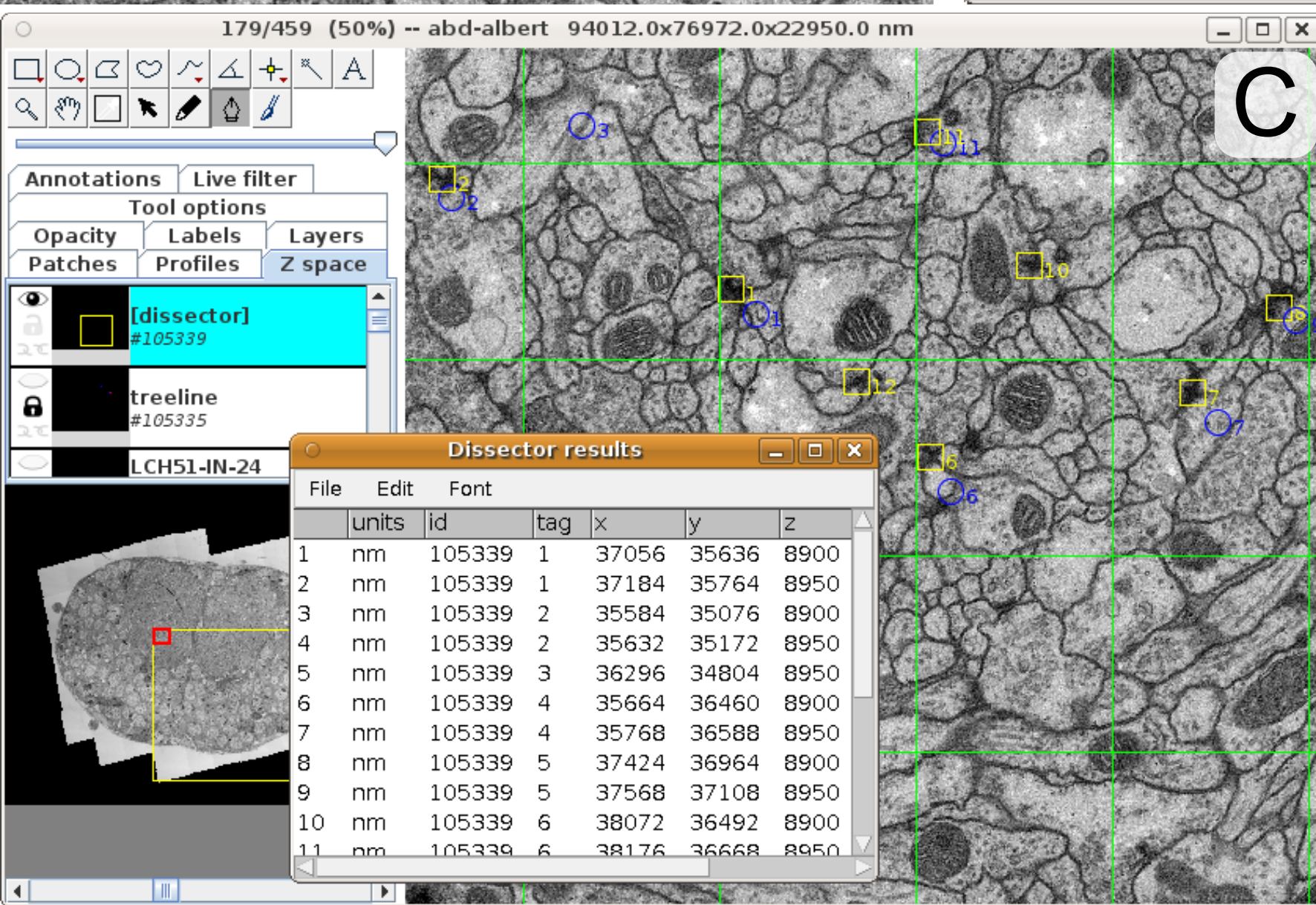


B

Tree path measurements

	units	id	XA	YA	Layer A	XB	YB	Layer B	distance	N nodes	N branch
1	nm	105311	5599.51	5833.33	118	8180.05	9960.61	196	35289.46	191	1
2	nm	105311	5599.51	5833.33	118	9221.50	9776.58	161	38985.44	191	6
3	nm	105311	5599.51	5833.33	118	8173.68	9962.16	197	35345.91	192	1
4	nm	105311	5599.51	5833.33	118	9657.42	9777.17	159	41330.34	194	7
5	nm	105311	5599.51	5833.33	118	9502.19	10048.04	150	41406.76	203	7
6	nm	105311	5599.51	5833.33	118	8458.57	9831.38	220	38744.73	214	2
7	nm	105311	5599.51	5833.33	118	9811.43	9655.46	232	46050.27	239	7
8	nm	105311	5599.51	5833.33	118	10259.96	9713.48	172	49170.01	240	13
9	nm	105311	5599.51	5833.33	118	9799.04	9547.12	225	46849.86	246	8
10	nm	105311	5599.51	5833.33	118	9776.57	9682.87	220	47764.86	251	9
11	nm	105311	5599.51	5833.33	118	10129.82	9404.53	83	51416.46	274	14
12	nm	105311	5599.51	5833.33	118	9895.04	9581.52	80	50334.41	275	12
13	nm	105311	5599.51	5833.33	118	10171.80	9558.58	99	53029.04	289	15
14	nm	105311	5599.51	5833.33	118	10213.79	9924.05	255	54210.45	293	12
15	nm	105311	5599.51	5833.33	118	8662.87	9059.87	37	61144.24	372	17
16	nm	105311	5599.51	5833.33	118	8676.82	9004.19	31	61702.25	378	18
17	nm	105311	5599.51	5833.33	118	8590.16	9015.95	24	63187.84	386	19
18	nm	105311	5599.51	5833.33	118	8736.79	8771.76	48	62754.20	386	20

pre [root] pre



D

Vesicle_to_Surface_statistics.py (Running)

```

File Edit Language Templates Run Tools Git Tabs
Vesicle_to_Surface_statistics.py
1 # The IDs of the Ball and AreaList instances
2 vesiclesID = 1543
3 synapticSurfaceID = 1541
4
5 # Obtain the two TrakEM2 instances
6 project = Project.getProjects()[0]
7 vesicles = project.findById(vesiclesID)
8 synapticSurface = project.findById(synapticSurfaceID)
9
10 # A set of unique vertices defining the synaptic surface
11 vertices = set(synapticSurface.generateTriangles(1, 2))
12
13 # For every vesicle, measure its shortest distance to a vertex
14 distances = [reduce(min, map(lambda v: p.distance(v), vertices))
15              for p in vesicles.asWorldPoints()]
16
17 # Compute average, median and standard deviation
18 mean = sum(distances) / len(distances)
19 stdDev = Math.sqrt(reduce(lambda s, e: s + pow(e - mean, 2),
20                          distances, 0)) / len(distances)
21 median = sorted(distances)[len(distances)/2]
22
23 print mean, stdDev, median

```

Run Kill Show Errors Clear

Started Vesicle_to_Surface_statistics.py at Fri Mar 25 11:02:39 CET 2011
98.3640799271 9.32777187286 104.904457092