells (all the "cells" on a single z slice, temporary)
  - allCells (all of the "permanent" cells)
  - oldCells (all of the permanent cells that were present on the previous z-slice)





3D Nucleus Detection



New z-slice

Bright spots

currentCells[ ]

Was an old cell present on the previous slice in the same location?

Yes

No

Phenotype checkpoints

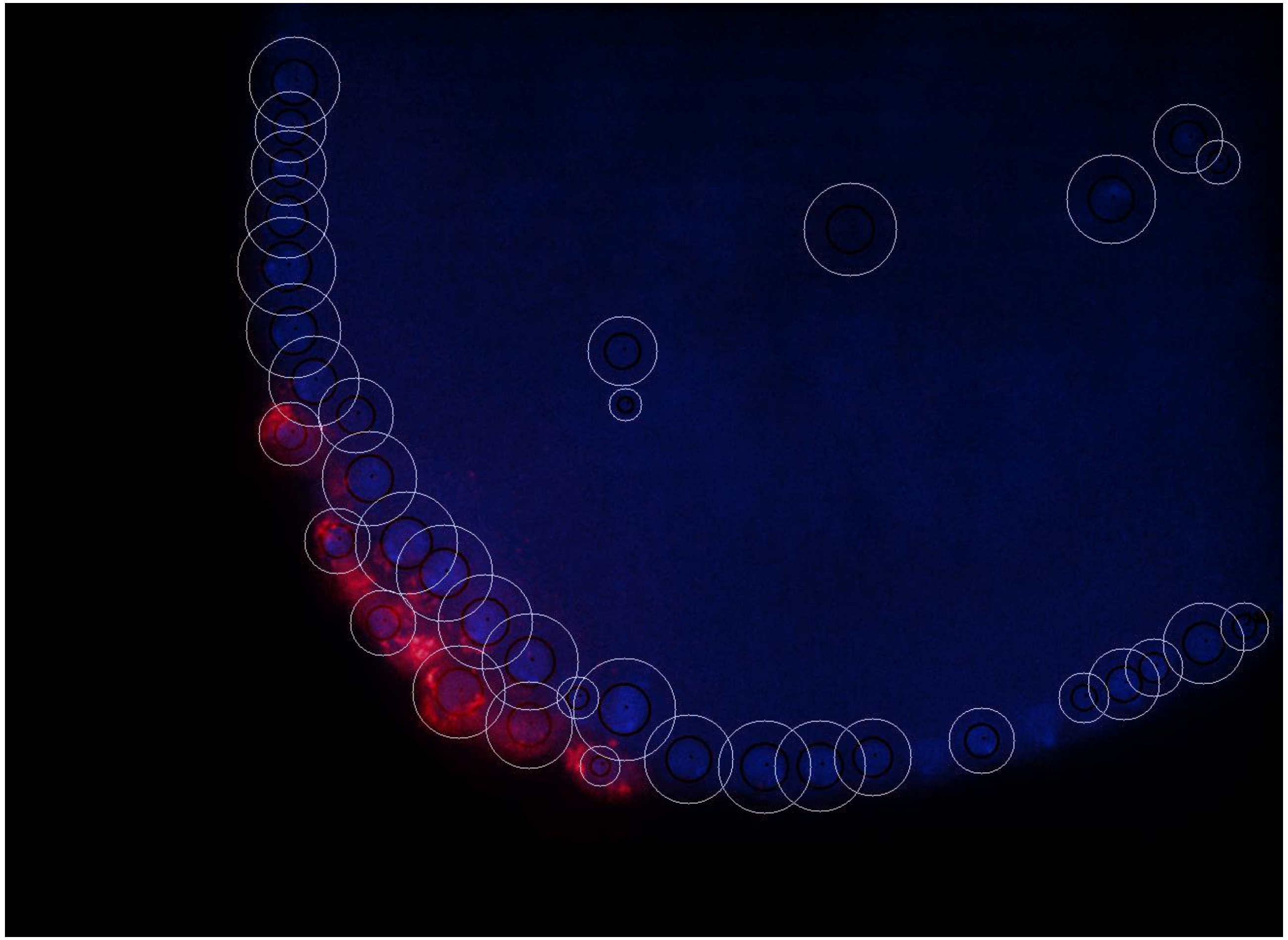
Add to the thickness of the old cell; remove this temporary cell slice

Add this current cell to **allCells[ ]**

Organize **oldCells[ ]**: remove any old cells that were not added to; Clear **currentCells[ ]**

Phenotype Check

- The phenotype detection area (white circle)’s radius is arbitrarily set to be twice the radius of the cell nucleus (black circle).
- The pixel intensity within the circles are obtained; if more than a certain percentage of the pixels have a pixel intensity of higher than a certain threshold, the cell object in that same location is marked as phenotypic



Acknowledgements

- All image used = copyright @ Trcek Lab @ Johns Hopkins University
- Referenced a glare detection algorithm by amittn on github for spot detection syntaxes, as well as various online library documentations
- Multiple python libraries were used, including opencv, the tifffile, xlsxwriter, numpy, etc.

Tatjana Trcek (PI)

Trcek Lab

Other members of Trcek Lab for providing imaging data & testing the program