Week of 10/2 Deliverables

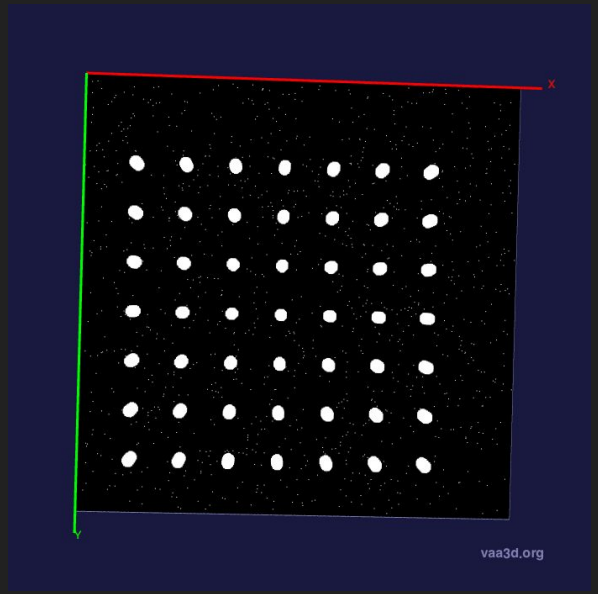
Team cobalt

Last week’s goals

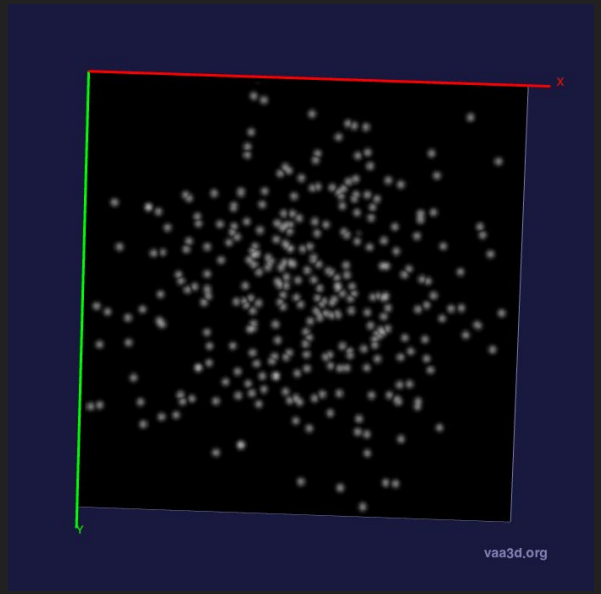
- Update sprint timeline
- Submitted AWS proposal
- Create simulated image generation tool
- Explore performance of watershed/blob detection qualitatively
  - ✓ Simple Blob Detector
  - ✓ Distance Transform for Cell Overlaps
- Identify 2 unsupervised cell detection algorithms from literature
- Profile ndreg code

Simulated data generator

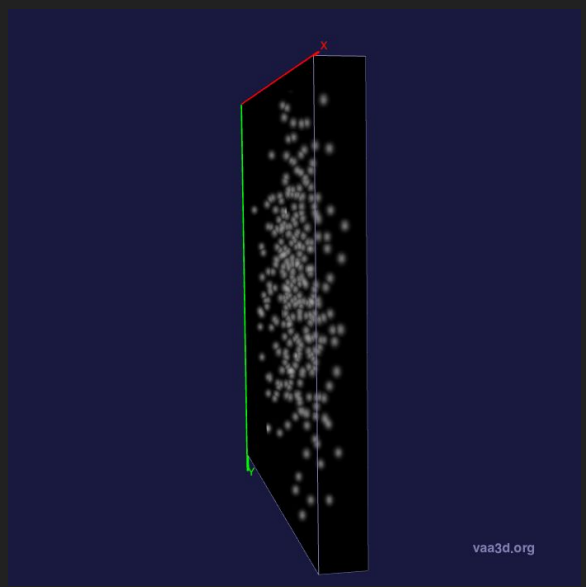
Built a tool that generates artificial cell counting datasets to use with our algorithms for validation and evaluation. Using this tool we created 8 datasets to be used as benchmarks. Capabilities of the tool are shown



Standard cell array with salt/pepper noise



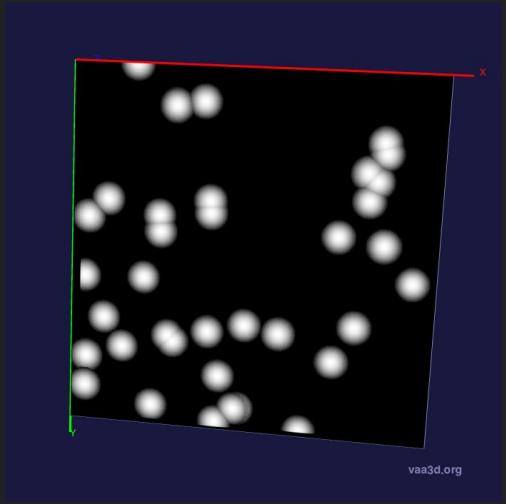
Randomly Normally distributed blurred cells



Z-axis view of the randomly distributed cells

Tool cont.

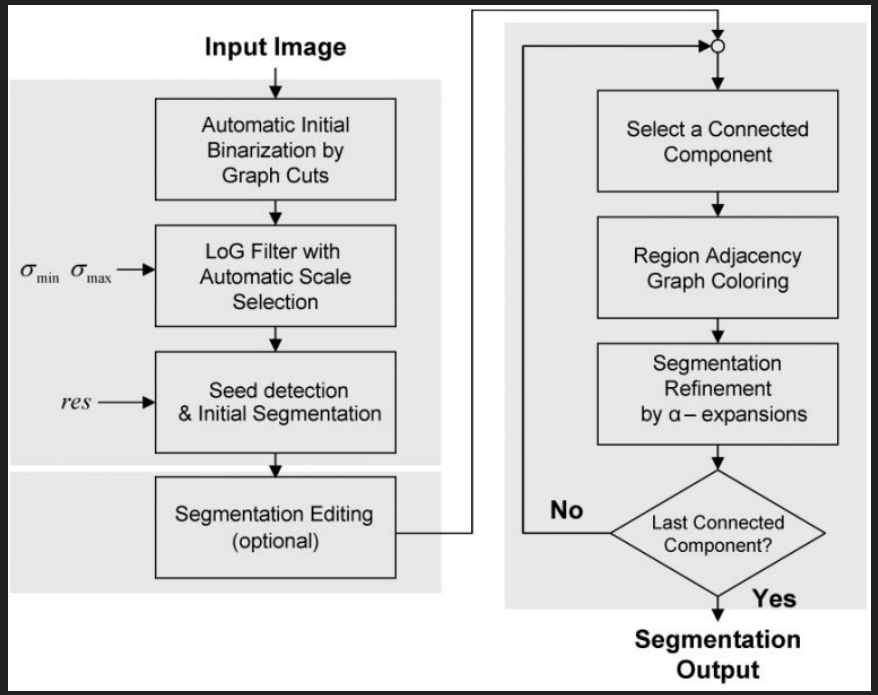
- We plan on using these images to evaluate and validate our algorithms. We can control how many cells are in each image, the overlapping of cells, their distribution, their shapes, the sizes of the volumes, etc (in future want to make the cells irregular shapes)
- if we go to deep learning then generating training data like this is invaluable.



- Code: https://github.com/NeuroDataDesign/clarity-f17s18/blob/master/source/jyim6/util/ImageGenerator.py
- Notebook (just my workspace): https://github.com/NeuroDataDesign/clarity-f17s18/blob/master/source/jyim6/Simulated%20data.ipynb

Unsupervised algorithm for cell detection

- Yousef Al-Kofahi et al is a survey of the automated cell-counting algorithms suggest that FARSIGHT's nuclei segmentation algorithm performed the best out of the modern algorithms. But still not good enough to replace supervised methods.
- The algorithm is detailed on the FARSIGHT website and in the paper although the paper only details the 2D variant.
- The algorithm exists in FARSIGHT toolkit. The algorithm in the paper is written in C++. We'll try to at least get it running in C++ or FARSIGHT and then port it to python if it's not too bad.

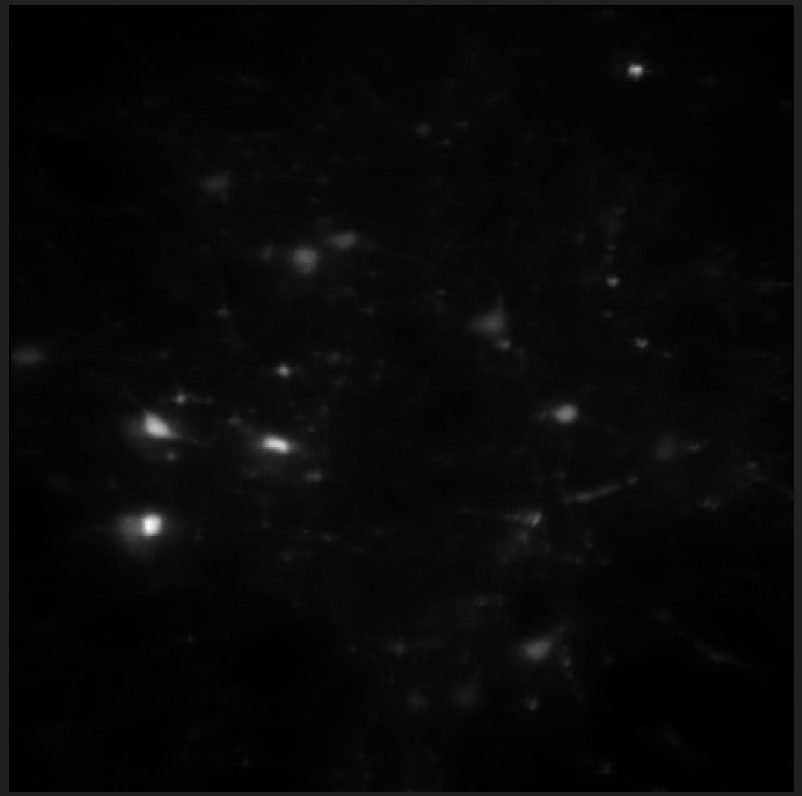


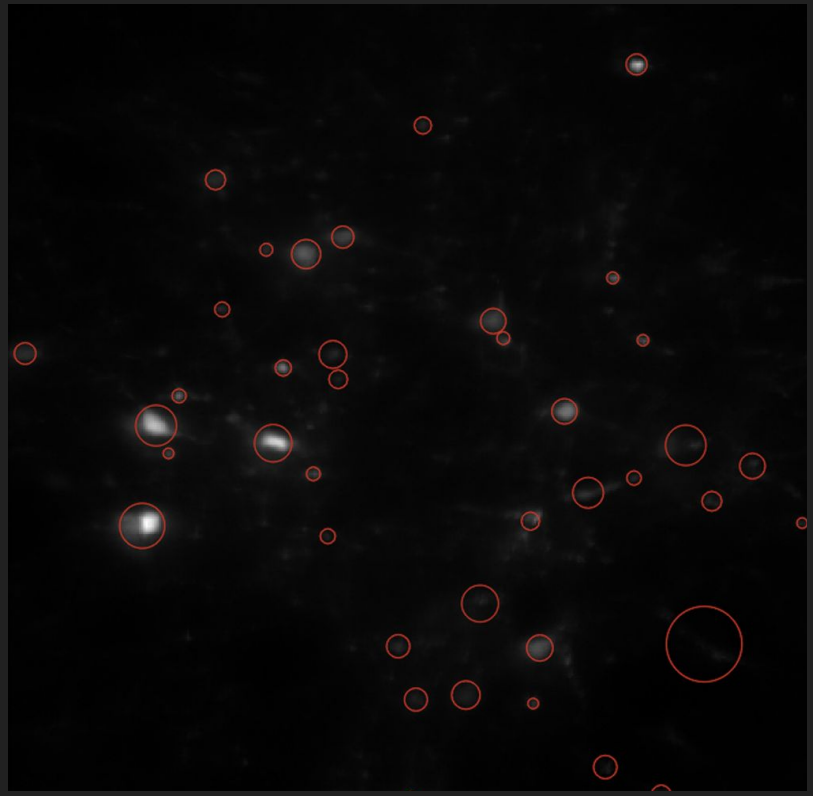
Flowchart of algorithm for 2D histology images

Simple Blob Detector

- Output of OpenCV's SimpleBlobDetector which is implemented based on
  - Watershed algorithm
  - OTSU's binarization threshold
  - Scale Invariant Feature Transform (SIFT)
- False positives are very less
- Can get very good results if combined with 3D reconstruction algorithms

Simple Blob Detector

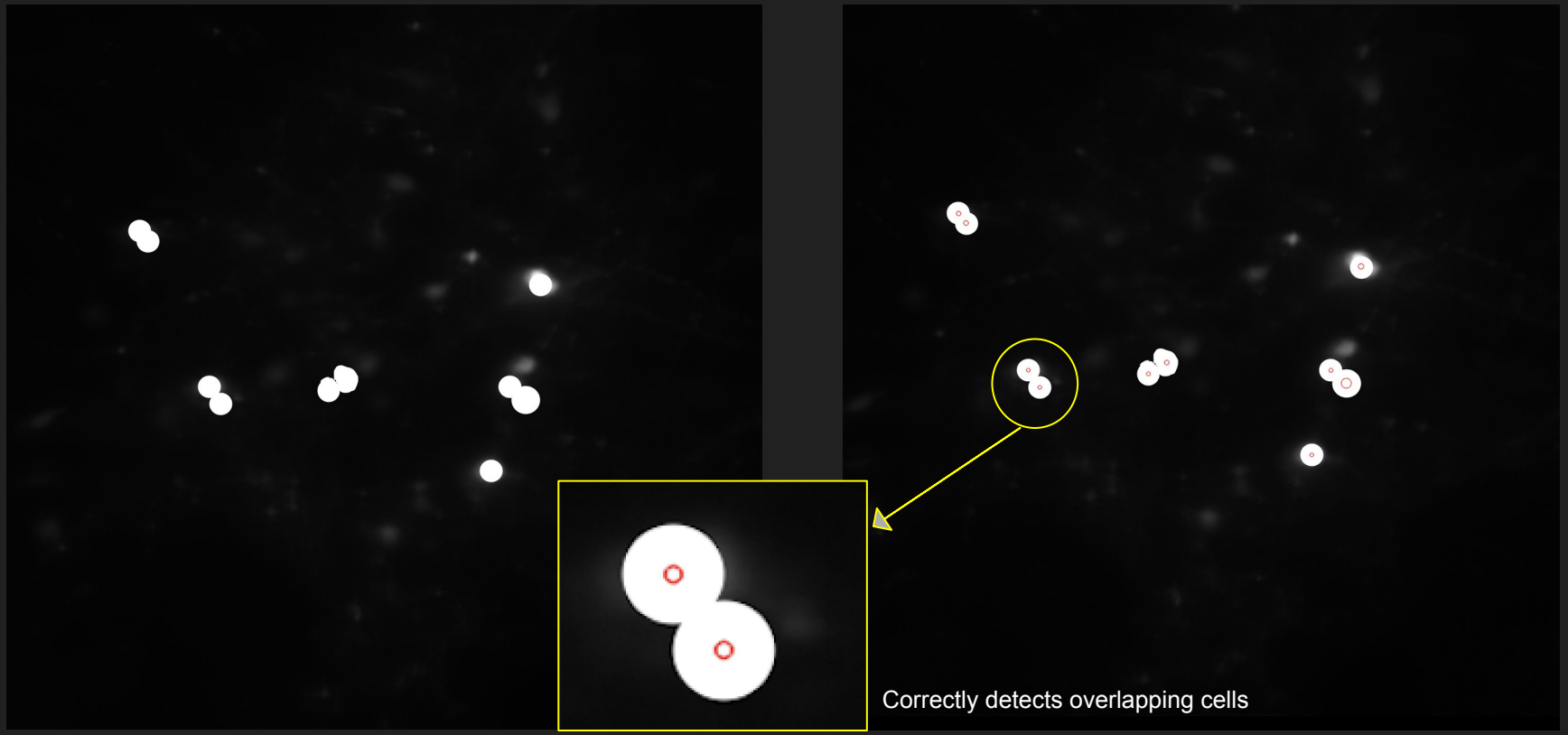




Distance transform for overlapping cells

- **Morphological opening** is used to create sure background area and **distance transform** is performed for sure foreground area.
- This removes ambiguity between overlapping cells and identifies them individually

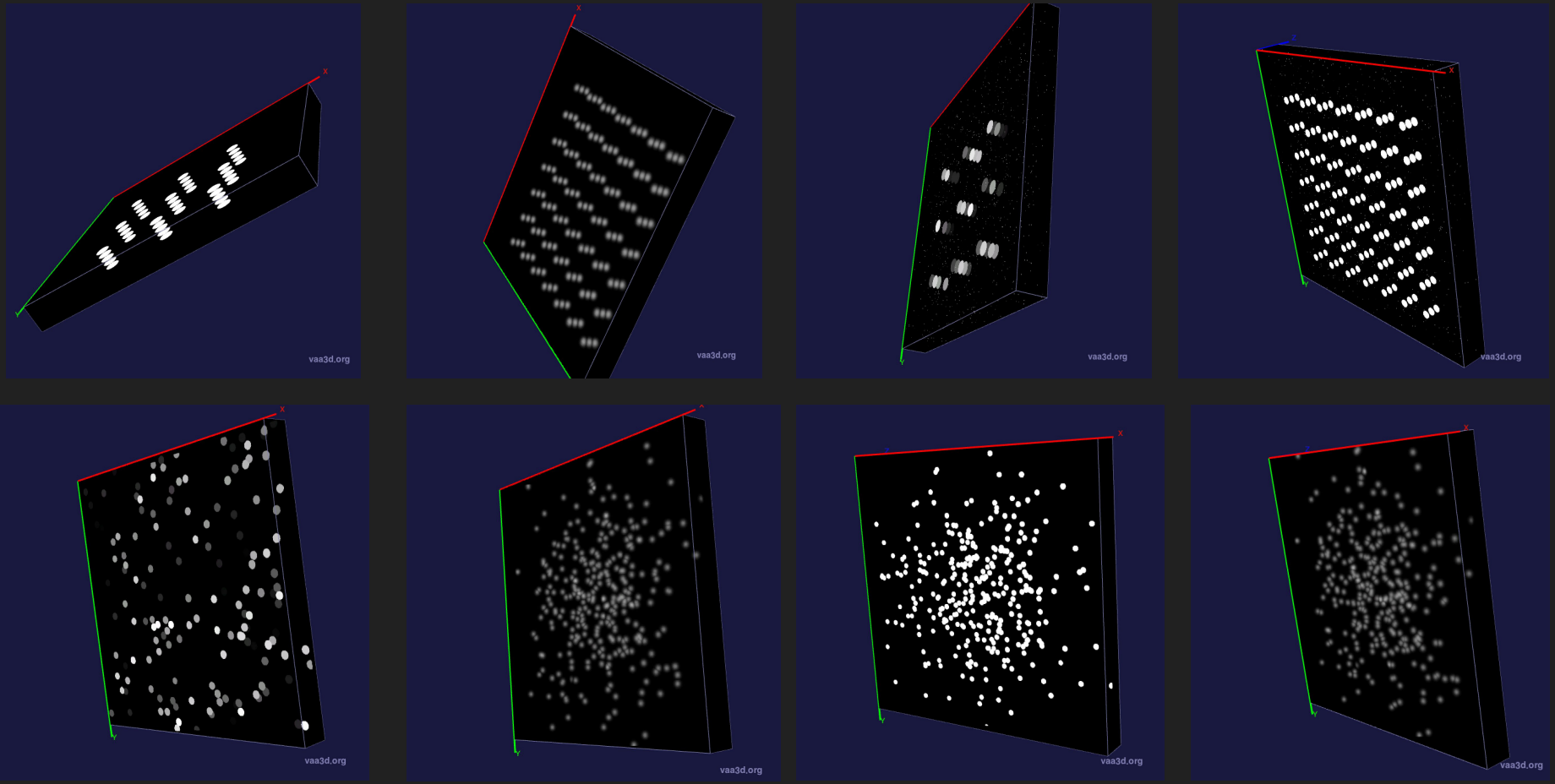
Distance transform for overlapping cells



Survey of Unsupervised Algorithms

- Surveyed the performance following unsupervised algorithms
  - Watershed Algorithm
  - Laplacian of Gaussian (LoG)
  - Difference of Gaussian (DoG)
  - Determinant of Hessian (DoH)
- The metrics taken were
  - Time taken
  - Accuracy
  - F1 Measure
- Dataset - 8 different types of images that was obtained from the simulation tool

Survey of Unsupervised Algorithms - Dataset



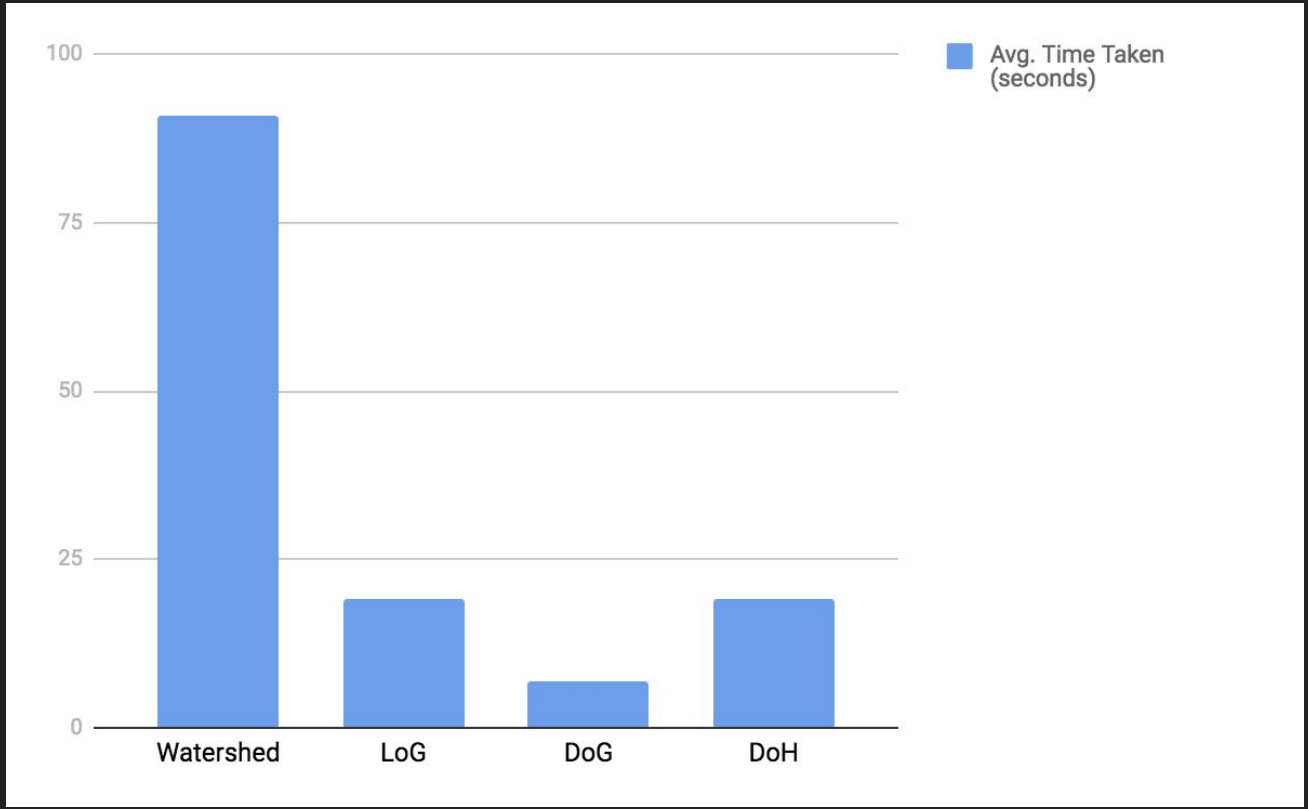
Algorithms

- **Watershed Algorithm:**
  - Starting from user-defined markers, the watershed algorithm treats pixels values as a local topography (elevation).
  - The algorithm floods basins from the markers, until basins attributed to different markers meet on watershed lines.
- **Laplacian of Gaussian (LoG):**
  - The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image.
  - The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection (see zero crossing edge detectors).
  - The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise, and hence the two variants will be described together here.

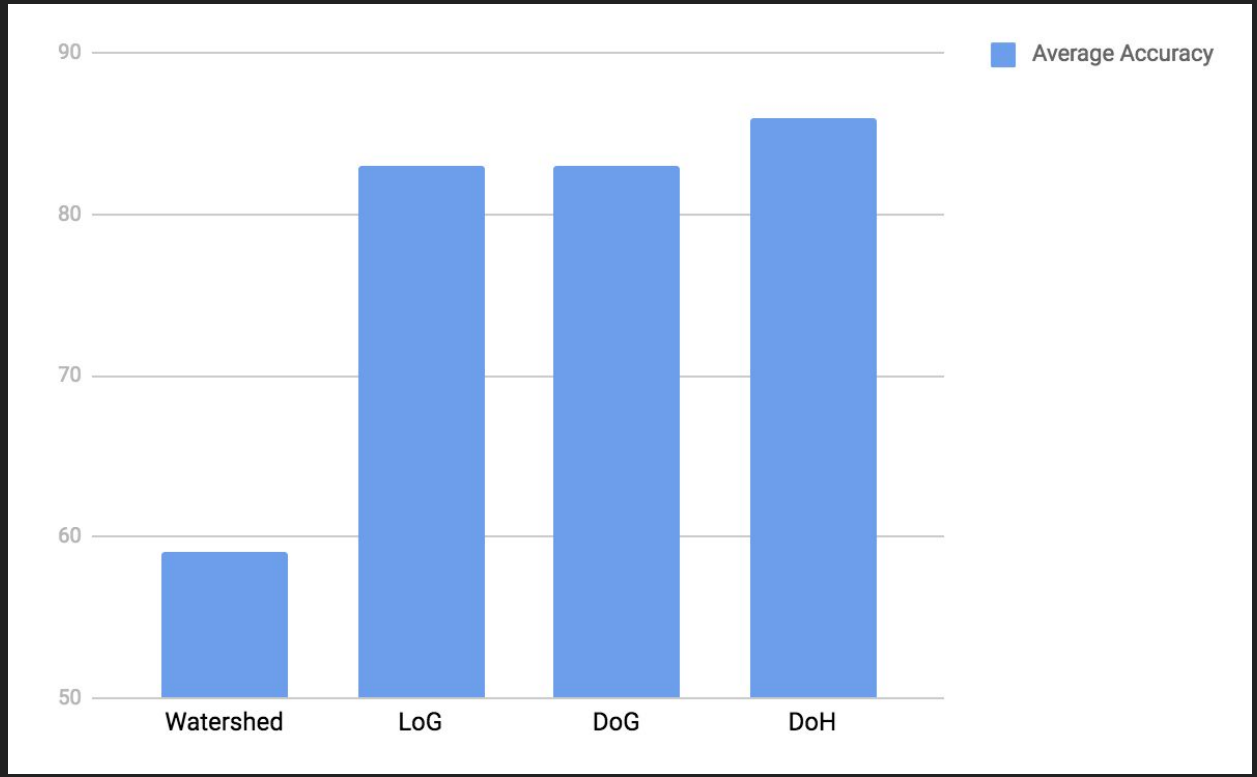
Algorithms

- **Difference of Gaussian (DoG):**
  - Difference of Gaussians is a feature enhancement algorithm that involves the subtraction of one blurred version of an original image from another, less blurred version of the original. In the simple case of grayscale images, the blurred images are obtained by convolving the original grayscale images with Gaussian kernels having differing standard deviations
- **Determinant of Hessian (DoH):**
  - In mathematics, the Hessian matrix or Hessian is a square matrix of second-order partial derivatives of a scalar-valued function, or scalar field.
  - The determinant of the above matrix is also sometimes referred to as the Hessian which is used for blob detection

Time Taken



Accuracy



F1 Measure

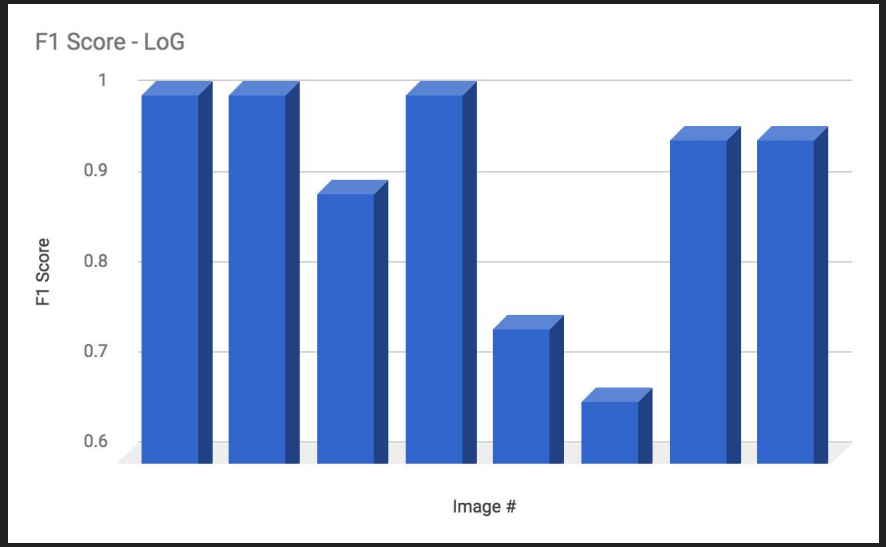
$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

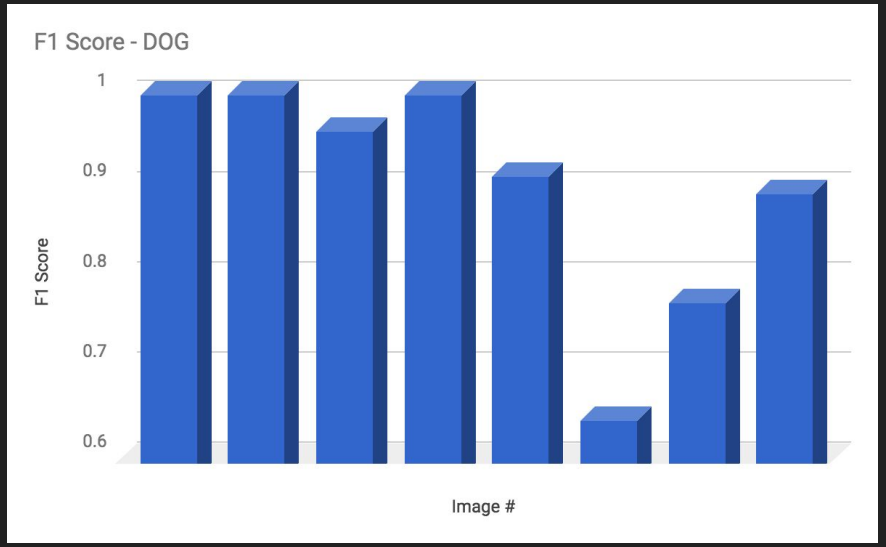
$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

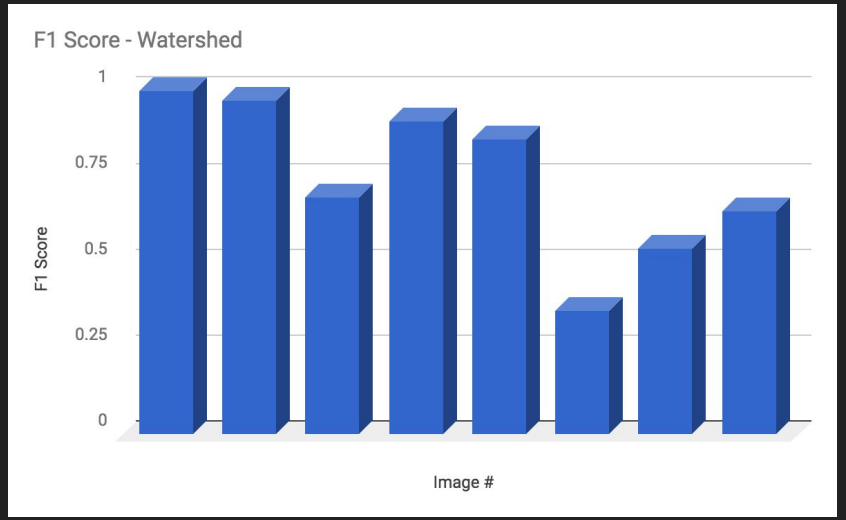
tp - true positives, fp - false positives, fn - false negatives

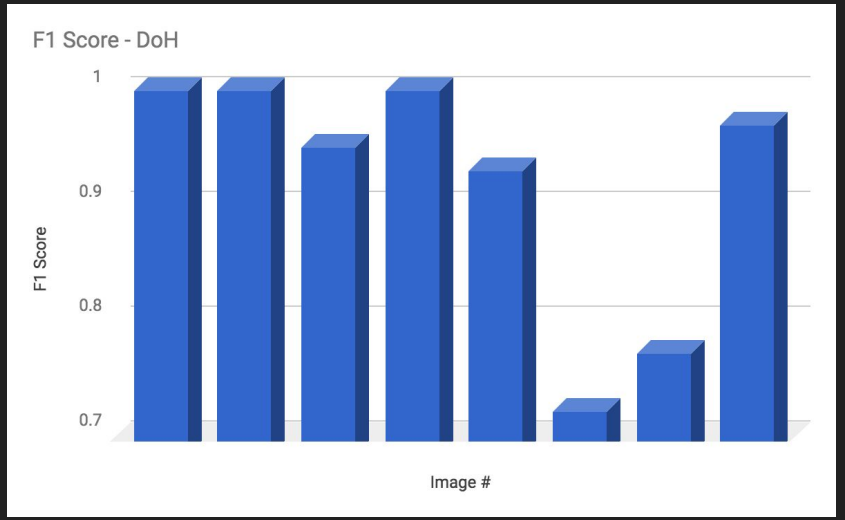
F1 Measure


F1 Score - LoG
1
0.9
0.8
0.7
0.6
F1 Score
Image #


F1 Score - DOG
1
0.9
0.8
0.7
0.6
F1 Score
Image #

F1 Measure


F1 Score - Watershed
1
0.75
0.5
0.25
0
F1 Score
Image #


F1 Score - DoH
1
0.9
0.8
0.7
F1 Score
Image #

NDREG Profiling - 50 um ARA registration of s3617

- notebook of profiling
- registration code
- Wasn't able to profile lddmm code
  - Will try running it on machine with more RAM

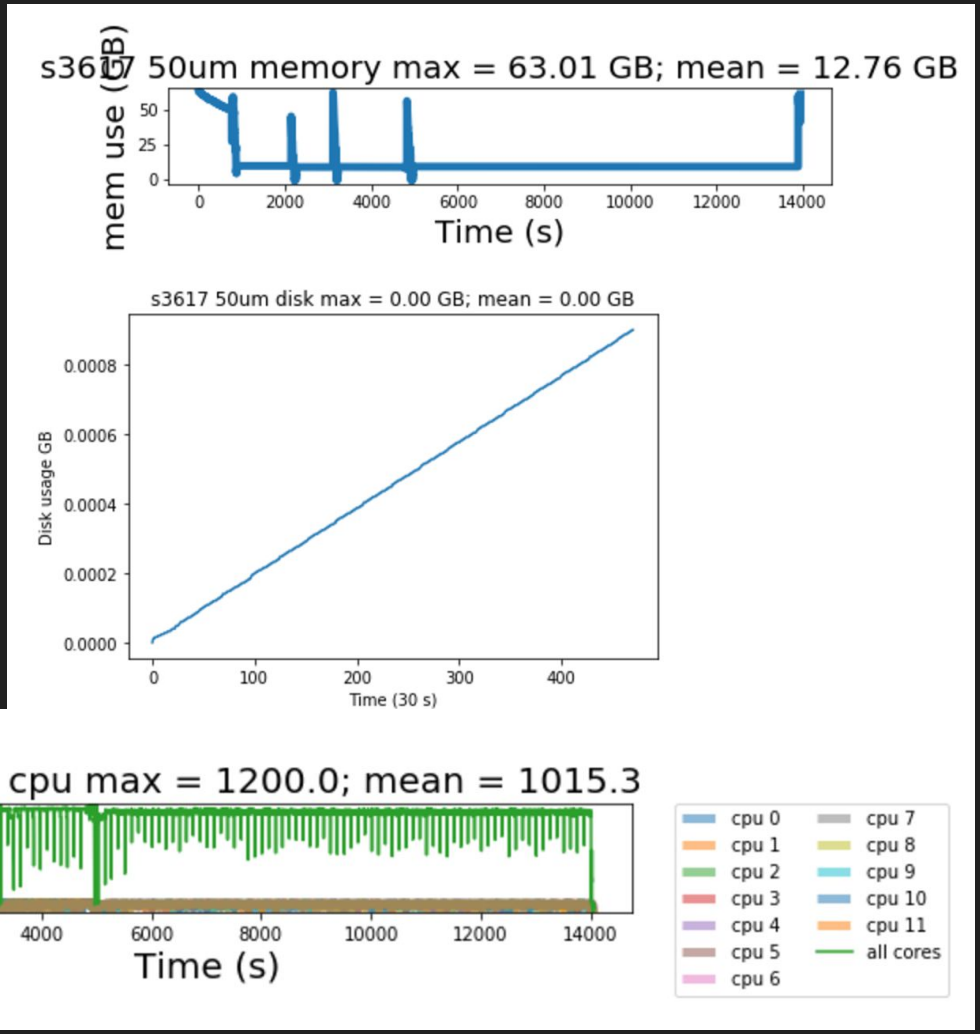
```
Step 0: alpha=0.2, beta=0.05, scale=1.0
Error: Metamorphosis did not terminate normally.

itk::MemoryAllocationError (0x7fc768000940)
Location: "TElement* itk::ImportImageContainer<TElementIdentifier, TElement>::AllocateElements(itk::ImportImageContainer<TElementIde
ntifier, TElement>::ElementIdentifier, bool) const [with TElementIdentifier = long unsigned int; TElement = itk::Vector<double, 3u>;
 itk::ImportImageContainer<TElementIdentifier, TElement>::ElementIdentifier = long unsigned int]"
File: /usr/include/ITK-4.9/itkImportImageContainer.hxx
Line: 199
Description: Failed to allocate memory for image.


Traceback (most recent call last):
  File "reg_s3617_50um.py", line 201, in <module>
    scaleList = 1.0, useMI=True, iterations=100, verbose=True)
  File "/root/.local/lib/python2.7/site-packages/ndreg-0.0.1-py2.7.egg/ndreg/ndreg.py", line 1329, in imgMetamorphosisComposite
    outDirPath=stepDirPath)
  File "/root/.local/lib/python2.7/site-packages/ndreg-0.0.1-py2.7.egg/ndreg/ndreg.py", line 1254, in imgMetamorphosis
    (returnValue, logText) = run(command, verbose=verbose)
  File "/root/.local/lib/python2.7/site-packages/ndreg-0.0.1-py2.7.egg/ndreg/ndreg.py", line 85, in run
    if checkReturnValue and (returnValue != 0): raise Exception(outText)
```

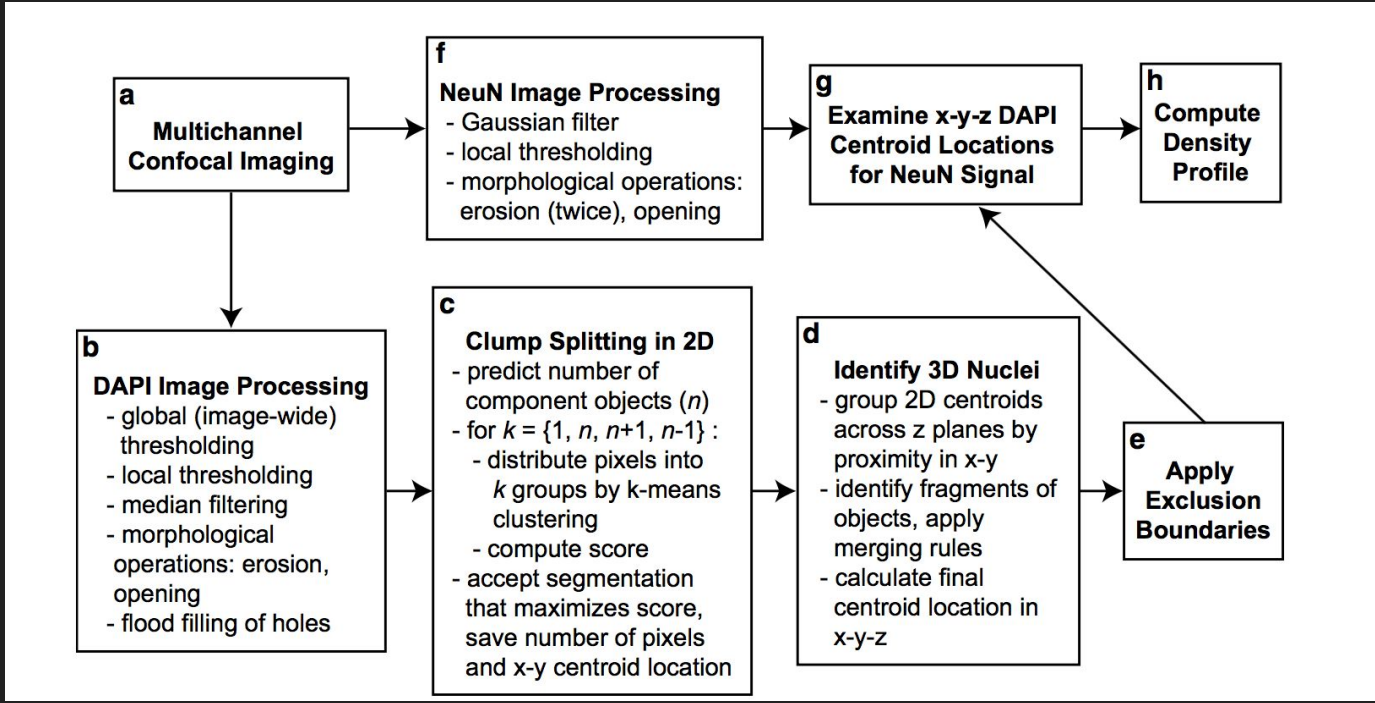
NDREG Profiling - graphs

- Memory Usage
- Disk Usage
- CPU Usage

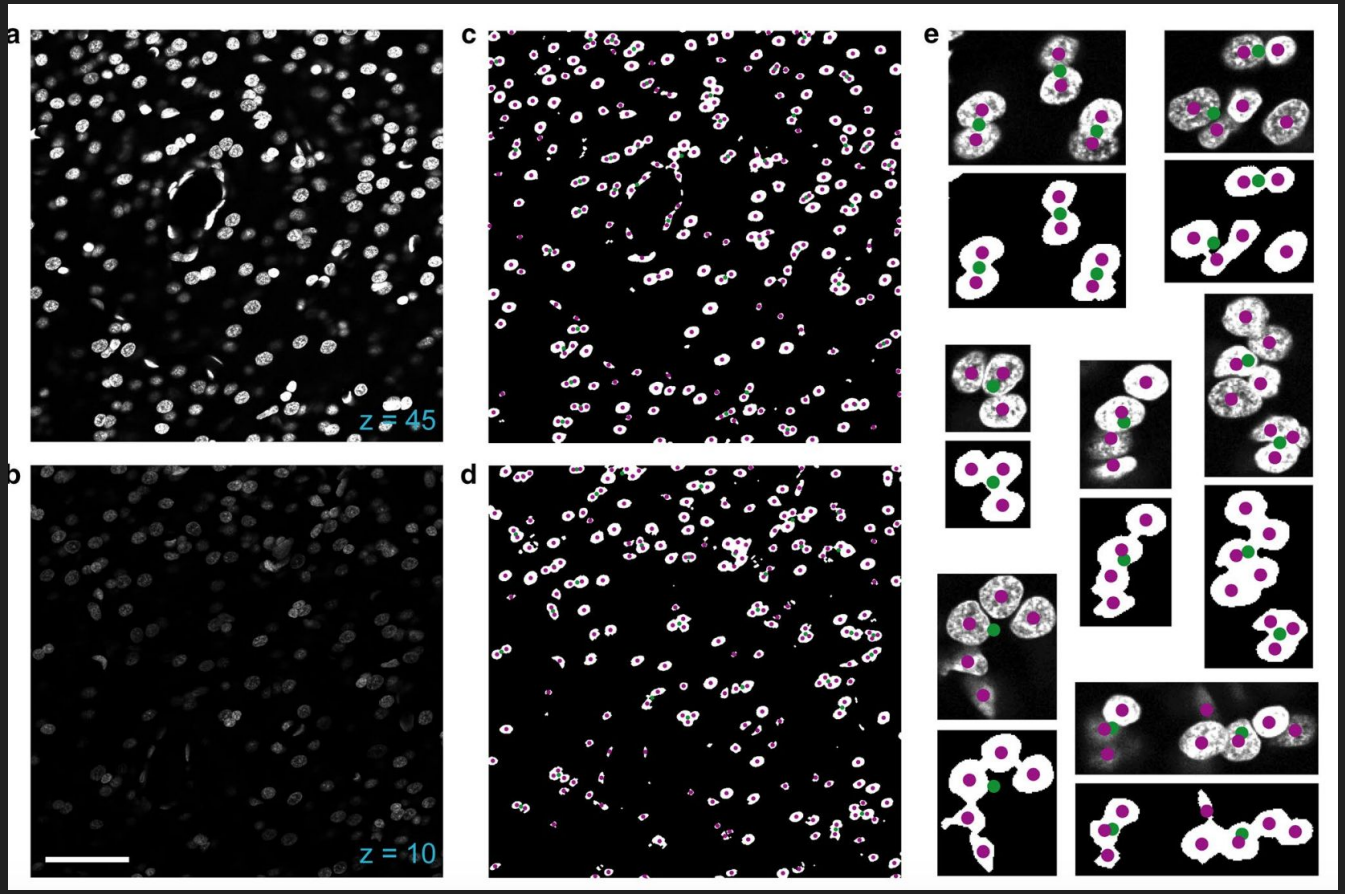


Quantification of neuronal density across cortical depth using automated 3D analysis of confocal image stacks

- Link
- Measuring densities of immunolabeled neurons across cortical layers



Continued



Next week

- Follow up with brian to get labelled COLM data
- Write detailed pseudocode for 3 unsupervised methods from literature (see algorithms.md)
- Quantitatively evaluate Thy1eYFP_Control_9 50um registration to ARA using manual fiducials
    - Compute mean-squared error