

An Experimental Comparison of Algorithms for Nodes Clustering in a Neural Network of Caenorhabditis Elegans

Jorge Hernandez and Hector Florez(✉)

Universidad Distrital Francisco Jose de Caldas, Bogota, Colombia
hrjorgee@correo.udistrital.edu.co, haflorezf@udistrital.edu.co

Abstract. Caenorhabditis Elegans (C. Elegans) is a worm, which has had several studies to search its nerve paths. In a neuronal network simulation, it is util to know which is the first weight to assign in each link if it is not presented to determine other characteristics (e.g. distances by weights). Normally, the weight is a heuristic to solve a problem. There is a data set about connections of C. Elegans which is a result of other authors. The weights by the connection are not set in the data set. In this work, we use the data set to determine experimental weights for each connection with four cluster algorithms. The weights are to use in future work. To compare the algorithms, we created several models for each algorithm. We used metrics to evaluate the results for each model. A spectral clustering algorithm was chosen how the better algorithm to generate the weights.

Keywords: Clustering · Caenorhabditis Elegans · MiniBatch K-means algorithm · DBSCAN algorithm · Spectral clustering algorithm · Clustering feature tree algorithm

1 Introduction

In animals, the inputs are received and managed in a neuronal network to make animal behavior. In comparison with machines, the behavior in robots is programmed previously by the human. There are several techniques and theories to calculate the movement in the machines [7,15,18]. The main purpose of robots is to simulate their behavior. In animals, movements are through external (environment) or internal stimulus and managed by the nervous system. In healthy conditions, the brain must be able to continuously maintain strict functional control between the inputs and outputs of the system [16,23]. Thus, the behavior is dynamic in comparison with the machines.

Creating neural networks based on real connections could allow us to understand how animal behavior works, thus, bio-inspired models for behavior in agents could be created. A neural network is formed with neuron cells that

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12957, pp. 327–339, 2021.
https://doi.org/10.1007/978-3-030-87013-3_25

receive or apply the chemical or electrical stimulus. Caenorhabditis Elegans is a worm, which has two sexes: hermaphrodite and male. The map of the neuronal connections of C. Elegans is almost complete and was proposed by Sydney Brenner [19].

The nervous system of C. Elegans, which consists of 302 neurons with three types (sensory, inter, motor) is relatively small to be analyzed [11]. To represent its neural network with a computational approach, it is necessary to create a connections graph. To make a simulation network, it is necessary to know the weights of connections to calculate the distances between nodes for future calculations. The nodes are cells of neuronal networks of C. Elegans. In the C. Elegans neural network, connections are structured and weights can be based on its structure. In others applications, each edge of a graph has a numeric value that corresponds to the weight. At the initial stage, weights can be set with random values or using a custom approach. We propose a custom approach to set weights based on the better group of the neuronal network.

This work is focused on searching the weights of a neural network with clusters in nodes with four applied algorithms. Additionally, we discussed from a computing perspective a data set that represents the connections of C. Elegans. Based on the results, we use metrics for the models of algorithms. In addition, in this work, we used the names of neurons according to Brenner [19], which corresponds to more than 282 neurons names.

The paper is structured as follows. Section 2 presents the main concepts of C. Elegans and the data set. Section 3 presents the related work. Section 4 presents the applied algorithms. In Sect. 5, we illustrate the proposed approach. Section 6 presents the results. Finally, Sect. 7 concludes the paper.

2 Neural Network Data Set

Normally, C. Elegans feed on microorganisms, mainly bacteria. There have been different works to design and validate a neuronal map of C. Elegans [17]. The data set of connections used in this work can be downloaded at Wormatlas[1]. C. Elegans has few neurons in comparison with other animals, most neurons can be unambiguously identified by morphology and position [3]. In addition, according to Brennen, neurons have a name convention [19].

Supervised and unsupervised algorithms can be used in machines to simulate intelligence. Supervised search algorithms find the best node based on a heuristic. The deployment in a search algorithm represents the possible paths that an agent can take when trying to find the best solution [10]. In consequence, it is interesting to observe the process in animals for finding possible ways to make artificial intelligence.

Generally, applying a cluster in a group of elements serves a problem in which the total characteristics are not set yet. Clustering is a division of data into groups or similar objects [1]. As an unsupervised classification technique,

[1] https://www.wormatlas.org/.

clustering identifies some inherent structures present in a set of objects based on a measure of similarity [5]. Before of search the clusters in node connections, we applied a transformation in data to discretize and then use data in each algorithm.

2.1 Connection Details

The electrical stimulus is called gap junctions and works as a relation between neurons. Another process is chemical synapses between neurons that correspond to a receiver and transmitter a signal. Thus, neuron A can have various links with neuron B. Also, neuron B can have links with neuron A.

C. Elegans neurons can be classified into 118 classes by morphology, dendritic specialization, and connectivity. There are the following three categories based on functional and structural properties [11]:

- **Sensory**. It responds to environmental conditions.
- **Inter**. Interconnects with other neurons.
- **Motor**. It has neuromuscular junctions.

The data set presents the connectivity of neuronal connections. Thus, the data set presents the number of synapses per neuron, its synaptic partner, and synapse type (incoming chemical synapse, outgoing chemical synapse, electrical junction, or neuromuscular junction). The data set contains 6417 records, where each row contains three categorical attributes and one numeric attribute. Nevertheless, this work uses 6414 records because three rows were removed. Removed records are explained in the next sections. The categorical attributes are *Neuron A* that indicates the point of origin, *Neuron B* that indicates the target neuron, *Type of Synapses* and *Links (NBR)*. *NBR* is a number of synapses between neurons A and B.

2.2 Data Preparation and Transformation

The data does not have missing values and the numeric column *Links NBR* does not have out-of-range or abnormal values. To check the range of values in *NBR*, a limit was created and calculated using Eq. 1.

$$limit = mean \pm 3.4 * standard_deviation \tag{1}$$

The max limit was *13*, while the min limit was *-8*; however, negative values were abnormal values in this case. Then, the range was between *0* and *13* for *NBR*. The connections can be modeled as a network or directed graph.

To implement the algorithms, the categories (neuron name A and neuron name B) were converted to numbers. We implemented a transformer[2]. The transformer was an ordinal type, where the characteristics are converted into ordinal integers. The result is a single column of integers from 0 to $n - 1$ features, where

[2] Sklearn Ordinal Encoder.

n is the number of features. Normally, machine learning algorithms tend to require numbers instead of strings as inputs [13]. So, we needed some methods of coding to convert categories to numbers.

2.3 Data Analysis

Figure 1 shows the structure of the network cell connections of C. Elegans, which has been generated using the dataset mentioned above. This network was created iterating each connection and connecting the nodes transmitters with the receivers. Furthermore, it shows that there are many neurons directed to the neuromuscular junction (NMJ) node. NMJ helps with the locomotive movement of C. Elegans.

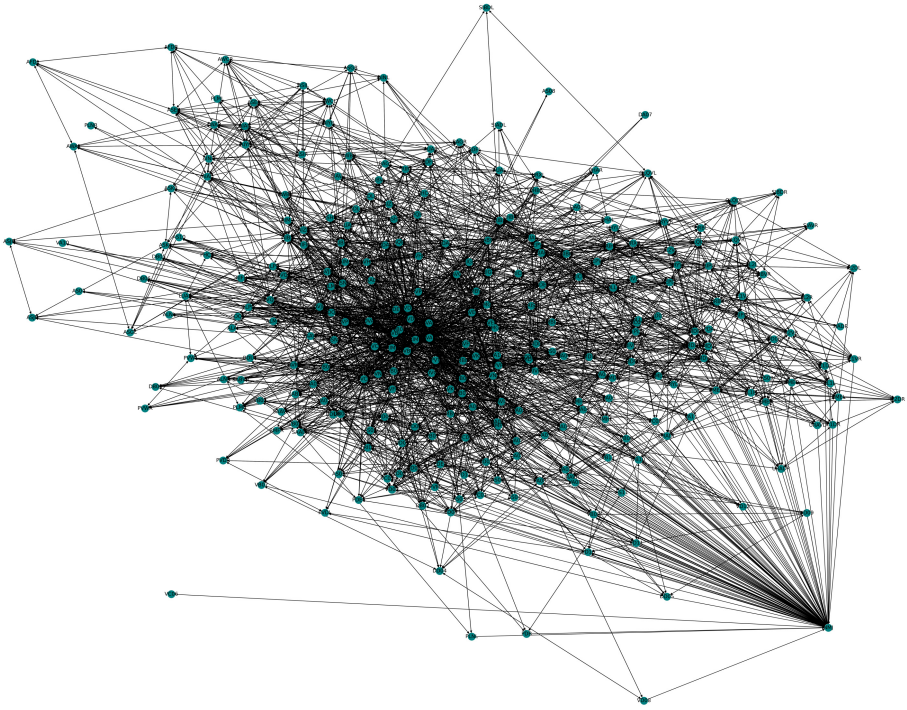


Fig. 1. Network connection structure

Same Origin. In the data set, the links that are connected to the same point were eliminated (i.e., point of origin is connected to the same point), since they will not generate additional characteristics. Table 1 shows the removed connections where the source node is equal to the end node. When the data was analyzed, these nodes do not represent additional features; therefore, the next neurons (RIBL, RIBR, and VA08) were removed. These removed nodes have elec-

Table 1. Deleted connections (A = B)

Node A	Type neuron	NBR
RIBL	EJ	1
RIBR	EJ	1
VA08	EJ	1

trical junctions (EJ). RIBL and RIBR are interneurons, while VA08 is a motor neuron.

Degree Centrality. The degree centrality of a node corresponds to the number of edges it has. The most central node is the one with a higher degree [8]. Centrality describes the importance of a node in the entire network. Thus, the degree centrality is a simple count of the total number of connections attached to a vertex. Figure 2, presents the degrees centrality of the nodes, where the highest peak is presented for the NMJ neuron node. In this figure, we can observe that the NMJ neuron is part of the core of the network since it is the most central node. In contrast, most of the other nodes have a degree of centrality between 0.05 and 0.10.

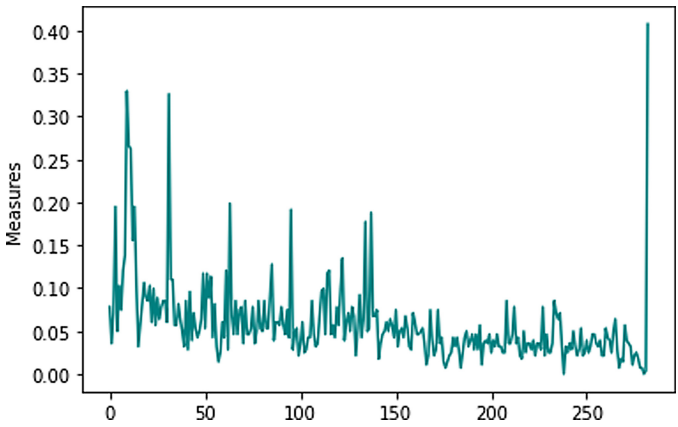


Fig. 2. Measures of centrality in the network

Isolated Nodes. Based on the graph, it is possible to find isolated nodes, which are those nodes that are not connected to the graph or network. In this case, two nodes were not connected to the network, which is neuron AVFL towards AVFR neuron. These nodes are neurons with type interneuron.

3 Related Work

Complex graph networks often represent relationships between sets of entities. One of the main questions in understanding a data set is knowing how many communities there and its memberships. [20]. As of the date of the article, we did not find an approach to generate the weights of the C. Elegans neural network. Next, we present related works about the analysis of C. Elegans neural network.

Bertin et al. [12] mapped a large fraction of the Caenorhabditis Elegans interactome network of protein-protein interaction. The authors examined the C. Elegans interactome network for the presence of highly connected neighborhoods by determining the mutual clustering coefficient between proteins in the network. C. Elegans is an ideal model for studying how protein networks relate to multicellularity. The authors analyzed the network of representing proteins and calculated the percentage of protein interaction pairs that share embryonic lethal phenotype.

Froese et al. [14] present a self-optimization process that has been replicated in various neural network formalisms. Nevertheless, it is not yet clear whether it can be applied to more biologically realistic network topologies and scale to larger networks. The authors demonstrated the self-optimization capacity for the C. Elegans connectome. In addition, they explored the possible beneficial effects of inhibitory connections between groups. They conclude that the self-optimization process can be applied to neural network topologies characterized by greater biological realism and that long-range inhibitory connections can facilitate the generalizability of the process.

Chen et al. [4] formulated an inference for graphs that represent a neural map of chemical and electrical connections. The formulation consists of a set of own graph inferences from the perspectives of the coincidence of graph matching (i.e., a set of edges that do not have a set of common vertices) and the classification of joint vertices. They concluded that jointly in the chemical and electrical connectivity maps, there is a conserved statistically significant similarity in the two synaptic connectome structures.

Bacik et al. [2] analyzed the weighted and directed C. Elegans connectome from a flow-based perspective. The authors' methods use graphical diffusion processes as a simple means of linking directed network characteristics and propagation dynamics. The analysis uses data from C. Elegans to represent the connectome of C. Elegans, we use the two-dimensional network design, that is, the neurons are placed in the plane according to their normalized Laplacian eigenvector (x-axis) and the processing depth (y-axis).

4 Algorithms

In this work, we created several models for each algorithm. Models are explained in Sect. 5. The models were created with Scikit Learn[3]. Scikit Learn is a software

[3] https://scikit-learn.org/.

tool package for Python[4] language. Scikit Learn permits the use of different algorithms related to data science.

4.1 Spectral Clustering

Spectral clustering has been successful in a wide variety of applications providing significant results at reasonable costs [22]. Spectral Clustering is used to group relevant data points. It also takes care of separating irrelevant data points. This algorithm is formulated like the partition of a graph, which is mapped by data points. It is called spectral since it is based on the theory of spectral graphs.

4.2 Density-Based Spatial Clustering (DBSCAN)

The DBSCAN algorithm is called a density-based spatial grouping for noisy data. It finds high-density core samples and expands groups from them. It is a successful algorithm for data containing groups of similar density [6].

DBSCAN looks at the clusters as areas that have high density and are separated by other areas that are of low density. It is the reason why the groups found by this algorithm can obtain the shape they want, unlike other algorithms, where their groups take a specific form. To define the word dense, we can include two parameters in the algorithm, which are *min_samples* and epsilon (EPS). EPS is the maximum distance to consider the neighborhood between two samples. Each one defines a density level, where *min_samples* is high and EPS is low indicating the density that is necessary to form a group.

The *min_sample* parameter is in charge of controlling the tolerance of the algorithm towards the noise, while the EPS parameter is in charge of choosing the data set and the function that determines the distance, which generally cannot take a default value. In addition, EPS controls the proximity of the points. When these points are chosen very small, most of the data cannot be grouped correctly, but if they are chosen very large, they tend to join with other clusters; then, all data is returned in a single cluster.

4.3 Mini Batch K-Means

Mini Batch is a variation of the K-Means algorithm that uses mini-batches to reduce computation time while trying to optimize the same objective function. Mini-batches are subsets of the input data obtained randomly in each training iteration. Mini batches drastically reduce the amount of convergence execution for a local solution.

4.4 Clustering Feature Tree (CFT)

This algorithm builds a tree called the Clustering Feature Tree (CFT) for the given data. The CFT algorithm has two parameters, the threshold, and the

[4] https://www.python.org/.

branching factor [21]. The branching factor limits the number of subgroups in a node and the threshold limits the distance between the incoming sample and existing subgroups. The algorithm creates subclusters called CF that contain the information necessary for clustering.

5 Proposed Approach

To implement these algorithms, different models were created to find the best grouping result. Six models were created for each algorithm varying the parameters according to each model. Table 2 presents 24 models and its parameters per algorithm. Each model was enumerated.

Table 2a presents the experimentation to find the best model for the data addressed. All DBSCAN models use the *ball tree* method to find point distances and to find the closest neighbors. *Ball tree* is a spatial partition data structure to organize points in a multidimensional space. The first parameter to vary is the maximum distance (EPS) between two samples for one to be considered in the vicinity of the other. The other parameter to vary is the number of samples (or total weight) in a neighborhood for a point to be considered as a central point. This includes the point itself.

Table 2b presents the experimentation to find the best model for the data approached with the Spectral Clustering algorithm. The parameter to be validated is the space that defines the dimension of the projection subspace. The

Table 2. Parameters in Models

Model	Max. Distance (EPS)	BDSCAN Weights
M1	0.2	10
M2	0.3	10
M3	0.4	10
M4	0.2	20
M5	0.3	20
M6	0.4	20

(a) BDSCAN

Model	Space	Neighbors
M7	28	20
M8	38	20
M9	48	20
M10	28	30
M11	38	30
M12	48	30

(b) Spectral Clustering

Model	Threshold	Branching factor
M13	0.2	40
M14	0.2	50
M15	0.5	60
M16	0.5	70
M17	0.8	80
M18	0.8	90

(c) CTF

Model	Num. clusters	Batch size
M19	3	20
M20	3	40
M21	8	20
M22	8	40
M23	10	20
M24	10	40

(d) MiniBatch K-Means

other parameter to vary is the number of neighbors to use when constructing the affinity matrix using the nearest neighbor method.

Table 2c presents the models for the CFT algorithm. Each model presented a variation in two parameters, the threshold, and the branching factor.

Table 2d presents the models for mini-batch K-Means. The parameters that varied were the number of clusters and the batch size. The initialization method k-means++ was used in each model, which selects the initial cluster centers for the clustering of k-mean in an automatic way to accelerate convergence.

5.1 Weights Calculations

With the best cluster model, the weights for edges were calculated. Equation 2 serves to make the weight calculations. M12 was chosen as the best model compared to the other models and based on the results of the metrics. The number of clusters for M12 is 48. Each cluster was given a score that corresponds to the total number of members in the cluster.

$$wi_j = 0.2 * cluster_score + \frac{Tn}{Tf + NBR} \tag{2}$$

In Eq. 2, NBR is the number of synapses, Tn is the number of appearances of that neuron in the connections, Tf is the total number of neurons or characteristics, and $O.2$ is a constant value associated with a fixed weight for each connection.

6 Results

In this section, the results of the models are presented. To compare each model, four metrics were established for the results of the groups. The chosen metrics do not need classes or target variables compared to forecasting algorithms. The four applied metrics are:

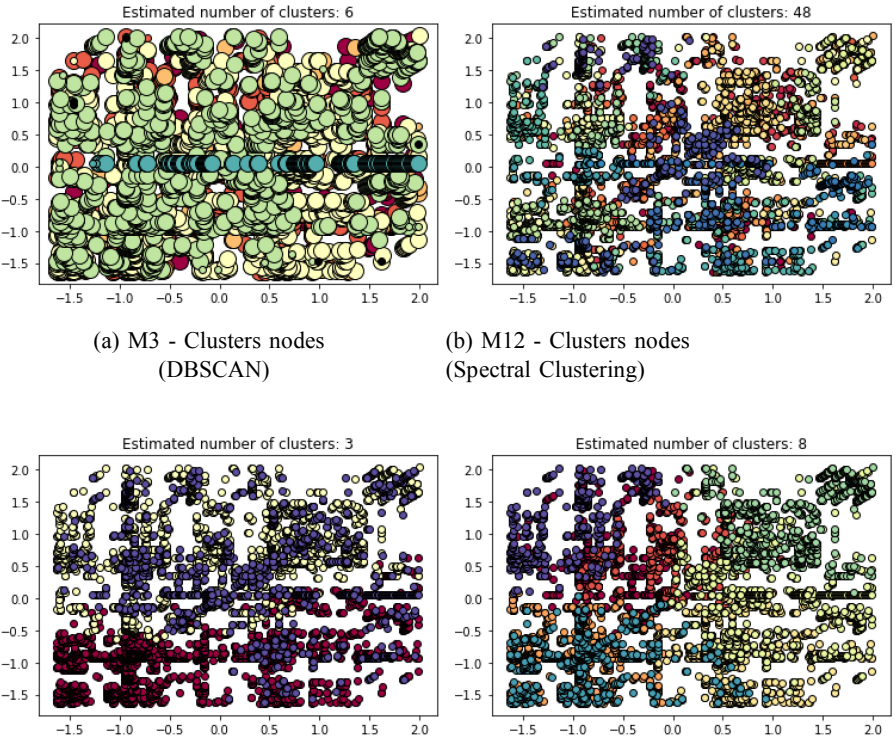
- **Silhouette coefficient**. It is calculated using the mean distance within group (a) and the closest mean distance to group (b) for each sample. The coefficient ranges from 1 to −1, where 1 is the best result.
- **Calinski harabasz**. It is known as the variance ratio criteria and defines the relationship between the sum of the dispersion between groups and the dispersion between groups for all groups, where the dispersion is defined as the sum of the squared distances. The highest value is the one chosen as the best.
- **Davies bouldin**. It stands for the average "similarity" between groups, where similarity is a measure that compares the distance between groups with the size of the groups. Values closest to zero perform the best.
- **Composed Density between and within clusters (CDBW)**. A reliable cluster validity index should match all "good" clustering requirements. This implies evaluating the cohesion of the groups, as well as the separation of the groups along with their compactness [9]. The lowest value is the one that performs best.

Table 3 presents the results of the metrics for each model. For the DBSCAN algorithm, the best model is M3, for the Spectral Clustering algorithm the best model is M12. CFT algorithm's best model is M13. Finally, in Mini Batch K-Means best model is M21.

Figure 3 represents the deployment of the nodes in the clusters for the best model chosen by an algorithm. The M12 model has a greater grouping. The sub-figures present a cluster grouping for each algorithm, where the nodes categories were transformed to ordinal numbers (i.e., the number associated with categories). Therefore, the X-axis can be seen as nodes A and the Y-axis as nodes B. Figure 3a presents a model M3 that has 6 clusters. Figure 3b presents a model M12 that has 48 clusters. Figure 3c presents a model M13 that has 3 clusters. Figure 3d presents a model M21 that has 8 clusters.

Table 3. Metrics results for models

Model	Silhouette	Calinski harabasz	Davies bouldin	CDBW
M1	−0.077	103.897	1.496	0.318
M2	−0.145	184.323	2.244	0.348
M3	0.046	554.804	3.337	0.821
M4	0.041	113.346	1.452	0.330
M5	−0.146	159.847	1.904	0.305
M6	−0.091	288.206	2.891	0.554
M7	0.191	1238.900	1.327	0.300
M8	0.236	1329.701	1.181	0.261
M9	0.253	1221.748	1.085	0.233
M10	0.198	1195.593	1.288	0.302
M11	0.250	1357.473	1.153	0.250
M12	0.285	1602.309	1.060	0.233
M13	0.247	2096.772	1.342	0.772
M14	0.245	1951.496	1.377	0.788
M15	0.271	2339.545	1.333	0.779
M16	0.271	2339.545	1.333	0.779
M17	0.271	2346.092	1.298	0.941
M18	0.271	2346.092	1.298	0.941
M19	0.265	2437.222	1.327	0.760
M20	0.297	2768.210	1.198	0.756
M21	0.301	2651.399	1.005	0.521
M22	0.295	2569.843	1.119	0.547
M23	0.296	2466.860	1.053	0.487
M24	0.289	2402.525	1.068	0.507



(a) M3 - Clusters nodes (DBSCAN)

(b) M12 - Clusters nodes (Spectral Clustering)

(c) M13 - Clusters nodes (CFT)

(d) M21 - Clusters nodes (Mini Batch K-Means)

Fig. 3. Best models chosen by algorithm

7 Conclusions and Future Work

C. Elegans is an animal model for implementing works about network cell structure with a computational perspective. Drawing network connections permits showing a structure and analyzing the main connections and nodes. Thus, The network can be represented as a directed graph.

In this work, we present an approach to calculate the weights of the connections according to the groupings between the connections. Thus, through grouping, the calculation of the weights was proposed. Different models were tested before the final models were established. The results allow establishing that for the case, the 6414 connections can be grouped according to their characteristics in 48 clusters. For this problem and data set, it is observed that DBSCAN had the worst results compared to the other algorithms. In our context, the CDBW and the silhouette coefficient were the most determining metrics to establish better models.

As future work, the weights might allow analyzing network structure. In addition, the future objective is to observe how the weights affect the propagation in a signal network of C. Elegans. Thus, we will create an interactive system that can allow seeing the structure of the network, the weights, and the propagation of the signals (chemical and/or electrical).

References

1. Abbas, O.A.: Comparisons between data clustering algorithms. Int. Arab J. Inf. Technol. (IAJIT) **5**(3) (2008)
2. Bacik, K.A., Schaub, M.T., Beguerisse-Díaz, M., Billeh, Y.N., Barahona, M.: Flow-based network analysis of the Caenorhabditis Elegans connectome. PLoS Comput. Biol. **12**(8), e1005055 (2016)
3. Bargmann, C.I., Horvitz, H.R.: Chemosensory neurons with overlapping functions direct chemotaxis to multiple chemicals in c. elegans. Neuron **7**(5), 729–742 (1991)
4. Chen, L., Vogelstein, J.T., Lyzinski, V., Priebe, C.E.: A joint graph inference case study: the c. elegans chemical and electrical connectomes. In: Worm, p. e1142041. Taylor & Francis (2016)
5. Du, K.L.: Clustering: a neural network approach. Neural Netw. **23**(1), 89–107 (2010)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
7. Florez, H., Cárdenas-Avendaño, A.: A computer-based approach to study the Gaussian moat problem. In: Florez, H., Misra, S. (eds.) ICAI 2020. CCIS, vol. 1277, pp. 481–492. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61702-8_33
8. Golbeck, J.: Analyzing the Social Web. Newnes (2013)
9. Halkidi, M., Vazirgiannis, M.: A density-based cluster validity approach using multi-representatives. Pattern Recogn. Lett. **29**(6), 773–786 (2008)
10. Hernandez, J., Daza, K., Florez, H.: Alpha-beta vs scout algorithms for the Othello game. In: CEUR Workshops Proceedings, vol. 2846 (2019)
11. Kim, J., Leahy, W., Shlizerman, E.: Neural interactome: interactive simulation of a neuronal system. Front. Comput. Neurosci. **13**, 8 (2019)
12. Li, S., et al.: A map of the interactome network of the metazoan C. elegans. Science **303**(5657), 540–543 (2004)
13. McGinnis, W.D., Siu, C., Andre, S., Huang, H.: Category encoders: a Scikit-learn-contrib package of transformers for encoding categorical data. J. Open Sour. Softw. **3**(21), 501 (2018)
14. Morales, A., Froese, T.: Unsupervised learning facilitates neural coordination across the functional clusters of the C. elegans connectome. Front. Robot. AI **7**, 40 (2020)
15. Sanchez, D., Florez, H.: Improving game modeling for the Quoridor game state using graph databases. In: Rocha, Á., Guarda, T. (eds.) ICITS 2018. AISC, vol. 721, pp. 333–342. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73450-7_32
16. Towlson, E.K.: Caenorhabditis elegans and the network control framework–FAQs. Philos. Trans. R. Soc. B: Biol. Sci. **373**(1758), 20170372 (2018)

17. Varshney, L.R., Chen, B.L., Paniagua, E., Hall, D.H., Chklovskii, D.B.: Structural properties of the Caenorhabditis elegans neuronal network. PLoS Comput. Biol. **7**(2), e1001066 (2011)
18. Velosa, F., Florez, H.: Edge solution with machine learning and open data to interpret signs for people with visual disability. In: CEUR Workshop Proceedings, vol. 2714, pp. 15–26 (2020)
19. White, J.G., Southgate, E., Thomson, J.N., Brenner, S.: The structure of the nervous system of the nematode Caenorhabditis elegans. Philos. Trans. R. Soc. Lond. B Biol. Sci. **314**(1165), 1–340 (1986)
20. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: Proceedings of the 2005 SIAM International Conference on Data Mining, pp. 274–285. SIAM (2005)
21. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. ACM SIGMOD Rec. **25**(2), 103–114 (1996)
22. Zhuzhunashvili, D., Knyazev, A.: Preconditioned spectral clustering for stochastic block partition streaming graph challenge (preliminary version at arxiv.). In: 2017 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–6. IEEE (2017)
23. Zuluaga, J.Y., Yepes-Calderon, F.: Tensor domain averaging in diffusion imaging of small animals to generate reliable tractography. ParadigmPlus **2**(1), 1–19 (2021)