Brain Network Study During Resting States

Mattia Podio, Konstantinos Sioumalas Christodolou, and Mattias Basso

(Dated: 16 January 2019)

Keywords: EEG data, brain network, connectivity graph, graph indices, motif analysis, community detection

The goal of this project is to conduct some analysis on two datasets of EEG data. In this report we explain briefly what we did and we try to get some insight from the results of the analysis. For a complete list of the tasks completed, see Appendix A.

Data[1] are taken from a public repository published from the Physionet Project. This data set consists of over 1500 one- and two-minute EEG recordings, obtained from 109 volunteers, but in our tasks we used just two runs from a specific subject (`S001`): the first run - `R01` - is recorded during eyes-open (EO) resting state, while the second one - `R02` - is recorded during eyes-closed (EC) resting state.

Generally we worked in Python, so in order to read EEG data, that are stored in `EDF` format we used the `pyedflib` library[2]

I. CONNECTIVITY GRAPHS

In the last decades, a big effort has been made to apply graph theory to neuroscience. For that purpose we used some basic but important tools of graph theory to identify the brain connectivity of the previous resting states.

Graph theory provides powerful means to assess topology and organization of brain connectivity networks

A network based approach is especially useful when it's important to comprehend and highlight non obvious interactions between agents in a system, their nature depending on the case study. In computational neuroscience we measure the neuronal activity of different brain regions through sensors applied on top of the scalp. Functional connectivity can be estimated using methods based on Granger causality, such as multivariate autoregressive models (MVAR).

A multivariate autoregressive model of order p can be defined as

$$X(t) = \sum_{r=1}^{p} A(r)X(t-r) + E(t) \tag{1}$$

In our case, $X(t)$ is a vector that contains measurements from 64 signals at time t; $A(r)\ r \in \{1, 2, ..p\}$ are matrices in $\mathbb{R}^{N\times N}$ that contain the parameters (to be estimated) that describe the dependence between the entries in $X(t)$ and $X(t-r)$; $E(t)$ is a vector of random variables and denotes the uncorrelated Gaussian process with zero mean.

The formula says that values at time t linearly depend from the last p values taken from our 64 time series through an appropriate set of $N \cdot N \cdot p$ coefficients.

It is possible to select the best value of p with respect to the Akaike information criterion. This selection required us to spend to much time on that, and since we considered this not crucial for our purposes, we set the order of the model to $p = 3$. Further investigations can surely be made in future studies.[3]

What is written above can be rewritten also like.

$$\sum_{r=0}^{p} A(r)X(t-r) = E(t) \tag{2}$$

if $A(0) = \mathbb{I}$, and $A(r)\ r \in \{1, 2, 3\}$ are estimated by solving the Yule Walker equations.

EEG signals are both defined in time as well as in frequency. We are interested in the reconstruction of the direction of information flows in the network.

Applying the Fourier Transform to both sides, hence passing to the frequency domain we have these two results

$$A(f)X(f) = E(f) \tag{3}$$

$$X(f) = A^{-1}(f)E(f) = H(f)E(f) \tag{4}$$

Partial Directed Coherence - PDC - is defined as

$$\pi_{ij}(f) = \frac{|A_{ij}(f)|^2}{\sum_{m=1}^{N} |A_{mi}(f)|^2} \tag{5}$$

Directed Fourier Transform - DTF - is defined as

$$\theta_{ij}(f) = \frac{|H_{ij}(f)|^2}{\sum_{m=1}^{N} |H_{mi}(f)|^2} \tag{6}$$

Once we calculated these estimators, choosing a particular frequency f^* we can effectively start using a graph formalism: in fact what we did is to construct a graph from using either $\pi(f^*)$ or $\theta(f^*)$.

We conduct the analysis below selecting the topology that arises choosing $f^* = 10$ Hz; they basically corresponds to the alpha waves ($8 - 12.5$ Hz), that predominantly originate from the occipital lobe during wakeful

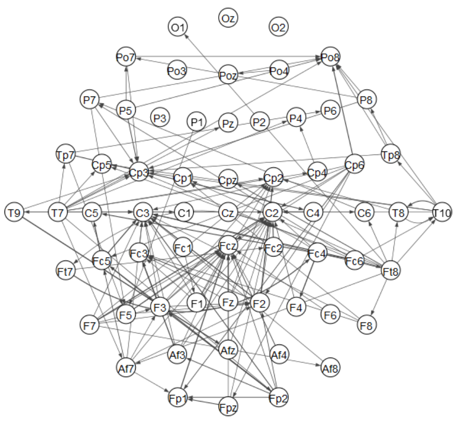


FIG. 1. Topographical representation of directed connectivity network for **open eyes** with density d = 0.05 (top view, nose down)

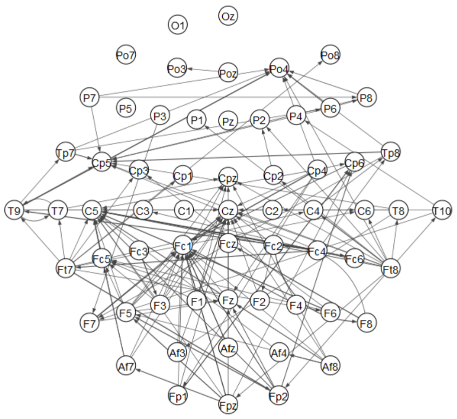
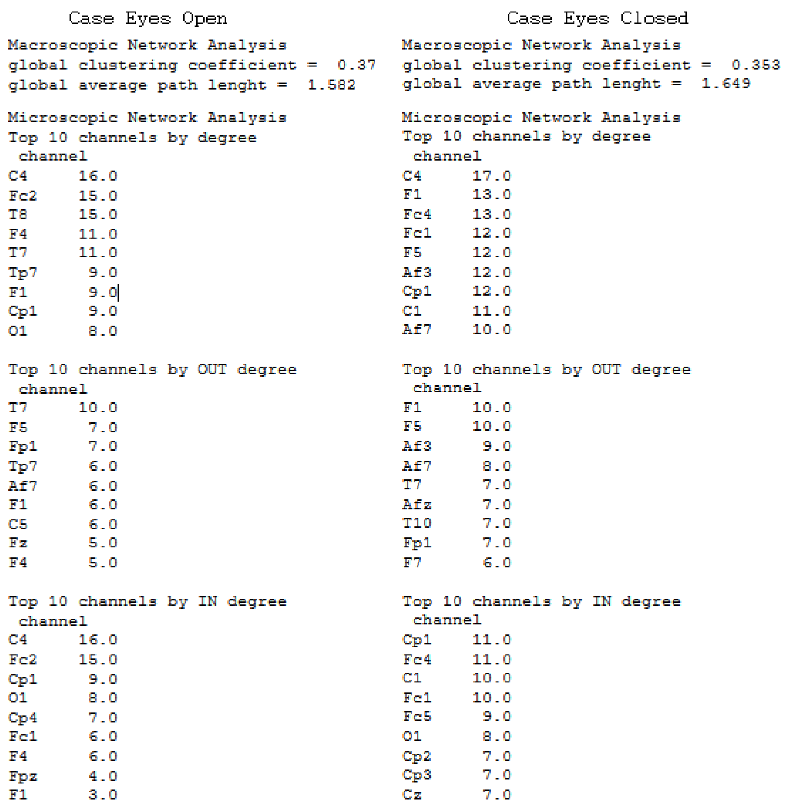


FIG. 2. Topographical representation of directed connectivity network for **closed eyes** with density d = 0.05 (top view, nose down)

relaxation with closed eyes. Alpha waves are reduced with open eyes, drowsiness and sleep[6].

We also generate the graphs relative to $f^* = 28$ Hz, i.e. to high beta waves (20.5–28 Hz); beta states are the states associated with normal waking consciousness[7].

When creating a graph object, we put a threshold on network density. Through a custom function we iteratively select the less meaningful edge - i.e. edge with the lowest weight - and we delete it, until criteria on the density is satisfied. In case of multiple edges with equal lowest weight, what the function does is to select one of them at random. Unless otherwise stated, we will refer to graphs with density d = 0.05.



```
      Case Eyes Open                                   Case Eyes Closed
Macroscopic Network Analysis                   Macroscopic Network Analysis
global clustering coefficient =  0.37          global clustering coefficient =  0.353
global average path lenght =  1.582            global average path lenght =  1.649

Microscopic Network Analysis                   Microscopic Network Analysis
Top 10 channels by degree                      Top 10 channels by degree
 channel                                        channel
C4     16.0                                    C4     17.0
Fc2    15.0                                    F1     13.0
T8     15.0                                    Fc4    13.0
F4     11.0                                    Fc1    12.0
T7     11.0                                    F5     12.0
Tp7     9.0                                    Af3    12.0
F1      9.0                                    Cp1    12.0
Cp1     9.0                                    C1     11.0
O1      8.0                                    Af7    10.0

Top 10 channels by OUT degree                  Top 10 channels by OUT degree
 channel                                        channel
T7     10.0                                    F1     10.0
F5      7.0                                    F5     10.0
Fp1     7.0                                    Af3     9.0
Tp7     6.0                                    Af7     8.0
Af7     6.0                                    T7      7.0
F1      6.0                                    Afz     7.0
C5      6.0                                    T10     7.0
Fz      5.0                                    Fp1     7.0
F4      5.0                                    F7      6.0

Top 10 channels by IN degree                   Top 10 channels by IN degree
 channel                                        channel
C4     16.0                                    Cp1    11.0
Fc2    15.0                                    Fc4    11.0
Cp1     9.0                                    C1     10.0
O1      8.0                                    Fc1    10.0
Cp4     7.0                                    Fc5     9.0
Fc1     6.0                                    O1      8.0
F4      6.0                                    Cp2     7.0
Fpz     4.0                                    Cp3     7.0
F1      3.0                                    Cz      7.0
```

FIG. 3. Summary of binary global and local graph indices (top ten channels are listed) for both case studies.

II. GRAPH THEORY INDICES

A. Global and Local Indices

We calculated basic indexes - both global and local - regarding the graphs generated with the techniques written above. In Figure 3 it is listed a summary that refers to binary networks. We also computed their respective weighted analogous that it is not shown in this report (for details see the attached material: `main.html`).

B. Small World Index

Duncan Watts and Steven Strogatz[8] found that graphs can be classified according to two important global indexes:

- clustering coefficient
- average shortest path

Empirical evidence shows that many real world systems appear to be in the middle, in the sense that they share with regular networks the fact that they have an high clustering coefficient (not true for random networks), but on the other hand on these graphs, it can

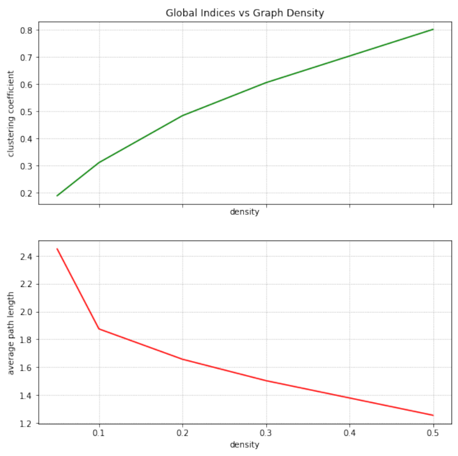


FIG. 4. With increasing density, we shall see that the global clustering coefficient is positively related to the density of the graph. This is expected, since the clustering coefficient is a measure of the the degree to which nodes in the graph generally tend to cluster together: when the density increases the average degree of the graph increases and as a result, it is more likely that neighbours of any given node are neighbours with each other.
At the same time, the more dense is a network, the more likely it is to find a shorter path that connects any two given nodes, and this gives the intuition why to higher densities correspond lower average shortest paths.

take only a few steps to reach, from one source, whatever target - the latter being a characteristic of random networks and not of lattices.

Watts and Strogatz measured that in fact many real-world networks have a small average shortest path length but also a clustering coefficient significantly higher than expected by random chance.

Small world networks can be considered as a trade off between two limiting topologies[10]: the regular lattice, and the random graph.

We consider a graph to be a *small world network* when

- for any given node, neighbours of that node are likely to be connected to each other (read: high clustering coefficient)
- the average distance L between two nodes (average shortest path) scales as the logarithm of the number of nodes in the network: $L \sim \log N$

Several indexes have been proposed to measure how much a given graph satisfies these requirements[9] and we will focus our attention on the small world index - *SWI*. It is defined[11] as follows:

$$\mathrm{SWI} = \frac{L - L_l}{L_r - L_l} \times \frac{C - C_r}{C_l - C_r} \tag{7}$$

where C and L denote the clustering coefficient and average path length respectively of the actual network; C_r and L_r refer to an equivalent random network with same degree distribution, C_l and L_l refer to an equivalent lattice network with same degree distribution.

Like Telesford et al[12], we created equivalent random graphs by assigning an edge to a node pair with uniform probability while maintaining the degree distribution of the original graph[15], as well as using a modified version of the "latticization" algorithm[16] found in the Brain Connectivity Toolbox[13] - we actually used `bctpy`[14], the Python version of the Brain Connectivity Toolbox (natively in 'MATLAB').

Regarding the estimation of L_r and C_r, since by definition they are obtained after the generation of a random network, we performed a bootstrap procedure so as to get their average value. Unfortunately, it has not been possible to sample the randomized network a lot of times because the implementation of the algorithm was not optimal time-wise.

We found

$$\mathrm{SWI} = -0.257 \quad \text{for the EO case}$$

$$\mathrm{SWI} = 0.665 \quad \text{for the EC case}$$

We actually doubt these results because of the aforementioned low number of simulations, and for the negative index we received as an output in the EO condition.

III. MOTIF ANALYSIS

Roughly speaking, motifs in a given network are induced sub-graphs which are recurrent and statistically significant: in other words, they appear with a higher frequency than expected in 'similar' random networks.

Let G be a network, and G_k one of its own sub-networks composed by k nodes. If we count how many times G_k appears in G for some k, we obtain something really analogous to a frequency spectrum, that can give a description about the basic building blocks of the net.

We can immediately guess that the action of counting how many times a given G_k occurs in G is highly computationally expensive (note also that the cardinality of the set $\{G_k \text{ for k} \in \mathbb{N}\}$ grows exponentially with k). That's why one usually concentrate just on sub-graphs of size $k = 3, 4$.

To be more formal, we consider an induced sub-graph G_k of G to be a **motif** (an over represented sub-graph) if, for some set of parameters p, U, D, N these three requirements are satisfied:

1 $\mathbb{P}(f_{rand}(G_k) > f_{original}(G_k)) < p$

2 $f_{original}(G_k) > U$

3 $f_{original}(G_k) - f_{rand}(G_k) > Df_{rand}(G_k)$

A Z score can be assigned to each sub-graph such that

$$Z = \frac{f_{original}(G_k) - < f_{rand}(G_k) >}{s(f_{rand}(G_k))} \quad (8)$$

where $s(\cdot)$ is the sample standard deviation. Another important feature due to its interpretation in many case studies is the **anti-motif** : an induced sub-graph that satisfies the following requirements:

1 $\mathbb{P}(f_{rand}(G_k) < f_{original}(G_k)) < p$

2 $f_{original}(G_k) - f_{rand}(G_k) < Df_{rand}(G_k)$

To perform motif analysis (i.e. to find motifs and anti-motifs in our network), we make use of the software `mfinder` version 1.2[20].

It has not a graphical interface, and we must use the terminal, running the following command:

```
mfinder1.2 ./data/motifAnalysis_CLOSED.txt

-s 3 -r 1500 -nu -ospmem 36
```

Here is the description of the flag commands we used:

`-s` is the desired sub-graph size

`-r` is the number of random networks to be generated

`-nu` ignores uniqueness threshold

`-ospmem` outputs members of a list of a specific sub-graph

In our case we generated r =1000 random networks searching for the motif with id=36 which corresponds to the triad schema (s = 3) $A \to B \leftarrow C$. Ignoring the uniqueness threshold we obtain the desired results about the pattern appearance in our graph as well as the random generated graphs.

The software outputs a table with some statistics regarding the candidate motifs that are repeated both in the randomized and our connections involved in the configured network.

By examining the frequency (z-score) and the statistical significance (p-value) of each candidate we chose as motifs all the candidates with $p \le 0.001$ and $Z \ge 3$, and as anti-motifs the sub-graphs with $p \ge 0.99$ and $Z \le -3$. Results are shown in Table I.

Given the results corresponding to the motif analysis we concluded in selecting two motifs (38, 108) and three anti-motifs (6, 36, 12) for case EO and just one anti-motif (id = 36) in case EC. Look at the Figure 5 to see the patterns which describe selected id.

Case Eyes Open				Case Eyes Closed			
id	frequency	z-score	p-value	id	frequency	z-score	p-value
38	86	4.37	0.0001	36	2559	-6.10	1.000
108	33	3.44	0.0001				
6	42	-6.43	1.000				
36	1338	-4.31	1.000				
12	100	-4.56	1.000				

TABLE I. motif-analysis

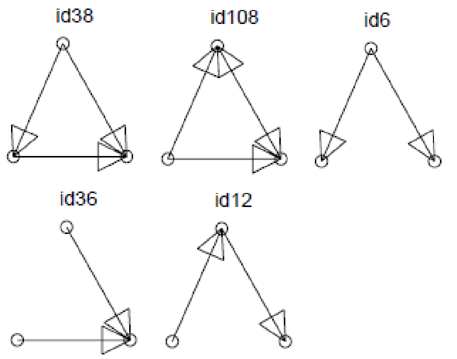


FIG. 5. Dictionary of the patterns that appear as motifs or anti-motifs in our case studies

As we shall see the anti-motif with id=36 appears in both resting cases.

A topographical representation (Figure 6 and Figure 7) is provided for the motif with pattern $A \to B \leftarrow C$ of the networks, considering only the connections involved in the configuration of the resting cases. Given the list of motifs with this pattern (id = 36) we draw edges with a larger width whether they connect two nodes more frequently.

Regarding motif analysis, it is also possible, given a channel, to obtain the list of motifs that channel is involved with. For example, in Figure 8 we list the count of

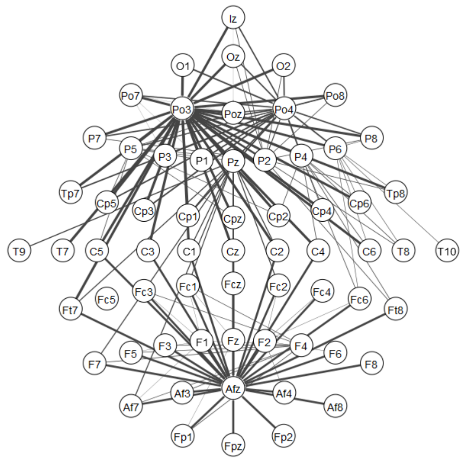


FIG. 6. Topographical representation of EO case for the motif with id = 36

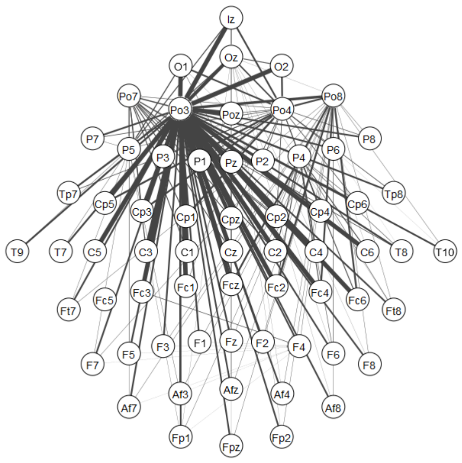
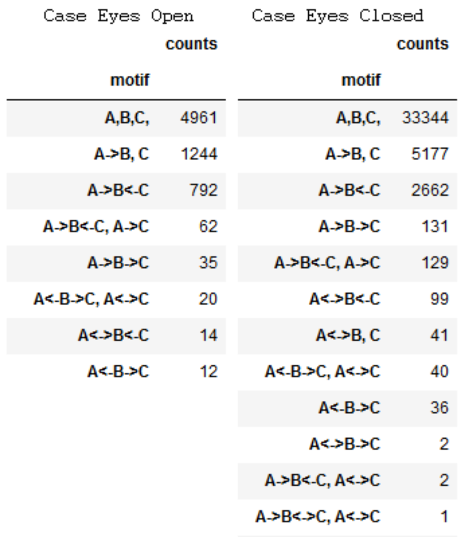


FIG. 7. Topographical representation of EC case for the motif with id = 36



Case Eyes Open

motif	counts
A,B,C,	4961
A->B, C	1244
A->B<-C	792
A->B<-C, A->C	62
A->B->C	35
A<-B->C, A<->C	20
A<->B<-C	14
A<-B->C	12

Case Eyes Closed

motif	counts
A,B,C,	33344
A->B, C	5177
A->B<-C	2662
A->B->C	131
A->B<-C, A->C	129
A<->B<-C	99
A<->B, C	41
A<-B->C, A<->C	40
A<-B->C	36
A<->B->C	2
A->B<-C, A<->C	2
A->B<->C, A<->C	1

FIG. 8. Motifs which involve the channel Po3

motifs that have channel Po3 as a node, which is placed in the parieto-occipital scalp region.

IV. COMMUNITY DETECTION

Based on the graph we have constructed before with a density equal to 5%, we identify the community structure of the network.

A network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally. The more general definition is based on the principle that pairs of nodes are more likely to be connected if they are both members of the same community(ies), and less likely to be connected if they do not share communities.

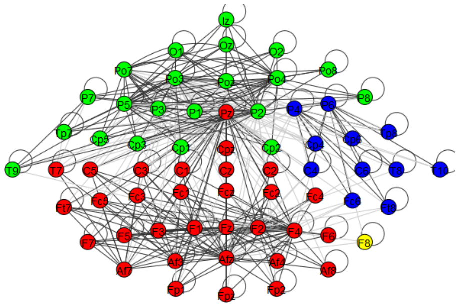


FIG. 9. EO case - Louvain method - Modularity based approach

Finding an underlying community structure in a network, if it exists, appears to be important since communities allow us to create a large-scale map of a network. This is because individual communities act like meta-nodes in the entire network, fact that makes the analysis easier[17].

For determining the number and composition (list of nodes and the corresponding community) of the communities, we used two methods which are based on different principles. So, in the context of community detection, we use the Louvain Method and the map equation known as Infomap. The former method is considered as a modularity-based approach in a sense that: first small communities are found by optimizing modularity locally on all nodes, then each small community is grouped into one node and the first step is repeated. The value to be optimized is modularity itself, defined as a value between -1 and 1 that measures the density of links inside communities compared to links between communities. Generally, for a weighted graph, (as in our case) modularity is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \qquad (9)$$

where A_{ij} represents the edge weight between node i and j, k_i and k_j are the sum of the weights of the edges attached to nodes i and j respectively, $2m$ is the sum of all the edge weights in the graph, c_i and c_j are the communities of the nodes.

This method when applied is like performing a clustering algorithm that results in hierarchical refinements of the network's partitions. For this purpose we used the implementation available on the `python-louvain` package which facilitates community detection of networks and builds on the `igraph` package[18].

The latter approach, the application of Infomap, is mainly used for directed weighted graphs in order to identify communities according to how the information flows in the network. Huffman codes are used to give short

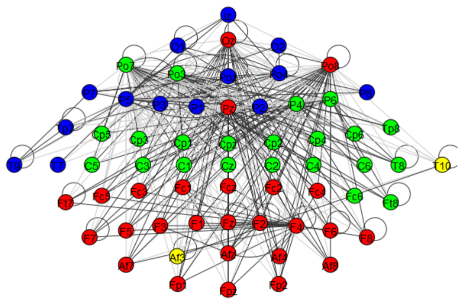


FIG. 10. EC case - Louvain method - Modularity based approach

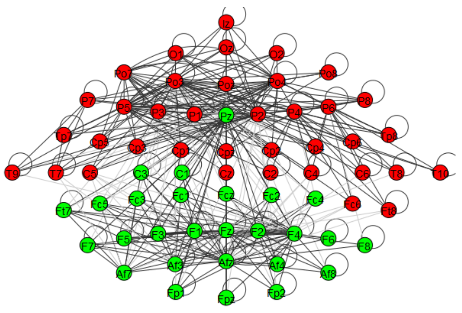


FIG. 11. EO case - Infomap method - Information theory based approach

codewords for commonly visited nodes and long codewords for rarely visited nodes. Below we cite the map equation and at the same time explain each component[19].

$$L(M) = q_{\curvearrowleft} H(Q) + \sum_{i=1}^{m} p_{i\circlearrowright} H(P_i) \tag{10}$$

$$L(M)$$

is the per-step description length for module partition M. That is, for module partition M of n nodes into m modules, the lower bound of the average length of the code describing a step of the random walker.

$$q_{\curvearrowleft} = \sum_{i=1}^{m} q_{i\curvearrowleft}$$

is the rate at which the index codebook is used. The per-step use rate of the index codebook is given by the total probability that the random walker enters any of them modules.

$$H(Q) = -\sum_{i=1}^{m} \left(\frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}} \right) \log \frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}}$$

is the frequency-weighted average length of codewords in the index codebook. The entropy of the relative rates to use the module codebooks measures the smallest average codeword length that is theoretically possible.

$$p_{i\circlearrowright} = \sum_{\alpha \in i} p_{\alpha} + q_{i\curvearrowright}$$

is the rate at which the module codebook i is used, which is given by the total probability that any node in the module is visited, plus the probability that the random walker exits the module and the exit codeword is used.

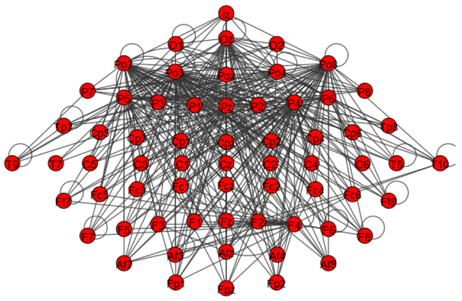


FIG. 12. EC case - Infomap method - Information theory based approach

$$H(P_i) = -\frac{q_{i\curvearrowright}}{p_{i\circlearrowright}} \log \frac{q_{i\curvearrowright}}{p_{i\circlearrowright}} - \sum_{\alpha \in i} \frac{p_{\alpha}}{p_{i\circlearrowright}} \log \frac{p_{\alpha}}{p_{i\circlearrowright}}$$

is the frequency-weighted average length of codewords in module codebook i. The entropy of the relative rates at which the random walker exits module i and visits each node in module i measures the smallest average codeword length that is theoretically possible.

Essentially, the function and the essence of Infomap is the minimization of the description length (Rissanen, 1978) of a random walker defined on the network through a set of processes. Consequently, we used the Infomap implementation which was available in the `igraph` package.

Regarding the performance of the two previously mentioned methods we can conclude that with decreasing the network density the number of communities increases in both cases. More specifically, with density equal to 5%, Infomap always finds few communities regarding both resting cases. However, when a modularity-based method is applied such as the Louvain method, more communities than Infomap are found (always strictly more than one for both rest cases) for the specific average degree of our network. See Figure 9 and Figure 10.

These results show that with increasing network density, the Louvain Method results are not representative enough since we have each time few small communities either our graph is relatively dense or sparse. On the other hand, the Infomap method (information theory approach) detects always, even when our graph is sparse enough, a single community in the rest case where the eyes are closed (fact that intuitively makes some sense) while in the eyes-open case when applied a mid-range density (5%-20%) the number of computed communities varies from 1-3. See Figure 11 and 12.

V. REFERENCES

[1] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages; `http://circ.ahajournals.org/cgi/content/full/101/23/e215`]; 2000 (June 13). Dataset at `https://physionet.org/physiobank/database/eegmmidb/`

[2] `pyedflib` module to read EDF files in `python`. `https://github.com/holgern/pyedflib`

[3] `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4873517/`

[4] `https://en.wikipedia.org/wiki/Brain_connectivity_estimators`

[5] Foster, J. J., Sutterer, D. W., Serences, J. T., Vogel, E. K., Awh, E. (2017). Alpha-Band Oscillations Enable Spatially and Temporally Resolved Tracking of Covert Spatial Attention. Psychological Science, 28(7), 929–941. `https://doi.org/10.1177/0956797617699167`

[6] New vistas for α-frequency band oscillations, Satu Palva,J. Matias Palva, Trends in Neurosciences, Elsevier, April 2007

[7] `https://en.wikipedia.org/wiki/Beta_wave`

[8] Watts, Duncan J.; Strogatz, Steven H. (June 1998). "Collective dynamics of small-world; networks";. Nature. 393 (6684): 440–442

[9] `https://en.wikipedia.org/wiki/Small-world_network`

[10] `https://arxiv.org/pdf/cond-mat/9903108.pdf`

[11] Neal, Zachary P. (2017). "How small is it? Comparing indices of small worldliness". Network Science. 5 (1): 30–44. doi:10.1017/nws.2017.5. ISSN 2050-1242.

[12] Telesford QK, Joyce KE, Hayasaka S, Burdette JH, Laurienti PJ. The ubiquity of small-world networks. Brain Connect. 2011;1(5):367-75.

[13] `www.brain-connectivity-toolbox.net`

[14] `https://github.com/aestrivex/bctpy`

[15] Maslov S. Sneppen K. Specificity and stability in topology of protein networks. Science. 2002;296:910–913.

[16] The small world of the cerebral cortex. Sporns O, Zwi JD Neuroinformatics. 2004; 2(2):145-62.

[17] `https://en.wikipedia.org/wiki/Community_structure`

[18] `https://en.wikipedia.org/wiki/Louvain_Modularity`

[19] `http://www.mapequation.org/assets/publications/mapequationtutorial.pdf`

[20] It is available at `http://www.weizmann.ac.il/mcb/UriAlon/research/network-motifs`

Appendix A: Extra Material

The code that produces these results is contained inside `main.html` that can be opened with a common browser. Many of the functions can be found into `mvar.py`, `mygraph.py`, `mylib.py`.

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