

- 
- 
- 
- 
- 
- 

### Neuron Network

```
1 import networkx as nx
2
3 from pandas import read_csv
4
5 from matplotlib.pyplot import subplots, show
```

### The C elegans Neuron Network

#### Data Import Introduction to C elegans neurons

C elegans neurons in the worm atlas

Import the connectivity matrix.

```
1 data = read_csv('data/celegans131matrix_50.csv', header=None, dtype = "
    int")
2
3 neurons = data.to_numpy()
4
5 print(len(neurons))
```

```
1 50
```

```
1 fig, ax = subplots()
2
3 ax.imshow(neurons);
4
5 show()
```

Import the labels and convert the resulting dataframe into a dictionary. The function `to_dict` wraps the dictionary within a dictionary and therefore indexing is used to access the ‘inner’ dict.

```
1 neuron_Names = read_csv('data/celegans131labels_50.csv', header=None)
2
3 neuronNames = neuron_Names.to_dict()
```

```

4
5 neuronLabels = neuronNames[0]
6
7 print(neuronLabels)

```

```

1 {0: 'ADFL', 1: 'ADFR', 2: 'ADLL', 3: 'ADLR', 4: 'AFDL', 5: 'AFDR', 6: 'AIAL', 7: 'AIAR', 8: 'AIBR', 9: 'AINL', 10: 'AINR', 11: 'AIZL', 12: 'AIZR', 13: 'ALA', 14: 'ASEL', 15: 'ASER', 16: 'ASGL', 17: 'ASGR', 18: 'ASHL', 19: 'ASHR', 20: 'ASIL', 21: 'ASIR', 22: 'ASJL', 23: 'ASJR', 24: 'ASKL', 25: 'ASKR', 26: 'AUAL', 27: 'AUAR', 28: 'AVAL', 29: 'AVAR', 30: 'AVBL', 31: 'AVBR', 32: 'AVDL', 33: 'AVDR', 34: 'AVEL', 35: 'AVER', 36: 'AVHL', 37: 'AVHR', 38: 'AVJL', 39: 'AVJR', 40: 'AVL', 41: 'AWAL', 42: 'AWAR', 43: 'AWBL', 44: 'AWBR', 45: 'AWCL', 46: 'AWCR', 47: 'BAGL', 48: 'BAGR', 49: 'CEPDL'}

```

## Graph Object and Network Display

The connectivity matrix can be converted to a Networkx graph.

```

1 neuronGraph = nx.from_numpy_matrix(neurons,
2                                   create_using=nx.DiGraph)
3
4 neuronLayout = nx.random_layout(neuronGraph, seed=12)
5
6 nx.draw_networkx(neuronGraph, neuronLayout,
7                  node_size=1000,
8                  labels=neuronLabels)

```

View the degrees:

```

1 fig, ax = subplots()
2
3 ax.plot(dict(neuronGraph.degree).values(), '-o');
4 ax.set_xlabel('Index');
5 ax.set_ylabel('Degree');
6
7 show()

```

The network displayed in circular layout:

```

1 neuronGraph = nx.from_numpy_matrix(neurons,
2                                   create_using=nx.DiGraph)
3
4 neuronLayout = nx.circular_layout(neuronGraph)
5
6 nx.draw_networkx(neuronGraph, neuronLayout,
7                  node_size=1000,
8                  labels=neuronLabels)
9

```

```
10 show()
```

Find node indices: The two BAG nodes 'BAGR' and 'BAGL' of the display (about 4 o'clock).

```
1 for index, name in enumerate(neuronLabels.values()):
2
3     if 'BAG' in name:
4
5         print(neuronLabels[index], index)
```

```
1 BAGL 47
2 BAGR 48
```

## Anatomical Mapping

File 'celegans131positions\_50.csv' contains information on how the nodes relate to each other in 2-D space. We can include this information to replace the layout.

```
1 neuronPos = read_csv('data/celegans131positions_50.csv', header=None)
2
3 neuronPos.items
```

```
1 <bound method DataFrame.items of                                0          1
2 0    0.082393 -0.000984
3 1    0.083279 -0.003184
4 2    0.082639 -0.013035
5 3    0.083279 -0.011512
6 4    0.086329 -0.002706
7 5    0.086463 -0.000980
8 6    0.065177  0.009346
9 7    0.059030  0.011512
10 8    0.075441  0.006123
11 9    0.061980 -0.006149
12 10   0.061969 -0.003429
13 11   0.048698  0.002706
14 12   0.057560  0.003184
15 13   0.094056 -0.013227
16 14   0.069112  0.000492
17 15   0.071767 -0.000735
18 16   0.077966 -0.007625
19 17   0.080095 -0.010287
20 18   0.075507  0.000492
21 19   0.078380 -0.000980
22 20   0.076982 -0.012789
23 21   0.077156 -0.011757
24 22   0.063455  0.006641
25 23   0.062704  0.009063
26 24   0.088542 -0.010084
```

```

27 25 0.088668 -0.009553
28 26 0.068620 0.006149
29 27 0.067113 0.005389
30 28 0.089526 0.001722
31 29 0.090872 -0.000245
32 30 0.069112 -0.004427
33 31 0.071767 -0.006368
34 32 0.061734 -0.001476
35 33 0.066378 -0.001470
36 34 0.082885 0.002214
37 35 0.084014 0.003184
38 36 0.072063 -0.008608
39 37 0.076421 -0.012737
40 38 0.067636 -0.009346
41 39 0.072992 -0.009798
42 40 0.060990 0.009063
43 41 0.077966 -0.002951
44 42 0.078380 -0.005389
45 43 0.079196 -0.003935
46 44 0.082789 -0.006858
47 45 0.078458 0.004919
48 46 0.078625 0.004164
49 47 0.112890 0.000492
50 48 0.114630 0.003184
51 49 0.094199 -0.016233>

```

```

1 nx.draw_networkx(neuronGraph, neuronPos.values,
2                 node_size=1000,
3                 labels=neuronLabels)
4 show()

```

```

1 new_pos = neuronPos.copy()
2 new_pos.values[:, 0] = -1*neuronPos.values[:, 0]
3
4 nx.draw_networkx(neuronGraph, new_pos.values,
5                 node_size=1000,
6                 labels=neuronLabels)
7 show()

```

The two BAG nodes to the right of the display are the sensory neurons used to monitor oxygen and carbon dioxide.

Here is some background: BAG genes, functions and connections

The graph display is in Matplotlib and can be handled as such. E.g. for possible Node shapes see: [https://matplotlib.org/stable/api/markers\\_api.html#module-matplotlib.markers](https://matplotlib.org/stable/api/markers_api.html#module-matplotlib.markers).

If you have a multi-panel figure, specify the axes for the network with keyword argument `ax`:

```

1 fig, ax = subplots(figsize=(14, 8), ncols=2)

```

```
2
3 nx.draw_networkx(neuronGraph, neuronLayout,
4                 node_size=1000,
5                 labels=neuronLabels,
6                 ax=ax[0])
7
8 nx.draw_networkx(neuronGraph, neuronPos.values,
9                 node_shape='H',
10                 node_color='tomato',
11                 node_size=1300,
12                 labels=neuronLabels,
13                 ax=ax[1])
14
15 ax[0].set_title('Circular view of C elegans network');
16 ax[1].set_title('Anatomical view of C elegans network');
17
18 show()
```

### Keypoints

- 
- 
-