Week of 10/9 Deliverables

Team cobalt

Last week's goals

- ✓ Jupyter notebook for summing intensity by region of ARA
- ✓ Create python package for blob detection metrics
- ✓ Write pseudocode for 2 unsupervised algorithms from literature
  - ✓ Ensemble methods
    - ✓ 2D Analysis + 2D Pseudocode
    - ✓ 3D Reconstruction
    - ✓ Code
  - ✓ FARSIGHT
    - ✓ A benchmarks.md thing

Summing intensity by region

- Simple proxy for cell count and can be used as a sanity check
- Notebook here

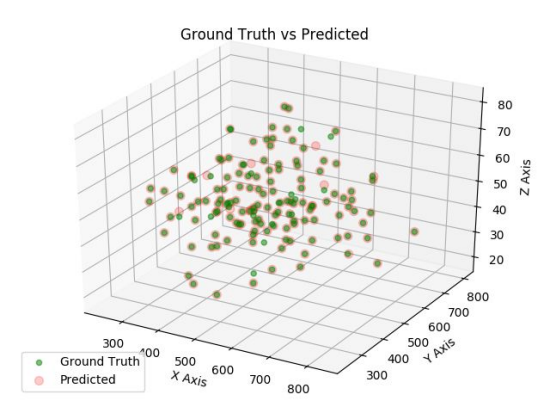
BlobMetrics Python Package

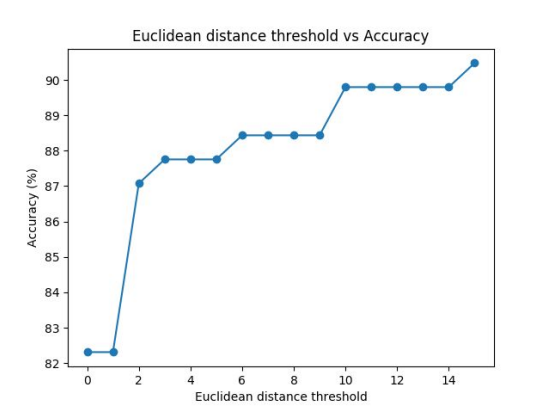
- Wrote a python package (“BlobMetrics”) to evaluate the results of a blob detector
  - Usage Documentation, Source Code, Notebook
- In process of making the package pip installable

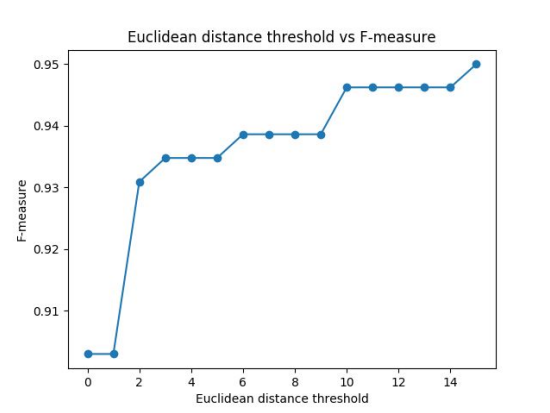
```
$ pip install blob-metrics
```

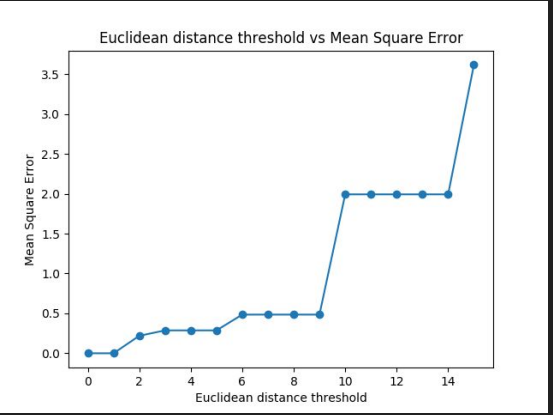
- Given the ground truth values and the predicted values the package computes
  - Accuracy
  - Precision
  - Recall
  - F-Measure
  - G-Measure
  - Mean Square Error

BlobMetrics Python Package - Visualization









Identified unsupervised algorithm

- Estimation of Small Blob Detection based on Local Convexity, Intensity and shape information
  - Pre-processing involves identifying blob candidate regions based on local convexity.
  - Regional blobness and regional flatness is extracted and HDoG (Hessian-based Difference of Gaussian) is applied
- Hysteresis Thresholding
  - The hysteresis mode uses a hysteresis loop to provide a more connected result. Any pixel above the upper threshold is turned white.
  - The surround pixels are then searched recursively. If the values are greater than the lower threshold they are also turned white.
  - The expected result is that there are many fewer specks of white in the resulting image.

Ensemble Methods implementation

- 2D Segmentation
- 3D Reconstruction

Algorithm

Pseudocode:

Inputs:

- img_stack
- z_dim, y_dim, x_dim

Psuedocode:

For every z_slice in the image stack:

1. Adaptive Threshold
2. Otsu's binarization to get binary image
3. Perform morphological erosion with kernal of radius of 5 voxels
4. Perform morphological opening with kernal of radius of 5 voxels
5. Get connected components using union-find
6. Compute centroids of each component

TODO: k-means for segmentation refinement

3D Reconstruction

Pseudocode

input:

- z_comps: a collection of components for all the z slice

pseudocode:

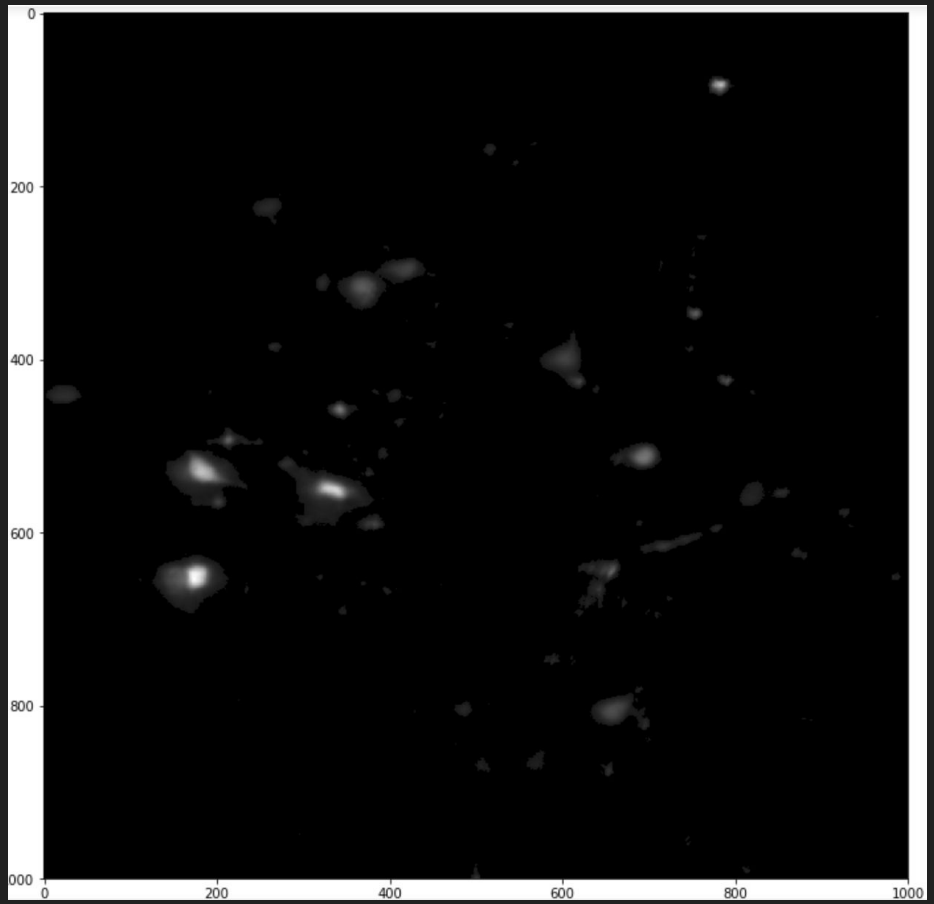
for each set of components for each z slice: for each component in this set:

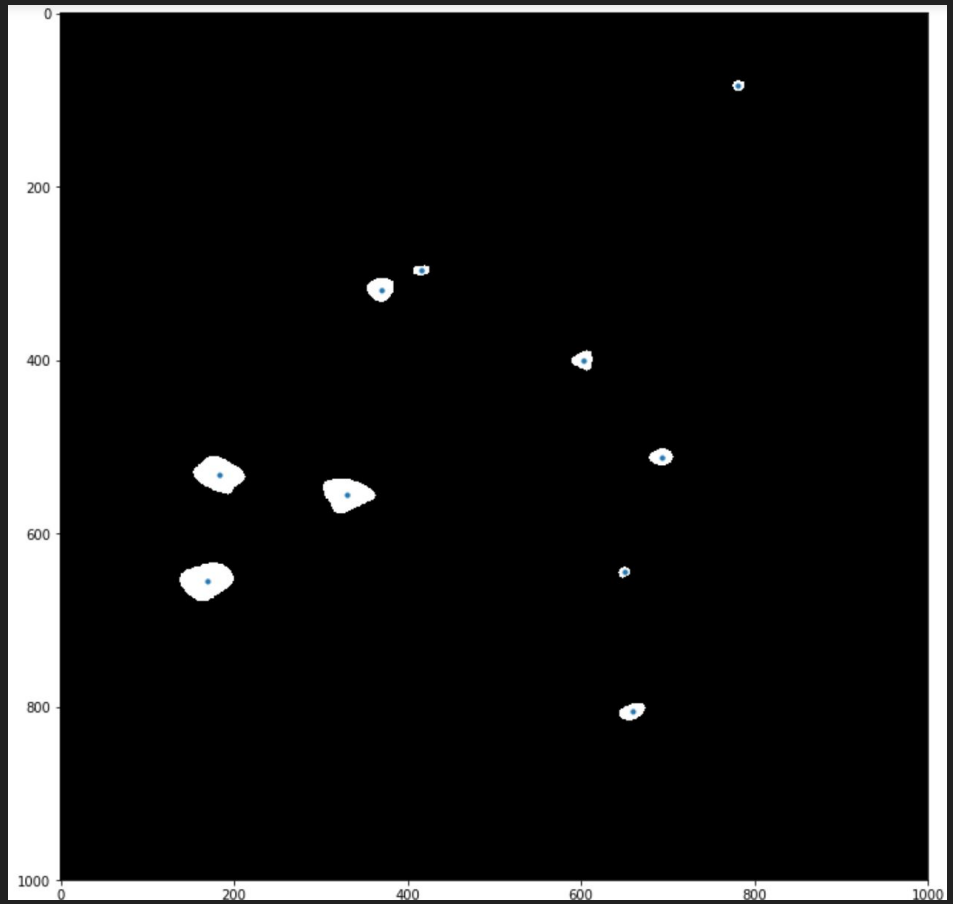
1. `If nearest centroids in z planes above and below the current plane are within a specified x–y radius, then current centroid is a part of that blob, so put it in that blob's collection`

Compute Centroid of all centroids associated with a blob

return list of blob centroids

2D Blob Detection





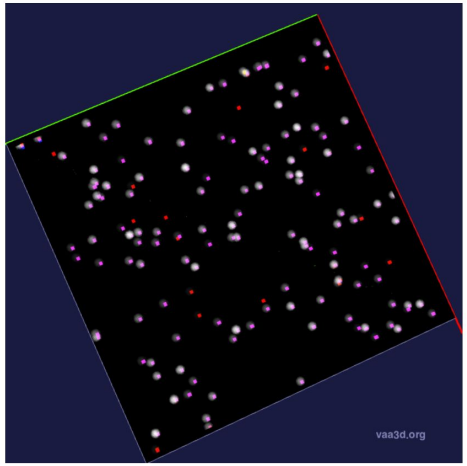
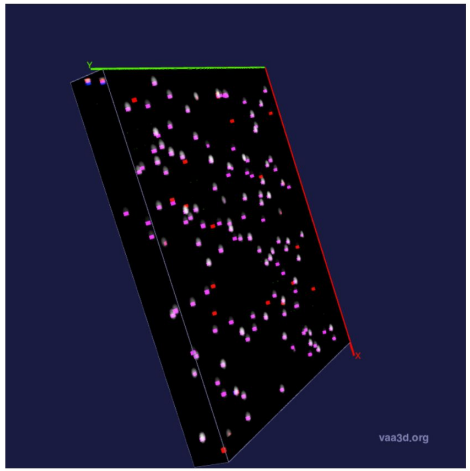
Yousef’s automatic cell detection pseudo code

- In-depth description: https://github.com/NeuroDataDesign/clarity-f17s18/blob/master/docs/jyim6/Automatic%20nuclei%20segmentation%20pipeline%20(Yousef).md
- Yousef’s algorithm is actually a pipeline of running multiple algorithms to do cell detection and segmentation
- The data medium: histopathology cells are also different than ours. We detail the steps of the algorithm relevant to us:
    a. **Binarization/Threshold:** Fit the data to a bimodal poisson mixture model, i.e. find the threshold parameter. Get the bi-modal poisson PDF for whether a cell is in the foreground or not
    b. **Labeling/large blob detection:** Run a max-flow/min-cut algorithm to discover the connected components (i.e. large blobs)
    c. **Edge detection:** Run the multiscale LoG for different scales and construct a response map.
    d. **Cell/small blob detection:** Find the local maximas in the response map.
    e. **Cell segmentation:** Run watershed or some clustering algorithm to do an initial cell segmentation.
- The rest of the steps in the actual pipeline include refining the cell segmentation and doing a graph coloring

Benchmarking

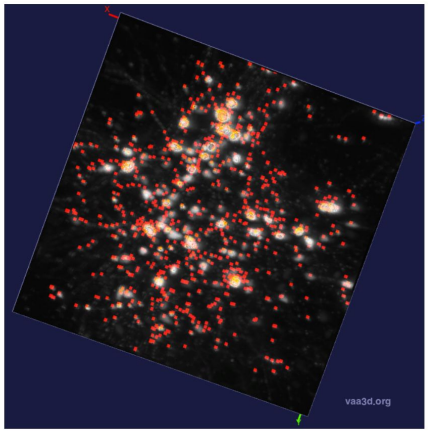
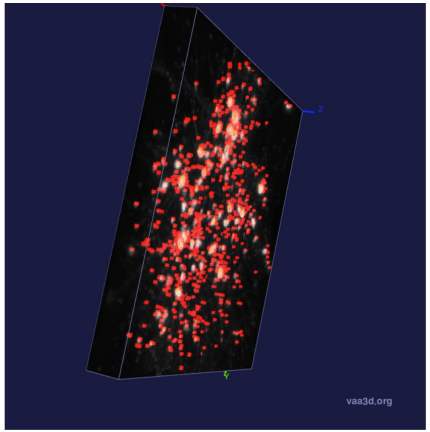
- More in depth: https://github.com/NeuroDataDesign/clarity-f17s18/blob/master/docs/jyim6/week6_deliverables.md

faded_147_randomized_cells_random_intensity

Front	Side
	

- Precision: 88%
- Recall: 100%

s3617_cutout

Front	Side
	

Next week

- Determine how to remove inhomogeneity in light sheet images (for registration)
- Identify 1 paper for each of the following in cell detection workflow:
  - Preprocessing
  - Thresholding/binarization
  - Edge detection/blob detection
  - Clustering /refinement
- Implement algorithm in this paper
- Make blob-metrics pip installable