



Week 4 deliverables

Claritron team



Last week's goals

- ✓ [Download Boss Subvolume](#)

Cell counting software survey:

- ✓ Vaa3D (Jason)
- ✓ Icy (Jon)
- ✓ DeepCell (Srivathsa)
- ✓ ClearMap (Vikram)

The shown results all used the same brain image cutout.



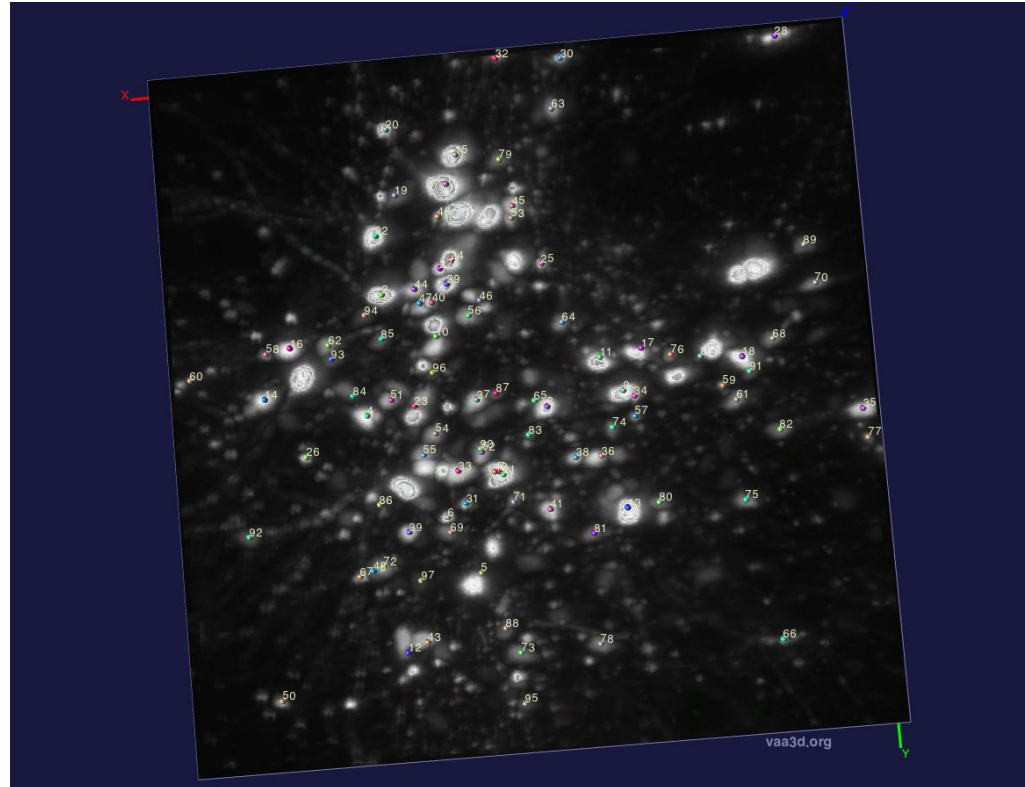
Draft of Proposal



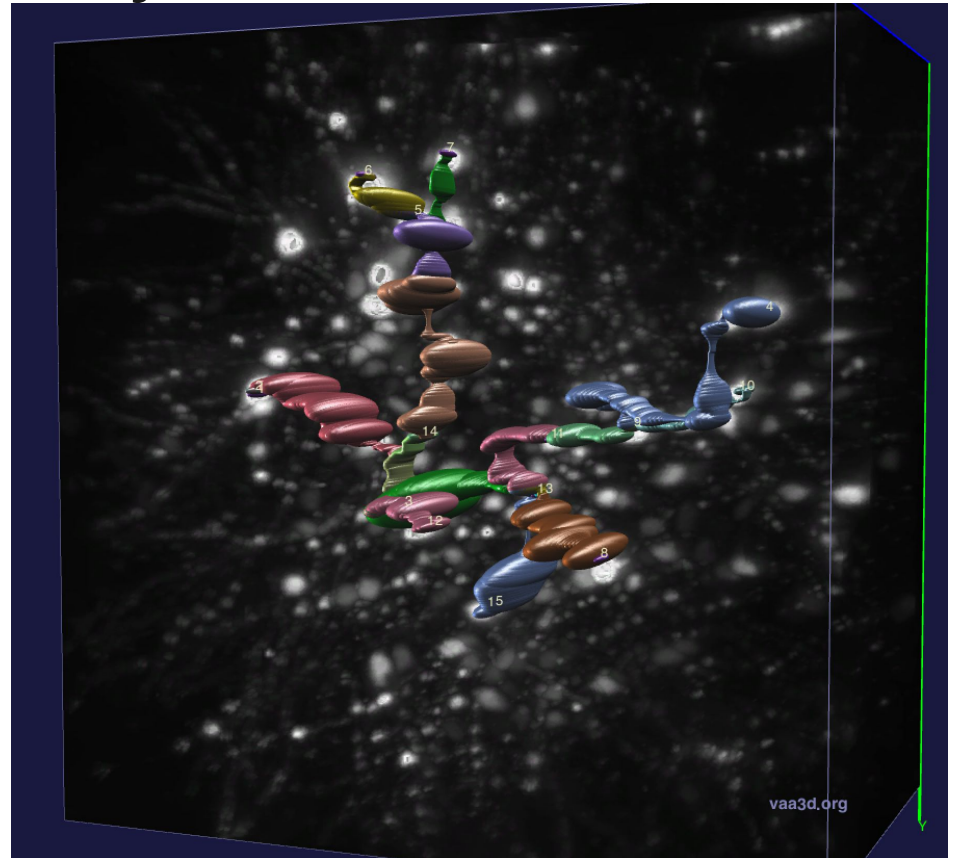
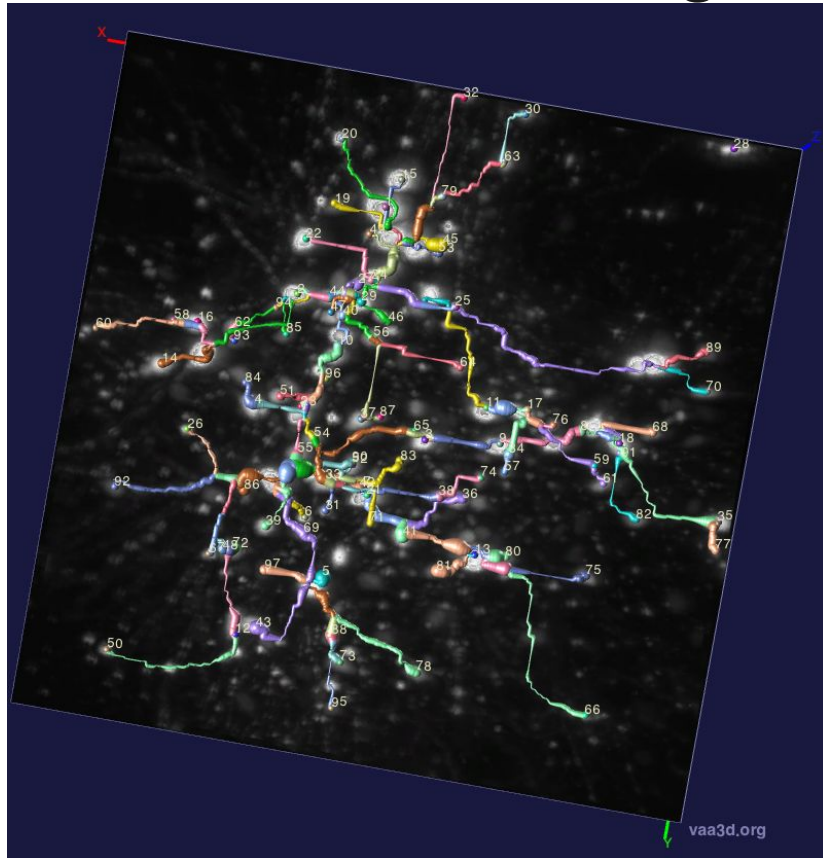
Vaa3D

- Background: Vaa3D is an open source 3D/4D/5D image visualization and analysis software for bioimage analysis
- Wrote notebook to get brain image data from BOSS and convert to tif
- Worked on getting Vaa3D to build successfully
 - After many hours debugging, discovered there are known issues with the latest OSX release:
https://www.nitrc.org/forum/forum.php?thread_id=7925&forum_id=1553
- Used the binary release to try out the software

Vaa3D cell-counting on ailey_dev



Vaa3D cell tracing on ailey_dev



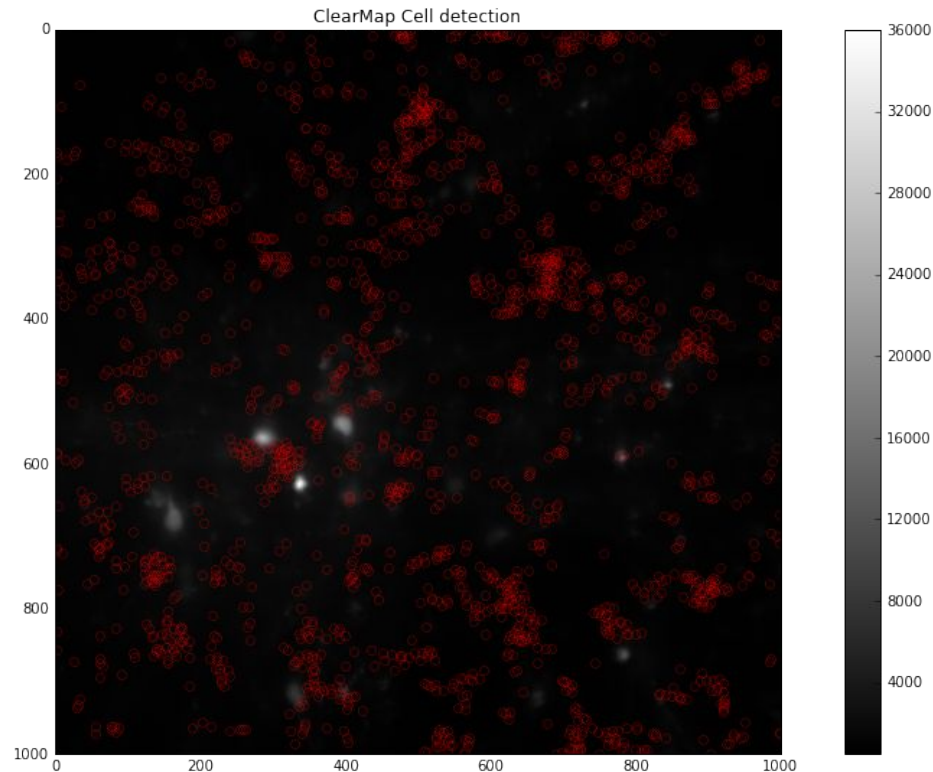


Vaa3D Summary

- Build doesn't work on OSX at the moment
- Has many bundled up plug ins written by the authors and other labs. May want to check them out.
- Nature paper summarizing Vaa3D:
http://home.penglab.com/papersall/docpdf/2014_NPROT_vaa3d.pdf
- Pros:
 - Works well, algorithms work, cell segmentation has high true positive rates, tracing works well when many exemplars are given.
- Cons:
 - Still misses a lot of cells that don't look like exemplars, need to manually select more exemplars
 - Requires manual input to perform cell-counting and tracing. The number of manual selection increases with larger images.
 - Only has a binary executable and written in C++
- Idea: talk with authors and consult them about how our pipeline can improve, build on their work?

ClearMap

- [Link to software](#)
- Wrote [Dockerfile](#) and [jupyter notebook](#)
- Seems like there are many false positives
- Many many parameters to tune so results vary wildly
- Total cells detected: 238,302
- Clearly not good





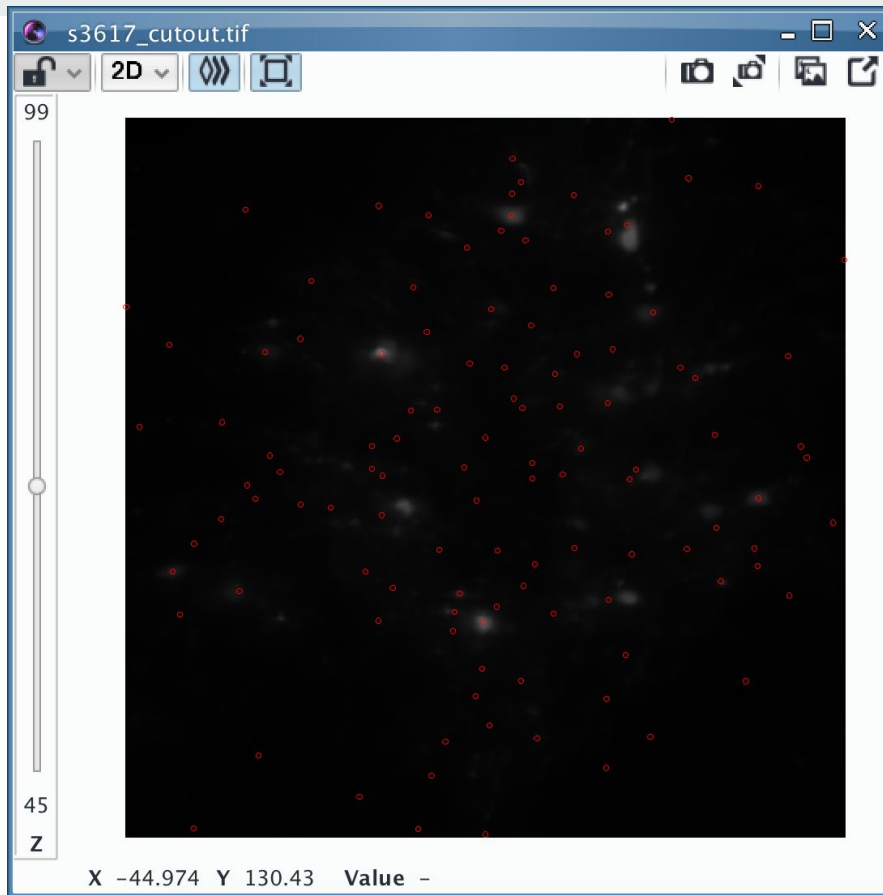
ClearMap

- Advantages
 - Designed for cleared brain tissue
 - Written in python (license for reuse)
- Disadvantages
 - Not very user-friendly (bad documentation)
 - Many parameters to tune in order to get results
- Conclusion
 - Possible submodules that may be interesting (SpotDetection) with the appropriate parameters
 - Explore the possible hyperparameter space to see which are ideal

Icy



- [Download and save subvolume using Boss Api](#) ✓
- [Do cell detection software survey for Icy](#) ✓



- [illegible]

	ROI number	ROI name	ROI surface	ROI nb detection	ROI tag(s)							
25	0	Rectangle2D	998001	115								
26												
27	Detection per ROI											
28												
29												
30	0											
31												
32												
33	Detection #	Surface	x	y	z	t	min intensit	max intensi	average intensity			
34	0	487	758.312115	2.642710472	96.9117	0	1890	5920	3420.778			
35	1	10381	516.2727098	600.7353819	94.68481	0	2145	7396	3833.93			
36	2	4918	708.2803985	488.1073607	94.89244	0	1428	21117	4682.365			
37	3	7059	351.3534495	121.972234	94.77334	0	1795	13895	3521.832			
38	4	7204	350.7284842	697.7872016	92.52485	0	2187	10804	3833.069			
39	5	8725	456.2834384	686.0179943	93.24917	0	1749	32155	4453.146			
40	6	15531	434.7837229	599.7069088	92.78965	0	1690	18072	4258.359			
41	7	17520	567.4512557	619.3127283	92.38139	0	1666	20312	4738.13			
42	8	24426	564.1198723	500.5247277	92.34864	0	1433	27788	4671.87			
43	9	2092	525.7466539	346.3804971	85.40583	0	2304	16208	5963.337			
44	10	17769	675.7579492	321.0290956	91.87422	0	1457	27178	4555.523			
45	11	2730	284.6227106	540.892674	83.49634	0	2154	15857	6405.476			
46	12	4189	504.7569826	843.0112199	83.34949	0	1949	14360	5042.585			
47	13	14840	514.7975741	678.4978437	84.63908	0	1767	16191	5435.56			
48	14	19533	200.0591307	468.3876517	88.4541	0	1168	18963	4177.994			
49	15	9543	536.5294981	105.1812847	86.65504	0	1638	36052	4050.837			
50	16	33466	693.595278	745.2377338	87.49621	0	1154	47752	5053.18			
51	17	17721	399.258789	235.2492523	87.05282	0	1612	11905	4065.847			
52	18	11	621.4545455	107.0909091	73.54545	0	1590	16946	10424.73			
53	19	71800	593.9708078	688.5770891	81.78756	0	1428	52747	6464.802			
54	20	871	61.0401837	314.7795637	72.14237	0	1864	4530	3068.135			
55	21	2209	606.6138524	494.449072	73.2947	0	1716	37478	7448.411			
56	22	7223	341.6490378	486.4759795	75.79635	0	1662	28261	5262.27			
57	23	6622	376.1780429	444.8086681	73.82664	0	1949	16003	4907.313			

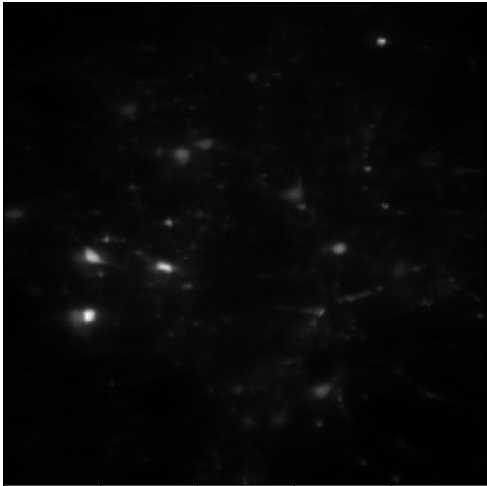


DeepCell

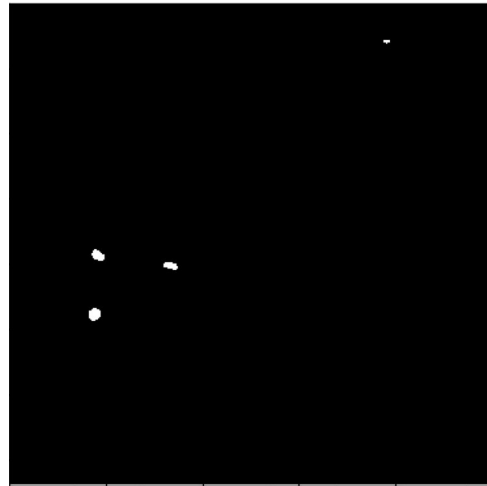
- Background: DeepCell is a Deep learning framework specifically fine tuned for cell segmentation
- Promising project overview but poor documentation
- The docker repository fails to load and the maintenance has stopped
- Couldn't run the image training samples successfully

Watershed Algorithm

- The Watershed ([`skimage.morphology.watershed\(\)`](#)) is a region-growing approach that fills “basins” in the image



Input Image



Highlighted cell regions



Watershed Algorithm

- The code shown here also counts and returns the number of cells (Cell Counting)

```
1 import numpy as np
2 import pylab
3 import mahotas as mh
4 dna = mh.imread('input.jpg')
5 pylab.imshow(dna)
6 pylab.show()
7
8 T = mh.thresholding.otsu(dna)
9 out_image = (dna > T)
10 out_image = out_image.astype(np.uint8) * 255
11 pylab.imshow(out_image)
12 pylab.show()
13
14 labeled, nr_objects = mh.label(out_image > T)
15 print nr_objects
16
17 pylab.imshow(labeled.astype(np.uint8) * 255)
18 pylab.jet()
19 pylab.show()
20
```



Next Week's Goals

- Complete Project proposal and sprint timeline
- Contact Vaa3D authors to determine which algorithms are ideal to implement
- Find Vaa3D/Icy papers describing cell detection/tractography
- Implement cell detection algorithm from Vaa3D
- Extend the watershed algorithm and implement it to return the cell centroids