Bioinformatics & Network Medicine

Brain network study during resting states

Laura Laurenti, Paolo Mandica, and Lucia Testa
Group no. 1
IDs 1751854, 1898788, 1764018

Data Science, Sapienza University of Rome, a.y. 2020-21

January 16, 2021

Abstract

The human brain is composed by millions of neurons which are interconnected together and which send and receive signals to initiate actions in the whole human body. To analyze and evaluate electrical activity and flow patterns in the brain, the EEG test is used. In this project, we analyze two datasets of EEG data recorded from 64 electrodes with subjects at rest in (i) eye-open and (ii) eyes-closed conditions, respectively. The analysis includes the topics of connectivity graphs, graph theory indices, motif analysis, and community detection. We also explore the differences of results obtained from the construction of the brain graph using different methods and densities, and we compute motifs and communities between channels in the brain using various algorithms.

Introduction

For this study we used EEG data of PhysioNet "EEG Moto Movement/Imagery Dataset", selecting the subject S002 and the first two runs: R01 (recorded in eyes-open condition), and R02 (recorded in eyes-closed condition). EEG is an electrophysiological monitoring method to record electrical activity of the brain. The main advantages of EEG are that it has the high temporal resolution on the order of milliseconds, its setups are portable and thus usable in realistic situations and it can be used in subjects who are incapable of making a motor response.
We performed several analysis in order to handle and understand these data in the best way possible. Our analyses are respectively on: spectral analysis, connectivity graph, graph theory indices, motif analysis, and community detection.
The list of tasks chosen for the project can be found in Table 1.

1 Connectivity graphs

Spectral density estimation (SDE),in statistical signal processing,means to estimate the spectral density (also known as the power spectral density) of a random signal from a sequence of time

samples of the signal. In order to perform this estimation, for this study, we used a AR/MVAR model. This model is generally used to describe time varying random processes in economics and natural sciences. To estimate the parameters in AR, we used the Yule-Walker equation. In Neuoroscience, when we talk about connectivity, we can refer to an anatomical or functional meaning. In this study, we are dealing with functional connectivity which means the existence of statistical or casual relationship between the activity recorded in different cerebral sites. Starting from data flows for every brain region we need to contruct a graph to follow brain connectivity, in order to understand the network topology and organisation. We used both Partial Directed Coherence (PDC) and Directed Transfer Function (DTF).

Introduced by Baccala and Sameshima in 2001 in the form [1]:

$$P_{ij}(f) = \frac{A_{ij}(f)}{\sqrt{(\boldsymbol{a}_j^*(f)\boldsymbol{a}_j(f)}}$$

where, Aij(f) is an element of A(f) - a Fourier transform of MVAR model coefficients A(t), where aj(f) is j-th column of A(f) and the asterisk denotes the transpose and complex conjugate operation, PDC has higher performances when the aim is to accurately reconstruct the exact pattern of interactions between signals and distinguish the direct influence from those mediated by other signals.

Instead, introduced ten years earlier by Kaminski and Blinowska, 1991 in the form [2]:

$$\theta_{ij} = \frac{|H_{ij}(f)|^2}{\sum_{m=1}^{L} |H_{mi}(f)|^2}$$

DTF measures the overall effect of one signal on another, considering both direct and indirect paths.

Whenever we identify a AR / MVAR model to estimate connectivity between signals, we must choose the order for the model since each dataset has its optimal order and the "real" order of the model for a given dataset is not known a priori. In our implementation, before proceeding with connectivity estimation, we used Akaike Information Criterion (AIC). Once estimated the "real" order for our datasets (see Table 2 and Figures 9 ,10) we used Connectipy library to perform both PDC and DTF on open eyes data and closed eyes data (see Figures 1 and 2). The PDC analysis has been performed on two different frequency values, 10Hz and 30Hz and a density threshold for connectivity matrix of 20% (see Figures 1, 8a, 8b). Maintaining the frequency value at 10Hz we also performed PDC connectivity analysis using as density thresholds 1%, 5%, 10%, 30%, 50% see (Figures 3 and 4).

Considering the following subset of 19 channels:

$$Fp1\ Fp2\ F7\ F3\ Fz\ F4\ F8\ T7\ C3\ Cz\ C4\ T8\ P7\ P3\ Pz\ P4\ P8\ O1\ O2$$

, calculating again the order for both open eyes and closed eyes (see Table 2 and Figure 10), we estimated again the PDC connectivity, using the resulting data to estimate the significance matrix (the p-value matrix for each couple of channels) From this significance matrix we took only those p-values less than 5% corresponding to those PDC(i,j) $\neq 0$ (see Figure 5).

Beyond these results we wanted to understand better the representation of connections between channels, so we generated topographical network plots for both PDC and DTF on the two considered resting states(see Figures 6, 7).

2 Graph theory indices

The second set of tasks (see Table), which we performed over our EGG data, consisted in computing and analyzing binary local (degree, in/out-degree) and global (average clustering coefficient, average path length) graph theory indices.
To perform the analysis, we created a new *GraphTheoryIndices* class, which extended the *ConnectivityGraph* class previously seen and included a set of methods to accomplish our goals. We also took advantage of multiple libraries and packages (*numpy, connectivipy, bctpy, networkx*).
A total of 6 tasks (Table 1) were completed for this section of the project. When it is not specified, the connectivity graph is computed with density 0.2, sample frequency 10 and *PDC* method.

2.1 Binary global and local indices

The goal of this task was to compute binary global (average clustering coefficient, average path length) and local (degree, in/out-degree) graph indices.
To do this we exploited *networkx* and its methods to work on our graph.

Results

For the *open eyes* graph we obtained an **average clustering coefficient** of 0.279 and an **average path length** of 2.533.
For the *closed eyes* graph we obtained an **average clustering coefficient** of 0.276 and an **average path length** of 2.115.
The local indices results, in terms of degree, in-degree and out-degree (first 10 channels for each index in descending order of degree), are present in tables 3 and 4.

2.2 Small-worldness index

The goal of this task was to search in the literature a definition of small-worldness index (i.e. an index describing the small-world organization of a network) and to compute it.

A **small-world network** is a type of mathematical graph in which most nodes are not neighbors of one another, but the neighbors of any given node are likely to be neighbors of each other and most nodes can be reached from every other node by a small number of hops or steps. This means that, despite the size of the network, each pair of nodes is connected by a not large number of edges [3].

A network of "large" dimensions has the property of a small world (small world) if its average path length is "small":

$$L << N \text{ per } N >> 1$$

with L = average path length and N = number of nodes in the network.
Finally, a network G is defined small-world if $L_G > L_{rand}$ and $C_G >> C_{rand}$ where:

- L_G, C_G represent respectively the average path length and the clustering coefficient of the graph G;
- L_{rand}, C_{rand} represent respectively the average path length and the clustering coefficient in random graphs.

The **small-worldness index** is defined as:

$$S = \frac{C_G/C_{rand}}{L_C/L_{rand}}, \text{ with } S >> 1.$$

Results

The *open eyes* graph has a **small-worldness index** of 0.965, while the *closed eyes* graph's one is 1.054. Both the networks have $S \approx 1$. This means that we cannot precisely classify them as small world networks.

2.3 Binary global and local indices (DTF)

The goal of this task was to compare the global indices extracted from PDC and DTF connectivity estimations. The results from PDC are present is task 2.1. Here are the ones obtained from the DTF method:

Results

For the *open eyes* graph we obtained an **average clustering coefficient** of 0.454 and an **average path length** of 0.892.
For the *closed eyes* graph we obtained an **average clustering coefficient** of 0.461 and an **average path length** of 0.617.
The local indices results, in terms of degree, in-degree and out-degree (first 10 channels for each index in descending order of degree), are present in tables 5 and 6.

2.4 Global indices behaviour

The goal of this task was to study the behaviour of global graph indices in function of network density (see point 1.3 for density values).
The resulted plots of the behaviours can be observed in Fig. 11 and Fig. 12.

2.5 Topographical representations of local indices

The goal of this task was to make a topographical representation of local indices. Specifically, we computed and visualized topological representations of *degree, in-degree and out-degree*, for both *open eyes* and *closed eyes* data (see figures 13, 14, 15, 16, 17, 18).

2.7 Binary global and local indices of weighted graph

The goal of this task was to replicate the computations of task 2.1 considering the weighted version of the graph indices definition.

Results

For the *open eyes* graph we obtained an **average clustering coefficient** of 0.071 and an **average path length** of 0.358.
For the *closed eyes* graph we obtained an **average clustering coefficient** of 0.163 and an **average path length** of 0.099.
The local indices results, in terms of degree, in-degree and out-degree (first 10 channels for each index in descending order of degree), are present in tables 7 and 8.

3 Motif analysis

For the detection of motifs we used the netsci package (https://github.com/gialdetti/netsci). The authors propose 2 algorithms to use in the optimization problem: brute-force and louzoun. According with their documentation:

- 'brute-force' - the naive implementation using brute force algorithm. The complexity is high ($O(|V|^3)$), counts all 16 triplets.
- 'louzoun' - an efficient algorithm for sparse networks. The complexity is low ($O(|E|)$), but it counts only the 13 connected triplets (the first 3 entries will be -1).

The results are in figure 19 and 20 for the brute force algorithm and in figure 21 and 22 for the louzoun algorithm. In the figures there are also barplots to understand motifs' frequency and statistical significance.
A topographical representation of the networks considering only the connections involved in the motif with pattern A $\rightarrow$ B $\leftarrow$ C are in figure 23 for eyes closed and in figure 24 for eyes open. Note that this analysis was performed using the results from louzoun algorithm, since the netsci package doesn't allow to do the same with brute force algorithm.

4 Community detection

For the detection of motifs we used the Louvain algorithm and the Infomap algorithm.

4.1 Dense graph

At the very begin we used the same graph we considered in previous analysis, with density equal to 20 %, obtaining the following results.
Using the Louvain algorithm we find 5 communities in the case of eyes closed, represented in figure 25, and 3 communities in the case of eyes open, represented in figure 26.
In table 9 there is the list for closed eyes using Louvain algorithm.
In table 10 there is the list for open eyes using Louvain algorithm.

Using the Infomap algorithm we find again 4 communities in the case of eyes closed, represented in figure 27, but this time only 2 communities in the case of eyes open, represented in figure 28. In table 11 there is the list for closed eyes using Infomap algorithm.
In table 12 there is the list for open eyes using Infomap algorithm.

4.2 Sparse graph

In binary network, the identification of communities makes sense only if graphs are sparse, i. e. if the number of edges m is of the order of the number of nodes n. Hence we create new graphs with density equal to 5 %, obtaining the following results.
Using the Louvain algorithm we find 6 communities in the case of eyes closed, represented in figure 29, and 6 communities in the case of eyes open, represented in figure 30.
In table 13 there is the list for closed eyes using Louvain algorithm.
In table 14 there is the list for open eyes using Louvain algorithm.

Using the Infomap algorithm we find 13 communities in the case of eyes closed, represented in figure 31 and 12 communities in the case of eyes open, represented in figure 32.

In table 15 there is the list for closed eyes using Infomap algorithm.
In table 16 there is the list for open eyes using Infomap algorithm.

5 Conclusions

Network analysis resulted to be an essential tool to understand better functional and anatomical brain connectivity. PDC and DTF methods allowed us to understand the basis of brain networks, and have been very useful to perform a complete analysis of global and local graph indices. At the end, we exploited all our new generated information to understand the network structure and node configurations and to find communities, using different algorithms, in our networks.

References

[1] Baccalá LA, Sameshima K. "Partial directed coherence: a new concept in neural structure determination". In: *Biol Cybern* (2001). DOI: 10.1007/PL00007990.

[2] Kaminski, M.J., Blinowska, K.J. "A new method of the description of the information flow in the brain structures". In: *Biol Cybern* (1991). DOI: 10.1007/BF00198091.

[3] S. Milgram. "Six degrees of separation". In: (1967).

Tables

Task	*Class*
1.1	*mandatory*
1.2	*A*
1.3	*A*
1.4	*D*
1.5	*C*
1.6	*B*
2.1	*mandatory*
2.2	*D*
2.3	*B*
2.4	*C*
2.5	*B*
2.7	*C*
3.1	*mandatory*
3.2	*C*
4.1	*mandatory*
4.2	*B*
4.3	*C*

Table 1: List of tasks chosen for the project.

Resting state	**# channels**	**Order**
open eyes	64	8
closed eyes	64	7
open eyes	19	17
closed eyes	19	17

Table 2: Order value for each resting state datasets with different number of channels.

Index	**List of (channel, degree)**
Degree	Af7 (45), T7 (43), Iz (42), Oz (41), Cp6 (39), P4 (38), F7 (37), C4 (34), Cp1 (34), Af8 (34)
In-degree	Cp6 (32), P4 (30), Iz (30), C4 (29), Oz (29), P7 (26), Cp1 (25), C3 (24), F2 (23), Po3 (23)
Out-degree	T7 (43), Af7 (38), Af8 (32), F7 (30), Fp1 (26), Fp2 (23), F8 (21), Fpz (20), Af4 (20), F5 (20)

Table 3: (Tasks 2.1) First 10 channels for each local index, in descending order of degree, for *open eyes* data.

Index	**List of (channel, degree)**
Degree	Iz (49), Oz (45), Poz (44), Po4 (44), T7 (37), Po8 (37), Fc3 (36), C4 (36), O2 (36), Af7 (34)
In-degree	Iz (33), C4 (30), Oz (30), Po4 (29), Fc3 (26), Poz (26), P4 (24), P8 (24), Po8 (24), Cp6 (23)
Out-degree	T7 (37), Af7 (28), O2 (25), Fp1 (24), Af3 (23), F5 (21), Fp2 (20), O1 (20), Af8 (18), F1 (18)

Table 4: (Tasks 2.1) First 10 channels for each local index, in descending order of degree, for *closed eyes* data.

Index	**List of (channel, degree)**
Degree	Oz (73), Iz (71), P8 (69), Po4 (68), C4 (64), Poz (56), Fc3 (52), P4 (52), P6 (52), Po8 (48)
In-degree	Oz (63), Iz (63), P8 (61), Po4 (61), C4 (54), Poz (48), P4 (43), P6 (43), Cp6 (39), P7 (39)
Out-degree	Af7 (20), Fp1 (19), Fpz (18), Fp2 (18), Af3 (17), Afz (17), F7 (16), F5 (16), F3 (16), Fc5 (15)

Table 5: (Tasks 2.3) First 10 channels for each local index, in descending order of degree, for *open eyes* data, from *DTF* method.

Index	**List of (channel, degree)**
Degree	Oz (73), Iz (71), P8 (69), Po4 (68), C4 (64), Poz (56), Fc3 (52), P4 (52), P6 (52), Po8 (48)
In-degree	Oz (63), Iz (63), P8 (61), Po4 (61), C4 (54), Poz (48), P4 (43), P6 (43), Cp6 (39), P7 (39)
Out-degree	Af7 (20), Fp1 (19), Fpz (18), Fp2 (18), Af3 (17), Afz (17), F7 (16), F5 (16), F3 (16), Fc5 (15)

Table 6: (Tasks 2.3) First 10 channels for each local index, in descending order of degree, for *closed eyes* data, from *DTF* method.

Index	**List of (channel, degree)**
Degree	Af7 (45), T7 (43), Iz (42), Oz (41), Cp6 (39), P4 (38), F7 (37), C4 (34), Cp1 (34), Af8 (34)
In-degree	Cp6 (32), P4 (30), Iz (30), C4 (29), Oz (29), P7 (26), Cp1 (25), C3 (24), F2 (23), Po3 (23)
Out-degree	T7 (43), Af7 (38), Af8 (32), F7 (30), Fp1 (26), Fp2 (23), F8 (21), Fpz (20), Af4 (20), F5 (20)

Table 7: (Tasks 2.7) First 10 channels for each local index, in descending order of degree, for *open eyes* data, from weighted graph.

Index	**List of (channel, degree)**
Degree	Oz (73), Iz (71), P8 (69), Po4 (68), C4 (64), Poz (56), Fc3 (52), P4 (52), P6 (52), Po8 (48)
In-degree	Oz (63), Iz (63), P8 (61), Po4 (61), C4 (54), Poz (48), P4 (43), P6 (43), Cp6 (39), P7 (39)
Out-degree	Af7 (20), Fp1 (19), Fpz (18), Fp2 (18), Af3 (17), Afz (17), F7 (16), F5 (16), F3 (16), Fc5 (15)

Table 8: (Tasks 2.1) First 10 channels for each local index, in descending order of degree, for *closed eyes* data, from weighted graph.

Community	**List of channels (nodes)**
0	Fc2 Fc4 Fc6 Cz C2 C4 Cpz Cp6 F4 F8 Ft8 T8 T10 Tp8 P2 P4
1	C1 C6 Cp1 Tp7 P7 P5 P1 Pz P6 P8 Po7 Poz Po4 Po8 O1 Oz O2 Iz
2	Fc1 Fcz Cp4 Fp1 Fpz Fp2 Af7 Af3 Afz Af4 F7 F1 Fz F2 F6 Po3
3	Fc5 Fc3 C5 C3 Cp5 Cp3 Cp2 Af8 F5 F3 Ft7 T7 T9 P3

Table 9: (Tasks 4) Dense graph. Nodes list for closed eyes using Louvain algorithm.

Community	List of channels (nodes)
0	Cz C2 Cp1 Cpz Cp2 T7 Tp8 P7 P5 Pz P2 P4 P6 P8Po7 Poz Po4 Po8 O1 Oz O2 Iz
1	Fcz Fc2 Fc4 Fc6 C4 C6 Cp4 Cp6 Af4 Af8 Fz F2 F4 F6F8 Ft8 T8 T10
2	Fc5 Fc3 Fc1 C5 C3 C1 Cp5 Cp3 Fp1 Fpz Fp2 Af7 Af3Afz F7 F5 F3 F1 Ft7 T9 Tp7 P3 P1 Po3

Table 10: (Tasks 4) Dense graph. Nodes list for open eyes using Louvain algorithm.

Community	List of channels (nodes)
0	z Oz P4 Po8 Cp6 Poz C3 C4 C6 P6 Po4 P8 Pz Fc3 P7Cp1 F4 P3 P1 C1 O1 Cz Fc4 Fc2 Cp2 O2 Fz Cp4 F1 C2 Po7 F2 Fc6 Cp3 Po3 Fc1 Fcz AfzCp5 P2 Cpz F3 Tp8 F6 Tp7 Af4
1	Fc5 Ft7 T9 C5 F7
2	Ft8 Af7 Fpz T8 T10 Fp2 Af3 Fp1 T7
3	P5 F8 Af8 F5

Table 11: (Tasks 4) Dense graph. Nodes list for closed eyes using Infomap algorithm.

Community	List of channels (nodes)
0	z Iz Oz Po7 P7 P4 Po4 Cp1 C4 Poz Cp6 P8 Po3 P6 O1Cz Pz F2 Po8 F4 P2 O2 C3 P1 P3 Fc6 Fcz Fz Cp4 Tp7 C2 Fc1 Cp2 C6 Afz F1 Fc4 F3 Cp3F6 Cp5 Fc2 Fc5 Fc3 T9 P5 C5 Ft7
1	F8 F7 Fpz Af7 C1 Af3 Fp2 T8 F5 Ft8 Cpz T10 Fp1Tp8 Af8 Af4 T7

Table 12: (Tasks 4) Dense graph. Nodes list for open eyes using Infomap algorithm.

Community	List of channels (nodes)
0	T9
1	Fc3 Fc2 Af3 F5 F3 F1
2	T8 Tp8 P7 P5 P2 P6 P8 Po7 Po3 Po4 Po8 O1 Oz O2 Iz
3	Fc1 Fcz Fc4 Fp1 Fpz Fp2 Afz Af4 Af8 Fz F2 F4 F6Ft8 T10
4	Fc5 C5 C3 C1 Cp5 Cp3 Cp1 Cp2 Af7 F7 Ft7 T7 Tp7P3 P1 Poz Iz
5	Fc6 Cz C2 C4 C6 Cpz Cp4 Cp6 F8 Pz P4

Table 13: (Tasks 4) Sparse graph. Nodes list for closed eyes using Louvain algorithm.

Community	**List of channels (nodes)**
0	Fc5 C5 C3 C1 Cp5 Cp3 F7 Ft7 T9 P3 P1
1	F1 T7 Tp8 P6 P8 Po7 Poz Po4 Po8 O1 Oz O2 Iz
2	Fc3 Fc1 Cp2 Fp1 Fpz Fp2 Af7 Af3 Afz F5 F3 Po3 P1
3	Fcz Fc2 Fc4 Fc6 C4 Af4 Af8 Fz F2 F4 F6 F8 Ft8 T10
4	Tp7 P7 P5
5	Cz C2 C6 Cp1 Cpz Cp4 Cp6 T8 Pz P2 P4

Table 14: (Tasks 4) Sparse graph. Nodes list for open eyes using Louvain algorithm.

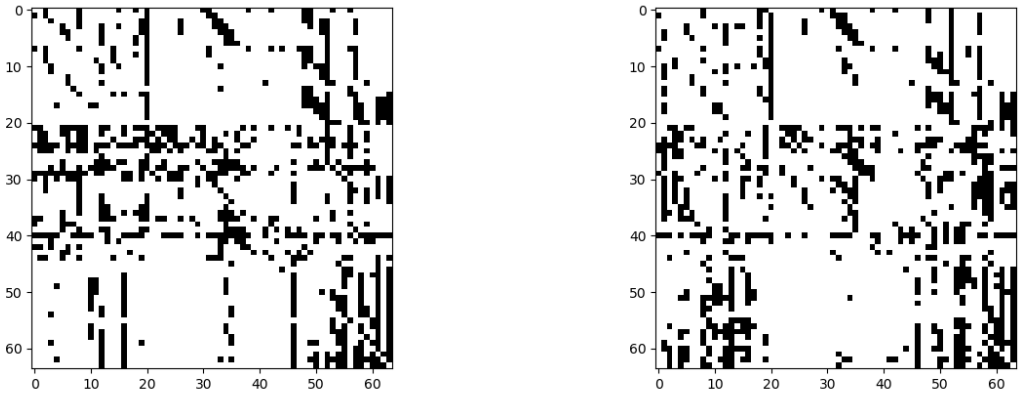
Community	**List of channels (nodes)**
0	P8 Po8 Po4 P6
1	O1 Po7 P7 P5
2	P1 C3 P3 Poz Cp3
3	F4 F2 Fcz Fz F6 Fp2 Af8 Afz Af4
4	P4 Cp6 C4 Fc6 Cp4 Fc2 Cp2 Fc4 C6 C2 F8
5	F1 Po3 F3 Fc1 Fc3 F5
6	Ft7 Af7 Fc5 C5 Fpz Af3 F7
7	Pz Cz Cpz Cp1
8	Ft8 T10
9	Cp5 C1 Tp7 T7
10	Oz Iz Fp1 O2
11	Tp8 P2 T8
12	T9

Table 15: (Tasks 4) Sparse graph. Nodes list for closed eyes using Infomap algorithm.

Community	**List of channels (nodes)**
0	Cp1 P3 P1 C1 Cp5 Cp3
1	P8 Po8 Po4 Poz Tp8 P6
2	Po7 O1
3	P4 F2 C4 F4 Fc4 Fc6 F6 F8 Fc2 Af4
4	F3 C3 Fc3 Fpz Po3 Af3 Fp2 Fc5 Af7 Afz Fp1 F5 Af8
5	Ft7 T9 F7 C5
6	Fz Fc1 Fcz
7	P2 Cp2 Cz Pz Cpz C2
8	Cp6 C6 Cp4
9	P7 Tp7 P5
10	Oz Iz F1 O2 T7
11	Ft8 T8 T10

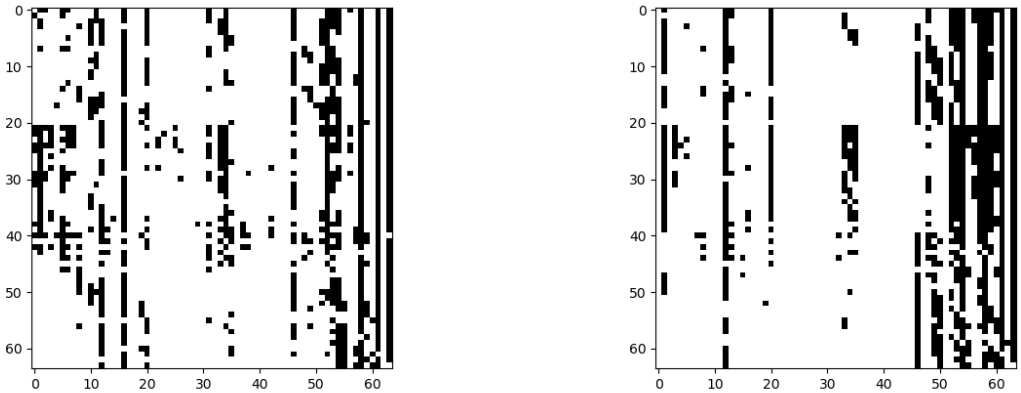
Table 16: (Tasks 4) Sparse graph. Nodes list for open eyes using Infomap algorithm.

Figures



(a) Open eyes. (b) Closed eyes.

Figure 1: PDC adjacency matrices with frequency value = 10 Hz and density = 20%.



(a) Open eyes. (b) Closed eyes.

Figure 2: DTF adjacency matrices with frequency value = 10 Hz and density = 20%.

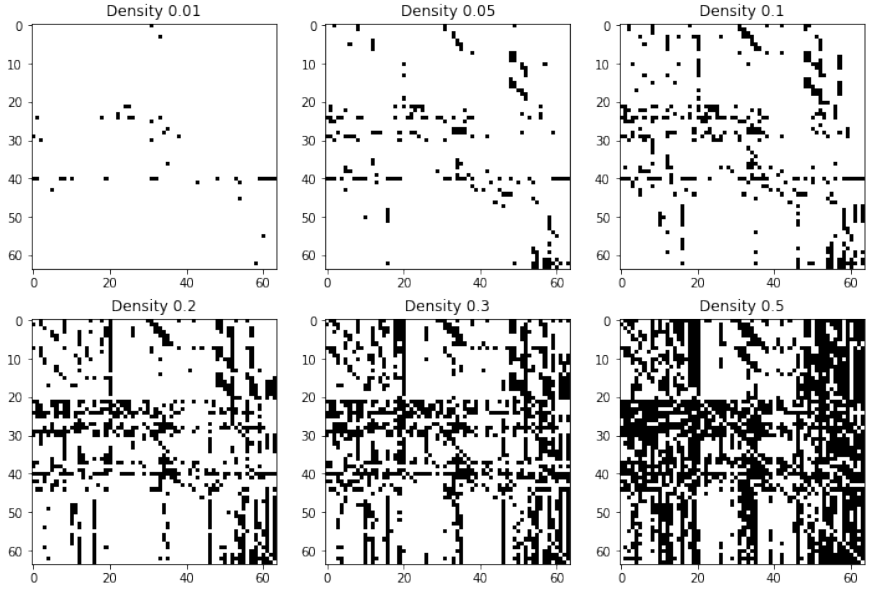


Figure 3: Open eyes PDC adjacency matrices with frequency value = 10 Hz and several densities values.

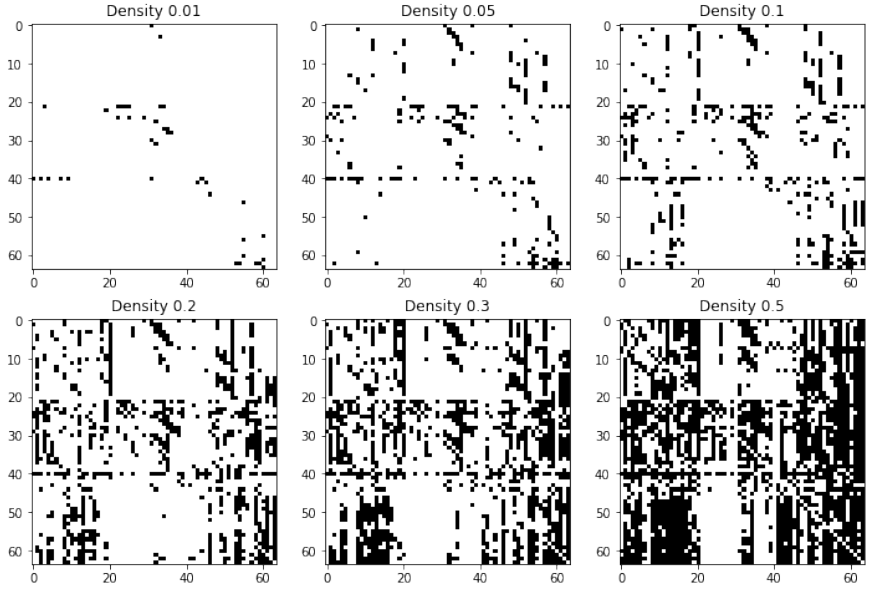
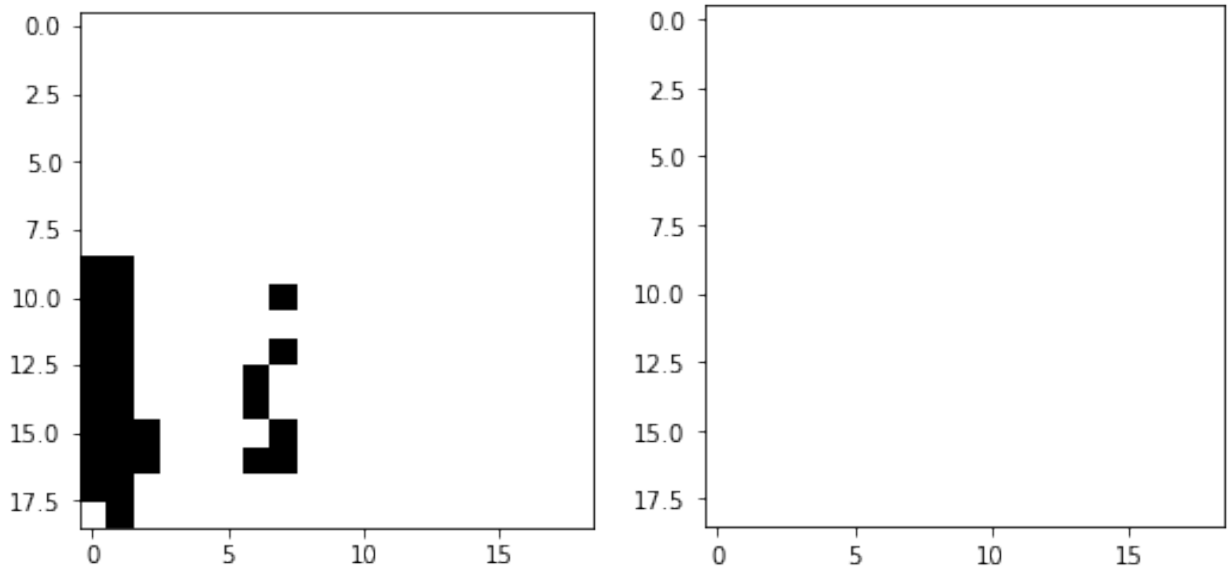
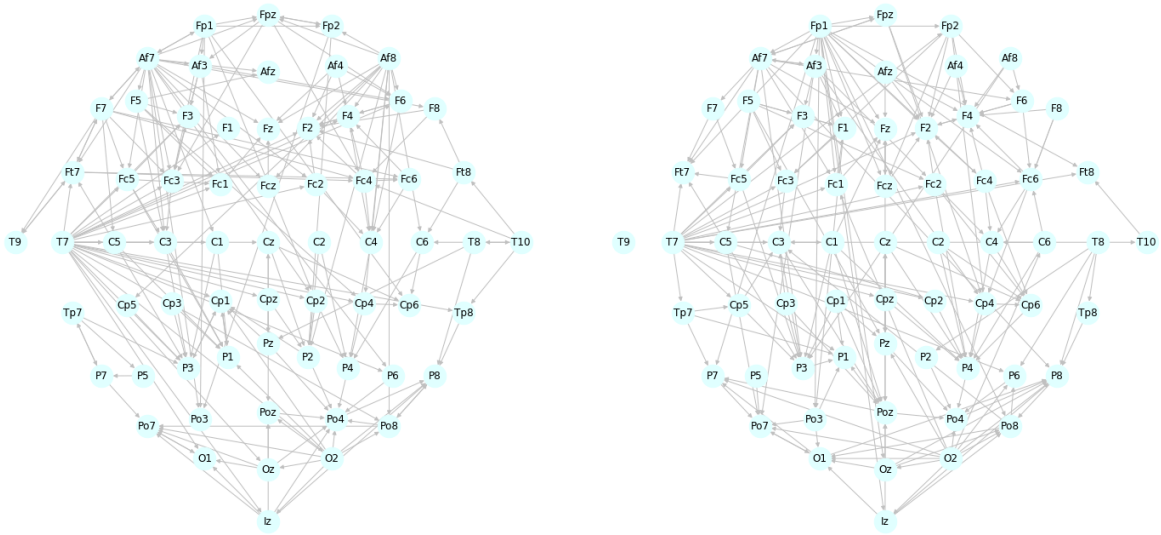


Figure 4: Closed eyes Open eyes PDC adjacency matrices with frequency value = 10 Hz and several densities values.



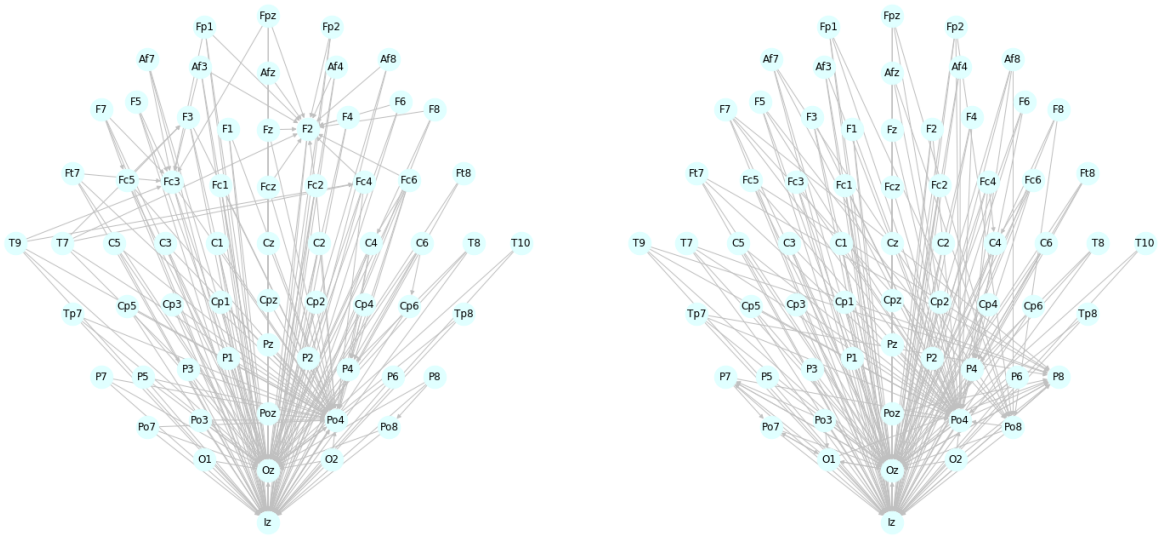
(a) Open eyes. (b) Closed eyes.

Figure 5: 19 channels data PDC adjacency matrices filtered without not significant values



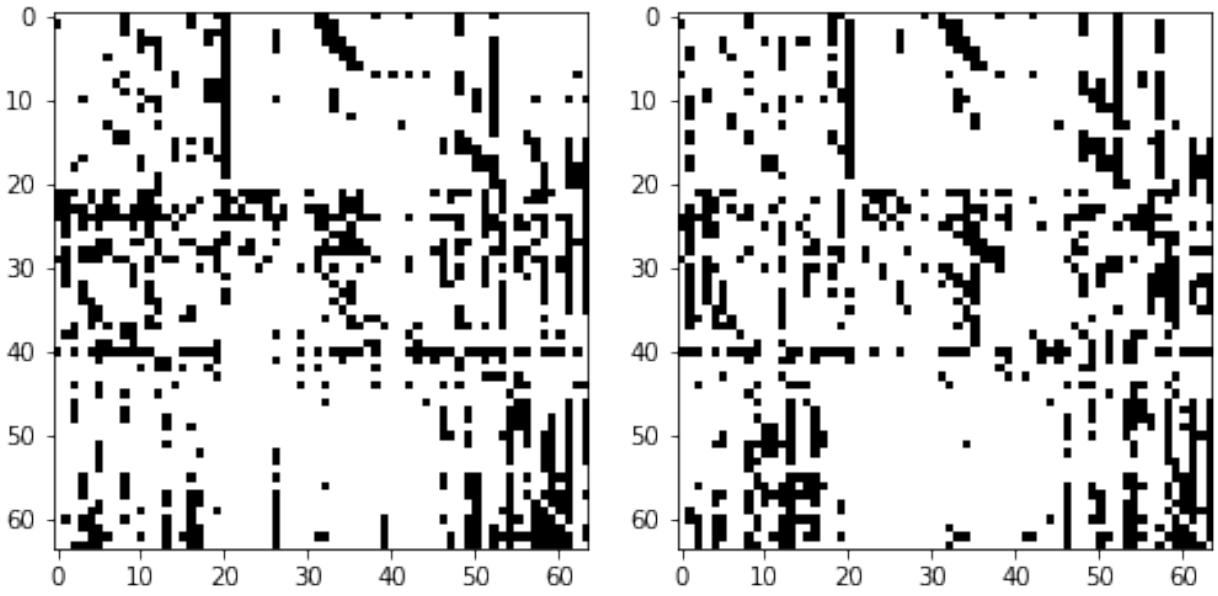
(a) Open eyes. (b) Closed eyes.

Figure 6: Topographical representation of PDC network with a density value = 5%



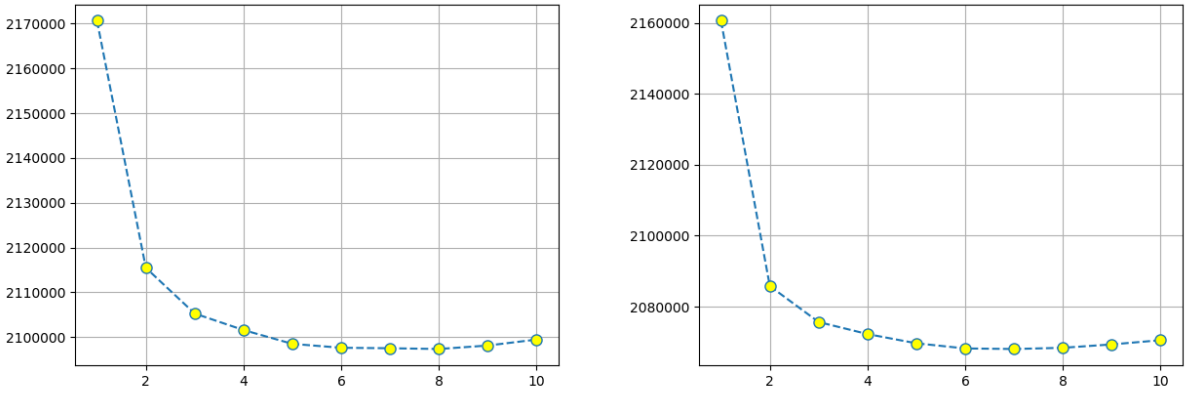
(a) Open eyes. (b) Closed eyes.

Figure 7: Topographical representation of DTF network with a density value = 5%



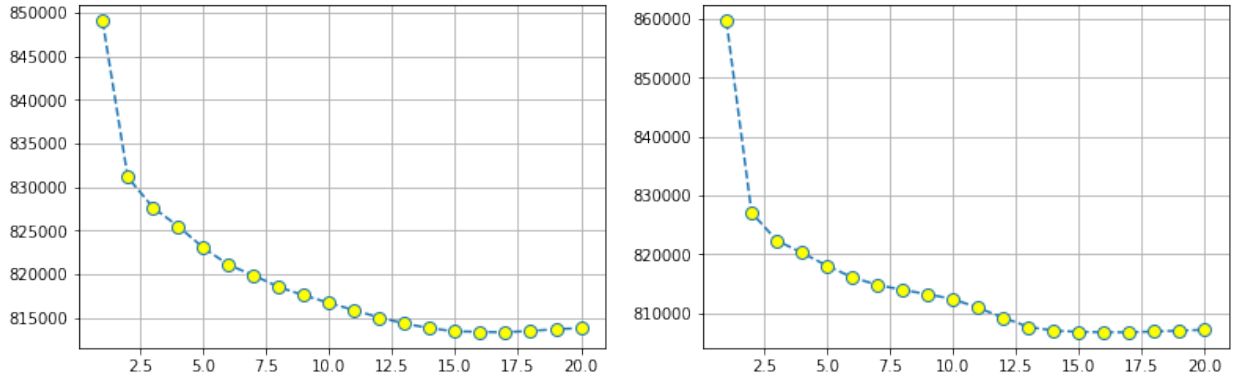
(a) Open eyes. (b) Closed eyes.

Figure 8: PDC adjacency matrices with frequency value = 30 Hz and density = 20%.



(a) Open eyes. (b) Closed eyes.

Figure 9: Orders behaviour using Akaike algorithm for 64 channels data



(a) Open eyes. (b) Closed eyes.

Figure 10: Orders behaviour using Akaike algorithm for 19 channels data

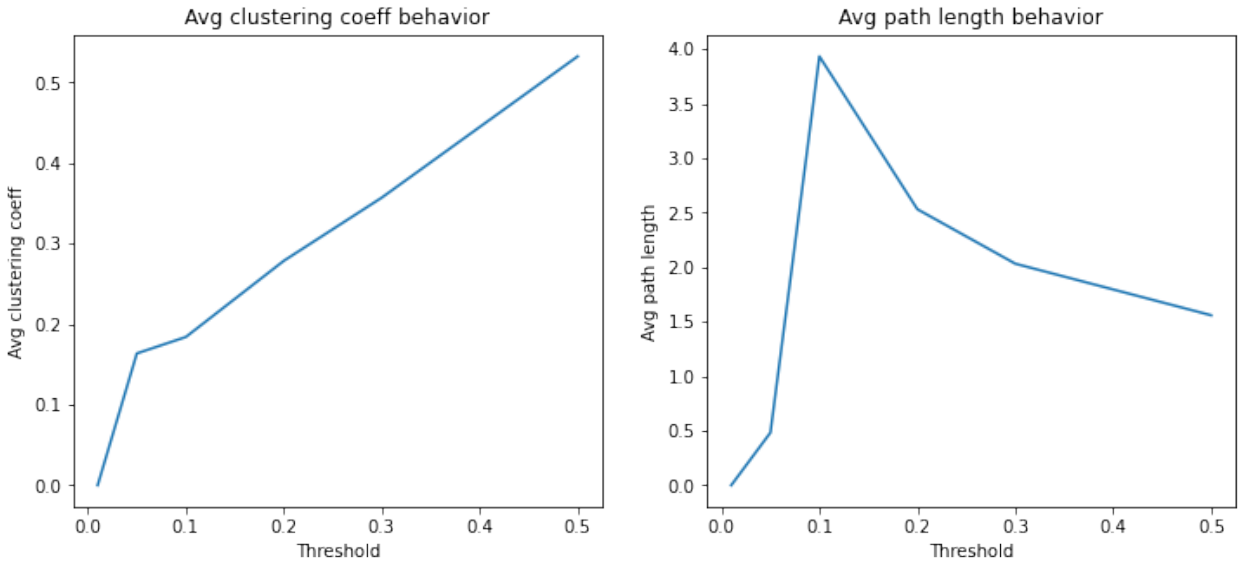


Figure 11: Global graph indices behaviour for *open eyes* data.

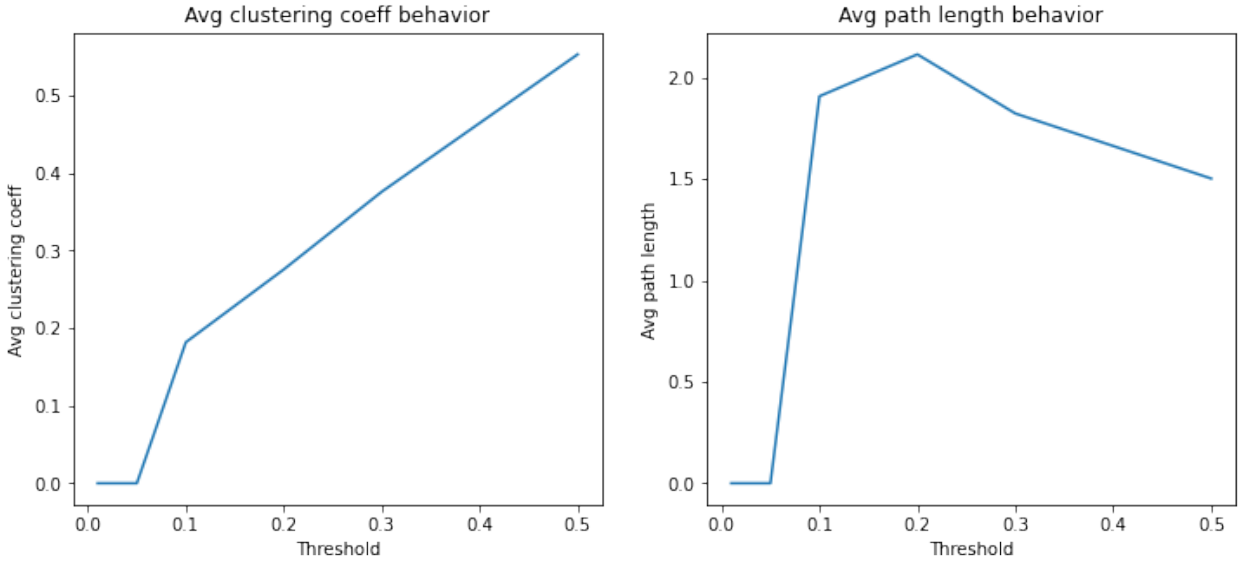


Figure 12: Global graph indices behaviour for *closed eyes* data.

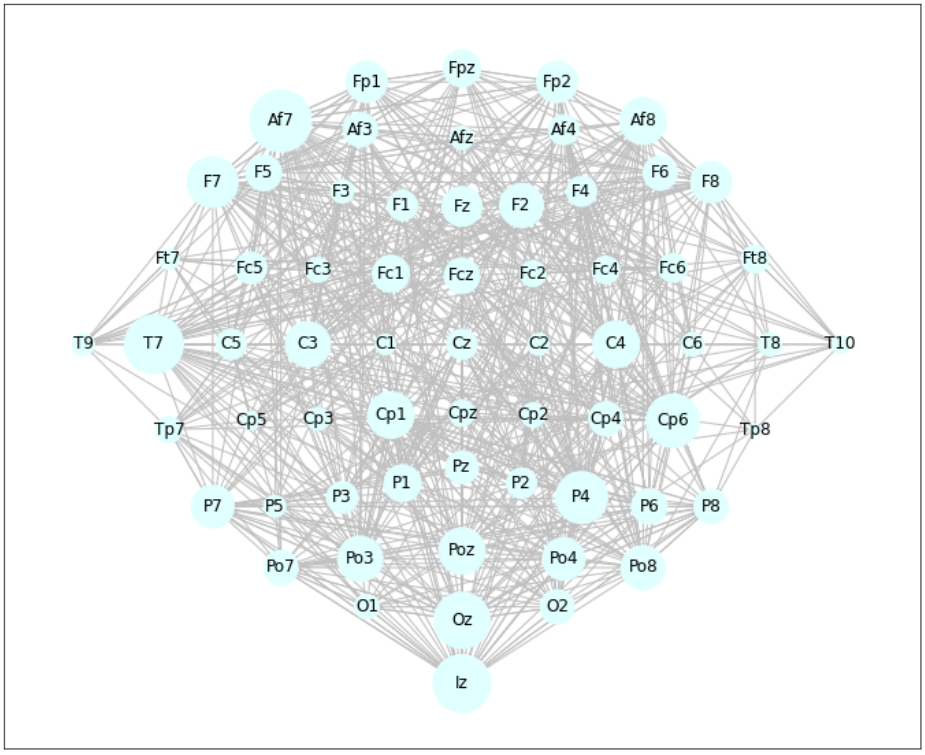


Figure 13: Topological representation of *degree* for *open eyes* data.

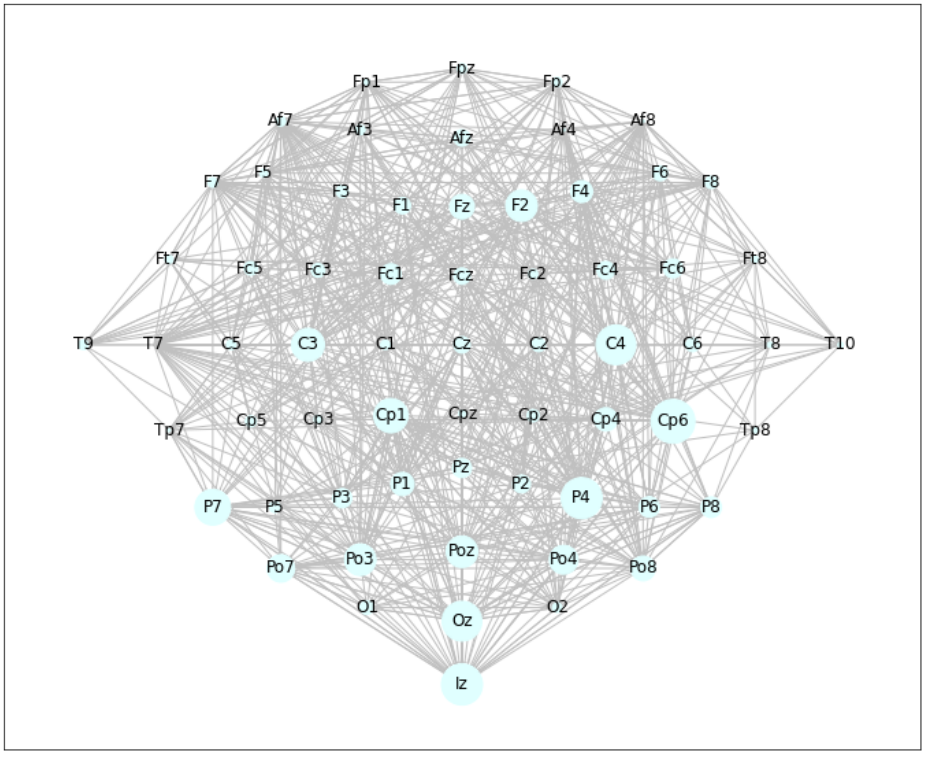


Figure 14: Topological representation of *in-degree* for *open eyes* data.

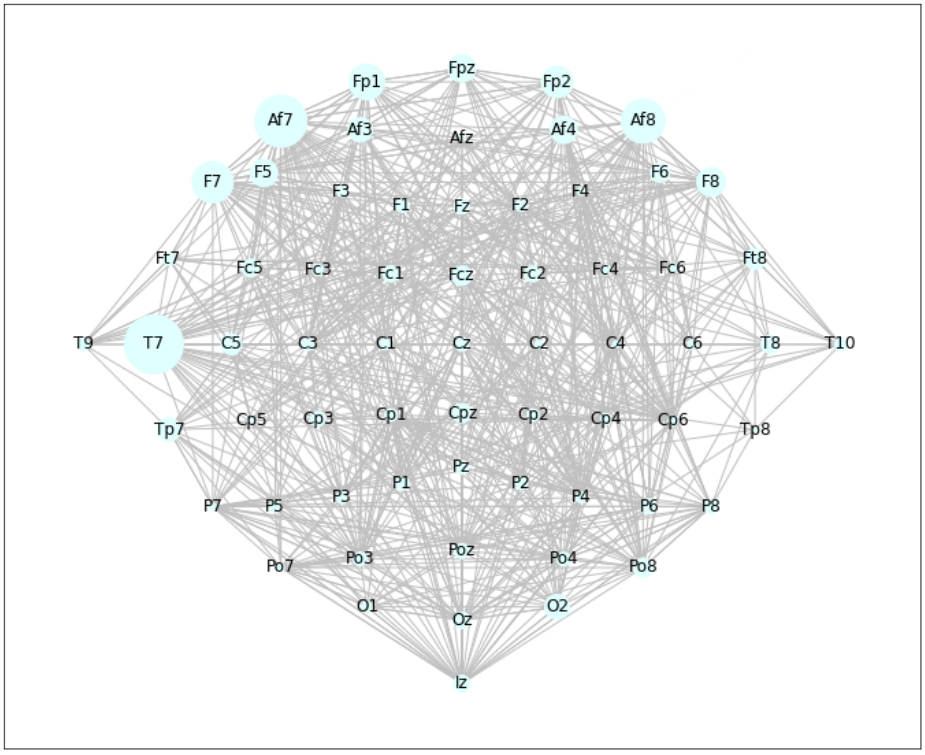


Figure 15: Topological representation of *out-degree* for *open eyes* data.

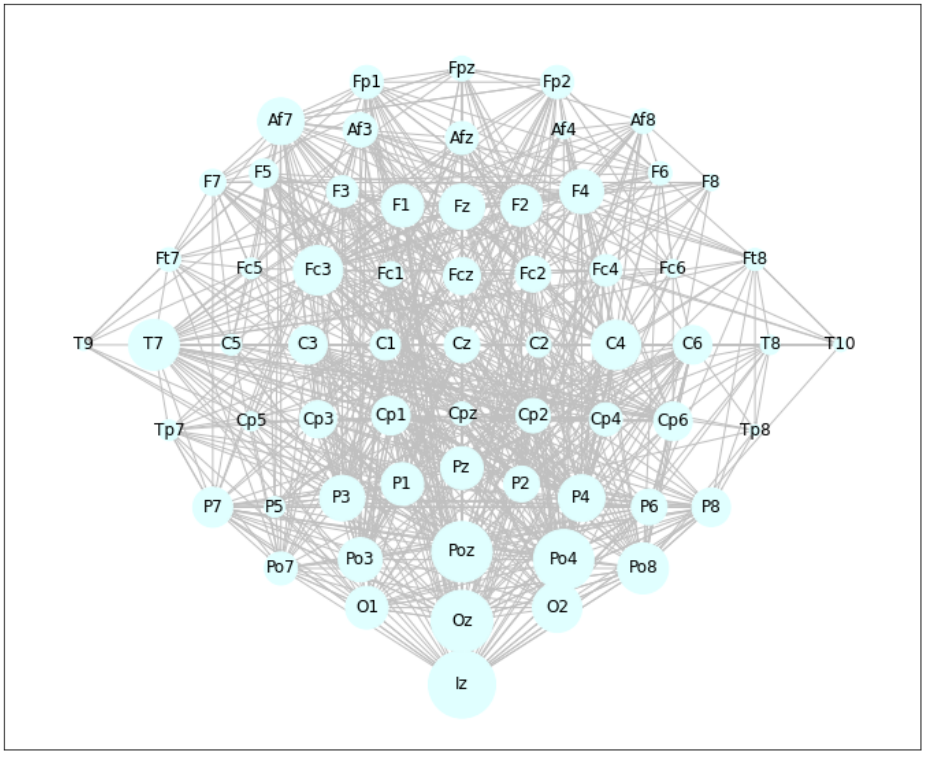


Figure 16: Topological representation of *degree* for *closed eyes* data.

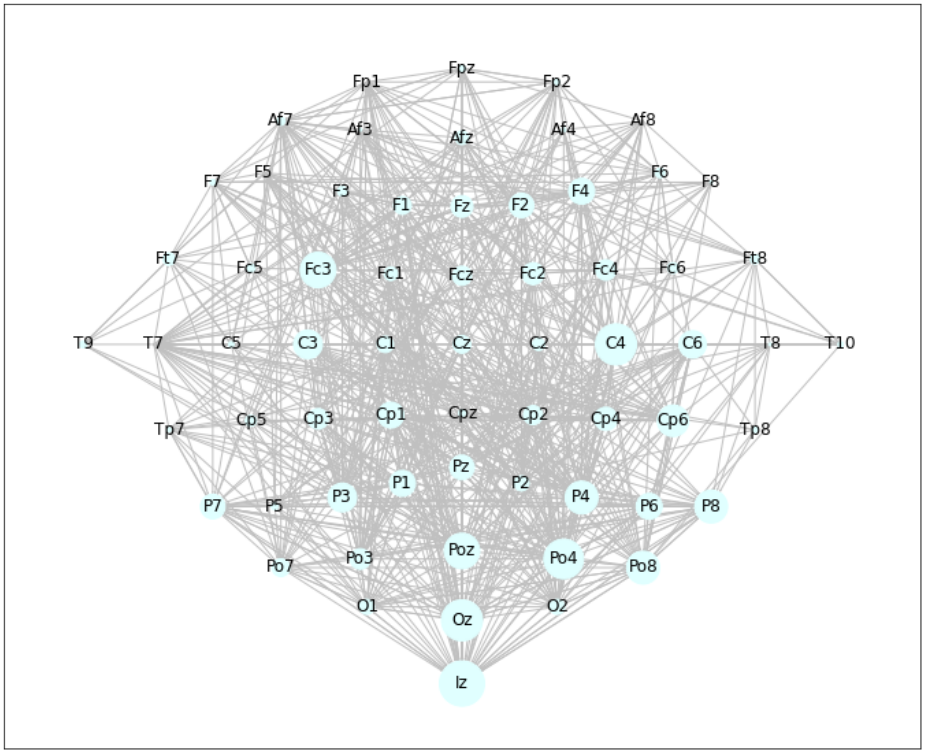


Figure 17: Topological representation of *in-degree* for *closed eyes* data.

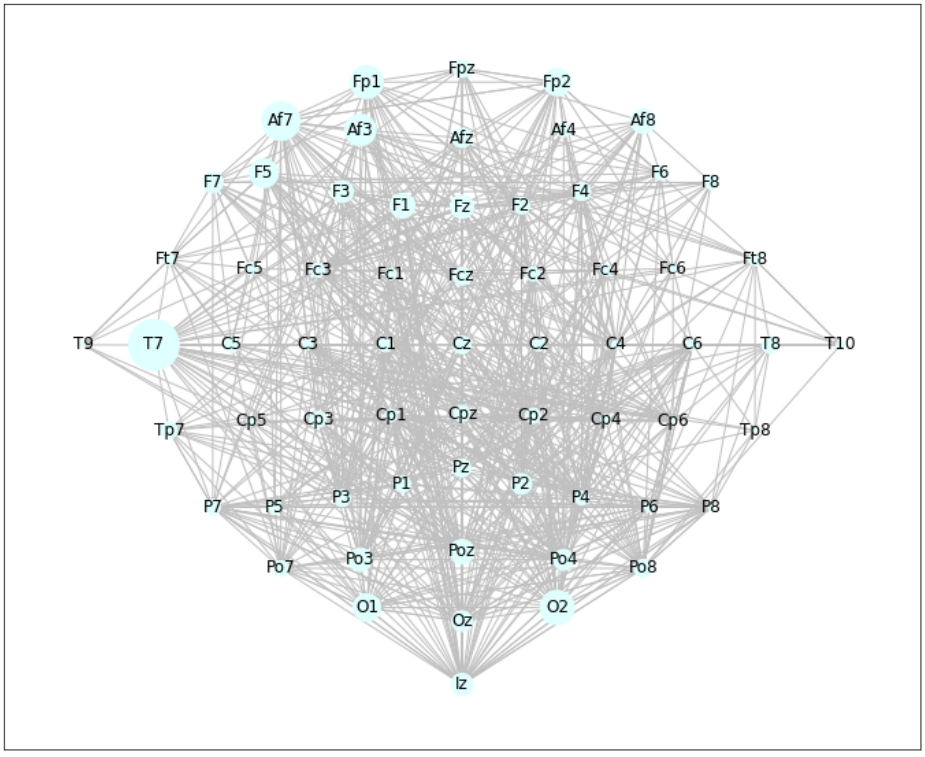


Figure 18: Topological representation of *out-degree* for *closed eyes* data.

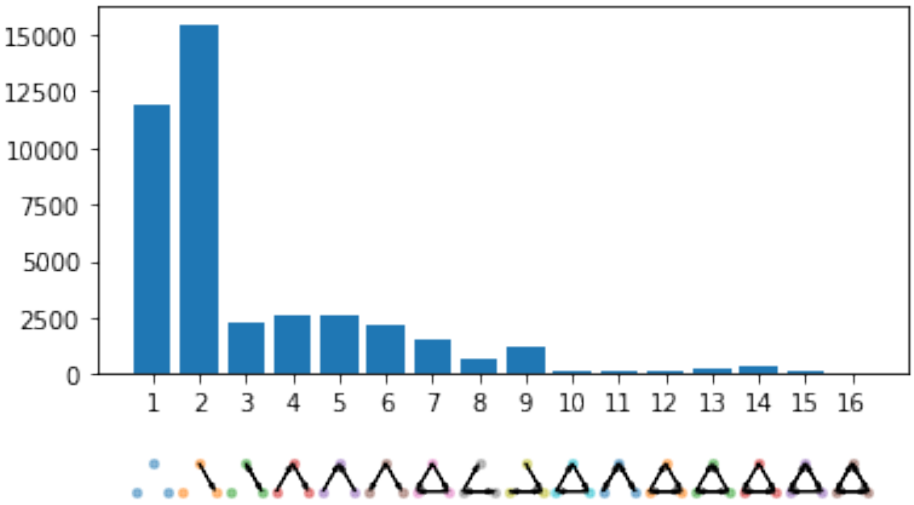


Figure 19: Motifs for eyes closed using 'brute-force' algorithm

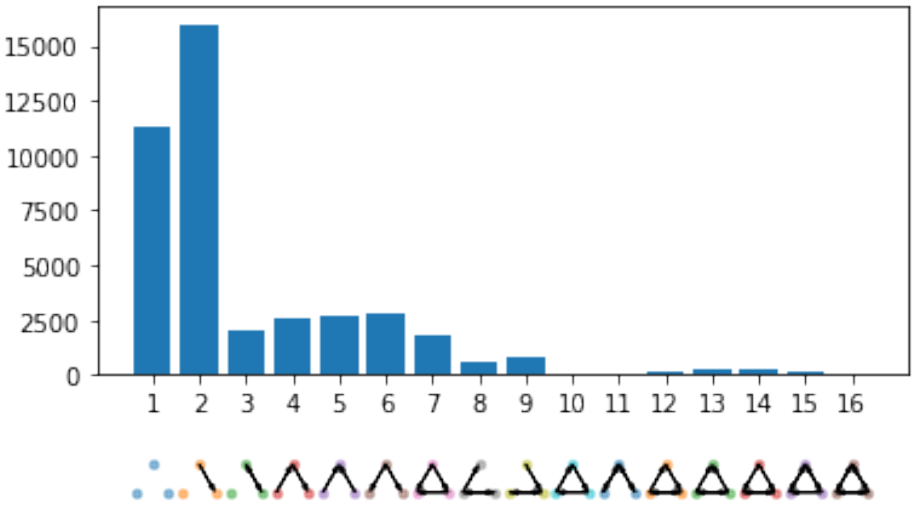


Figure 20: Motifs for eyes open using 'brute-force' algorithm

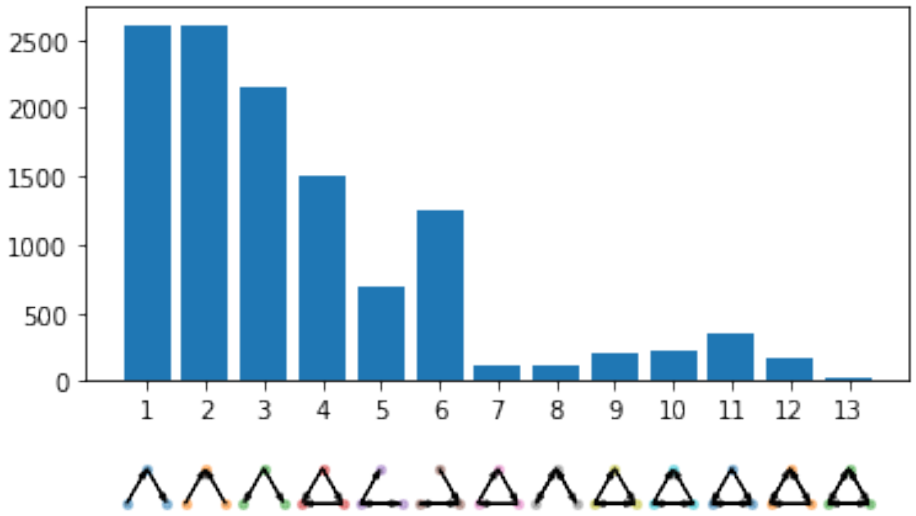


Figure 21: Motifs for eyes closed using 'louzoun' algorithm

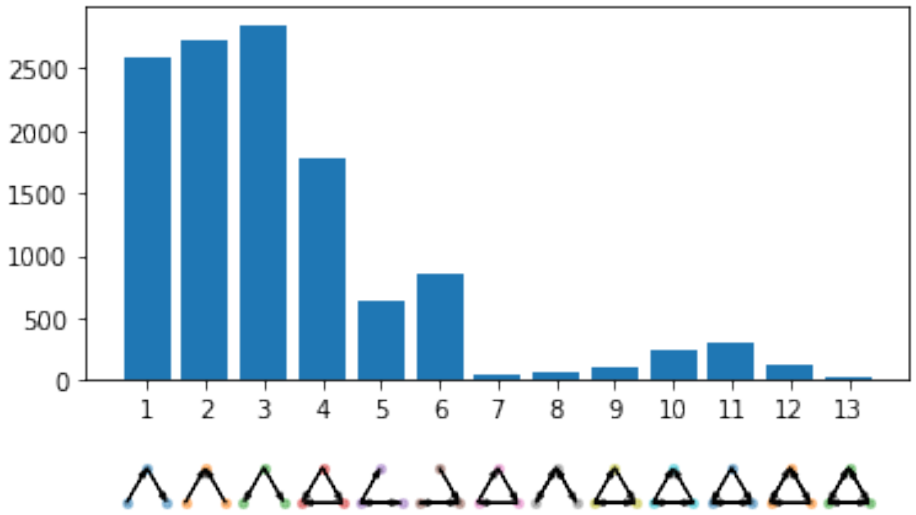


Figure 22: Motifs for eyes open using 'louzoun' algorithm

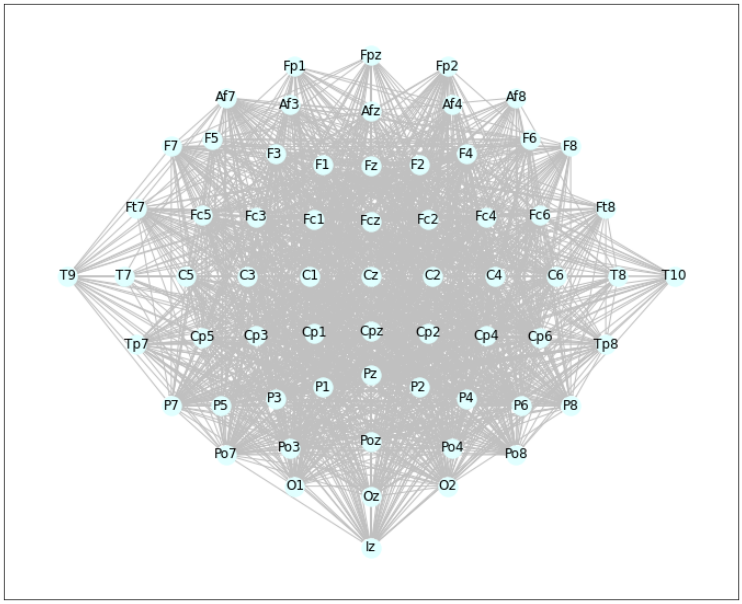


Figure 23: Topographical representation of the closed eyes network, considering only the connections involved in the motif with pattern A $\rightarrow$ B $\leftarrow$ C

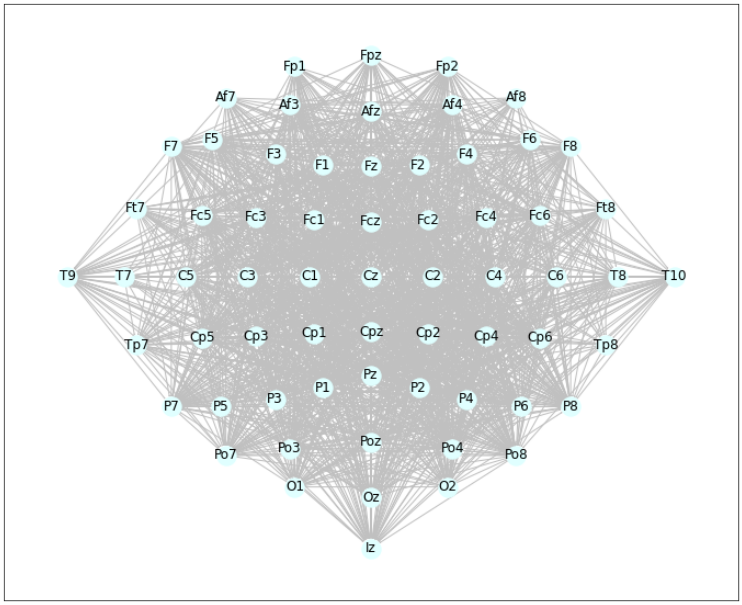


Figure 24: Topographical representation of the open eyes network, considering only the connections involved in the motif with pattern A $\rightarrow$ B $\leftarrow$ C

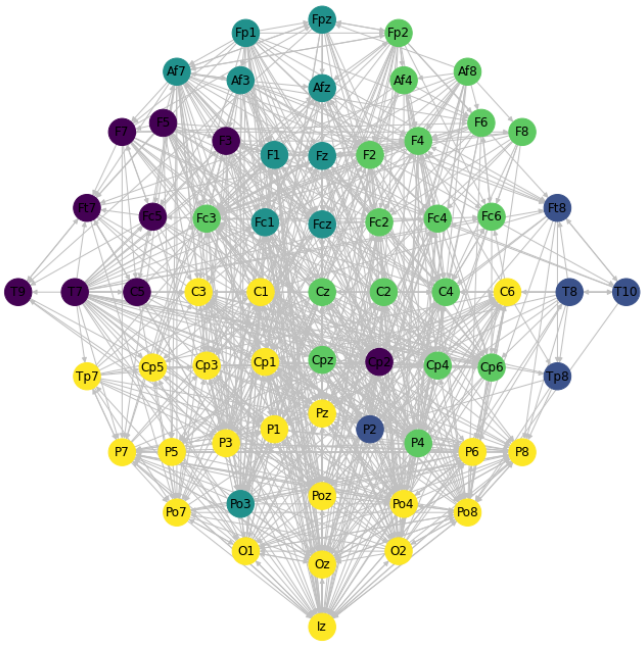


Figure 25: Community detection using Louvain algorithm in the case of eyes closed in a dense graph

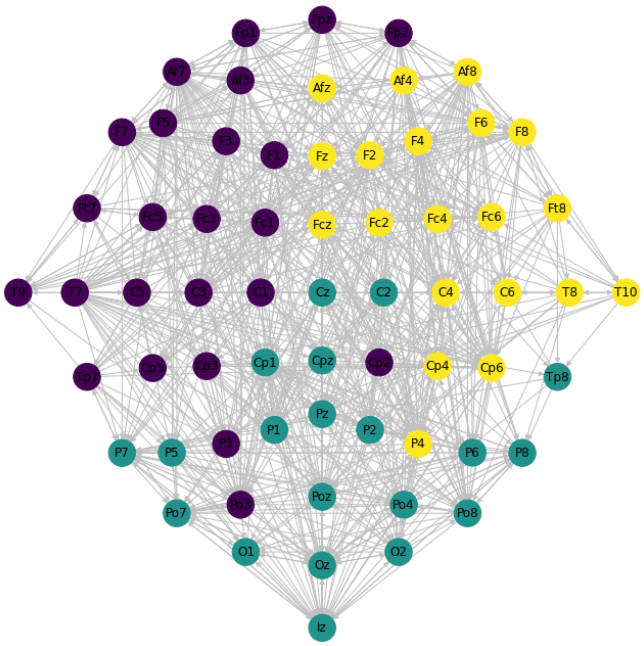


Figure 26: Community detection using Louvain algorithm in the case of eyes open in a dense graph

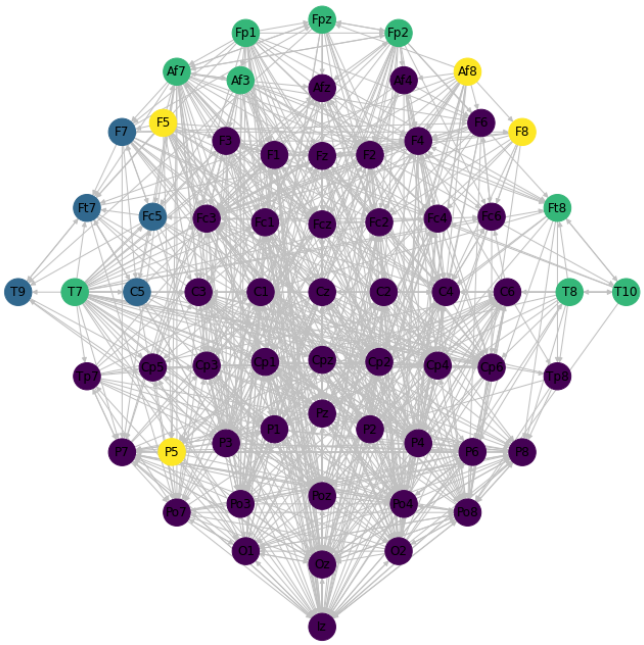


Figure 27: Community detection using Infomap algorithm in the case of eyes closed in a dense graph

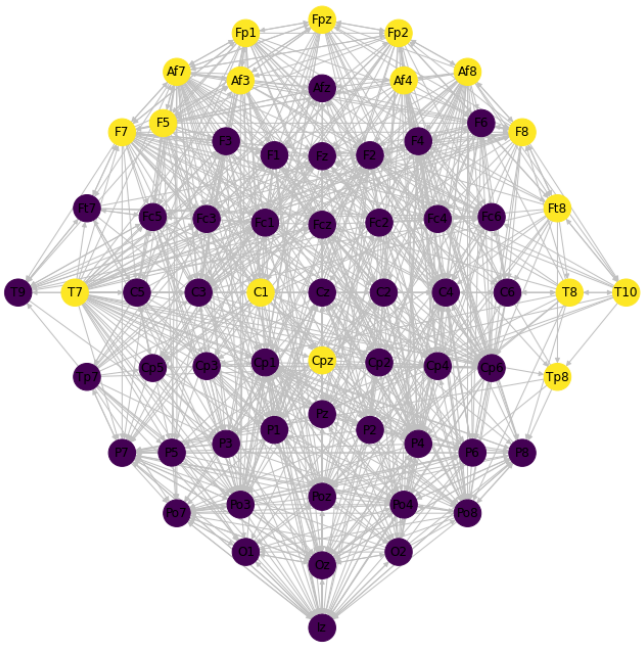


Figure 28: Community detection using Infomap algorithm in the case of eyes open in a dense graph

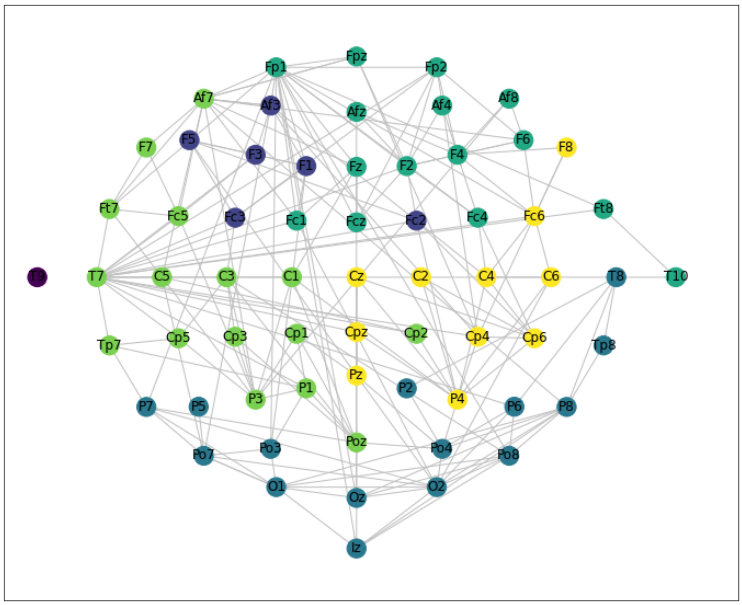


Figure 29: Community detection using Louvain algorithm in the case of eyes closed in a sparse graph

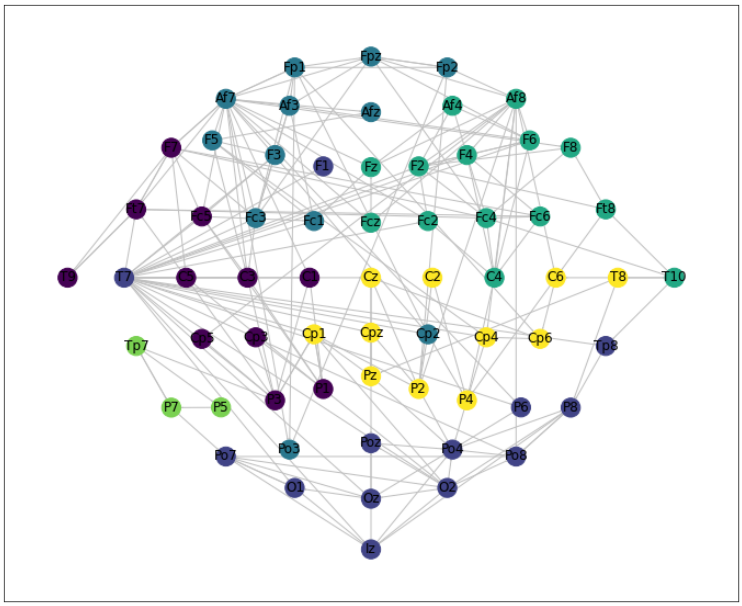


Figure 30: Community detection using Louvain algorithm in the case of eyes open in a sparse graph

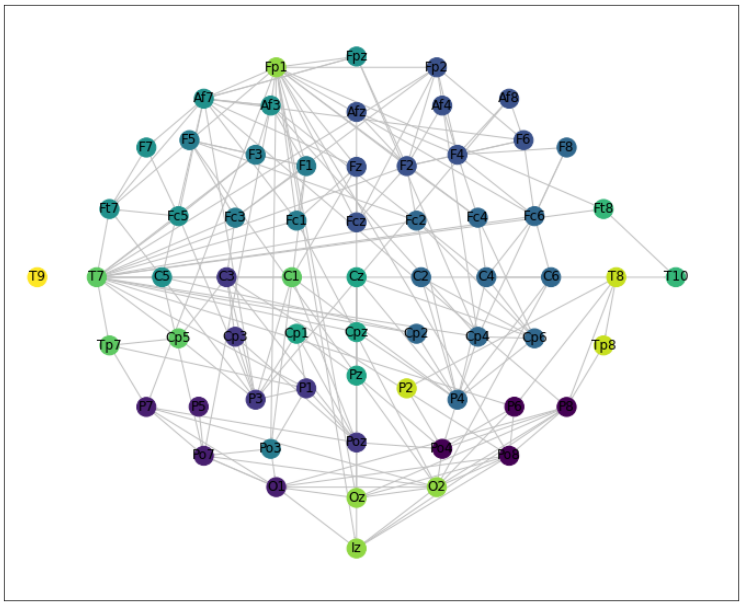


Figure 31: Community detection using Infomap algorithm in the case of eyes closed in a sparse graph

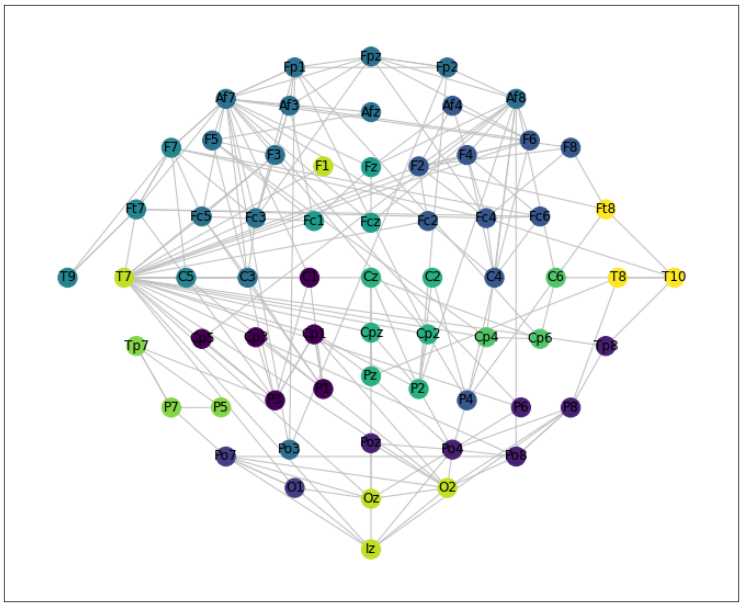


Figure 32: Community detection using Infomap algorithm in the case of eyes open in a sparse graph