Community Detection in the C. elegans Connectome

BY MARK HORENI
My Research

Previous work the C. elegans connectome with link analysis

Was curious about community detection in this data set

Currently working on alternative artificial neural network structures that are closer to biology
The Data

280 neurons (humans have 100 billion)
6393 chemical synapses
890 electrical junctions
1410 neuromuscular junctions

Neurons can be either Sensory, Inter, Motor, or a combination
Synapses
Characteristics about the graph

Directed
Not a true “Feed Forward Network”
Includes cycles, but no self loops
Weighted
Densely connected
Shows a high degree of modularity
More Visualization

https://elegans.herokuapp.com/

http://wormweb.org/neuralnet#c=BAG&m=1
Modularity

Metric to determine communities

Number between -1 and 1 of how strong communities are

Global Property

Goal: Maximize Modularity

\[ Q = \frac{1}{2m} \sum_{ij} \left[ A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \]
Louvain

Each node starts in a community by itself

Put node $i$ in a neighboring community

A change in $Q$ is computed:  
$$\Delta Q = \left[ \frac{\Sigma_{in} + 2k_{i,in}}{2m} - \left( \frac{\Sigma_{tot} + k_i}{2m} \right)^2 \right] - \left[ \frac{\Sigma_{in}}{2m} - \left( \frac{\Sigma_{tot}}{2m} \right)^2 - \left( \frac{k_i}{2m} \right)^2 \right]$$

If the change is positive, $i$ becomes part of that community

Once these communities are formed, communities themselves become nodes and intra community weights are treated as self loops, and inter community weights are treated as connections

Repeat
Example
Pseudocode

\[ O(N \log N) \]

1: \( V \): a set of vertices
2: \( E \): a set of edges
3: \( W \): a set of weights of edges, initialized to 1
4: \( G \leftarrow (V, E, W) \)
5: repeat
6: \( C \leftarrow \{\{v_i\} \mid v_i \in G(V)\} \)
7: calculate current modularity \( Q_{\text{cur}} \)
8: \( Q_{\text{new}} \leftarrow Q_{\text{cur}} \)
9: \( Q_{\text{old}} \leftarrow Q_{\text{new}} \)
10: repeat
11: for \( v_i \in V \) do
12: \( Q_{\text{cur}} \leftarrow Q_{\text{new}} \)
13: remove \( v_i \) from its current community
14: \( N_{v_i} \leftarrow \{c_k \mid v_i \in G(V), v_j \in c_k, e_{ij} \in G(E)\} \)
15: find \( c_x \in N_{v_i} \) that has \( \max \Delta Q_{\{v_i\},c_x} > 0 \)
16: insert \( v_i \) into \( c_x \)
17: end for
18: calculate new modularity \( Q_{\text{new}} \)
19: until no membership change or \( Q_{\text{new}} = Q_{\text{cur}} \)
20: \( V' \leftarrow \{c_i \mid c_i \in C\} \)
21: \( E' \leftarrow \{e_{ij} \mid \forall e_{ij} \text{ if } v_i \in C_i, v_j \in C_j, \text{ and } C_i \neq C_j\} \)
22: \( W' \leftarrow \{w_{ij} \mid \sum w_{ij}, \forall e_{ij} \text{ if } v_i \in C_i \text{ and } v_j \in C_j\} \)
23: \( G \leftarrow (V', E', W') \)
24: until \( Q_{\text{new}} = Q_{\text{old}} \)
Future Work

Use a directed version of the algorithm
See if the communities form a biological purpose
Make the algorithm work in parallel