École polytechnique fédérale de Lausanne

Institute of Mathematics

Bachelor's thesis

Spring Semester 2021

Spatial Modularity of EEG networks

Carried out in the Chair of Statistical Data Science

Under the supervision and direction of
Professor Sofia Olhede

Authored by
Kieran Vaudaux

Contents

1 Introduction

This project will study methods to characterise the brain function of coma patients based on electroencephalography (EEG) measurements. Although the invention of electroencephalography is generally attributed to British physician Richard Caton in 1875, this decade will mark the first 100 years of electroencephalography as we know it today, a relatively simple tool for the non-invasive measurement of brain activity. Since the first EEG recordings in humans performed by the physiologist and psychiatrist Hans Berger in 1924, EEG has considerably evolved and now represents a cornerstone for the instrumental assessment of patient with disorders of consciousness(DoC). Consciousness is a complex concept, divided into two main components: Arousal and Awareness. Arousal is the physiological and psychological state of being awoken, while awareness is the state of being conscious of something, that is the ability to directly know and perceive or to be cognizant of events. The disorders of consciousness can generally be categorised into three states : minimally conscious state, vegetative state and coma.

In recent years, research has progressed rapidly and has developed an evidence base for neurotechnology in the assessment of consciousness after brain injury, such as may occur after cardiac arrest, and which may result in prolonged disorders of consciousness. One of these prolonged disorders of consciousness is coma. Perhaps even more than for other prolonged disorders of consciousness, from a clinical point of view, the availability of predictive markers at this early stage can complement the current neurological tests performed during the first hours of coma in order to eventually improve the accuracy of prognosis. Thus, in the long term, this study aims to investigate the predictive capacity of topological measures obtained from EEG recordings of patients during the first 24 hours after entering coma. For this purpose, we have at our disposal the 62-channel resting EEG recording during the first 24 hours of coma after cardiac arrest while patients were mildly sedated and treated with targeted temperature management. In total, we have recordings from 98 patients, 57 of whom had a favourable outcome, and for each patient the continuous EEG data were first segmented into epochs of five seconds. However, in this first study we have a sample of about ten EEG recordings. Patients who are considered to have had a favourable outcome are those who have fully recovered from their coma and those who have recovered but remain moderately impaired at any time in the three months after cardiac arrest, i.e. patients who score 1 or 2 on the Cerebral Performance Categories (CPC) test at any time in the three months after cardiac arrest. For more details on these recordings, the article [1] provides more information on the conditions and protocols of these.
In this paper, we will focus on the analysis of potential clusters among the 62 sensors, in order to further investigate whether or not the "quality" of these clusters can help us to significantly predict the outcome of a patient, as early as the first day after his cardiac arrest.

To be able to group the sensors into different clusters, we created networks from the EEG recordings for each patient and for each 5-second epoch. Networks are a practical method used to mathematically model sparse interaction patterns. Concretely, a network is a graph (i.e. a set of nodes numbered from $i = 1, ..., n \in \mathbb{N}^+$ and some of which are connected by edges) where nodes and/or edges have attributes. In this project, we will only study networks representing binary and symmetrical interactions between sensors (modelled by the nodes of the graph). In the literature many ways have been introduced to mathematically represent networks. The one we are going to use consists in representing a network by its adjacency matrix, i.e. the symmetric matrix $A = (A_{ij}) \in \mathbb{R}^{n \times n}$ such that all entries take the value 0, unless there is a edge between the nodes i and j, in which case $A_{ij} = 1$. Moreover, we only consider networks that do not have self-edges, therefore we always have $A_{ii} = 0$.

In network theory, there are many structures for a network. These include co-peripheral networks, hierarchical networks, assortative networks and conversely dissasortative networks. Co-peripheral networks consist of a single core sub-graph in which the nodes are highly inter-connected, with additional nodes on the periphery that are loosely connected to the core. Hierarchical networks have a hierarchical structure between nodes or between hubs. However, these two network structures are not the most appropriate to describe the network structure obtained from EEG recordings. According to current knowledge, the brain is made up of several regions, each with distinct and complementary functions. The co-peripheral and hierarchical structures therefore do not seem to be the most suitable to describe the structure of our EEG networks. The assortative and dissasortative networks, which will be presented later, seem to better represent the structure of the brain. For the rest of the project, we will

assume the presence of an assortative network structure, as it seems to be the most plausible structure for our networks. In order to be able to separate the nodes of a network into different communities, it is necessary to be able to quantify the community structure of a given partition of a network. In this spirit, several "cost measures" or "quality measures" have been proposed. For example, for a network given by its adjacency matrix A, one might want to minimize the number C of edges running between two groups of nodes, also called the *cut size*, $C = \frac{1}{2}\sum_{i,j=1}^{n} A_{ij} \cdot \delta(g_i, g_j)$, where $\delta(g_i, g_j) = 1$ if nodes i and j are in the same vertex group and 0 otherwise. However, as discussed in [2], if the size of the different communities is not known a priori, cut sizes are not the right function to minimise because they do not accurately reflect our intuitive concept of network communities. According to Newman [2, 3], an intuitive way of thinking about the accuracy of a division of a network into communities is to look at whether the number of edges within community is higher than expected. This leads us to consider a modified cost measure Q defined by $Q = \frac{1}{2m}\sum_{i,j=1}^{n}(A_{ij} - P_{ij})\delta(g_i, g_j)$, where P_{ij} is the number of edges expected between the nodes i and j. What is meant by the "expected number of edges between the i and j nodes" may vary across the literature and is highly dependent on additional assumptions about the process that generated our network. But in our case, we will refer to the meaning Newman gave it. Namely, $P_{ij} = \frac{d_i d_j}{2m}$, where $d_i = \sum_j \mathbb{E}[A_{ij}]$ is the expected degree of the node i, which will be estimated by the observed degree $d_i \approx k_i = \sum_j A_{ij}$. So for the rest of this project, our cost function will be given by :

$$Q(G) = \frac{1}{2m}\sum_{i,j=1}^{n}(A_{ij} - \frac{k_i k_j}{2m})\delta(g_i, g_j) \tag{1}$$

where $A = (A_{ij})$ is the adjacency matrix of the graph G, $k_i = \sum_j A_{ij}$ is the observed degree of the node i, $m = \sum_{i<j} A_{ij}$ is the number of edges in the graph and δ is as in the definition of the cut size C.

As Christian Hennig states in his article "Pattern Recognition Letters", [4], for reals clustering aims, true cluster need to be defined based on a certain truth behind the data. According to him, there are two possibilities for doing this. Firstly, one could assume that in the "real world" there is true clustering information for all observations, which is available in principle but not used by the clustering method. Secondly, one could assume that the data are generated by a true probability model, and then define the truth in terms of this model. In this article, we will assume that we are in this second case, and therefore assume that the community structure we are looking for has been generated by a probability model, the stochastic block model. This model allows us to model the intuitive structure of a network formed by several communities, namely a number of edges within groups higher than expected and a number of edges between groups smaller than expected.

After making the connection between the stochastic block model and modularity, which will support the use of modularity as a cost function, we will then introduce different methods of communities detection based on modularity maximization, then test their robustness for group detection and test our methods on some real EEG networks. Finally, having somewhat set aside the fact that the networks we are studying come from real EEGs, we will use the spatial positions of the sensors, together with our community detection methods to study the spatial structures of these networks.

2 Modularity and Stochastic Block Model

In this section, we will motivate the use of modularity maximization methods for our project. Indeed, we will first present two of the most popular and widely used methods for community detection in undirected and unweighted networks : the modularity maximization method and the likelihood maximization method applied to a stochastic block model. Then, following Newman's article [5], we will see that these two methods are equivalent in a special case of the stochastic block model, in which all communities of a network are assumed to have similar statistical properties.

2.1 Community detection based on modularity maximization

Modularity maximisation is one of the most widely used methods of community detection in assortative networks. As introduced earlier, modularity is a benefit function that measures the quality of a network division into communities. The method of maximising modularity consists in optimising the cost

function among all possible divisions of the network and choosing the final division of the network as the one that gives the highest score. However, in our case, and especially in the case of large networks, the number of possible divisions of a network is too large to find the global maximum of modularity. To overcome this problem, we will have to use different methods to approximate our optimisation problem. Many approximation methods already exist and will be discussed in more detail in section 3.

As mentioned in the introduction, a natural way of thinking about modularity is not just to look at the number of edges within groups, but rather the difference between the number of edges within groups and the expected number of such edges, if the same number of edges were placed randomly in the network. In the case of a network divided into q assortative communities, this idea led Newman to define modularity, in its general form, as :

$$Q = \frac{1}{2m} \sum_{i,j=1}^{n} (A_{ij} - P_{ij}) \delta(g_i, g_j) \tag{2}$$

where $A = (A_{ij})_{i,j=1}^{n}$ is the adjacency matrix of the network, g_i the group of node i, $\delta(g_i, g_j)$ the Kronecker delta, m the total number of edges in the network and $P_{ij} = \mathbb{E}[A_{ij}]$ is the expected number of edges between nodes i and j. The factor $\frac{1}{2m}$ makes Q to be a proportion of edges, which makes it easier to compare the value of Q between different networks. Furthermore, as discussed in [2] about the *cut size*, it is natural to try to exclude the extreme cases where all vertices belong to the same group. We then notice that the hypothesis on the constant number of sides in the network implies that :

$$\sum_{ij} P_{ij} \approx 2m \left(= \sum_{ij} A_{ij} \right) \tag{3}$$

And so that $Q = 0$ if all the nodes assigned to the same community. Thus, putting all nodes in the same community is no longer a viable way to maximise modularity.

Now we need to estimate the values of P_{ij}. Estimation of P_{ij} must be done under the assumption of a correct model specification. Otherwise, we would estimate P_{ij} by A_{ij}, which is too noisy. As discussed in [2], assigning values to P_{ij} is essentially equivalent to choosing a "null model" that would describe the distribution of sides in our network. A null model is a model that we believe is at the origin of the process that generated the networks we observe. One of the simplest models one could consider would be that of an Erdos-Rényi network with an edge probability $0 < p < 1$. In this model, the adjacency matrix of the network is generated component by component as $A_{ij} \overset{iid}{\sim} Bernoulli(p)$ for $j < i$ and we complete the matrix by setting $A_{ii} = 0$ and $A_{ij} = A_{ji}$. For a p well-chosen ($p = \binom{n}{2}$) and estimating P_{ij} by $\mathbb{E}(A_{ij}) = p$, we would have that our model verifies the condition Eq.(3). However, as pointed out by Newman [5], this model is not a good representation of most real-world networks. A more interesting model to consider is the stochastic block model. This model will be presented in more detail in section 2.2, but, concisely, in this model we consider the existence of a distribution of nodes in q distinct communities, with an edge probability between nodes i and j which depends only on the communities of these, resp. g_i and g_j. In practice, this model mainly fails to respect the degree of the nodes in the network, [5]. To overcome this, we can restrict our null model with the additional condition that the expected degree of node i is d_i, i.e. :

$$\sum_{j} \mathbb{E}[A_{ij}] = \sum_{j} P_{ij} = d_i \tag{4}$$

Let us note that if the condition Eq.(4) is verified then Eq.(3) is automatically verified when we approximate the expected degree d_i by the observed degree $k_i = \sum_j P_{ij}$, knowing that $\sum_i k_i = 2m$. According to [2], the simplest null model under these conditions is to assign the sides randomly under the constraints Eq.(4). We thus have that the probability that the end of an edge attaches to a vertex i depends only on the expected degree of this vertex d_i, and that the probability of the two ends of a single edge are independent from each other. This implies that we can write P_{ij} as the separate function product of d_i and d_j : $P_{ij} = f(d_i)f(d_j)$, where the functions are identical because $P_{ij} = P_{ji}$.

Eq.(4) then gives us :

$$\begin{aligned} d_i \overset{(4)}{=} \sum_j P_{ij} = f(d_i) \sum_j f(d_j) \ , \forall i = 1, ..., n \\ \Rightarrow f(d_i) = C d_i \ , \text{ for a constant } C = \sum_j f(d_j) \end{aligned} \tag{5}$$

Finally, by estimating $\sum_{ij} P_{ij}$ using Eq.(3), the expected degree d_i by the observed degree k_i and using the fact that $\sum_{ij} k_i k_j = \left(\sum_i k_i \right)^2 = 4m^2$ this allows us to conclude that :

$$\begin{aligned} 2m = \sum_{ij} P_{ij} = C^2 \sum_{ij} k_i k_j = 4m^2 C^2 \\ \Rightarrow C = \frac{1}{\sqrt{m}} \end{aligned} \tag{6}$$

This gives us :

$$P_{ij} = \frac{k_i k_j}{2m} \tag{7}$$

By combining Eq.(2) and Eq.(7), we obtain the modularity as we had defined it in Eq.(1), for this project :

$$Q = \frac{1}{2m} \sum_{i,j=1}^{n} (A_{ij} - \frac{k_i k_j}{2m}) \delta(g_i, g_j) \tag{8}$$

Nevertheless, as Fortunato and Barthélémy have shown [6], the detection of assortative community structure by maximising modularity using the definition Eq.(8) is efficient in many situations, but fails in particular to detect a community structure in large networks with many small communities. But this will not be a problem in our case, as we will study networks with a small number of vertices. However, to address this problem, Reichardt and Bornholdt [7] proposed a generalized modularity function :

$$Q(\gamma) = \frac{1}{2m} \sum_{i,j=1}^{n} (A_{ij} - \gamma \frac{k_i k_j}{2m}) \delta(g_i, g_j) \tag{9}$$

When $\gamma = 1$, the generalized modularity is the same as in Eq.(8), but other choices of gamma allow us to vary the relative weight given to the observed sides. We introduce this generalised form of modularity mainly to be able to make the link with the stochastic block model that we will present in the next section. The next section will also allow us to introduce in a more rigorous framework the stochastic block model, which will allow Newman, in [5] to propose an algorithm to approximate the optimal value of γ. The latter, again according to Newman's results, will tend to find smaller communities as γ increases.

2.2 Community detection based on maximum likelihood and stochastic block model

The other method of community detection we consider is the method of statistical inference, as applied to the stochastic block model and its variants, which will be defined below. With this method, one fits a generative model of a network to observed network data, and the parameters of the fit tell us about the structure of the network in much same way that fitting a straight line through a set of data points tell us about their slope.

2.2.1 Stochastic block model

In order to perform statistical inference on networks, one must first specify a general model on which to work. In our case, in the context of community detection in networks, one of the most commonly

used models is the stochastic block model in its Bernoulli version. This model considers a network with n nodes, initially with no sides, and divides them into q groups, with g_i being the group to which node i is assigned. Then the sides are placed randomly and independently in the graph, so that the probability, ω_{rs}, of a side between a given pair of nodes depends only on the groups r and s to which these nodes belong. We can therefore group these parameters into a symmetric matrix $q \times q$, which determines the probability of having a side in or between each pair of groups. In a natural way, one will tend to attribute a community structure to networks that have been generated by a model where the diagonal elements, ω_{rr}, of the parameters matrix are larger than the off-diagonal elements. This type of model, with a higher probability of edges within groups than between them, is known as an assortative model, and conversely, a model for which the probability of an edge between groups is greater than that of an edge within a group, is known as a disassortative model.

However, we will look at the stochastic block model in a slightly different formulation, in which we do not just place one side between a pair of vertices, but a number of sides that have a Poisson distribution with mean ω_{rs}. In this configuration, ω_{rs} is no longer the probability of a side between two nodes that are in the groups r and s, but is now the average number of sides between these two vertices. Moreover, in this configuration we also allow the presence of self-edges, which also follow a Poisson distribution, with mean $\frac{1}{2}\omega_{rr}$ for a vertex in the r group (the factor one $\frac{1}{2}$ is there only to simplify the calculations afterwards) Most data networks encountered in the real world do not contain multi-edges or self-edges, which might make our configuration not very realistic. Nevertheless, as most real-world networks are very sparse, we have that the average number of sides ω_{rs} is also very small. Thus, the amount of self-edges and multi-edges in the network is very small and according to Newman the Poisson version of this model is, in practice, indistinguishable from the Bernoulli version. As in Newman's paper [5], we will focus on the Poisson version for simplicity, but similar results can be derived from the Bernoulli version.
The model defined above can be used to generate networks. However, as we are interested in community detection, we will use this model in the reverse direction to infer a community structure by fitting it to data. To this end, we will assume that an observed network, with an adjacency matrix $\mathbf{A}$, has been generated by a Poisson version of the stochastic block model and try to estimate the parameters ω_{rs} and g_i using the maximum likelihood principle.
In our Poisson model, the probability of observing A_{ij} sides between vertices i and j is given by :

$$\mathbb{P}(\mathbf{A}_{ij} = A_{ij}) = \begin{cases} \frac{(\frac{1}{2}\omega_{g_i g_i})^{\frac{A_{ii}}{2}}}{(\frac{1}{2}A_{ii})!} e^{-\omega_{ii}/2} & \text{if } i = j, \\ \frac{\omega_{g_i g_j}^{A_{ij}}}{A_{ij}!} e^{-\omega_{ij}} & \text{if } i \neq j. \end{cases} \tag{10}$$

where we have adopted the convention that a self-edge is represented by a diagonal adjacency matrix element $A_{ii} = 2$.
For given values of the parameters, we can write the likelihood that an observed network is generated by our model:

$$\mathbb{P}(\mathbf{A}|\mathbf{\Omega}, \mathbf{g}) = \prod_i \frac{(\frac{1}{2}\omega_{g_i g_i})^{\frac{A_{ii}}{2}}}{(\frac{1}{2}A_{ii})!} e^{-\omega_{ii}/2} \prod_{i<j} \frac{\omega_{g_i g_j}^{A_{ij}}}{A_{ij}!} e^{-\omega_{ij}} \tag{11}$$

where Ω denotes the matrix $q \times q$ of parameters ω_{rs}.
As we are interested by the group assignments $\mathbf{g}$, which tell us how the network divides into groups, we are looking for the parameters Ω and $\eth$ which allow to maximize the likelihood of the observed network. As in most likelihood maximisation problems, we can equivalently reduce ourselves to the log likelihood maximisation problem which is often simpler to handle :

$$\log \mathbb{P}(\mathbf{A}|\mathbf{\Omega}, \mathbf{g}) = \sum_i \left[\frac{1}{2} A_{ii} \log(\frac{1}{2}\omega_{g_i g_i}) - \frac{1}{2}\omega_{g_i g_i} - \log\left\{ (\frac{1}{2}A_{ii})! \right\} \right] + \sum_{i<j} \left(A_{ij} \log(\omega_{g_i g_j}) - \omega_{g_i g_j} - \log\left\{ (A_{ij})! \right\} \right) \tag{12}$$

But, since we want to maximise the log-likelihood, we can ignore the constant terms in Eq.(12). This gives us, when simplified :

$$\log \mathbb{P}(\mathbf{A}|\mathbf{\Omega}, \mathbf{g}) = \frac{1}{2} \sum_{ij} \left(A_{ij} \log(\omega_{g_i g_j}) - \omega_{g_i g_j} \right) \tag{13}$$

Then, it only remains to maximize this quantity according to $\mathbf{\Omega}$ and $\mathbf{g}$ to find the optimal distribution of the network in various communities. However, maximising such an expression could be very complicated as soon as the size of our network is large.

2.2.2 Degree-corrected block model

As in the section 2.1, on the maximization of modularity, this approach gives sub-optimal results on networks which come from the real world. Indeed, it fails in particular to respect the nodes degrees in the network, the stochastic block model described previously generates networks which have a Poisson distribution of degrees, which is not the case for the majority of the observed networks. In our situation, the classical solution to this problem is to consider a slightly different model to the one used in the previous section. This new model is called the degree-corrected block model, which better fits the degree distribution to the observed one. In this model, the nodes are still assigned to g_i groups and the sides are placed independently at random between the nodes following a Poisson distribution, but the difference is that now the expected number of sides between nodes i and j is $\frac{k_i k_j}{2m}\omega_{g_i g_j}$ if $i = j$ and $\frac{k_i k_j}{4m}\omega_{g_i g_j}$ if $i = j$, where k_i is the degree of node i, m the number of sides in the network. For this new model we obtain the following likelihood:

$$\mathbb{P}(\mathbf{A}|\mathbf{\Omega}, \mathbf{g}) = \prod_i \frac{(\frac{k_i k_i}{4m}\omega_{g_i g_i})^{\frac{A_{ii}}{2}}}{(\frac{1}{2}A_{ii})!} e^{-\frac{k_i k_i}{4m}\omega_{ii}} \prod_{i<j} \frac{(\frac{k_i k_j}{2m}\omega_{g_i g_j})^{A_{ij}}}{A_{ij}!} e^{-\frac{k_i k_j}{2m}\omega_{ij}} \tag{14}$$

Considering as before the logarithm of the likelihood and neglecting the constant terms, we obtain the log-likelihood for this model:

$$\log \mathbb{P}(\mathbf{A}|\mathbf{\Omega}, \mathbf{g}) = \frac{1}{2}\sum_{ij}\left(A_{ij}\log(\omega_{g_i g_j}) - \frac{k_i k_j}{2m}\omega_{g_i g_j}\right) \tag{15}$$

However, there is still too much possibility of group assignment g_i to be able to exhaustively maximise the log-likelihood, nevertheless the researchers have successfully found a variety of approximate methods [8, 9, 10]„ on which we will not focus.

2.3 Link between stochastic block model and modularity : The planted partition model

We now come to the central result of Newman's article [5]: the equivalence between the modularity maximization problem, discussed in the section 2.1, and a particular case of the maximum likelihood method described in the two previous subsections.
The planted partition model is a special case of the degree-corrected block model in which the parameters $\omega_{g_i g_j}$, which characterize the community structure of the network, take only two values :

$$\omega_{g_i g_j} = \begin{cases} \omega_{in} & \text{if } g_i = g_j, \\ \omega_{out} & \text{if } g_i \neq g_j. \end{cases} \tag{16}$$

This model assumes that all communities in the network are similar in the sense of having the same in-group and between-group connection rates. If one restricts oneself to this model, one still encounters the problem due to a bad distribution of degrees. This is why in practice when applying the panted partition model, we also use a degree-corrected version of that model, as in subsection 2.2.2.Then, let us explore the form of the log-likelihood, Eq.(15), for such model. First, we note that we can write $\omega_{g_i g_j}$ and $\log \omega_{g_i g_j}$ as :

$$\begin{aligned} \omega_{g_i g_j} &= (\omega_{in} - \omega_{out})\delta_{g_i g_j} + \omega_{out} \\ \log \omega_{g_i g_j} &= (\log \omega_{in} - \log \omega_{out})\delta_{g_i g_j} + \log \omega_{out} \end{aligned} \tag{17}$$

where $\delta_{g_i g_j}$ is the Kronecker delta. Then, using Eq.(17) in Eq.(15) and after some algebraic manipulation, we find the log-likelihood for the degree-corrected planted partition model :

$$\begin{aligned}\log \mathbb{P}(\mathbf{A}|\mathbf{\Omega}, \mathbf{g}) &= \frac{1}{2}\log\frac{\omega_{in}}{\omega_{out}}\sum_{ij}\left(A_{ij} - \frac{\omega_{in}-\omega_{out}}{\log\omega_{in}-\log\omega_{out}}\frac{k_i k_j}{2m}\right)\delta_{g_i g_j} + m(\omega_{out}+\log\omega_{out}) \\ &= B\frac{1}{2m}\sum_{ij}\left(A_{ij} - \gamma\frac{k_i k_j}{2m}\right)\delta_{g_i g_j} + C\end{aligned} \tag{18}$$

where B and C are constants that depend on ω_{in} and ω_{out} but not on $\mathbf{g}$, and

$$\gamma = \frac{\omega_{in}-\omega_{out}}{\log\omega_{in}-\log\omega_{out}} \tag{19}$$

Then, at this stage to perform community detection, the first idea that one could have is to maximize this expression with respect to both the group assignments g_i and the parameters ω_{in} and ω_{out}. But, let us suppose that we already know the correct values of the parameters ω_{in} and ω_{out}, then we only have to maximize with respect to the group assignments. Under this assumption and apart from all the constants, we notice that Eq.(18) is exactly the generalized modularity $Q(\gamma)$ defined in Eq.(9), and therefore their extrema with respect to g_i are the same. However, a further point of interest is that while the value of γ in Eq.(19) is always positive, the value of the constant $B = m\log\frac{\omega_{in}}{\omega_{out}}$ in Eq.(18) changes sign depending on which of ω_{in} and ω_{out} is larger. Thus, in the an assortative model, we can conclude that community detection by maximizing the likelihood for the planted partition model with known values of ω_{in} and ω_{out} is equivalent to maximizing the generalized modularity for an appropriate value of γ, given by Eq.(19). Whereas in the case of a disassortative structure,i.e. $\omega_{out} > \omega_{in}$, then the maximization of the likelihood becomes equivalent to the minimization of the modularity. Furthermore, we note that the conventional choice $\gamma = 1$ which would allow us to relate to modularity(Eq.(2)) is not correct in most cases. However, even though the consistency of modularity maximization has already been demonstrated in other ways, the equivalence with likelihood maximization, under certain assumptions, gives a good intuition for the use of modularity in Eq.(2).

On the other hand, the equivalence between the maximisation of likelihood, in the case of an assortative structure, and the maximisation of modularity tends to reveal hidden assumptions and some limitations of modularity. Indeed, the assumption that the parameters $\omega_{g_i g_j}$ take the same values for each community is much more restrictive than the full stochastic block model. The modularity maximization is then less powerful as a result, since it assumes all communities in a network to be statistically similar, which is generally not always true.

This equivalence is also a potential direction, which it would be interesting to exploit in the future development of this project, where instead of considering the modularity defined by Eq.(2) we could use the general form, Eq.(9), of it with γ as in Eq.(19). By adopting this approach, we could then use the algorithm described at the end of the article, [5], on which this section is based, to be able to iteratively approximate the parameters ω_{in} and ω_{out}. Another point that could be developed in an extension of this paper would be the assumption of assortativity of our networks. This would put our modularity maximization approach in default.

3 Community Detection Methods

One can argue that community detection is similar to clustering. Clustering is a machine learning technique in which similar data point are grouped into the same cluster based on their attributes. Even though clustering can be applied to network, it is a broader field in unsupervised machine learning which deals with multiple attribute types. On the other hand, community detection is specially tailored for network analysis which depends on a single attribute type called edge. In this section, we will present some community detection methods that the igraph package puts at our disposal, then we will test these methods on different networks that we will have generated in several ways, and finally we will test these methods on some real networks from EEG of some patients.

3.1 Presentation of several community detection methods

3.1.1 The Fast Greedy method

We are going to present here the method fast greedy method implemented in igraph. It is an algorithm described as "gluttonous" which allows the constitution of a partition from a criterion based on modularity. It was first proposed by Newman, then by Clauset, Newman and Moore in a second, faster version, which works in $\mathcal{O}(md \log n)$, where d is the depth of the "dendrogram" describing the network's community structure, m the number of edges and n the number of vertices. They use modularity in the following form:

$$Q = \sum_i (e_{ii} - a_i^2) \tag{20}$$

where we use the notation :

$$e_{ij} = \frac{1}{2m} \sum_{vw} A_{vw} \delta(c_v, i) \delta(c_w, j)$$
$$a_i = \frac{1}{2m} \sum_v k_v \delta(c_v, i)$$

with k_v the degree of vertex v. Then, they define the quantity noted ΔQ_{ij} corresponding to the variation of the modularity when the i community and the j community are grouped in the same community. The details of the algorithm, with the indications related to the storage of information can be found in the article [11]. The general scheme of this algorithm is as follows:

Algorithm 1: Cluster Fast Greedy

Input: A adjacency matrix A = $(A_{ij})_{i,j=1}^n$, corresponding to a graph G
Output: A dendrogram, which represent the hierarchical structure that maximize the modularity

```
We start by assigning a different community to each vertex.
while There is more than one community do
    for Each pair (g_i, g_j) of communities do
        Calculate and store ΔQ_ij
    Merge the pair of community that increase modularity the most.
Cut the dendrogram at the value corresponding to the highest modularity.
```

3.1.2 The Leading Eigen-Vector method

Newman also proposed a spectral version of the partitioning of a graph based on the modularity. In this version, he introduces the modularity matrix $\mathbf{B}$ which gives the expression of the modularity defined by Eq.(1): $\mathbf{B}_{ij} = A_{ij} - \frac{k_i k_j}{2m}$. In the case of a division into two communities, Newman introduces the vector $\mathbf{s}$, such that $\mathbf{s}_i = 1$ if the vertex i belongs to the first group and $\mathbf{s}_i = -1$ if the vertex i appears in the second group. In the article [2], Newman shows that the maximization of modularity reduces to a problem that can be formalized by :

$$\text{Find } \mathbf{s} \in \mathbb{R}^n \text{ such that } \mathbf{s} = \arg\max_{\mathbf{x} \in \Omega} \frac{1}{4m} \mathbf{x}^T \mathbf{B} \mathbf{x} \text{ , where } \Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{s} = \pm 1\}.$$

As with ordinary spectral partitioning, this would be a simple task if our choice of $\mathbf{s}$ were unconstrained : we would just choose $\mathbf{s}$ proportional to the leading eigenvector $\mathbf{u}$ of the modularity matrix, which have the largest eigen value. However, as the components of $\mathbf{s}$ are restricted to ± 1 values, we cannot generally choose a vector $\mathbf{s}$ parallel to $\mathbf{u}$. But, as introduced in [2], a good approximate solutions can be achieved by setting :

$$\mathbf{s}_i = \begin{cases} +1 & \text{if } \mathbf{u}_i \geq 0, \\ -1 & \text{if } \mathbf{u}_i < 0. \end{cases} \tag{21}$$

Then, to divide the network into more than two communities, the simplest approach is to repeat this community division into two sub-communities, until there is no further increase in modularity.

For this purpose, Newman defines the quantity ΔQ which corresponds to the difference between the modularity of the network before and after the subdivision of the communities. Further details on the implementation of the algorithm can be found in [2]. In addition, the clustering function which implements this method can be found in the package igraph.

3.1.3 The Louvain method

Researchers at the University of Louvain have proposed another greedy method, which is faster than most other approaches. Its particularity is that it is based on a local approach to modularity. In a first phase, a different community is assigned to each vertex. Then, we are interested in the neighbours of each vertex i, and we compute the modularity gain by removing the vertex i and placing it in the community j. A positive and maximum gain is sought for moving i. This is done sequentially until no improvement is possible.
The second phase of the algorithm consists in building a new network whose vertices are the communities identified in the first phase, the weights of the links between the communities being determined by the sum of the weights of the links of the vertices of the initial graph.
Once this second phase is completed, phase one is applied again to this new weighted network. And, we proceed in this way until a maximum of modularity is reached.
The algorithm is presented in more details in the article [12]. This algorithm is implemented in the Louvain method which is also in the package igraph. Moreover, it is often presented as the most efficient in terms of partitioning time and quality.

3.1.4 The Spinglass method

With this method, we move away from the usual methods. To present this method in a rigorous way would take a certain time and would go somewhat beyond the framework of this work, we will thus be satisfied with a summary presentation. This method is inspired by spinglass, which are alloys corresponding to impurities, one spin being associated with each impurity. The coupling between the different spins can be more or less intense. This method is used in theoretical physics. The spin pairs are associated in a graph. We define a Hamiltonian graph (a graph with at least one cycle passing through all the vertices at most once) and a probability distribution of the couplings. Each vertex is characterised by a spin with q possible values, and the communities correspond to the values of vertices with equal spin values. The energy of the system is defined by a Hamiltonian using the adjacency matrix of the graph. The minimisation of this expression is done by simulated annealing. This makes this method a non-deterministic method, unlike the previous methods, i.e. we will potentially obtain a different result each time we run this algorithm.
As said above, a description in detail of this algorithm is left as it would go out of scope of this project. Note however that the description of this method is detailed in the following publication in greater details, [7]. Moreover, it is implemented in the Spinglass method which is also in the package igraph.

3.2 Testing methods on Erdös-Rényi random graphs

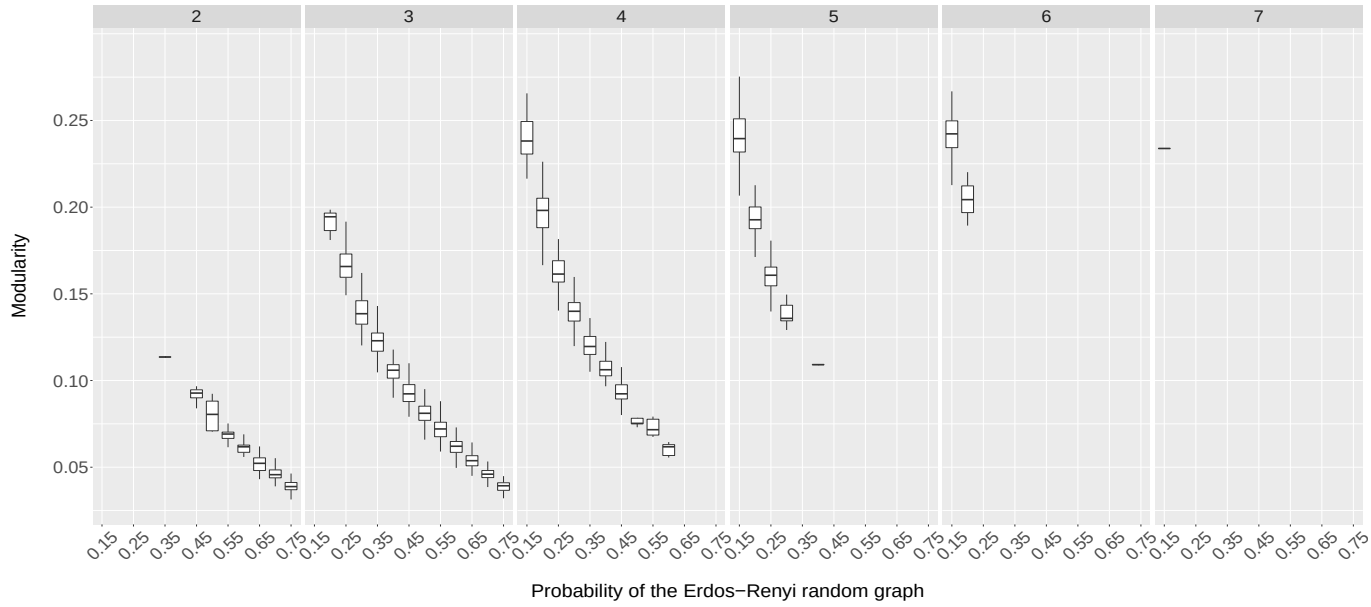
A natural question to ask is : "To what point can our clustering methods be led to put forward a community structure in a graph, whereas this graph is not supposed to have any?" To investigate this, we will in this section test these methods on randomly generated graphs, based on the Erdös-Rényi random graph model. As presented in Section 1.1, an Erdös-Rényi random graph, $ER(n, p)$, on n nodes with an edge probability $0 < p < 1$, such that $E(n, p)$ does not contain self-loop and such that the probability of the presence of a side between two distinct nodes is $Bernoulli(p)$ distributed and is independent across all pairs of distinct nodes.

To be able to test our methods, we have therefore randomly generated graphs using a method already implemented in igraph : *sample_gnp*. This method allows us to generate graphs according to the random model of Erdös-Rényi graphs, with $n = 62$ vertices and an edge probability p, where we varied $p \in 0.15, 0.2, ..., 0.75$ (for each p we generated 200 distinct graphs). Once these graphs were generated, we used the methods introduced in section 3.1 to see if they detected significant community structures in these different graphs.
From Fig.1 and Fig.13, we have that each facet represents the number of group detected by the method,

on each of these facets the x-axis represents the edge probability p of the generated graphs and the y-axis represents the modularity of the detected structures. For each facet and each probability p we have used box-plots to represent the modularity, which allows us to show the average of our observations as well as the dispersion of our observations around it.

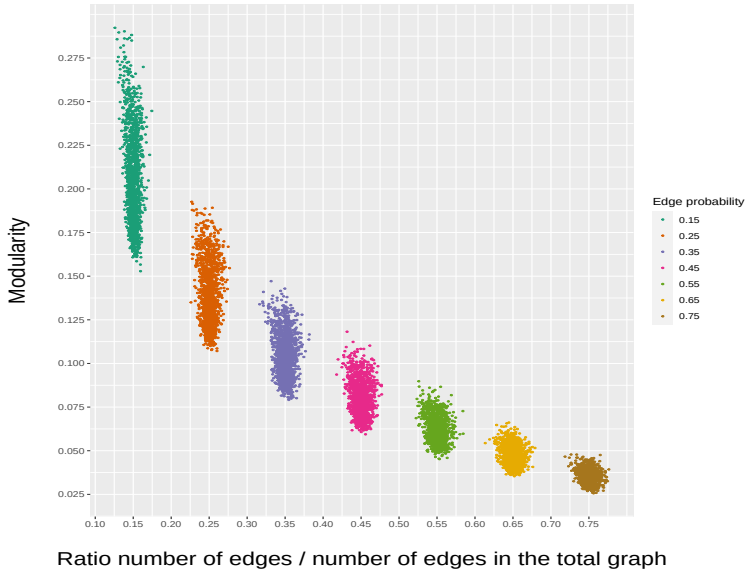
Fig.1 is obtained using the fast greedy method, letting the method choose the number of groups that maximizes modularity. We see that whatever the number of groups detected, the modularity decreases approximately linearly as a function of p. For example, for $p = 0.15$, we observe a modularity of the order of 0.225, which is not very far from the threshold of 0.3 which, in practice, indicates a significant community structure. However, the modularity quickly becomes less significant as p increases. This phenomenon could, quite naturally, be explained by the fact that for a low probability p, the average number of edges in the generated graphs is not very high, and that therefore our method would tend to overestimate the presence of a group structure when the number of edges of the graph is low compared to the number of vertices.Moreover, in the appendix on Fig.11, Fig.12 and Fig.13, we observe the same tendency with the other methods, which reinforces us in the idea that the fact of finding a significant modularity for graphs having been generated by a small edge probability is more due to a lack of density in the network than to a deficiency of our methods.

Figure 1: Analysis of the fast greedy method with box-plot in the case of ER(n,p) networks



To analyse this phenomenon a little more, for each probability $p \in \{0.15, 0.25, ..., 0.75\}$ we generated a certain number of graphs having between $n = 60$ and 120 vertices, then still with the fast greedy method we identified a group structure for each of these graphs. Then, as we can see on Fig.2, we plotted the modularity of these graphs according to the ratio R between the number of edges of the graphs and the number of edges in the complete graphs, where the number of edges in the complete graph is $\frac{(n-1)n}{2}$ with n the number of vertices of the graph. Not surprisingly we notice that the ratios R of the graphs are concentrated around the probabilities that generated them.

Figure 2: Modularity analysis as a function of edges density in graphs of varying sizes



As a result of these observations, it is not possible to draw any definitive conclusion on the consistency of our methods for graphs that are not expected to have any group structure.However, it seems reasonable to assume that for graphs with not too small side density (> 0.2) our methods should not detect any significant group structure, if there is no underlying structure.

3.3 Testing methods on stochastic block model random graphs

In the previous section, we were interested in the consistency of the results of our methods on networks that, a priori, should not have a community structure. We will now test our methods on graphs that have been generated with an underlying community structure. To do this we will randomly generate graphs using the *sample_sbm* method implemented in the igraph package.This method is based on a stochastic block model for networks, i.e. we assume the existence of k groups in the network and that the probability of having a side between nodes i and j, knowing the groups z_i and z_j of these vertices, follows a Bernoulli($\theta_{z_i z_j}$), with $0 < \theta_{z_i z_j} < 1$ and where each realization is independent.

To generate our graphs we followed the same procedure. Indeed, to generate a graph with a number k of groups and a given distribution of vertices in these various groups, we generated this graph for multiple probabilities $\theta_{z_i z_j}$. We placed ourselves in the case of the planted partition model (see section 2.3), i.e. we assigned the same probability of within edges θ_{in} and between edges θ_{out} for all groups.

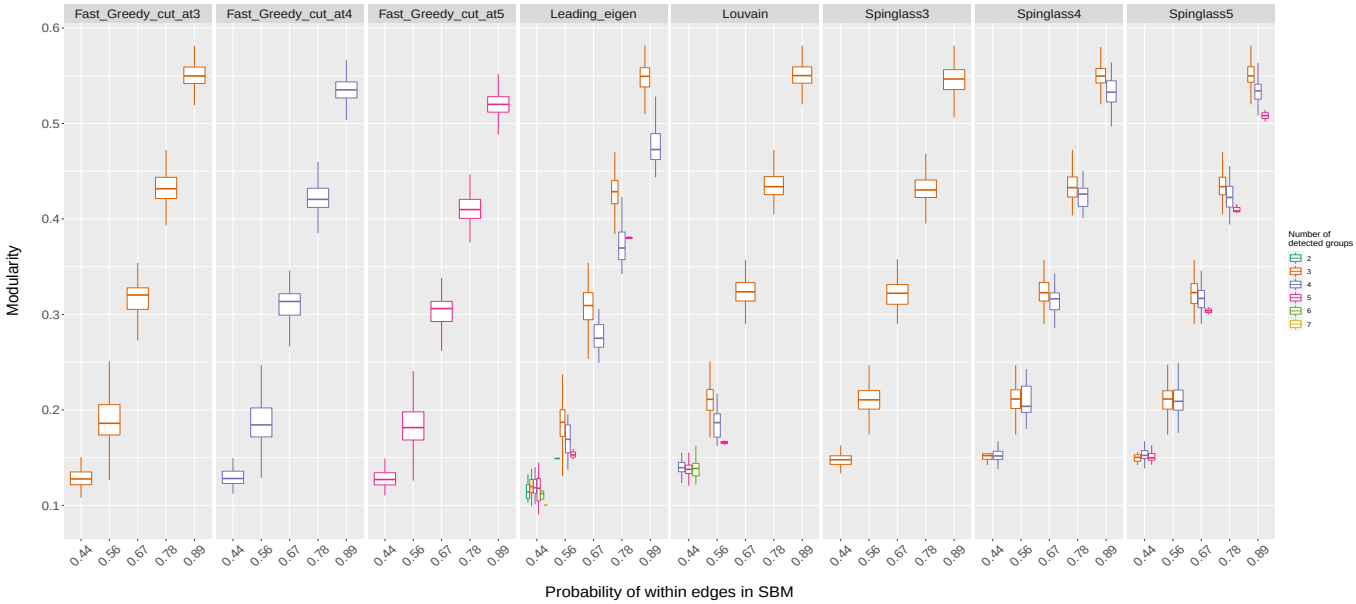
First, we focused on an assortative model ($\theta_{in} > \theta_{out}$). For a number of groups k, we generated graphs for five values of θ_{in} equally spaced in the interval $[\frac{1}{k}, 1]$, bounds not included. With θ_{out} given by $\theta_{out} = \frac{1-\theta_{in}}{k-1}$.

3.3.1 Uniform groups size

On Fig.10, we have generated a large number of graphs with 3 groups of vertices of uniform size (21 vertices per group). Each facet is dedicated to a method (or a variant) whose efficiency we want to test. The x axis represents the probability of within edges in our model. As in the previous section, we have used box-plots to represent the average and the dispersion of the modularity of the detected group structures, on the y axis. We also note that for the same probability θ_{in}, there can be several slightly shifted box-plots, with a different colour according to the number of groups, and whose size is proportional to the number of graphs with the same number of groups.
We observe that for all the methods the modularity increases when θ_{in} increases, and that for almost

all the methods the modularity passes the significant threshold of 0.3 for $\theta_{in} \approx 0.67$. We also notice that for the methods free to choose the number of groups, i.e. leading eigen method and Louvain method, the majority choose 3 or 4 groups.This is a good sign, given that we have generated our graphs assuming a structure of 3 groups. For the methods Spinglass1, Spinglass2 and Spinglass3, which are respectively limited to the detection of a maximum of 3, 4 and 5 groups, we also note that they mainly detect structures with 3 and 4 groups.

Figure 3: Analysis of clustering across methods in the case of SBM networks with 3 uniform groups (21/21/21)



Moreover, in the appendix B, on Fig.14 and Fig.15, for which we have generated graphs in the same way with respectively 4 uniform groups (of 15, 15, 16 and 16 vertices per group) and 5 uniform groups (of 13, 13, 13 ,12 and 12 vertices per group), we see that for each method the modularity of the detected structures behaves in the same way as on Fig.10.

Nevertheless, it is logical to ask if the detected group structures correspond to the underlying structures which allowed us to generate our graphs. Unfortunately, the function *sample_sbm* which generates our graphs does not allow us to know the distribution of the vertices of the graph in the groups. We cannot therefore formally compare the detected structure with the "real structure" of the graph. Moreover, we could have limited ourselves to analyze the number of vertices in the detected groups, with plots for example. But here again, the task will be complicated because we would need a plot for each graph to be able to interpret the results, which is not possible given the number of graphs generated. We will therefore have to make do with an informal analysis which I will develop below.

If we first restrict ourselves to the case of uniform 3-group graphs, as on Fig.10, we observe that the Fast_Greedy_cut_at3 method detects three groups whose size is of the same order as the size of the true SBM groups, and that for θ_{in} large enough (≈ 0.77) it even becomes accurate for almost every graph. However, if we look at the Fast_Greedy_cut_at4 and Fast_Greedy_cut_at5 methods, which are forced to detect respectively 4 and 5 clusters, we notice that whatever θ_{in}, these methods tend to detect three clusters whose size is of the same order as the true clusters, and that the last clusters generally contain 1 to 5 vertices each (a large part contains only one vertex). This phenomenon also appears for the other two cases of SBM with uniform clusters. Moreover, when it comes to detecting a number of clusters smaller than the true number of clusters, these methods seem to be consistent in detecting one large cluster and the rest of the clusters with a size of the same order as the true cluster

size. Thus, although these methods give equivalent results for modularity, it seems that the group structures they detect are largely more consistent when underestimating the true number of groups.
Then, as regards the Spinglass3, Spinglass4 and Spinglass5 methods, we observe globally the same accuracy in the detection of groups as the Fast_Greedy methods. However, as these methods only have an upper bound for the number of groups to be detected, and no longer a fixed number of groups, the recurrent phenomenon that appeared with the Fast_Greedy methods when the number of groups was overestimated almost does not appear with the Spinglass methods. Therefore, the Spinglass methods seem to be relatively robust in terms of the size of the groups detected.
As far as the Leading_Eigen method is concerned, we can see on Fig.14 and Fig.15, in the appendix, that this method is much less stable in terms of the number of groups detected. However, we could observe that even if it often detects too many groups, the size of these groups remains homogeneous most of the time. Finally, the Louvain method seems to be the method that provides the most stable and accurate group detection. Indeed, in addition to being accurate in terms of the number of groups detected, this method is particularly stable in terms of the size of the groups detected, which are mostly of the same order as the size of the real groups.

3.3.2 Heterogeneous groups size

In order to test other configurations of the stochastic block model, we proceeded as in the uniform case but this time with groups of different sizes. As we see below on Fig.15 in appendix B, for five groups of sizes 25, 25, 4, 4 and 5, the general behaviour of the modularity as a function of the within edge probability is similar to the uniform case. However, on Fig.10, in the case of three groups of sizes 53, 5 and 5, we observe a reversal of the trend, i.e. the modularity does not increase anymore when the probability of within edge increases, but it decreases with a maximum of the modularity which reaches 0.125, which is far from being a significant value for the presence of a group structure. Moreover, for each distribution of vertices in groups of different sizes, we notice that all the methods have a lot of difficulties to detect groups of the same size as the underlying groups. With the possible exception of the Spinglass and Louvain methods which, for a large within edge probability, manage to detect group structures similar to the true underlying structures, in the case of heterogeneous 4 and 5 group structures. However, in the case of three heterogeneous clusters, the distribution of cluster sizes is too extreme for our methods to detect the underlying structure, indeed, they detect either homogeneous cluster sizes or two large clusters with other clusters having only 1 or 2 vertices.

Figure 4: Analysis of clustering across methods in the case of SBM networks with 4 heterogeneous groups size (28,25,5,5)

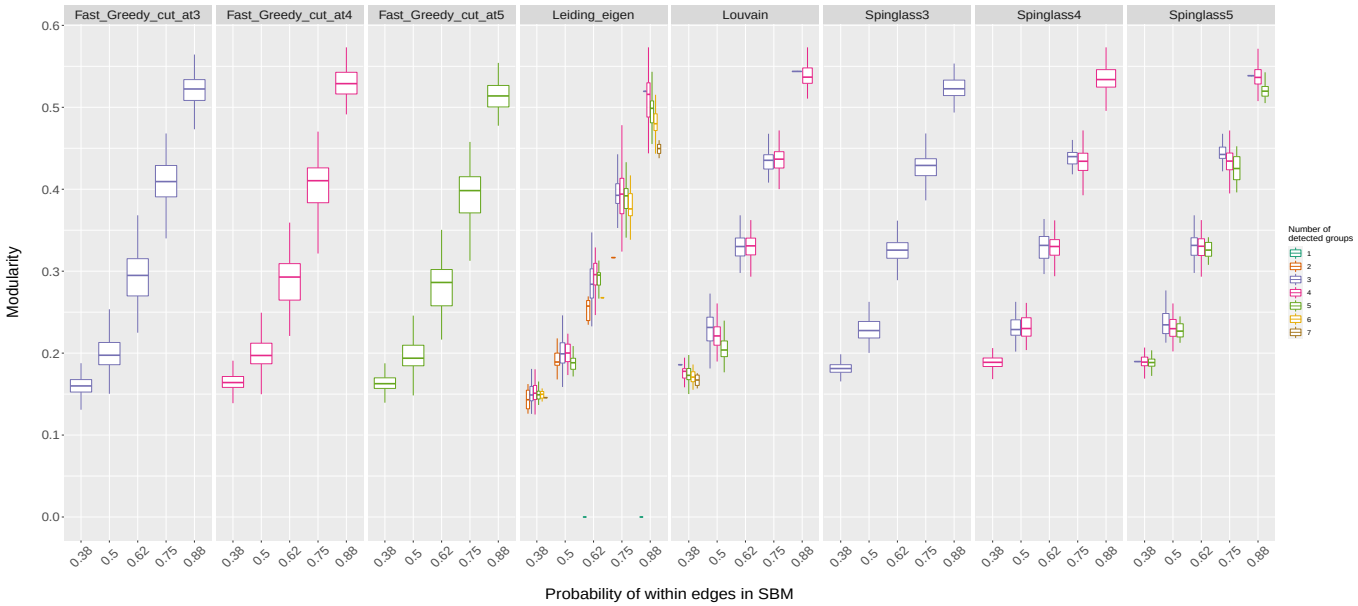
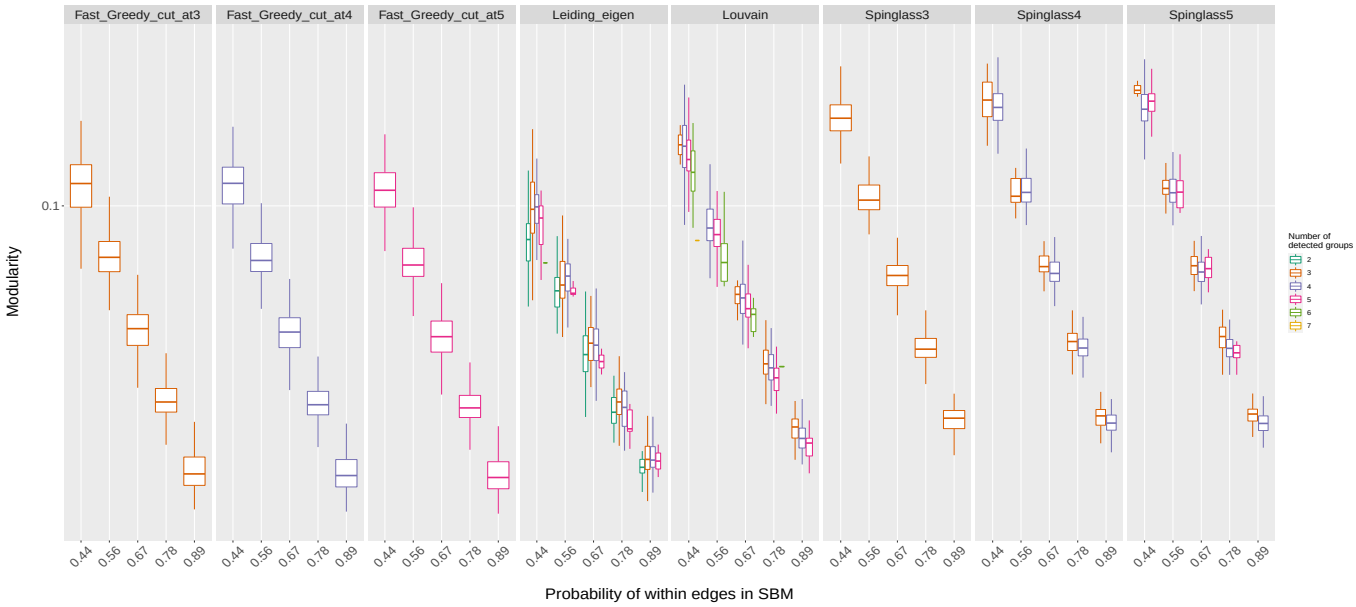


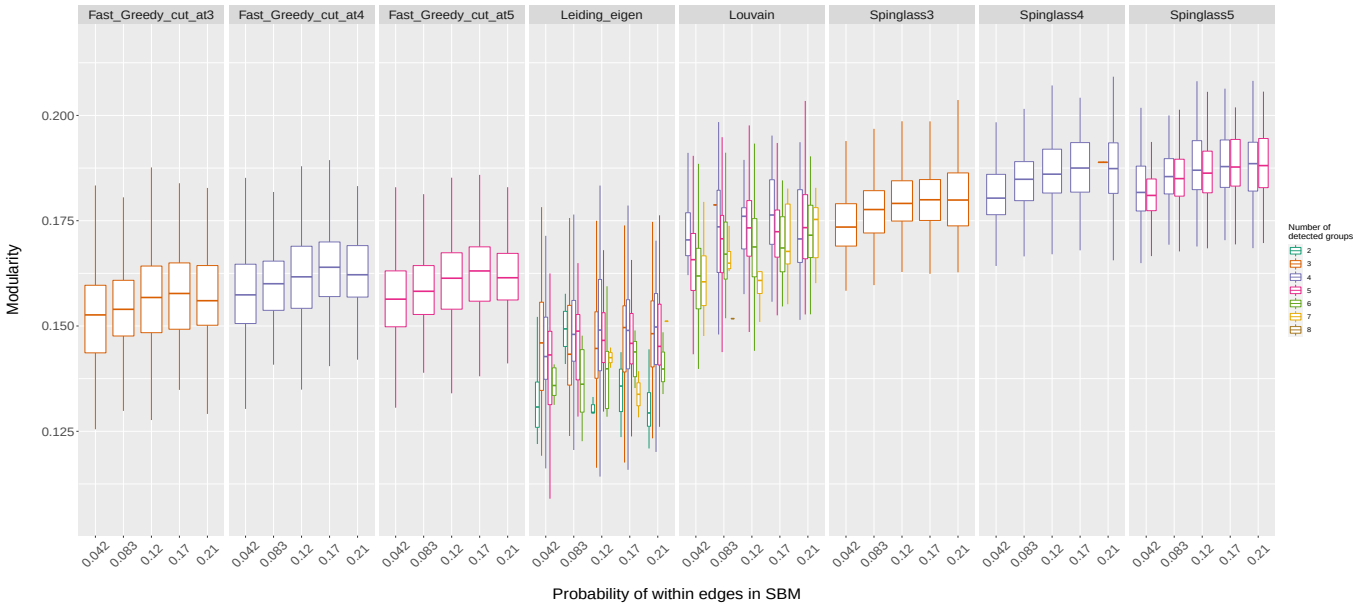
Figure 5: Analysis of clustering across methods in the case of SBM networks with 3 heterogeneous groups size (53/5/5)



3.3.3 Short tests on a disassortative model

We will now look briefly at how our methods behave when we generate our graphs with a disassortative model, i.e. with a within edge probability smaller than the between edge probability ($\theta_{in} < \theta_{out}$). On Fig.6, that for a disassortative model, our methods detect only insignificant group structures with a modularity between 0.125 and 0.2. Moreover, when studying in more detail the group structures detected by our methods, we observe that the size of the detected groups are of the same order as the size of the underlying groups. Even when the methods do not detect the right number of groups, the size of the groups remains homogeneous. In view of these few observations, we are led to believe that our methods manage to detect the underlying groups, but that since these groups have been generated by a disassortative model, their group structure is less significant than in the assortative case.

Figure 6: Analysis of clustering across methods in the case of SBM networks with 4 dissasortive groups size (15/16/16/16)



3.4 Discussion of the methods on some real EEG networks

Based on our previous tests, the Louvain and Spinglass methods tend to stand out for their efficiency and robustness. Moreover, in their paper Z. Yang, R. Algesheimer and C. J. Tessone [13] performed a comparative analysis of community detection algorithms on artificial networks, in order to determine which methods has the highest accuracy. After conducting their analysis, they also came to the conclusion that the Louvain (Multi-level) and Spinglass methods were, in general, the most efficient in detecting communities. In order not to spread ourselves too thinly in the rest of our analysis, we will focus on the use of these two methods, which have provided the best results so far.

The table below shows the first results we obtained for the networks we have at our disposal. Because of the way the networks were obtained from the EEG recordings, some networks are composed of several connected components. When a network is not connected, it is always composed of a large connected component with between 1 and 3 nodes that are isolated from this main connected component. As the only way to assign a group to an isolated sensor is to assign a group to it alone, we have chosen to focus only on the detection of communities in the main connected component. In the table below, we present the modularity of the community structures detected in the main connected component of each network, as well as the number of communities detected and the number of nodes and sides in this component. On the basis of [11], it as be shown that in practice, a modularity of more than 0.3 is a good indicator of significant community structure in a network. The only networks that seem to have a significant community structure are the networks Network160 and Network187.

Table 1: Clustering method comparison

Clustering method	Modularity	Number of communities detected	Number of vertices/edges
Network 159			
Spinglass	0.1413183	3	59/483
Louvain	0.126399	4	59/483
Network 160			
Spinglass	0.3055287	4	62/323
Louvain	0.3040142	4	62/323
Network 165			
Spinglass	0.2706869	4	61/368
Louvain	0.2697712	4	61/368
Network 172			
Spinglass	0.201875	4	61/431
Louvain	0.1904275	4	61/431
Network 173			
Spinglass	0.2159149	4	62/435
Louvain	0.2061646	4	62/435
Network 181			
Spinglass	0.2305017	5	61/331
Louvain	0.199446	4	61/331
Network 184			
Spinglass	0.1583153	4	59/429
Louvain	0.1427916	4	59/429
Network 187			
Spinglass	0.3206509	3	61/260
Louvain	0.3142382	4	61/260

Moreover, we have noticed that the only networks where the presence of communities on the plots of these networks can be well distinguished are the networks Network160 and Network187 which have a modularity greater than 0.3.

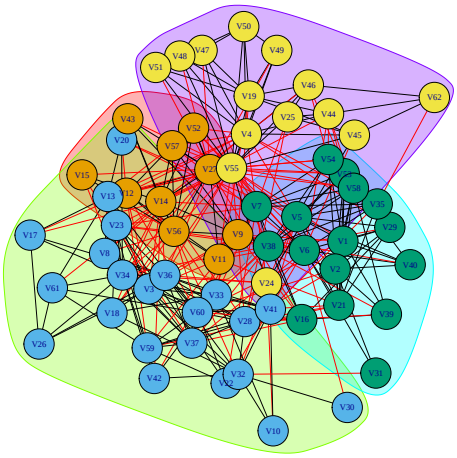


Figure 7: Network160 using Louvain method

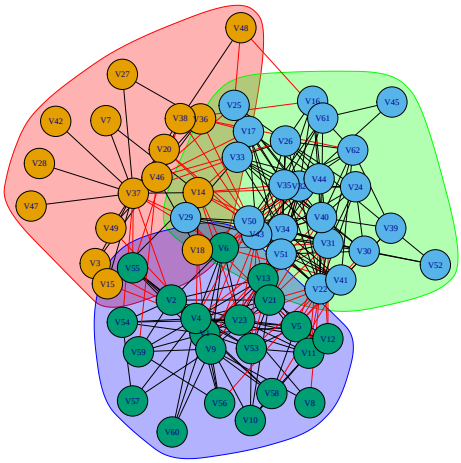


Figure 8: Network187 using Spinglass method

On the other networks, the plots are not very helpful for the visualisation of the detected communities, as we see on the plot below.

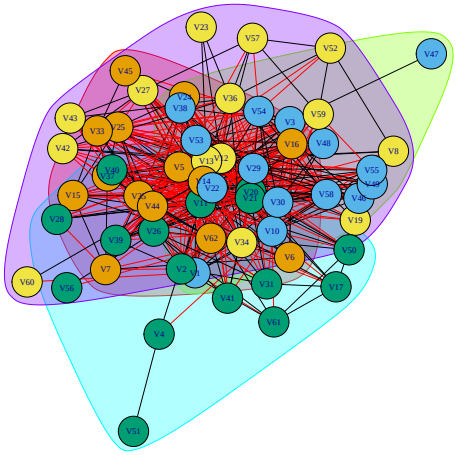


Figure 9: Network159 using Louvain method

4 Dispersion and spatial cohesion of communities

In the remainder of this project, we would like to develop a generalized linear model to explore the predictive capabilities of EEG recordings from comatose patients. In this way, we hope to find a meaningful alternative measure to standard clinical tests for inferring the potential recovery or not of a comatose patient in the first days of coma. For this model, our basic idea is to consider modularity as being related to the dispersion of communities around their centroid, i.e. the spatial cohesion of communities. In order to explore this possibility, we will in this section analyse the dispersion of communities around their centroid.

4.1 Spatial dispersion

Here we will discuss the most consistent way of calculating the dispersion of sensors in a community around their average position. First of all, we have to specify that we have normalised the position of the sensors so that they are located on the unit sphere. This assumption on the normalized position of the sensors will greatly simplify our next calculations, without being unreasonable given that the sensors were already almost all on the same sphere.

First, we need a way to calculate the average position of the sensors on the unit sphere. Calculating the centroid of a group of points is done by solving an optimisation problem :

$$\arg\min_{\mu\in\mathbb{S}^2} \sum_{i=1}^{n} d(\mathbf{x},\mu)^2 \tag{22}$$

where $d(\cdot,\cdot)$ is a distance to be defined and $\mathbb{S}^2$ is the unit sphere in $\mathbb{R}^3$. A naive way to proceed would be to consider the Euclidean distance :

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{3}(\mathbf{x}_i-\mathbf{y}_i)^2} \tag{23}$$

And then approximate the optimal solution by projecting the solution of the following problem onto the unit sphere :

$$\mu(\mathbf{x_1},\dots,\mathbf{x_n}) = \arg\min_{\mu\in\mathbb{R}^3} \sum_{i=1}^{n} d_E(\mathbf{x},\mu)^2 = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{24}$$

However, given the high symmetry of the sensor positions and the linearity in $\mathbf{x}_1,\dots,\mathbf{x}_n$ of $\mu(\mathbf{x}_1,\dots,\mathbf{x}_n)$, this method is not very robust because, for example if we obtain groups with a high symmetry with respect to the origin, we risk obtaining an average position close to the origin, which would make the projection of this position on the unit sphere not very significant.

The problems associated with this first method led us to question our approach. Until now we had treated the problem in a mechanical and naive way, we wanted to calculate the centroid of a group on a sphere, without even using the specific and well known geometry of the sphere. With this in mind, we reformulated our basic problem to use the geometry of the sphere. To do this, we redefined the distance between two points on the unit sphere as the geodesic distance between these points :

$$d_S(\mathbf{x},\mathbf{y}) = \arccos(\mathbf{x}\cdot\mathbf{y}) \tag{25}$$

With this new distance, the problem (22) then takes the form :

$$\mu(\mathbf{x_1},\dots,\mathbf{x_n}) = \arg\min_{\mu\in\mathbb{S}^2} \sum_{i=1}^{n} d_S(\mathbf{x},\mu)^2 = \arg\min_{\mu\in\mathbb{S}^2} \sum_{i=1}^{n} \arccos(\mathbf{x},\mu)^2 \tag{26}$$

However, even if considering the geodesic distance on the sphere seems much more natural in our situation, the problem still becomes much more complicated to solve. Indeed, unlike problem Eq.(24), problem Eq.(26) does not have an explicit form for its solution. In order to solve it, we must therefore use numerical analysis methods. Samuel R. Boss and Jay P. Fillmore have treated this problem in their article [14], in which they propose two algorithms that converge to a single optimal solution, under the condition that all points lie on the same hemisphere. We could not find these algorithms already implemented on R, and importing or rewriting them on R would take too much time in this project. We were therefore unable to use these algorithms to solve our optimisation problem. However, the article [14] brings us an important result for the continuation. Indeed, Buss and Fillmore show that if we restrict the points $\mathbf{x}_1,\dots,\mathbf{x}_n$ to a single hemisphere of $\mathbb{S}^2$ then our problem Eq.(26) has a unique solution, under the only hypothesis that there is at least one point $\mathbf{x}_i$ which is not on the edge of the hemisphere (which is a relatively weak hypothesis in our case). This is rather a good sign of the existence of a unique solution to our problem because the sensors are almost all in the upper hemisphere of the unit sphere, and if we also take into account that the sensors will be divided into

several groups then we have strong reason to think that our problem is well posed.
Finally, we found a method already implemented on R which allows us to calculate the centroid of a group of points on the sphere, it is called *surfacecentroid* and comes from the icosa package. Unfortunately, we have not managed to find a detailed description of this method. We only know that this function implements great circle calculation to infer on the place of the centroid. Furthermore, we were able to test this method in some cases that could prove to be pathological, but it provided correct results for these different tests. It is therefore reasonable to assume that this method is efficient and correct.

Now that we have an efficient way of calculating the centroid of a group of points, we need a way of calculating σ^2, the dispersion of the different communities around their centroids. Again, there would be many ways to define σ^2, but if we are using the geometry of the sphere again, a fairly natural way to define it would be :

$$\sigma^2 = \frac{1}{n} \sum_g \frac{1}{n_g} \sum_{i \in C_g} d_S(\mathbf{x}_i, \mu_g)^2 \tag{27}$$

where the first sum is over all communities g,n is the number of communities, $n_g = |C_g|$ the size of community g, C_g the set of all nodes in community g, $d_S(\cdot, \cdot)$ is the geodesic distance on a unite sphere as define above and μ_g is the spherical centroid of the community g.

4.2 Study of the dispersion of randomly generated communities

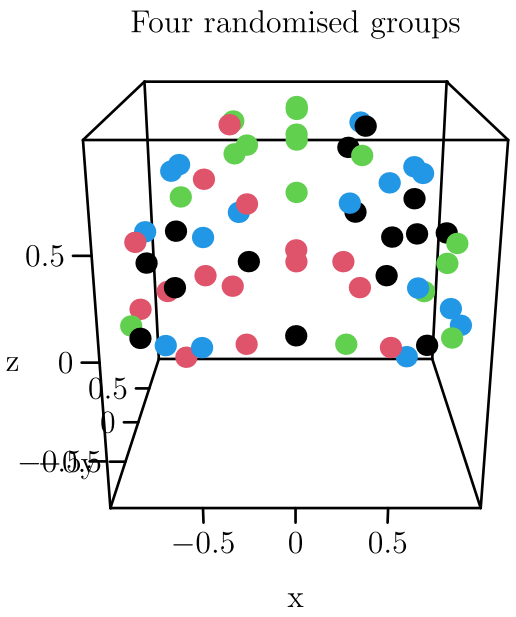
In this section, we will generate communities of homogeneous size in a random way, and then we will look at the dispersion of these communities around their centroids. This will allow us to have an order of magnitude of a potential "upper bound" for the dispersion of sensor groups around their centroid, which will allow us to know if communities detected in our networks show some spatial cohesion or on the contrary if the detected communities seem to be randomly distributed. For this we considered the fixed positions of 62 sensors, and assigned them a group at random, and repeated this procedure for different numbers of groups, namely 3, 4 and 5. Given the homogeneous group sizes that we could observe in our networks on which we work, for each number of groups we divided the 62 sensors into groups of homogeneous size, namely (20,21,21) for 3 groups, (15,15,16,16) for 4 groups and (12,12,12,13,13) for 5 groups.

In the following table, we see the values of the empirical means and variances that we obtain after having randomly generated a thousand distributions of the sensors in 3, 4 and 5 groups and then having calculated their dispersion around their centroid. We observe that the average dispersion seems to decrease slightly when the number of groups increases, and we see that the variance of the observed dispersion is very small for each number of groups. In view of these results, it would seem reasonable to think that a dispersion close to 1.5 would be an indication of the absence of spatial cohesion of the communities detected.

Table 2: Mean values and variances of the dispersion of communities around their centroid

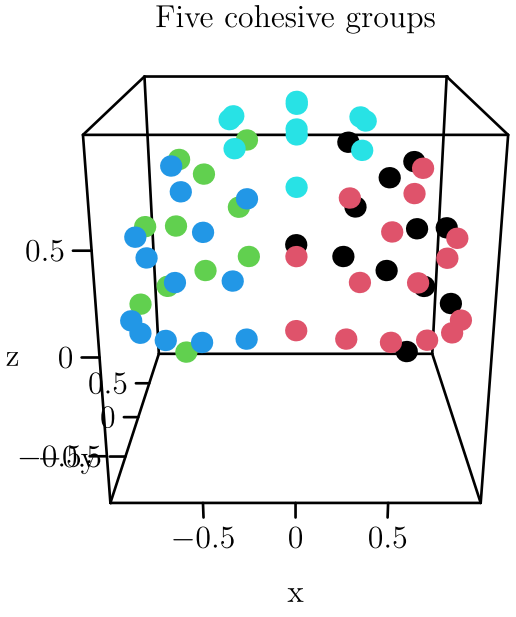
Number of groups	Mean of the observed dispersion	Variance of the observed dispersion
3	1.557195	0.0031273
4	1.517012	0.004570265
5	1.489634	0.005441371

In the interactive 3D graph below, we can see a random distribution of the sensors into 4 groups of uniform size.



4.3 Study of the spatial dispersion of highly cohesive communities

We are now interested in the dispersion of groups of sensors around their centroids, having previously "designed" these groups so that they have a high spatial cohesion. In contrast to the previous section, this will allow us to have an order of magnitude for a potential "lower bound" for the dispersion of sensor groups around their centroids, which will allow us to detect networks with a community division having a strong spatial cohesion. Here, we will also consider three different cases with respectively three, four and five groups of uniform size (we will also add the values of the dispersion for a division into 1 and 2 cohesive groups, without commenting too much on them since our clustering methods almost systematically detect a number of communities in 3 and 5). In the figure below, we see an example of the distribution of sensors in five uniform groups (for the distribution in three and four groups see Appendices 3 and 4).



In the table below, we see the values of the dispersion for a division of the sensors into 1,2,3,4 and 5 highly cohesive groups. In the case of 4 and 5 groups, we see that the dispersion is much lower than in the case of randomly generated communities. For the case of three groups, the dispersion is about double that for 4 and 5 groups, and it is a little less than half that calculated for randomly generated

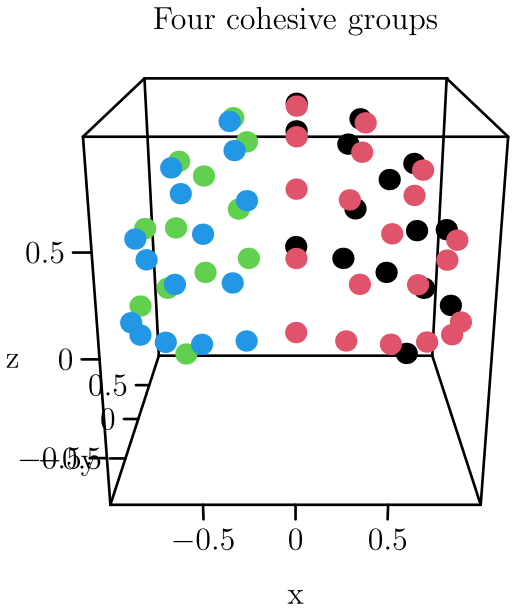
groups. This is because even if we consider three groups with high spatial cohesion, as the positions of the sensors are fixed the groups necessarily have a higher dispersion when they are few in number. It is therefore not surprising that the dispersion decreases as the number of groups increases.

Table 3: Dispersion of communities around their centroid on some highly spatially cohesive communities

Number of cohesive communities	Dispersion
1	1.625651
2	0.7996573
3	0.6645357
4	0.3377237
5	0.2635111

4.4 Spatial dispersion of detected communities on network generated with SBM

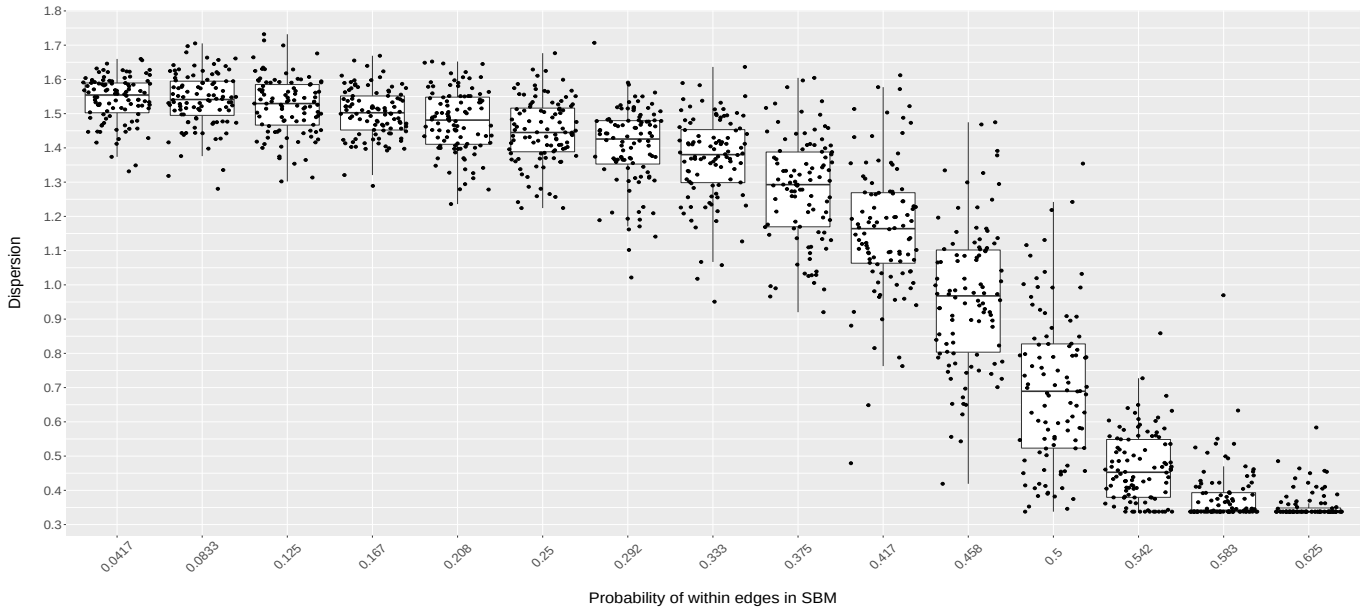
In this section, we will test the Louvain method by randomly generating networks using the stochastic block model (SBM) as we did in the section 3.3. However, this time we will generate networks with a stochastic block model composed of four communities of sizes 15, 20, 12 and 15 respectively. We have chosen this distribution of nodes so that we can match the spatial location of the nodes of each group with the position of the four groups we "created" in the previous section so that they have a strong spatial cohesion. By proceeding in this way we will be able to see how the different methods succeed in detecting the underlying structure of the randomly generated networks, moreover we will see how the spatial cohesion of the detected structures evolves as a function of the probability of within edge in the generated networks (we recall that in section 3.3, we generate networks with probability of within edge p_{in} and probability of between edge given by $p_{out} = \frac{1-p_{in}}{n_{grp}-1}$ where n_{grp} is the number of groups we want to generate, here $n_{grp} = 4$).
The 3D graph below shows the distribution and location of the four groups on which our stochastic block model is based:



In the following figure, we plot the calculated dispersion of communities around their centroid for a large number of networks generated with the stochastic block model we described earlier for different values of p_{in}, and where the communities are given by the Louvain method. We see that for $p_{in} < \frac{1}{3}$ the average value of the calculated dispersion is very close to 1.5, which as we saw in the subsection 4.2 is about what we obtain when we randomly assign groups to the nodes of the network. Then, the dispersion only decreases when p_{in} increases. For $p_{in} \geq 0.625$, the dispersion stabilises around the

value 0.3377237, which is the value of the structure underlying the SBM and which we calculated in the previous section.

Figure 10: Spatial dispersion of communities around their centroid for networks generated with SBM



4.5 Spatial dispersion of communities on some reals world networks

Now that we have tested our method on randomly generated or self-created communities with high spatial cohesion, we will test our method on some real networks, which come from EEGs of real patients who have recovered from their comas. Based on the previous results, we now have an idea of the value of the dispersion of communities around their centroid, for communities with high spatial cohesion ($\sigma^2 \propto 0.5$) and for communities with low spatial cohesion ($\sigma^2 \propto 1.5$). In the two tables below we have reported the values of the modularity and the dispersion of the communities around their centroid that we obtained, when we used respectively the method of Louvain and that of Spinglass to identify these communities.

If we compare the values of the dispersion we obtain for these networks with the values we found in the subsections 4.3 and 4.2, we observe that the Network160 and Network165 networks have a spatial dispersion of their communities around their centroids which is very close to that obtained for randomly generated groups, and this with both the Louvain and Spinglass methods. This would translate into the absence of spatial cohesion in the communities detected for these two networks. Similarly, if we observe the values obtained for the networks Network172 and Network173 (in the case of the Louvain method), we see that even if the values indicate a more important spatial cohesion, we remain far from values indicating a significant spatial cohesion.

These tables also give us an insight into the behaviour of the modularity as a function of the dispersion of a network. Even if we cannot draw global conclusions with a small sample of networks such as the one considered here, it is nevertheless disturbing to observe that dispersion tends to decrease when modularity decreases. Indeed, we rather expected to see dispersion increase when the community structure of the network increased, and therefore when modularity increased.

Table 4: Modularity and dispersion of some networks (Clustering with Louvain method)

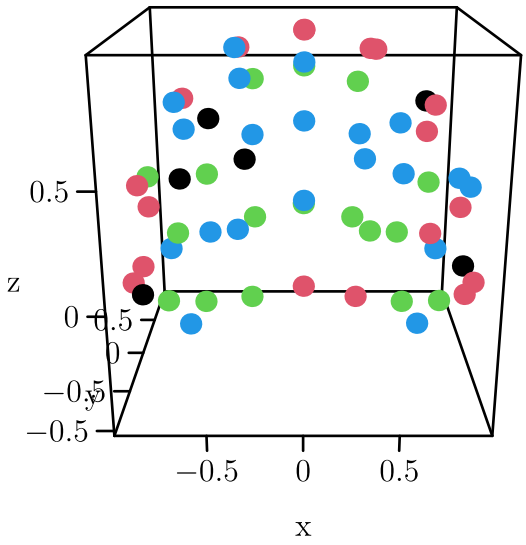
Name of the network	Modularity	Dispersion	Number of group
Network160	0.3040142	1.47847	4
Network165	0.2697712	1.536399	4
Network172	0.1904275	1.194577	4
Network173	0.2061646	1.106855	4

Table 5: Modularity and dispersion of some networks (Clustering with Spinglass method)

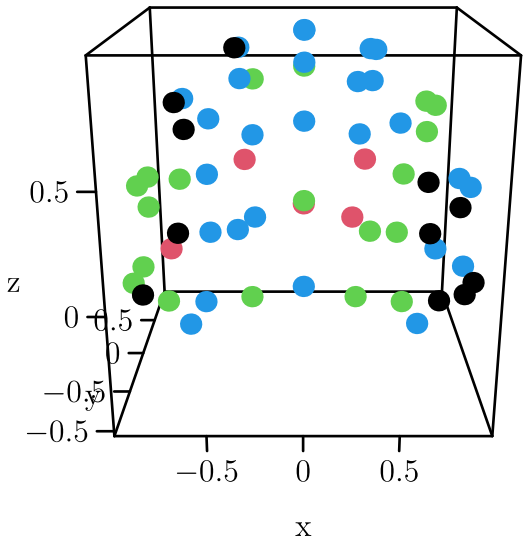
Name of the network	Modularity	Dispersion	Number of group
Network160	0.3049536	1.481206	4
Network165	0.2706869	1.525914	4
Network172	0.201875	1.265443	4
Network173	0.2115471	1.334773	4

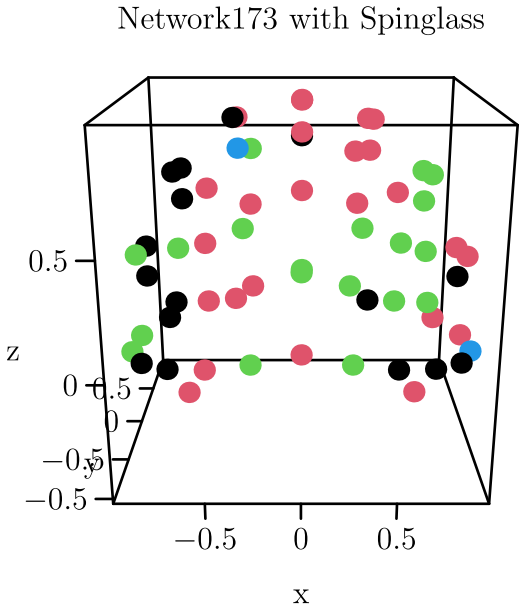
On the interactive 3D graphs below, we can observe a little more visually the spatial distribution of the communities detected on the networks Network165, Network172 and Network173 for the Louvain or Spinglass methods. Informally, we see that on these networks the communities appear to have some spatial cohesion, either by being separated into small cohesive groups, or by being spatially united into one cohesive but spatially spread out group.

Network165 with Louvain



Network173 with Louvain





5 Discussion of the results

In this paper, we studied the accuracy and robustness of some methods of detecting communities in networks that we generated artificially so as to have knowledge of the presence or not of real communities underlying the generation of these networks. When the networks were randomly generated without any underlying structure, no significant presence of communities was detected, as long as the density of sides in the network was not too low. Conversely, when the networks were generated with an underlying community structure, we found, for communities whose node distribution was not extremely heterogeneous, that the more the underlying structure had a marked community structure ($p_{in} >> p_{out}$ in the stochastic block model), the more significant the structure detected was. The methods we consider, therefore, appear to be to be relatively accurate and robust. Among these methods, the methods of Louvain and Spinglass seem to be more efficient than the others, both by their robustness and by their ability to detect the true underlying structure in the case of the SBM. When applying these methods to a few networks created from the records of patients with a favourable outcome, we obtain structures whose modularity is generally close to being significant. Interpretation of these results is therefore complicated, as they could either imply the absence of an underlying structure with a low density of sides, or the presence of an underlying structure with a probability of sides within a group slightly greater than the probability of sides between groups. To overcome this interpretation problem, it would first be interesting to increase the number of networks (from patients with a favourable outcome) taken into account, in order to have a more robust interpretation than the one that can be made on a small number of networks. Furthermore, it would also be interesting to contrast these results with those obtained for networks from patients with an unfavourable outcome. In this way, we would have a better idea of whether or not there is a real difference in compartmentalisation between the networks of patients with a favourable outcome and those with an unfavourable outcome.

Secondly, we investigated the spatial cohesion of sensor communities detected with the Louvain and Spinglass methods. First, we computed orders of magnitude for an upper bound and a lower bound on the dispersion of sensor clusters around their centroids, for different numbers of clusters. The upper bound was obtained by randomly assigning communities to the sensors, and the lower bounds were calculated by artificially creating groups with strong spatial cohesion. This allowed us to have an approximate scale, to better interpret the spatial cohesion of the EEG network communities. The communities detected by the Louvain and Spinglass methods tend, in the few networks at our disposal, to have a dispersion around their centroids that seems to be more that of randomly generated groups than communities with strong spatial cohesion. However, by visually inspecting the spatial distribution of sensor communities, there appears to be some cohesion of these communities, but the structure of these communities appears to be either spatially divided into small cohesive groups, or to be spatially

united into one cohesive but spatially spread out group. However, the interpretation of these results is also limited, as we have few networks to increase the significance of these results. It would also be interesting to compare these results with those obtained for patients with an unfavourable outcome, to see if a distinct pattern emerges between these two categories of patients.

During this project, we have found some potential directions that would be interesting to explore in the future of this project. First of all, as we presented in section 2, modularity maximization is related to a special case of the likelihood maximization method. More precisely, the special case of the likelihood maximization method is equivalent to the generalized modularity maximization (9). This equivalence is a potential direction, which it would be interesting to exploit in the future development of this project, where instead of considering the modularity defined by the equation ((2)) we could use the general form, (9), of it with γ as in ((19)). By adopting this approach, we could then use the algorithm described at the end of the article, [5], to be able to iteratively approximate the parameters ω_{in} and ω_{out}. Another point that could be developed in an extension of this paper would be the assumption of assortativity of our networks. As we saw in the section 2.3, this would put our modularity maximization approach in default. Finally, another point that would be interesting to develop is the calculation of the dispersion of sensor communities around their centroids. Indeed, in our calculations to find the centroid of a group of sensors and then in the calculation of the dispersion of these sensors around their centroid, we assign the same weight to each sensor. This implies that each sensor has the same importance in our calculations, but in doing so we do not take into account the network structure that exists between the sensors. In a community, it would seem natural to give less weight to a sensor that has a minor role in the community and more weight to a sensor that has a central role in the community structure. One possibility would be to assign a weight to each sensor according to its degree of connection with the other sensors in that community. Thus, to calculate the centroids of communities, the algorithms described in the article by S. Buss and J. Fillmore [14] would be ideal.

Appendices

Appendix A Other plots on Erdös-Rényi graphs

Figure 11: Analysis of the leading eigen method with box-plot in the case of ER(n,p) networks

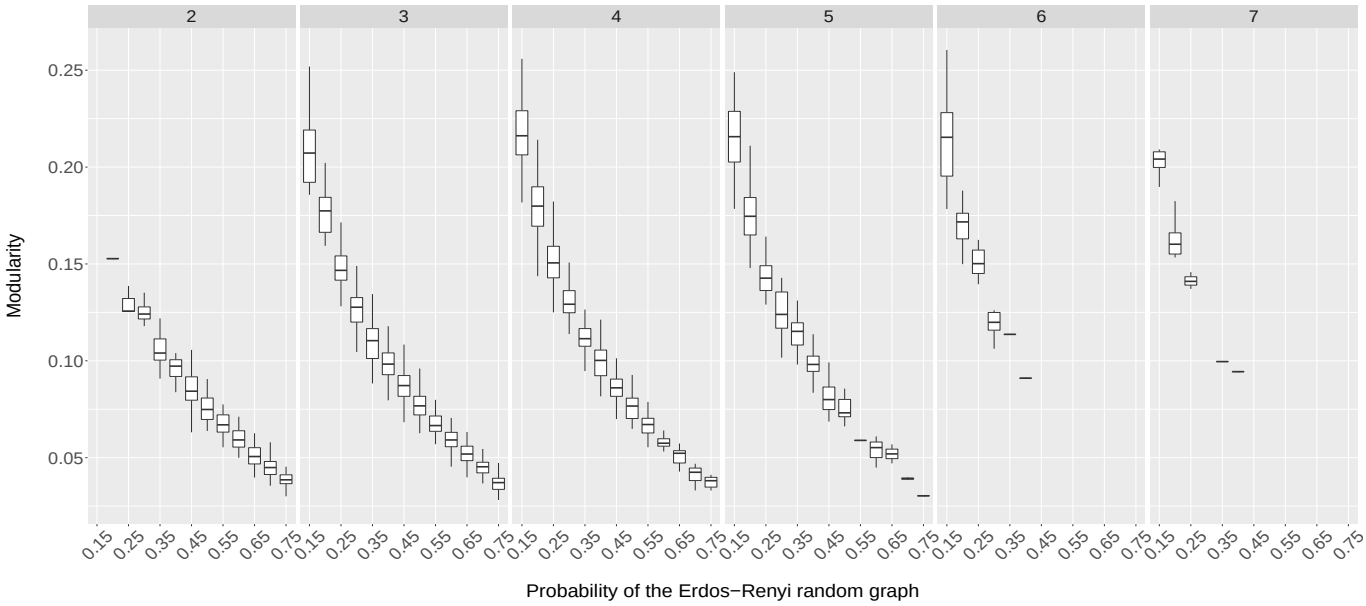


Figure 12: Analysis of the Louvain method with box-plot in the case of ER(n,p) networks

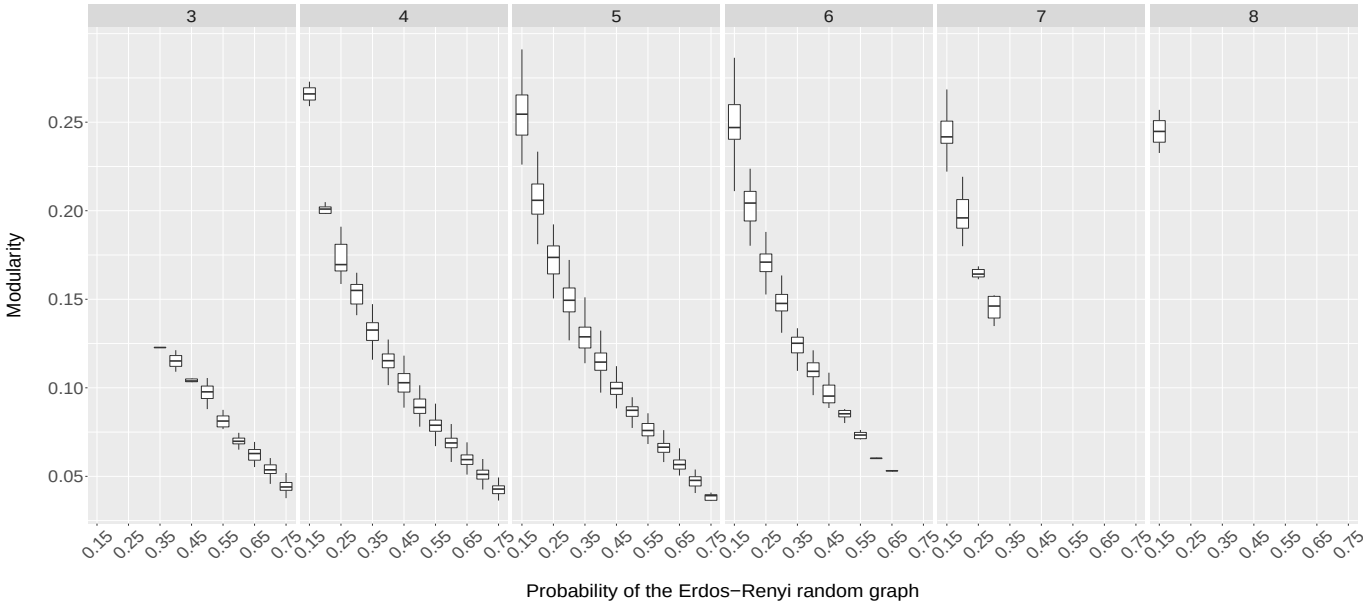
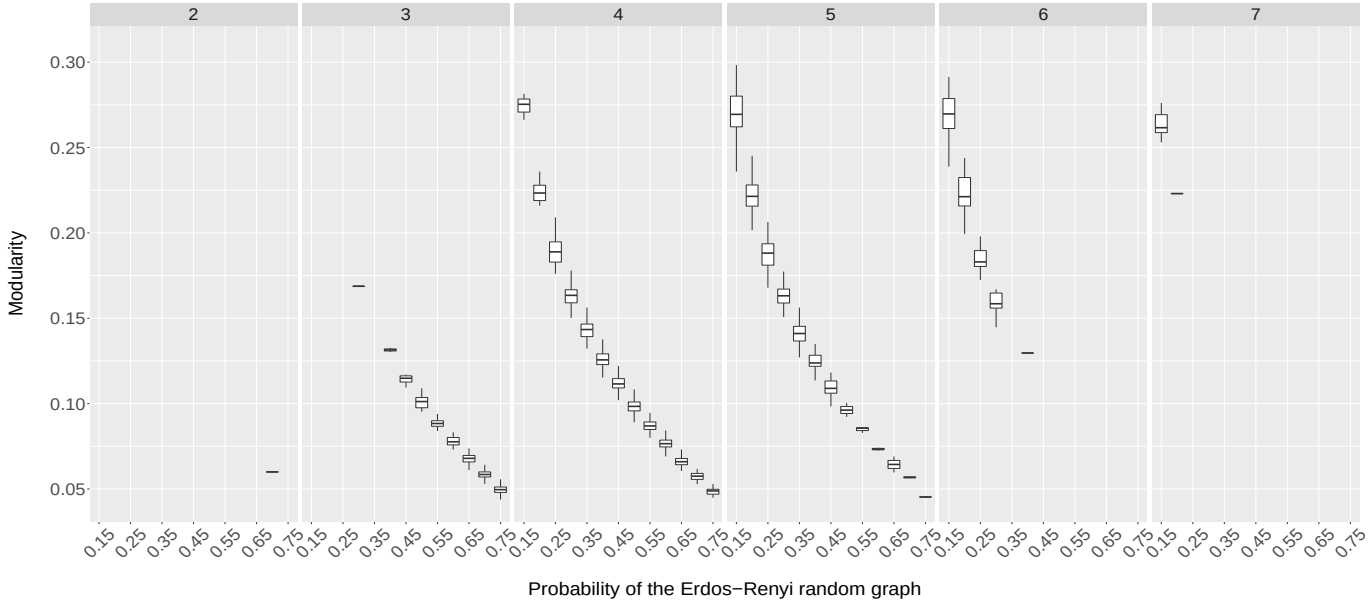


Figure 13: Analysis of the Spinglass method with box-plot in the case of ER(n,p) networks



Appendix B Other plots of Stochastic Blocks Models graphs

Figure 14: Analysis of clustering across methods in the case of SBM networks with 4 uniform groups (15/16/16/16)

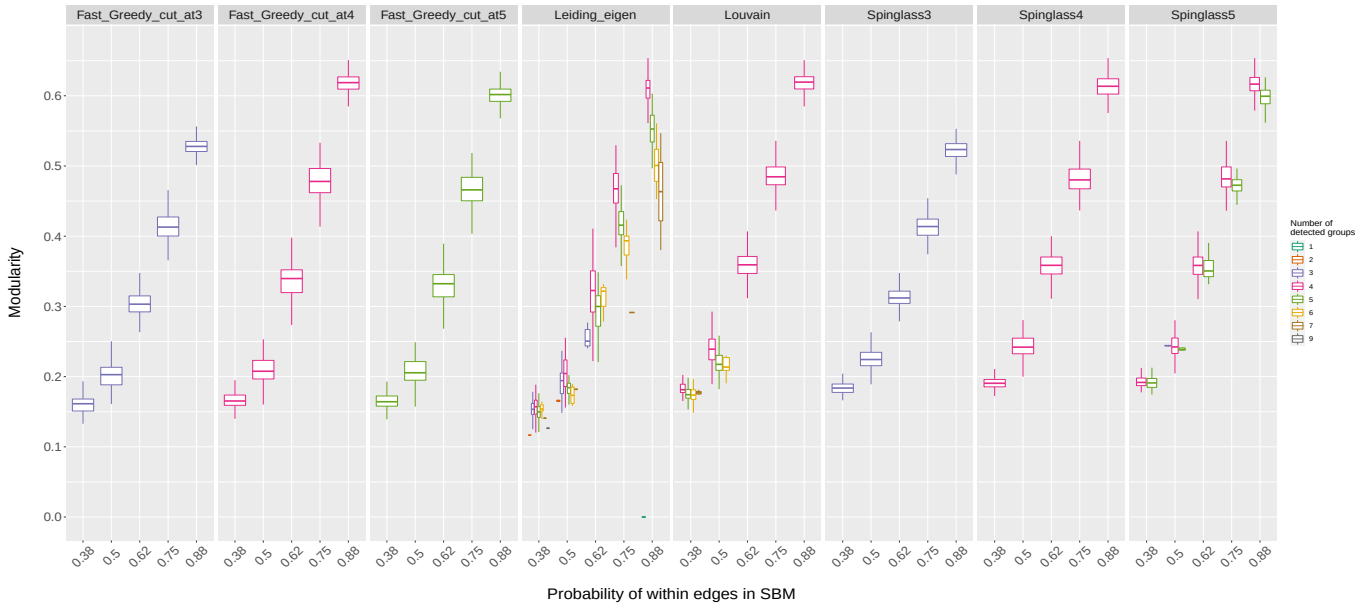


Figure 15: Analysis of clustering across methods in the case of SBM networks with 5 uniform groups (13/13/13/12/12)

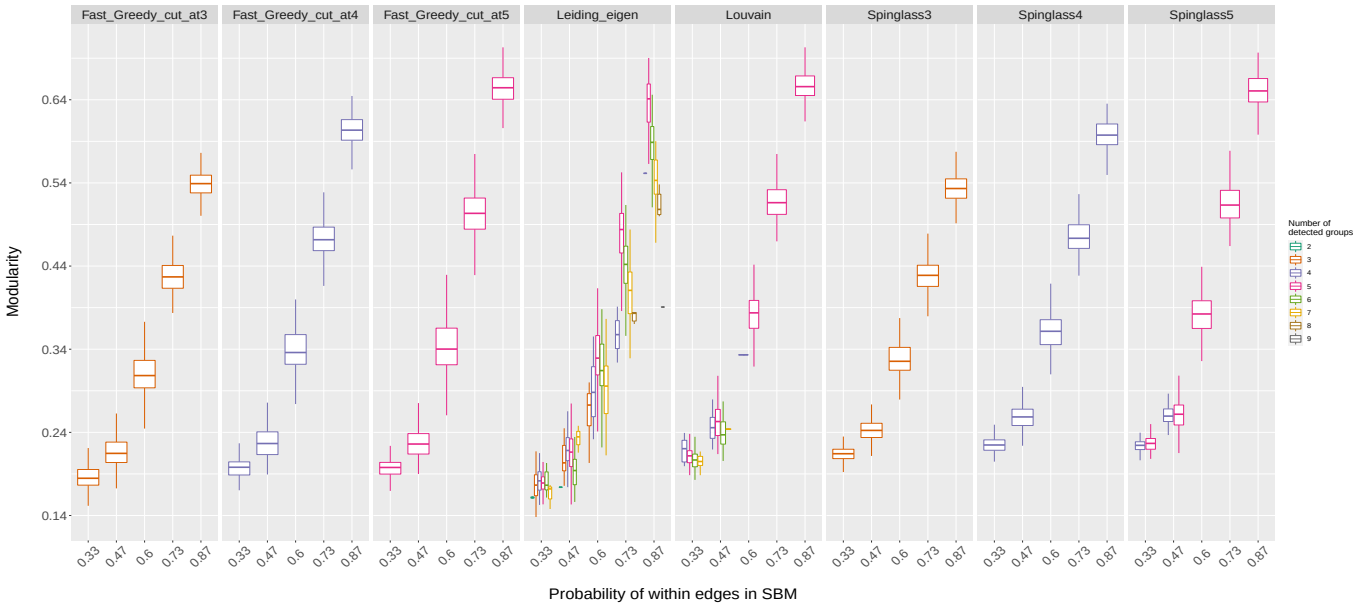


Figure 16: Analysis of clustering across methods in the case of SBM networks with 3 heterogeneous groups size (53/5/5)

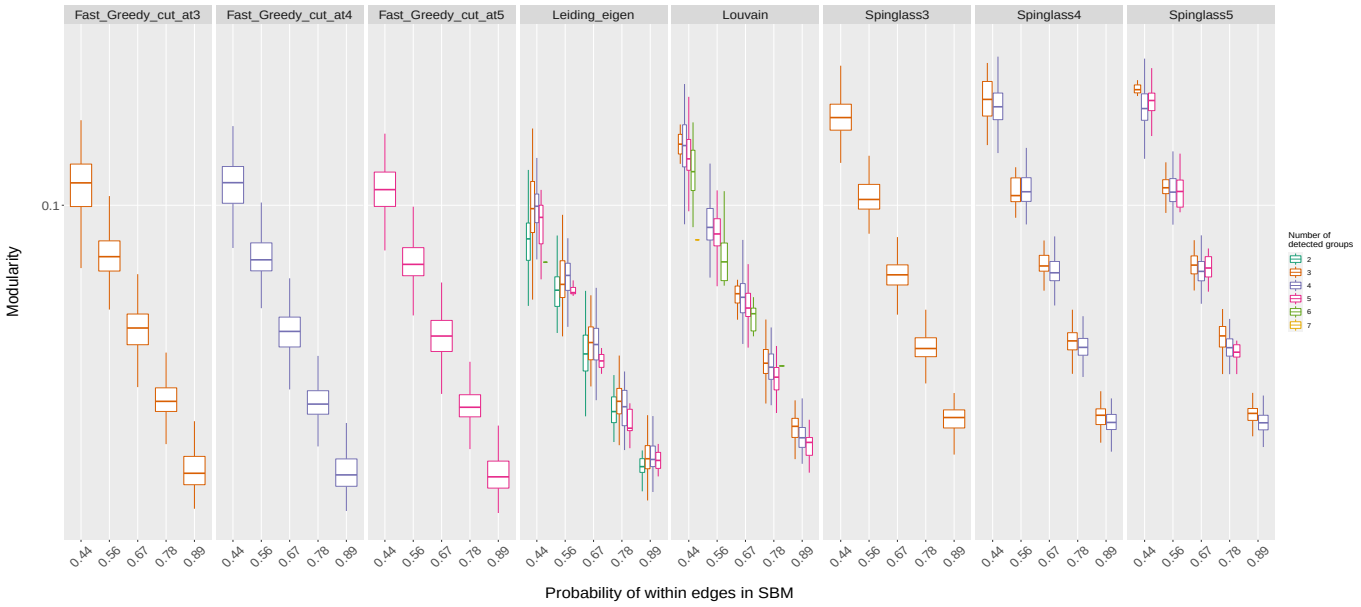
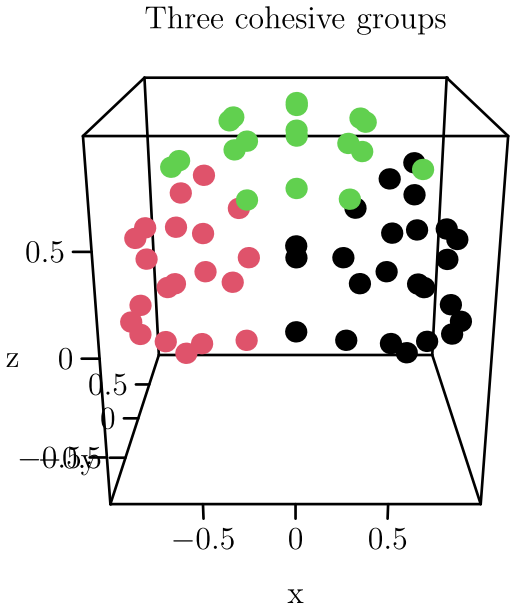
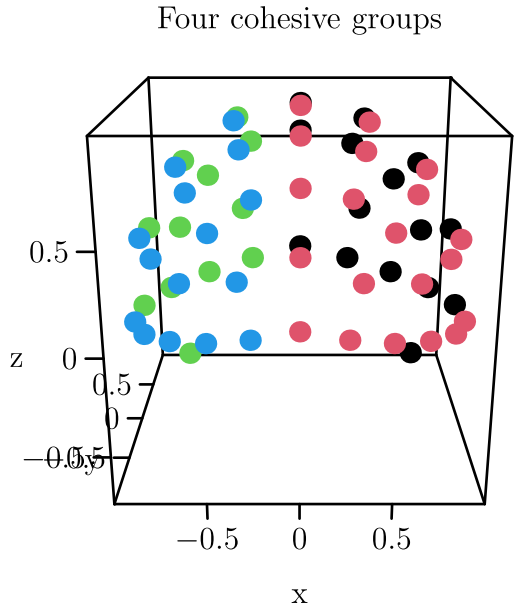


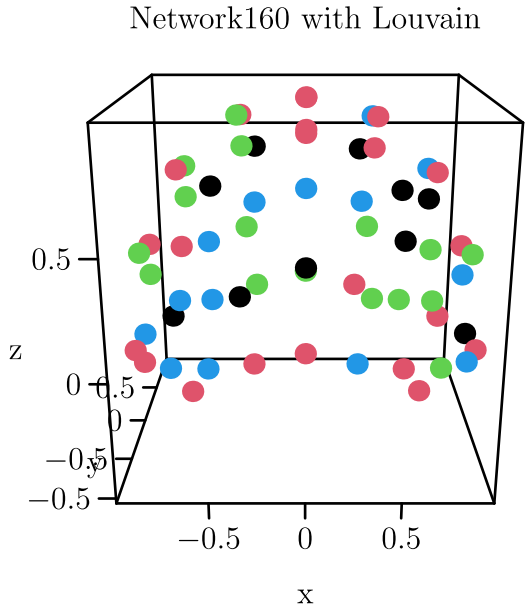
Figure 17: Analysis of clustering across methods in the case of SBM networks with 5 heterogeneous groups size (25/25/5/4/4)

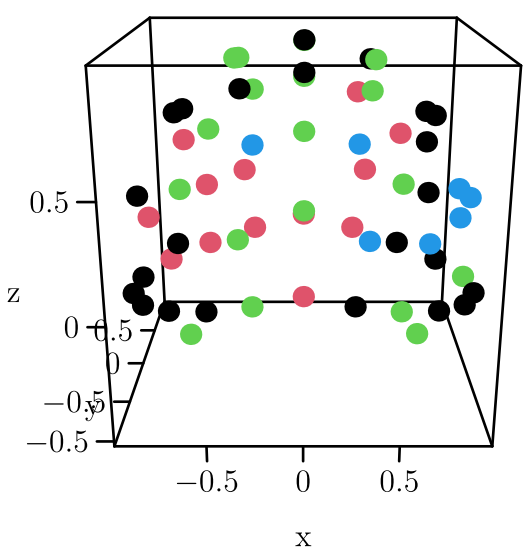
Appendix C Other 3D graphs of cohesive group





Appendix D Other 3D graphs of real networks




Network172 with Louvain
0.5
z
0
−0.5
0.5
0
−0.5
y
−0.5
0
0.5
x

References

[1] Thomas Kustermann, Nathalie Ata Nguepnjo Nguissi, Christian Pfeiffer, Matthias Haenggi, Rebekka Kurmann, Frédéric Zubler, Mauro Oddo, Andrea O. Rossetti, and Marzia De Lucia. Brain functional connectivity during the first day of coma reflects long-term outcome. *NeuroImage: Clinical*, 27:102295, 2020.

[2] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), Sep 2006.

[3] M. E. J. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, 2012.

[4] Christian Hennig. What are the true clusters?, 2015.

[5] M. E. J. Newman. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. *CoRR*, abs/1606.02319, 2016.

[6] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, Dec 2006.

[7] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1), Jul 2006.

[8] Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1), Jan 2011.

[9] Tiago P. Peixoto. Efficient monte carlo and greedy heuristic for the inference of stochastic block models. *Physical Review E*, 89(1), Jan 2014.

[10] M. E. J. Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4), Oct 2013.

[11] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6), Dec 2004.

[12] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, Oct 2008.

[13] Zhao Yang, René Algesheimer, and Claudio J. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6(1), Aug 2016.

[14] S. Buss and Jay P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graph.*, 20:95–126, 2001.