Studying the connectome of *Caenorhabditis elegans* using graph theory

Sergio Peignier
sergio.peignier@insa-lyon.fr

Prerequisites

- Python (libraries: Matplotlib, Numpy, Pandas).
- Basic biological knowledge about neurons (neuron, axon, synapse, dendrite, ...).
- Basic knowledge about graph theory (graph, node, edge, degree, adjacency matrix, graph traversal algorithms).

Complementary sources

- Book: Dynamical Process On Complex Networks, BARRAT et al., 2008
- Article: Using graph theory to analyze biological networks, Pavlopoulos et al., 2011.
- Article: Network Motifs: Simple building blocks of complex networks, Milo et al., 2002.
- Article: The rich club of the *C. elegans* neural connectome, Towlson et al., 2013.

Objectives

General Competencies

- Analyze a system: Model a complex system by means of mathematical tools.
- Exploit a real system model: Implement a computer simulation of the model.
- Analyze data: Select/implement analysis tools, detect tendencies and interpret results.

Specific Knowledge

- Network motifs, graph metrics (connection distance, efficiency, rich club coefficient, connection distance) based on graph traversal algorithms.
- NetworkX Python package functions to create, manipulate, and study graphs.

Specific Competencies

- Model a complex biological network as a graph and implement/study it in python using Networkx.
- Interpret results in biological terms.

1 Introduction

In this practical work we apply graph algorithms to study the connectome (neural network) of the *Caenorhabditis elegans* organism. The different steps presented here are based on two research papers, that analyzed specific aspects of this complex biological network: [MSOI+02] that studied the constitutive motifs of complex networks such as the connectome of *Caenorhabditis elegans*; and [TVA+13] that focused on the study of the so-called "rich club neurons", a small number of important neurons, using different tools from graph theory. In section 2 we give some information regarding the *Caenorhabditis elegans* organism, and the Openworm project that allowed to determine its connectome. In section 3, we recall some basic graph terminology. In Section 4, we will proceed to load the connectome, and visualize it. In Section 5, we use the networkx package to implement the graph based model. In Section 6, we focus on the connectome network motifs, we implement three motif search algorithms, and introduce the concept of random configuration model. We compare the proportion of motifs in the connectome and in random configuration graphs. Finally, in Section 7, we study the so called Rich Club neurons, computing important metrics: the *Rich club coefficient*, the *connection distance*, the *Efficiency*, the *Betweenness Centrality* and the *Modularity*. Then we investigate their role in the connection of groups of neurons at a larger scale.

2 *Caenorhabditis elegans* and the Openworm project

Caenorhabditis elegans is a small (∼1mm length) not parasitic transparent nematode (roundworm) that lives in temperate soil environments. *C. elegans* worms live around 3 years and have large progeny (around 300 children). These worms are mostly hermaphrodite (∼ 99%), and like other nematodes they lack a respiratory and a circulatory system. They have a digestive system that includes a mouth, pharynx and intestine. The locomotion of their bodies (i.e., dorsal and ventral bending) is ensured by four longitudinal bands of muscles that receive signal from the neural system. The movements of the head are controlled by four muscles wired independently. When the waves of contractions of the body muscles proceed from the back (resp. front), then the animal moves backward (resp. forward). An interesting video that illustrates the locomotion of *C. elegans* can be found in `https://www.youtube.com/watch?v=GgZHziFWR7M`.

C. elegans is often used as a model organism and recently an interesting international project called the OpenWorm project [KHL98] has been developed (`http://www.openworm.org/`). This project aims to simulate C. elegans at the cellular level. The first stage is to study the locomotion system of the worm by modeling the 302 neurons of the hermaphrodite worm and the 95 muscular cells. The long term goal of the OpenWorm project is to simulate the entire worm (959 somatic cells).

In this practical course we will analyze the connectome of *C. elegans* that has been found by the OpenWorm project so far, using graph traversal algorithms.

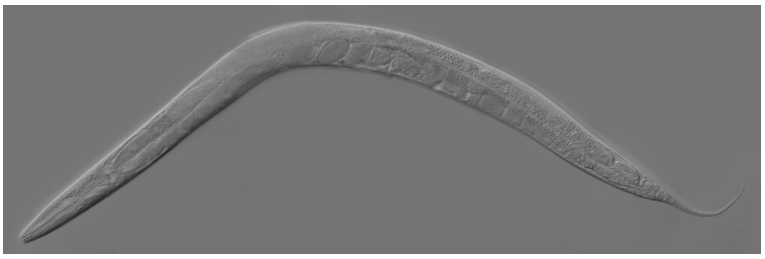

Figure 1: Adult *Caenorhabditis elegans*

3 Networks and Graphs (reminder)

In this practical course, we are going to represent the connectome as a graph. A graph, is a mathematical structure that allows to model pairwise "relations" between "objects". Objects are often called vertices or nodes ("noeuds" or "sommets" in french), and relations between objects are called edges, arcs, or lines ("arêtes", "arcs" or "liens" in french). Figure 2 shows the classic

representation of a graph: nodes are represented as circles (with labels form 1 to 6) and lines between circles represent edges. Formally, a graph $G = \langle V, E \rangle$ is a pair of two sets, the set of nodes $V = \{v_1, v_2, \dots\}$ and the set of edges $E = \{e_1, e_2, \dots\}$, such that $e = \{v_i, v_j\}$ belongs to E if and only if $v_i \in V$, $v_j \in V$ and there is a connection between v_i and v_j.

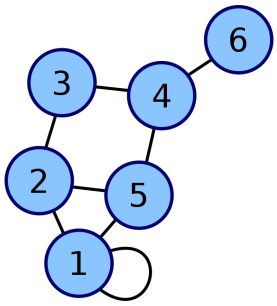

Figure 2: Classic representation of a graph

The edges between nodes can be undirected or directed, in the first case the relation is symmetric. In the second case the order matters, and an edge is an ordered pair of nodes $e = (v_i, v_j)$ such that the connection goes from v_i to v_j. Figure 3 shows the classical way to represent undirected and directed arcs between two nodes.

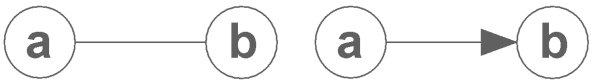

Figure 3: Undirected (left) and directed (right) arcs between nodes a and b

In this practical course, we represent the *Caenorhabditis elegans* connectome as a directed graph, were each neuron is a node and a synapse from a neuron a to a neuron b will be represented as a directed arc between these two nodes, as shown in Figure 4.

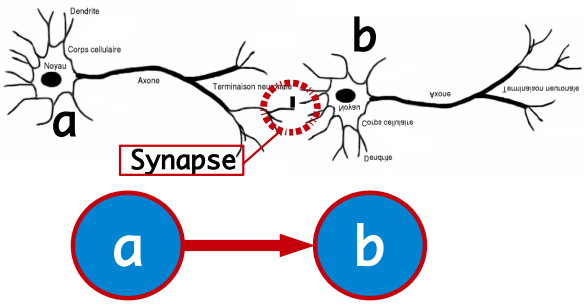

Figure 4: Pair of connected neurons modeled as two nodes connected by an arc.

4 Connectome exploration

- Download the data from http://www.biological-networks.org/pubs/suppl/Kaiser2006.html (file Individual CSV files), provided by [KH06].
- Three files are available at the previous url:
  - "celegans277labels.csv": contains the names of each neuron.
  - "celegans277positions.csv": the location in the Anterior - Posterior / Ventral - Dorsal plan.

  - "celegans277matrix.csv": Matrix M of connections, rows and columns correspond to neurons, and the element $M_{i,j} = 1$ if the i-th neuron has a synaptic connection with the j-th neuron (directed connection), and $M_{i,j} = 0$ otherwise. Considering the network as a graph, this matrix would correspond to the so-called adjacency matrix.

- Create a Jupyter notebook, load the files using *Pandas*.
- Use the function to plot the connectome (the result you should obtain is depicted in Figure 5). What can you say about the connectome?
- Plot the connection matrix using the function imshow of matplotlib library. Compute the proportion of ones with respect to the number of elements in the matrix. What does it mean? Which data structure would you use to represent such a network?
- Search what is the in-degree and the out-degree of a node. And compute such values for each node in the network. Rank this values and plot them.
- Plot the histogram of in-degrees and out-degrees. What can you observe?

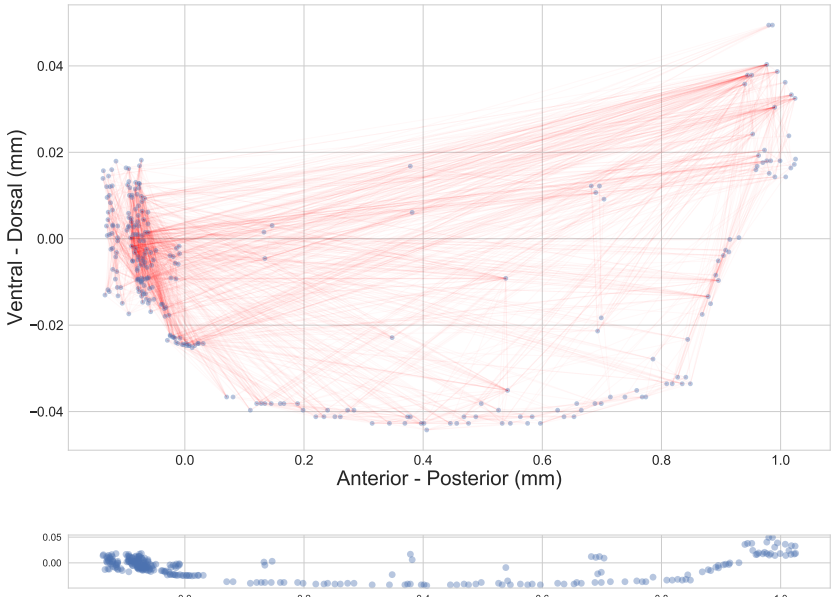

Figure 5: Spatial representation of the connectome, blue circles represent nodes (neurons) and red lines represent edges (synapses). The bottom figure represents the location of the neurons using the same scale to represent both axis (now we can see the shape of the worm more clearly!).

5 Networkx

- In order to represent our graph in networkx, we need to create a data frame with three columns:
  - The first one called "input" contains the pre-synaptic neuron id.
  - The second one called "output" contains the post-synaptic neuron id.
  - The third one called "weight" will contain the distance between the two neurons (euclidean distance).
- import the networkx library and use the function from_pandas_edgelist function to build a networkx Graph object from our data frame. Does it create a directed graph? Instantiate a "DiGraph" object and use the create_using parameter to create a directed graph with the from_pandas_edgelist function.

- Explore the networkx library and find out another way to build the graph (e.g., https://networkx.github.io/documentation/networkx-1.10/tutorial/index.html).
- Explore the functions that this library contains, and the methods and attributes of a Graph object. Try for example the nodes() and edges() methods. How can you access the neighbors of a node? How can you get the edges weights?

6 Network motifs

- Download the paper [MSOI+02][1] and read relevant parts that should allow you to answer the following questions.
- How would you define a network motif?
- Which are the most important motifs in *C. elegans* connectome?
- Think about the possible role of each one of these motifs.
- Design and then implement algorithms to search the patterns depicted in Figure 6. Your functions should take a Networkx Directed graph as an input and should output the instances of the given motif, found in the graph.
- Take a look to the directed_configuration_model generator from networkx[2], take a look at the source code, and explain what it does, and what could it be useful for?
- Compute the number of motifs in the connectome graph and many corresponding configuration graphs, what do you observe (interpretation)?

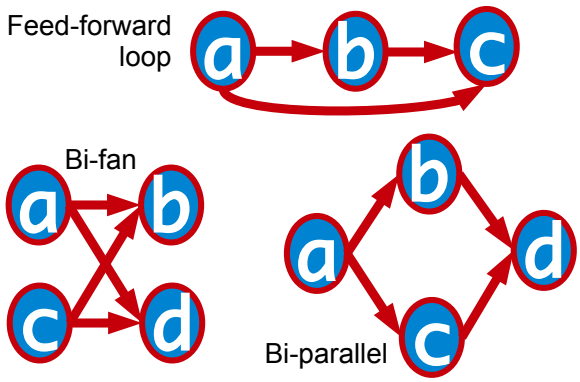

Figure 6: Feed-forward loop, bi-fan and bi-parallel motifs

7 Rich Club neurons

In [TVA+13], the authors focused on the study of the so-called "rich club neurons", a small number of important neurons, using different metrics from graph theory (*Rich club coefficient*, the *connection distance*, the *Efficiency*, the *Betweenness Centrality* and the *Modularity*), in order to understand better the topology and the functioning of the connectome. Many of these measures rely on graph traversal algorithms that allow to compute the shortest path between two nodes, such as the Dijkstra's algorithm that we have studied previously.

[1] http://science.sciencemag.org/content/sci/298/5594/824.full.pdf?casa_token=2wi8QoyQjSoAAAAA:sU-tGpwK17Te9Z8-s6FAlvQQLX1UWszyWQKCQANPOFyIhuMTKqyNQu-t_qW92oeBfV2Yke6c6oab

[2] https://networkx.github.io/documentation/networkx-1.10/reference/generators.html

7.1 Studying the Rich Club

- Download the paper [TVA+13][3].
- read the Materials and Methods, and understand each of the metrics that will be used to characterize the connectome:
  - *Rich club coefficient*
  - *Connection distance*
  - *Efficiency*
  - *Betweenness Centrality*
- Search the corresponding implementations of these metrics in the networkx package.
- Detect the rich club of the connectome (should be the same found in [TVA+13]).
- Compute the metric for the Rich-club neurons, for the poor periphery, and for the whole network.
- Plot the results using box-plots, and interpret.

7.2 Communities

The Girvan Newman method is an algorithm that relies on the edge betweenness centrality to detect communities in graphs. This algorithm computes the betweenness of each existing edge in the graph, then the edge with the highest betweenness is removed, these two steps are repeated until it remains a given number of connected components. This algorithm separates communities of nodes that are strongly intra-connected, by removing iteratively edges with high betweenness centrality, that tend to inter-connect such communities.

- Search the Networkx implementation of this algorithm, and apply it to detect communities in the connectome (test different number of communities).
- Compute for number of inter-community and intra-community connections involving rich club neurons, and poor periphery neurons.
- Plot the results for different number of communities produced, and interpret.
- Did you find, like in the paper [TVA+13], that rich club neurons are often involved in the inter-community communications?
- Regarding all these results, how important are rich club neurons? What could be there roles?

Bonus

- Take a look to [SCA+11][4], and explain the possible role of communities.
- Plot communities, rich club neurons and poor periphery neurons, in the Ventral-Dorsal and Anterior-Posterior plan of the animal. Are communities clustered in space? Where are located rich club neurons?

References

[KH06] Marcus Kaiser and Claus C Hilgetag. Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems. *PLoS computational biology*, 2(7):e95, 2006.

[KHL98] Hiroaki Kitano, Shugo Hamahashi, and Sean Luke. The perfect c. elegans project: An initial report. *Artificial Life*, 4(2):141–156, 1998.

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4104292/
[4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3098222/

[MSOI+02] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[SCA+11] Yunkyu Sohn, Myung-Kyu Choi, Yong-Yeol Ahn, Junho Lee, and Jaeseung Jeong. Topological cluster analysis reveals the systemic organization of the caenorhabditis elegans connectome. *PLoS computational biology*, 7(5):e1001139, 2011.

[TVA+13] Emma K Towlson, Petra E Vértes, Sebastian E Ahnert, William R Schafer, and Edward T Bullmore. The rich club of the c. elegans neuronal connectome. *Journal of Neuroscience*, 33(15):6380–6387, 2013.