Brain network study during resting states

Alessandro TAGLIERI, *Davide* ZINGARO, *Yao* APPETI

University of Rome, MSc Data Science - Bioinformatics class 2020/2021, Group N° 8

Introduction:

The goal this project is to analyze two datasets of EEG data and to prepare a report on their comparison. EEG data are recorded from 64 electrodes with subject at rest in eyes-open (EO) and eyes-closed (EC) conditions, respectively. These input files are in the following web pages: https://physionet.org/content/eegmmidb/1.0.0/ . For this project we considered S038R01.edf and S038R01.edf files.

Analyses will span the following topics:

1. *connectivity graphs*
2. *graph theory indices*
3. *motif analysis*
4. *community detection*

In Table 1 we've defined all task that we've done for the project. We've written also their complexity (as written in the track). To perform all task that we considered we use a Jupyter Notebook where we've written python code using several libraries, like netwrokX[1], iGraph[2], Louviain[3], Infomap[4], smallworld[5], connectivipy[6] and pyedflib[7]. The complete jupyter notebook is in the following GitHub repository:
https://github.com/AlessandroTaglieri/Bioinformatics-Project2

To execute correctly the project, it's necessary to move two input files and *channels.txt* file in a directory named 'data'. This directory must be allocated in the same path where you put the jupyter notebook.

Part 1. Connectivity graph

In this part, we were aimed to estimate functional brain connectivity among 64 channels. In order to do that, we used two types of Multivariate Autoregressive models: Partial Directed Coherence (PDC), Direct Transfer Function (DTF). The Multivariate Autoregressive (MAR) model characterizes interregional dependencies within data, specifically in terms of the historical influence one variable has on another. We used python implementation of Multivariate Autoregressive modes. From that implementation, we used methods that perform estimation using PDC, DTF. We were aimed to obtain a network with a density of 20% applying threshold. In order to find a threshold, we wrote a function that calculates it using the formula of the density of directed graphs. Applying that value of the threshold to the output of estimation gives us an adjacency matrix and we can build the target graph. We performed this procedure for both conditions. As shown in *figure 1* (using DTF) and in *figure 2* (using PDC), in both conditions (EO and EC), graphs are still connected.

In addition, we had a task where we needed to consider subset with 19 electrodes. We needed again to repeat process that was described above but in addition we were required to filter out values that are not significantly different from 0 (PDC (i, j) $\neq$ 0 with $p < 5\%$). In the python implementation there is a method that is written to solve that. As the result we obtained adjacency matrix that can be used for visualizing a graph. *Figure 3* illustrates graphs obtained with 19 channels. In this figure where we are considering first condition (EO), we see that we have 3 disconnected nodes while in the second condition we have 1 disconnected node. They are different in both cases.

Using cartesian coordinates of channels, we made topological representation of the network. We performed it to both variants of the graphs that we had with 64 and 19 channels. Topological representation of 64 channels with density 0.05 is shown in *figure 4* (using PDC) and for 19 channels it is shown in *figure 5* (using DTF). When we are considering 64 channels with density 0.05 in condition when eyes are closed, we see less connections in right-side of central lobe in comparison with condition when eyes are opened. And with opened eyes condition there are more outgoing edges from frontal lobe to parietal lobe in comparison with closed eyes condition. On the contrary, in closed eyes condition we can see that we have more outgoing edges from parietal lobe to frontal lobe than in opened eyes condition. In *figure 4,* we can see that in opened eyes case there are less connections in frontal lobe and on the left side in comparison with closed eyes case.

Part 2. Graph theory indices

In the second part, we were aimed to compute binary global (clustering coefficient, path length) and local (degree, in/out degree) graph indices. In order to perform this task, we started from the DTF graph generated in the previous point. Results of the global graph indices are shown in *Table 2*, in which we compared the values obtained both in the eyes-open (EO) condition and in the eyes-closed (EC) condition. Then, we observed the global graph indices through PDC estimator (*Table 3*). Observing these two tables, we can notice a low average clustering coefficient and a short average path length in both eyes-open and eyes-closed conditions. Results of the local graph indices are shown in *Table 4* and in *Table 5*, in which we summarized the top 10 nodes sorted by highest degree in EO and EC condition, respectively. We computed these values using the DTF estimator. The node that has the greater degree, it's the more important for the whole system. In a directed graph, as in our case, the degree can be split into in-degree and out-degree. Observing the tables related to the local indices, we can notice that the degree distribution when the subject has his eyes open is pretty much similar to the other case.

Then, performing another task, we plot a topographical representation of local indices using cartesian coordinates of EEG channels. *Figure 8* represents in-degree network, *Figure 9* stays for out-degree and *figure 10* about degree indices. We highlighted in light-blue color the 10 highest degree nodes. We can observe that, in both cases, the nodes with high in-degree values are homogeneously located in the different brain lobes, with a slight tendency to the right hemisphere; instead, the highest out-degree nodes are mostly located in the frontal/parietal lobe, with a high tendency to the left hemisphere.

In addition, we tried to compute the small-world organization of our network. A graph has a small-world property if it is characterized by a high clustering coefficient and a short path length. The main mechanism to construct small-world networks is the Watts-Strogatz model. We used this model to compute a small-world network with same number of nodes and same average degree of our directed graph. Small-world properties are found in many real-world phenomena, including biological and neural networks. We started from a regular ring lattice graph with N vertices and k edges per vertex; then, we rewire each edge at random with probability β. We built three graphs with probability β equal to 0, 0.25 and 1, respectively; both for EO and EC subject-condition (*Figure 6* and *Figure 7*). When $\beta=0$, we obtain a completely regular lattice graph, with no rewiring and each node connected to k of its neighbors. If no edge is rewired, we expect that the distances between each pair of nodes would be long. Also, if k is large enough, the ring lattice starts to form many triangles. So, it is characterized by a high clustering coefficient and a long path length. With $\beta=1$, we rewire every single edge; so, we obtain a completely random graph with short path length and small clustering coefficient. With $\beta=0.25$, we obtain a small-world network. Some edges are rewired; so, the distances between nodes are reduced but a high clustering coefficient is maintained. Hence, we can notice that the average clustering coefficient and the shortest path of a small world network depend on the parameters k and β. We tried to quantify network small-worldness by a small-coefficient, σ, calculated by comparing clustering and path length of a given network to an equivalent (directed) random network with same degree on average. If σ coefficient is greater than 1, the network presents small-world properties. Results of the σ coefficient are 3.89 (in the case of Eyes-Opened) and 2.073 (in the case of Eyes-Closed), using the DTF estimator. W can notice that σ coefficient values are greater than 1. We obtained a small-world network in both cases. However, this metric is known to perform poorly because it is influenced by the network size.

Part 3. Motif analysis

In this third part we are aimed to analyse motifs presence in the network. Network motifs are sub-graphs that repeat themselves in a specific network. Each of these sub-graphs, defined by a particular pattern of interactions between vertices, may reflect a framework in which particular functions are achieved efficiently. Indeed, motifs are of notable importance largely because they may reflect functional properties. We considered configurations which contain 3 nodes. There are possible 13 configurations. We implemented our function that counts their frequencies in python. Additionally, we needed to find their significance. In order to do that we needed to consider also random graph with the same number of nodes and density. We performed Monte Carlo simulations to get mean and standard deviation of frequencies of configurations in random graphs. We needed these two values in order to count Z values of configurations. Larger value of Z means that its significance is larger. Results of all frequencies and Z values for each configuration are shown in *Table 7*. Among 13 configurations seven of them have high Z-value in condition where eyes are opened. Especially, Z-value of configuration when 3 nodes construct a triangle where they go in both directions (A->B->C->A->C->B->A), it has really small frequency in both cases. Configuration ‘A->B<-C’ has the highest frequency and quite high Z-value in both cases.

Configurations that have Z-value less than -0.9 can be considered as anti-motifs. From first condition we have 3 anti-motifs. Two of them are also anti-motifs for second configuration however one of them is not the same and it is configuration that was not nor motif or anti-motif in first condition.

We performed also topological representation of network considering only configuration 'A->B<-C'. In order to perform this task, we wrote a procedure that takes in consideration only this configuration. Since it has the highest frequency it covers all the nodes. In order to see the difference views, we used graphs with density 0.05. Graphs are represented in *Figure 11*. When eyes are closed there are few edges in nodes in right central lobe.

In parieto-occipital scalp region we selected one channel to consider and show in which motifs they are involved. This is 'Po8'. We have four configurations in eyes opened and eight in case when eyes are closed. Motifs related to node 'Po8' are shown in *figures 12* and *figures 13*.

Part 4. Community detection

Over the years, many algorithms are developed for understanding brain connectivity and information flows using graph theory. However, we will see here how Louvain and Infomap algorithms behave with two analyzed datasets containing EEG data.
The main research question regarding this part is making inference by seeing how different parts of brains can be grouped by using different techniques that are information-based (Infomap algorithm) or based on the modularity of the network (Louvain). Clustering algorithms has the goal to capture the intuitive notion that nodes should be connected to many nodes in the same community (intra-cluster density) but connected to few nodes in other communities (inter-cluster sparsity).

Figures 14, 15 show community structures as mentioned in the previous sections, for two different conditions: eyes-closed (EC), and eyes-opened (EO). Colored vertices correspond to the identified communities and their location according to the channel labels. Tables 8 and 9 show the composition of the communities obtained by Louvain algorithm. By observing the results of the modularity-based algorithm, we see that communities in the results of the Louvain algorithm in both states look quite differentiable. Moreover, the infomap algorithm created one community in the EC state. While, in the EO state, a lot of nodes from frontal, right temporal, parietal and occipital parts are labeled as part of the bigger community(region) of the brain. While comparing the execution times of the algorithms, Louvain was the most efficient.

References:

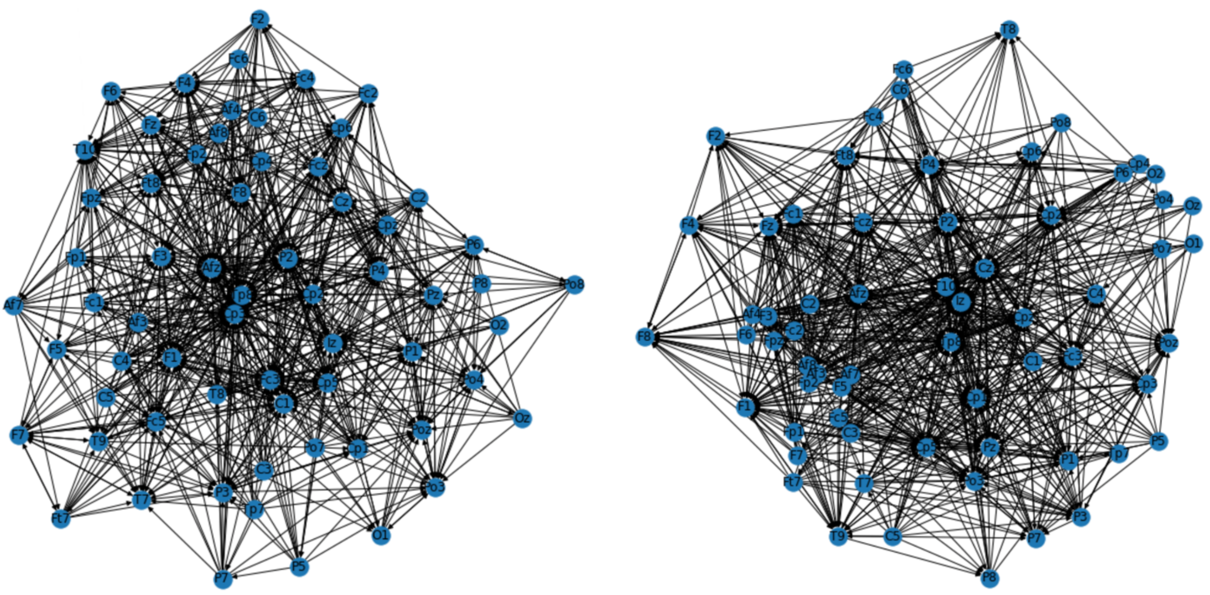
1 – https://networkx.org Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks:
2 – https://igraph.org Collection of network analysis tools with the emphasis on efficiency, portability and ease of use
3 –https://github.com/taynaud/python-louvain : This module implements community detection with Louvian method
4 –https://mapequation.github.io/infomap/python/ : This module implements community detection with INFOMAP method
5 - https://github.com/benmaier/smallworld Functions for estimating the small-world-ness of graphs
6 - https://github.com/dokato/connectivipy **:** module for connectivity analysis
7 -https://pyedflib.readthedocs.io/en/latest/ : python library to read/write EDF+/BDF+ files based on EDFlib.

Figures and Tables

Table 1. The list of task performed in this project

Task	Class
1.1	Mandatory
1.2	A
1.4	D
1.5	C
2.1	Mandatory
2.2	D
2.3	B
2.5	B
3.1	Mandatory
3.2	C
3.3	C
4.1	Mandatory
4.2	B
4.3	C

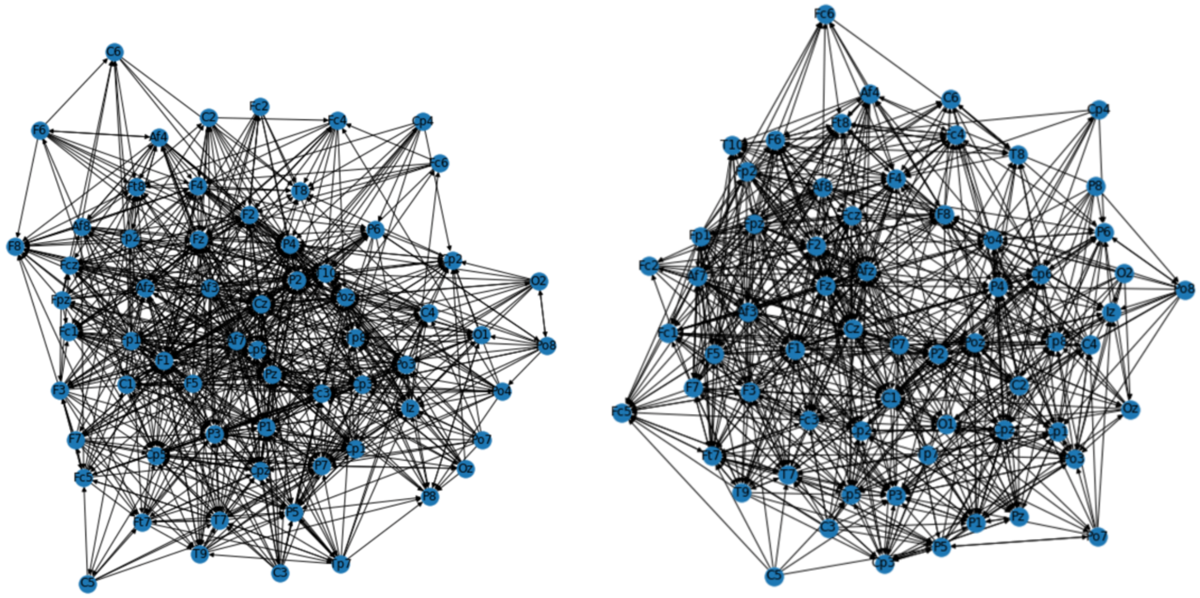
Figure 1. Network with a density 20% applying threshold: using DTF estimator



Eyes-Opened

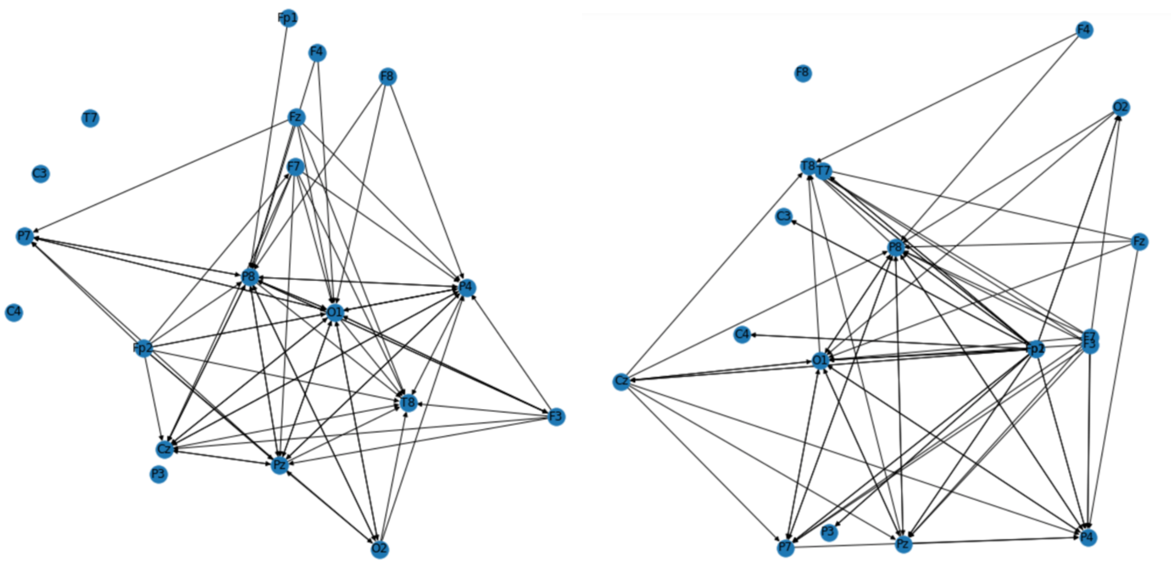
Eyes-Closed

Figure 2. Network with a density 20% applying threshold: using PDC estimator



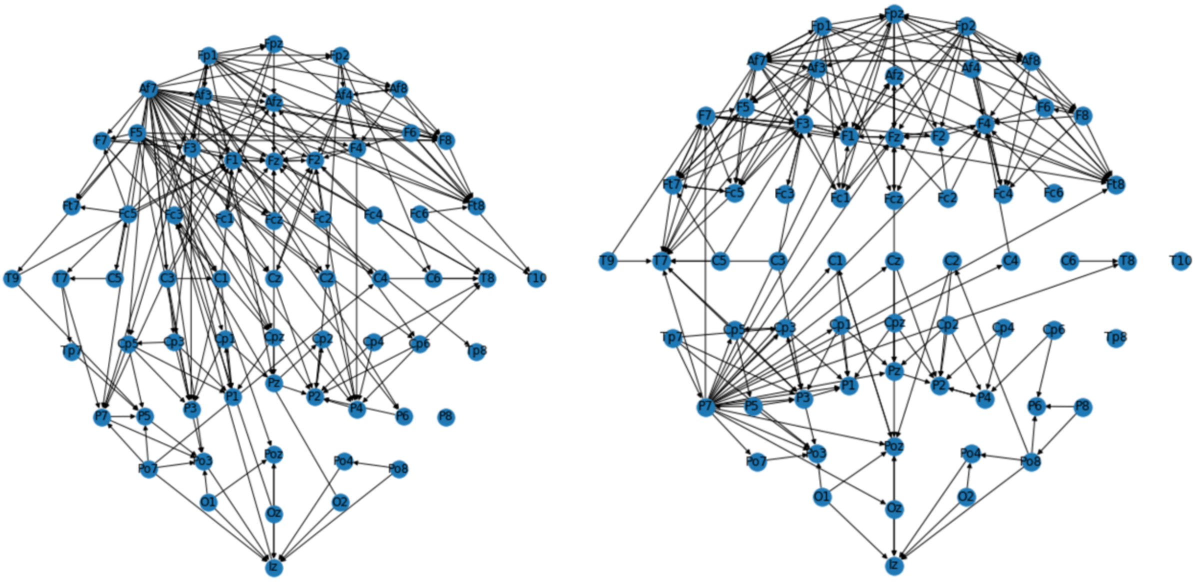
Eyes-Opened Eyes-Closed

Figure 3. DTF of 19 suggested channels



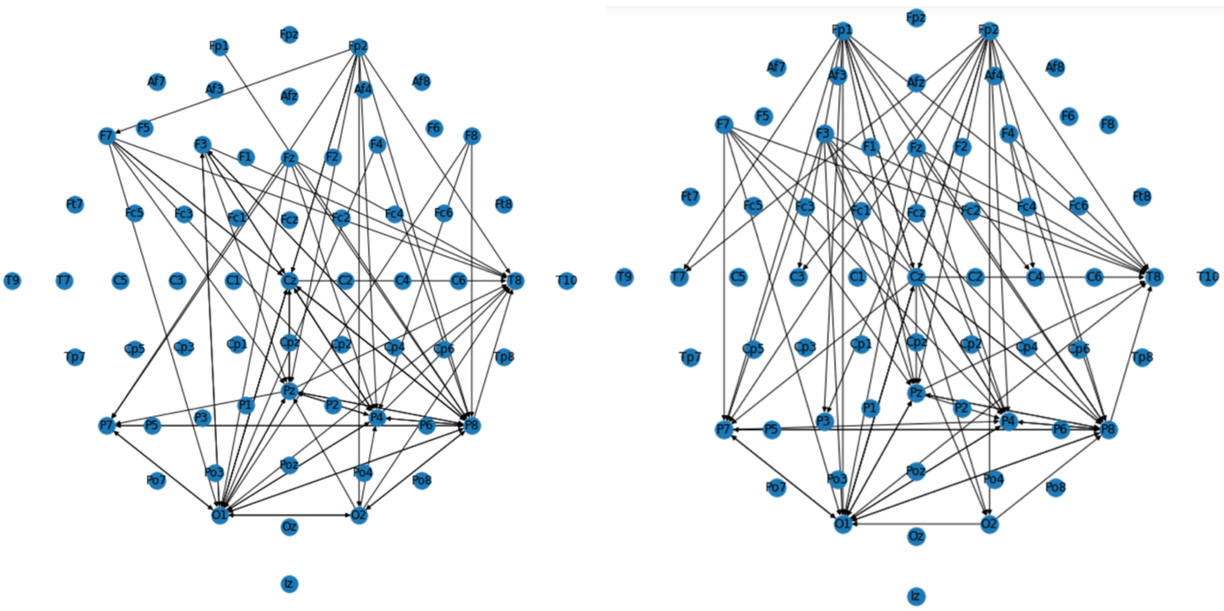
Eyes-Opened Eyes-Closed

Figure 4. Topographical representation of the PDC networks with density 0.05 using cartesian coordinates



Eyes-Opened Eyes-Closed

Figure 5. Topographical representation of the DTF network with density 0.02 (< 5%) for 19 channels using cartesian coordinates



Eyes-Opened Eyes-Closed

Table 2. Binary global graph indices - TDF

Condition	Index	TDF
Eyes-open	Average clustering coefficient	0.409
Eyes-open	Average path length	0.843
Eyes-close	Average clustering coefficient	0.41
Eyes-close	Average path length	1.556

Table 3. Binary global graph indices - PDC

Condition	Index	PDC
Eyes-open	Average clustering coefficient	0.326
Eyes-open	Average path length	2.673
Eyes-close	Average clustering coefficient	0.36
Eyes-close	Average path length	1.987

Table 4. Local indices DTF – eyes opened

Node	In - Degree	Node	Out - Degree	Node	Degree
T10	60	Cp5	21	Iz	68
Iz	60	Afz	20	Cz	61
Tp8	56	Cp3	19	Tp8	61
Cz	47	Af7	19	T10	60
P2	38	Fc5	18	Cp5	49
Cp2	33	Fc3	18	Cp1	49
Cp1	32	Fp1	18	Afz	49
Fz	30	Fp2	18	Fc3	47
Fc3	29	Af3	18	P2	46
Afz	2	F5	18	Cp2	44

Table 5. Local indices DTF – eyes closed

Node	In - Degree	Node	Out - Degree	Node	Degree
Afz	53	Cp3	19	Afz	71
Cp3	49	Fpz	19	Cp3	68
P2	48	Af3	19	P2	62
Tp8	46	Pz	19	Tp8	52
Fc3	38	Fc5	18	Fc3	51
C1	32	Fp1	18	C1	46
F1	31	Afz	18	Iz	44
Iz	28	P1	17	Cp2	40
T10	26	Cp5	16	F1	39
Cp2	25	Af7	16	F3	37

Figure 6. DTF Small-World network (EO)

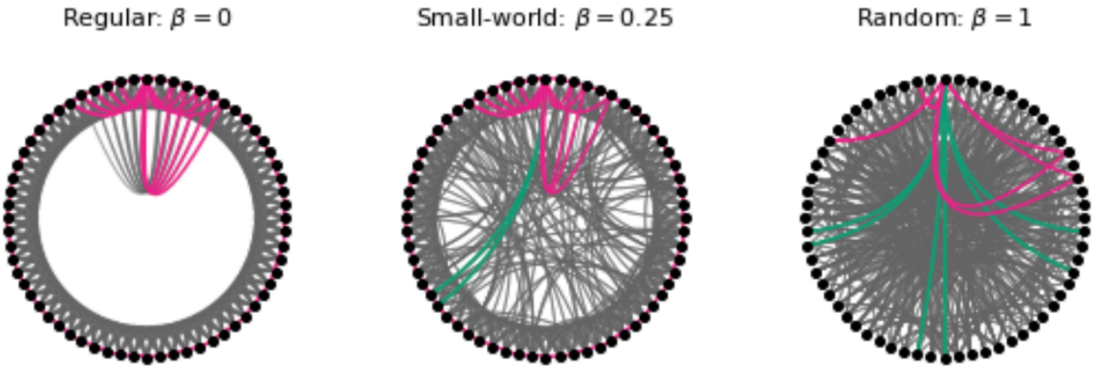


Figure 7. DTF Small-World network (EC)

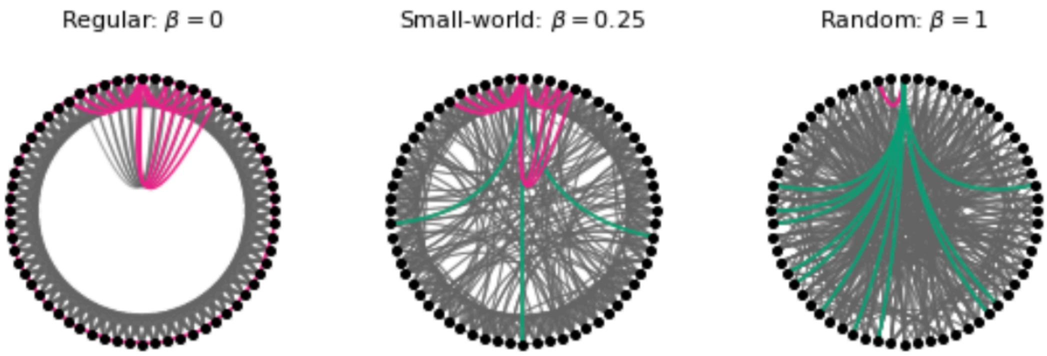
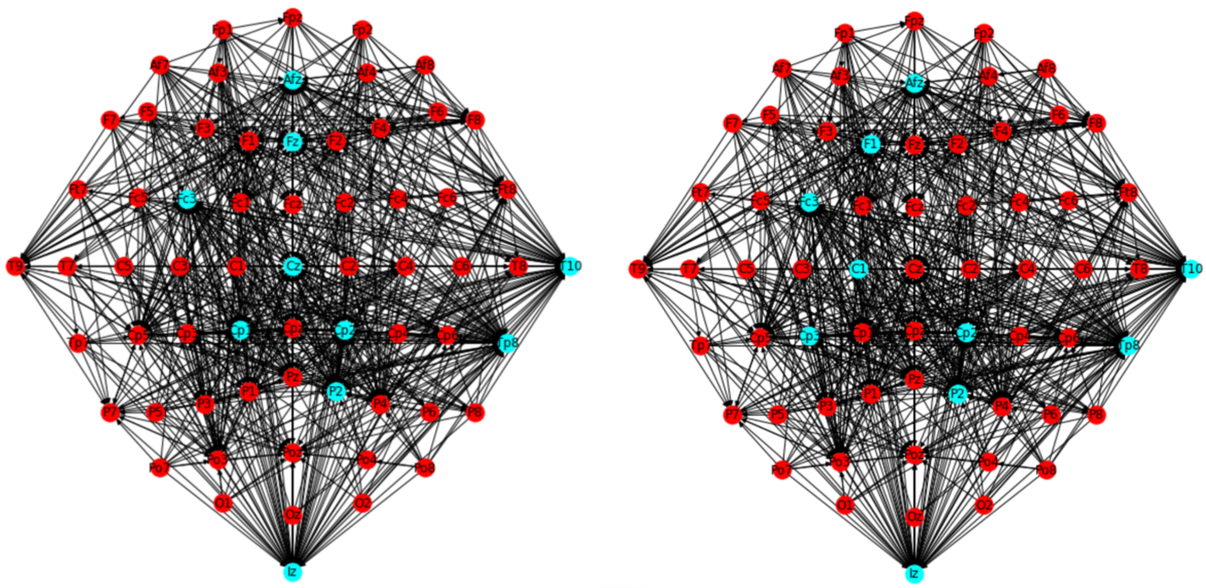
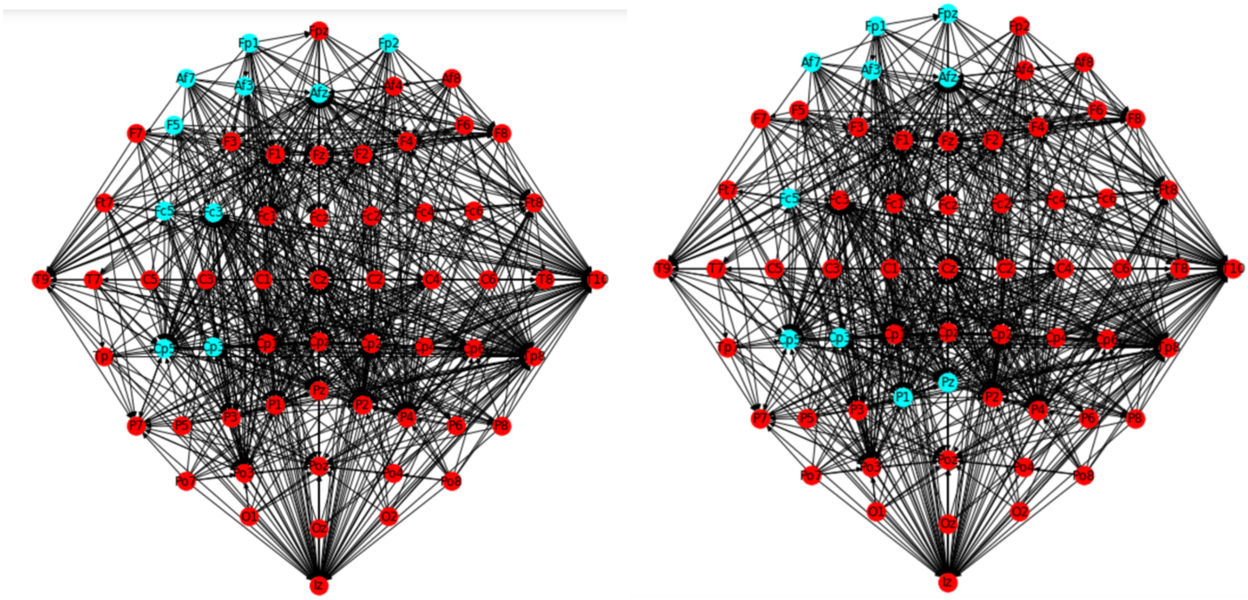


Figure 8. DTF In-Degree



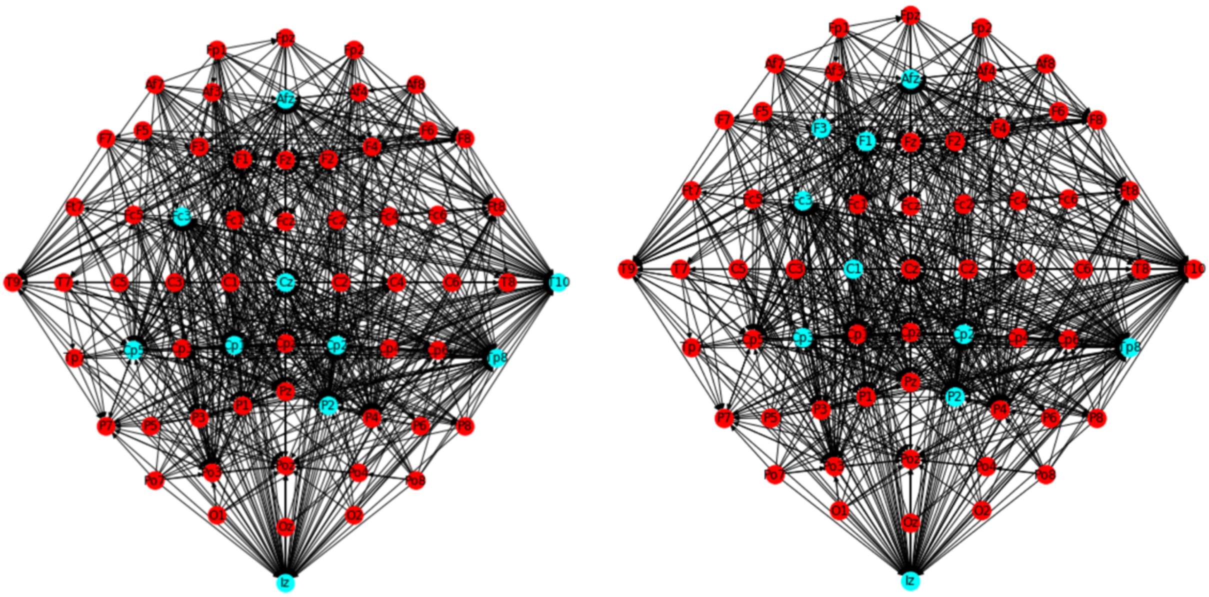
Eyes-Opened Eyes-Closed

Figure 9. DTF Out-Degree



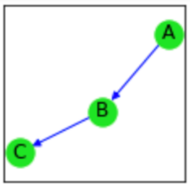
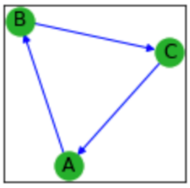
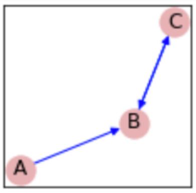
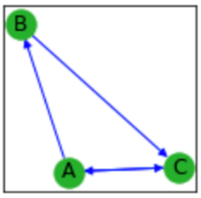
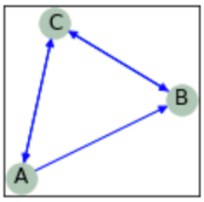
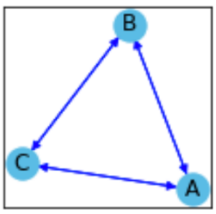
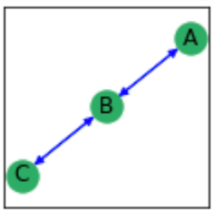
Eyes-Opened Eyes-Closed

Figure 10. DTF Degree



Eyes-Opened Eyes-Closed

Table 6. Frequencies and Z-values of motifs in network

CONFIGURATION	Graphic representation	Frequency (EO)	Z (EO)	Frequency (EC)	Z (EC)
A->B->C		6536	-3.178391	7068	-2.679838
A->B->C->A		376	-3.653741	651	-0.114797
A->B->C->B		2631	5.559106	3356	9.382549
A->B->C->A->C		371	1.045722	616	6.461059
A->B->C->A->C->B		3883	-0.455113	7259	4.110619
A->B->C->A->C->B->A		58	29.151113	72	39.942987
A->B->C->B->A		355	3.907596	578	9.181598

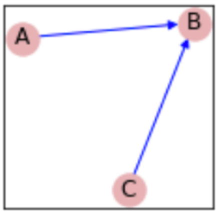
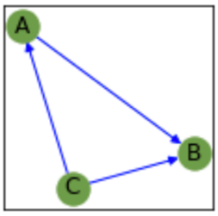
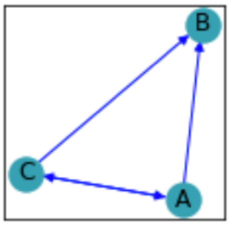
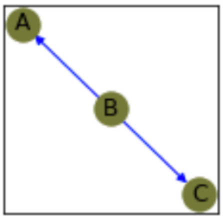
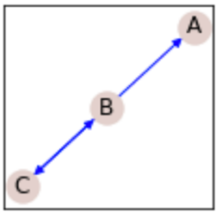
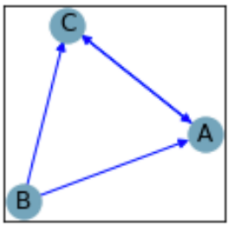
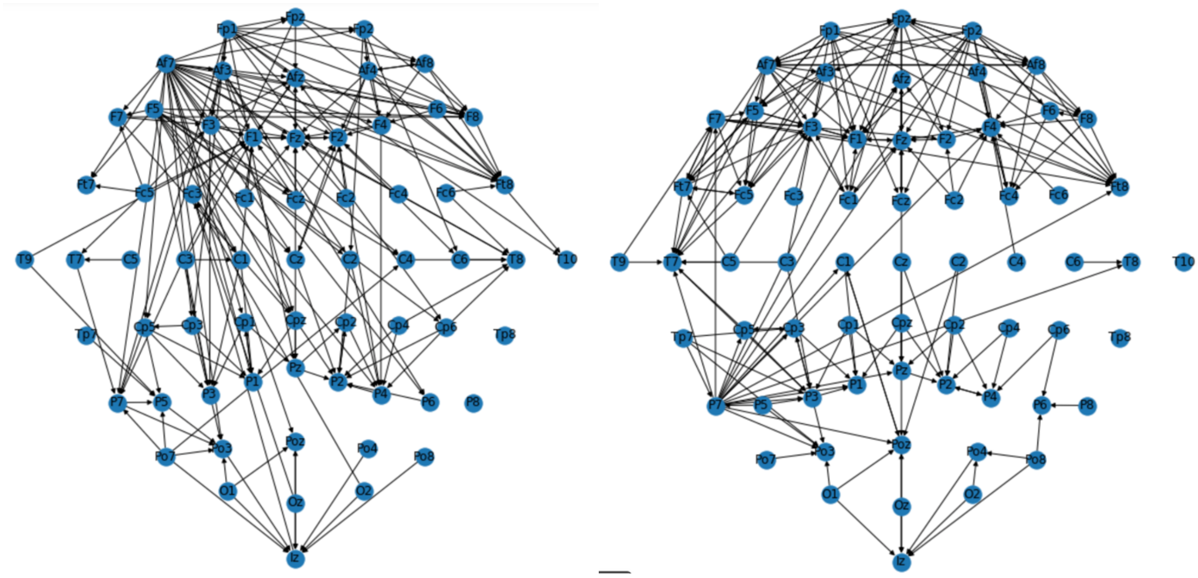
A->B<-C		11879	20.547587	9278	16.090764
A->B<-C->A		3660	11.629491	3123	11.182618
A->B<-C->A->C		516	9.463630	649	14.788225
A<-B->C		4812	0.003570	4475	-1.122909
A<-B->C->B		900	-3.601386	1367	-1.077191
A<-B->C->A->C		1040	26.995669	1191	34.00327

Figure 11. Topographical representation of the networks involved in A -> B <- C



Eyes-Opened Eyes-Closed

Figure 12. Paretio-Occipital - EO

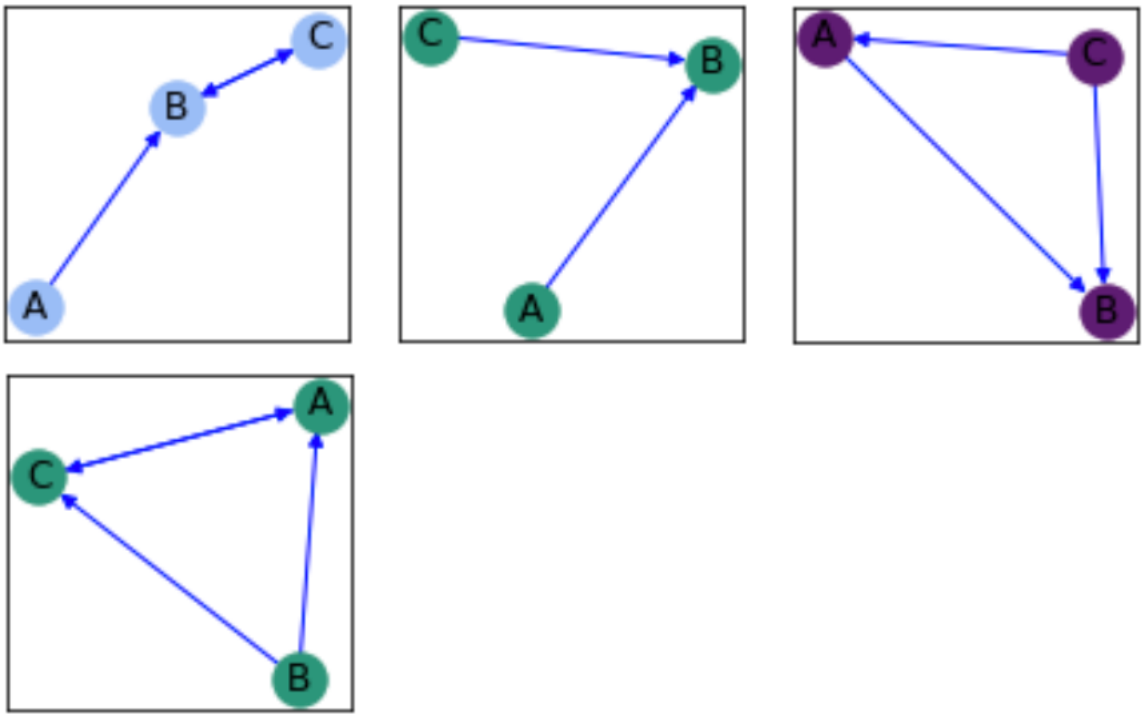


Figure 13. Paretio-Occipital - EC

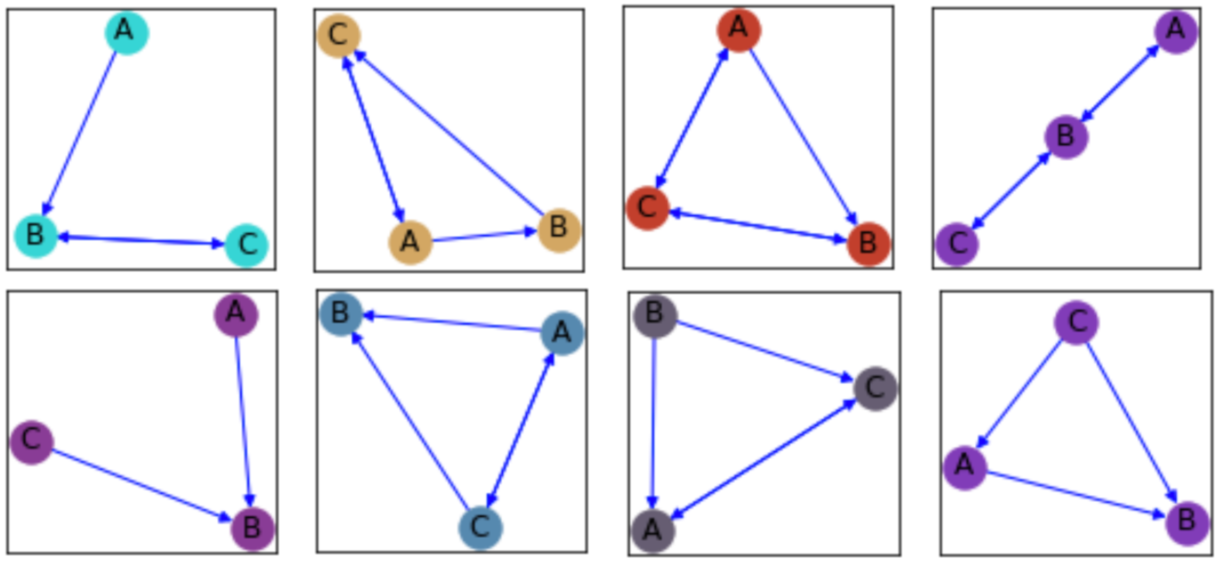


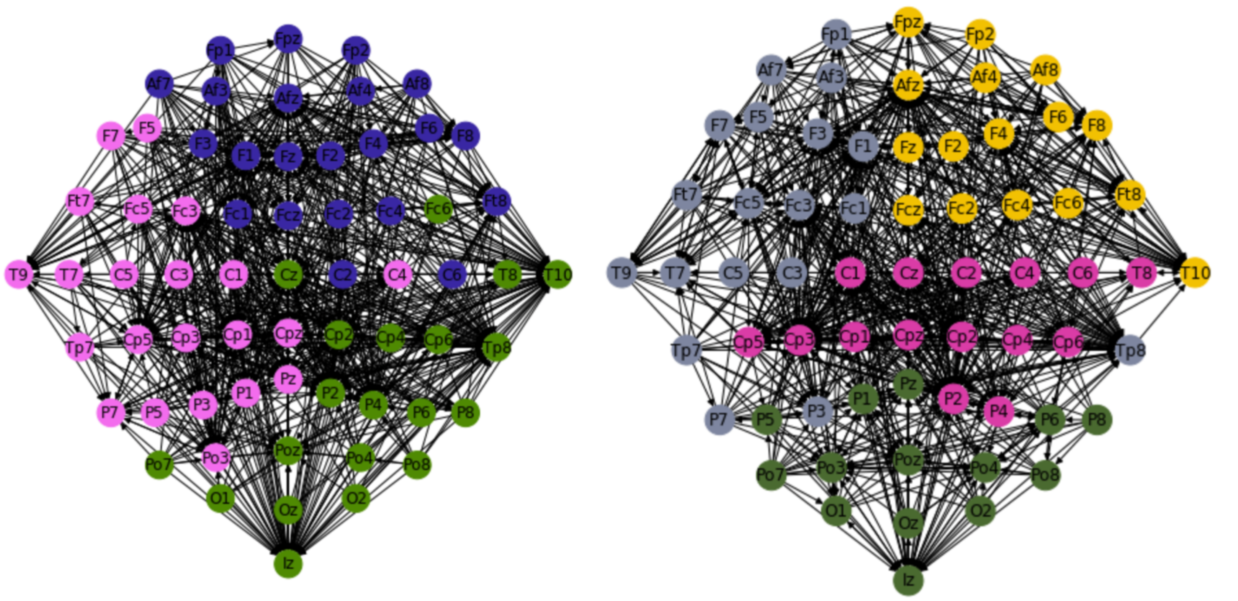
Table 7. Composition of communities using Louvian algorithm (EO)

Size	Composition of communities
22	'Fc1', 'Fcz', 'Fc2', 'Fc4', 'C2', 'C6', 'Fp1', 'Fpz', 'Fp2', 'Af7', 'Af3', 'Afz', 'Af4', 'Af8', 'F3', 'F1', 'Fz', 'F2', 'F4', 'F6', 'F8', 'Ft8'
22	'Fc5', 'Fc3', 'C5', 'C3', 'C1', 'C4', 'Cp5', 'Cp3', 'Cp1', 'Cpz', 'F7', 'F5', 'Ft7', 'T7', 'T9', 'Tp7', 'P 7', 'P5', 'P3', 'P1', 'Pz', 'Po3'
20	'Fc6', 'Cz', 'Cp2', 'Cp4', 'Cp6', 'T8', 'T10', 'Tp8', 'P2', 'P4', 'P6', 'P8', 'Po7', 'Poz', 'Po4', 'Po8', 'O1', 'Oz', 'O2', 'Iz'

Table 8. Composition of communities using Louvian algorithm (EC)

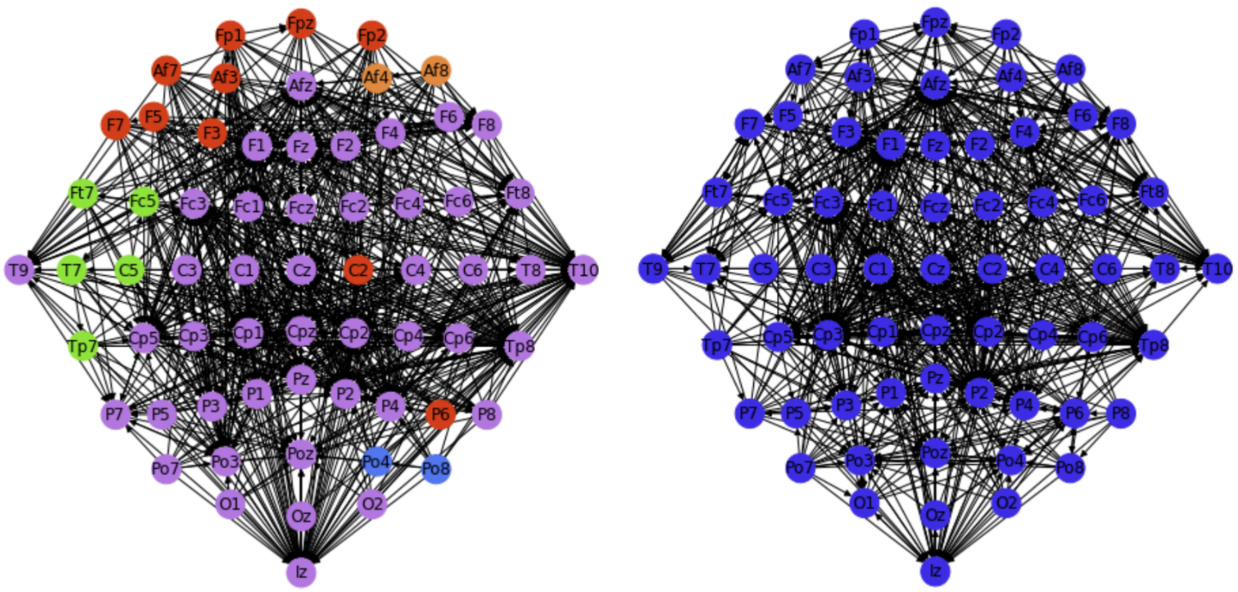
Size	Composition of communities
19	'Fc5', 'Fc3', 'Fc1', 'C5', 'C3', 'Fp1', 'Af7', 'Af3', 'F7', 'F5', 'F3', 'F1', 'Ft7', 'T7', 'T9', 'Tp7', 'Tp8', 'P7', 'P3'
16	'Fcz', 'Fc2', 'Fc4', 'Fc6', 'Fpz', 'Fp2', 'Afz', 'Af4', 'Af8', 'Fz', 'F2', 'F4', 'F6', 'F8', 'Ft8', 'T10'
15	'C1', 'Cz', 'C2', 'C4', 'C6', 'Cp5', 'Cp3', 'Cp1', 'Cpz', 'Cp2', 'Cp4', 'Cp6', 'T8', 'P2', 'P4'
14	'P5', 'P1', 'Pz', 'P6', 'P8', 'Po7', 'Po3', 'Poz', 'Po4', 'Po8', 'O1', 'Oz', 'O2', 'Iz'

Figure 14. Communities detected by Louvian algorithm



Eyes-Opened Eyes-Closed

Figure 15. Communities detected by InfoMap algorithm



Eyes-Opened Eyes-Closed