Network Analysis of Separated Cognitive States in the Human Brain

Candidate Number: 267458

March 24th, 2014

Abstract

Network science is an insightful approach to studying data gained from fMRI scans. In this thesis we attempt to see if it is appropriate to make a distinction between "Task-positive" (TP) and "Task-negative" (TN) networks based on these scans, where the former is associated with the subjects' brain states when performing cognitive tasks and the latter when not performing cognitive tasks. The data used to study these networks was fMRI data parcellated to involve 600 regions of the brain across 94 human subjects. We use task-positive and task-negative region assignments that already exist in literature. We examine simple network diagnostics. We then study two more complex methods; spectral clustering and multilayer community detection. We find inconclusive results based on the simple network diagnostics, but promising results in spectral clustering. We generalise the already existing concept of flexibility and find that attention tasks generally have higher average flexibility in the task-negative network. We also find a correlation between ordinal and categorical flexibility.

Contents

1 Introduction

1.1 Motivation

A *network* is a set of *nodes* connected by *edges*, and network science is the study of these networks. This thesis will involve an investigation, from a network science perspective, into two specific networks of brain regions, which are associated with different types of brain activity - the "Task-positive network" and the "Task-negative network." The idea of these networks dates back to 1929 when Hans Berger, the inventor of the electroencephalogram, proposed the idea that the brain is continually active, even in *resting state,* that is, the state of the brain when it is not performing an explicit task. His ideas were not taken seriously by neurologists; the general perception at this point was that the brain (or at least a portion of it) only became active when performing an explicit task [37, 43]. At the very end of the 20th century, research by Shulman et al. identified a brain network, the *default mode network*, associated with task-induced deactivation or, more simply, a brain network in which activity in resting state differs from activity when performing an explicit task [40]. This was supported by Raichle et al. in 2001 [38].

In this thesis, we focus on the two terms "Task-positive network" and "Task-negative network" which were introduced by Fox et al. in 2005 [15]. These networks are composed of regions known to become more active (task-positive), and regions known to become less active (task-negative) during the task relative to their behavior in resting state. The activity in these networks have been shown to underpin attention [31] and memory [25, 19] processes. Furthermore, disruptions to these networks have been observed in certain neurological disorders [18, 5, 24, 27]. In relation to these networks, several questions have been posed [20]:

- Do the strength of the connections within and between these networks vary systematically across resting state and task-driven cognitive states?
- How does the anatomical organisation of the brain affect this variation?
- Does the relationship between anatomical structure and brain function shape behaviour?

These questions motivate rather specialised research which would span several disciplines, and because of this we are not able to answer them in this thesis. However we will use a variety of methods from network science to examine these task-positive and task-negative networks in order to provide a solid mathematical basis from which such research could benefit. This is motivated by several recent criticisms on the origin of task-positive and task-negative networks. R Nathan Spreng notes the following in his opinion article [42]:

- Recent studies [45, 34] show that the task-positive network actually consists of at least two functionally and anatomically distinct networks which serve different roles in cognition.

- The task-negative network is often engaged during goal-directed cognition but depends on the nature of the task, hence the "task-negative" label is not entirely appropriate.
- Recent work [42] demonstrates that the components of these networks interact differently with one another depending on the task at hand. Such interactions challenge the veracity and utility of a "task-positive network versus task-negative network" comparison.

Although R Nathan Spreng writes many criticisms about the classification of task-positive and task-negative networks, his judgment is based upon the definition of these networks and not their existence. As we will see later, the assignment of these task-positive and task-negative nodes (the nodes which together make up the task-positive and task-negative networks) seems rather specific, and because the human brain is an extremely complex organ, these assignments could interact with subject variability. The methods for defining these networks are not entirely rigorous [15] and would benefit from more research, especially from the perspective of network science. Again, although we will not be directly answering the questions posed, we take a basic approach and study these networks from the ground up. At the same time, wherever possible, we will attempt to relate these results to the questions posed by other authors.

1.2 Network Science

Network science is an extremely useful tool that allows mathematicians and scientists alike to model complex organisation structures from a mathematically sound perspective. In particular, it is well suited to the study of neuroscientific data [30, 6, 7]. On a microscopic level the structure of the brain is built from *neurons* which are connected physically via *synapses*. This automatically gives rise to a mathematical network. If we were to model the brain this way, we would have access to an enormous amount of information about both the anatomical structure and function of the brain. However, the human brain is estimated to have approximately 10^{11} neurons and 10^{14} synapses [46]. This causes an immediate problem as neuroscientists do not yet have the methods to physically identify such a large number of entities, and computers are currently not powerful enough to be able to process this data. The only species to have been modeled this way is *Caenorhabditis elegans*, which has around 300 neurons and 7,600 synaptic edges [7].

We can however produce brain networks by a wide a variety of methods, which are able to approximate the underlying system by measuring activity from an area inside the brain. The type of brain activity is dependent on the method. These methods include functional magnetic resonance imaging (fMRI), electroencephalography (EEG), and magnetoencephalography (MEG), single-photon emission computed tomography (SPECT) and positron emission tomography (PET). The choice of method is typically dependent on what types of property one wishes to study. For example, EEG recording has the advantages of cost, lack of dependency on subject health and responsiveness in measuring the brain's reactions to stimuli, especially when compared to, say, fMRI. However, the data gleaned from fMRI analysis is of higher spatial resolution and, most importantly, the area in the brain from which a signal oringiates can be identified with greater precision [41]. This renders the data from fMRI analysis particularly suitable for this thesis, as the physical location of the task-positive and task-negative nodes occupy very specific areas of the brain and we are required to identify them. For this reason, it is fMRI data that we will use in order to explore the "Task-positive network (TPN)" versus "Task-negative network (TNN)" debate discussed in Section 1.1.

1.3 The Data

The data involves parcellated time series of fMRI blood oxygen level-dependent (BOLD) signals recorded from 95 subjects performing one of three tasks: *(i)* an attention task, *(ii)* a memory task and *(iii)* a resting state task [20]. Despite predictions that resting-state neural activity would appear noisy and uncontrolled, the human brain exhibits patterns of correlated neural activity even when the subject has not been instructed to perform any task [48, 14, 38]. Such correlations in this resting state are thought to support the functional organization of the brain [14]. The resting state will be compared to the other tasks as if it were a directed task itself[1]. In prior studies, the data has been "parcellated" which involved selecting regions by subdividing the Automated Anatomical Labeling (AAL) Atlas into 600 regions of similar volume [44]. This is an important step, as this step directly affects the size and nature of the networks which will be produced in Section 2.2, and different parcelations can reveal different structures in the brain [8]. After this step, we have a $600 \times T$ matrix of values for each subject, where T is the number of time points, giving the BOLD signal for each of these 600 ROIs (regions of interest) across a series of time points. There are 240 time points for the attention tasks and 140 in the resting state, but there is no available time series for the memory tasks, as this data has already been pre-processed with wavelet correlation at scale 2 (this will be outlined in Section 2.2). The data includes two versions of the same attention task for each subject and both versions will be included in all the discussions that follow. We will name these two versions "Attention task 1" and "Attention task 2." We expect that we will find similar results in these two versions. There are also two types of memory task; a "face memory" task and a "word memory" task. We will refer to these as "Memory task 1" and "Memory task 2." Again, we will include both types of memory task in all calculations. The data includes the co-ordinates of each of the 600 regions (see 2.2).

Each of the 600 ROIs has been assigned exactly one of the following labels: *(i)* "Task-positive," (TP) *(ii)* "Task-negative," (TN) and *(iii)* "Other." As mentioned in Section 1.1, Fox et al. identified certain non-overlapping areas of the brain which were labeled "Task-positive" or "Task-negative". The co-ordinates of each of our 600 regions have been compared to co-ordinates of these TP and TN regions and were assigned accordingly. There are 368 such regions, 240 task-positive and 128 task-negative. The remaining 232 regions are assigned the "Other" label.

2 Constructing the Networks

2.1 Formal Definition of a Network

The simplest type of network is an undirected and unweighted graph. An unweighted graph will be referred to as a *binary* graph. Such a simple graph consists of a set of entities called *nodes* (or ROIs) that are connected to each other by an *edge*. An edge between nodes i and j will be denoted by (i, j). The number of nodes will always be denoted by N, and a network with N nodes will be said to have *size* N. An unweighted network can be represented by an $N \times N$ adjacency matrix $\mathbf{A}$, where

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge between } i \text{ and } j \\ 0, & \text{otherwise.} \end{cases}$$

[1] However, although we say we will be comparing the *tasks*, we will in fact be comparing the condition of the brain of the subjects during these tasks. It is important to make this distinction.

In this thesis we will usually study weighted networks, an example of which is shown in Fig. 1. A weighted network has its "topology" given by an unweighted adjacency matrix and its "geometry" given by a weighted adjacency matrix. A weighted adjacency matrix is defined in the same way, except that we now denote it by $\mathbf{W}$, and W_{ij} represents the strength of the edge connecting i and j. In the brain networks we will be studying, we will take $W_{ij} \in [0, 1]$. The procedure for obtaining these matrices is detailed in Section 2.2. It will be assumed that all networks in this thesis are undirected, hence $\mathbf{W}$ will be symmetric, i.e. $\mathbf{W^T} = \mathbf{W}$, and hence $\mathbf{W}$ will have real eigenvalues [33].

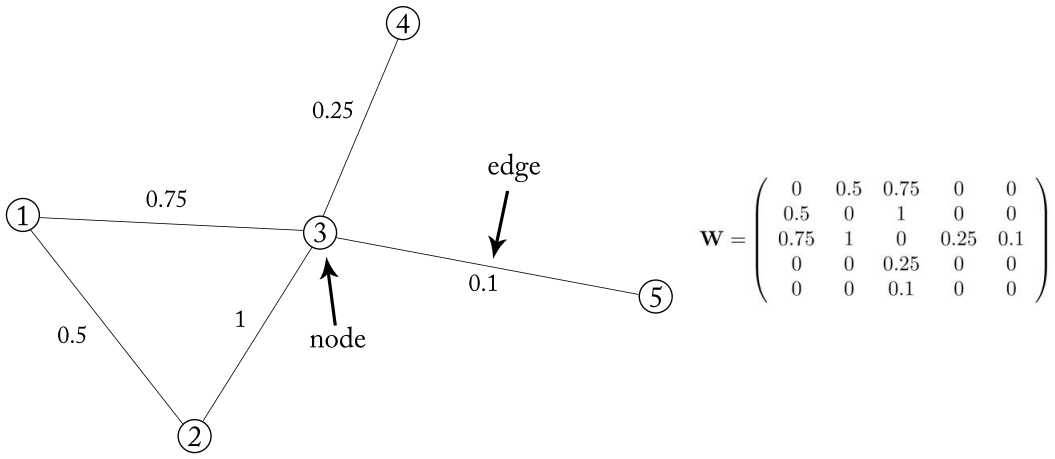

Figure 1: An example of an undirected, weighted network with its respective adjacency matrix. The values along the edges correspond to the weights.

2.2 Obtaining The Adjacency Matrices from the Data

Before investigating the networks, we need to create them. The steps we will be discussing are shown in Fig. 2. As outlined in Section 1.3, the data was previously "parcellated" by selecting regions. The regions were selected by subdividing the Automated Anatomical Labeling (AAL) Atlas into 600 regions of similar volume [44]. This is the first step in Fig. 2; it identifies 600 ROIs which will constitute the nodes of our network. Each node i therefore has a location in the brain specified as a vector (x_i, y_i, z_i). This location is the same across all subjects, and adjustment has been made for differently sized brains. The form of the data at this stage is a 600×240 matrix (where 240 is the number of time points[2]) giving the BOLD signal for each node over a series of time points. This matrix is visualised in Fig. 3.

The second step in Fig. 2 is to estimate a similarity between each pair of nodes. We can choose to create a single network over the entire time interval. However, if we want to study how the network evolves over the time period, we will have to "chop" the time interval into a specified number of windows, and then we will estimate an adjacency matrix for each time window. These windows can either be overlapping or non-overlapping. Our time windows to be non-overlapping for simplicity. If the time interval chosen is too short, we are more prone to noisy data skewing the similarity between the nodes. However if it is too long, we will not have sufficiently many adjacency matrices to study

[2]There are 140 for the resting state case.

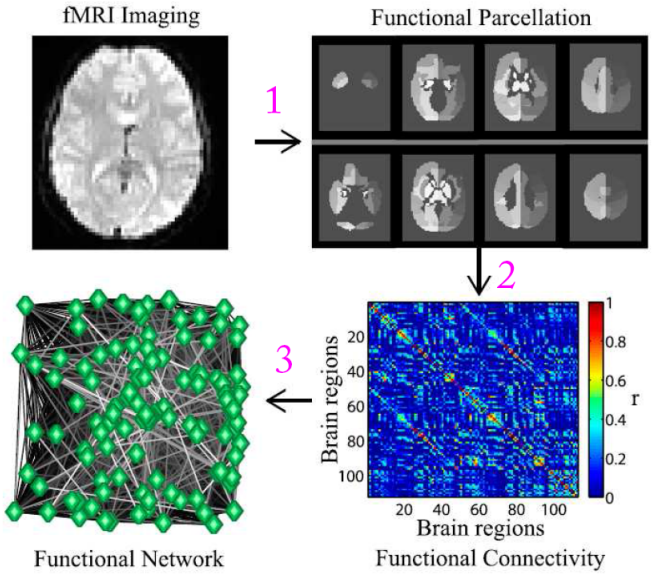

Figure 2: The three steps from raw fMRI data to the adjacency matrices. The first step, the parcellation, was done for us. There are many ways to perform the second step, which is calculating the weighted adjacency matrices (in the picture, blue represents weaker connections while red represents stronger). We use wavelet correlation at scale 2, which has been successful in extracting important information in other studies with similar data [29] and also used in a study on the same data [20]. The final step is thresholding, which is outlined in Section 2.3. This figure is from Ref. [3] and is used with permission.

the structure of the network over the time period [13]. Hence we will show results for three different choices of time windows: 24, 8, and 4 time windows of size 10, 30, and 60 time points respectively. These time windows are visualised in Fig. 3.

We use wavelet correlation at scale 2 (0.06–0.125Hz) to estimate similarity between each pair of ROIs. This uses the partial maximal overlap discrete wavelet transform, and then the correlations are estimated by calculating the correlation coefficient $r_{ij} \in [0, 1]$ of each pair of nodes i and j. We use the WMTSA Wavelet Toolkit for MATLAB[3].

For the two attention tasks and the resting state task[4], we run this method over the entire time interval for each subject individually. We remove all data for subject 56 from all 5 tasks due to a missing signal from a node. We are left with five sets of ninety-four 600×600 fully weighted, fully connected matrices giving a correlation coefficient representing the weight of the edge (i, j) between each pair of nodes i and j. The matrices produced in MATLAB are not actually symmetric at machine precision. The values r_{ij} and r_{ji} usually differ by a value of less than 10^{-16}. This causes complications with some programs in MATLAB, hence we force the matrices to be symmetric by taking the upper triangular part $\mathbf{U_W}$ of the correlation matrix $\mathbf{W}$ and setting $\mathbf{W} = (\mathbf{U_W^T} + \mathbf{U_W})/2$. Because of the

[3] HTTP://WWW.ATMOS.WASHINGTON.EDU/~WMTSA/

[4] Except the memory tasks, since the memory data has already been processed with wavelet correlation at scale 2 over the whole time interval.

nature of the wavelet correlation method, we are left with negative correlations for a small portion of the edges. There are usually very few negative values and this is typical of wavelet correlation processed fMRI data [29]. The large prevalence of positive correlations is due to a "global signal" from the blood flow in the subject's neck. Negative correlations could be due to the pooling of deoxyhaemoglobin in the veins carrying oxygen away from the brain [40].

There are a few ways to factor in these negative values. One option is to set $W_{ij} := |r_{ij}|$. This appears to be a natural choice, though we can no longer make a distinction between the negative correlations with equally strong positive correlations. The method we will use is to set $W_{ij} := (r_{ij} + 1)/2$. This has the advantage of still making a distinction between negative and positive correlations[5]. We also remove self-edges (i.e. we set $W_{ii} = 0$ for all nodes i).

To produce adjacency matrices over the time series, we follow a similar process as above except we apply the wavelet correlation method to each time window instead. We will only perform this process for attention task 1.

We now have our weighted adjacency matrices. The third step in Fig. 2, thresholding, is outlined in Section 2.3 and we will perform this later when required.

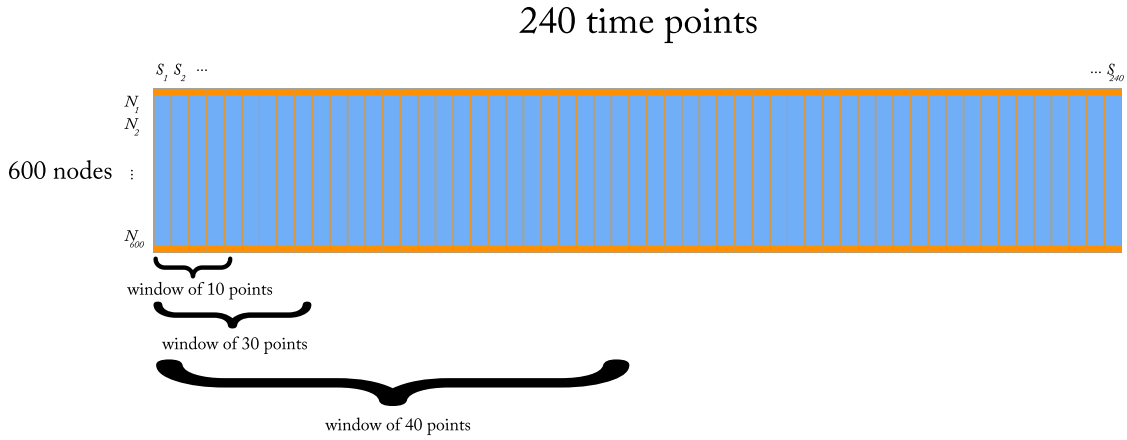

Figure 3: A visualisation for the 600×240 BOLD signal matrix. For practical purposes there are fewer than 240 signal bars shown in this picture. The 600×1 signal vector for node i is represented by S_i. Three lengths of the first window are shown below the matrix. For example, if we choose our window size to be 10, we use wavelet correlation to compute an adjacency matrix first on $[S_1, S_2, \ldots, S_{10}]$ which will become our first adjacency matrix. The second will be computed from $[S_{11},, \ldots, S_{20}]$ and so on. Note that the brackets are not to scale and do not correspond exactly to the number of signal vectors.

[5]There is also the option of using wavelet coherence instead of wavelet correlation to produce the correlation coefficients, as negative values are filtered out. We do not follow this method here due not having the time series data for the memory tasks.

2.3 Thresholding

In most cases we will be dealing with fully connected (i.e, an edge exists between each pair of distinct nodes), weighted networks. However it will be sometimes necessary to study unweighted networks. One reason may be that we need to extract topological information from certain ranges of weights. Another case arises when analysing certain results created from the weighted matrices, in that the results could be somewhat dependent on the average strength of the network, hence studying the unweighted case removes this possibility. Thresholding is a process which transforms our weighted networks into binary ones.

We will use a common type of thresholding [28] . It involves choosing a single threshold percentage parameter $p \in [0, 1]$, and given a weighted network $\mathbf{W}$ with size N, we calculate pN. In most cases pN will not be an integer, so we take $\lceil pN \rceil$. We then order the edges by weight and set the strongest $\lceil pN \rceil$ edges to 1 and set all others to 0. It is also possible to produce a weighted version of this type of thresholding, which is simply just setting a certain percentage of the edge weights to 0. However in most cases in this thesis, when using thresholding we will have an active interest in the binary matrices as we will be wanting to remove the possibility of higher connectivity contributing bias towards results.

Varying p produces different binary networks. A disadvantage of this type of thresholding is that it based on the assumption that edges with weights below τ are significantly different to those above. Therefore, the choice of p is an important one but is usually chosen arbitrarily, and when this type of thresholding is used we will show results over a range of p.

2.4 Constructing The TPN and TNN Sub-Networks

The next step is to define our *sub-networks*. A sub-network of a network $\mathbf{W}$ is a network $\mathbf{W_S}$ whose nodes are a subset of the nodes of $\mathbf{W}$. As stated in Section 1.3, every region is assigned exactly one of: *(i)* TP, *(ii)* TN, or *(iii)* Other. We will consider six sub-networks. We keep their associated edges the same but we if remove a node in the process, we also remove all its associated edges. We will define them in a similar way to other research on this data [20]. We create the networks by deleting the asociated edges and nodes from the original 600×600 adjacency matrices. We will name them the following:

1. Task-positive network (TPP-N),
2. Task-negative network (TNN-N),
3. "Other" network (TOO-N),
4. Task-positive-negative network (TPN-N),
5. Task-positive-other network (TPO-N),
6. Task-negative-other network (TNO-N).

These sub-networks are visualised in Fig. 4. To produce sub-network (1), we remove all of the non task-positive nodes from the original 600×600 adjacency matrices and all of the edges associated with the respective nodes. There are $128 + 232 = 360$ such nodes, and therefore we are left with a fully connected (except on self-edges; these values are 0), weighted 240×240 adjacency matrix. This network is denoted by "TPP", because we are preserving only the connections between task-positive nodes and other task-positive nodes. Cases (2) and (3) are analogous. The TNN-N has size 128 and the TOO-N has size 232.

Cases (4), (5), and (6) are constructed in a similar, but not identical, way. In sub-network (4), we remove all the "other" nodes, and all of their associated edges. We are left only with the edges connecting task-positive and task-negative nodes to task-positive and task-negative nodes. We are left with a fully connected, weighted 368×368 adjacency matrix. Again, the other two cases are analogous, the TPO-N is of size 472 and the TNO-N is of size 360.

We apply the process above to all 94 subjects across both versions of the attention task, the resting state task and both memory tasks. Table 1 shows all the combinations of sub-networks and their associated dimensions. Figure 4 shows a visualisation of the different connections in each of the sub-networks.

We note that some of the network diagnostics that we will compute (see Sections 3 and 4.0.5) will scale with network size. We therefore have to be careful when making comparisons between columns in Table 1, as the sub-networks have different sizes. Comparing rows causes no immediate complications.

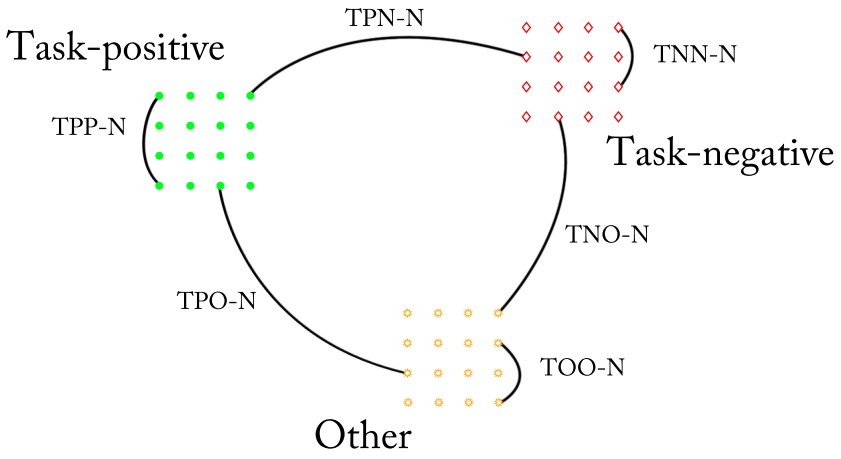

Figure 4: A visualisation of the connections in each of the sub-networks defined in Section 2.4. This figure shows the three types of nodes, with a typical edge in each of the sub-networks. Note that the TPO, TNO and TPN networks also include the connections in the TPP, TNN and TOO networks respectively.

	PP-N	NN-N	OO-N
Attention task 1	240×240	128×128	232×232
Attention task 2	240×240	128×128	232×232
Resting state	240×240	128×128	232×232
Memory face	240×240	128×128	232×232
Memory word	240×240	128×128	232×232
	PN-N	PO-N	NO-N
Attention task 1	368×368	472×472	360×360
Attention task 2	368×368	472×472	360×360
Resting state	368×368	472×472	360×360
Memory face	368×368	472×472	360×360
Memory word	368×368	472×472	360×360

Table 1: The six possible sub-networks across all five tasks. We expect to see similar results in attention tasks 1 and 2, because they represent two versions of the same task. We also expect to see similar results in both memory tasks, because although the tasks themselves are different, it has been demonstrated that they activate similar areas of the brain [17].

3 Simple Network Diagnostics

Here we will explore some simple network diagnostics and their results after applying them to our networks. Recall the size of a network $\mathbf{W}$ is denoted N.

3.0.1 Connectivity

The *degree* of a node i is the number of edges associated with it, and is denoted k_i. In our fully connected adjacency matrices, $k_i = N - 1$ for all nodes i. The *strength* of a node i is given by the column sum (or row sum, because our adjacency matrices are symmetric) of the adjacency matrix:

$$S_i = \sum_{j=1}^{N} W_{ij}.$$

The *normalised strength* is

$$S_i' = \frac{1}{N} \sum_{j=1}^{N} W_{ij}. \tag{1}$$

This gives the possibility to compare connectivity between columns in Table 1. The *connectivity* of a network $\mathbf{W}$ is the average of the normalised strength of all nodes i [29], and it will be denoted by $\text{Connec}_{\mathbf{W}}$.

We also define the *density* of a network $\mathbf{W}$ [33] to be:

$$\text{density}_{\mathbf{W}} = \frac{\text{number of edges in } \mathbf{W} \text{that exist}}{\text{total possible number of edges}}.$$

We will also study the average strength of the task-positive, task-negative and "other" nodes. This is different to studying the connectivity of the respective sub-network since we consider the weights of all edges associated with the respective node label.

3.0.2 Clustering coefficient

The *local clustering coefficient* for a node i in a network with a binary adjacency matrix $\mathbf{A}$ is

$$C_i^{\text{bin}} = \frac{1}{k_i(k_i-1)} \sum_{j,m} A_{jm}A_{ji}A_{mi}. \tag{2}$$

This formula counts the number of "closed triangles" for which node i participates in, divided by the maximum number of possible "closed triangles". More precisely, for distinct nodes i, j, and k, if (i,j) and (j,k) exist, then we can form a path between i and k, we will choose the notation $[ijk]$ to denote this. If (i,k) also exists, we can form the path $[ijki]$ which forms a closed triangle. Averaging over all nodes i, the *global clustering coefficient* $C_{\mathbf{A}}^{\text{bin}}$ of a network with binary adjacency matrix $\mathbf{A}$ is the mean of C_i^{bin} over all nodes i.

There are numerous generalisations for the clustering coefficient of a weighted network $\mathbf{W}$ [39]. We use the one defined by Onnela et al. [35]. The local clustering coefficient for a node i in a weighed network $\mathbf{W}$ is

$$C_i = \frac{1}{k_i(k_i-1)} \sum_{j,k} (\hat{W}_{ij}\hat{W}_{ik}\hat{W}_{jk})^{1/3}, \tag{3}$$

where

$$\hat{W}_{ij} = \frac{W_{ij}}{\max(\mathbf{W})}$$

and the global clustering coefficient $C_{\mathbf{W}}$ is the mean over C_i over all nodes i.

3.0.3 Centrality

The centrality of a vertex measures the relative importance within the network. There are several types of centrality measures [12] but we will focus on two of them.

The *geodesic betweenness centrality* considers the shortest paths within a network, known as the *geodesic paths*. We define geodesic betweenness centrality only for a binary network $\mathbf{A}$, because geodesic betweenness centrality is not a suitable diagnostic for a fully connected weighted network, for reasons which we will explore after the definition.

A geodesic path γ_{ij} between nodes i and j is the path of shortest length. We define σ_{ij} to be the number of geodesic paths between distinct nodes i and j. We also define $\sigma_{ij}(k)$ to be the of geodesic paths between distinct nodes i and j that contain node k. We then define geodesic betweenness centrality for a node i in a binary network $\mathbf{A}$ as

$$B_i = \sum_{i,j,k} \frac{\sigma_{jk}(i)}{\sigma_{jk}}, \tag{4}$$

where i, j, k are distinct nodes in $\mathbf{A}$. The geodesic betweenness centrality $B_{\mathbf{A}}$ is the average of B_i over all nodes i.

There are several complications which occur when applying this definition to a weighted network. One of them is that the notion of a geodesic path between nodes i and j for a fully connected network $\mathbf{W}$ becomes meaningless; all nodes are connected to all other nodes hence the geodesic path length between all nodes will be equal to 1. We would therefore have to use the weight information to define the *length* between two nodes and use an algorithm to find the shortest path. Another complication is that nodes with very high strengths typically have short edges to *all* other nodes and will therefore participate in most of the shortest paths in a network. Hence except for a very small set of nodes, a node i in the network will typically have $B_i \approx 0$. We therefore will use the thresholding technique defined in Section 2.3 when computing geodesic betweenness centrality.

Eigenvector centrality is another type of centrality measure and it is defined recursively. We assign a relative score to a node based on the relative score of the nodes in which it is connected to. Thus for a weighted network $\mathbf{W}$ (in the binary case we can replace $\mathbf{W}$ by $\mathbf{A}$) we define the eigenvector centrality $e(i)$ of a node i to be proportional to the sum of the eigenvector centralities of the nodes connected to it:

$$e(i) = \frac{1}{\lambda} \sum_{j \in M(i)} e(j) = \frac{1}{\lambda} \sum_{j} W_{ij} e(j)$$

where $M(i)$ is the set of nodes that i is directly connected to and λ is a non-zero constant. Using the notation $\mathbf{e} = (e(i))_{1 \leq i \leq N}$ we rearrange to obtain

$$\mathbf{We} = \lambda \mathbf{e}$$

which is the eigenvector equation for the matrix $\mathbf{W}$ with eigenvector $\mathbf{e}$ corresponding to its eigenvalue λ. Perron-Frobenius theorem states that a real square matrix with positive entries has a unique largest real eigenvalue and that the corresponding eigenvector has strictly positive components [33]. Thus if we choose $\mathbf{e}$ to be the eigenvector corresponding to largest eigenvalue of $\mathbf{W}$ we are guaranteed a positive value for each $e(i)$. We normalise $\mathbf{e}$ so that $|\mathbf{e}| = 1$. We could define eigenvector centrality $E_{\mathbf{W}}$ of a network $\mathbf{W}$ ($\mathbf{W}$ can represent a weighted or a binary network) to be the average of $e(i)$ over all nodes i, however as $|\mathbf{e}| = 1$, $E_{\mathbf{W}}$ is directly correlated with the size of $\mathbf{W}$ as $N \to \infty$ [22]. We can however calculate the average eigenvector centrality for the task-positive, task-negative and "other" nodes in the original 600×600 networks.

3.1 Results

3.1.1 Connectivity

We first calculate the average connectivity, defined in Section 3.0.1, of the 94 subjects of the original 600×600 weighted adjacency matrices for each task. We also calculate the standard deviation σ of

the group average for each task, via the MATLAB command STD, this range (2σ) is shown as an error on the displayed result charts. The results for the original 600×600 are shown in Fig. 5. We observe no distinctions between the tasks.

We now calculate the average connectivity of the 94 subjects, for each of the weighted six sub-networks defined in Section 2.4, and for all tasks. The results are shown in Fig. 6 for the TPP, TNN and TOO sub-networks, and in Fig. 7 for the TPN, TPO and TNO sub-networks.

At first glance it appears that the memory tasks have a higher connectivity across all sub-networks. However, the error bars are far too large to make any real distinctions. We also observe no differences between any of the sub-networks nor between any of the tasks.

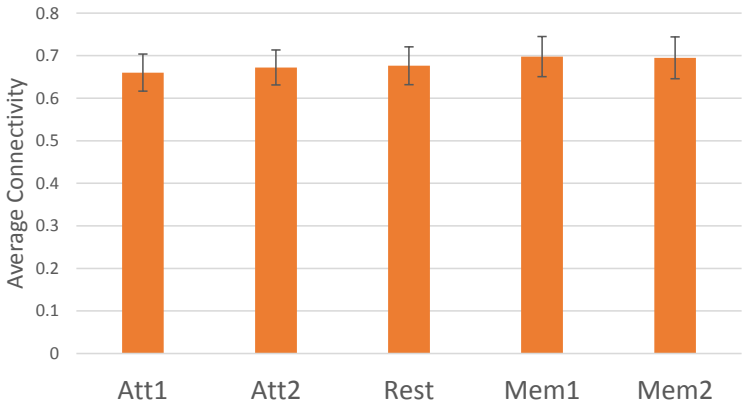

Figure 5: Average connectivity of the 94 subjects in the original 600×600 adjacency matrices.

We now calculate the average strength of the task-positive, task-negative and "other" nodes in the original 600×600 networks. We then average over all subjects in each of the tasks and include the standard deviation over the subjects. Note that as stated in Section 3.0.1, this calculation differs from connectivity in that we consider the weights of *all* connections associated with each node type. The results are shown in Fig. 8. We observe no significant differences, neither between any of the types of node nor between any of the tasks.

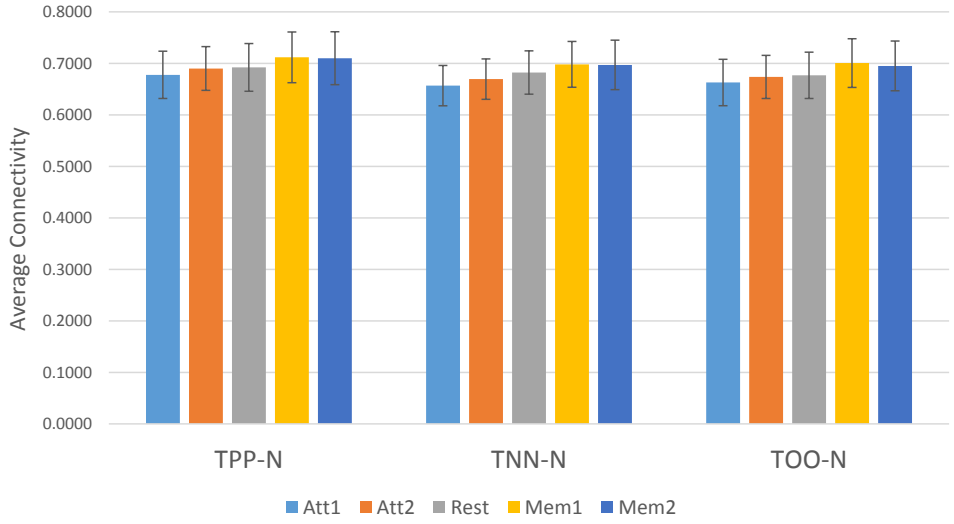

Figure 6: Average connectivity of the 94 subjects in each of the tasks for the TPP, TNN and TOO networks.

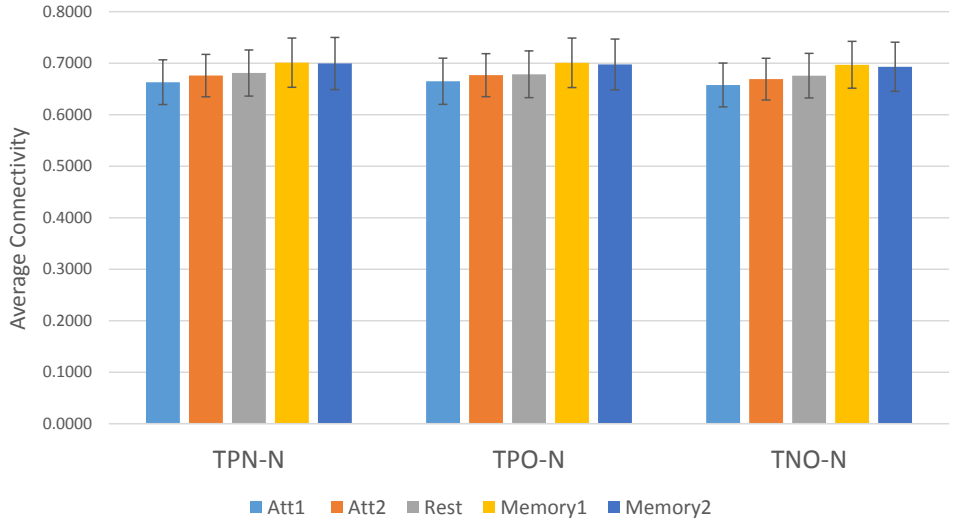

Figure 7: Average connectivity of the 94 subjects in each of the tasks for the TPN, TPO and TNO networks.

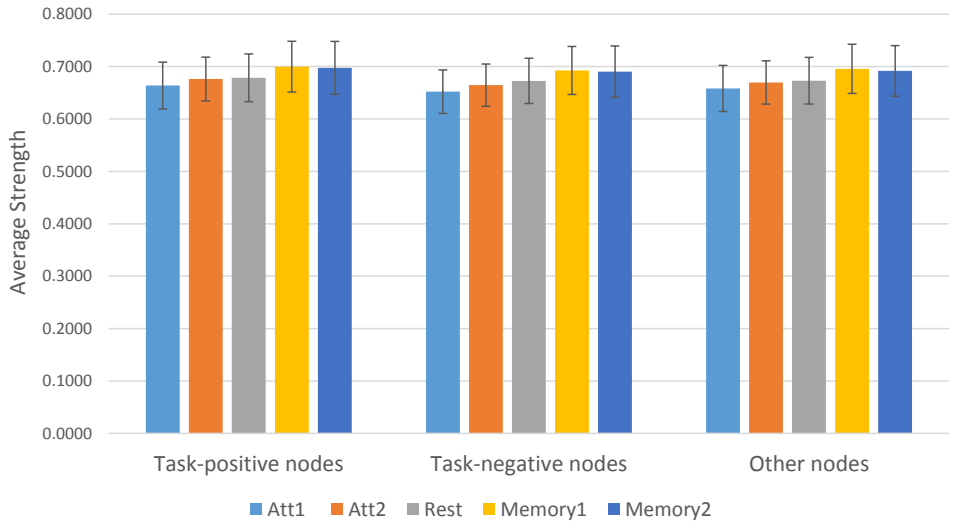

Figure 8: Average strength for each node assignment in the original 600×600 networks.

3.1.2 Clustering Coefficient

We mirror the procedure in Section 3.1.1 but using the clustering coefficient as defined in Section 3.0.2. We calculate the average clustering coefficient of the 94 subjects for the original 600×600 networks, and for each of the weighted six sub-networks. The results are shown in Fig. 9 for the original 600×600 matrices, in Fig. 10 for the TPP, TNN and TOO sub-networks, and in Fig. 11 for the TPN, TPO and TNO sub-networks. We make the same observation as in Section 3.1.1; because of the large error bars we recognise no significant differences between any of the tasks nor between any of the sub-networks.

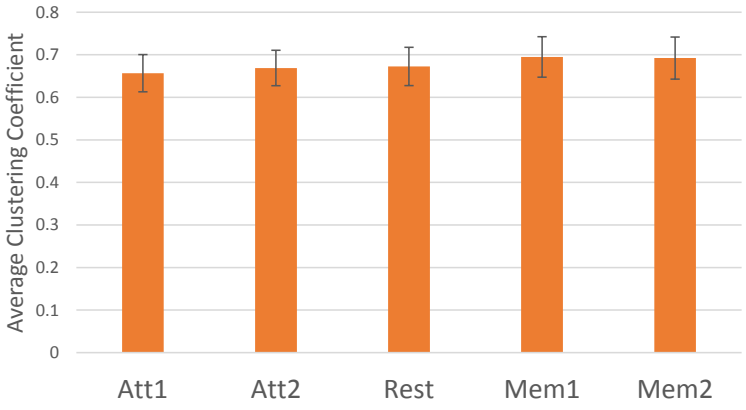

Figure 9: Average clustering coefficient of the 94 subjects in the original 600×600 adjacency matrices.

We now calculate the average clustering coefficient of the task-positive, task-negative and "other" nodes in the original 600×600 networks. We then average over all subjects in each of the tasks and include the standard deviation over the subjects; the results are shown in Fig. 12. We again observe no significant differences, neither between any of the types of node nor between any of the tasks.

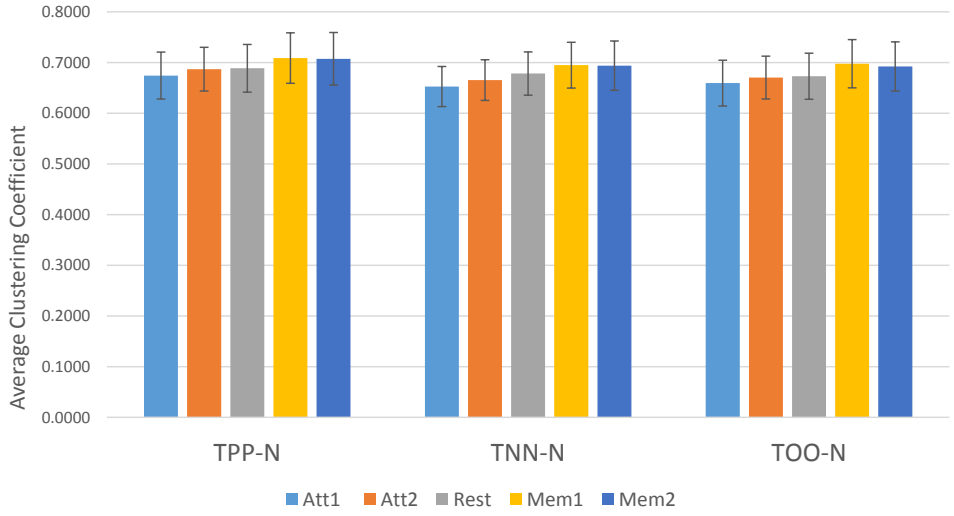

Figure 10: Average clustering coefficient of the 94 subjects in each of the tasks for the TPP, TNN and TOO networks.

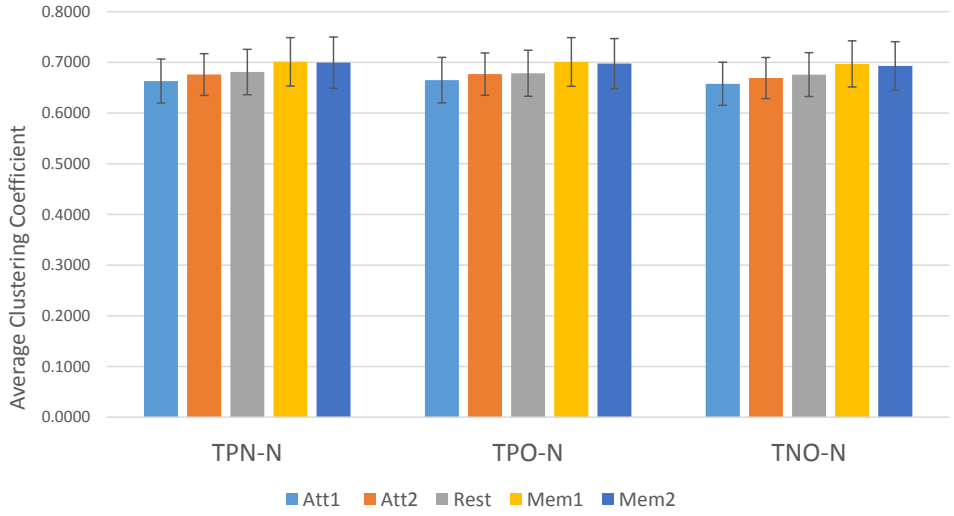

Figure 11: Average clustering coefficient of the 94 subjects in each of the tasks for the TPN, TPO and TNO networks.

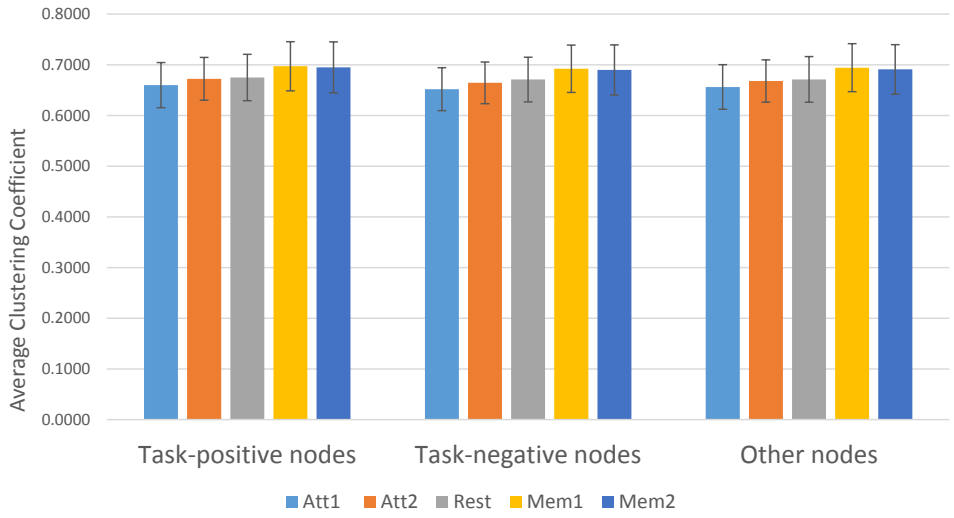

Figure 12: Average clustering coefficient for each node assignment in the original 600×600 networks.

3.1.3 Eigenvector Centrality

We now explore eigenvector centrality. Recall from Section 3.0.3 that if we were to calculate the average eigenvector centrality of the 94 subjects for any network or sub-network we would obtain a similar result across all tasks, as $E_{\mathbf{W}}$ is dependent only on network size. This is confirmed in Fig. 13.

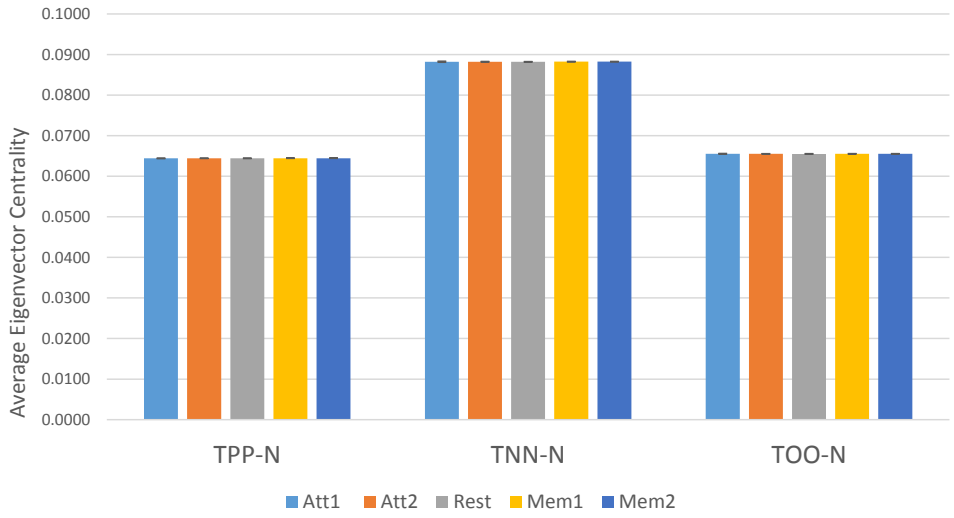

Figure 13: Average eigenvector centrality of the 94 subjects in the original 600×600 adjacency matrices.

We will calculate the average eigenvector centrality of the task-positive, task-negative and "other" nodes in the original 600×600 networks. We average the eigenvector centrality of each node assignment, and for each subject in each of the tasks. We include the standard deviation of the average over the subjects; the results are shown in Fig. 14. We are then able to make a comparison between tasks, because each network has size 600. We observe that the task-positive nodes have greater eigenvector centrality in attention tasks 1 and 2 and memory tasks 1 and 2, compared to task-negative nodes. We also observe that the task-positive nodes have greater eigenvector centrality than the "other" nodes in the resting state task.

3.1.4 Thresholding

Because the results examined in Sections 3.1.1 and 3.1.2 show no separation between tasks and sub-networks, we now explore connectivity, clustering coefficient and betweenness centrality using the thresholding technique as outlined in Section 2.3. Our aim is to analyse only the topology of the networks; the weights of the edges are now all equal and we focus on the connections that remain.

To make the chart easier to digest, we define *thresholding strength* j as $1/j = p$. Thus a thresholding strength of 1 corresponds to $p = 1$, which sets all the weights (except self edges) in a network $\mathbf{W}$ to be equal to 1. A thresholding strength of $j = 20$ corresponds to $p = 1/20 = 0.05$, which sets the 5% strongest edges in $\mathbf{W}$ to be equal to 1, and all others to 0. We explore the range $j \in \{1, 2, \ldots, 20\}$. Our thresholded networks are binary. Connectivity is not a suitable diagnostic for binary networks as

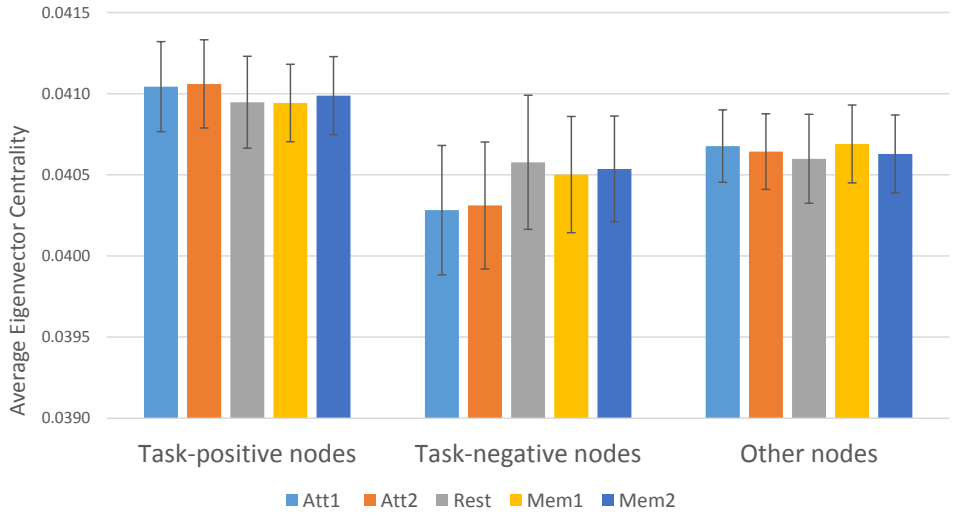

Figure 14: Average eigenvector centrality for each node assignment in the original 600×600 networks.

all networks of a given size will have the same connectivity for the same thresholding strength. We will also not be showing results for eigenvector centrality; our only option would be to calculate the eigenvector centrality for the node assignments in the 600×600 networks, as thresholding would give a huge variance to the number of connections each node would be left with, which would skew the results.

We start by showing results for the clustering coefficient. We use the binary clustering coefficient as defined in Section 3.0.2. We note that because of the large amount of information, there are many ways to represent these results in charts. We take the approach that is most visually appealing; we show the thresholding results across the range $j \in \{1, 2, \ldots, 20\}$ across all tasks for each sub-network separately. This has the disadvantage of not being able to visually compare across sub-networks. This comparison was done, however we will not explore the results in order to not overload this section with too many charts. No results were found across sub-networks.

Figure 15 shows the average clustering coefficient of the TPP sub-networks of the 94 subjects for each task over a range of thresholding strengths. The standard deviation error bars are shown for each task and for each thresholding strength. We observe no distinction between any of the tasks at all thresholding strengths; if there were to be any distinction we would at least see some separation between the error bars. We find similar results in the TNN, TOO, TPN, TPO and TNO sub-networks. We show the TPP and TNN sub-network results in Fig. 16, the TOO and TPN results in Fig. 17 and the TPO and TNO results in Fig. 18.

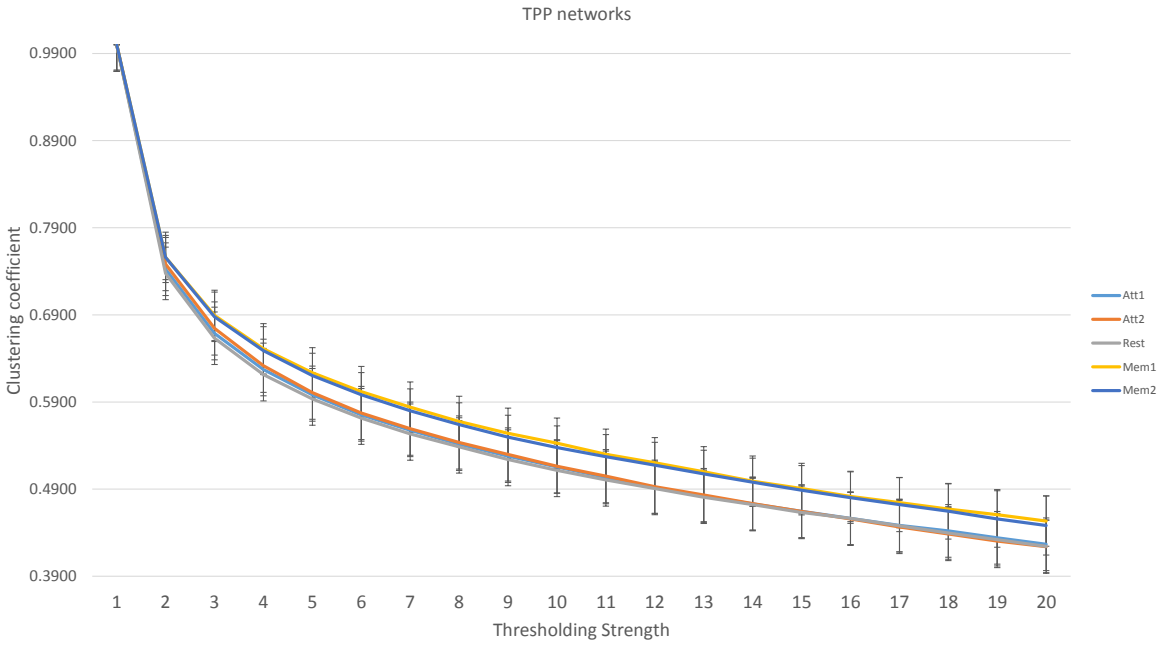

Figure 15: Average clustering coefficient of the TPP sub-network of the 94 subjects for each task over a range of thresholding strengths.

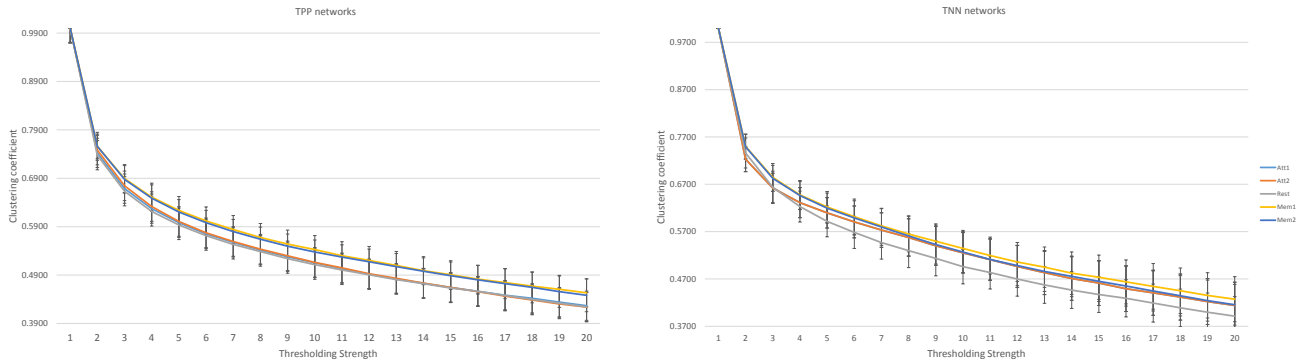

Figure 16: Average clustering coefficient of the TPP (also shown in Fig. 15) and TNN sub-networks of the 94 subjects for each task over a range of thresholding strengths.

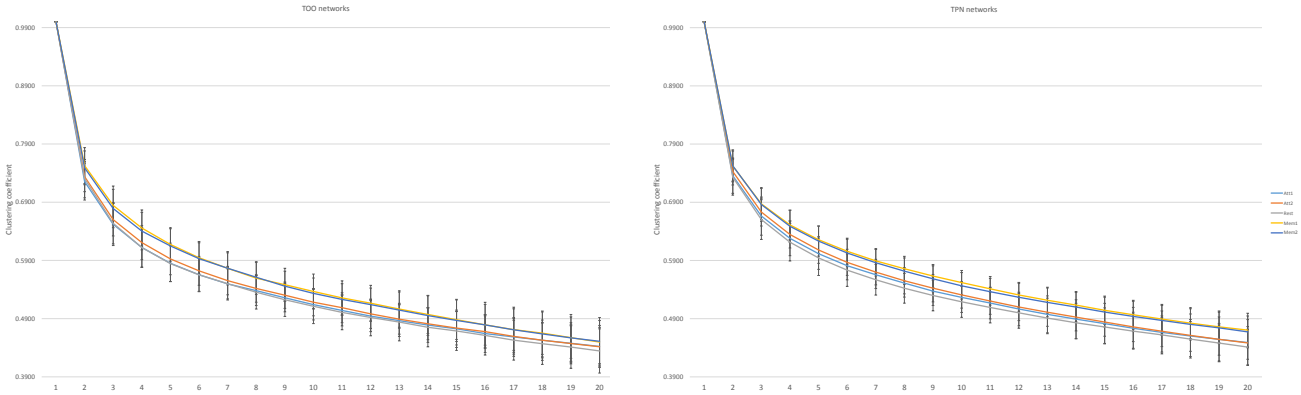

Figure 17: Average clustering coefficient of the TOO and TPN sub-networks of the 94 subjects for each task over a range of thresholding strengths.

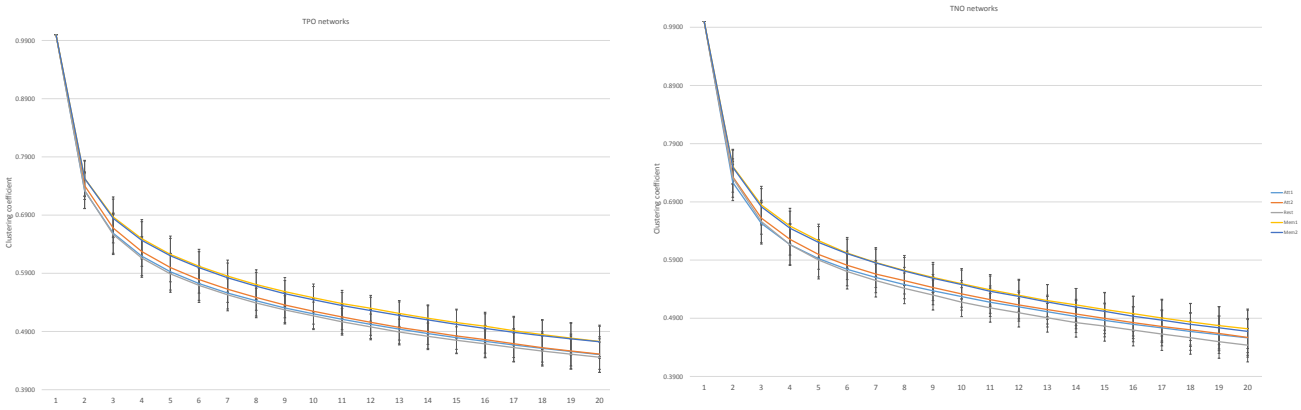

Figure 18: Average clustering coefficient of the TPO and TNO sub-networks of the 94 subjects for each task over a range of thresholding strengths.

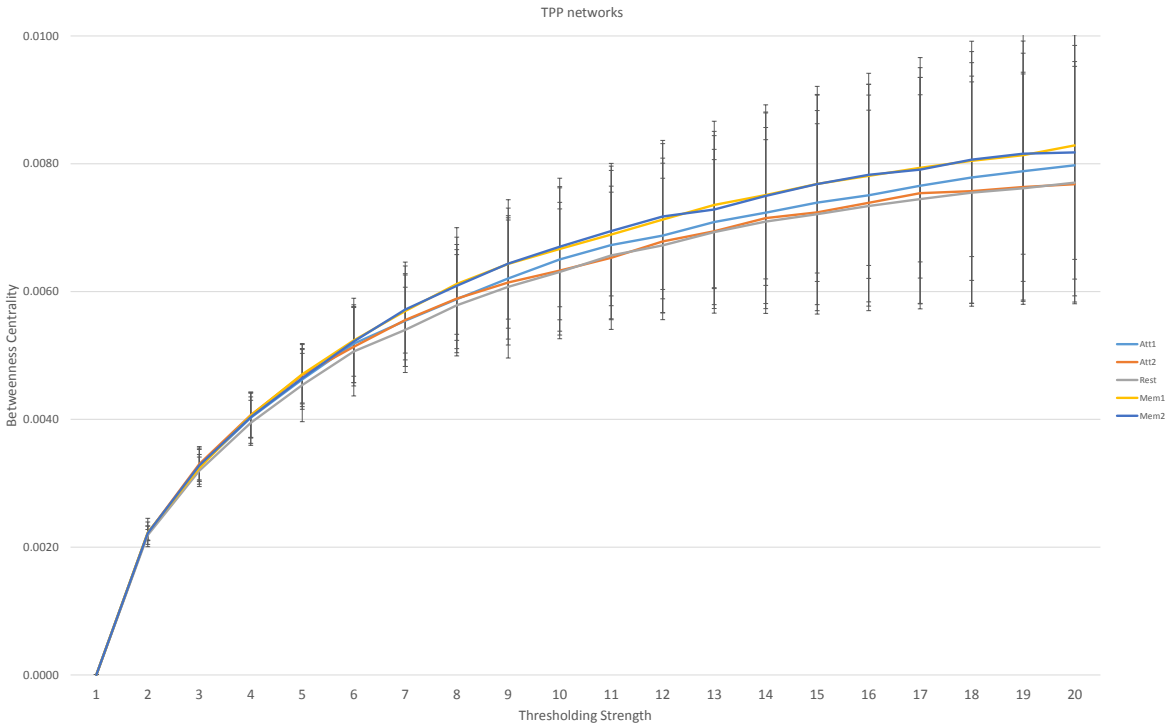

Figure 19: Average betweenness centrality of the TPP sub-network of the 94 subjects for each task over a range of thresholding strengths.

We now move on to betweenness centrality as defined in Section 3.0.3. Our thresholded networks are binary, hence betweenness centrality is a suitable diagnostic. As in Section 3.1.4 we show the results for each sub-network separately over the same range of thresholding strengths (i.e. $j \in \{1, 2, \ldots, 20\}$). The results for the TPP network are shown in Fig. 19. We are not able to make any distinction between the tasks, however we can observe that the standard deviation grows as the thresholding strength increases. We find similar results in the TNN, TOO, TPN, TPO and TNO sub-networks. We show the TPP and TNN sub-network results in Fig. 20, the TOO and TPN results in Fig. 21 and the TPO and TNO results in Fig. 22. Betweenness centrality scales with network size [22] hence we are not able to compare between sub-networks.

There is one sub-network which is distinct from the rest, the TNN sub-network. We take a closer look; the larger chart is shown in Fig. 23. It appears that the resting state task breaks away from the other tasks, however the error bars become too large to make any proper distinctions. We can however state that the error bars become larger than in the other sub-networks as thresholding strength increases.

Our initial conclusion is that we are unable to distinguish between tasks and sub-networks based on connectivity, clustering coefficient and betweenness centrality alone. There exist other types of thresholding methods [28] and many other diagnostics [2, 7] but we do not have the space in this thesis

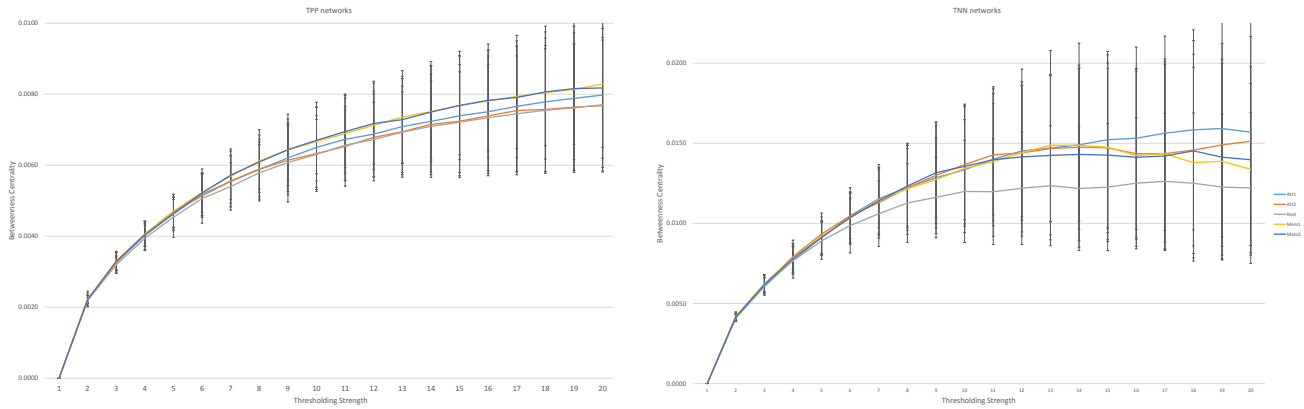

Figure 20: Average betweenness centrality of the TPP (also shown in Fig. 19) and TNN sub-networks of the 94 subjects for each task over a range of thresholding strengths.

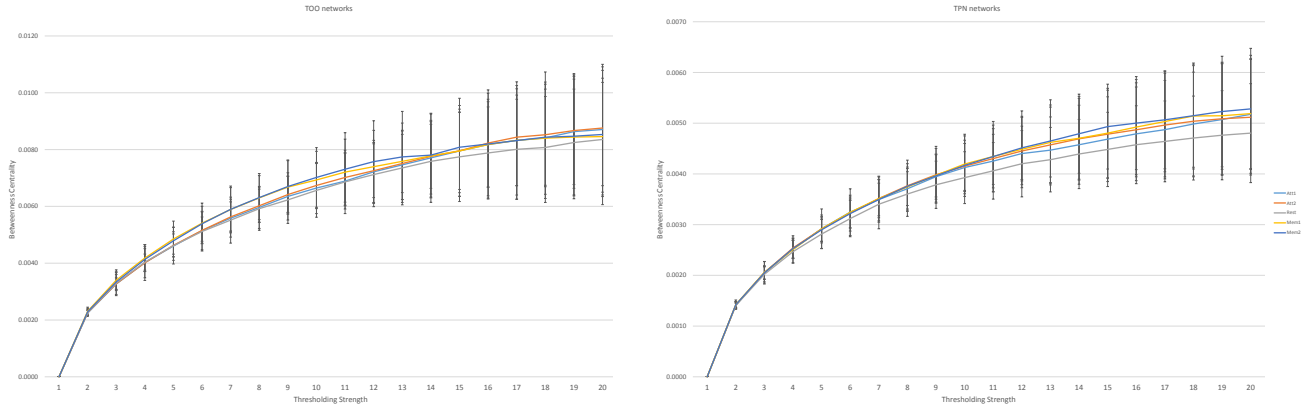

Figure 21: Average betweenness centrality of the TPP and TNN sub-networks of the 94 subjects for each task over a range of thresholding strengths.

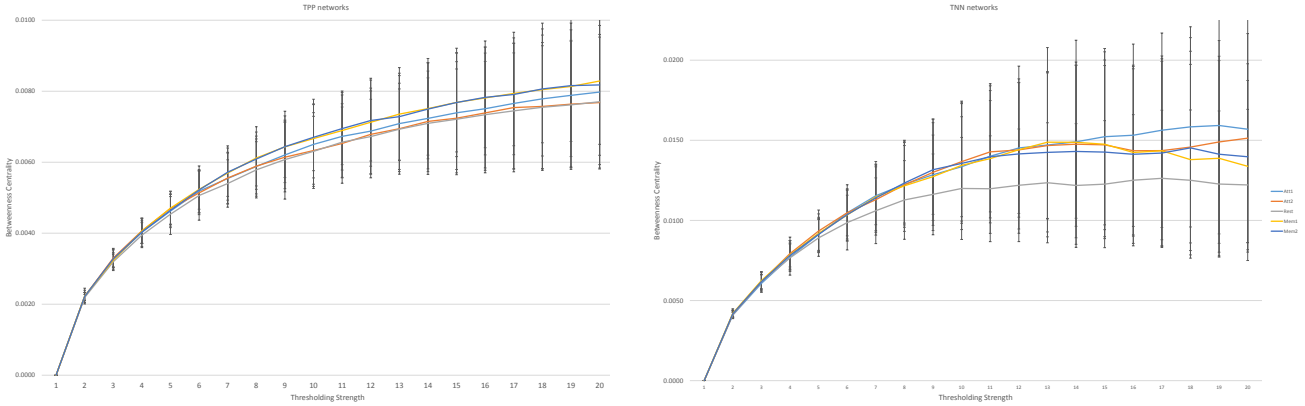

Figure 22: Average betweenness centrality of the TPO and TNO sub-networks of the 94 subjects for each task over a range of thresholding strengths.

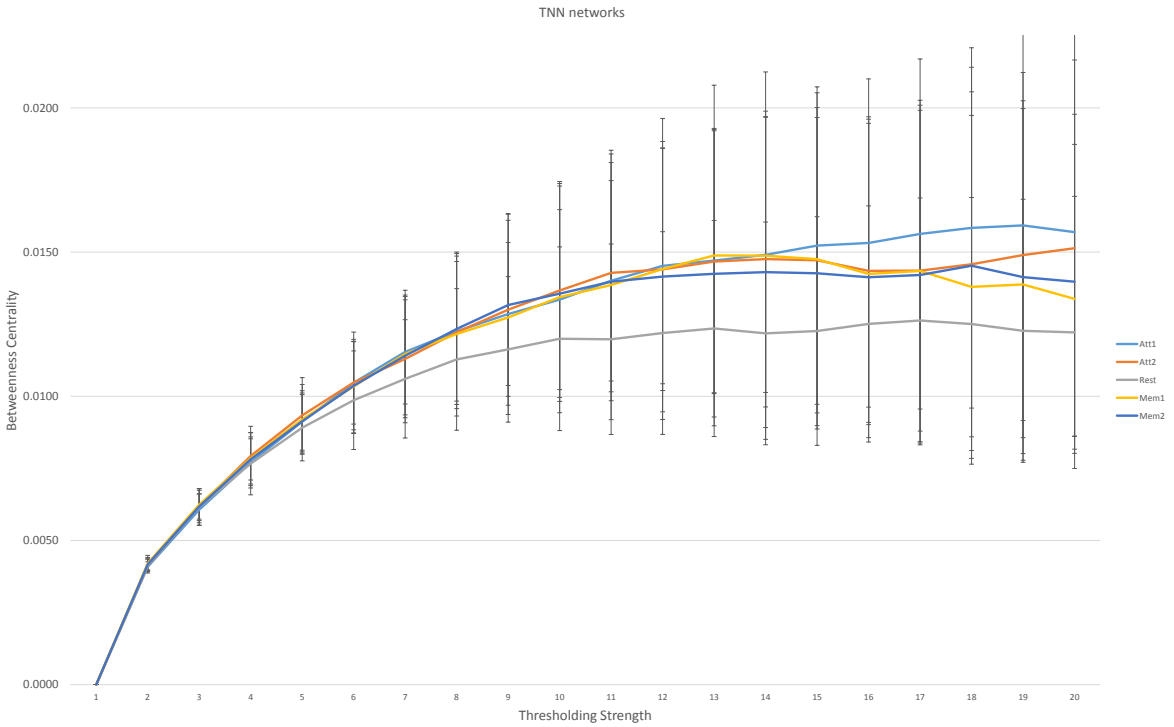

Figure 23: Average betweennesse centrality of the TNN sub-network of the 94 subjects for each task over a range of thresholding strengths.

to explore these fully. We are thus motivated to explore the more complex methods, spectral clustering and community detection. Our hope is that these methods are able to distinguish between the tasks and sub-networks.

4 Spectral clustering

One of the most important aims in many disciplines is to separate a set of objects in to groups such that objects within each group are similar and objects in distinct groups are dissimilar. The general task is called *cluster analysis* and it is a rich subject, with a wealth of methods to choose from [23]. We focus on one in particular; *spectral clustering,* and we follow the method specified in the paper by Higham et. al [21], with a few modifications for our purposes. We choose this method due to its previous success with other fMRI data [8]. Spectral clustering makes use of the spectrum of the *Laplacian* matrix of the adjacency matrix of a network. This method can be applied directly to the adjacency matrices but it is suggested that the *communicability* matrices, which we will define in Section4.0.6, give the most effective results [10].

4.0.5 The Spectral Clustering Method

Consider a network $\mathbf{W}$ (the network can be weighted or binary). We proceed by performing eigenvalue decomposition on the *Laplacian* of $\mathbf{W}$, which is defined as

$$\mathbf{L} = \mathbf{D_W} - \mathbf{W}, \tag{5}$$

where $\mathbf{D_W} = \text{diag}(S_i)$. We will also use the *normalized Laplacian*

$$L_{\text{norm}} = \mathbf{D_W}^{-\frac{1}{2}}(\mathbf{D_W} - \mathbf{W})\mathbf{D_W}^{-\frac{1}{2}}.$$

The Laplacian matrix is symmetric and positive semi-definite (i.e. all of its eigenvalues are positive). The smallest eigenvalue is 0 and the corresponding eigenvector is$\mathbf{1}$ (the vector with all elements equal to 1) [33]. We order the eigenvalues as follows:

$$0 = \lambda_1 < \lambda_2 < \lambda_3 \leq \ldots \leq \lambda_S,$$

and each eigenvector corresponds to the respective mutually orthonormal eigenvectors

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_S.$$

We stipulate a normalisation; $\mathbf{v}_1 = \mathbf{1}/\sqrt{S}$.

The normalised Laplacian matrix is also symmetric and positive definite with smallest eigenvalue 0, but the corresponding eigenvector is $\mathbf{D_W}^{\frac{1}{2}}\mathbf{1}$. We order the eigenvalues as follows:

$$0 = \mu_1 < \mu_2 < \mu_3 \leq \ldots \leq \mu_S,$$

with mutually orthonormal eigenvectors

$$\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_S$$

This time the normalisation is $\mathbf{w}_1 = \mathbf{D}_{\mathbf{W}}^{\frac{1}{2}}\mathbf{1} / \parallel \mathbf{D}_{\mathbf{W}}^{\frac{1}{2}}\mathbf{1} \parallel_2$. We refer to $\mathbf{v}_1$ as the *Fiedler vector* and to $\mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}}\mathbf{w}_2$ as the *normalised Fiedler vector*. The sum of all the Fiedler vector components must equal 0 [21], however this is not the case for the normalised Fiedler vector. The information in these eigenvectors form the basis for the clustering process and have the potential to differentiate between the nodes of the network. This is done by sorting the components of the Fiedler vector in to ascending order and identifying which group each component belongs to. We call this the *sorted* Fiedler vector. It is also possible to study $\mathbf{v}_3, \mathbf{v}_4$, and so on, in the unnormalised case and $\mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}}\mathbf{w}_3$, $\mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}}\mathbf{w}_4$, and so on, in the normalised case [1]. We will restrict our attention to the Fiedler vectors in this thesis.

4.0.6 Communicability

Many properties of networks are based on the assumption that most of the transport in the network flows along the shortest paths (for example, geodesic betweenness centrality as defined in Section 3.0.3). However, there are different scenarios in which other paths are important, especially in brain networks [16]. Thus only considering the shortest paths does not account for the global communicability of a network. We use this intuition to generalise the concept of communicability, following a definition first proposed by Estrada et al. [11].

We start with a binary adjacency matrix $\mathbf{A}$. We discuss the concept of a *walk* between nodes i and j. A walk is a sequence of (not necessarily different) nodes $a_1, a_2, \ldots, a_m$ such that for each $i = 1, 2, \ldots, m-1$, there exists an edge between a_i and a_{i+1}; such a walk is said to have length m. The important distinction between a walk and a path (defined in Section 3.0.3) is that a walk can revisit nodes and edges along the way [11]. We note the following important observation, which is well-known in graph theory [4]:

$$(A_{ij}^m) = \text{the number of walks of length } k \text{ between } i\text{and } j. \tag{6}$$

We can now define the *communicability* between a node i and j as a function of the number of walks of length $m = 1, 2, 3$, etc. Estrada et al. [11] noted that we can sum the number of walks of length m over all $m \in \mathbb{N}$, and in particular if we use a penalisation factor of $\frac{1}{m!}$ we have

$$\left(\sum_{m=1}^{\infty} \frac{A^m}{m!} \right)_{ij} = (e^A)_{ij}, \tag{7}$$

and note this is always convergent [11].

The *communicability* of a network with adjacency matrix $\mathbf{A}$ is then defined as the connectivity (as defined in Section 3.0.1) of the communicability matrix in (7).

We have discussed the communicability for a binary network and we must now consider a weighted network $\mathbf{W}$. Equation (7) remains valid when applied to $\mathbf{W}$, but it does not count the number of walks of length m; rather it contributes the product of all the weights from all the edges along the particular walk. However it has been noted that the above method does not give the desired results when applied directly to $\mathbf{W}$, as the resulting communicability matrix is disproportionately influenced by nodes with higher strengths [9]. Similar effects have been observed in spectral clustering where it has proved successful to judge the size of a cluster not by the number of nodes, but by the connectivity of the cluster [21]. We follow the method first specified by Crofts et al. [9]. To combat the nodes with

higher strengths, Crofts et al. introduce normalisation factor in which the weight W_{ij} is divided by the product $\sqrt{S_i S_j}$. Recalling that $\mathbf{D_W} = \text{diag}(S_i)$, which is a diagonal $N \times N$ matrix, we can define the communicability between nodes i and j in a weighted network in a similar way as before:

$$(\exp(\mathbf{D_W}^{-\frac{1}{2}}\mathbf{W}\mathbf{D_W}^{-\frac{1}{2}}))_{ij},$$

but we quickly run in to a complication; the average communicability is heavily dependent on the size and density of the networks. In this thesis we introduce a generalisation of the normalisation factor, and for $k \in \mathbb{N}$ we define the communicability of factor k of a weighted network $\mathbf{W}$:

$$(\text{Comm}_k(\mathbf{W}))_{ij} = (\exp(\mathbf{D_W}^{-\frac{1}{k}}\mathbf{W}\mathbf{D_W}^{-\frac{1}{k}})_{ij}. \tag{8}$$

We are now ready to explore what we can do with communicability and spectral clustering. It is possible to use the spectral clustering method to attempt to partition the task-positive, task-negative, and "other" nodes. There are five sets of 600×600 adjacency matrices, and six sub-networks from which we can apply the spectral clustering method. However as discussed in Section 1.1, Fox et al. [15] defined the task-positive nodes to be associated with attention tasks, and task-negative nodes to be associated with resting state and memory tasks. We do not apply the method to the original 600×600 adjacency matrices as the "other" nodes are not intrinsically important. Thus we direct our attention to the TPN, TPO and TNO sub-networks (we are not able to partition the other three sub-networks as each sub-network only contains one type of node), but in particular the TPN network is the most interesting case to us as it contains the two important assignments.

4.1 Results

4.1.1 Communicability Results

We initially direct our attention to the TPP, TNN and TOO sub-networks, and calculate the average communicability over the 94 subjects for all tasks, for $k \in \{1, 2, 3, 4, 5\}$. We first examine the case $k = 1$. The results are shown in Fig. 24. Recall from Section 4.0.6 that we are not permitted to compare columns, as communicability scales with network size. Thus we direct our attention to comparing the tasks within each sub-network. We observe that there is almost no difference between any of the tasks. As we shall see in the next few graphs, this is a result of the choice of k and is not a result of the definition of communicability itself.

The results for the cases $k = 2, 3, 4$ and 5 are shown in Figs. 25, 26, 27 and 28 respectively. For the case $k = 2$, we again observe almost no difference between any of the tasks. This is not the case when $k = 3$, we start to see a small variation between tasks. We are still not able to distinguish between the tasks. In the cases of $k = 4$ and $k = 5$, we observe that the standard deviation grows very large to the point where the interval 2σ crosses the x-axis.

Based on this information, we take $k = 2$ and calculate the average communicability over all subjects in the TPN, TPO and TNO sub-networks. We observe no distinctions between tasks, however we do note that $k = 3$ seems a valid choice for the normalisation factor for these network sizes (the size of these sub-networks is slightly larger than that of the previous examples) as there is a variation between tasks.

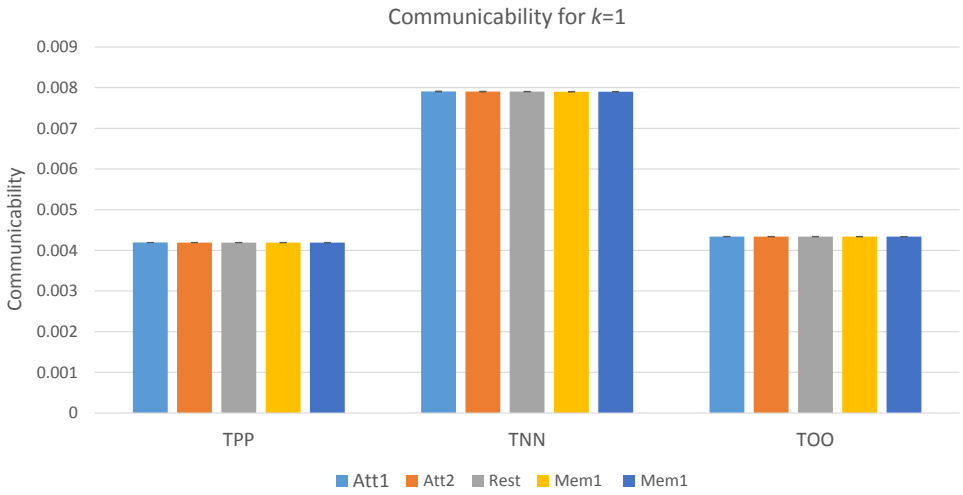

Figure 24: Average communicability over the 94 subjects for the case $k = 1$.

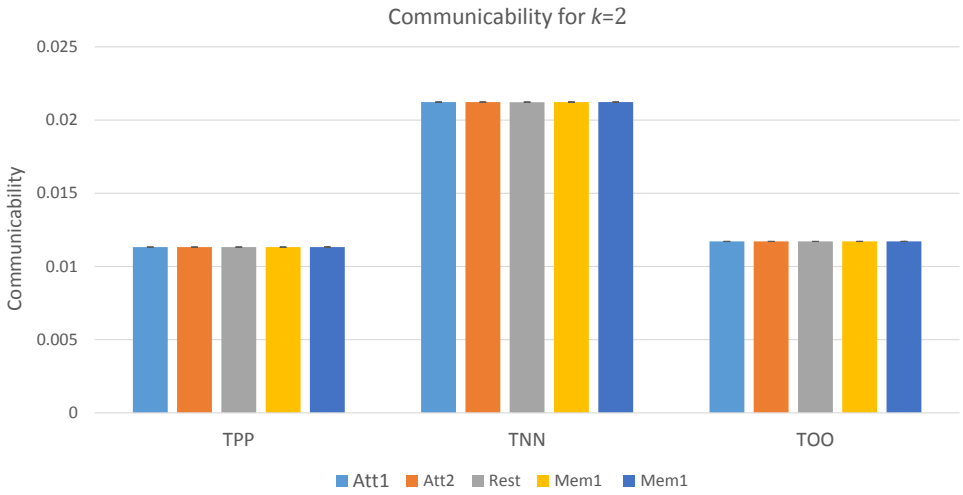

Figure 25: Average communicability over the 94 subjects for the case $k = 2$.

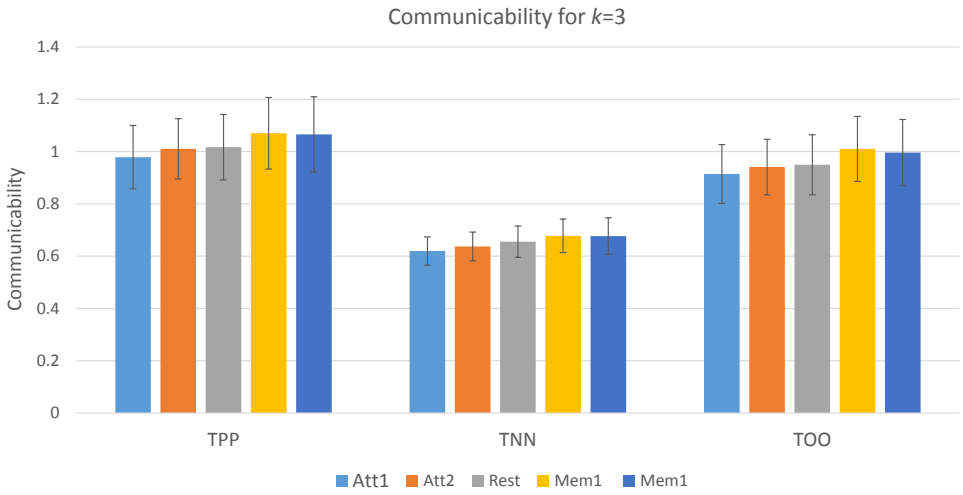

Figure 26: Average communicability over the 94 subjects for the case $k = 3$.

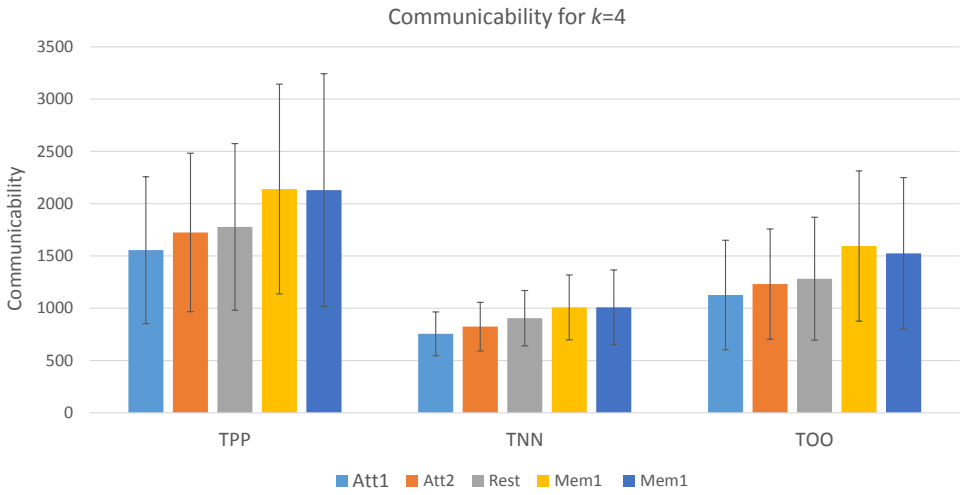

Figure 27: Average communicability over the 94 subjects for the case $k = 4$.

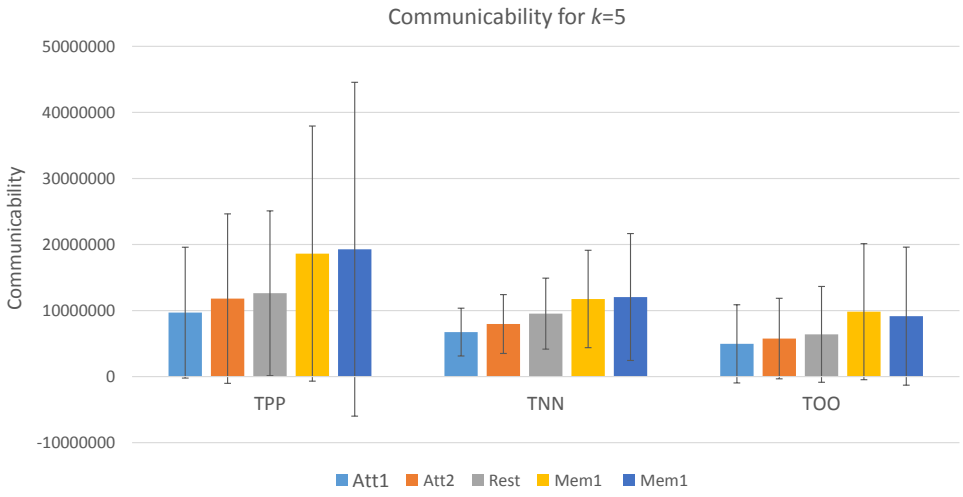

Figure 28: Average communicability over the 94 subjects for the case $k = 5$.

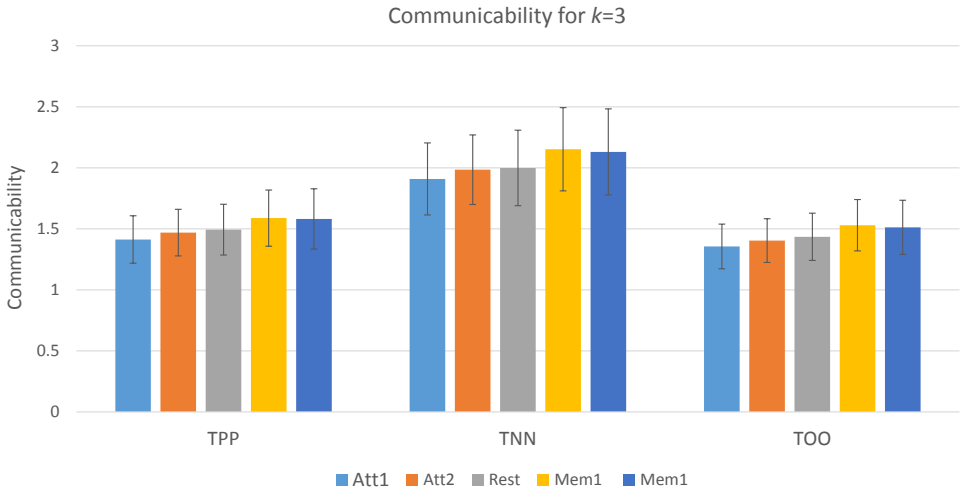

Figure 29: Average communicability over the 94 subjects for the case $k = 5$ in the TPP, TNN and TOO sub-networks.

4.1.2 Comparing Node Assignments

We first explore the way in which we can show the results. By the method of Section 4.0.6, we can apply the spectral clustering method to the TPN sub-networks of each of the 94 subjects across all tasks. This results in $94 \times 5 = 470$ Fiedler vectors, thus it would be impractical to show in detail the results for all of these. Instead, we can perform a two-sample $t-$test on the positions of the task-positive and task-negative nodes in the sorted Fiedler vector after sorting the components in to ascending order. This is done by using the MATLAB command TTEST2, and we obtain a p-value for which we can use to examine the results. We use the convention that $p < 0.05$ corresponds to a significant result.

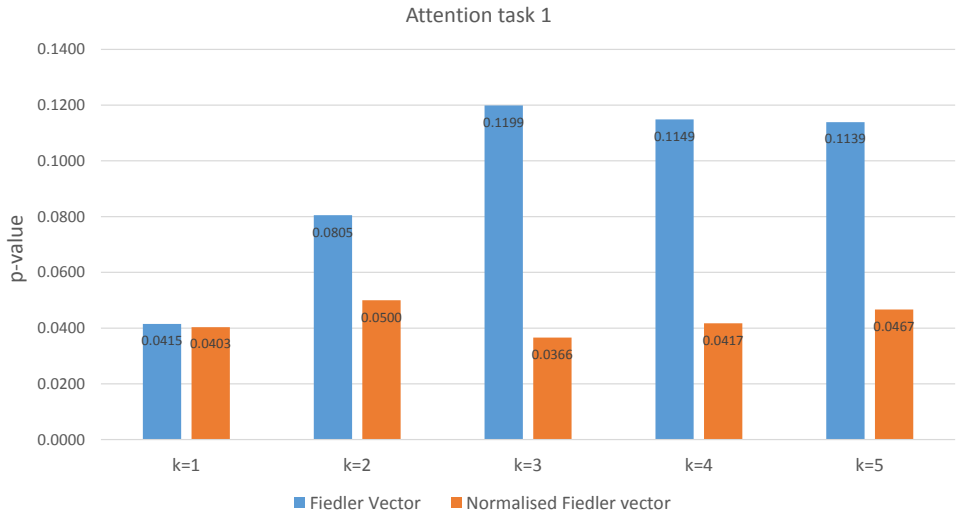

Figure 30: The p-value averaged over all subjects, associated with the various Fiedler vectors for differing values of k

We show results for varying k, the constant associated with the normalisation power in Section 4.0.6. However we must first make a choice whether to use the Fiedler vector or normalised Fiedler vector. As Higham et al. discussed [21], the nature of the results obtained depend upon the type of data. We direct our attention only to attention task 1. We work our the p-value for each subject corresponding to positions in the sorted Fiedler vector and normalised Fiedler vector. We average the p-value over all 94 subjects. The results are shown in Fig. 30. Based on this result we direct our attention to the normalised Fiedler vector for this subsubsection. We can now compare the results across tasks. Varying k, compute the p-value corresponding to the positions of the task-positive and task-negative in the sorted normalised Fiedler vector for the TPN sub-network. The results are shown in Fig. 31. We observe that significant results are obtained for all values of k in the attention task 1, but no other tasks. This means that there is an important distinction between task-positive and task-negative nodes in the attention task 1. We can view a typical normalised Fiedler vector plot; we choose subject 2 with $k = 1$. This is shown in Fig. 32. We observe that although the task-positive nodes appear equally spaced over the 368 components (x-values), the task-negative nodes are clusterered on

the right side of the plot.

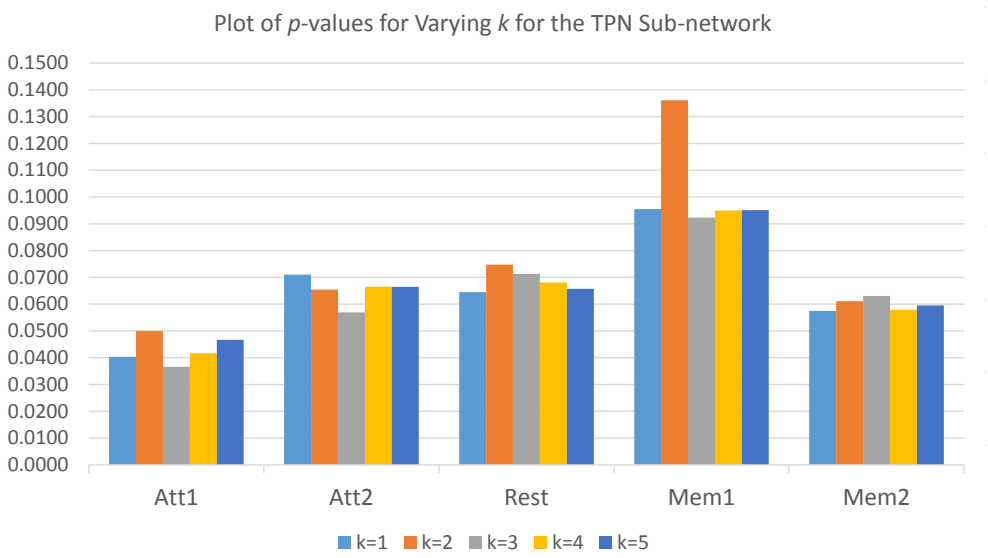

Figure 31: Plot of p-values corresponding to the positions of the task-positive and task-negative nodes in the sorted normalised Fiedler vector for the TPN sub-network for varying k, across all tasks.

We repeat this process for the TPO and TNO sub-networks and we continue our restriction to the normalised Fiedler vector. For the TPO sub-network, we again vary k, and we compute the p-value corresponding to the positions of the task-positive and "other" nodes in the sorted normalised Fiedler vector for the TPO sub-network. The results are shown in Fig. 33 . We observe for all values of k that no task has a significant average p-value. However we do note that for attention task 1 with $k = 1$, there exist 77 subjects with a p-value of less than 0.05, and we find similar numbers across all others values of k and all tasks. Thus in most subjects there seems to be a distinction between task-positive and "other" nodes.

We repeat this process for the TNO sub-network. The results for the p-values are shown in Fig. 34. We again observe that for all values of k that no task has an significant average p-value.

We briefly examine the Fiedler vector components after applying the spectral clustering method to the original 600×600 network of attention task 1 of subject 2, with $k = 1$. The results are shown in Fig. 35 . We observe that the task-positive nodes are clustered to the left of the graph, while the remaining nodes are clustered to the right. We do not have space in this thesis to fully explore these results.

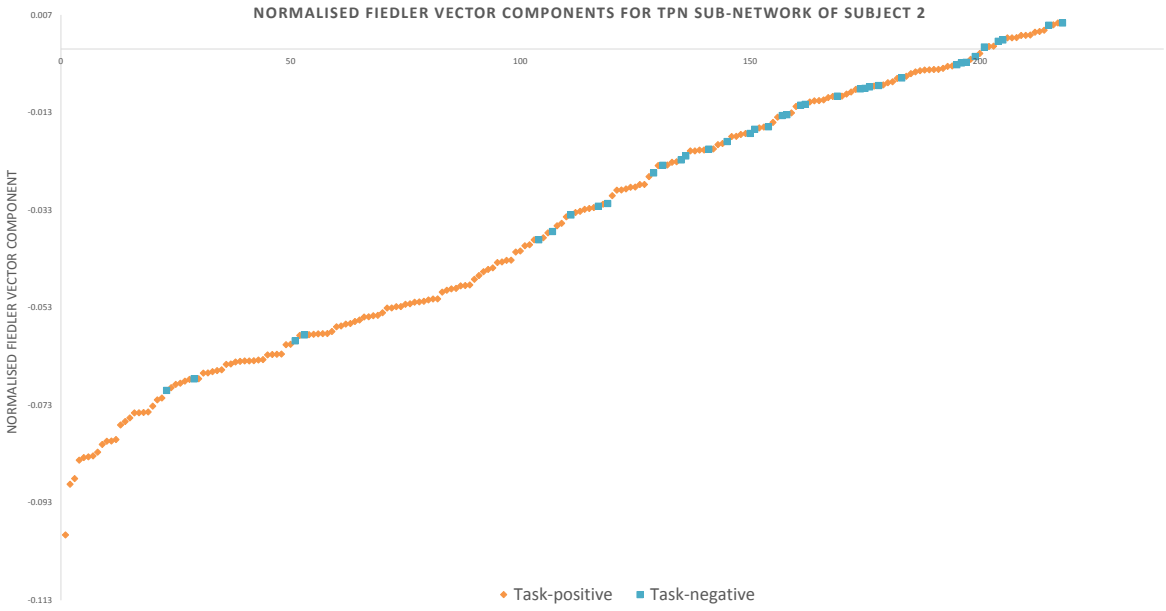

Figure 32: Normalised Fiedler vector components for the TPN sub-network of subject 2 with $k = 1$. Task-positive nodes are shown in orange; task-negative nodes are shown in blue squares.

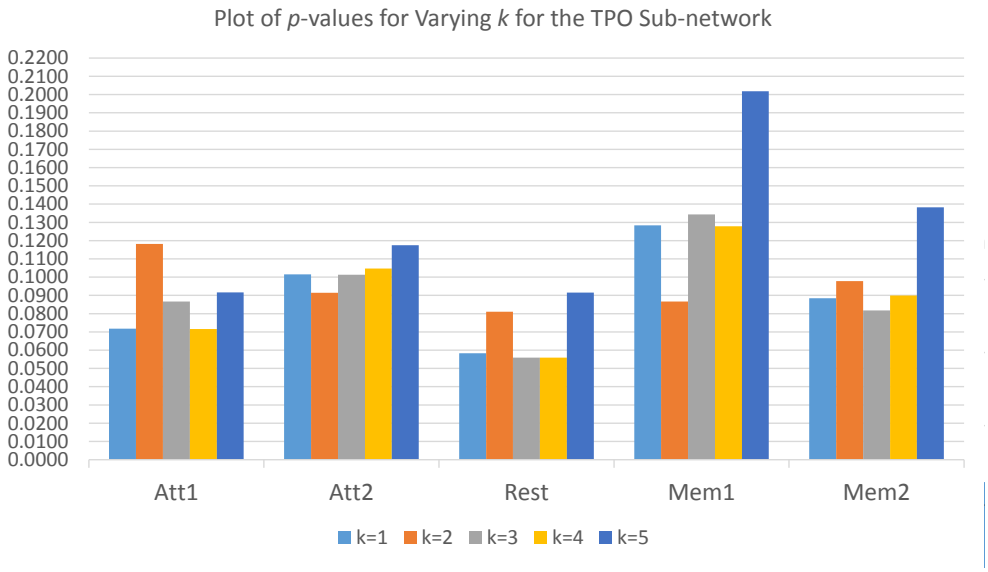

Figure 33: Plot of p-values corresponding to the positions of the task-positive and "other" nodes in the sorted normalised Fiedler vector for the TPO sub-network for varying k, across all tasks.

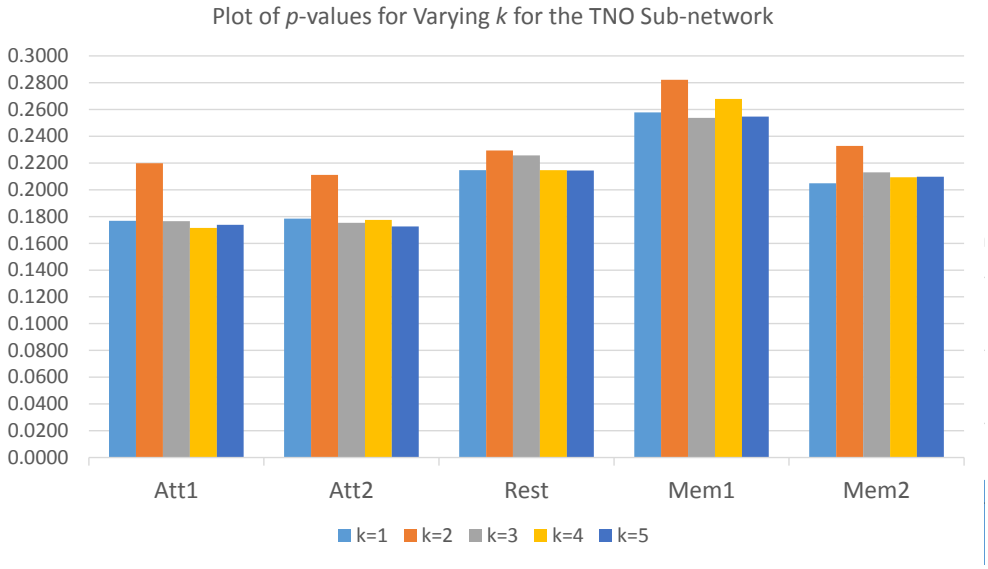

Figure 34: Plot of p-values corresponding to the positions of the task-negative and "other" nodes in the sorted normalised Fiedler vector for the TPN sub-network for varying k, across all tasks.

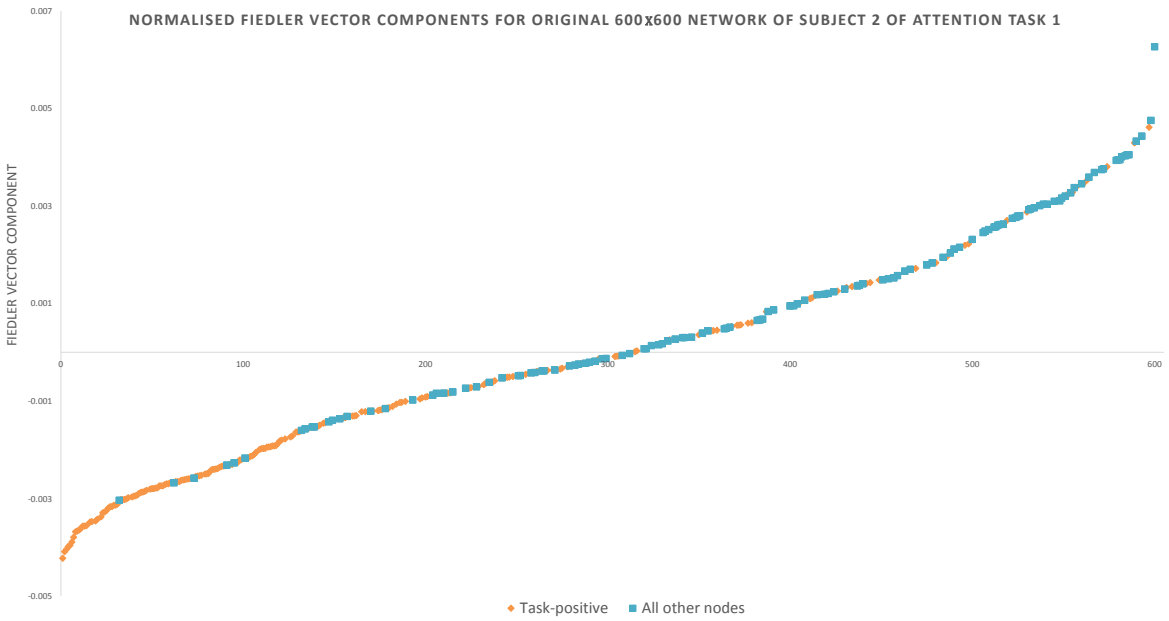

Figure 35: Fiedler vector components for the original 600×600 sub-network of subject 2 with $k = 1$. Task-positive nodes are shown in orange; the remaining nodes (both task-negative and "other" nodes) are shown in blue squares.

5 Community detection

A widely studied problem in network science concerns identifying tightly connected groups of nodes known as *communities* [7]. Intuitively, it seems reasonable find certain sets of nodes such that there exists a greater number of connections (in the weighted case, we replace the number of connections by the total edge weight) within the sets, but fewer connections between the sets compared to a random *null model* (which we define in this section). This intuition becomes clearer by considering a simple example; see Fig. 36. We consider two types of community detection. The first will consider only a single network when identifying communities. The second will concern *multilayer community detection* [32], which will involve assigning each node a community across a series of networks called *layers*. The layers could be the networks of the points in a time series, or the networks corresponding to the subjects themselves.

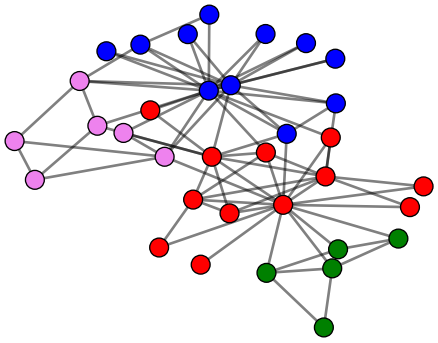

Figure 36: A well known example; The Zachary Karate Club. The edges represent social ties. Each member has been assigned to one of four communities. Data was taken from the original paper [47] and procssed using python (HTTPS://BITBUCKET.ORG/TAYNAUD/PYTHON-LOUVAIN, see also the acknowledgements section).

In both cases, we will be detecting communities by maximising a *quality function* Q, but first we must make a few definitions.

We will be defining this for a weighted network $\mathbf{W}$. Let $2m$ be the total edge weight in the network and let g_i be the community assignment of node i. We have

$$\delta(g_i, g_j) = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are assigned the same community} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

For a static network we define Q as in [32]:

$$Q = \frac{1}{2m} \sum_{ij} (W_{ij} - P_{ij}) \delta(g_i, g_j),$$

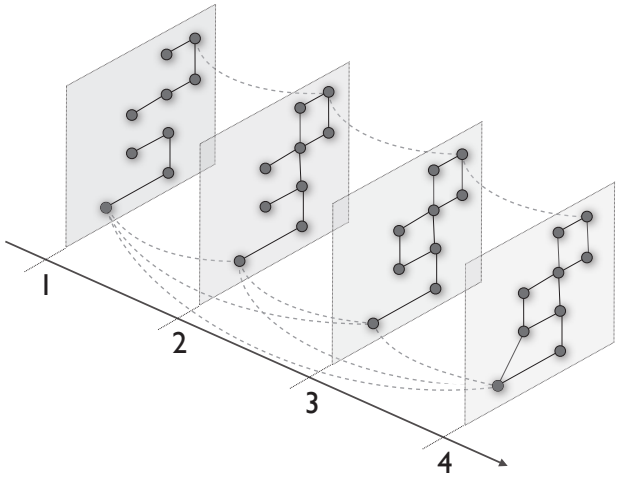

Figure 37: Pictorial representation of a binary multilayer network. The intralayer couplings are represented by the solid lines; the interlayercouplings by the dashed lines. This figure is from Ref. [32] and used with permission.

where P_{ij} is the expected connection strength between nodes i and j under the null model. The choice is the random null model is a crucial one when studying network community structure [36]. The most common choice for this null model is $P_{ij} = S_i S_j/(2m)$. This is the expected connection strength if the weight of edge (i, j) is to arise by chance [33]. Thus, high values of Q indicate network partitions in which there is a larger edge weight within the group than expected by chance. With this choice of P_{ij}, the function that we are maximising is

$$Q = \frac{1}{2m} \sum_{ij} \left(W_{ij} - \frac{S_i S_j}{2m} \right) \delta(g_i, g_j). \tag{10}$$

We can now generalise this for the multilayer case. Denoted the layers by $t = 1, 2, \ldots, S$. We must consider *interlayer couplings* and *intralayer couplings* which are shown pictorially in Fig. 37. The intralayer coupling for nodes i and j on a layer t is denoted by A_{ijt}. The interlayer coupling that connects node j in layer r to itself in layer t is denoted by C_{jrt}. We let

$$2\mu = \sum_{ijt} A_{ijt} + \sum_{jrt} C_{jrt}, \tag{11}$$

which is the sum of all of the edge weights in the multilayer network. For a specific layer t, we generalise the parameters g_i, m, and S_i in equation (10) to g_{it}, m_t, and S_{it}. We now introduce γ, which is a *resolution parameter*; it is responsible for the number of communities across the multilayer network and the number of nodes in them. Lambiotte et al. [26] identified γ as the inverse of the timescale of a random walk on the network. Smaller values of γ result in larger communities; larger values result

in smaller communities. The default resolution is $\gamma = 1$ however γ is allowed to vary between 0 and ∞ but we will typically be interested in $\gamma < 1.05$ (this will be explored in Section 5.1). It is also possible to specify a different resolution parameter for each layer but for simplicity we will keep γ constant. As for C_{jrt}, we presume for simplicity that $C_{jrt} \in \{0, \omega\}$, where $C_{jrt} = 0$ represents the absence of an interlayer edge and $C_{jrt} = \omega$ represents the presence of an interlayer edge. We will explore a range of values for ω; it is allowed to vary between 0 and ∞, but we will direct our interest to the interval $[0, 0.3]$ (again, this will be explored in Section 5.1). The case $\omega = 0$ corresponds to no coupling between layers and results in single-layer community detection. We thus have for a multilayer network, as in [32];

$$Q^{\text{multi}} = \frac{1}{2\mu} \sum_{ijtr} \left[\left(A_{ijt} - \gamma \frac{S_{it} S_{jt}}{2m_t} \right) \delta(t, r) + \delta(i, j) C_{jrt} \right] \delta(g_{it}, g_{jt}). \tag{12}$$

The output gives a community assignment for each node in each layer. When we consider a multilayer network consisting of subjects as the layers, we will use the word *categorical (multilayer) network*. We use the "generalized Louvain" MATLAB code[6] to maximise Q and Q^{multi}.

5.0.3 Community diagnostics

We now explore in detail one of the diagnostics we can compute after performing community detection. This is the notion of the *flexibility* of a node i. Flexibility was first defined by Bassett et al. [3] in 2011. It is a measure that captures changes in the local properties of of a node.

Two definitions of flexibility have been proposed thus far [3]. To state them precisely we introduce community notation as follows: if node i, over layers $t = 1, 2, \ldots, H$, belongs to the (not necessarily distinct) communities $g_{i1}, g_{i2}, \ldots, g_{iH}$ we write $\text{comm}(i) = (g_{i1}, g_{i2}, \ldots, g_{iH})$, where as before g_{it} is the community assignment of i in layer t. The g_{it} will be integers that index the communities.

We define $\text{flex}_1(i)$ of a node i to be the number of times i changes communities in successive layers (i.e. the number of times we have $g_{it} \neq g_{i(t+1)}$). We define $\text{flex}_2(i)$ of a node i to be the total number of distinct communities node i belongs to. We refer to the first definition as *flexibility 1* and the second as *flexibility 2*. To compare these two definitions of flexibility, suppose we have 8 layers with $\text{comm}(i) = (1, 2, 1, 2, 2, 2, 1, 3)$. We then have $\text{flex}_1(i) = 5$ and $\text{flex}_2(i) = 3$.

This definition was motivated by a multilayer network composed of time series networks. This means measuring the community assignment of a node over a period of time. Our multilayer networks include the possibility of the layers being the subjects themselves, hence although it is possible to calculate both definitions of flexibility in this case (the calculation is performed in the same way), it does not make sense to use the first definition because we are not interested in the behavior of the node over a time period. Shuffling the layers in the catagorical case (which would not affect the community assignments as the layers in this case are not ordinal) would give us a different answer. It is therefore not appropriate to call this calculation (either flex_1or flex_2) "flexibility" in the catagorical case. However it raises the question of whether these two diagnostics are correlated:

1. Calculating the flexibility for each node (averaging the flexibility of each node across all subjects) in a multilayer network composed of a time series, where the layers make up the adjacency matrices of each time window (as in the method specified in Section (2.2)); we call this *ordinal flexibility*.

[6] HTTP://NETWIKI.AMATH.UNC.EDU/GENLOUVAIN/GENLOUVAIN

2. Calculating the flexibility of each node via the catagorical multilayer network; we call this *categorical flexibility*.

This process is illustrated in Fig. 38 . It is not immediately mathematically clear if there exists such a link. Note that this comparison can only be done on attention tasks 1 and 2 and resting state because there is no time series data for the memory tasks. We explore results only from attention task 1. As we shall see later in Section 5.1, it is advantageous to create a third definition of flexibility. We will call this definition *flexibility 3*. For a node i with community assignment $\text{comm}(i) = (g_{i1}, g_{i2}, \ldots, g_{iH})$ suppose that i is in h distinct communities $g_{i1}, \ldots, g_{ih}$, and let $n(g_{it})$ be the total number of appearances in community g_{it}. Suppose further that $n(g_{i1}) > n(g_{i2}) > \ldots > n(g_{ih})$. This is allowed as we are permitted to reorder the layers in the categorical case. Define

$$\text{flex}_3(i) = \sum_{j=1}^{h} f(j)n(g_{ij}) \tag{13}$$

where $f(j)$ is a weighting function. The default choice for the weighting function is $f(j) = j$, but we will explore what happens when we increase the "strength" of this weighting factor. The idea of this definition is to give a greater variability of flexibility of nodes when using multilayer community detection. For example, if there were a total of 7 communities identified across the 600 nodes and 94 layers, there would only be a possible of 7 values for $\text{flex}_2(i)$ across all nodes i. Our motivation for this definition will become clearer in Section 5.1.

The flexibility of a network $\text{flex}_k(\mathbf{W})$ where $k \in \{1, 2, 3\}$ is defined as the mean flexibility k over all nodes i.

To explore this connection, we will need to define a random-graphs null model to which to compare the results. There are many models of random-graphs [33], however we introduce a basic random-graph model. Recall the matrix in Fig. 3 in Section 2.2. We randomly permute the signal vectors S_i, and re-create the matrices with the method from Section 2.2 and follow the method specified in Fig. 38. We perform this step multiple times and record the results in the form of a correlation coefficient between ordinal and categorical flexibility and compare the equivalent result obtained from the unaltered data.

5.1 Results

As discussed in Section 5 we must first choose a value for the resolution parameter γ and the interlayer coupling ω. There is no *best* choice, however we can vary these parameters and exclude values which give us unsuitable results. For example, if γ is sufficiently large, each node will be assigned its own individual community thus the flexibility (under all versions of flexibility) for all nodes would be equal. If ω is sufficiently large, the community assignment of node i will remain constant throughout all layers resulting in a flexibility of 0.

We calculate the categorical flexibility 2 of the TNN sub-network for attention task 2, resting state, and memory task 1. Let $\mathbf{A}_k$ represent the TNN sub-network in the attention 1 task for subject k, $\mathbf{B}_k$ the TNN sub-network in the resting state for subject k and $\mathbf{C}_k$ the TNN sub-network in memory task 1 for subject k. We calculate the quantity

$$\text{diff}(k) = |\text{flex}(\mathbf{A}_k) - \text{flex}(\mathbf{B}_k)| + |\text{flex}(\mathbf{A}_k) - \text{flex}(\mathbf{C}_k)| + |\text{flex}(\mathbf{C}_k) - \text{flex}(\mathbf{B}_k)|$$

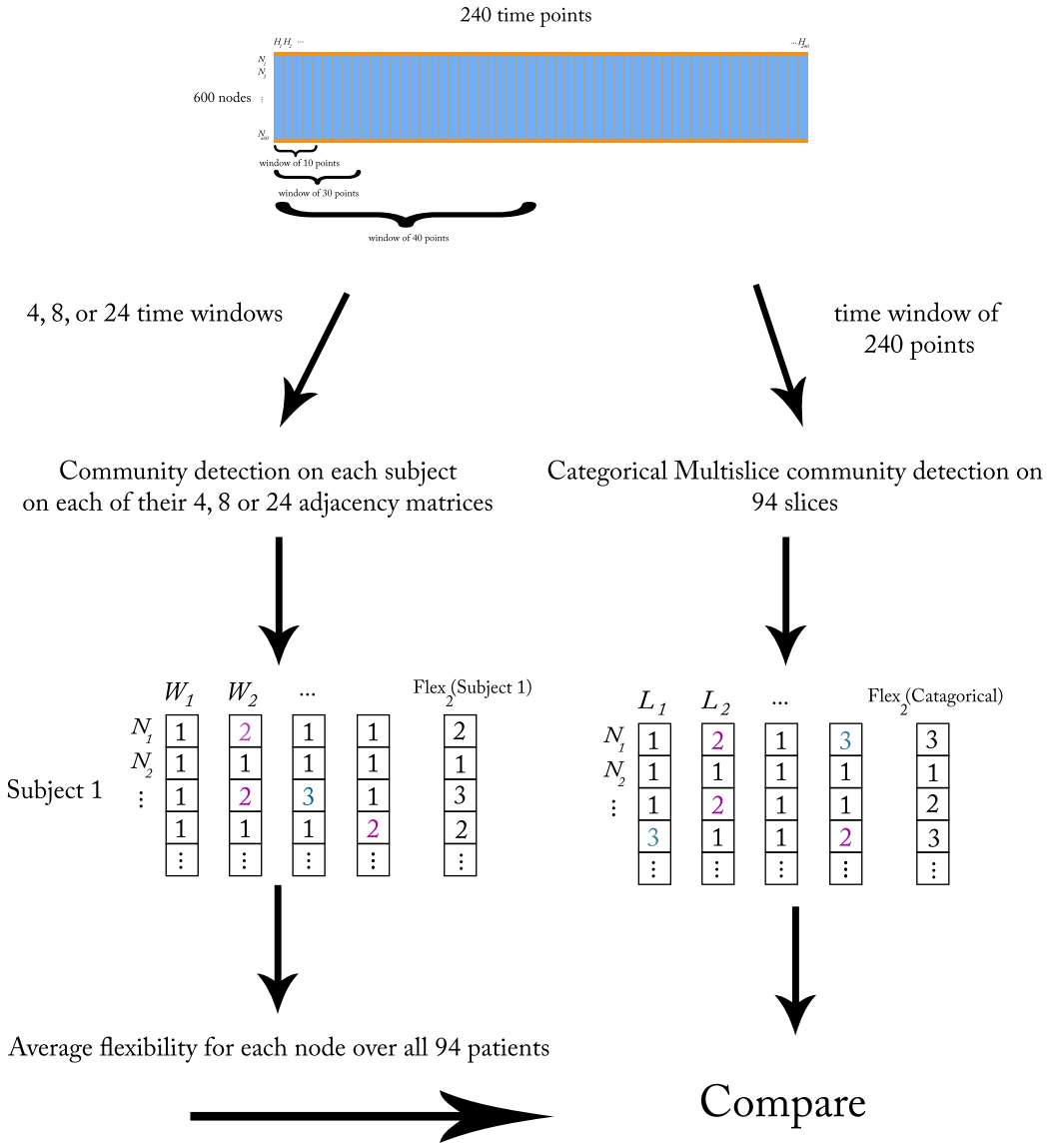

Figure 38: An illustration of the method we use to compare ordinal flexibility and categorical flexibility. Recall Fig. 3 from Section 2.2. The left hand route explores ordinal flexibility; the layers W_i represent time points in the time series (for a single subject). These layers are ordinal as they are ordered with respect to time. The right hand route explores categorical flexibility; the layers L_i represent the adjacency matrices of the subjects. The order of these layers is irrelevant and we can shuffle them as we please. The comparison will be done node-wise, producing a scatter plot of 600 points.

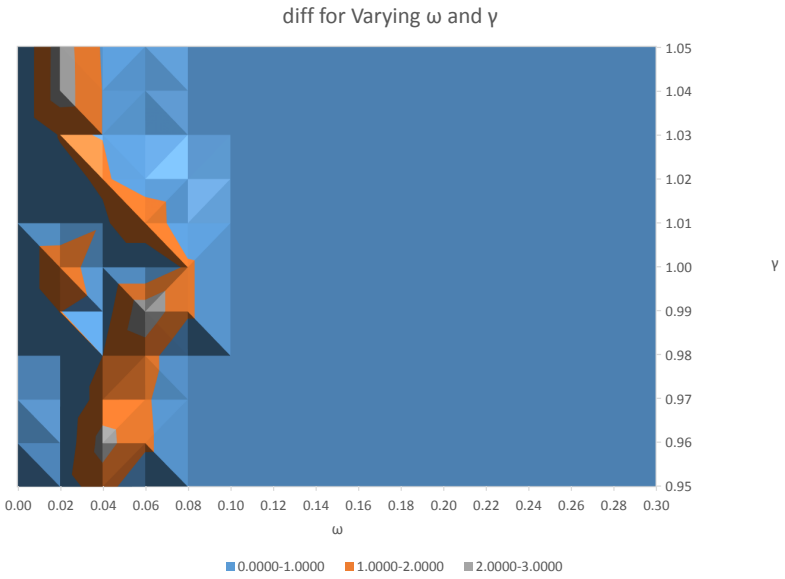

Figure 39: Contour plot of diff for varying ω and γ. Pale blue areas represent no difference between tasks in terms of categorical flexibility; orange areas represent some different and grey areas represent a large difference. We vary γ in steps of 0.01 and ω in steps of 0.02.

for each subject k. We then calculate:

$$\text{diff} = \frac{1}{94}\sum_{k=1}^{94}\text{diff}(k)$$

The results are shown in Fig. 39. We observe that for values of $\omega > 0.1$ we obtain no difference between flexibility across attention 1 task, resting state and memory task 1. We choose $\omega = 0.05$. The parameter γ seems less important and we choose $\gamma = 1.02$. We use these values in all the community detection that follows.

The justification for only observing this particular sub-network is solely due to computing power. Calculating diff(k) for all subjects k and all networks and sub-networks while varying γ and ω would be extremely computationally intensive; it would take many months on a home computer. Thus we choose the TNN sub-network which is of size 128 which increases computational speed, and we only look at three of the tasks.

5.1.1 Standard Flexibility Results

We calculate categorical flexibility using flexibility 2 of the original 600×600 adjacency matrices across all tasks. We also calculate the standard deviation of the flexibility of the nodes (we cannot calculate the standard deviation over the subjects as each node is assigned a single community for each subject). The results are shown in Fig. 40. We observe no distinction between the tasks.

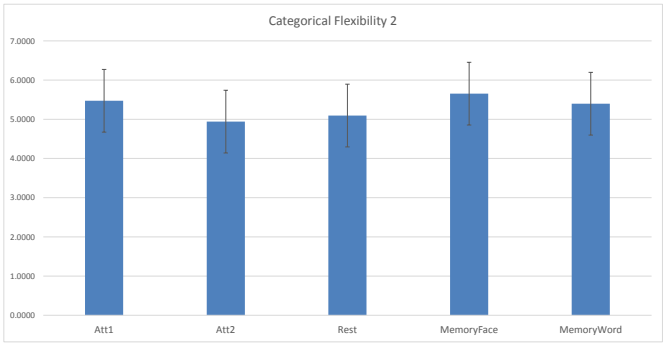

Figure 40: Categorical flexibility using flexibility 2 of the original 600×600 adjacency matrices across all tasks.

We now calculate the categorical flexibility, in the same way as above, of the three sub-networks TPP, TNN and TOO. The results are shown in Fig. 41. We observe a clear distinction between the attention and memory tasks, and also between the resting state task and memory tasks, in the TNN sub-network. We are not able to make a comparison between tasks as the sub-networks have different sizes, and as γ remains constant, a varying number of communities are produced.

We repeat this calculation for the TPN, TPO and TNO sub-networks. The results are shown in Fig. 42. We observe no differences across tasks in any of the sub-networks.

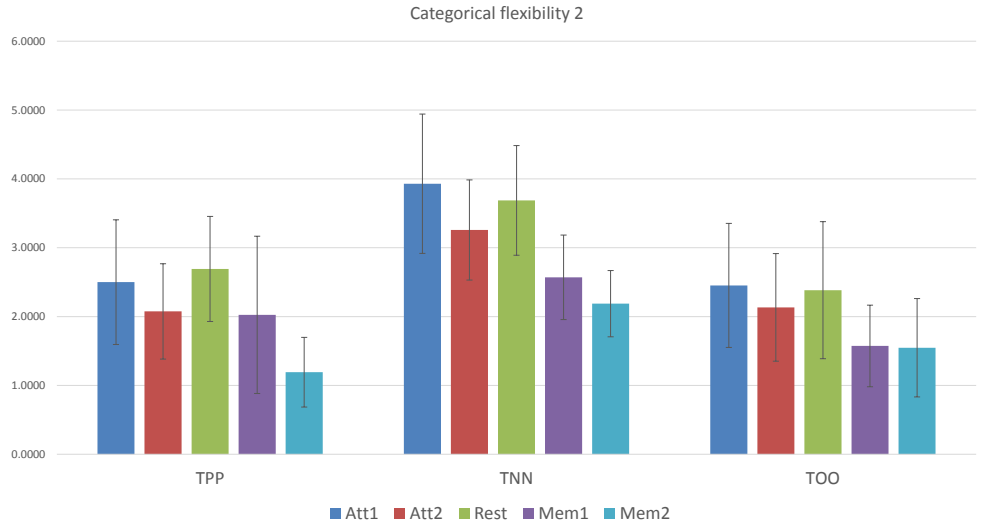

Figure 41: Categorical flexibility using flexibility 2 of the three sub-networks TPP, TNN and TOO across all tasks.

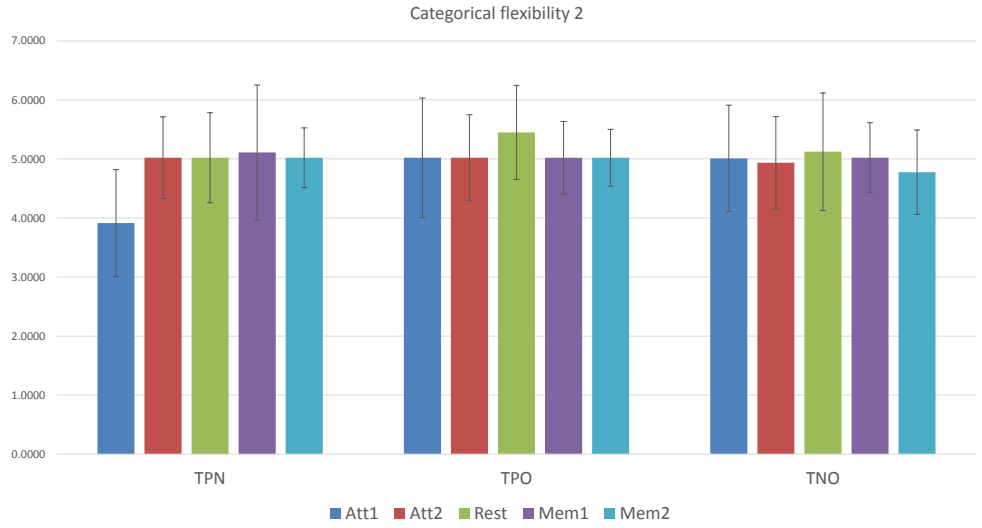

Figure 42: Categorical flexibility using flexibility 2 of the three sub-networks TPN, TPO and TNO across all tasks.

5.1.2 Ordinal Flexibility Versus Categorical Flexibility

We will now attempt to make a comparison between ordinal flexibility and categorical flexibility with the method outlined in Section 5.0.3. We first illustrate the complication that appears when we use the second definition of flexibility, *flexibility 2*. We take 6 time windows of size 40 each, perform single layer community detection on the adjacency matrix corresponding to each time window (for each subject) and calculate the flexibility of each node averaged over all 94 subjects. We then calculate the categorical flexibility by performing the multilayer community detection and then computing the flexibility of each node. Recall this result is for the 600×600 adjacency matrices for attention task 1. The results are shown in Fig. 43. The complication is that when using flexibility 2, we obtain only three possible values for flexibility. The correlation coefficient (calculated using CORRCOEF) is -0.0778 with associated p-value of 0.0568.

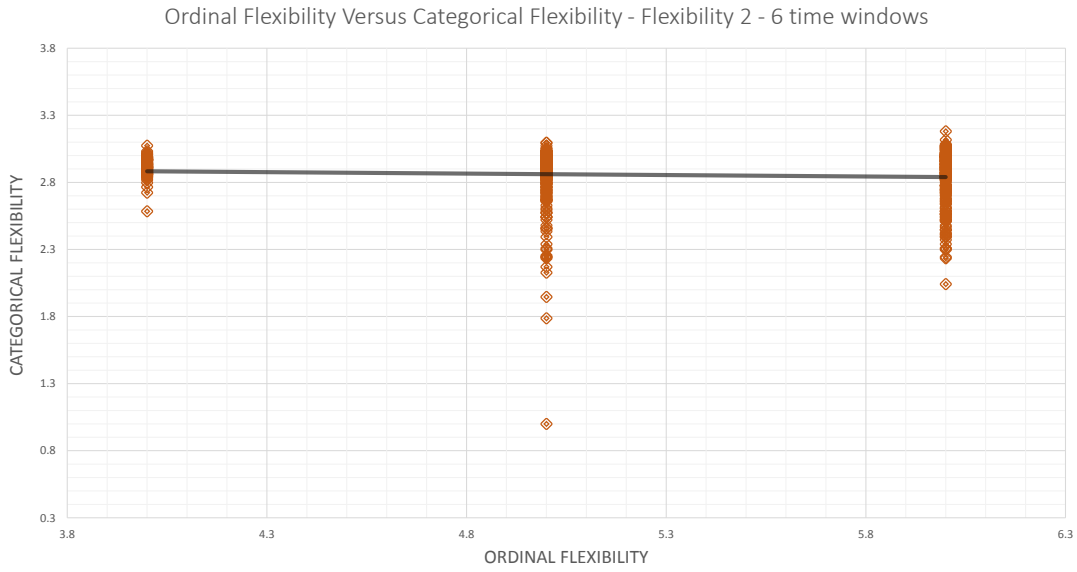

Figure 43: A scatter plot of the 600 nodes in the case of 6 time windows using flexibility 2. The x-axis represents the ordinal flexibility of the respective node; the y-axis the categorical flexibility. A linear line of best fit is shown in black.

We repeat the calculations using flexibility 3 as defined in Section 5.0.3. We take $f(j) = j$, and note that if we increase the strength of this normalisation function, i.e. if we take $f(j) = j^{j^j}$ for example, then each new additional community a node i belongs to increases the flexibility by an increasingly large amount. This isolates the values of flexibility corresponding to how many unique communities each node belongs to. Thus if we took this as our normalisation factor, we would obtain a similar result to Fig. 43. The results for the case $f(j) = j$ are shown in Fig. 44. This time, the points are distributed more appropriately around the chart and we see a small positive correlation. The correlation coefficient is 0.2611 with associated p-value of 0.0001.

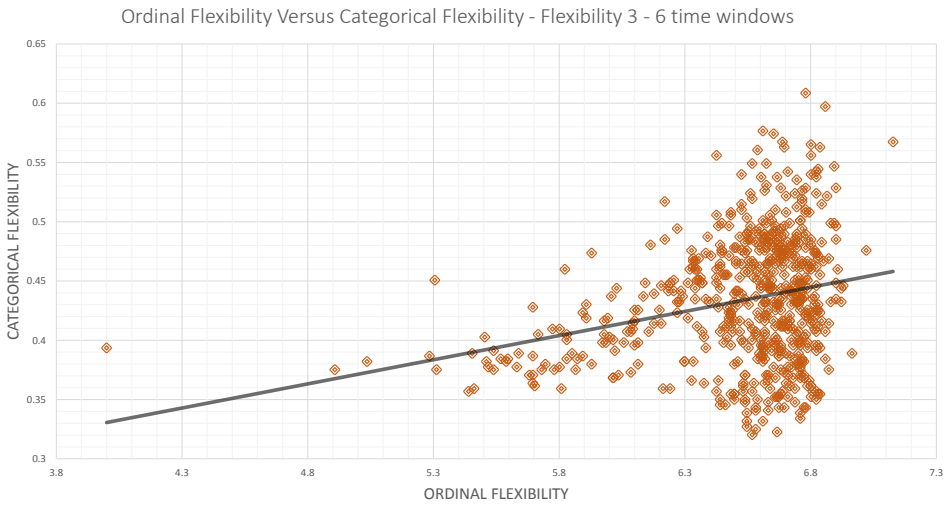

Figure 44: A scatter plot of the 600 nodes in the case of 6 time windows using flexibility 3. The x-axis represents the ordinal flexibility of the respective node; the y-axis the categorical flexibility. A linear line of best fit is shown in black.

We repeat the calculations again, using flexibility 3, but with 24 time windows. We observe similar results to those in the case of 6 time windows. The correlation coefficient is 0.1507 with associated p-value of 0.0002. This process was repeated again for the case of 8 time windows and we again observed similar results.

These results appear to be significant, but we must now compare them to a random-graph null model. We use the null model as described in 5.0.3. We choose 6 time windows, in attention task 1. We perform the randomisation of the time signals and calculate the associated p-value of the correlation between ordinal and categorical flexibility. We perform this process 10 times. We note that we only perform this on the first 50 subjects due to computational constraints[7]. The results are shown in Fig. 46. A typical scatter plot is shown in Fig. 47. We observe no significant results. This support the theory that there is a correlation between ordinal flexibility and categorical flexiblity as hypothesised in Section 5.0.3, however significantly more trails need to be performed.

[7] Multilayer community takes a long time to computer in MATLAB. The computational time increases with the number of layers.

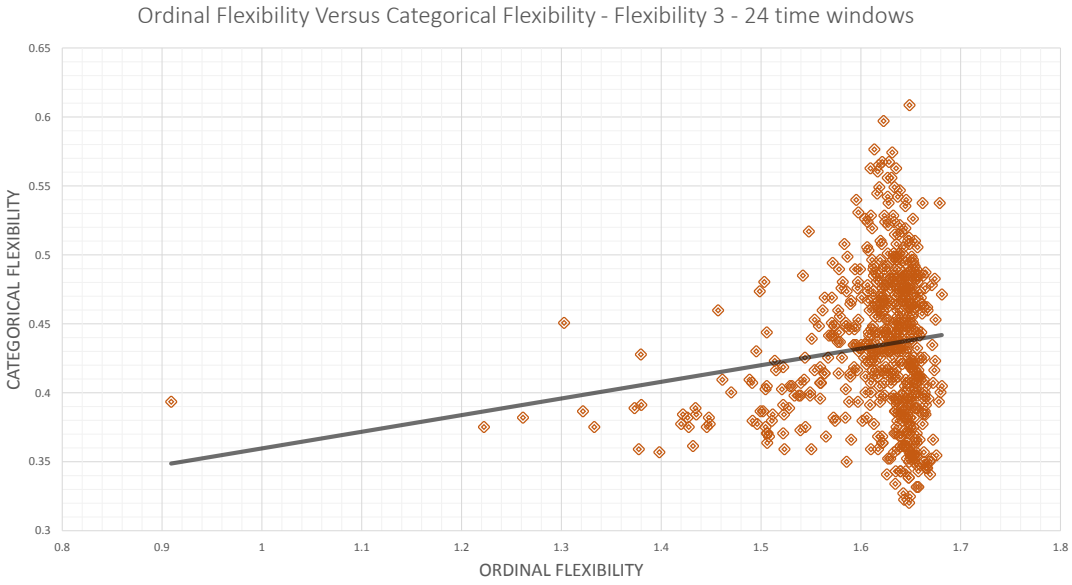

Figure 45: A scatter plot of the 600 nodes in the case of 24 time windows using flexibility 3. The x-axis represents the ordinal flexibility of the respective node; the y-axis the categorical flexibility. A linear line of best fit is shown in black.

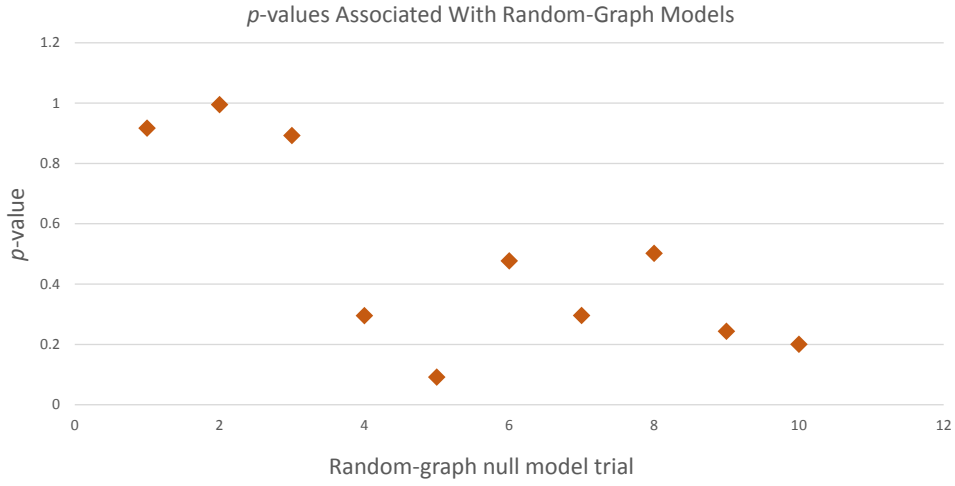

Figure 46: A scatter plot showing the p-values associated with the 10 random-graph models we compute.

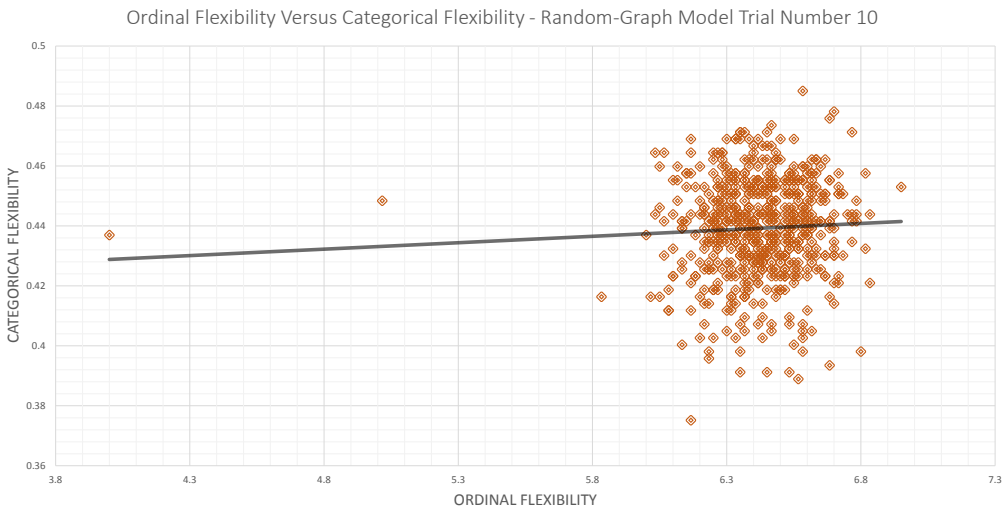

Figure 47: One of the the scatter plots showing the ordinal flexibility (x-axis) and categorical flexibility (y-axis) of each of the 600 nodes.

5.1.3 Categorical flexibility of the Three Node Assignments

We finish with a plot of the average categorical flexibility, using the new definition, of the task-positive, task-negative and "other" nodes across all tasks. The results are shown in Fig. 48. We observe no significant results.

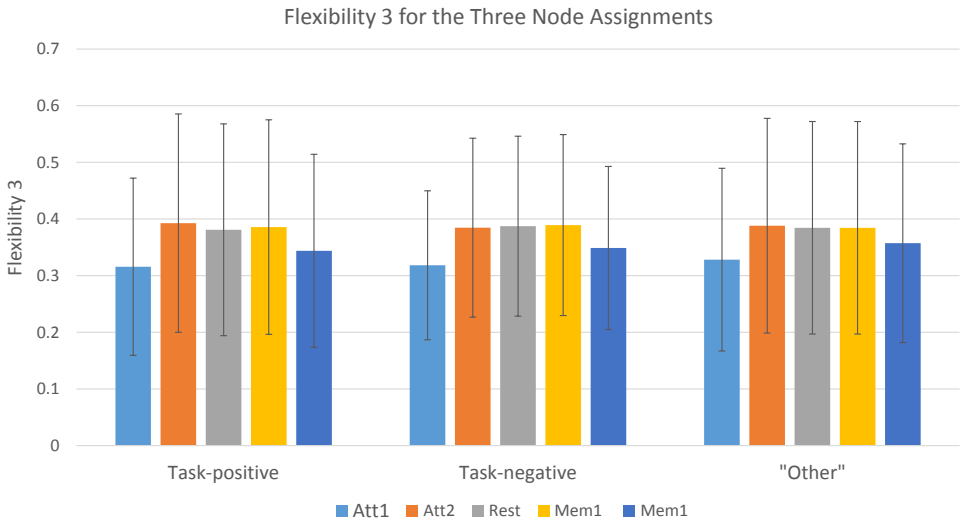

Figure 48: Average values for flexibility 3 for the three node types. The error bars show the variation in each node type.

6 Discussion

The first results explored in Sections 3.1.1, 3.1.2 were not promising, as we found no distinction in both the original 600×600 adjacency matrices and the six sub-networks, across all tasks, based on connectivity and the clustering coefficient alone. Eigenvector centrality, explored in Section 3.1.3 started to show signs of some distinction between the task-positive and task-negative nodes themselves. However, the original aim of Fox et al.[15] was to identify certain networks of the brain associated with certain types of tasks and not just the regions by themselves, hence the eigenvector centrality results do not give us enough information about the *structure* of the networks. However, we can conclude that the task-positive nodes, across all tasks, are generally connected more strongly to other nodes of the same type.

The thresholding technique deployed Section 3.1.4 did not yield any interesting results. We can only conclude that the tasks are not distinguishable between each other in any of the sub-networks, based on the simple diagnostics; connectivity, clustering coefficient and betweenness centrality.

The techniques explored in Section 4.1 yielded more promising results. As stated in Section 4.0.5, the aim of the spectral clustering method is separate a set of objects in to groups such that objects within each group are similar and objects in distinct groups are dissimilar. The method failed to produce a perfect (or almost perfect) separation between the node assignments; however the results

obtained show signs of separation. It therefore suggests there are common properties shared within each category of node. The generalisation of the normalisation in Eq. 8 is justified as we observed in the results. This is because the nature of the communicability values obtained depend upon the size (and to a lesser degree, the average strength) of the network. The original definition given by Crofts et al. [9] is unsuccesful in producing suitable communicability values. However, even with an appropriate value of k (recall we chose $k = 3$) we were unable to make any conclusions on the nature of the six sub-networks related to the task-positive and task-negative node assignments.

The flexibility results obtained from community detection explored in Section 5.1 show some distinction, based on average flexibility, between tasks in the two sub-networks TPP and TNN. We also observed a curious result in Section 5.1.2 which was based upon our generalisation of the notion of flexibility. We noted that this new definition, *flexibility 3,* is more suitable for calculating categorical flexibility than the previously proposed definitions (which were motivated by an ordered time series). With this new definition, we were unable to make any distinctions between task-positive and task-negative nodes in the original 600×600 networks.

7 Conclusion

In this thesis we explored many methods for which only a small portion showed promising results. This confirms R Nathan Spreng's critisisms [42] of the nature of these assignments are well-founded. The promising results here motivate further research into *redefining* these networks, as although other research in to these networks networks have yielded important results [19, 27], the wide variety of methods used in this thesis do not support the original aims of the definition of the task-positive and task-negative subnetworks. However there are many other methods in network science which could be used to explore further research into these assignments. A small selection of these are discussed in Section 7.1.

7.1 Further Investigations

This thesis only explores a small portion of the methods which were used in this project. There are other avenues in which we can explore to analyse the structure of task-positive and task-negative networks.

7.1.1 Spectral Clustering Applied to Directly to Sub-networks

It is possible to use the spectral clustering method outlined in Section 4.0.5 to compare two groups of networks. For example, we could compare the group of 94 TPN networks in attention task 1 to the group of 94 TPN networks in memory task 1. This is done by transforming the data in each network in to a column vector and constructing a matrix $\mathbf{V}$ for which V_{ij} represents the similarity beween networks i and j. The spectral clustering method can then be applied to $\mathbf{V}$. The disadvantage of this method is that there are a large number of combinations of groups of networks we could study and hence analysing the results becomes tricky. This was explored in detail during the project, but needs further work in order to understand the results.

7.1.2 Additional Thresholding Techniques

There exist more thresholding techniques which are able to reveal different structures in networks [28]. We explored only one type of thresholding.

7.1.3 Redefining Sub-networks Based on the Data

It is theoretically possible to define new sub-networks, based on the original aims of the task-positive and task-negative networks, using the network data itself. We can, for example, construct a sub-network consisting of the n strongest nodes in a particular task. We can then reuse the methods in this thesis to explore these new sub-networks.

Acknowledgements

Many thanks to my supervisors: Mason Porter and Sang Hoon Lee, for both introducing me to network science and giving me the possibility to do this project. I thank the Oxford Maths Institute for awarding me with a bursary which enabled me to study networks in Oxford, under supervision by Mason Porter and Sang Hoon Lee, in the summer of 2013, which led me to consider a networks project for my 4th year. I thank Danielle Bassett and Ann Hermundstad for providing me with the data. Thanks to Sang Hoon Lee for producing the plot in Fig. 36 using code from HTTPS://BITBUCKET.ORG/TAYNAUD/PYTHON-LOUVAIN. Thanks to the authors of Ref. [3] for kindly allowing the replication of Fig. 2 and to the authors of Ref. [32] for kindly allowing the replication of Fig. 37. Many thanks to the authors of the "generalized Louvain" MATLAB code[8] which I used to perform the community detection, and to the authors of the WMTSA Wavelet Toolkit for MATLAB[9] which I used to perform the wavelet correlation. I acknowledge that various programs in the Brain Connectivity Toolbox[10] were used as part of my own programs.

[8] HTTP://NETWIKI.AMATH.UNC.EDU/GENLOUVAIN/GENLOUVAIN
[9] HTTP://WWW.ATMOS.WASHINGTON.EDU/~WMTSA/
[10] HTTPS://SITES.GOOGLE.COM/SITE/BCTNET/)

References

[1] C. J. Alpert and S.-Z. Yao. Spectral partitioning: The more eigenvectors, the better. In *Proceedings of the 32nd Annual ACM/IEEE Design Automation Conference*, DAC '95, pages 195–200, New York, NY, USA, 1995. ACM.

[2] D. S. Bassett, E. T. Owens, K. E. Daniels, and M. A. Porter. Influence of network topology on sound propagation in granular materials. *Physical Review E*, 86(4):041306, 2012.

[3] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton. Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, 108(18):7641–7646, 2011.

[4] N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1993.

[5] S. J. Broyd, C. Demanuele, S. Debener, S. K. Helps, C. J. James, and E. J. Sonuga-Barke. Default-mode brain dysfunction in mental disorders: a systematic review. *Neuroscience & Biobehavioral Reviews*, 33(3):279–296, 2009.

[6] E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[7] E. T. Bullmore and D. S. Bassett. Brain graphs: graphical models of the human brain connectome. *Annual Review of Clinical Psychology*, 7:113–140, 2011.

[8] R. C. Craddock, G. A. James, P. E. Holtzheimer, X. P. Hu, and H. S. Mayberg. A whole brain fmri atlas generated via spatially constrained spectral clustering. *Human brain mapping*, 33(8):1914–1928, 2012.

[9] J. J. Crofts and D. J. Higham. A weighted communicability measure applied to complex brain networks. *Journal of the Royal Society Interface*, 6(33):411–414, 2009.

[10] J. J. Crofts, D. J. Higham, R. Bosnell, S. Jbabdi, P. M. Matthews, T. Behrens, and H. Johansen-Berg. Network analysis detects changes in the contralesional hemisphere following stroke. *NeuroImage*, 54(1):161–169, 2011.

[11] E. Estrada and N. Hatano. Communicability in complex networks. *Physical Review E*, 77(3):036111, 2008.

[12] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Physical Review E*, 71(5):056103, 2005.

[13] D. J. Fenn, M. A. Porter, P. J. Mucha, M. McDonald, S. Williams, N. F. Johnson, and N. S. Jones. Dynamical clustering of exchange rates. *Quantitative Finance*, 12(10):1493–1520, 2012.

[14] M. D. Fox and M. E. Raichle. Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nature Reviews Neuroscience*, 8(9):700–711, 2007.

[15] M. D. Fox, A. Z. Snyder, J. L. Vincent, M. Corbetta, D. C. Van Essen, and M. E. Raichle. The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proceedings of the National Academy of Sciences*, 102(27):9673–9678, 2005.

[16] J. Goñi, M. P. van den Heuvel, A. Avena-Koenigsberger, N. V. de Mendizabal, R. F. Betzel, A. Griffa, P. Hagmann, B. Corominas-Murtra, J.-P. Thiran, and O. Sporns. Resting-brain functional connectivity predicted by analytic measures of network communication. *Proceedings of the National Academy of Sciences*, 111(2):833–838, 2014.

[17] M. D. Greicius, B. Krasnow, J. M. Boyett-Anderson, S. Eliez, A. F. Schatzberg, A. L. Reiss, and V. Menon. Regional analysis of hippocampal activation during memory encoding and retrieval: fmri study. *Hippocampus*, 13(1):164–174, 2003.

[18] M. D. Greicius, K. Supekar, V. Menon, and R. F. Dougherty. Resting-state functional connectivity reflects structural connectivity in the default mode network. *Cerebral Cortex*, 19(1):72–78, 2009.

[19] M. Hampson, N. Driesen, J. K. Roth, J. C. Gore, and R. T. Constable. Functional connectivity between task-positive and task-negative brain areas and its relation to working memory performance. *Magnetic Resonance Imaging*, 28(8):1051–1057, 2010.

[20] A. M. Hermundstad, D. S. Bassett, K. S. Brown, E. M. Aminoff, D. Clewett, S. Freeman, A. Frithsen, A. Johnson, C. M. Tipper, M. B. Miller, et al. Structural foundations of resting-state and task-based functional connectivity in the human brain. *Proceedings of the National Academy of Sciences*, 110(15):6169–6174, 2013.

[21] D. J. Higham, G. Kalna, and M. Kibble. Spectral clustering and its use in bioinformatics. *Journal of computational and applied mathematics*, 204(1):25–37, 2007.

[22] K. E. Joyce, P. J. Laurienti, J. H. Burdette, and S. Hayasaka. A new measure of centrality for brain networks. *PLoS One*, 5(8):e12200, 2010.

[23] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*, volume 344. John Wiley & Sons, 2009.

[24] D. P. Kennedy and E. Courchesne. The intrinsic functional organization of the brain is altered in autism. *NeuroImage*, 39(4):1877–1885, 2008.

[25] H. Kim, S. M. Daselaar, and R. Cabeza. Overlapping brain activity between episodic memory encoding and retrieval: roles of the task-positive and task-negative networks. *Neuroimage*, 49(1):1045–1054, 2010.

[26] R. Lambiotte, J.-C. Delvenne, and M. Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*, 2008.

[27] H. Liu, Y. Kaneko, X. Ouyang, L. Li, Y. Hao, E. Y. Chen, T. Jiang, Y. Zhou, and Z. Liu. Schizophrenic patients and their unaffected siblings share increased resting-state connectivity in the task-negative network but not its anticorrelated task-positive network. *Schizophrenia Bulletin*, 38(2):285–294, 2012.

[28] C. Lohse, D. S. Bassett, K. O. Lim, and J. M. Carlson. Resolving structure in human brain organization: Identifying mesoscale organization in weighted network representations. *ArXiv e-prints*, Dec. 2013.

[29] M.-E. Lynall, D. S. Bassett, R. Kerwin, P. J. McKenna, M. Kitzbichler, U. Muller, and E. Bullmore. Functional connectivity and brain networks in schizophrenia. *The Journal of Neuroscience*, 30(28):9477–9487, 2010.

[30] A. V. Mantzaris, D. S. Bassett, N. F. Wymbs, E. Estrada, M. A. Porter, P. J. Mucha, S. T. Grafton, and D. J. Higham. Dynamic network centrality summarizes learning in the human brain. *Journal of Complex Networks*, 1(1):83–92, 2013.

[31] M. Mennes, C. Kelly, X.-N. Zuo, A. Di Martino, B. B. Biswal, F. X. Castellanos, and M. P. Milham. Inter-individual differences in resting-state functional connectivity predict task-induced bold activity. *NeuroImage*, 50(4):1690–1701, 2010.

[32] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[33] M. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[34] T. A. Niendam, A. R. Laird, K. L. Ray, Y. M. Dean, D. C. Glahn, and C. S. Carter. Meta-analytic evidence for a superordinate cognitive control network subserving diverse executive functions. *Cognitive, Affective, & Behavioral Neuroscience*, 12(2):241–268, 2012.

[35] J.-P. Onnela, J. Saramäki, J. Kertész, and K. Kaski. Intensity and coherence of motifs in weighted complex networks. *Physical Review E*, 71(6):065103, 2005.

[36] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

[37] M. E. Raichle. The brain's dark energy. *Scientific American*, 314(5803):1249, 2006.

[38] M. E. Raichle, A. M. MacLeod, A. Z. Snyder, W. J. Powers, D. A. Gusnard, and G. L. Shulman. A default mode of brain function. *Proceedings of the National Academy of Sciences*, 98(2):676–682, 2001.

[39] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, and J. Kertesz. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105, 2007.

[40] G. L. Shulman, J. A. Fiez, M. Corbetta, R. L. Buckner, F. M. Miezin, M. E. Raichle, and S. E. Petersen. Common blood flow changes across visual tasks: Ii. decreases in cerebral cortex. *Journal of cognitive neuroscience*, 9(5):648–663, 1997.

[41] S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich. Network modelling methods for fmri. *NeuroImage*, 54(2):875–891, 2011.

[42] R. N. Spreng. The fallacy of a task-negative network. *Frontiers in Psychology*, 3(145), 2012.

[43] M. Tudor, L. Tudor, and K. Tudor. Hans berger (1873-1941)–the history of electroencephalography. *Acta medica Croatica: casopis Hravatske akademije medicinskih znanosti*, 59(4):307–313, 2004.

[44] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *NeuroImage*, 15(1):273–289, 2002.

[45] J. L. Vincent, I. Kahn, A. Z. Snyder, M. E. Raichle, and R. L. Buckner. Evidence for a frontoparietal control system revealed by intrinsic functional connectivity. *J Neurophysiol*, 100(6):3328–3342, 2008.

[46] R. W. Williams and K. Herrup. The control of neuron number. *Annual Review of Neuroscience*, 11(1):423–453, 1988.

[47] W. Zachary. An information flow modelfor conflict and fission in small groups1. *Journal of anthropological research*, 33(4):452–473, 1977.

[48] Q. Zhu, J. Zhang, Y. L. Luo, D. D. Dilks, and J. Liu. Resting-state neural activity across face-selective cortical regions is behaviorally relevant. *The Journal of Neuroscience*, 31(28):10323–10330, 2011.