

# Structural analysis of the Zebrafish connectome

**Karan Kabbur Hanumanthappa Manjunatha (1236383), Nora Nikoloska(2013006), Iriarte Delfina (1231682)**

March 29, 2021

---

## Abstract

In this project, the structural network properties of the connectome of larval zebrafish are analyzed. The network consists of 3163 sequenced neurons and their connections. The network analysis includes: computation of some general network properties, measuring degree distribution and assortativity, computation of different centralities, community detection with various algorithms, robustness of the network with different forms of attack, results commentary and visualisation.

---

## 1 Introduction

The larval zebrafish is increasingly being used to address fundamental questions of nervous-system organization and circuit function at the whole-brain level yet its neuro-anatomy is still relatively uncharted.

Kunst et al [1] generated a digital atlas of the larval zebrafish brain containing over 2,000 individually GFP-labeled neurons. Prof. Marco Dal Maschio (UNIPD and also co-author of [1]) provided us a new dataset consisting of information of 3163 neurons in SWC format [2].

## 2 Theory

In this section, the dataset is described and all of the computed network measures are briefly explained.

### 2.1 Dataset

The dataset consists of 3163 SWC files [2] corresponding to each neuron of larval zebrafish. Each file with the name format as *name-of-the-neuron.swc*. In each of the file, the first row which consists of 1 in the second column representing the soma/cell body of neuron while the remaining rows which has 6 in the second column represents "end points". However, it should be noted that there is no distinction done between which end points it is being referred to i.e. it can be either axonal terminals or dendrites. Columns 3, 4 and 5 represents x, y and z spatial coordinates, given in micrometers for the corresponding structure(i.e. either soma or endpoints). While column 7 refers to radius for soma, it instead refers to thickness radius for the dendritic endpoints with unit of measurement again in micrometers. The last column consists of numbering based on parent-child samples with next higher number representing a parent with respect to lower number, which is considered a child of it.

### 2.2 Network Measures

A small description of each of the basic network properties is provided to guide the reader, however it is given in depth in the literature of Network Science [3]. Networks can be **undirected** or **directed**. In directed nodes all edges have a source and target node. In addition, if each edge is associated with a weight, the network is **weighted**. The number of edges of a given node is node's **degree**. By calculating the degree distribution of a network the most and least connected nodes can be pointed out. For directed graphs, both in-degree and out-degree are calculated. Another very useful network property is the **shortest path**(or average distance or average shortest path length, all mean the same). This is defined as the path with the fewest number of edges. The path's length is the number of edges in the path. For the whole network, the average shortest path length is computed between all pairs of nodes. The **diameter** of the network is defined as the longest shortest path in a graph, or the distance between the two farthest nodes. Regarding network connectivity, the network is considered **connected** if a path exists from every node to every other node. In a disconnected network, it is often the case that one big component is present in which there is a path between every two nodes. This largest connected

component is referred to as **Giant Component** (GC). The degree to which the neighbours in a network are connected to each-other can be measured with the **clustering coefficient**:

$$C_i = \frac{2L_i}{k_i(k_i - 1)} \quad (1)$$

where  $L_i$  represents the number of links between the  $k$  neighbors of node  $i$  [3]. The **average clustering coefficient**  $\langle C \rangle$  of the whole network is usually computed to measure the degree of clustering. To explore the connectivity of the network further, the number of **cliques** can be computed. They are defined as complete subgraphs where each node is connected to every other.

## 2.3 Assortativity

The hubs (nodes with very high degree) tend to link to each other and avoid linking to small-degree nodes while small-degree nodes tend to connect to other small-degree nodes. But it is the opposite in the case of disassortative networks while in neutral networks the wiring is completely random. [3]

To classify whether our network is assortative, disassortative or neutral it can be visualized the degree correlation matrix  $e_{ij}$  which is probability of finding a node with degrees  $i$  and  $j$  at the two ends of a randomly connected links. For assortative networks it is expected to find high values of  $e_{ij}$  along the main diagonal while for disassortative networks values are high along secondary diagonal instead. However, for neutral networks there is not any correlation and the  $e_{ij}$  values are symmetric around the average degree. This just gives us a visual inspection of the type of assortativity that should be expected from our network. However, to measure degree correlation there is an alternative approach.

$K$  nearest neighbours or Average degree connectivity of graph( $k_{nn}(k)$ ) is simply the average nearest neighbours with degree  $k$ . In mathematical terms:

$$k_{nn}(k) = \sum_{k'} k' P(k'|k) \quad (2)$$

where  $P(k'|k)$  is the conditional probability starting from a link of  $k$  degree reaching a node of  $k'$  degree[3]. From the log-log plot of  $k_{nn}(k)$  as a function of  $k$ , the equation

$$k_{nn}(k) = Ck^\mu \quad (3)$$

is to be fit so as to obtain the parameter  $\mu$  known as *correlation exponent*. The sign of this exponent decides the type of assortativity observed in our network. If  $\mu > 0$  then it is an **Assortative network**. When  $\mu = 0$  representing that  $k_{nn}$  is independent of  $k$ , it can be conclude that this is a **Neutral network**. Lastly, if  $\mu < 0$ , this corresponds to a **Disassortative network**.

## 2.4 Robustness

Robustness refers to the ability of the network to withstand the failures or perturbations caused by either directed attack on the hubs or random removal of nodes in the network [3]. In the case of directed attack on the hubs, the nodes with highest degree are removed first along with their links and further nodes are removed in the decreasing order of the degree.

## 2.5 Efficiency

The efficiency of a network is a measure of how efficiently it exchanges information [5]. Moreover, it is used to help determine how cost-efficient a particular network construction is, as well as how fault tolerant it is. This is divided into 2 types: **Global efficiency** - which quantifies the exchange of information across the whole network where information is concurrently exchanged. **Local efficiency** - Quantifies network's resistance to failures on small scale. Focusing only on

global efficiency for our project, mathematically it is defined as follows [5]:

$$E(G) = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \frac{1}{d(i,j)} \quad (4)$$

$$E_{glob}(G) = \frac{E(G)}{E(G^{ideal})} \quad (5)$$

where n is total nodes in the network,  $d(i,j)$  is the length of the shortest path between node i and node j,  $G^{ideal}$  is the ideal graph on n nodes where all possible edges are present which is same as clique but instead all the nodes of the network are taken into consideration here and finally computing  $E_{glob}(G)$  give us the Global efficiency where all the nodes are exchanging packets of information with each other.

## 2.6 Community detection

Detecting communities in networks provides a means of coarse-graining the complex interactions of the network. It should be mentioned that since our network is directed, the algorithms used were only restricted to those that can be implemented in such network. In particular the following algorithms were implemented:

### 1. Based on dynamics:

- *Infomap*: based on the concepts of information theory; it uses the probability flow of random walks on a network as a proxy for information flows in the real system and it decomposes the network into modules by compressing a description of the probability flow [10].
- *Walktrap*: Walktrap uses short random walks to merge separate communities. The general idea is that if you perform random walks on the graph, then the walks are more likely to stay within the same community because there are only a few edges that lead outside a given community [11].
- *Label Propagation*: The intuition behind the algorithm is that a single label can quickly become dominant in a densely connected group of nodes, but will have trouble crossing a sparsely connected region. Labels will get trapped inside a densely connected group of nodes, and those nodes that end up with the same label when the algorithms finish can be considered part of the same community [12].

### 2. Based on Statistical Mechanics:

- *Springlass*: relies on an analogy between a very popular statistical mechanics model called *Potts spin glass*, and the community structure. It applies the simulated annealing optimization technique on this model to optimize the modularity [14]. Based on the statistical mechanics of spin around the network, community detection can be interpreted as finding the ground state of an infinite range spin glass [13].
- *RBER*: A variant of the previous method that instead of a configuration null-model [16]. uses a Erdös-Rényi (ER) null-model in which each edge has the same probability of appearing [16].

### 3. Based on Optimisation:

*Louvain* maximizes a modularity score for each community, where the modularity quantifies the quality of an assignment of nodes to communities. This means evaluating how much more densely connected the nodes within a community are, compared to how connected they would be in a random network. The partitions obtained through these methods maximize intra-community edge weights relative to a specific random network null model [15]. Several method options are applied:

- *Modularity*: compares the actual graph to the expected graph, taking into account the degree of the nodes [16].
- *CPM*: compares to a fixed resolution parameter, so that it finds communities that have an internal density higher than the resolution parameter, and is separated from other communities with a density lower than the resolution parameter [16].
- *Significance*: This is a probabilistic method based on the idea of assessing the probability of finding such dense subgraphs in an (ER) random graph [16].
- *Surprise*: Another probabilistic method, but rather than the probability of finding dense subgraphs, it focuses on the probability of so many edges within communities [16].

In order to get a measure of the strength of each community structure found by the several algorithms, the modularity Q is calculated for the several communities. This quantity is defined as:

$$Q = \frac{1}{2m} * \sum_{ij} A_{ij} - \frac{k_i k_j}{2m} \delta(c_i, c_j) \quad (6)$$

where  $m$  is the number of edges,  $A_{ij}$  is the element of the  $A$  adjacency matrix in row  $i$  and column  $j$ ,  $k_i$  is the degree of node  $i$ ,  $k_j$  is the degree of node  $j$ ,  $c_i, c_j$  are the types of the two vertices and  $\delta$  is the delta of Dirac. This quantity measures the fraction of the edges in the network that connect vertices of the same type (i.e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices. Values approaching  $Q = 1$ , which is the maximum, indicate strong community structure. In practice, values for such networks typically fall in the range from about 0.3 to 0.7. Higher values are rare [17].

Furthermore, in order to compare the communities several distance metrics were used:

- **Variation of information (VI):** measures the amount of information lost and gained in changing from clustering  $C$  to clustering  $C'$ . In particular, the VI is positive, symmetric and obeys the triangle inequality. Thus, surprisingly enough, it is a true metric on the space of clustering types [18].
- **Normalised mutual information measure:** the mutual information (MI) of two random variables is a measure of the mutual dependence between the two variables. More specifically, it quantifies the "amount of information" obtained about one random variable through observing the other random variable. The concept of mutual information is intimately linked to that of entropy of a random variable, a fundamental notion in information theory that quantifies the expected "amount of information" held in a random variable. The normalised mutual measure is more discriminatory and appears more sensitive to errors in the community identification procedure [19].
- **Split-join distance:** The split-join distance is a distance measure defined on the space of partitions of a given set. It is the sum of the projection distance of one partition from the other and vice versa. Split/Join is easily interpretable as the number of nodes that need to be moved to obtain one cluster from the other [20].

Finally, the network community profile (NCP) of each community is computed, which characterizes the “best” possible community according to the conductance measure. The **conductance**  $\phi$  is the quotient of the cut size and the smaller of the volumes of the two sets [21]. By this measure, the best communities are densely linked sets of nodes attached to the rest of the network via few edges. Intuitively, the NCP plot measures the quality of the best possible community in a large network as a function of the size of the purported community. Formally, the conductance may be defined as the value of the best conductance set of cardinality  $k$  in the entire network, as a function of  $k$ :

$$\Phi(k) = \min_{S \in V, |S|=k} \phi(S) \quad (7)$$

Just as the conductance of a set of nodes provides a quality measure of that set as a community, the shape of the NCP plot provides insight into the community structure of a graph. For example, the magnitude of the conductance tells us how well clusters of different sizes are separated from the rest of the network [22].

### 3 Methods

#### 3.1 Extracting adjacency matrix from dataset

From the basic knowledge of neurons, it is well-known that axonal terminals of one neuron forms a connection to dendrites of another neuron through a neuronal junction (which is a small gap between axonal terminal and dendrite) known as a "synapse". Since the information provided in the dataset does not differentiate the endpoints (neurites) as dendrites and axonal terminals certain assumptions have been done on how the neurons are connected to each other.

*Assumptions and Algorithm:*

- Given 2 SWC files representing 2 neurons data, let us for eg. name them as neuron A and neuron B. The Euclidean distances were computed from all of the neurites of A to the soma of neuron B.
- By knowing that radius of soma of all neurons is on average  $6\mu m$ , the dendrites are assumed to be able to spread at most  $30\mu m$  from the center of soma.
- If the Euclidean distance is less than or equal to  $30\mu m$  in length, then they can be imagined as axonal terminals of A within the reach of dendrites of neuron B and it is assumed that they form a "synaptic connection".
- The count of such distances less than  $30\mu m$  from all neurites of neuron A to the soma of neuron B, refer as weight of

directed edge/link from node A to node B. This is done again for for neurites of B to soma of A giving us the weighted directed link from B to A.

- To convert this to adjacency matrix representation, if we refer node A as index 0 and node B as index 1, then the element (0,1) of the adjacency matrix represents weight of the directed link from node 0 to node 1 while (1,0) represents the weight of directed link from node 1 to node 0 (or from our simple example above neuron/node B to neuron/node A).
- When this algorithm is applied to all the neurons for all possible combinations of neurons (avoiding self loops), a asymmetric adjacency matrix is finally obtained with 0's along the principle diagonal which can be finally converted to NetworkX weighted directed graph G in Python.

### 3.2 Assortativity in directed network

The earlier definition for assortativity and Eq. 2.3 refers to undirected network. For the directed graph such as ours, the definitions and concept as explained by Praveen et al[6] are used since their measures of assortativeness are more meaningful than the one proposed by Newman[9].

- **In-Assortativity** of a network is the tendency whereby nodes tend to connect with other nodes with similar IN DEGREES.
  - **Out-Assortativity** of a network is the tendency where nodes tend to connect with other nodes with similar OUT DEGREES.
- Thus, to plot and fit the data, the equations 2.3 are modified as:

$$k_{nn}(k_{in}) = C k_{in}^{-\mu} \quad (8)$$

$$k_{nn}(k_{out}) = C k_{out}^{-\mu} \quad (9)$$

and for fitting the data, by taking log on both sides of Eqs. 8 and 9, the equations result in:

$$\ln(K_{nn}(K_{in})) = \mu_{in} \ln(K_{in}) \quad (10)$$

$$\ln(K_{nn}(K_{out})) = \mu_{out} \ln(K_{out}) \quad (11)$$

Finally, using curve fit tools of scipy Python and Eqs.10, 11 linear fitting is performed to obtain the parameters  $\mu_{in}$  and  $\mu_{out}$ .

### 3.3 Robustness:

The algorithm implemented in Python to quantify the robustness of the directed network is as follows:

- In case of directed attacks on the network, if the measure is chosen is centrality, the removal begins with the nodes with highest centrality measure. NetworkX package in python updates the edges too i.e. when nodes are removed the corresponding edges also gets removed. For the random type of attack, the order of removal of the nodes from the graph is random.
- In the first step, for the directed type of attack, given NetworkX graph G (full network), the centrality measure is computed and the nodes are sorted in the decreasing order of the centrality measure. In the case of random attack there is no need to sort.
- Next, the first 100 nodes of the whole network G are removed,
- After the removal of nodes, at every iteration the Giant Component (GC) is computed and updated from the list of connected components. The number of nodes present in that GC is then computed.
- At each iteration of removal of nodes from G, the number of nodes present in GC is stored. In addition, the cumulative number of nodes removed at every step in is recorded.
- At every iteration the centrality measures of G are recomputed (since nodes were removed from it) and then sort them again in the case of directed attack before removing the nodes.
- Finally, the following fractions can be obtained:

$$GC\ fraction = \frac{No.\ of\ nodes\ in\ GC}{Total\ no.\ of\ nodes\ in\ GC@f = 0} \quad (12)$$

$$frac.\ of\ nodes\ removed = \frac{Cumulative\ no.\ of\ nodes\ removed}{Total\ no.\ of\ nodes\ in\ G} \quad (13)$$

and then plot GC fraction vs fraction of nodes removed to visualize how the node removal affects the network of neurons.

#### **Diameter and Average Shortest path length as a measure of Robustness of network:**

The same algorithm as explained above is applied and instead of computing number of nodes of Giant connected component at every step of removal of nodes, the diameter and average shortest path length of the GC are computed. Plots of these provide an idea of how the diameter and average path length vary with respect to removal of nodes based on various types of attacks i.e. attacks based on various centrality measures and random removal of nodes.

### 3.4 Efficiency

The algorithm for computing the efficiency of the network is same as the one explained in Robustness. However, there are 2 main differences in the implementation:

- At every step the NetworkX directed graph G is converted to undirected graph before computing the efficiency value since NetworkX takes only undirected graph as input to `global_efficiency()` function.
- The efficiency is computed on the whole network G instead of Giant Connected Component (GC) used for Robustness.

## 4 Results and discussion

This section focuses on the presentation of the results. The network properties described in the previous section are applied to the zebrafish connectome dataset.

### 4.1 Network Properties

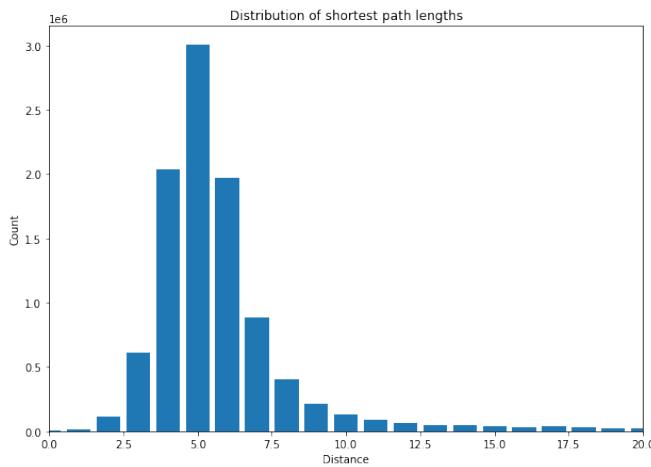
By means of NETWORKX and IGRAPH, which are libraries that can be implemented in Python, the properties of the network are studied. Table 1 shows the results obtained for the main properties of the graph. As already mentioned, this is a **directed** graph that contains 3163 nodes and 349194 edges. The size of the Giant Component (GC), which is defined as the largest connected sub-network of the graph, has 3154 nodes and 349027 edges. Thereafter, it can be said that the graph itself is highly connected. Nevertheless, it should be pointed out that the node and edge connectivity of the GC is only 1, meaning that only one node/edge is necessary to disconnect the graph.

The obtained density of the graph is 0.0349 meaning that the existing connection are few compared to all possible connections. This is similar to the densities obtained for the normal and cancer protein interaction networks for bone cell which correspond to 0.034 and 0.029 respectively [23]. In fact, [24] provide a list of small world datasets with the correspondent density and it can be seen that the range is  $d \sim [0.01, 0.09]$ . Furthermore, it was found that the graph is not connected, meaning that there are some isolated components, namely: T\_190909\_3\_3, T\_190531\_10\_4, T\_161121\_6\_2 and T\_190711\_1\_1. The diameter of the network is 10 with the farthest two nodes being: T\_190220\_1026xBG\_7\_2 and T\_160708\_HuCxBG\_23\_3.

Figure 1 shows the distribution of the shortest path lengths in the network. As it can be seen, the graph consist of mostly small path length with few long-range connections. In fact, the average value obtained corresponds to 4.1595. This is indeed a phenomena that occur in small world networks and reflects an important role that is how information is transmitted. Furthermore, the average global efficiency of the graph obtained is 0.4775, as it can be seen from Table 1. Hence, the ability of parallel information exchange across the whole network is rather high, which is something that it is expected since the brain is considered as a multi-activity parallel system. The local efficiency obtained for the network was 0.7106 which is a high value that corresponds to the ability of information exchange of the sub-network consisting of itself and its all direct neighbors. It can be concluded that information is transmitted rather quickly reducing brain consumption and working efficiently. A comparison of Zebrafish connectome with respect to a random directed unweighted network of similar size and a real cancer network as investigated by Sahoo *et al* in works[23] is shown in Table 1.

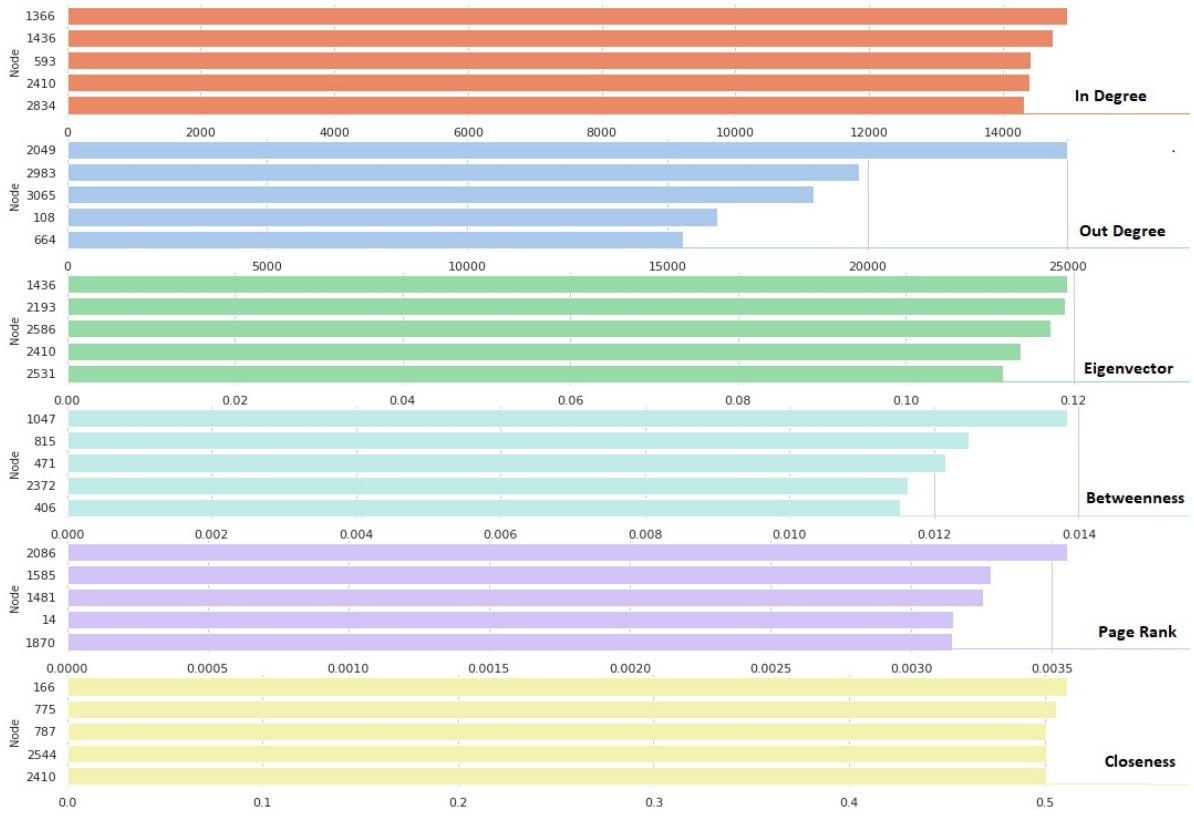
**Table 1.** Main network properties of the Zebrafish connectome in comparison with a similar size directed unweighted random network and Cancer network investigated by Sahoo *et al*[23]

Network Property	Zebrafish	Cancer network	Random network
Graph type	Directed	Undirected	Directed
Number of Nodes(Neurons)	3163	351	3163
Number of Edges	349194	1783	349948
N/W Density	0.03491	0.029	0.035
Diameter	10	7	11
Avg. Shortest path length	4.1595	*	2.03
Global Efficiency	0.4775	*	0.5349
Local Efficiency	0.7106	*	0.4818
Clustering coefficient	0.2947	0.261	0.035
Assortativity	*	0.12	*
In-Assortativity exponent	0.488	*	0.482
Out-Assortativity exponent	0.287	*	0.482
Giant Components (Number of Nodes)	3154	107	3163
Giant Components (Number of Edges)	349027	*	349948
Number of cliques	2264453	149677	1071503
Max clique size	81	15	6



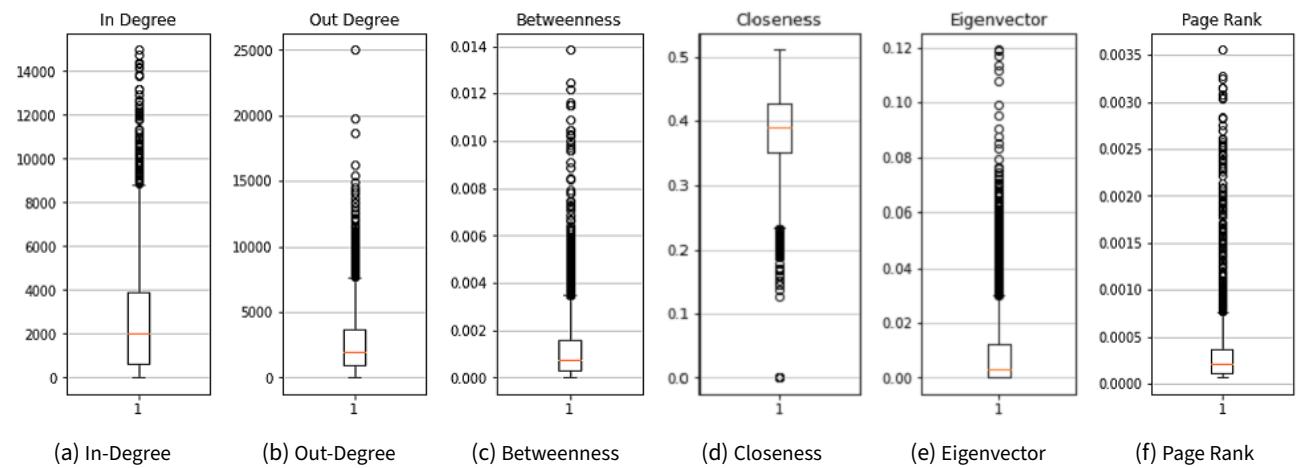
**Figure 1.** Distribution of the shortest path length in the Zebrafish connectome

The clustering coefficient of the Zebrafish connectome is 0.29472 which measures the fraction of all possible triangles presented in the graph reflecting connectivity of a given region to its neighbors. It can be observed that this value is similar to the ones of the normal and cancer protein interaction networks for bone cell [23]; and that it is larger than the values of random graph with the same size  $\sim 0.2$ . Thereafter, the graph has a higher clustering coefficient, which is again a typical property of small-world networks, meaning that the neurons becomes more specialize within highly interconnected functional sub networks leading into locally efficient networks [4]. Furthermore, this can be more evident by noticing that the number of cliques found was 2264453 with the size of the largest one being 81 which it is a large value in comparison with the other networks [23].



**Figure 2.** Top five nodes by centrality

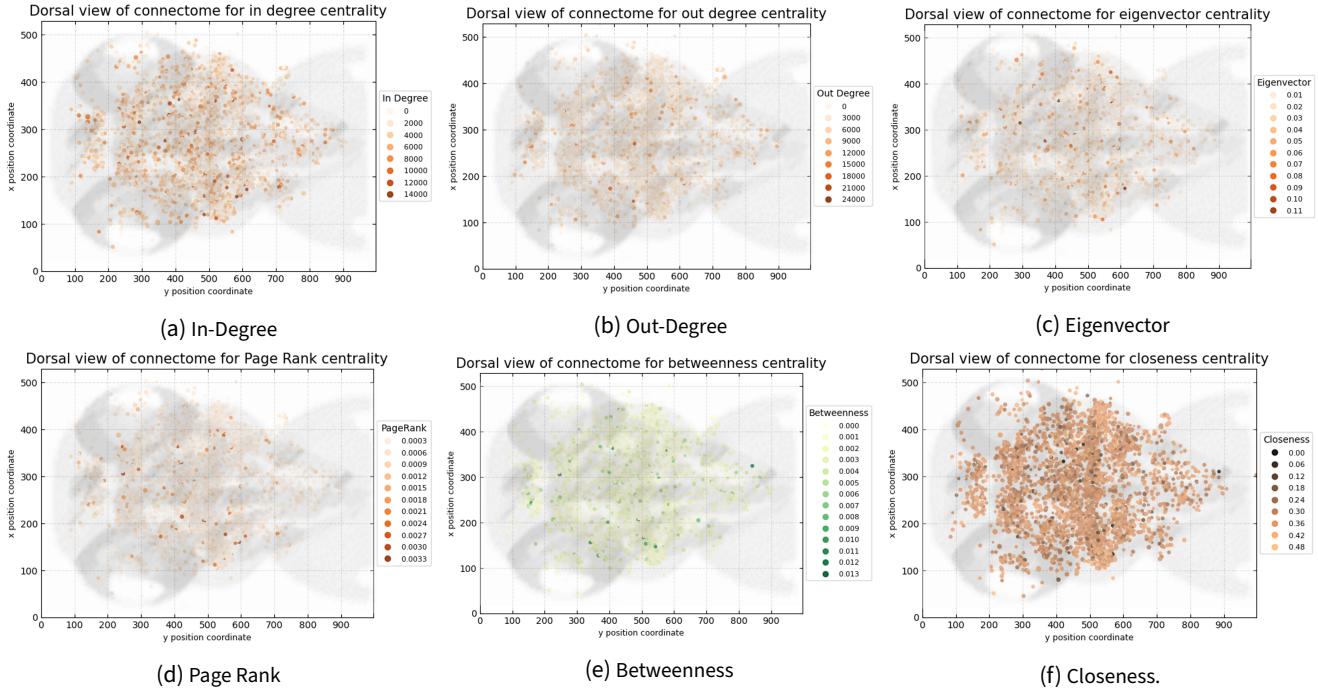
Figure 2 shows the top five node by its correspondent centrality measurement. It can be seen that different nodes are more important than others when computing the different measurements. In particular, the in-degree and out-degree measurements, which are the numbers of incoming and outgoing edges respectively, have a straightforward interpretation: the higher the number the more the node is interacting with other neurons. Thereafter, it is clear that the neuron 166 (neuron name: T\_161129\_19\_1) is the one that receive larger information whereas neuron 329(neuron name: T\_190523\_6\_2) is the node that spread more information in the network. While calculating the in-degree and out-degree values, the weights were also taken into consideration.



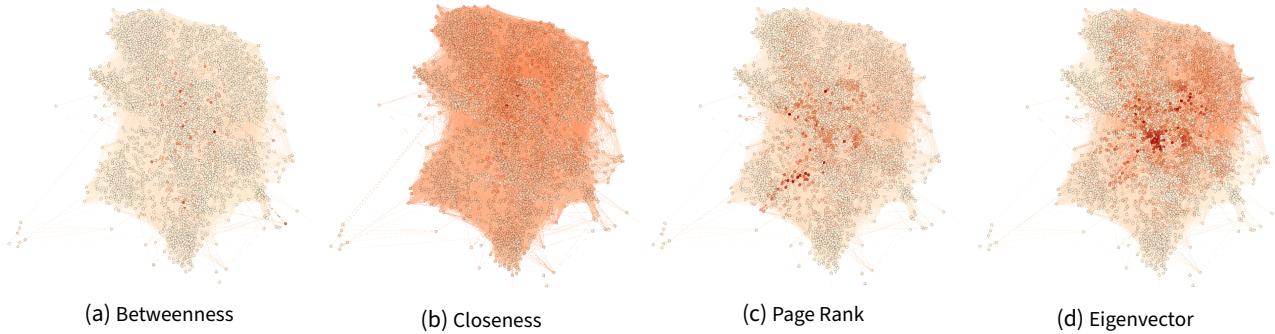
**Figure 3.** Box plots for the distribution of centrality measurements

Figure 3 shows box plots for the distributions for each centrality measure. All of them have a high number of outliers and most have a high positive skew apart from closeness and in-degree centrality. This is especially evident for eigenvector

and Page Rank centralities. We also notice that 50% of the nodes have higher closeness centrality of 0.35 - 0.43. However, for 50% of nodes have betweenness, page rank and eigen vector extremely small. Most of the nodes are seen to be having high in-degree rather than out-degree.

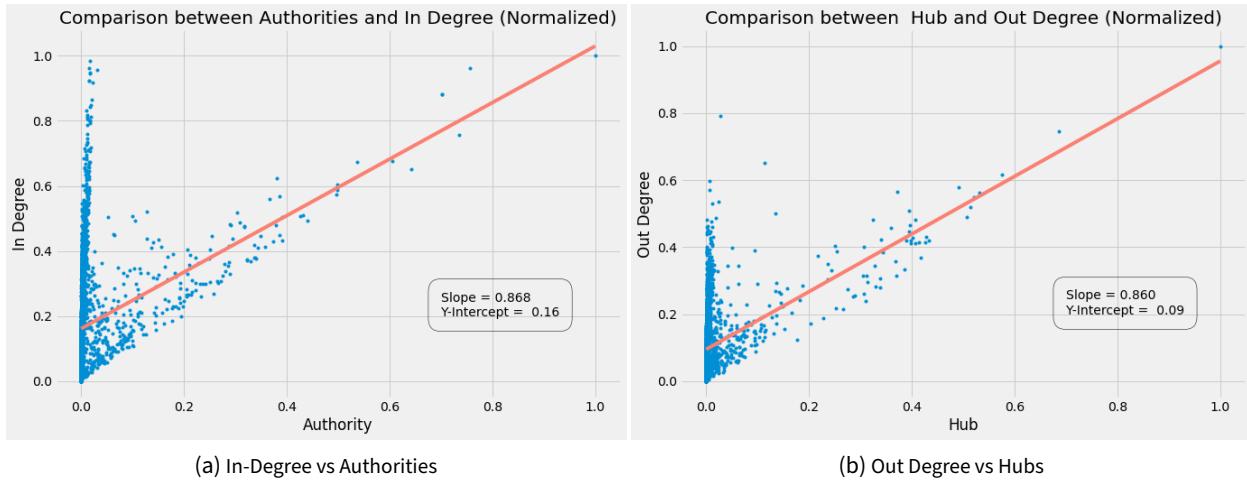


**Figure 4.** Network visualization according to centrality measurements.



**Figure 5.** Visual comparison of the centrality distribution in Gephi

Figure 4a presents a visualization of the graph according to its In-Degree centrality. It is evident that there is a large amount of neurons that are **authorities** (nodes with the highest number of incoming links), whereas by observing the visualization graph by its Out-Degree in Figure 4b, it can be observed that there are only few nodes that are the **hubs** (nodes with the highest number of outgoing links). The concept relating In-Degree and Out-Degree with authorities and hubs seems to be consistent and it is verified by applying the HITS algorithm. In fact, it can be observed in Figure 6 a linear correlation among the parameters. Moreover, it can be easily verified from this graph that there are more neurons with low hub score as already mentioned. Figure 5 shows the different centrality measurements computed in Gephi and visualised using the Force Atlas 2 algorithm. It can be noted that the closeness centrality has overall higher values for most nodes. On the other hand few nodes are high in betweenness centrality. Page Rank and eigenvector produce similar results, with the difference being that in the visualisation more nodes high in eigenvector centrality are grouped together.



**Figure 6.** Comparison between HITS and In/Out Degree centrality measurements.

Figure 4c shows the visualization of the graph according to its eigenvector centrality. It can be seen that in general there are only few nodes with high values. However, these neurons are well spread in the whole brain. Eigenvector centrality is much like degree centrality since it favours nodes that have high correlations with many other nodes but it specifically does it for those that are connected to nodes that are themselves central within the network. Thereafter it can be obtained information about weighting connections to the region based on how connected each other region is. Hence, it can be deduced that regions in the brain are highly connected by just a few neurons.

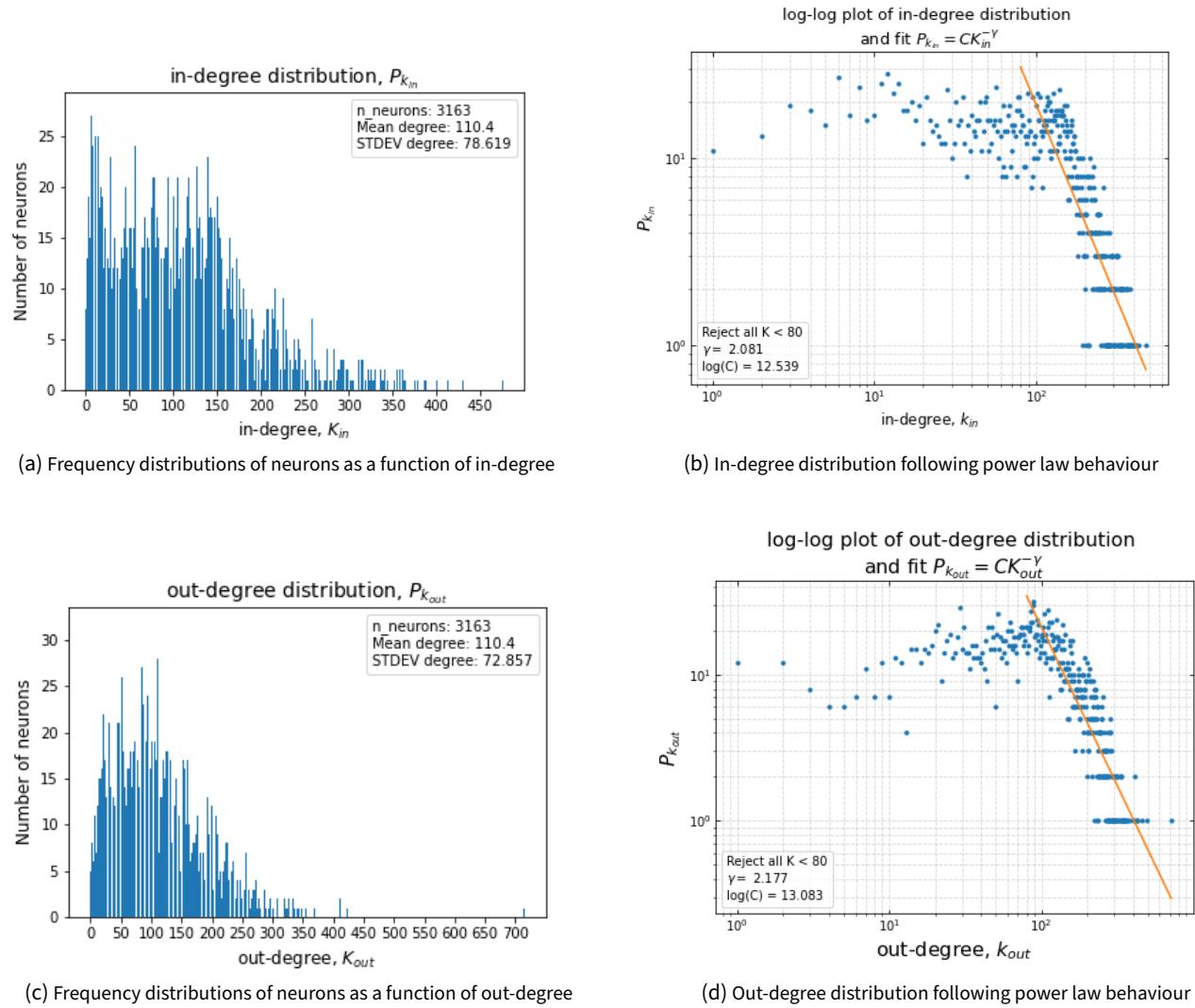
A similar behaviour can also be noticed in Figure 4d and Figure 4e, which represent respectively the brain organized by its Page rank centrality and its betweenness centrality. Page Rank is a method conceptually similar to eigenvector centrality that adjusts for the biased effect of largely connected regions on regions with low connections.

Betweenness centrality quantifies the number of times that a node acts as a bridge along the shortest path between two other nodes, that is, how much is a brain region involved in the flow of information across the brain network. As before, it can be concluded that there are only some meaningful neurons that can spread widely information over all of the network. This is similar to what it was found with the out-Degree centrality. However, it is important to point out that as a whole, information are well spread. This concept can be further speculate by means of the Closeness centrality measurement. It can be noticed from Figure 4f that most of the neurons have high values of this centrality measure. The closeness centrality quantifies how fast a given node in a connected graph can access all other nodes, hence the more central a node is, the closer it is to all other nodes. This means that regions of the brain have an indirect impact on other regions. In general, it can be seen that the whole network can be accessed from a chosen region to any other region.

Interestingly and not surprisingly, it should be point that that most of the neuron with high centrality measures are in the middle of the brain. Indeed, this is something that it is expected since there it can be found the "cerebellum" which plays a central role in coordinating motion and behavior [26]

In conclusion, the Zebrafish connectome acts as a small world network having similar density and clustering coefficients. In particular it was found that the efficiency of the graph is really high, leading into a high information flow among the networks. This was further evidence by the fact that the average path length was short leading into a high degree of closeness centrality. It was found that there is a large amount of neuron that are authorities and few nodes that acts as hubs. Regarding the other centrality measurements, it was found that there are few neurons with high values of them (Page Rank, Betweennes, Eigenvector). However, it was found that these nodes were widely spread over the anatomical graph.

## 4.2 Degree distribution



**Figure 7.** Plot of In and Out Degree distributions with curve fitting of power law to extract exponents,  $\gamma_{in}$  and  $\gamma_{out}$

Figure 7a and Figure 7c presents the frequency distribution plots of the In Degree and Out Degree distribution respectively. It can be observed, that while mean in-and-out degrees are the same, the standard deviations are slightly different. However, the number of neurons with in-degree in the range 0-30 are more similar to neurons with out degree in the same range. The Log-Log plot of in-and-out degree distributions gives us heavy-tailed distribution following a power law  $P_k = CK^{-\gamma}$  as shown in Figure 7b and Figure 7d. With this formula, the exponents  $\gamma_{in} = 2.081$  and  $\gamma_{out} = 2.177$  are extracted by curve fitting. For the curve fitting, all the degree values less than 80 are rejected since they are not significant while estimating the power law exponent  $\gamma$ . Following the Albert-Barabasi[3] definition "A scale-free network is a network whose degree distribution follows a power law.", the directed network of Zebrafish Connectome is a **scale free network**.

## 4.3 Small World property

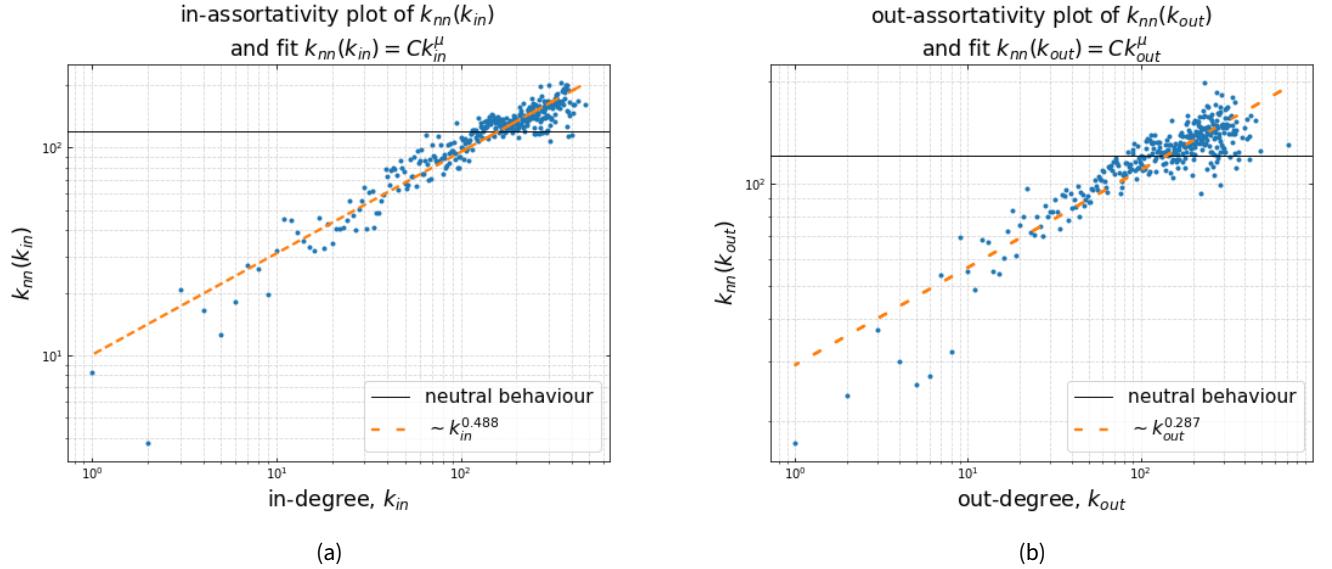
R. Cohen and Havlin[7][8] have showed analytically that "*Scale-free networks are ultra-small worlds*" where the diameter  $d$  is actually given by[8]

$$d \approx \ln(\ln N)/\ln(\gamma - 2) \quad (14)$$

instead of simply proportional to  $\ln N$ . From Eq. 14,  $\gamma$  is the degree exponent obtained from degree distribution  $p(k) \propto k^{-\gamma}$  which is probability of having  $k$  links outgoing from a site i.e.  $k = k_{out}$  and  $\gamma = \gamma_{out} = 2.177$  as seen from Figure 7d. So,

our network which is scale free with both the in-degree exponents and out-degree exponents in the range  $2 < \gamma < 3$  should actually follow the Eq.14. This is found to be relatively exact in our case since diameter  $d$  of our network(largest connected component to be specific) is computed to be 10 while Eq.14 with number of neurons,  $N = 3163$  and  $\gamma = 2.177$  gives  $d \approx 11.79$  (rounded to 2 decimals).

#### 4.4 Assortativity



**Figure 8.** Computing correlation exponents  $\mu_{in}$  and  $\mu_{out}$  from the fit which is a measure of In-Assortativity and Out-Assortativity

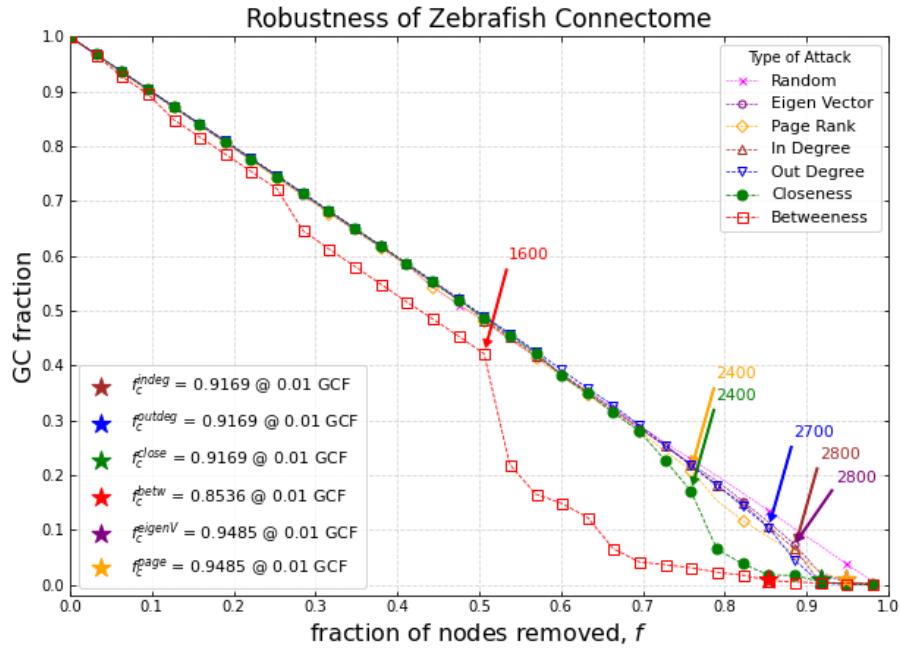
By fitting the data  $K_{nn}(K_{in})$  with  $K_{in}$  as seen from Figure 8a and  $K_{nn}(K_{out})$  with  $K_{out}$  as shown in Figure 8b in log-log scale is in accordance with Eq.10 and Eq.11 respectively,  $\mu_{in} = 0.488$   $\mu_{out} = 0.287$  are obtained. The correlation exponent  $\mu_{in} > 0$  shows that the neurons of the Zebrafish connectome tend to connect with other nodes which have similar in-degrees or in other words the nodes which have high in-degree values/ incoming connections tend to connect with other nodes which have high in-degree values while the nodes with low in-degree values tends to connect with other nodes which have similarly low in-degree values. The other correlation exponent  $\mu_{out} > 0$  provides the information that neurons of Zebrafish connectome tend to connect with other neurons which have similar out-degrees. That means, nodes which have high out-degree/hubs/outgoing connections prefer to connect with other nodes which have high out degrees/hubs/outgoing connections and low out-degree nodes prefer to connect with other nodes which have lower out going connections.

#### 4.5 Robustness:

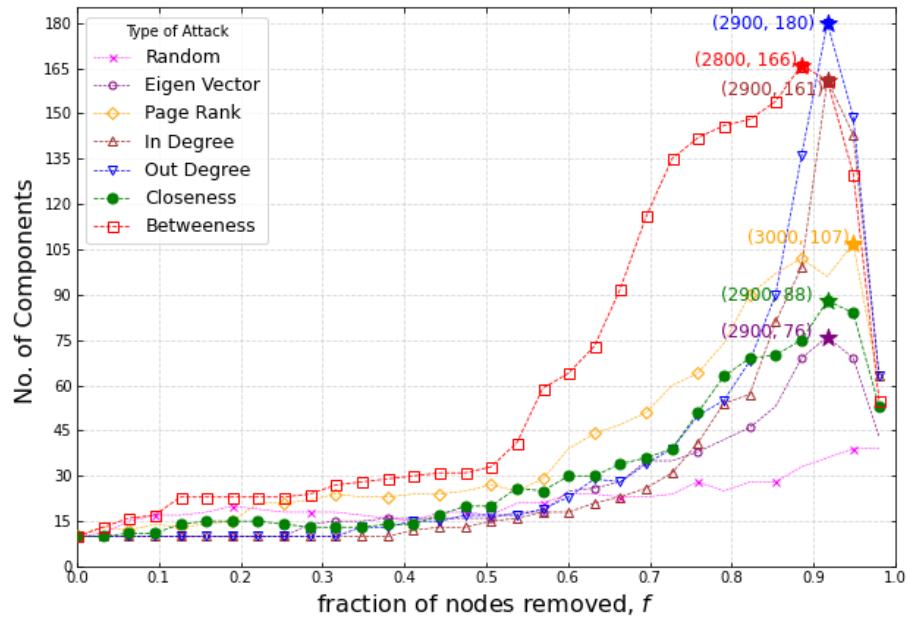
The robustness of the network is presented in Figure 9. It can be seen that the network refuses to break apart even under rather extensive node failure either for directed attacks or for even random failures. While the size of the largest component decreases linearly for random attacks, they vanish only in the vicinity of  $f = 1$  for directed attacks by in-degree, out-degree, closeness, page rank and eigen vector centralities. Except for attack based on betweenness centrality, where the largest component size decreases relatively faster in comparison with others. This means that the network behind the Zebrafish Connectome shows an unusual robustness to almost all the node failures: all of its nodes must be removed to destroy its giant component.

The point where the networks break the maximum or the size of giant component shrinks faster is highlighted by arrow marks in the Figure 9. From this, it can be infer that nodes with high betweenness centrality value or the nodes that acts as a bridge in the network. When nodes with high betweenness centrality are removed the network is less stable in comparison with respect to other centrality measures. Next in line is closeness centrality curve from the plot which are the top nodes from where the information could spread faster, is the next most important with respect to stability of network since the size of GC decreases faster than the other centralities. From the plot it can be seen that, removal of nodes by Eigen vector,

page rank, in degree, out degree centralities does not seem to destabilize the network.

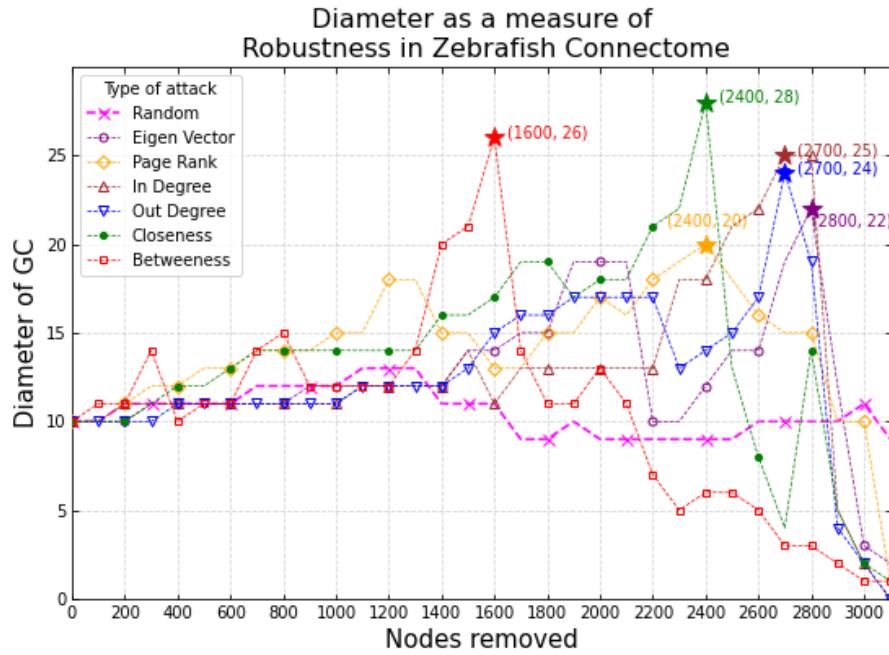


**Figure 9.** Robustness of the network measured in terms of number of nodes present in largest connected component after every stage of node removal based on centralities and random attack.

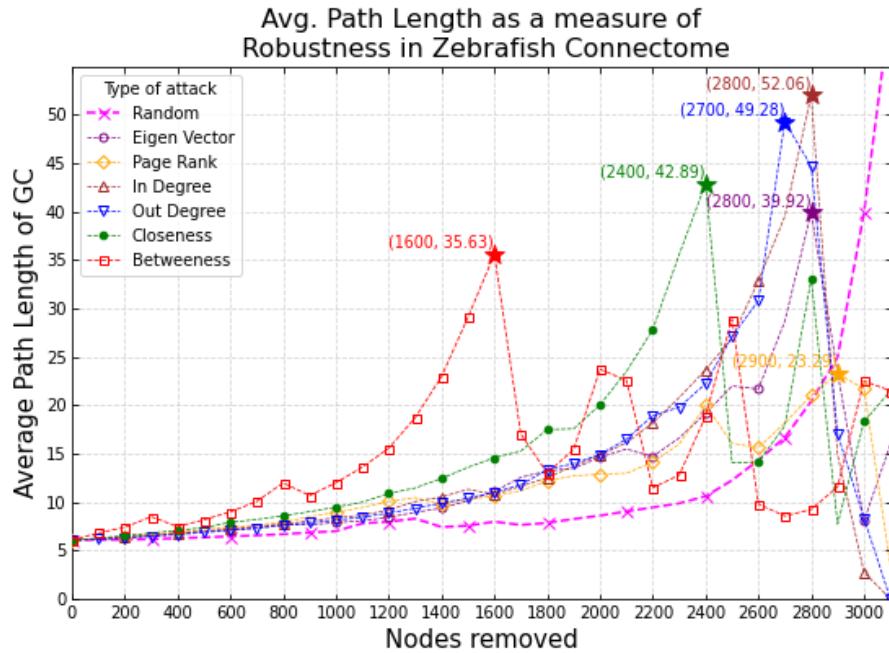


**Figure 10.** Number of components obtained at every 100 node removals on the basis of various centrality measures along with random attack.

The behavior observed above in the case of random node removal is not unique to only this network but all the scale free networks whose degree exponents are  $2 < \gamma < 3$ [3]. This means that to fragment a scale-free network by random node removal all of its nodes must be removed. In other words, the random removal of a finite fraction of its nodes does not break apart a large scale-free network[3].



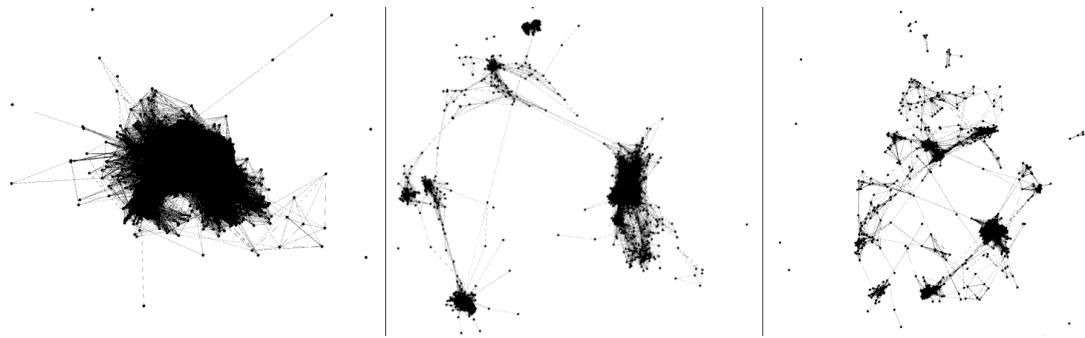
**Figure 11.** Plot of the variation of diameter of the giant component of the network as a function of removal of nodes based on various types of centralities along with random removal.



**Figure 12.** Plot of variation of Average Path length of the giant component of the network as an alternative to diameter.

As seen from Figure 10, with removal of nodes the number of connected components of network increases. However, if the nodes are removed based on betweenness centrality, the network tends to break faster as compared with respect to other centrality measures and so the number of components of the graph increases. It can be seen that all the curves peak at around 2800 - 3000 after which they drop down, meaning that in general when the nodes are getting removed the number of components increases until a certain point after which those components tend to have size of few nodes (on average 1 or 2) within them and in the next removal iteration they get completely removed which results in a sudden drop in the curve. It is however surprising to notice that number of components curve with respect to node removal by Page Rank attack is higher in comparison with respect to closeness curve because as seen in Figure 9, the GC fraction drops down quicker and goes lower in comparison with respect to Page Rank curve. The rate of increase in number of components at every random removal of nodes is seen to be lower than other curves as expected since random attack process removes the nodes in random fashion in contrast with directed attacks on hubs or nodes with higher centrality measures which tends to disintegrate the network quickly into a number of components.

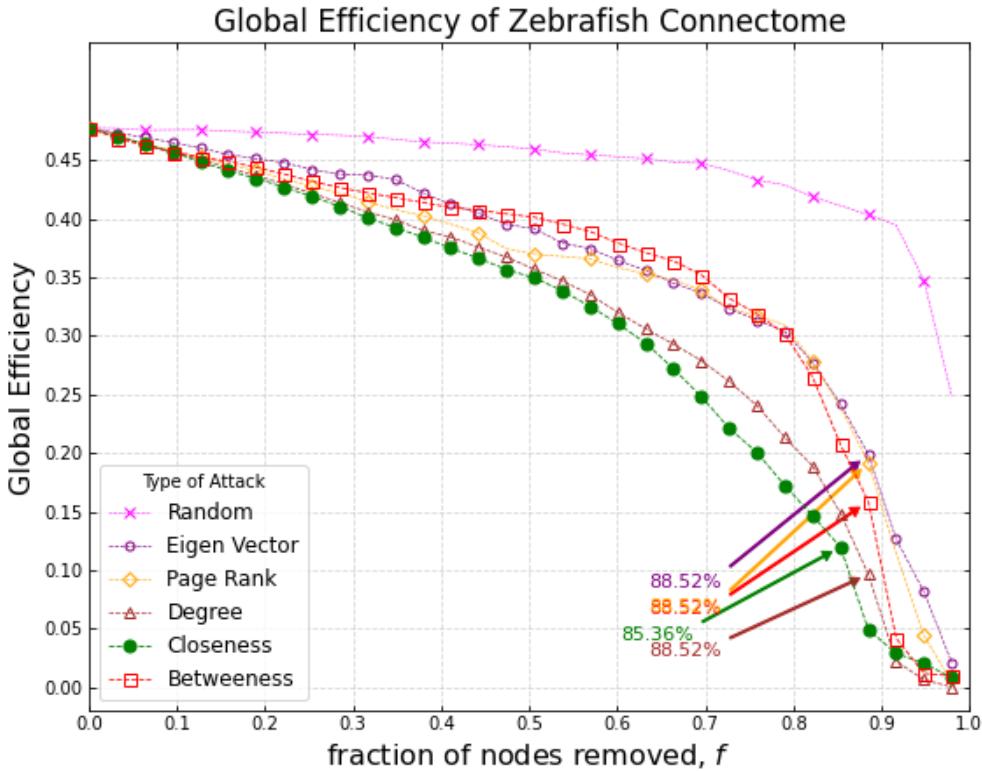
The diameter of the Giant Component which is the maximum of the shortest paths between any two nodes of the network and the average shortest path length which is average number of steps of all the shortest paths between any 2 given nodes, gives us more insight into the network robustness. Figure 11 shows us how the diameter of GC varies with the network dismantling. The same kind of behaviour is observed when considering average shortest path length as seen from Figure 12 i.e. from both figures the peaks at exact points when the number of nodes removed are observed. They are all similar except for the curve that defines the attack based on in-degree centrality. But the average path length Figure 12 is less noisy as compared to the plot of diameter Figure 11. The coinciding peaks of these two plots corresponds exactly to the point just before a huge drop in GC fraction as seen from Figure 9 which are marked with arrows. This gives an actual insight and understanding on how network dismantling is taking place. The arrow marked points in Figure 9 refer to the iterations in which giant component in the network has the maximum diameter and average path length. With the removal of nodes, the distance measures (diameter and path length) increase, until a breaking point (shown by arrows in Figure 9 and stars in figure 11, 12) in the network where with the further removal of nodes, the network breaks down into a number of components among which the largest of them is smaller in size in comparison with the previous giant component which was at the threshold. Another thing to observe is that, this break point/threshold is reached when the type of attack is based on betweenness centrality first and afterwards on closeness and Page Rank.



**Figure 13.** Visual progression of the robustness: original graph - betweenness checkpoint - closeness checkpoint

Figure 13 uses the Yifan Hu visualisation algorithm which is used for large and connected networks [27]. The progression of the split of the GC is depicted in three checkpoints. The first one is the shape of the original graph with 3163 nodes. The second is the graph shape after removing nodes with the betweenness attack at the point of the largest drop in the GC fraction i.e. after removal of 1700 neurons. The same is applied in the third graph, for the attack on closeness centrality, after removal of 2500 neurons. It can be seen that the visualisation algorithm depicts the separation of the GC, even though some components are weakly connected.

## 4.6 Efficiency:



**Figure 14.** Global efficiency of the network measured as a function of removal of nodes based on various centralities.

As seen from Figure 14, when the type of attack is based on closeness centrality, the efficiency curve quickly dips down as compared to other centrality measures. This is reasonable since closeness property refers to measure of center of gravity i.e. top nodes from which to spread information. However, removal of nodes based on betweenness centrality has much lesser effect on efficiency of the network and random removal provides the smallest effect.

## 4.7 Community Detection

Several clustering algorithms were implemented using available communities detection in IGRAPH. Table 2 shows the number of communities and the modularity found for each of the algorithms computed for the whole graph and for the Giant component (GC). In general, the algorithms which are high in modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. As anticipated, since Louvain Modularity optimizes the modularity metric, it produces the highest value for the whole graph by dividing the nodes into 12 communities. Notably, the Spin Glass model is able to capture the highest modularity for the case of the GC when the final temperature is 0. The number of communities decreases for the GC whereas the modularity increases, meaning that communities structures becomes better, which is to be expected. Moreover, the number of communities for the cases of all algorithms but the Louvain Significance and Louvain Surprise are similar to the number of communities found in the normal protein network [23].

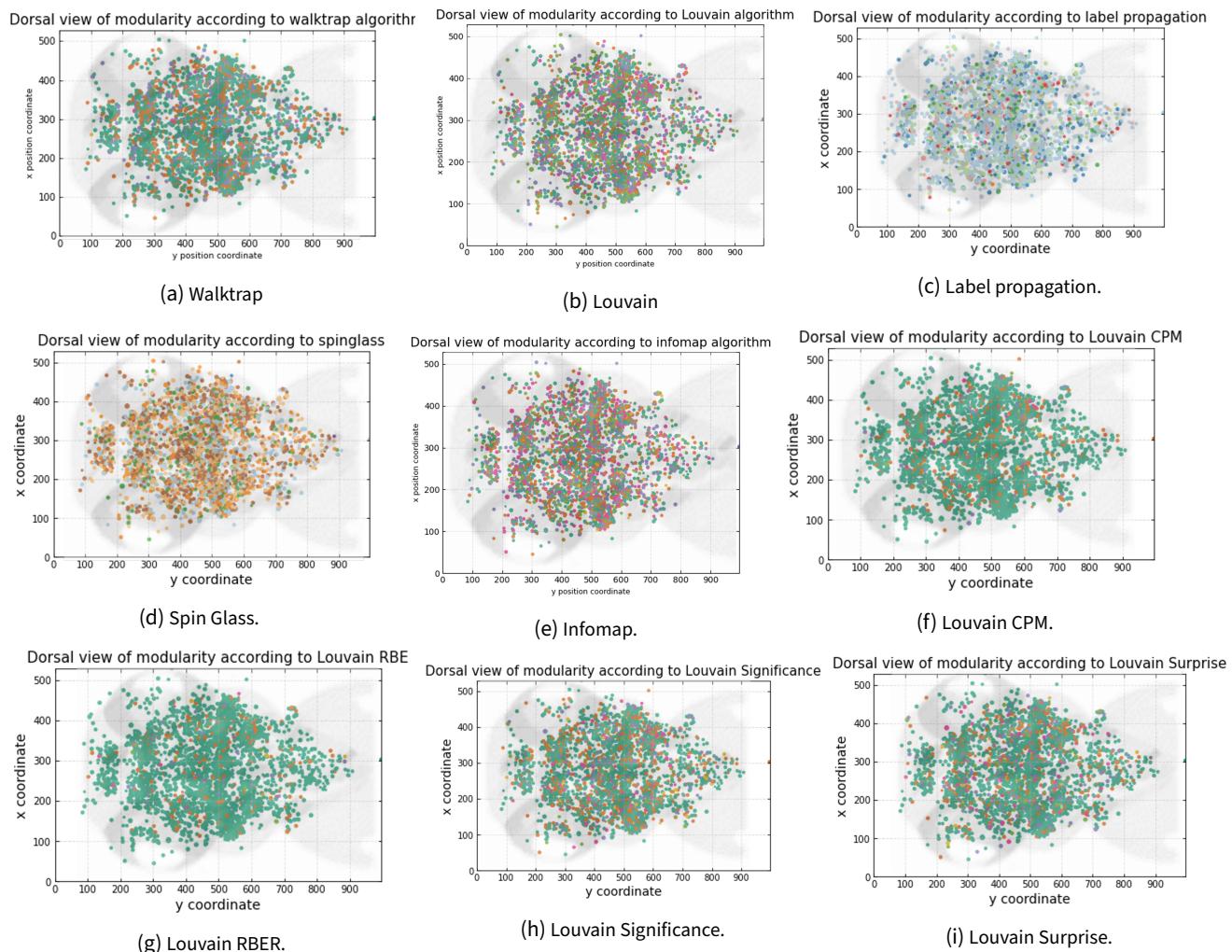
Figure 15 shows the visualization of the communities in the network by its anatomical position. Unfortunately, no underlying clustering structure is visualized. Thereafter, no anatomical mapping with communities was found. A possible cause for this random effect is the fact that the selected and traced neurons in the dataset are not sampled uniformly from the whole brain. There are a lot of neurons sampled from brain regions that are believed to be most connected and only few neurons from other regions.

**Table 2.** Communities in Zebrafish connectome

Community algorithm	Number of Clusters	Modularity	Number of Cluster (GC)	Modularity (GC)
Walktrap	41	0.380	33	0.395
Infomap	47	0.351	34	0.399
Label Propagation	44	0.312	8	0.184
Louvain Modularity	12	0.460	9	0.463
Louvain RBER	54	0.454	48	0.452
Louvain CPM	54	0.444	52	0.441
Louvain Significance	117	0.323	108	0.333
Louvain Surprise	147	0.305	135	0.324
Spinglass	*	*	17	0.468 ( $T_f = 0$ )

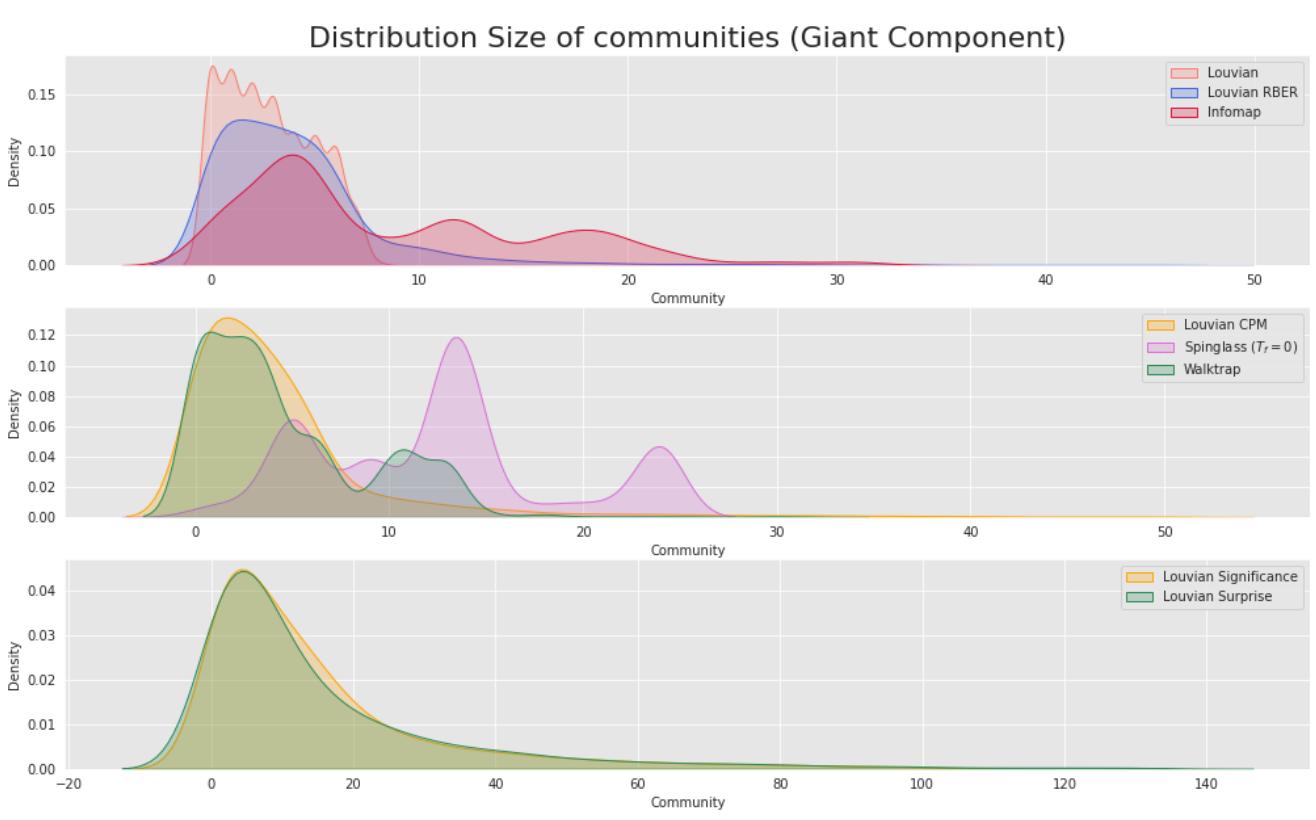
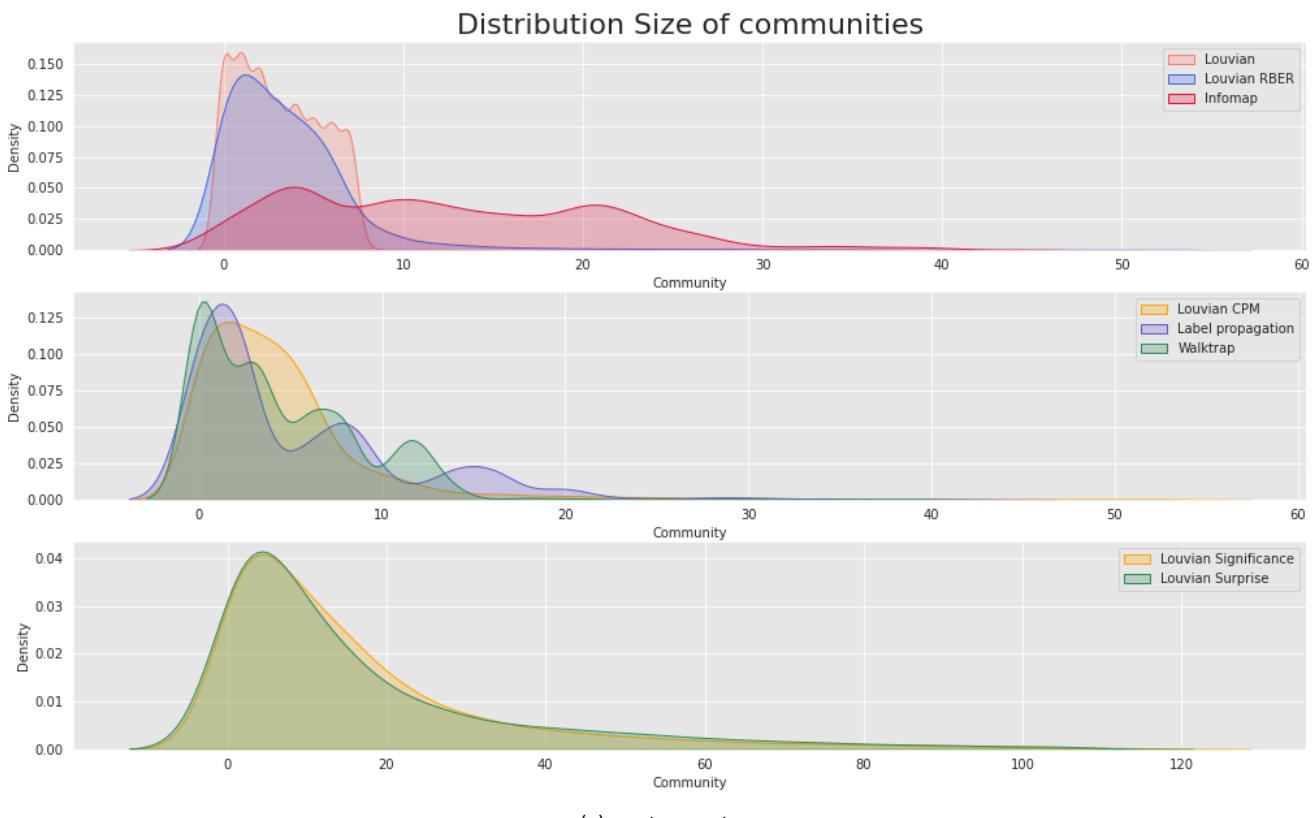
<sup>1</sup> Spinglass can only be performed in connected networks.

<sup>2</sup> Label propagation did not allow the use of weight when using GC leading to erroneous results.



**Figure 15.** Network visualization of the communities.

The numbers of the communities found highly depends on the community algorithm leading into different values as shown in Table 2. Figure 16 presents the distribution size of each of the communities found for the whole graph and the GC.



**Figure 16.** Distribution of size the communities found.

First of all, when observing the communities found for the whole graph, it should be noticed that despite having a large number of communities, there are only few communities that can be considered important. For example, taking out the cases of `infomap`, `Louvain Significance` and `Louvain Surprise`; the algorithm tends to cluster most of the nodes of the networks in the first  $\sim 10$  communities leading into larger cluster sizes. Thus, these algorithms are able to capture larger regions of the brain whereas, the other algorithms create more specific clusters with less nodes inside of them. In particular, this is notorious for the case of `infomap`, which is able to find clusters with an uniform number of nodes in each of them.

Thereafter, it can be seen that the number of meaningful communities of the networks are no more than  $\sim 10 - 20$ . Recall that it was found that this graph has positive assortativity, therefore neurons tends to group with neuron which have similar in-degree and out-degree. In particular, it can be seen that the neurons tends to group in only a few large sized communities and this resembles to the assortativity plots obtained in Figure 8. Furthermore, the fact that communities are low are also correlated to the fact that when attacking the network, no "abrupt" change can be found (Figure 9). This may be to the fact that neurons are well organized in large communities among the networks, as it was already shown, hence, attacking to a single neuron does not affect to the whole brain.

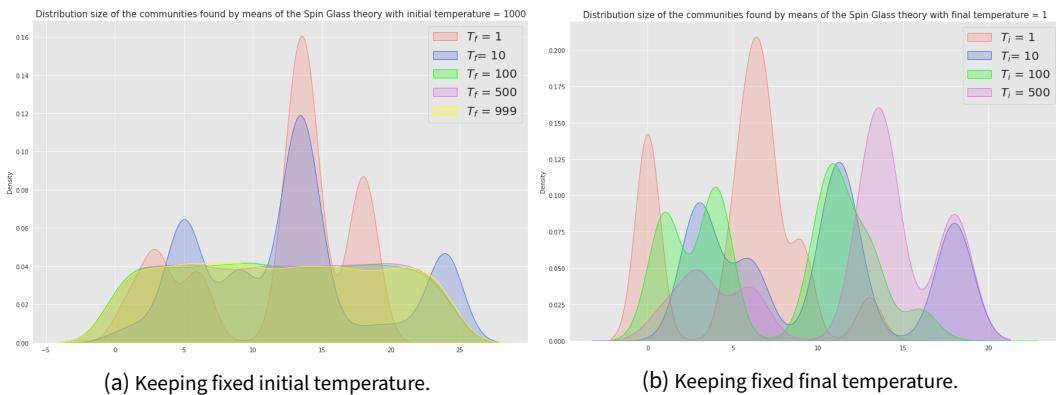
The `Louvain` algorithm tends to group edges that are in multiple shortest paths. It was already found that every neuron of the graph has a really high degree of closeness. Thereafter, it should be no surprise that `Louvain` is able to cluster the neurons in only 10 communities with a uniform size. The `Label propagation` algorithm and the `Walktrap` algorithm are able to capture a bit more than 10 communities, with sizes decreasing exponentially with oscillations. Finally, the `Louvain Significance` and the `Louvain Surprise` are able to capture much more communities but consisting of only few nodes.

On the other hand, when comparing with the GC (Figure 16b) it can be observed that for the case of the `Louvain RBER`, the size of the communities are more uniform in the first ten communities. The `Louvain Surprise`, `Louvain CPM` and `Louvain Significance` algorithm has a similar trend as for the whole case.

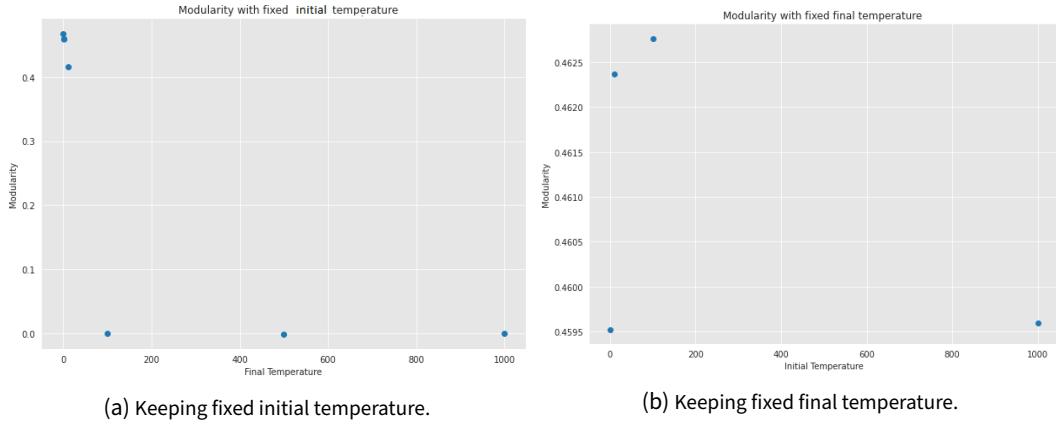
The `infomap` algorithm looses the uniformity exhibited in the whole graph, and makes a distinction in size between the first 10 communities and the 10 following up. For the `Walktrap` algorithm, something similar happens, in the region of 0-10 communities the density becomes larger, and in the next communities the density abruptly decreases.

Finally, the `Spin glass` model is the most interesting one. As already mentioned, it groups the neurons according to their spin states, and the energy of the spin system is the quality function of the communities. The aim of this algorithm is to find the ground state of a spin glass model with a Potts Hamiltonian. In this scenario, it can be noticed that the number of communities founded corresponds to 20 with 4 notable sizes. This behaviour can be observed when changing the initial and final temperature of the Spin Glass model as shown in Figure 17. It can be seen that changing the final temperature into higher values, when keeping fixed the initial temperature ( $T_i = 1000$ ) leads to bad results. Indeed this can be seen in Figure 18b where the modularity decreases when increasing the final temperature. This somehow is related to the magnetization vs temperature found in the Potts model where there is an abrupt change from lower values of the magnetization [25].

The distributions between fixed final temperate ( $T_f = 1$ ) and initial temperature are similar in comparison. In figure 18a it can be observed that the modularity is always high, being the one with initial temperature ( $T_i = 100$ ) the best one.



**Figure 17.** Distribution of size of the Spin Glass communities when varying the temperature.

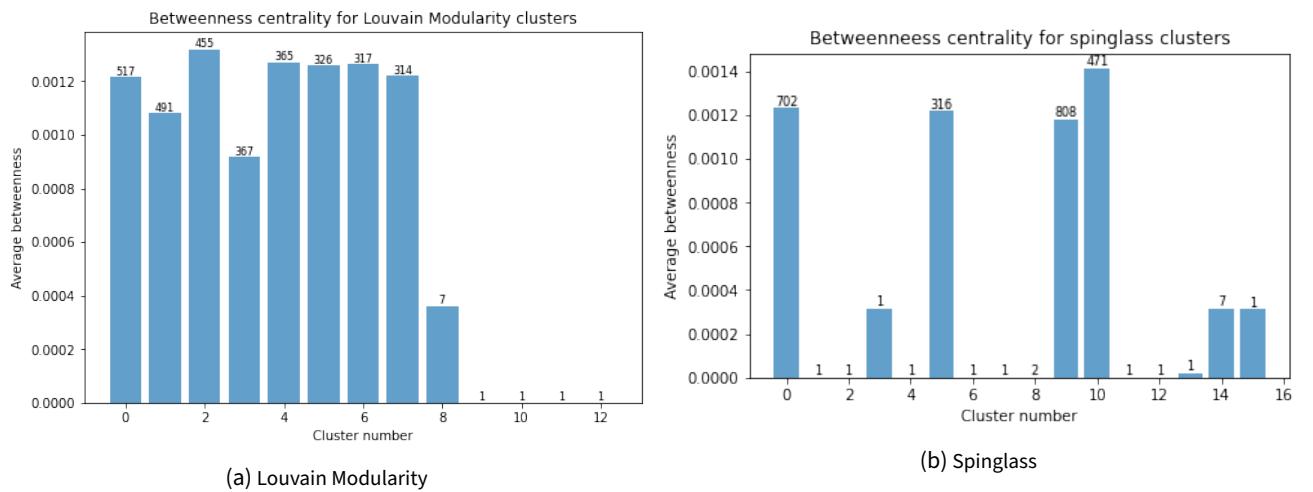


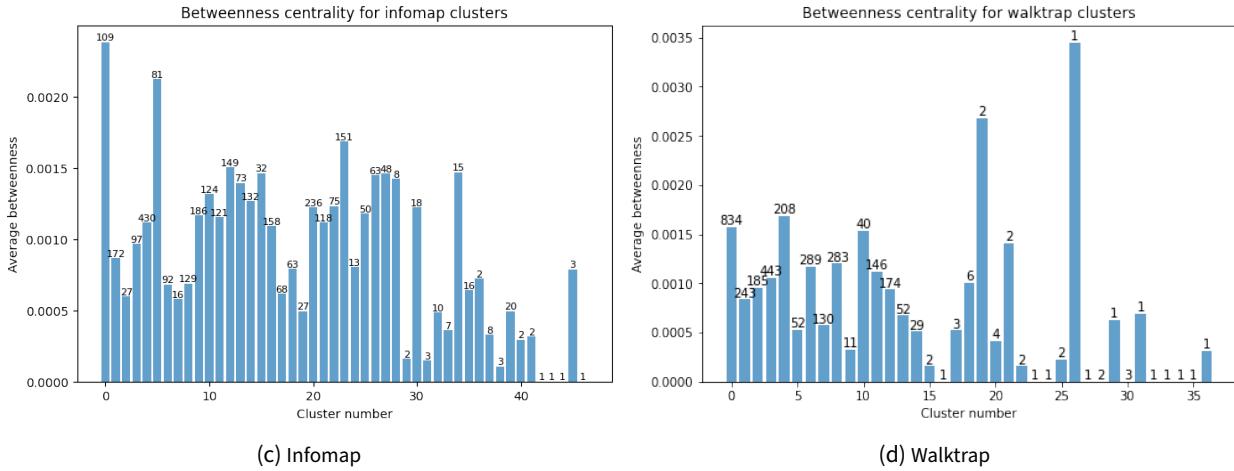
**Figure 18.** Modularity of the Spin Glass communities when varying the temperature.

Figure 19 is motivated by the fact that betweenness centrality plays an important role in the attacks that tackle the network robustness. For each algorithm, the average betweenness by cluster is calculated in order to point out the clusters with the highest values. In addition, the cluster size must be taken into account since some clusters that have very large values consist of only few nodes and are considered outliers. This is the case with some clusters from the *Walktrap*, *Louvain CPM* and *Louvain RBER* algorithms. On the other hand *Louvain Modularity* produces almost uniform results in the 7 clusters which contain a significant number of nodes. *Infomap* produces the most uniform betweenness among the algorithms, with two clusters significantly higher in betweenness centrality.

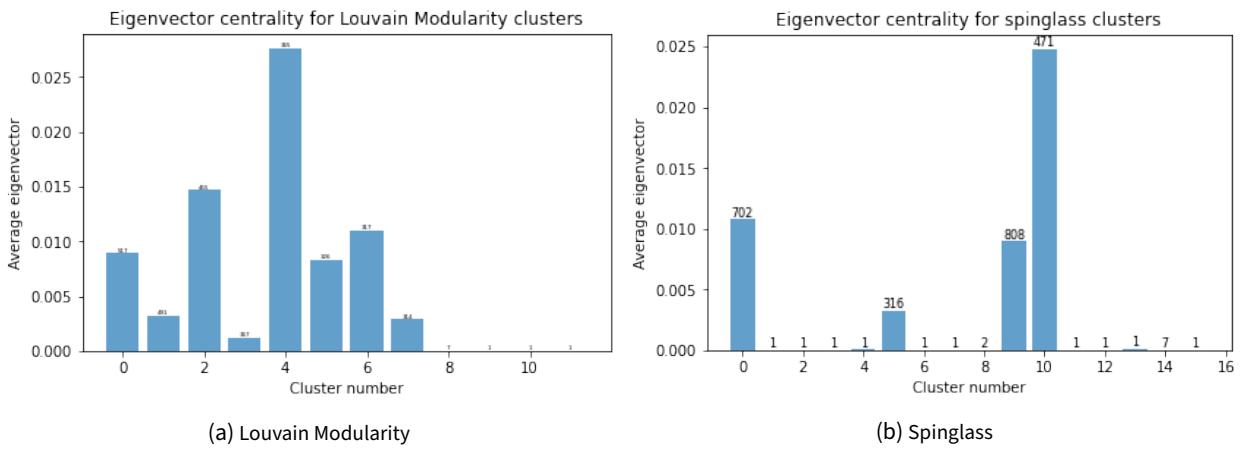
Moreover, the robustness was study for the case of communities ordered by betweenness centrality. Nothing notably was found, in fact the robustness does not decrease the GC fraction any faster than the random attack. This may be attributed to the fact that a lot of nodes (whole communities) are removed in each iteration.

Figure 20 shows that for some algorithms are able to differentiate between communities high in a particular centrality. In this case, the algorithms who detect smaller number of clusters such as *Louvain Modularity* and *spinglass* are able to distinguish nodes whose influence on the network is high and cluster them together. However, these nodes do not make up a particular region of the brain anatomically.





**Figure 19.** The barplots represent the average betweenness centrality by cluster for different algorithms. The size of each cluster is denoted with a label on top of the bar.

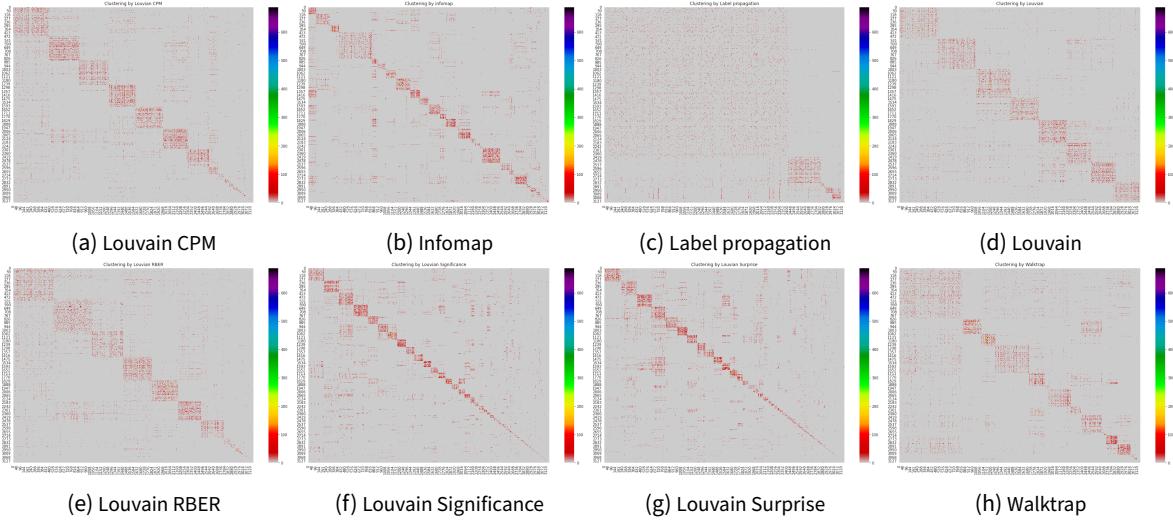


**Figure 20.** Average eigenvector centrality by cluster for Louvain Modularity and spinglass algorithms.

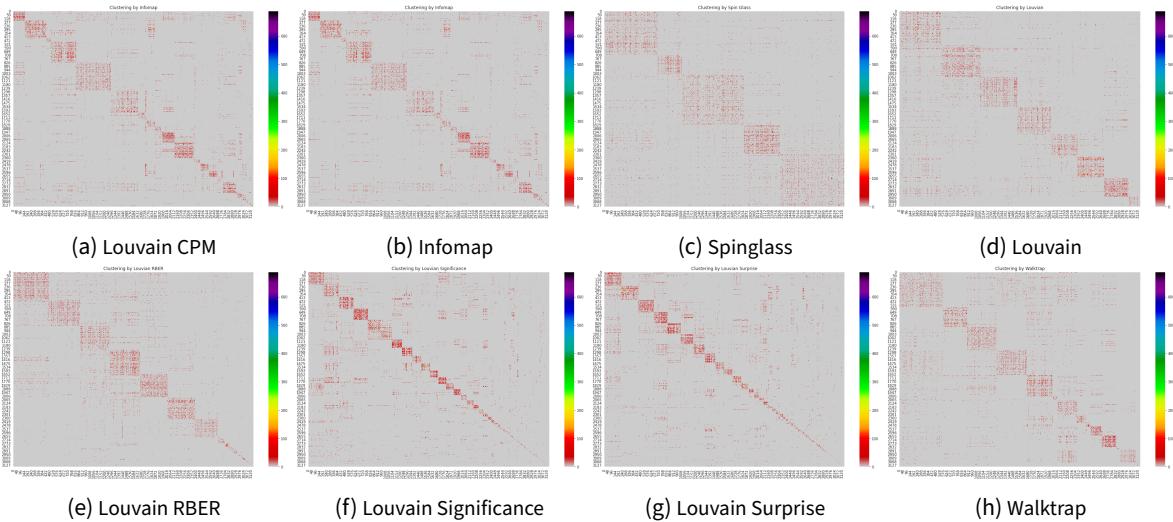
Figure 21 shows the adjacency matrix of the whole network organized by the different communities. It can be seen that the infomap, Louvain Significance and the Louvain Surprise communities are able to capture much smaller regions than the other ones as previously discussed. As already seen in Figure 16a, the Louvain, Louvain CPM and the Louvain RBER algorithms are able to organize the brain in big structures with similar sizes. Instead the Walltrap and the Label propagation are clustering the neurons with different size clusters.

In contrast, for the case of the GC (Figure 22), it can be observed that the communities are similar to the whole graph. However, it should be pointed out that now the "background" becomes neater in the sense that the clustering becomes better. On the other hand, the Label Propagation was performed without weights leading to unwanted results. The Walktrap algorithm achieves more uniform cluster sizes for the GC than for the whole graph, exhibiting large size communities and small communities, as for the case of infomap.

In order to have further comparison with the communities, the split-join distance among communities are computed. The split-join distance between partitions A and B is the sum of the projection distance of A from B and the projection distance of B from A. Figure 23a shows the comparison values of the split distance of the whole graph. As expected, the split distance among Louvain, Louvain RBER and the Louvain CPM are in between  $\sim [300, 1000]$ , which is a small value, with the two last ones being more similar, respectively. A similar trend can also be found with the Louvain Significance and Louvain Surprise with a value of  $\sim 1400$ . Interestingly, the difference between infomap and Louvain Significance and Louvain Surprise is only  $\sim 1700$  neurons; with Louvain CPM and Louvain  $\sim 1700$  neurons and finally  $\sim 2000$  with the Walktrap algorithm.



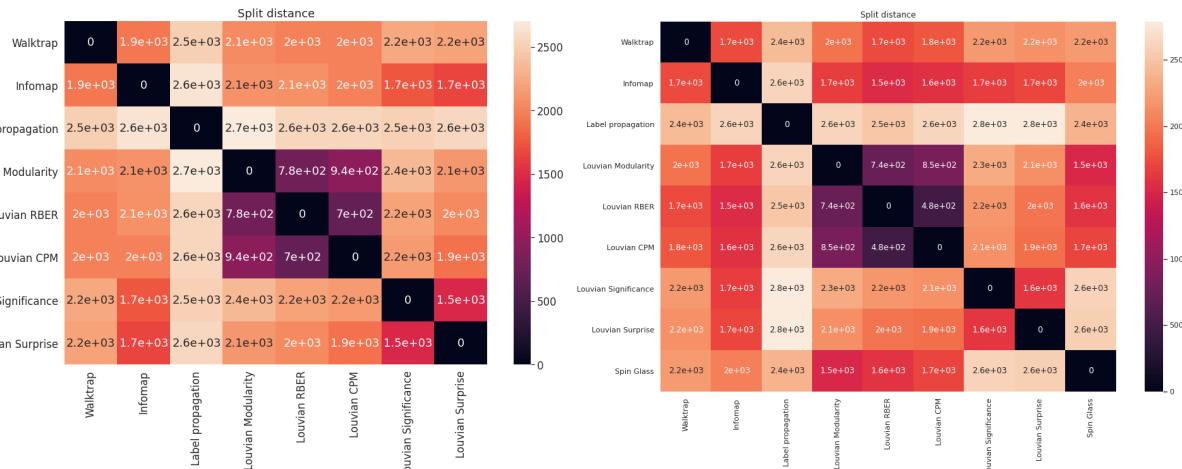
**Figure 21.** Adjacency matrix organized by the different communities.



**Figure 22.** Adjacency matrix organized by the different communities (Giant Component).

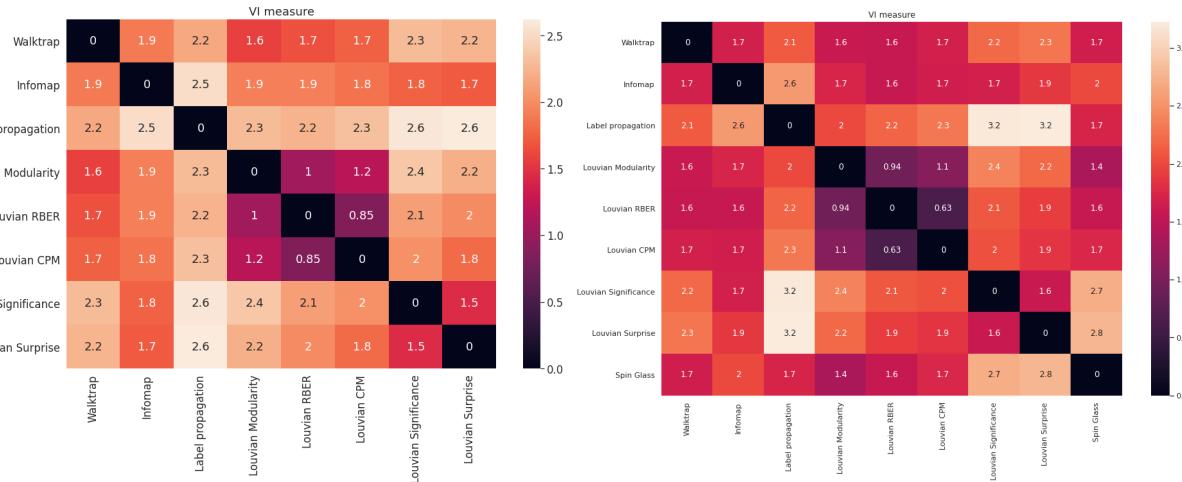
The Walktrap algorithm instead is more related with the Louvain RBER, Louvain CPM and Louvain than the Louvain Significance and Louvain Surprise as it happens for the case of Infomap. Finally, it should be noticed that the Label propagation algorithm is the farthest distanced one with respect to the other algorithms. The same information can be obtained by calculating the Variation of information and the Normalized Mutual Information as it can be seen from Figure 23c and Figure 23e.

Comparing with the Giant component (Fig 23b, 23d and 23f), one can observe again the two regions corresponding of the Louvain's as before. Nevertheless, it should be highlighted that in this scenario the Walktrap is more similar to the infomap. For the case of infomap it can be seen that now is more related to the Louvain Modularity, Louvain RBER and Louvain CPM rather than the Louvain Significance and Louvain Surprise. Lastly, for the case of spin glass it can be seen that it is similar to the Louvain Modularity, Louvain RBER and Louvain CPM following with the Infomap.



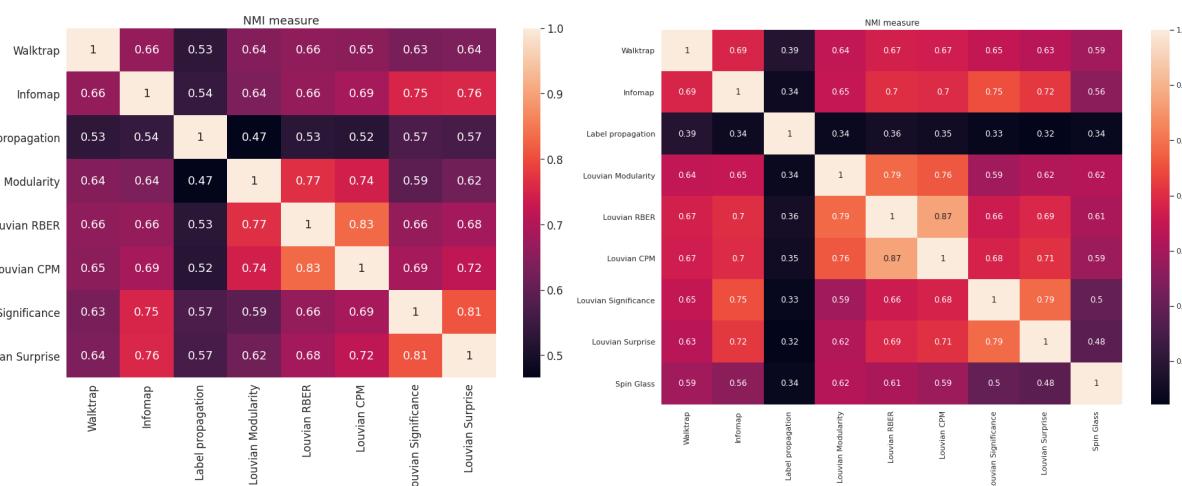
(a) Split distance (Total network).

(b) Split Distance (Giant component).



(c) Variation of Information (Total network).

(d) Variation of Information (Giant component).



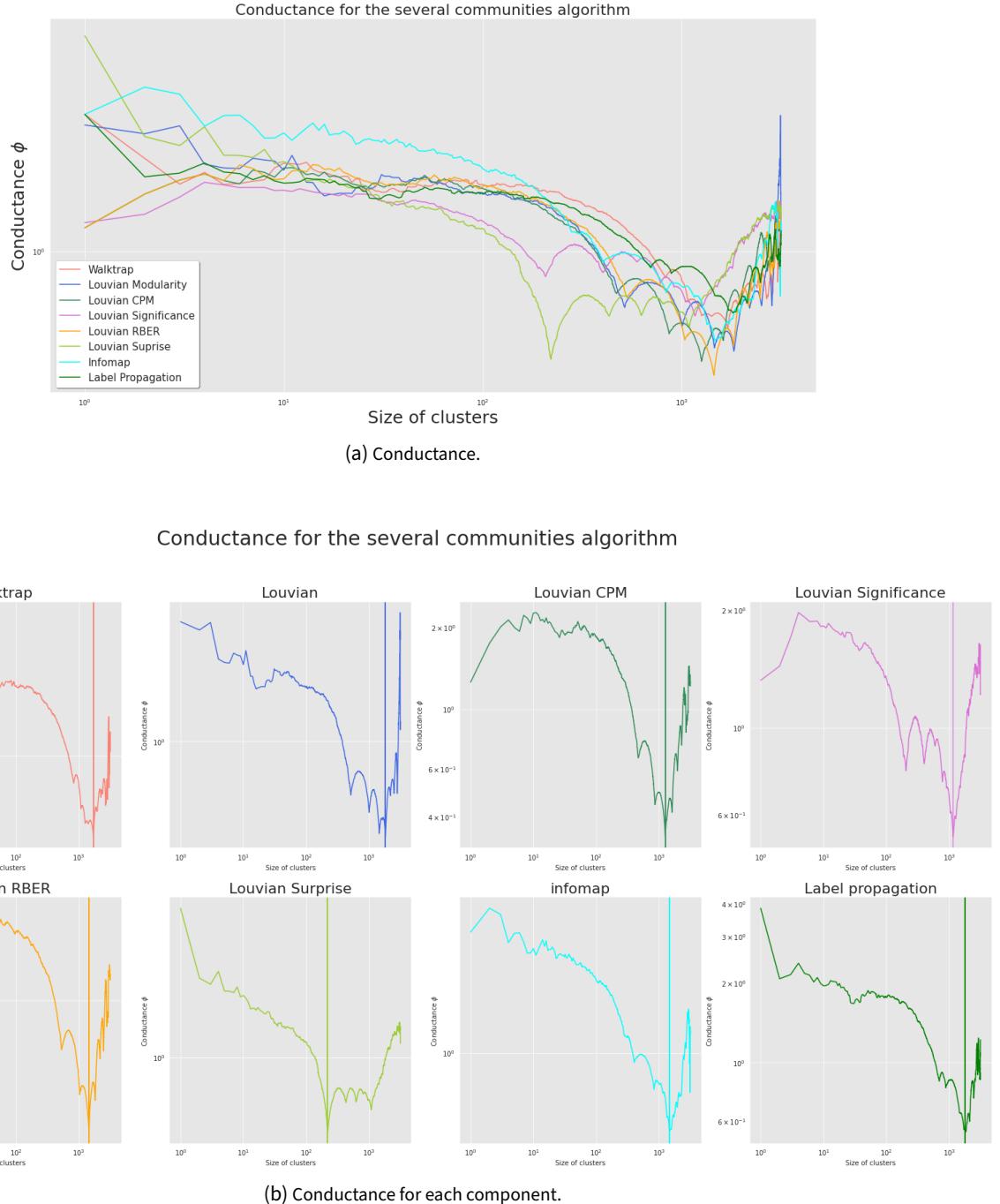
(e) Normalized Mutual Information (Total network).

(f) Normalized Mutual Information (Giant component).

**Figure 23.** Comparison of the communities by means of several metrics.

The conductance is the probability that a random edge leaves the set, hence the smaller, the better. Figure 24 shows

the Network Community profile (NCP) for the whole network. The first thing to notice is that the NCP tends to have the universal "V" shape when the size of the community increases. Furthermore, the conductance using the several methods are quite similar. It should be noticed that the minimum of this graph is in very large communities size(it will be discussed further later). This means that this structure can be treated as an "onion-like nested core-periphery" structure, where in this case the network consists of a large core with a large number of small very well-connected communities barely connected to the core. Nevertheless, it should be remarked that for the case of small-worlds this behaviour is really common. The NCP is mostly important for larger networks.



**Figure 24.** Network community profiles for the different algorithm for the whole graph

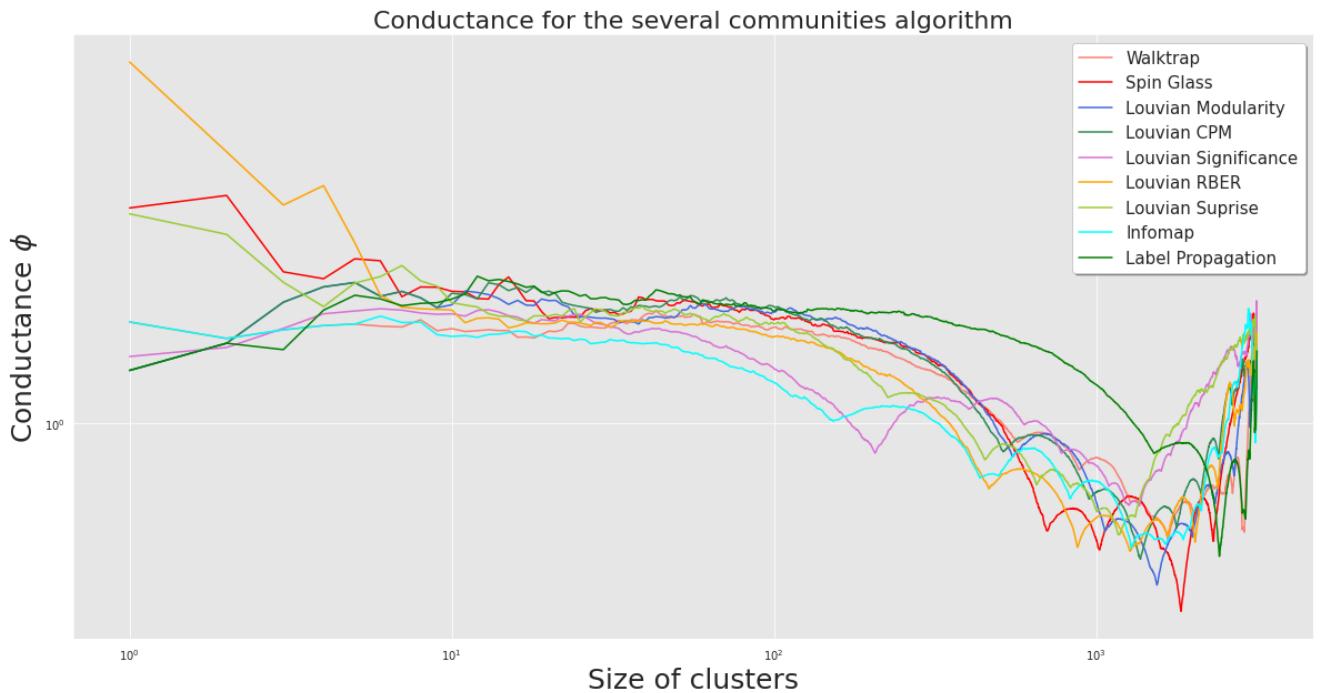
It can also be seen that the optimal size of the communities such that the value of the conductance is minimum can be founded in a similar value (taking out the case of the Louvain Surprise). Table 15 shows the minimum values achieved

by the several algorithms. It can be seen that the best conductance value corresponds to the Louvain RBER algorithm with a community size of 1449.

**Table 3.** Conductance

Community algorithm	Optimal size	Min conductance	Optimal size (GC)	Min conductance (GC)
Walktrap	1703	0.467	1668	0.511
Infomap	1461	0.412	1284	0.487
Label Propagation	1801	0.545	2411	0.464
Louvain Modularity	1828	0.374	1543	0.394
Louvain RBER	1449	0.295	1272	0.478
Louvain CPM	1257	0.338	1371	0.457
Louvain Significance	1163	0.528	1360	0.621
Louvain Surprise	219	0.345	1179	0.526
Spin Glass	-	-	1832	0.338

A similar behaviour also occurs for the case of the GC. Figure 25 shows the network community profiles of the several communities implemented in the GC. The V shape is also present as before, nevertheless it should be pointed out that by observing Table 3 one can observe that the minimum conductance found is increased for every algorithm. The community optimal size remains more or less the same but for the case of the Louvain Surprise in which it is highly increased. In this case the minimum conductance corresponds to the case of the Spin Glass.



**Figure 25.** Level of conductance related to the size of clusters

It was found that the optimal size clustering can be indeed really large ( $O(1000)$ ) for the case of this network. As already mentioned the conductance is related to cut of the edges. It was found that that each neuron of this graph posses high degree of closeness and the average path length among them is really low. Furthermore, the flow of information in the whole brain was well spread among the regions. Thereafter, having a huge size of optimal community it is not strange for this scenario. Since all plots are similar, it could be thereby conclude that, according to the conductance value, all methods are similar and present an akin behaviour.

In conclusion, for the **whole graph** the following similarities can be identified among the communities:

- Large brain region:** consisting on the algorithms Louvain Modularity, Louvain RBER and Louvain CPM. These are the communities which lead to higher modularity in all cases. The size distribution for each of them were highly compressed into just [0, 10] communities. This is evident in the form of the adjacency matrices (Fig 21)
- Specific brain region:** consisting on the algorithms Louvain Significance, Louvain Surprise, Infomap. The similitude with *Louvain Significance* and *Louvain Surprise* are extremely high. Once again the modularity for the communities are similar as shown in Table 2. The similitude is highly evidence of the form of the adjacency matrices show in Figure 21.
- Mixture of Large and small:** Walktrap, Label Propagation. The modularity values obtained by the two algorithms were similar as were the number of clusters (Table 2). In principle they should be considered as two different types of clustering since the metrics found in Figure 23 provide evidence that the two algorithms does not share many elements in common.

In respect to the **Giant component**:

- Large brain regions:** it can be also categorized as follows:
  - Uniform cluster size: consisting on Louvain Modularity, Louvain RBER and Louvain CPM. The modularity values of these models were similar (and high!) although the number of communities vary among them.
  - Somewhere in between the two of them: Walktrap. The modularity increases with respect to the whole graph, the distribution size also seems to be more uniform than before (Fig 22). In this scenario, it is more similar to the Infomap algorithm.
  - Irregular cluster size: consisting on Infomap and Spin Glass. The Infomap algorithm is able to considerably increase its modularity in regard to the the GC compared to the whole graph (Table 2)). Remarkably, the Spin glass algorithm is the one with highest modularity. With some intuition, one may think that brain regions should not be proportional, then this situation may be the one leading to more "real" results than the other ones.
- Specific brain regions:** Louvain Significance and Louvain Surprise. Same argument as before.
- Label Propagation is discarded for our own sake.

## 5 Conclusion

- The  $\gamma$  exponents from the power law extracted from in degree and out degree distributions are within range  $2 < \gamma < 3$  and thus the structural network of Zebrafish Connectome is "Scale free" whose degree distribution is heavy tailed.
- Since our network is both Scale free and follows exactly the Eq.14, the Structural Network of Zebrafish Connectome is an Ultra Small World. In fact, the values of density and clustering coefficient resembles to Small world networks.
- The neurons of Zebrafish connectome are In-Assortative and Out-Assortative meaning that they prefer to form connections with other neurons of similar in-degree and out-degree.
- The network shows unusual robustness to almost all the node failures based on various centralities with critical threshold within the vicinity of 1. Even though network does not break down completely to node removal by betweenness centrality, it shows some response unlike other types of attack(see Figure 9).
- The network is highly robust to random attacks whose critical threshold  $f_c \approx 1$  and confirms the theoretical result that for scale free networks with  $\gamma < 3 \implies f_c \rightarrow 1$ [3].
- The network is more susceptible to the attack based on betweenness centrality i.e. removal of nodes that acts as bridges and then comes closeness and page rank centrality in that order.
- The efficiency of the graph is notable, meaning that the flow of information among the graph is high, this is evidence by the fact that the average path length was short leading into a high degree of closeness centrality. However, the efficiency of the network is more susceptible to network dismantling based on closeness centrality measure and less susceptible to attack based on betweenness centrality measure.
- Many nodes act as authorities and few nodes act as hubs.
- The communities found were classified according to the sizes of the region, founding at every time around 10 – 20 number of meaningful communities. Furthermore, the network communities profiles of each method behaves similarly, all leading into the well known "V-shape".
- The Louvain modularity method was the one with biggest modularity value for the case of the whole graph whereas the highest value correspond to the Spin Glass method when reaching final temperatures near to 0 for the case of the GC. In

fact, it has been deduced that these type of algorithms leads into regions with irregular cluster size instead of uniformly, which is something that it is expected when dealing with the "real" brain region.

- Most of the clusters have almost equal value for average betweenness. It can be inferred that the nodes high in betweenness centrality are distributed among different clusters and all of the clusters with significant number of nodes tend to communicate to each other in a connected manner.
- Robustness attack by communities ordered by betweenness centrality does not decrease the GC fraction any faster than the random attack. This may be attributed to the fact that a lot of nodes (whole communities) are removed in each iteration.

## References

- [1] Kunst, Michael et al. "A Cellular-Resolution Atlas of the Larval Zebrafish Brain." *Neuron* vol. 103,1 (2019): 21-38.e5. doi:10.1016/j.neuron.2019.04.034
- [2] Standardised SWC format for Neuro-morphological dataset  
<http://www.neuronland.org/NLMorphologyConverter/MorphologyFormats/SWC/Spec.html>
- [3] Network Science by Albert-Laszlo Barabasi  
<http://barabasi.com/networksciencebook>
- [4] Rubinov M, Sporns O. Complex network measures of brain connectivity: uses and interpretations. *NeuroImage* 52: 1059-1069
- [5] Latora, Vito, and Massimo Marchiori. "Efficient behavior of small-world networks." *Physical Review Letters* 87.19 (2001): 198701. <https://doi.org/10.1103/PhysRevLett.87.198701>
- [6] Mahendra Piraveenan, Mikhail Prokopenko, Albert Zomaya, "ASSORTATIVE MIXING IN DIRECTED NETWORKS TECHNICAL REPORT 637", University of Sydney, March 2009 [https://www.researchgate.net/profile/Albert\\_Zomaya/publication/237329695\\_ASSORTATIVE\\_MIXING\\_IN\\_DIRECTED\\_NETWORKS\\_TECHNICAL\\_REPORT\\_637/links/53f879c90cf2c9c3309dfb53/ASSORTATIVE-MIXING-IN-DIRECTED-NETWORKS-TECHNICAL-REPORT-637.pdf](https://www.researchgate.net/profile/Albert_Zomaya/publication/237329695_ASSORTATIVE_MIXING_IN_DIRECTED_NETWORKS_TECHNICAL_REPORT_637/links/53f879c90cf2c9c3309dfb53/ASSORTATIVE-MIXING-IN-DIRECTED-NETWORKS-TECHNICAL-REPORT-637.pdf)
- [7] Cohen R, Havlin S, ben-Avraham D (2002). "Structural properties of scale free networks". *Handbook of Graphs and Networks*. Wiley-VCH, 2002 (Chap. 4).
- [8] Cohen R, Havlin S (February 2003). "Scale-free networks are ultrasmall". *Physical Review Letters*. 90 (5): 058701. arXiv:cond-mat/0205476. Bibcode:2003PhRvL..90e8701C. doi:10.1103/PhysRevLett.90.058701. PMID 12633404. S2CID 10508339.
- [9] Newman M. E., *Phys Rev Lett*, 89(2002) 208701.
- [10] Martin Rosvall and Carl T. Bergstrom (January 2008). "Maps of random walks on complex networks reveal community structure". <https://doi.org/10.1073/pnas.0706851105>
- [11] Pascal Pons and Matthieu Latapy (December 2005). "Computing communities in large networks using random walks". arXiv:physics/0512106
- [12] Raghavan, U.N. and Albert, R. and Kumara, S (2007). "Near linear time algorithm to detect community structures in large-scale networks". *Phys Rev E* 76:036106
- [13] Reichardt J and Bornholdt S (2006). "tatistical mechanics of community detection". *Phys Rev E* 74:016110 (2006).<http://arxiv.org/abs/cond-mat/0603718>.
- [14] Hoffman M, Steinley D, Gates KM, Prinstein MJ, Brusco MJ.(2018). "Detecting Clusters/Communities in Social Networks". *Multivariate Behav Res*. 2018;53(1):57-73. doi:10.1080/00273171.2017.1391682
- [15] Louvain algorithm description  
<https://neo4j.com/docs/graph-algorithms/current/algorithms/louvain/>
- [16] Louvain python package  
<https://pypi.org/project/louvain/0.5.3/>
- [17] MEJ Newman and M Girvan (2004) "Finding and evaluating community structure in networks." *Phys Rev E* 69 026113.
- [18] Meila M: Comparing clusterings by the variation of information. In: Scholkopf B, Warmuth MK (eds). *Learning Theory and Kernel Machines: 16th Annual Conference on Computational Learning Theory and 7th Kernel Workship, COLT/Ker-*nel 2003, Washington, DC, USA. *Lecture Notes in Computer Science*, vol. 2777, Springer, 2003. ISBN: 978-3-540-40720-1.

- [19] Danon L, Diaz-Guilera A, Duch J, Arenas A: Comparing community structure identification. *J Stat Mech* P09008, 2005.
- [20] van Dongen D: Performance criteria for graph clustering and Markov cluster experiments. Technical Report INS-R0012, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, May 2000.
- [21] David Gleich. Hierarchical Directed Spectral Graph Partitioning. <<https://www.cs.purdue.edu/homes/dgleich/publications/Gleich%202005%20-%20hierarchical%20directed%20spectral.pdf>>
- [22] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. 2008. Statistical properties of community structure in large social and information networks. In Proceedings of the 17th international conference on World Wide Web (WWW '08). Association for Computing Machinery, New York, NY, USA, 695–704. DOI:<https://doi.org/10.1145/1367497.1367591>
- [23] R. Sahoo, T.S. Rani, S.D. Bhavani, Chapter 17 - Differentiating Cancer From Normal Protein-Protein Interactions Through Network Analysis, Editor(s): Quoc Nam Tran, Hamid R. Arabnia, In Emerging Trends in Computer Science and Applied Computing, Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology, Morgan Kaufmann, 2016, Pages 253-269, ISBN 9780128042038, <https://doi.org/10.1016/B978-0-12-804203-8.00017-1>. (<http://www.sciencedirect.com/science/article/pii/B9780128042038000171>)
- [24] Guy Melançon. Just how dense are dense graphs in the real world? A methodological note. BELIV 2006: BEyond time and errors: novel evaLuation methods for Information Visualization (AVI Workshop), May 2006, Venice, Italy, pp.75-81. [fflirmm-00091354f](#)
- [25] Chen, Q; Qin, Patia, Chen, J, Wei, Zhiyi, Zhao, Horic, Normand, B, Xiang, Tao. (2011). Partial Order and Finite-Temperature Phase Transitions in Potts Models on Irregular Lattices. *Physical review letters*. 107. 165701. 10.1103/PhysRevLett.107.165701.
- [26] Laura D. Knogler, Andreas M. Kist, Ruben Portugues Motor context dominates output from Purkinje cell functional regions during reflexive visuomotor behaviours *eLife*, 25. January 2019
- [27] Hu, Yifan. “Efficient, High-Quality Force-Directed Graph Drawing.” *The Mathematica journal* 10 (2006): 37-71.