Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Silje Marie Anfindsen

A statistical framework for identifying neuroplasticity after stroke

Master's thesis in Applied Physics and Mathematics
Supervisor: Benjamin Adric Dunn
Co-supervisor: Riccardo Iandolo
June 2022

NTNU
Norwegian University of
Science and Technology

Silje Marie Anfindsen

A statistical framework for identifying neuroplasticity after stroke

Master’s thesis in Applied Physics and Mathematics
Supervisor: Benjamin Adric Dunn
Co-supervisor: Riccardo Iandolo
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Preface

This work concludes my five-year Master's programme in Applied Physics and Mathematics at the Norwegian University of Science and Technology (NTNU). The thesis is my assignment for the course TMA4900 - Industrial Mathematics Master's Thesis and was carried out in the spring semester of 2022 as a continuation of my Project Thesis for the course TMA4500, in the autumn semester of 2021.

The thesis is written in cooperation with a recently initiated study at St.Olav Hospital on stroke rehabilitation. The thesis began as a contribution to applying and developing mathematical and computational methods to predict how stroke rehabilitation can rearrange the neural network connections to improve and facilitate the restoration of lost motor function after stroke. The recruitment of stroke patients for this project has been difficult in the preceding year, influencing my thesis. However, with the support and assistance of my wonderful supervisor, Benjamin Adric Dunn, we have always been able to come up with a solution and new exciting problems to investigate. I would also like to thank my co-supervisor, Riccardo Iandolo, for guidance in this new, exciting field, neuroscience. Your contributions with knowledge, creative ideas and help have been crucial and are well appreciated. I am very grateful for the new insight into neuroscience and statistics that this thesis has given me, and I wish you and this project tremendous success in the future.

Finally, I want to express my gratitude to the most important people around me for their unwavering support and assistance. You are well aware of who you are. Thank you!

Trondheim, 17th June, 2022

....

Silje Marie Anfindsen

Abstract

The brain is a fascinating organ—billions of electric signals between neurons transport information continuously. Using functional Magnetic Resonance Imagining (fMRI), we can approximate these processes. We consider a network of nodes and edges reflecting groups of neurons and their functional connections, respectively. The nodes are defined according to a partition of the human brain, referred to as a brain *parcellation.* The edges are computed using a statistical dependence of fMRI signals from the nodes, known as functional connectivity (FC). During a stroke, parts of the brain network become damaged. *Neuroplasticity,* or the brain's ability to *rewire* itself, is presumed to occur after a stroke. Rewiring entails the formation of new connections or the reinforcement of existing ones. This thesis investigates whether we can detect rewiring using fMRI. An understanding of this process can contribute better to facilitating the restoration of lost motor function after stroke. We examined three fMRI scans performed over three months on healthy elderly participants in resting state (rs). We have established a statistical framework to ensure that observed changes in FC after stroke are associated with recovery. The framework includes a Test-Retest (trt) reliability evaluation of a variety of brain parcellations and FC methods and a prediction of how many patients should be recruited for a stroke rehabilitation study at St.Olav Hospital to identify FC changes after stroke. As far as we know, the framework, which can readily incorporate new methodologies and data quality, does not exist. Results reveal that the estimated brain networks cannot provide a reliable conclusion on the number of required stroke patients. This is due to fMRI noise, violation of model assumptions, and a limited number of control subjects. Nevertheless, the estimated sample sizes indicate that the stroke rehabilitation study will most likely be unable to enrol enough stroke patients.

Sammendrag

Hjernen er et fascinerende organ – milliarder av elektriske signaler mellom nevroner transporterer kontinuerlig informasjon. Ved å bruke funksjonell magnetresonansavbildning, også kalt fMRI, kan vi modellere disse prosessene. Vi tar for oss et nettverk av noder og kanter som reflekterer grupper av nevroner og deres funksjonelle forbindelser. Kantene beregnes ved å bruke en statistisk avhengighet av fMRI-signaler innenfor hver node, kjent som funksjonell konnektivet. Under et hjerneslag kan deler av hjernen bli skadet. *Nevroplastisitet*, hjernens evne til å *reorganiseres*, er antatt å oppstå i etterkant av et slag. Reorganisering betyr at nye forbindelser kan dannes, og eksisterende forsterkes. I denne masteroppgaven undersøker vi om det er mulig å observere denne reorganiseringen ved bruk av fMRI. En forståelse av denne prosessen kan bidra til å forbedre gjenoppretting av tapt motorisk funksjon etter hjerneslag. Vi undersøkte tre fMRI skanninger utført over tre måneder på friske eldre deltakere i hvilende tilstand. Vi har opprettet et statistisk rammeverk for å være sikker på at observerte endringer i funksjonell konnektivet etter slag representerer reorganisering. Rammeverket evaluerer påliteligheten til ulike modeller av det nevrale nettverket og predikerer hvor mange pasienter som bør rekrutteres til et studie på St.Olav Hospital for å oppdage reorganisering. Videre kan rammeverket inkludere nye metoder og datakvalitet, og eksisterer så vidt kjent ikke i dag. Resultatene viser at de estimerte nettverkene ikke er i stand til å produsere pålitelige prediksjoner av antall nødvendige slagpasienter. Dette er på grunn av fMRI støy, brudd på modellantagelser og et begrenset antall deltakere i studiet. Imidlertid indikerer de estimerte utvalgsstørrelsene at studiet sannsynligvis ikke vil være i stand til å rekruttere nok slagpasienter.

Acronyms

ACF Autocorrelation function.

BOLD Blood-Oxygen-level-dependent.

CV cross validation.

DCM Dynamical causal model.

EM Expectation maximization.

FC functional connectivity.

fMRI functional Magnetic Resonance Imagining.

FSL FMRIB Software Library.

GLasso Graphical lasso.

GLM Generalized linear model.

lasso least absolute shrinkage and selection operator.

LH left hemisphere.

LS least squares.

MAP Maximum a Posterior.

MLE Maximum Likelihood Estimation.

MNI Montreal Neurological Institute.

MR Magnetic Resonance.

MRI Magnetic Resonance Imaging.

PC principal component.

PCA Principal component analysis.

RH right hemisphere.

ROI region of interest.

rs resting state.

spDCM Spectral Dynamic causal modelling.

SPM Statistical Parametric Mapping.

TR reptition time.

trt Test-Retest.

Contents

Chapter 1

Introduction

This first chapter starts with presenting the problem description along with the motivation and main objectives of this thesis. A brief introduction to neuroscience follows in Section 1.2, providing the reader with an understanding of the data utilized in this study. Lastly, an outline of the following chapters of this thesis is presented.

1.1 Problem Description and motivation

Networks are commonly utilized to describe real-life phenomena, such as social relations, economic dependencies or the world wide web (better known as the internet). Fundamentally, networks are all composed of nodes and edges. The nodes are identifiable entities that interact with other nodes. The connection that binds the nodes together is called edges. How we characterize these two components largely influences our perception of the underlying system represented by the network [1].

The neural network, a system of dynamic interactions between neurons in our brain, is one fascinating network. Similarly, as your social network, linking people you have a relationship to, changes yearly or even daily, the neural connections in the brain can change. The brain‘s ability to *rewire* itself is called *neuroplasticity*. Rewiring refers to either a production of new or strengthening existing connections. Neuroplasticity is believed to occur in a damaged neural network, resulting from the reduced blood supply following a stroke [2]. However, recent studies on resting state (rs) functional Magnetic Resonance Imagining (fMRI) of humans following a stroke, yield conflicting results [3] [4]. The primary objective of this thesis is to investigate whether it is possible to detect rewiring related to stroke recovery using rs fMRI. We consider a large-scale neural network where the nodes represent groupings of neurons. The nodes are connected through edges whose weights and direction are determined by the statistical dependence of each node‘s fMRI response signal, known as functional connectivity (FC) [5]. We refer to this system as a *functional brain network*.

Nodes and edges are clearly defined in most networks that model physical systems. For the social network mentioned above, individual persons and their friendships are obvious choices of nodes and edges. It is currently impossible to imagine a full-scale replica of the human brain, with nodes and edges reflecting the individual neurons and their interactions. Functional neuroimaging technology is limited to nodes larger than a millimetre. Consequently, there is no consensus on defining the nodes of a functional brain network at present [6]. In fMRI research, this results in a wide range of distinct partitions of the human brain, known as *parcellations schemes* [7]. Furthermore, there are also a variety of approaches for inferring FC, resulting in various edges in terms of weight and direction, using the same fMRI data [8]. Low reproducibility is one of the most severe consequences of the absence of standardized pipelines for analyzing neuroimaging data. The issue was brought to light by a recent study in which seventy teams were given identical fMRI datasets but produced varied results due to various analysis procedures [9]. A first step in addressing the reproducibility problem is to evaluate how consistent measurements between scans of the same individuals are, known as Test-Retest (trt) reliability. According to Elliott et al. fMRI research examining trt reliability of their analysis methodologies before predicting clinical outcomes are few [10]. Another aspect of the reproducibility difficulty in neuroscience is low statistical power, the likelihood that statistically significant findings reflect true effects [11]. Statistical power deficiencies have been related to small sample sizes of participants [12]. According to Poldrack et al., researchers should justify a required sample size before initiating clinical trials to boost statistical power and, as a result, reduce false-positive findings [13].

We investigate and evaluate different models of a functional brain network to detect neuroplasticity after stroke. That involves different parcellation schemes to define the nodes and different measures of FC to define the edges. The models are further incorporated into a framework which is utilized to (1) evaluate the trt reliability of the methods to ensure that observed abnormalities following stroke represent rewiring; and (2) decide how many patients are needed to observe significant longitudinal changes in FC, following a stroke. The outcomes of this thesis are being used to determine the prospects of a current trial at St.Olav Hospital aimed at improving motor recovery after stroke [14].

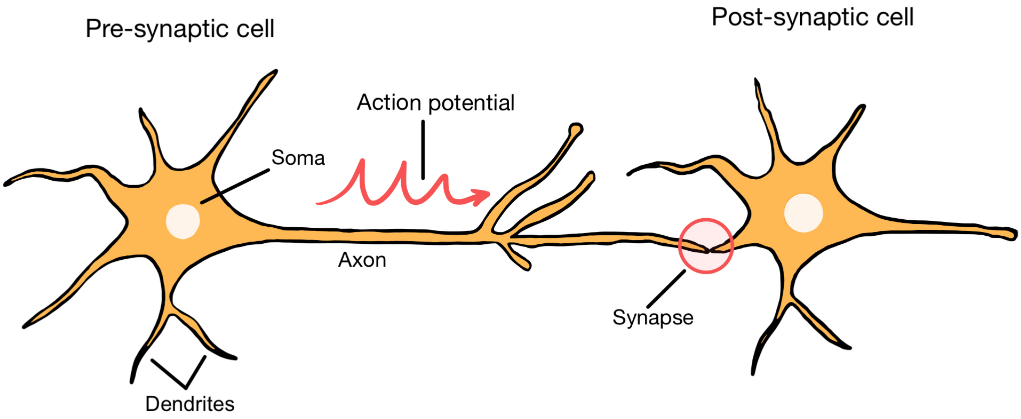


Figure 1.1: The synapse between two neurons. This is where the action potential is transported from the axon of a presynaptic (sending) neuron and to the dendrites of a postsynaptic (receiving) neuron. The action potential is further sent to the soma before possibly continuing its journey.

1.2 Background in Neuroscience

Neuroscience is the study of brain function and structure. Basic concepts within this field are here presented on microscopic and macroscopic levels. Section 1.2.1 gives a biological presentation of the individual neurons and their synapses. Section 1.2.2 examines the whole brain network and gives insight into how fMRI is related to neural activity. Finally, Section 1.2.3 describes the mechanisms behind stroke recovery and synaptic plasticity.

1.2.1 The Neuron

The fundamental functional units of the human brain, in charge of processing environmental inputs, are called nerve cells or equivalently neurons. Understanding the composition of such cells is helpful before analysing the connections in the complete neural network.

The neuron consists of three main components: the cell body or *soma*, the *dendrites* and the *axon*, as illustrated in Figure 1.1. The dendrites are the neuron‘s input that receives signals from other neurons and transports them back to the soma. The axon is the output structure that carries the impulses away from the cell body. These two components of the neurons continuously send and receive electrical messages to each other across gaps, called *synapses* [15].

The area between the inside and outside of a nerve cell is called the *membrane*. Positive and negative ions travel in and out of the membrane, leading to a voltage difference between these two areas called the *membrane potential*. The membrane potential is responsible for carrying the electrical signal throughout the nerve cell. Chemical *neurotransmitters* are released at the very end of the nerve cell

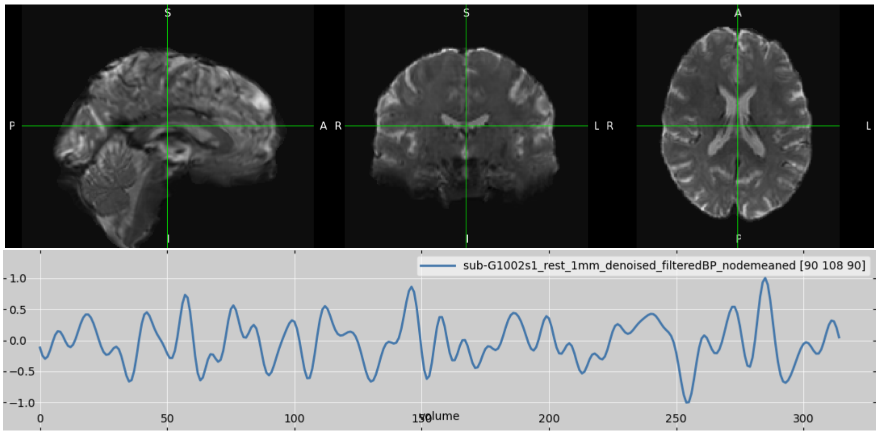

S
P
A
R
I
S
L
R
I
A
L
P
sub-G1002s1_rest_1mm_denoised_filteredBP_nodemeaned [90 108 90]
1.0
0.5
0.0
−0.5
−1.0
0
50
100
150
200
250
300
volume

Figure 1.2: A functional image of a healthy brain at one specific time point. The brain is visualised in all three anatomical planes: sagittal (left image), coronal (centre image) and axial (right image) [20]. The time series represents the preprocessed, normalised Blood-Oxygen-level-dependent (BOLD) signal from a voxel whose location is shown by the green cursor. The sampling rate is TR = 1.53 seconds.

in order to transport the electrical signals between the gap of adjacent neurons. Ion channels open when neurotransmitters bind to the dendrite receptors belonging to the next neuron. This opening results in a change in the membrane potential of the receiving cell. With a net influx of negative ions, the synapse is called *inhibitory*. Oppositely, for a net influx of positive charge, it is called *excitatory*. This membrane potential is constantly changing, and when it reaches a specific threshold, an *action potential* is created in the axon. The neuron is said to *fire a spike*, signalling that it is active and ready to communicate with new nerve cells [16].

1.2.2 The Brain Network

The human brain is assumed to contain about 86 billion neurons [17], each communicating with possibly thousands of other nerve cells. The number of synapses per neuron can vary between $1,000$ and $10,000$, implying up to a quadrillion connections in the entire brain [18]. A full-scale human brain with nodes and edges corresponding to individual neurons and their structural, axonal connections is computationally costly and difficult to comprehend. As a result, we examine groups of neurons, each corresponding to distinct, separated brain regions that can be functionally connected. The fMRI derived FC is widely used to study the operational structure of the human brain [19]. This section aims to provide the reader with a brief overview of fMRI and how to interpret the output signal.

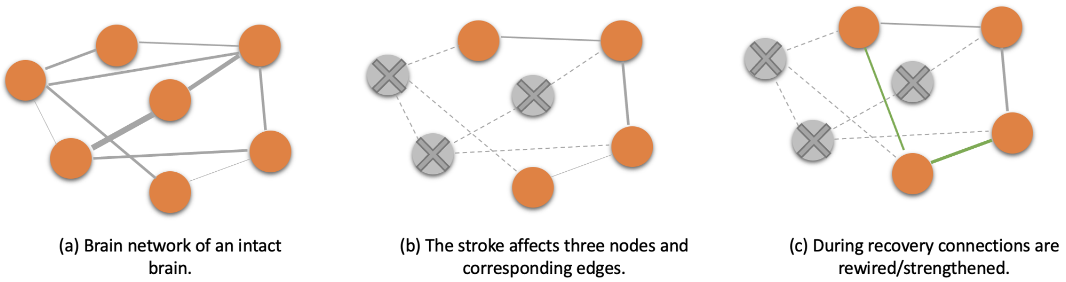
Structural Magnetic Resonance Imaging (MRI) detects the different tissues of the brain, areas of axons and areas of somas [21]. A strong magnetic field surrounds the brain, and the protons of the hydrogen nuclei, most of which are in water molecules, align their spin in the presence of this field and absorb energy to create what is known as an Magnetic Resonance (MR) signal. The distribution of this signal, which varies for the two different tissues, can be transformed from the frequency domain into image space via inverse Fourier transforms [22]. The MRI scanner acquires several slices of the brain, which are stacked to create a 3D image of the entire brain, called a *volume*. The MR signal depends on the blood's oxygen level, which can be mapped over time. Active brain cells need oxygen to produce energy through glucose metabolism, supplied by an increased blood flow. Because of this increased demand for oxygen, blood in areas where there is brain activity is more oxygenated than in other parts. Since oxygen-rich and oxygen-poor blood have different magnetic properties, the MR signal is an indirect measure of neural activity, called the BOLD signal [23]. Functional MRI acquires several volumes of the entire brain with a sampling rate of 1-2 seconds, referred to as repetition time (TR). Each volume contains a grid consisting of several uniformly spaced cubic elements called *voxels*. The voxels represent the spatial distribution of the nuclear spin in an area. Figure 1.2 visualizes a functional scan of a control subject and the preprocessed BOLD signal across all volumes for given voxel.

The exact relationship between the BOLD signal and the underlying neural activity is unclear [19] [24]. However, previous studies have shown that the fMRI response signal is proportional to average neuronal firing rates [25] [26]. Findings from Logothetis et al. [19] suggest that local field potentials yield a better estimate of the BOLD responses compared to the spiking activity of a neural population. The local field potential is believed to reflect the weighted average of synchronized current from the dendrites contained in a larger volume of tissues [24].

Connectivity research aims to understand the *strength*, *directness* and *directionallity* of the edges in a brain network. Weights denote the strength of the functional connection between two neuronal entities. The term "directness" refers to whether a third, hidden node influences any of the two other nodes (indirect connection) or if the link between the two nodes is immediate and undisturbed (direct connection). Ultimately, a connection between nodes A and B is bidirectional or directed if it defines whether the activity travels from A to B or B to A [27]. Generally, FC does not imply causality. Two neural events A and B, are said to causally interact when A results from event B and B's existence is required for A to be present [27]. A more complex technique of measuring the underlying, causal interactions between brain regions is referred to as *effective* connectivity. Complexity stems from the fact that effective connection is heavily reliant on determining the real link between brain activity and the observable BOLD signal [5], which is yet unknown, as discussed above. The effective connections are directed and contain strength and direction. The Dynamical causal model (DCM), introduced in the next chapter, is one such method. According to Horwitz [28], functional and effective connectivity definitions appear inconsistent across studies. Effective connectivity is in this work treated as a subset of FC. This follows from the fact that both metrics represent the similarity between different BOLD time series but differ in how the edges are inferred [27]. Furthermore, in this work, we study the *cerebral cortex*, which is the brain's outermost layer.

1.2.3 Stroke recovery

Neurons are dependent on energy from nutrients and oxygen provided by the blood to maintain synaptic activity, as mentioned in the last section. When blood flow to a portion of the brain is disrupted, the affected cells will die. This medical incident is called a stroke. A usual result is an impairment in function depending on what the damaged brain cells used to control. According to the World Stroke Organization, a stroke will affect 1 in 4 adults above the age of 25 during their lifetime [29]. Annually, approximately 15 million people are suffering from a stroke worldwide; 5 million die and 5 million are left permanently disabled [30]. The stroke rehabilitation study at St.Olav Hospital, to which this thesis is connected, recruits patients suffering from their first stroke resulting in arm motor deficits [14]. The study‘s objective is to investigate if there are any significant associations between improved motor behaviour and changes in brain network functional connectivity.


(a) Brain network of an intact brain.

(b) The stroke affects three nodes and corresponding edges.

(c) During recovery connections are rewired/strengthened.

Figure 1.3: An illustration of stroke is believed to impact the functional brain network. The nodes indicate brain areas, whereas the edges represent functional connections, whose thickness determines interaction strength. The network in (a) is intact, representing a healthy brain. The grey nodes in (b) reflect the lesion or injured brain regions. Tissue that surrounds the stroke core region may be damaged [2]. Following a stroke, connections to these locations may become dysfunctional. These are the grey dotted lines. During the first several months following a stroke, connections might rewire; new connections are formed, or old ones are strengthened (c). The rewired connections are coloured green.

The human brain changes as we develop as human beings during our entire lives. Neuroplasticity, as previously described, is the brain's ability to change its structure and function [31]. A varying amount of synaptic connections are lost as a cause of stroke. Fortunately, the human brain's remaining intact circuits can *rewire*. We refer to rewiring of the brain network as the development of new or strengthening of old functional connections. Figure 1.3 depicts the brain network before and immediately after a stroke, as well as throughout rewiring. We believe that it follows that stroke patients can restore impaired function effectively by maximizing neuroplasticity through correct recovery. Recovery refers to behavioural compensations provided by remaining or new brain connections, leading to improved functional performance. Murphy et al. [2] claim that the human brain is highly plastic during a time-limited window following a stroke. Increased knowledge about the changes in pattern and strength of FC during stroke recovery can therefore contribute to optimizing the rehabilitation of the thousands of stroke incidents occurring annually.

1.3 Outline

Chapter 2 is divided into three parts, presenting (1) how the nodes are defined along with data processing, (2) the three FC methods yielding the edges of the network; and (3) a framework for how to examine trt reliability and conducting power tests to answer our research question. The results obtained for each of the three parts of the preceding chapter are then presented in Chapter 3. The chapter includes presentations of each FC method‘s brain network and an assessment of how various parcellations affect the trt reliability and the predicted sample sizes of each method. Finally, in Chapter 4, the impacts of key findings are examined and compared to similar research. In addition, certain notable limitations and how they might be improved in the future are discussed. This chapter comes to a close with a conclusion.

Chapter 2

Materials and Methods

The purpose of this chapter is to cover data processing and theory, both of which are required to comprehend the measures of FC, also referred to as connectivity measures. The nodes are first defined, followed by a presentation of how the fMRI response signals are retrieved from each node. Then each of the three methods for inferring node interactions is described. Finally, a framework is established for calculating trt reliability and the estimated number of patients necessary to identify post-stroke rewiring.

2.1 Data processing: Defining the nodes

2.1.1 Subjects

The dataset used in this work belongs to an ongoing clinical trial, Personalised rehabilitation of stroke patients evaluated with multimodal quantitative tools (PROTEQT) [14], located at St.Olavs Hospital in Trondheim, Norway. The project has, before the 1st of June 2022, enrolled 32 volunteers; 28 healthy control subjects without previous central nervous system disease or trauma, and 4 stroke patient admitted to the stroke unit, St Olav‘s Hospital, Trondheim, Norway, without previous stroke, neurological disease or central nervous system trauma. The subjects undergo three fMRI sessions with inter-session intervals equal to one and three months between sessions 1 to 2 and sessions 1 and 3, respectively. Recall that we want to calculate the Test-Retest (trt) reliability, which requires many scans of the same subject. As a result, only the individuals who have undergone 3 scanning sessions, as listed in Table 2.1, are considered in this study.

Subject nr.	Age	Sex	Handedness
G1002	64	F	R
G1014	76	M	R
G1018	74	F	R

Table 2.1: Control subject demographics. Data is acquired at St.Olav Hospital.

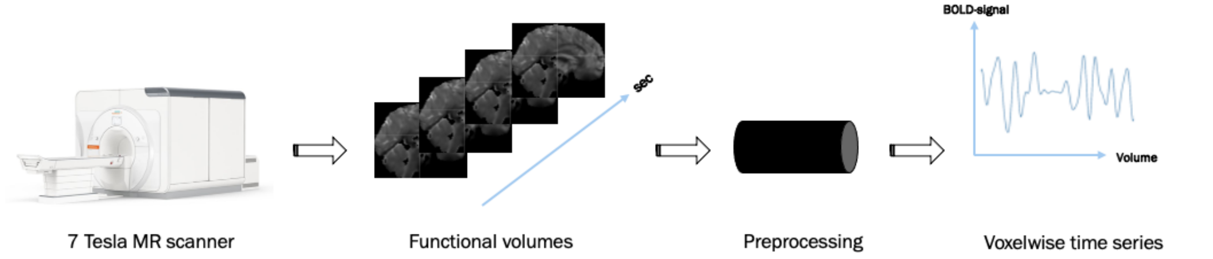


Figure 2.1: An illustration of the data pipeline. Brain volumes are collected and preprocessed from the 7 Tesla MR scanner. The volumes, which are 3D images of the brain, are acquired with a sample rate of 1.53 s. There are 315 volumes corresponding to 8 minutes of scan duration. The volumes contain equal grids of voxels. The voxels hold a Blood-Oxygen-level-dependent signal, which is preprocessed according to Appendix B.1.

2.1.2 Data acquisition and preprocessing

A 7 Tesla Siemens MAGNETOM Terra System was used to obtain up to three sessions of resting state fMRI for each participant – that is where the subject is lying in the scanner, at rest, eyes open. Each session consists of $T = 315$ functional volumes of the whole brain, acquired with a TR = 1.53s. The volumes are divided into a grid of 1.5mm × 1.5mm × 1.5mm voxels. The voxels each contain a BOLD signal defined in Section 1.2.2. The signals are preprocessed and extracted from each voxel. Appendix B.1 contains information on the preprocessing procedures. The pipeline from data acquisition to the extraction of the preprocessed BOLD signal vectors of each voxel is depicted in Figure 2.1.

2.1.3 Cortical parcellation and signal extraction

Due to the demanding computational load required to assess a brain network of nodes represented by individual voxels, we parcellate the cortex into larger regions before extracting the BOLD signals. The brain regions are functionally and structurally separate, each containing a different number of voxels that are presumed to be functionally homogeneous. An *atlas* contains instructions on parcellate the brain. There are a multitude of brain parcellations, or atlases, as discussed in Chapter 1. The fMRI data in this work are parcellated according to the Kong-Schaefer atlas [32]. This functional atlas clusters similar voxels in terms of their functional connections. The clusters are referred to as *parcels*. Figure 2.2 illustrates how the left hemisphere can be parcellated into smaller brain regions. Nodes refer to either an individual parcel or groups of parcels, called *networks*. The number of voxels in

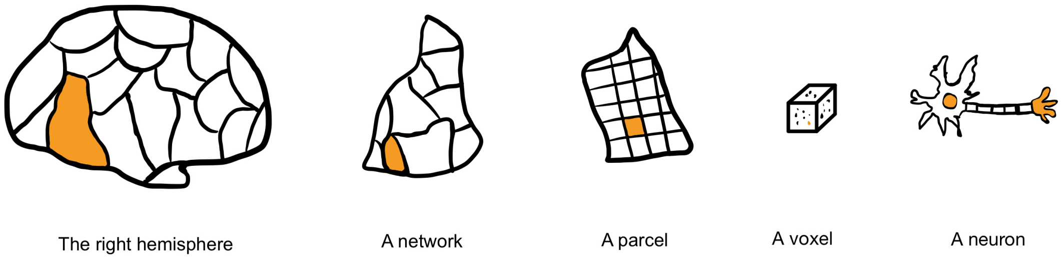


Figure 2.2: An illustration of the parcellation process of the right hemisphere (RH). According to Yeo [33], the hemispheres may be classified into 17 functional networks. Parcels are stored in the networks. Each network has a different amount of parcels. Furthermore, parcels consist of thousands of voxels representing a small cube of neurons.

each node varies depending on the quantity of brain tissue covered, that is, the node's spatial dimension. We seek to investigate how variations in parcellation schemes affect trt reliability, and hence examine seven different parcellation resolutions $(8,16,17,34,100,300,500$ nodes) of the entire cortex. Appendix B.2 contains detailed information on the parcellation process.

We can decrease the data dimension before further analysis by extracting one representative time series from each node. As showcased in Figure 2.1, every voxel holds a time series representing the BOLD signal in a particular brain area. Principal component analysis (PCA) is an unsupervised statistical tool for dimensionality reduction of multivariate data sets. The method computes a single representative time series for each network node.

Let $\boldsymbol{y_j} = (y_{1j},\dots,y_{nj})^T$ denote $T = 315$ observations of the normalised BOLD signal from the jth voxel, for $j = 1,\dots,v$. There are v voxels contained in the node. The column matrix $\mathbf{Y} \in \mathbb{R}^{n\times v}$ consists of the time series of all voxels contained in one node. The columns of $\mathbf{Y}$ are linearly transformed onto a lower dimensional space spanned by orthogonal principal components $\boldsymbol{x_1}, \boldsymbol{x_2}\dots, \boldsymbol{x_v} \in \mathbb{R}^{n\times v}$ while retaining most of the variability [34] as follows

$$\boldsymbol{x_1} = \phi_{11}\boldsymbol{y_1} + \phi_{21}\boldsymbol{y_2} + \dots + \phi_{v1}\boldsymbol{y_v} = \sum_{j=1}^{v} \phi_{j1}\boldsymbol{y_j} = \mathbf{Y}\boldsymbol{\phi}_1, \tag{2.1}$$

where $\boldsymbol{\phi}_1 = (\phi_{11},\phi_{21},\dots,\phi_{v1}) \in \mathbb{R}^{v\times 1}$ is referred to as the *loadings* of the first principal component (PC). Appendix A.1 contains more details on how the first PC is retrieved from each node's data matrix $\mathbf{Y}$.

2.2 Connectivity measures: Infering the edges

In the following section, let $\boldsymbol{x_k} \in \mathbb{R}^T$ denote a representative time series of length $T = 315$, corresponding to the first PC introduced at the end of last section, for the kth node of a parcellation scheme. The cortex is parcellated into p nodes. The number of nodes contained in a brain network is defined by $p \in (8, 16, 17, 34, 100, 300, 500)$. Let $\mathbf{X} \in \mathbb{R}^{T\times p}$ be a data matrix of p column vectors, $\boldsymbol{x_k}$ for $k = 1,\ldots,p$, each with T observations. Each of the three methods presented next results in a weight matrix, $\mathbf{W} \in \mathbb{R}^{p\times p}$. The weights represent the functional connections between all pairs of nodes in a brain network of p nodes.

2.2.1 Pearson product-moment correlation

Functional connectivity is, according to Chapter 1, defined as the statistical dependency between nodes. In the following section, we aim to compute the strength of the pairwise association between the activity of two neural entities. A usual strategy utilized to assess FC, is to disregard temporal dependency and assume a linear relationship between BOLD signals.

Correlation is a measure of dependecy between two continous random variables $\boldsymbol{x}_i \in \mathbb{R}^T$, $\boldsymbol{x}_j \in \mathbb{R}^T$, independent of the scale of measurement. The empirical correlation between the two vectors denoted by $r_{\boldsymbol{x}_i,\boldsymbol{x}_j} \in \mathbb{R}$ is defined as follows

$$r_{\boldsymbol{x}_i,\boldsymbol{x}_j} = \frac{\widehat{\mathrm{Cov}}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{\widehat{\mathrm{Var}}(\boldsymbol{x}_i)\widehat{\mathrm{Var}}(\boldsymbol{x}_j)}} = \frac{\sum\limits_{t=1}^{T}(x_{ti}-\bar{x}_i)(x_{tj}-\bar{x}_j)}{\sqrt{\sum\limits_{t=1}^{T}(x_{ti}-\bar{x}_i)^2\sum\limits_{t=1}^{T}(x_{tj}-\bar{x}_j)^2}}. \tag{2.2}$$

The observations are centered in the numerator by subtracting out the mean of each variable, defined by $\bar{x} = \frac{1}{T}\sum\limits_{t=1}^{T} x_t$. The numerator corresponds to the empirical covariance of the two variables. The denominator scales the variables to have equal units through a division of the empirical variance of each variable. The above formula, (2.2), corresponds to the definition of Pearson correlation coefficient of the two variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The coefficient‘s range is $r \in [-1, 1]$. The numerical part of r denotes the strength of the linear relationship between two variables. The sign signifies whether the nodes are positively or negatively correlated. Two variables are perfectly correlated when $r = 1$, meaning that one variable can be explained by a linear function of the other [35].

Rodgers et al. [36] present 13 ways to interpret the Pearson correlation coefficient, illustrating the diversity of this broadly applied index in statistics. The correlation coefficient between a pair of variables is directly analogous to the slope of a simple linear regression model of $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$, as defined in

Appendix A.2,

$$x_{ti} = \beta_0 + \beta_1 x_{tj} + \epsilon_t, t = 1, \dots, T, \tag{2.3}$$

The slope β_1 denotes the average change of the response x_{ti} for every one unit increase in x_{tj}. Observe that the correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in (2.2) equals the standardized slope of the regression line, whose least squares (LS) estimate of β_1 is given in (A.9),

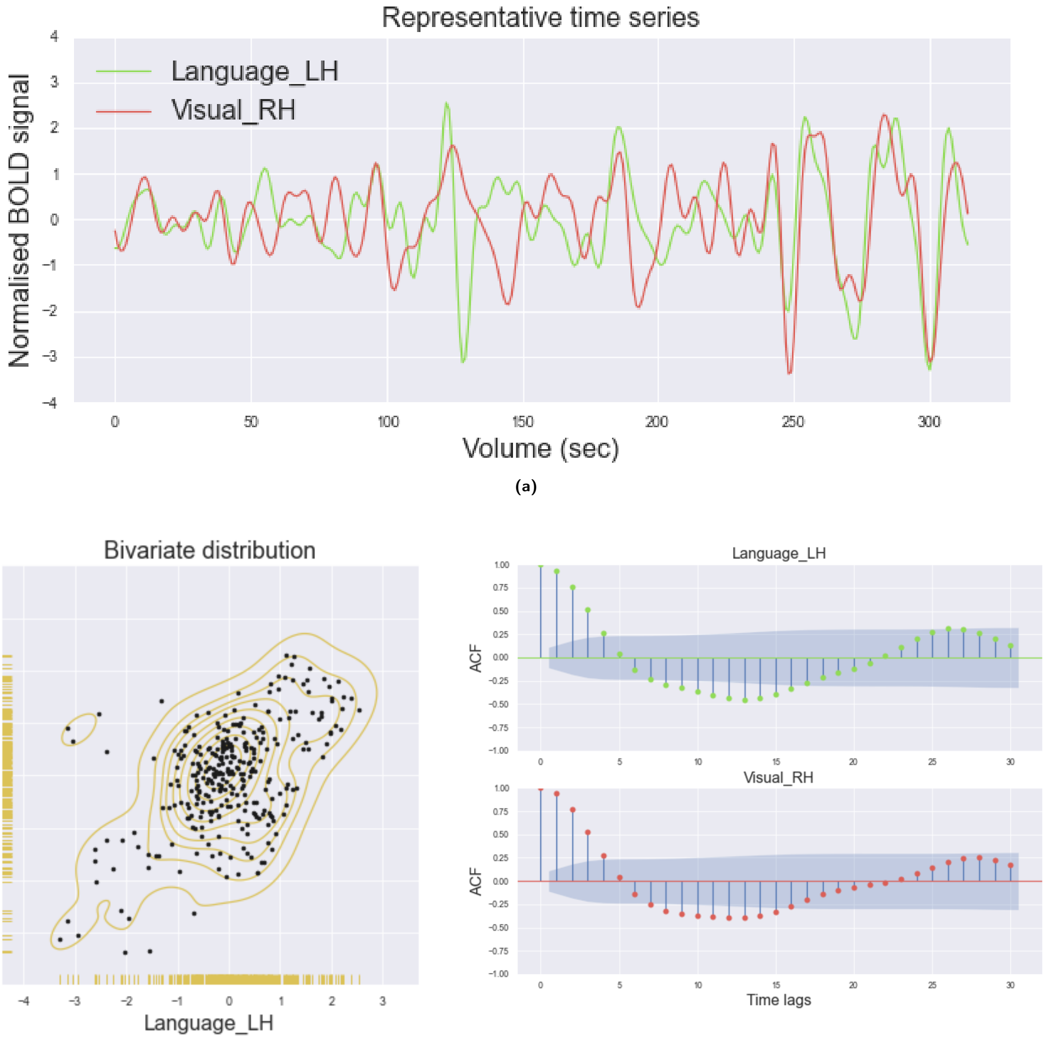
$$r_{\boldsymbol{x}_i,\boldsymbol{x}_j} = \hat{\beta}_1 \frac{\widehat{\text{Var}}(\boldsymbol{x}_j)}{\widehat{\text{Var}}(\boldsymbol{x}_i)}. \tag{2.4}$$

The BOLD vectors, considered in this work are normalized with zero mean and unit variance such that the correlation coefficient $r_{\boldsymbol{x}_i,\boldsymbol{x}_j}$ equals $\hat{\beta}_1$ from (2.4).

Although it is well-understood that Pearson correlation only assesses the linear relationship between two variables, explicit assumptions of the method are intensely debated [37] [38]. However, one should consider whether the data sample meets certain assumptions required for the Pearson correlation coefficient to be a sufficient statistic [39]. These are summarised as follows:

1. Data is drawn from a random or representative sample.
2. Both variables, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are continous and jointly normally distributed.
3. Each observation pair $(x_{ti}, x_{tj}), t = 1, \dots, T$ is measured independently of every other observations.
4. There are no outliers, that is observation pairs that differ significantly from the rest.

To avoid misleading correlations caused by data not behaving in accordance with the underlying assumptions outlined above, Anscombe recommends analysing both computations and graphs [40]. It is too time-consuming to plot and examine all pairs of time series in all parcellation schemes of p nodes. Therefore, we provide graphs of different pairs of BOLD time series in Appendix B.3, of which one pair is presented in Figure 2.3. The nodes, each holding a time series, are chosen arbitrarily to assess a representative subset of node pairs. Figure 2.3a showcases the normalised BOLD signal from each of the two nodes. First, we evaluate the data behaviour in Assumption 2, qualitatively through an inspection of the bivariate distribution of the two variables in Figure 2.3b. The bivariate normal distribution belongs to the large family of elliptical distributions [35]. It is bell-shaped with either circular or elliptical contours depending on the correlation magnitude between the two variables. The variables of a normal joint distribution are marginally normally distributed. The small bars along the axes of Figure 2.3b prove that the highest frequency of observations is around zero.



(a)

(b)

(c)

Figure 2.3: Three different visualisations of the relationship between a representative pair of BOLD time series are related to the underlying assumptions of Pearson correlation. The BOLD signals correspond to the first PCs within two nodes of a parcellation scheme of 16 nodes. The nodes are labeled according to the networks defined in Table B.2 and their associated hemisphere, left hemisphere (LH) or RH (a): The normalised BOLD time series with $T = 315$ observations, and a sample rate of 1.53s. (b) The bivariate distribution, achieved from kernel density estimation in Seaborn (Appendix C.1), is bell-shaped. The number of contours is analogous to the sampling density. The observations of the two samples are visualised as ticks along the axes. (c) The Autocorrelation function for the 40 first lags in each variable. The lags within the shaded region, the 95% confidence interval, are assumed to have no significant correlation with the previous value. There is evidence for significant autocorrelation between the first time points in both variables.

A time series is defined as a sequence of random variables with indices $t = 1, 2 \ldots, T$ ordered according to the time they are obtained [41]. Define the Autocorrelation function (ACF) for a signal x_t as a function of delay s as follows:

$$ACF(s) = \frac{\text{Cov}(x_t, x_{t-s})}{\sqrt{\text{Var}(x_s, x_s)}}. \tag{2.5}$$

Figure 2.3c visualises the ACF defined in (2.5) for each of the two representative time series in Figure 2.3a. Significant autocorrelations between the first few time lags are present in both vectors. Autocorrelated errors are a usual occurrence in time series data. Periodic noise present in the BOLD signal, such as heart rate and respiration, referred to as seasonal components, might give rise to autocorrelated errors [22]. Assumption 3 above states that observations from each variable must be independent of past and coming observations of the same pair of variables. Autocorrelated errors violate this assumption and can lead to "nonsense correlations", which according to Harris et al. [42], result in significant correlations for unrelated signals.

2.2.2 Partial correlation with Graphical lasso

Pearson correlation which was introduced in the last section, reflects the pairwise relationships between BOLD signals. It was mentioned in Section 1.2.2 that a third, unobserved node might influence the connection between the node-pair being considered. The FC method introduced next adjusts for such confounding factors, yielding both the strength and directness of a brain network's edges.

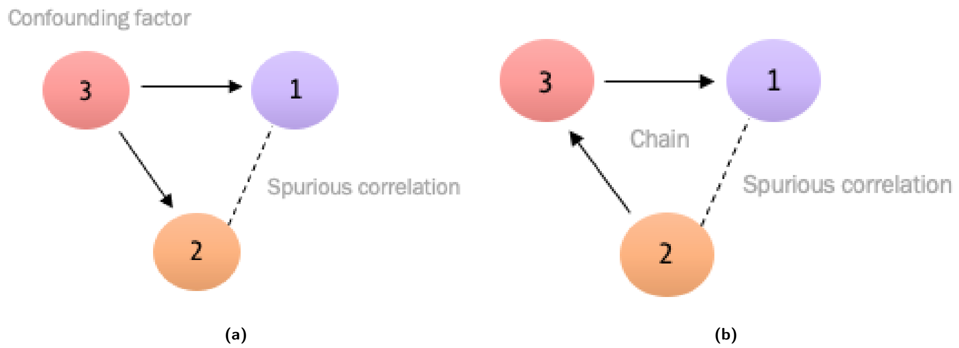


Figure 2.4: An illustration of two possible scenarios leading to spurious correlations when considering the pairwise correlation between node 1 and 2. (a) Node 3 confound the pairwise relationship between 1 and 2. (b) A loop between all three nodes is observed, although only two connections are true.

In Section 2.2.1, we studied the node-pairs independently. We isolated two nodes from the rest and computed their statistical dependency. Consider the three nodes in Figure 2.4a. Node 3 is correlated with both node 2 and node 1. The connection between node 1 and 2, called *spurious*, occurs because the two nodes appear casually related. The spurious connection is caused by a confounding factor, which in this case is node 3. Partial correlation, introduced next, improves causal inferences in the case of confounders and chains, such as visualised in Figures 2.4a and 2.4b [27].

The partial correlation coefficient between a pair of random variables $\boldsymbol{x}_i \in \mathbb{R}^T$ and $\boldsymbol{x}_j \in \mathbb{R}^T$, originating from $\mathbf{X} \sim \mathbb{R}^{T\times p}$, is directly analogous to the multiple linear regression model [43] defined Appendix A.2. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be samples from $\mathcal{N}(\boldsymbol{\mu},\Sigma)$. To compute the partial correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, consider two linear regression problems, one for each of the variables as the response function

$$\begin{aligned} \boldsymbol{x}_i &= \mathbf{X}_{\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}}\boldsymbol{\beta} + \boldsymbol{\epsilon}_i \\ \boldsymbol{x}_j &= \mathbf{X}_{\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}}\boldsymbol{\beta} + \boldsymbol{\epsilon}_j \end{aligned} \tag{2.6}$$

where $\mathbf{X}_{\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}} \in \mathbb{R}^{T\times(p-2+1)}$ resembles the design matrix of all but the two variables to be correlated. The residual vectors, $\boldsymbol{\epsilon}_i \in \mathbb{R}^T$ and $\boldsymbol{\epsilon}_j \in \mathbb{R}^T$, have indices corresponding to their related response variable, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Solving the problems in (2.6) amounts to estimating the regression coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{p-2})$. The estimation can be performed using least squares (LS) defined in (A.8), or Maximum Likelihood Estimation (MLE) as in (A.16). The partial correlation coefficient equals the sample correlation of the two estimated residual vectors, defined as

$$\begin{aligned} \hat{\boldsymbol{\epsilon}}_i &= \boldsymbol{x}_i - \mathbf{X}_{\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}}\hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\epsilon}}_j &= \boldsymbol{x}_j - \mathbf{X}_{\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}}\hat{\boldsymbol{\beta}} \end{aligned} \tag{2.7}$$

such that

$$\rho_{\boldsymbol{x}_i,\boldsymbol{x}_j|\mathbf{X}\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}} = \widehat{\text{Corr}}(\hat{\boldsymbol{\epsilon}}_i, \hat{\boldsymbol{\epsilon}}_j). \tag{2.8}$$

The sample correlation, estimating $\text{Corr}(\hat{\boldsymbol{\epsilon}}_i, \hat{\boldsymbol{\epsilon}}_j)$ above is defined in (2.2). The partial correlation coefficient quantifies the linear relationship between two BOLD signals $\boldsymbol{x}_i, \boldsymbol{x}_j$ conditioned on the $p-2$ other signals of $\mathbf{X}$.

Scaling of the inverse covariance matrix of $\mathbf{X}$, denoted $\Sigma^{-\mathbf{1}}$, is a less computationally intensive approach for estimating partial correlation coefficients. The inverse covariance matrix is also represented by Θ, called the precision matrix of $\mathbf{X}$. The partial correlation coefficient of $\Theta = \{\theta_{ij}\}_{p\times p}$, denoted by $\rho_{\boldsymbol{x}_i,\boldsymbol{x}_j|\mathbf{X}\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}}$, is defined as follows:

$$\rho_{\boldsymbol{x}_i,\boldsymbol{x}_j|\mathbf{X}\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}}. \tag{2.9}$$

Details behind this result can be found in the book *Graphical Models* by Lauritzen [44], page 130. The precision matrix contains the relationship between pairs of variables conditioned on all other variables. If an entry of the precision matrix is zero, that is $\theta_{ij} = 0$, the two nodes, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, are conditionally independent. While for a zero element in the correlation matrix, the two nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are only marginally independent. According to Hastie et al., the latter kind of independence between nodes does not represent the real world [45]. As a result, when examining network connections, the precision matrix is generally favoured over the correlation matrix.

Next, we model the brain network by a Gaussian graphical model, that is, a conditional independence graph with multivariate Gaussian distributions:

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma). \tag{2.10}$$

The edges of a Gaussian graphical model correspond to the entries of a precision matrix $\Theta \in \mathbb{R}^{p \times p}$ [43]. We need to estimate Θ to yield a network of connections between $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$. Consider the log-likelihood function (defined in in Appendix A.3) of the multivariate Gaussian variables contained in $\mathbf{X}$, as a function of the precision matrix

$$\log L(\Theta \mid \mathbf{X}) = \log\det\Theta - \operatorname{tr}(\hat{\Sigma}_{\mathrm{ML}}\Theta). \tag{2.11}$$

A derivation of (2.11) can be seen in (A.24). The log-likelihood in (2.11) is maximized with respect to Θ. As a result, in (2.11), the terms of (A.24) that are constants with respect to Θ are removed.

Recall, $T = 315$ and $p \in (8, 16, 17, 34, 100, 300, 500)$ in this work. In theory, the estimated precision matrix denoted $\hat{\Theta}_{ML}$ converges towards Θ when the number of samples T tends to infinity. However, for problems where the number of nodes p is large compared to T, a MLE may not exist. That is because the sample covariance matrix $\hat{\Sigma}_{\mathrm{ML}}$ is not invertible in these cases. A solution is to consider a regularized form of the MLE to constrain the precision matrix [46] while being estimated. In Appendix A.4 we define one such regularization technique, called least absolute shrinkage and selection operator (lasso). We add a penalty to the log-likelihood in (2.11). The lasso penalty is based on the ℓ_1 of the off-diagonal elements of Θ:

$$\begin{aligned} \hat{\Theta} &= \max_{\Theta} \log\det\Theta - \operatorname{tr}(\hat{\Sigma}_{\mathrm{ML}}\Theta) - \lambda \|\Theta\|_1 \\ \|\Theta\|_1 &= \sum_{s \neq t} |\,\theta_{st}\,|\,. \end{aligned} \tag{2.12}$$

λ is a non-negative *tuning parameter*. The lasso penalty has the effect that, when the tuning parameter is large, some of the entries of $\hat{\Theta}$ will be exactly zero. As a result, the optimization problem above can be solved for $p > T$. An efficient algorithm that estimates a sparse inverse covariance matrix, $\hat{\Theta}$ by solving (2.12) using coordinate-descent is called Graphical lasso (GLasso), and was proposed by Friedman et al. [47]. To demonstrate coordinate-descent, a problem analogous to (2.12) is solved in Appendix A.4. The main focus for the remainder of this section is the GLasso algorithm defined in Algorithm 1. The theory presented is based on the book *The Elements of Statistical Learning* by Hastie et al. [45].

We start by setting the gradient of Θ in (2.12) to $\mathbf{0}$, using the fact that the derivative of $\log\theta$ equals θ^{-1}, as follows

$$\Theta^{-1} - \hat{\Sigma}_{MLE} - \lambda\Gamma = \mathbf{0}. \tag{2.13}$$

Γ is a matrix of Lagrange parameters with nonzero values for all node pairs without any edges. In other words, when $\theta_{ij} = 0$ then $\gamma_{ij} \neq 0$. Let $\mathbf{W} = \Theta^{-1}$ denote the current working estimate of the inverse of the precision matrix. Furthermore, let $\mathbf{S}$ represent the sample covariance matrix $\hat{\Sigma}_{MLE}$. The GLasso algorithm estimates the entries $\mathbf{W}$ and Θ, updating one row and column at a time. The system (2.13) is solved via blockwise coordinate descent, partitioning all matrices, $\mathbf{W}$,Θ and $\mathbf{S}$, into two parts: the first being the block of $j-1$ rows and columns, and the second the jth row and column. By definition, we know that: $\mathbf{W}\Theta = \mathbf{I}$, which for the blockwise partitioning is expressed as follows:

$$\begin{bmatrix} \mathbf{W}_{p-1} & \boldsymbol{w}_p \\ \boldsymbol{w}_p^T & w_{pp} \end{bmatrix} \begin{bmatrix} \Theta_{\mathbf{p-1}} & \boldsymbol{\theta}_p \\ \boldsymbol{\theta}_p^T & \theta_{pp} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{2.14}$$

$\mathbf{W}_{p-1}$ is the upper left $(p-1)\times(p-1)$ block, $\boldsymbol{w}_p$ are non-diagonal elements of the pth row and column and w_{pp} the corresponding diagonal element. The same accounts for Θ and $\mathbf{S}$. Let p denote the last row and column such that the upper right of (2.13) becomes

$$\boldsymbol{w}_p - \boldsymbol{s}_p - \lambda\boldsymbol{\gamma}_p = \mathbf{0}. \tag{2.15}$$

The relation in (2.14) implies that

$$\begin{aligned} \boldsymbol{w}_p &= -\frac{1}{\theta_{pp}}\mathbf{W}_{p-1}\boldsymbol{\theta}_p \\ &= \mathbf{W}_{p-1}\boldsymbol{\beta}, \end{aligned} \tag{2.16}$$

where $\boldsymbol{\beta} = -\frac{1}{\theta_{pp}}\boldsymbol{\theta}_p$. Substitute (2.16) into (2.15) such that

$$\mathbf{W}_{p-1}\boldsymbol{\beta} - \boldsymbol{s}_p + \lambda\boldsymbol{\gamma}_p = \mathbf{0}. \tag{2.17}$$

The above equation (2.17) is analogous to the objective function in (A.29). Consider a usual regression setup with response variable $\boldsymbol{y}$ and predictor matrix $\mathbf{X}$, such as in (A.26). A *subgradient* equation of the expression in (A.26) becomes

$$\frac{1}{n}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - \frac{1}{n}\mathbf{X}^T\boldsymbol{y} + \lambda \text{sign}(\boldsymbol{\beta}) = 0, \tag{2.18}$$

which compared to (2.17) show that $\frac{1}{n}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}$ is analogous to $\mathbf{W}_{p-1}$ and $\frac{1}{n}\mathbf{X}^T\boldsymbol{y}$ to $\boldsymbol{s}_p$. We can solve the blocks of equations in (2.17), treating each variable as the response variable and the other $p-1$ variables as the predictors by the GLasso algorithm defined in Algorithm 1 [46].

Furthermore, to infer $\hat{\Theta}$, notice that the diagonal elements of Θ can be found from

$$\frac{1}{\theta_{pp}} = w_{pp} - \boldsymbol{w}_p^T\boldsymbol{\beta}. \tag{2.19}$$

using partitioned inverse formulas of (2.14). Notice that the diagonal elements w_{pp} stay unchanged and equal to s_{pp} as initialized.

Algorithm 1: Graphical Lasso

```
Let $\mathbf{W}$ be current working estimate.
Initialize $\mathbf{W}$ with the empirical covariance matrix $\boldsymbol{S}$.
while not converged do
    for $j = 1,2,\ldots,p,1,2,\ldots,p,\ldots$ do
        Partition $\mathbf{W}$ into a block matrix as shown in (2.14).
        Solve $\mathbf{W}_{p-1}\boldsymbol{\beta} - \boldsymbol{s}_p + \lambda\text{sign}(\boldsymbol{\beta})$ using a coordinate descent algorithm.
        Update $\boldsymbol{w}_j = \mathbf{W}_{j-1}\hat{\boldsymbol{\beta}}$ from (2.16).
for $j = 1,2\ldots,p$ do
    Recover $\Theta$ by solving (2.19) for $\theta_{jj}$ and then (2.16) for $\boldsymbol{\theta}_j$, using the final estimate of $\boldsymbol{W}$.
```

The convergence criteria in Algorithm 1 can be set to either a maximum number of iterations or a specified tolerance t such that the average absolute change in $\mathbf{W}$ is less than $\mathrm{t}\cdot$ average $|\ \mathbf{S}^{-\text{diag}}\ |$. $\mathbf{S}^{-\text{diag}}$ are the off-diagonal elements of the empirical covariance matrix $\mathbf{S}$. The tolerance is set to $t = 10^{-3}\forall p$ in our implementation. We perform a 5-fold-cross validation (CV) with loss function equal the negative log-likelihood in (2.11), to find the optimal hyperparameter λ. The cross validation (CV) procedure is defined in Appendix A.4.

2.2.3 Dynamical causal model

None of the two FC methods described so far has been able to account for all of the characteristics of edges in a brain network: strength, directness, and directionality, as defined in Section 1.2.2. Measures of effective connectivity attempt to examine a brain network with edges that contain all three features by establishing the genuine relationship between the recorded BOLD signal and neural activity.

The Dynamical causal model (DCM), primarily proposed by Friston et al. [48], is one such method. The method is mainly exploited on a reduced brain network to test hypotheses about whether experimental input will activate specific brain connectivity patterns. Connectivity weights are represented by parameters tuned to generate a predicted BOLD signal. The computational cost of accounting for the human brain's complex biophysical processes is substantial.

More efficient DCM variants exist, such as the spectral DCM (spDCM) implemented using rs fMRI data. This model may be utilized to discover the structure of larger networks [49] compared to the original procedure. However, due to the exponential increase of free parameters as the network grows, we have only been able to fit the spDCM to a network with a maximum of 17 nodes. The following section provides a brief introduction to the original DCM. We consider experimental stimuli, such as hand movement, when defining this model. Finally, the spectral DCM is defined, primarily based on the same theoretical ideas as the original model.

State equations

Consider a small brain network of three connected nodes, as illustrated in Figure 2.5. The DCM estimates the effective connectivity, defined in Section 1.2.2 as a subset of FC, represented as *coupling* parameters between nodes. Central to estimating these parameters is the *neuronal state* variable defined for each node. The variables correspond to the assumed synaptic activity of each node and are denoted as $\boldsymbol{Z} = (\boldsymbol{z_1}, \boldsymbol{z_2}, \boldsymbol{z_3})^T$, where each state variable is $\boldsymbol{z_i} \in \mathbb{R}^T, i = 1,2,3$. Recall that T denotes the number of observations, or time points of the BOLD signal. The time derivative of each state variable is a function of the states of other nodes, some experimental input vectors, $\mathbf{U} = (\boldsymbol{u_1}, \boldsymbol{u_2})^T \in \mathbb{Z}^{2\times T}$, believed to drive the activity in the nodes, and some matrices holding the coupling parameters, $\mathbf{A} \in \mathbb{R}^{3\times 3}$ and $\mathbf{C} \in \mathbb{R}^{3\times 3}$.

$$\frac{d\mathbf{Z}}{dt} \approx \mathbf{A}\mathbf{Z}(t) + \mathbf{C}\mathbf{U}. \tag{2.20}$$

$\mathbf{A}$, whose entries represent the *intrinsic* connectivity between all nodes represent the weight matrix $\mathbf{W} \in \mathbb{R}^{p\times p}$ for p nodes. $\mathbf{C}$ denotes the effect of influence from the experimental input vectors $\boldsymbol{u_1}, \boldsymbol{u_2}$, and is not of main interest in further analysis.

We can now understand most of the variables in Figure 2.5. The nodes each holds a T-dimensional state vector $\boldsymbol{z}_i$. Recall $T = 315$. The arrows denote the connection weights, either from experimental

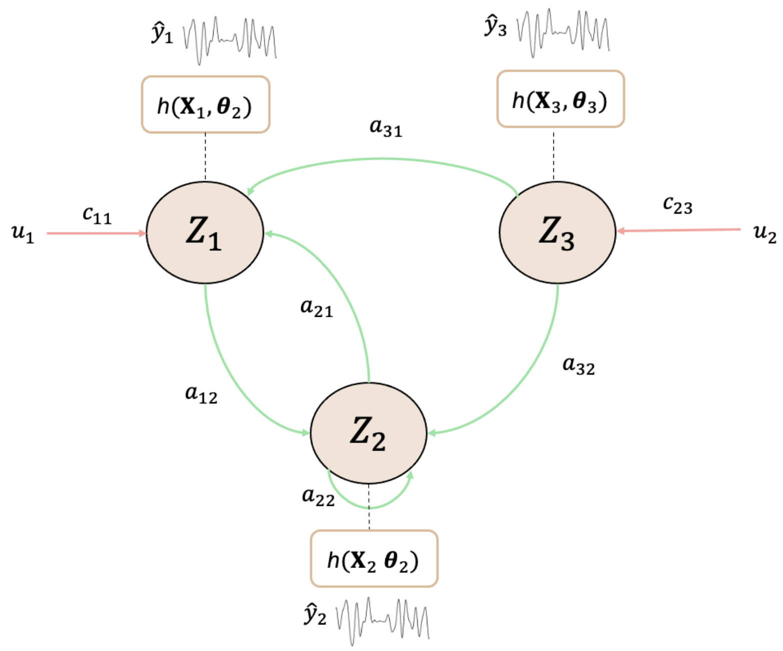

$\hat{y}_1$
$h(\mathbf{X}_1, \boldsymbol{\theta}_2)$
a_{31}
$\hat{y}_3$
$h(\mathbf{X}_3, \boldsymbol{\theta}_3)$
u_1
c_{11}
Z_1
Z_3
c_{23}
u_2
a_{21}
a_{32}
a_{12}
Z_2
a_{22}
$h(\mathbf{X}_2\ \boldsymbol{\theta}_2)$
$\hat{y}_2$

Figure 2.5: Three nodes each holding a neural state z, connected with each other trough coupling matrices A and C is displayed. The driving input affects both states directly. The parameters are inferred when estimating the time series $\hat{y}$ from each node through a function g depending on the hemodynamic state vector $\boldsymbol{x}$ and the neural state z.

input or from other nodes. Now, expand equation (2.20) for the 3-nodes case in Figure 2.5:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} 0 & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} c_{11} & 0 \\ 0 & 0 \\ 0 & c_{32} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix} \tag{2.21}$$

To understand the rest of the details in Figure 2.5, we introduce the remaining state variables belonging to each node. That is, the biophysical states that are believed to engender the BOLD signal by translating the inferred synaptic activity, $\boldsymbol{z}_i$ from (2.20), into some *hemodynamic state* variables $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4 \in \mathbb{R}^T$. These variables enter a nonlinear function h to estimate the observed BOLD signal $\boldsymbol{y}_i \in \mathbb{R}^T$, with a measurement error, denoted $\boldsymbol{\epsilon}_i$ for node i as follows

$$\boldsymbol{y}_i = h(\mathbf{X}_i, \theta_i) + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{C}_{\boldsymbol{\epsilon}}). \tag{2.22}$$

The error term $\boldsymbol{\epsilon}_i$ is assumed zero mean additive Gaussian noise with covariance $\mathbf{C}_{\boldsymbol{\epsilon}}$. Let $\mathbf{X}_i$ be the state vector for node i, holding the four hemodynamic states $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$, in addition to the neuronal state variable $\boldsymbol{z}_i$. The parameter vector $\theta = (\mathbf{A}, \mathbf{C}, \ldots)$ holds the coupling matrices and biophysical parameters related to some hemodynamic differential equations. These differential equations, transforming synaptic activity from $\boldsymbol{z}_i$ into the hemodynamic states $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4$, were proposed by Buxton et a. [50] aiming to explain the changes in biological states linked to blood flow during brain activation. The state equations are not of main focus in this work, and the reader is referred to the original work by Friston et al. [48] for further details.

Parameter estimation

In what follows we have simplified notation by dropping the node index i. The predicted BOLD time series, $\hat{\boldsymbol{y}} = h(\mathbf{X}, \theta)$ from (2.22), are generated by integrating the differential equations of the neural state (2.20) and the hemodynamic states. There is need for some prior parameters to initialize $\boldsymbol{\theta}$ and yield a complete estimate $\hat{\boldsymbol{y}}$. The posterior distributions of the parameters in $\boldsymbol{\theta}$ are then estimated iteratively by maximizing *the negative free energy* $\tilde{F}$ in (A.38) and thus minimizing the difference between the observed and predicted BOLD signal.

Assume the prior distribution of θ to be Gaussian: $\pi(\theta) \sim \mathcal{N}(\eta_\theta, \mathbf{C}_\theta)$. Bayesian inference, defined in Appendix A.5, aims to infer parameter estimates after data is observed. Sine the sample size of the fMRI time series is sufficiently large, the posterior distribution asymptotically approaches a Gaussian distribution around the Maximum a Posterior (MAP). The posterior distribution of θ is therefore assumed Gaussian

$$\theta \mid \boldsymbol{y} \overset{a}{\sim} \mathcal{N}(\eta_{\theta|\boldsymbol{y}}(\hat{\boldsymbol{\theta}}_{\text{MAP}}), \mathbf{I}^{-1}(\hat{\boldsymbol{\theta}}_{\text{MAP}})). \tag{2.23}$$

Here $\mathbf{I}^{-1}(\hat{\boldsymbol{\theta}}_{\text{MAP}})$ is the inverse Hessian or Fisher information matrix from (A.21) evaluated at $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ [51]. We aim to find $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ that maximizes the posterior distribution. This is computed using the Fisher scoring in (A.22), where $\eta_{\theta|\boldsymbol{y}}$ is updated in the direction of the gradient of the log-posterior function $l_p(\boldsymbol{\theta})$, evaluated at the current estimated posterior mean $\hat{\eta}^{(t)}_{\theta|\boldsymbol{y}}$:

$$\hat{\eta}^{(t+1)}_{\theta|y} = \hat{\eta}^{(t)}_{\theta|y} + \frac{\partial^2 l_p(\theta)}{\partial\theta\partial\theta^T}^{-1}\Big|_{\theta=\hat{\eta}^{(t)}_{\theta|y}} \frac{\partial l_p(\theta)}{\partial\theta}\Big|_{\theta=\hat{\eta}^{(t)}_{\theta|y}} \tag{2.24}$$

From Bayes rule (A.33) the log-posterior distribution can be derived as follows

$$\begin{aligned}
l_p(\theta) &= \pi(\theta \mid \boldsymbol{y}) \\
&\propto \log\pi(\theta) + \log p(\boldsymbol{y} \mid \theta) \\
&= -\frac{1}{2}\Big(\log|\,\mathbf{C}_\theta\,| + (\theta-\eta_\theta)^T\mathbf{C}_\theta^{-1}(\theta-\eta_\theta)\Big) + c \\
&- \frac{1}{2}\Big(\log|\,\mathbf{C}_\epsilon\,| + (\boldsymbol{y}-h(\theta))^T\mathbf{C}_\epsilon^{-1}(\boldsymbol{y}-g(\theta))\Big) + c,
\end{aligned}$$

where the marginal distribution is omitted as it disappears under maximization. c is a constant with respect to θ. To solve (2.24) the two first derivatives of $l_p(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is needed

$$\begin{aligned}\frac{\partial l_p(\theta)}{\partial \theta}\bigg|_{\theta=\hat{\eta}^{(t)}_{\theta|y}} &= -\frac{\partial h(\hat{\eta}^{(t)}_{\theta|\boldsymbol{y}})}{\partial \theta}^T \mathbf{C}_\epsilon^{-1}(y - h(\hat{\eta}^{(t)}_{\theta|\boldsymbol{y}})) - \mathbf{C}_\theta^{-1}(\hat{\eta}^{(t)}_{\theta|\boldsymbol{y}} - \eta_\theta) \\ \frac{\partial^2 l_p(\theta)}{\partial \theta \theta^T}\bigg|_{\theta=\hat{\eta}^{(t)}_{\theta|y}} &\approx -\frac{\partial h(\hat{\eta}^{(t)}_{\theta|\boldsymbol{y}})}{\partial \theta}^T \mathbf{C}_\epsilon^{-1}\frac{\partial h(\hat{\eta}^{(t)}_{\theta|\boldsymbol{y}})}{\partial \theta} - \mathbf{C}_\theta^{-1}.\end{aligned} \tag{2.25}$$

Notice that the second-order derivative of the nonlinear function $h(\hat{\eta}^{(t)}_{\theta|\boldsymbol{y}})$ is ignored, as we assume the function in (2.22) to be weakly nonlinear [52]. The iterative scheme requires a specified initial value for the posterior mean. The initial value is set equal to the prior density, $\hat{\eta}^{(0)}_{\theta|\boldsymbol{y}} = \hat{\eta}_\theta$.

The optimization scheme in (2.24) cannot be evaluated due to the unknown error covariance $\mathbf{C}_\epsilon$ present in (2.25). The error covariance can be estimated through a linear mixture of some precision components $\mathbf{Q_i}$ weighted by hyperparameter λ_i for each node i, that is $\mathbf{C}_\epsilon = \sum_i \lambda_i \mathbf{Q}$. The hyperparameters θ has to be estimated in addition to the parameters θ suggesting a generalized Expectation maximization (EM) algorithm similar to Algorithm 2. As stated in Appendix A.5 the EM algorithm aims to maximize the likelihood of the observed data $\boldsymbol{y}$ conditional on the hyperparameters λ with the presence of unobserved data. In this case $\boldsymbol{\theta}$ is treated as the unobserved data, such that (A.36) becomes

$$\log p(\boldsymbol{y} \mid \lambda) = \int p(\boldsymbol{y}, \theta \mid \lambda) d\theta. \tag{2.26}$$

In Appendix A.5 it was demonstrated that there exists a lower bound on (2.26), $\tilde{F}(q,\lambda) \leq \log p(\boldsymbol{y} \mid \lambda)$. Thus, a maximization of the functional $\tilde{F}(q,\lambda)$ implies a maximization of (2.26), see (A.39).

The **E-step** in the generalized EM algorithm searches for an approximate posterior distribution of the parameters denoted as $q(\theta) \sim N(\hat{\mu}, \hat{\boldsymbol{\Sigma}})$. It maximizes $\tilde{F}(q,\lambda)$, holding the hyperparameters from the last M-step fixed,

$$q^{(t)} \leftarrow \underset{q}{\operatorname{argmax}}\Big(\tilde{F}(q, \lambda^{(t-1)})\Big). \tag{2.27}$$

The maximization of an approximate posterior distribution in (2.27) is equivalent to the iterative search for the MAP-estimate $\hat{\eta}_{\theta|\boldsymbol{y}}$ in (2.24). The hyperparameters λ are then estimated in the **M-step** through a maximization of $\tilde{F}(q,\lambda)$, now with respect to λ, holding the parameters, estimated from the previous E-step, fixed

$$\lambda^{(t)} \leftarrow \underset{\boldsymbol{\lambda}}{\operatorname{argmax}}\Big(F(q^{(t)}, \lambda)\Big). \tag{2.28}$$

The M-step is then solved iteratively under a Fisher Scoring Scheme of $\tilde{F}(q,\lambda)$ evaluated at the current estimated hyperparameter $\hat{\lambda}^{(t)}$.

$$\hat{\lambda}^{(t+1)} = \hat{\lambda}^{(t)} + \frac{\partial^2 F(q,\lambda)}{\partial\lambda\partial\lambda^T}^{-1}\Big|_{\lambda=\hat{\lambda}^{(t)}} \frac{\partial F(q,\lambda)}{\partial\lambda}\Big|_{\lambda=\hat{\lambda}^{(t)}} \tag{2.29}$$

The negative variational free energy, $F(q,\lambda)$ defined in (A.38), can be approximated using Variational Laplace, provided by Penny [53].

The EM scheme described above alternates between the E- and M-steps until $q(\theta) \approx \pi(\theta \mid \boldsymbol{y})$. As the model is nonlinear, the difference between the two distributions, $q(\theta)$ and $\pi(\theta \mid y)$, never reaches zero [52]. The convergence criterium is defined as

$$\sum(\eta_{\theta|\boldsymbol{y}} - \hat{\mu})^2 \leq \mathrm{c}. \tag{2.30}$$

In this work, the estimation procedure converges according to (2.30) for $\mathrm{c} = 10^{-6}$ [54] or when reaching a maximum number of iterations, set to 128.

Spectral DCM

The DCM described so far explains how experimental input U influence the neural connections through the coupling parameters of $\mathbf{A},\mathbf{C}$. For resting state fMRI data, the neuronal model in (2.20) becomes

$$\frac{d\mathbf{Z}}{dt} = \mathbf{A}\mathbf{Z}(t) + \mathbf{V}(t) \tag{2.31}$$

where $\mathbf{V}(t)$ contains $\boldsymbol{v}_i(t) \in \mathbb{R}^n$ for node i, and models the endogenous neuronal fluctuations assumed to drive neural activity at rest. Now the neuronal state for node i, $\boldsymbol{z}_i$ depends upon the states of the other nodes and the endogenous fluctuations $\boldsymbol{v}_i(t)$ [55]. Furthermore, the state equation in (2.31) is equipped with hemodynamic state equations equal to the ordinary DCM. The full model is then mapped to measured BOLD responses similarly as in (2.22), as follows:

$$\boldsymbol{y}_i(t) = h(\mathbf{X}_i,\theta_i) + \boldsymbol{\epsilon}(t) \tag{2.32}$$

Due to the addition of the new stochastic term $\mathbf{V}(\mathbf{t})$ in (2.31), we must now estimate not just the model parameters in θ, but also the neuronal states in $\mathbf{Z}(t)$, that have become random variables, for all nodes – a computationally demanding task. A solution is to convert the cross-correlation between pairs of time series into the frequency domain, called the cross spectrum. Cross-correlation is similar to the autocorrelation function in (2.5). However, instead of correlating time points within one time series, cross-correlation represents the correlation between two time series as a function of the displacement of one relative to the other. The time series are then replaced by their cross spectra. As a result, a time-invariant covariance of the random fluctuations of the BOLD signals is predicted instead of the neural states $z_i(t)$, as for the ordinary DCM [49]. The cross spectra is inverted and fitted for the observed data using variational Bayesian techniques (definitions in Appendix A.5), similar as in the preceding section. The aforementioned approach is known as Spectral Dynamic causal modelling (spDCM). The reader is urged to read [56] for further information on this transition and model fitting.

When compared to the regular DCM, the spDCM manages to execute parameter inference on a larger collection of nodes.

2.3 Comparison of brain networks

The previous sections introduced three different connectivity measures, each of which yields a weight matrix $\mathbf{W} \in \mathbb{R}^{p \times p}$. The weight matrix holds the strength of all connections in a network of p nodes. The diagonal elements of **W** represent self-connections, which are no interest in our analysis. Therefore, only off-diagonal elements are utilized in the subsequent analysis of FC. For the two correlation-based measures, that is Pearson and partial correlation defined in Section 2.2.1 and 2.2.2, the edges contain no direction. Thus, for each of the two methods, only the lower triangle of **W**, i.e. $w_{ij} = 0, i > j$, is used during the comparison of brain networks. Let $\boldsymbol{w} \in \mathbb{R}^e$ denote the vectorized $p \times p$ dimensional weight matrix **W**. The number of edges e depends on whether the FC method yields symmetric or asymmetric matrices.

In the following section, we establish a framework for examining the Test-Retest reliability of a connectivity measure. This framework entails calculating a reliability score by quantifying the similarity of two weight matrices representing distinct brain networks. Additionally, the framework considers the individual edges of a functional brain network and how these connections change over time. Based on the premise that relearning emerges as new, strong connections post-stroke, it estimates the minimum number of stroke patients required to detect significant FC changes during the first month following stroke. The connections of a healthy network are presumed to be constant throughout a month. It is worth noting that the power calculations are based on the observed distribution of temporal FC changes across controls as the project has not acquired enough stroke data yet.

2.3.1 Test-Retest Reliability Measure

Reliability of a measure refers to its ability to give consistent results under repeated measurements with similar conditions [10]. The term is related to *precision* denoting the degree of similarity between measured values. Identifying possible post-stroke connectivity changes in a brain network requires a reliable measure capable of quantifying the FC pattern with sufficient precision.

Let $\mathbf{W}^{(s)}$ denote the estimated weight matrix from session s, where $s = 1,2,3$. We examine two test-retest intervals in this work: one month and three months, or equally session 1 vs. 2 and session 1 vs. 3, respectively. A reliability score coefficient, quantifying the net degree of similarity between all weights of $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(s)}$ for session 1 and s such that $s \neq 1$, is Pearson correlation coefficient, previously defined in (2.2). Let $r(1,s)$ denote the reliability score of session 1 and $s = 1,3$. The correlation coefficient represents the strength of a linear relationship between two variables. Recall that the coefficient ranges from -1 to 1, where the latter implies a perfect linear relationship between the weights of the two matrices.

2.3.2 Power Analysis

Branscheidt et al. [4] state that stroke-related damage may alter the connectivity weights among only a few brain regions within or across hemispheres. We believe that new functional connections appear following a stroke, implying that there is a rewiring. Most edges of the brain network should remain stable during the first months following a stroke, while a few particular connections will change in magnitude from zero (that is, no connection) to powerful. Figure 2.6 illustrates expected connectivity changes post-stroke compared to a healthy network. It was stated in Chapter 1 that the primary objective of this thesis is to estimate how many stroke patients are required for the expected temporal change in FC after a stroke to be significant. We conduct power tests to estimate this sample size in the following section. Relevant theory regarding hypothesis testing is also presented in Appendix A.6.

Let $\boldsymbol{w}^{(\text{ses } s, \text{sub } g)} \in \mathbb{R}^e$ be the estimated weight vector for a network of e weighted edges derived from session s of a control subject g. The change in the ith connection over the course of a month is defined as

$$d_i^{(\text{sub } g)} = w_i^{(\text{ses } 1, \text{sub } g)} - w_i^{(\text{ses } 2, \text{sub } g)} \tag{2.33}$$

where $w_i^{(\text{ses } 1, \text{sub } g)} \in \boldsymbol{w}^{(\text{ses } 1, \text{sub } g)}$ and $w_i^{(\text{ses } 2, \text{sub } g)} \in \boldsymbol{w}^{(\text{ses } 2, \text{sub } g)}$. In this study we consider only the difference between the weights of sessions 1 and 2. Assume that the edgewise differences among the subjects, i.e. $d_1^{(\text{sub } g)}, \ldots, d_e^{(\text{sub } g)} \forall g$, are independent and identically normally distributed with zero mean and common variance σ^2. To put it another way, the difference between the ith edges of subject h, denoted as $d_i^{(\text{sub } h)}$ is both independent of (1) the difference between matching edges of another subject, $d_i^{(\text{sub } k)}$, and (2) the difference of mismatching edges of the same subject, $d_j^{(\text{sub } h)}$.

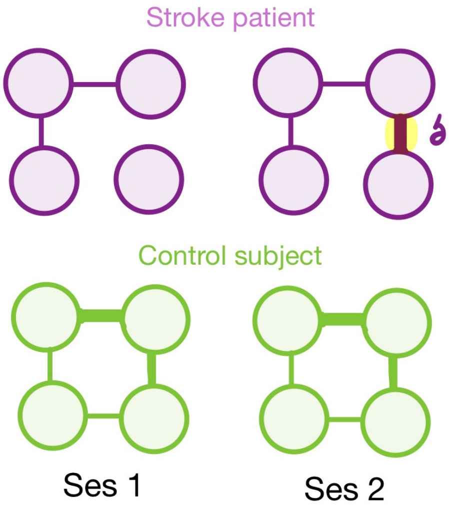


Figure 2.6: An illustration of the post-stroke assumed temporal connectivity change in comparison to a healthy network. The time interval between session 1 and 2 is one month. The averaged FC within each group is represented by the two brain networks. During a month, we anticipate the connection pattern in the averaged control group to remain consistent. After a stroke, we expect new connections to form; the weights of certain edges in session 1 shift from zero (that is, no connection) to powerful (a strong connection), denoted by δ in session 2.

The mean weight differences across n subjects is defined as follows

$$\bar{d}_i = \frac{1}{n}\sum_{g=1}^{n} d_i^{(\text{sub } g)}. \tag{2.34}$$

Because $d_i^{(\text{sub }1)},\ldots,d_i^{(\text{sub } n)}$ are approximately normally distributed with zero mean and variance σ, the mean $\bar{d}_i$ is likewise normally distributed with zero mean and variance σ^2/n. As a result, when evaluating the mean across subjects, the variance decreases as the number of subjects, n rises. Let $\bar{d}_i^{(s)}, \bar{d}_i^{(c)}$ indicate the mean weight difference within the stroke population and the control group, respectively. We wish to identify a substantial difference in the weights of a brain network after stroke compared to a healthy subject‘s network. To test our research topic, create some edgewise hypotheses.

$$\begin{aligned} \mathrm{H}_0 &: \bar{d}_i^{(\mathrm{c})} = \bar{d}_i^{(\mathrm{s})} \\ \mathrm{H}_1 &: \bar{d}_i^{(\mathrm{c})} < \bar{d}_i^{(\mathrm{s})}, \end{aligned} \tag{2.35}$$

or equivalently

$$\begin{aligned} \mathrm{H}_0 &: \bar{d}_i^{(\mathrm{s})} - \bar{d}_i^{(\mathrm{c})} = 0 \\ \mathrm{H}_1 &: \bar{d}_i^{(\mathrm{s})} - \bar{d}_i^{(\mathrm{c})} > 0. \end{aligned} \tag{2.36}$$

The hypothesis in (2.35) tests whether the two population means, $\bar{d}_i^{(c)}$ and $\bar{d}_i^{(s)}$, belong to separate distributions. The degree of overlap between these two distributions is called *effect size* [57]. The further apart the two distributions are, the greater the effect size. Previous studies have concluded that the best way to increase power in neuroimaging research is by using sample sizes justified by power calculations based on estimated effect sizes from prior data [13] [58]. Because we do not have any stroke data available when this thesis is finished, we estimate $\bar{d}_i^{(s)}$ based on the distribution of $d_i^{(c)}$. As mentioned, we assume that a stroke patient will have the relearning appear like a new connection with magnitude δ, see Figure 2.6. Concatenate all of the controls' weight vectors from session 1 into a single weight vector consisting of $w_1^{(\text{ses1,sub}g)}, \ldots, w_e^{(\text{ses1,sub}g)}, \forall g$. Then sort the components of the resultant vector. The predicted temporal weight change of a stroke patient is given by the qth quantile of the sorted list, denoted as $\delta(q)$.

We also need to assume that the FC network of controls stays constant during a month. Consequently, since the expected weight change of controls is zero, $\delta(q)$ represents the estimated effect size between the populations. Furthermore, the test statistic T under the nullhypothesis (2.35) can be defined similarly as the test statistic in (A.42),

$$T = \frac{\bar{d}_i^{(s)} - 0}{\sqrt{s^2/n_s}} \sim t_{n_s-1}. \tag{2.37}$$

s^2, which is an estimator of σ^2,equals the empirical variance of a sample consisting of all distance elements of all controls, $d_1^{(\text{sub } g)}, \ldots d_e^{(\text{sub } g)}, \forall g$. The statistic T is t-distributed with $n_s - 1$ degress of freedom, where n_s refers to the sample size of stroke patients.

Rewiring after a stroke affects just a few edges of the brain network. We must execute the test in (2.35) for the individual, ith edge because we do not know which edges this accounts for. As a result of multiple testing, the likelihood of a Type I error, which according to the definition in (A.41) equals α, rises. Therefore, to obtain a correct significance level α, adjust α_i for each observation as follows

$$\alpha_i = \frac{\alpha}{e}, i = 1, \ldots, e. \tag{2.38}$$

For e edges, there are e tests. The correction in (2.38) is called a *Bonferroni correction* [59].

We can now estimate the sample size n_s of the stroke population necessary to achieve a power of $1-\beta$ with significance level α_i and effect size $\bar{d}_i^{(s)} = \delta(q) > 0$ from

$$n_s = \frac{s^2}{\delta^2(q)} (t_{\beta, n_s-1} + t_{\alpha_i, n_s-1})^2, \tag{2.39}$$

The above equation is based on the derivation in (A.43). Notice that the quantiles of the t-distribution in (2.39) depends on n_s. To estimate the sample size, set an initial guess for n_s. To do this, assume known variance and replace the t-distributed quantiles, t_{β,n_s-1} and t_{α_i,n_s-1}, in (2.39) with z_β and z_{α_i}, from a standard normally distributed test statistic T. The sample size n can be estimated by iteratively evaluating (2.39) until convergence. That is, $n_s^{(i+1)} - n_s^{(i)} < t$ where t is a set threshold.

Chapter 3

Results

We now present the results of applying the procedures specified in the preceding chapter, using a variety of whole-brain parcellations. First, an exploratory analysis of the time series from each parcellation scheme is provided in Section 3.1. The estimated weights for each of the last chapter's three FC methods are shown in Section 3.2. Then each FC measure and parcellation scheme combination's trt reliability scores are presented in Section 3.3. Finally, the predicted sample sizes of stroke patients are determined through power analyses in Section 3.4.

3.1 Exploratory Analysis

The main objective of this thesis is to study prerequisites for detecting longitudinal changes in FC after stroke using rs fMRI data. We examine three sessions of rs fMRI scans performed over three months on healthy elderly (control subjects), see Table 2.1. We conduct power analyses to estimate the minimum required sample size of stroke patients necessary to find longitudinal post-stroke FC changes if it is present in our data. To ensure that these changes are related to motor recovery after stroke, we evaluate the Test-Retest reliability of various parcellations and FC methods. These analyses are incorporated into a framework. The Kong-Schaefer atlas [32] was used to parcellate the brain network into seven different resolutions: $8, 16, 17, 34, 100, 300, 500$ nodes. In Appendix B.2, the parcellation technique is described in depth.

Consider the 16 node brain network, composed of 8 nodes for each hemisphere. The nodes correspond to networks, each of which is given a label depending on the brain area's anticipated functionality. Figure 3.1 depicts this brain network and the label and position of the 8 networks for each hemisphere. The spDCM cannot be implemented on a network of more than 17 nodes as it is compu-

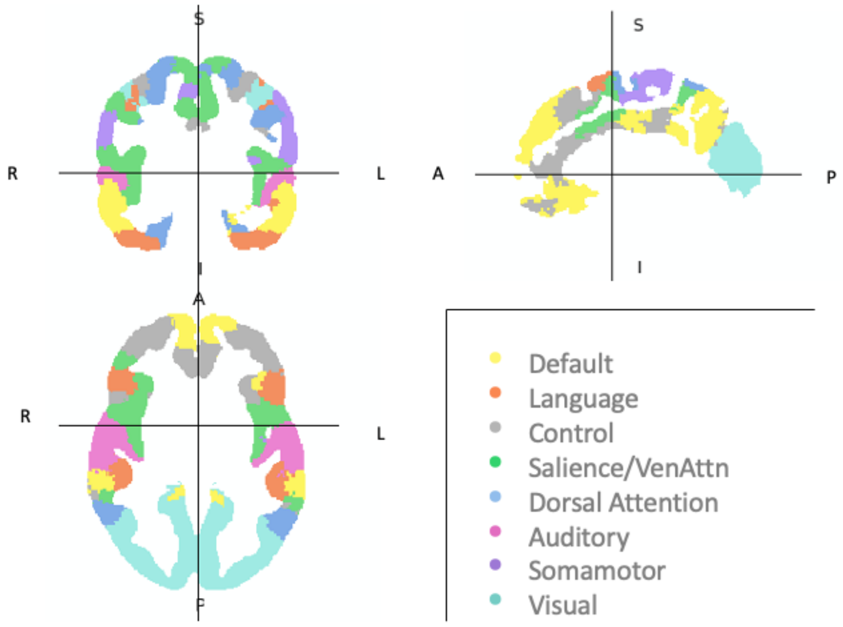


Figure 3.1: An illustration of the 8 nodes of each hemisphere, 16 nodes in total. The networks are coloured according to corresponding labels. The coordinates are defined as follows: L-R(left-right hemispheres), P-A(posterior-anterior axis) and I-S (inferior-superior axis). The brain is visualised in the same three anatomical planes as for Figure 1.2: axonal (lower left), coronal (upper left) and sagittal (upper right). A clarification of how the networks can be found in Appendix B.2. The figure was made using FSLeyes [60], see Appendix C.1.

tationally intensive. The network of $p = 17$ nodes is constructed by a merging of nodes across the two hemispheres; see Appendix B.2. It is not convenient to visualise the nodes' position for this network. Due to the advantages described above, the next chapter primarily employs the network of 16 nodes as a sample parcellation technique for visualisation.

Figure 3.2 showcases the time series extracted from all parcellation schemes. The data come from a representative control subject's first session. Recall from Chapter 2 that $T = 315$ refers to the number of BOLD observations in one signal or equivalently the number of volumes acquired during one scan. The number of nodes in a network, $p = 8, 16, 17, 34, 100, 300, 500$ depends on the selected parcellation scheme. Further details on how time series are extracted from NIfTI-1 data is available in Appendix C.2.

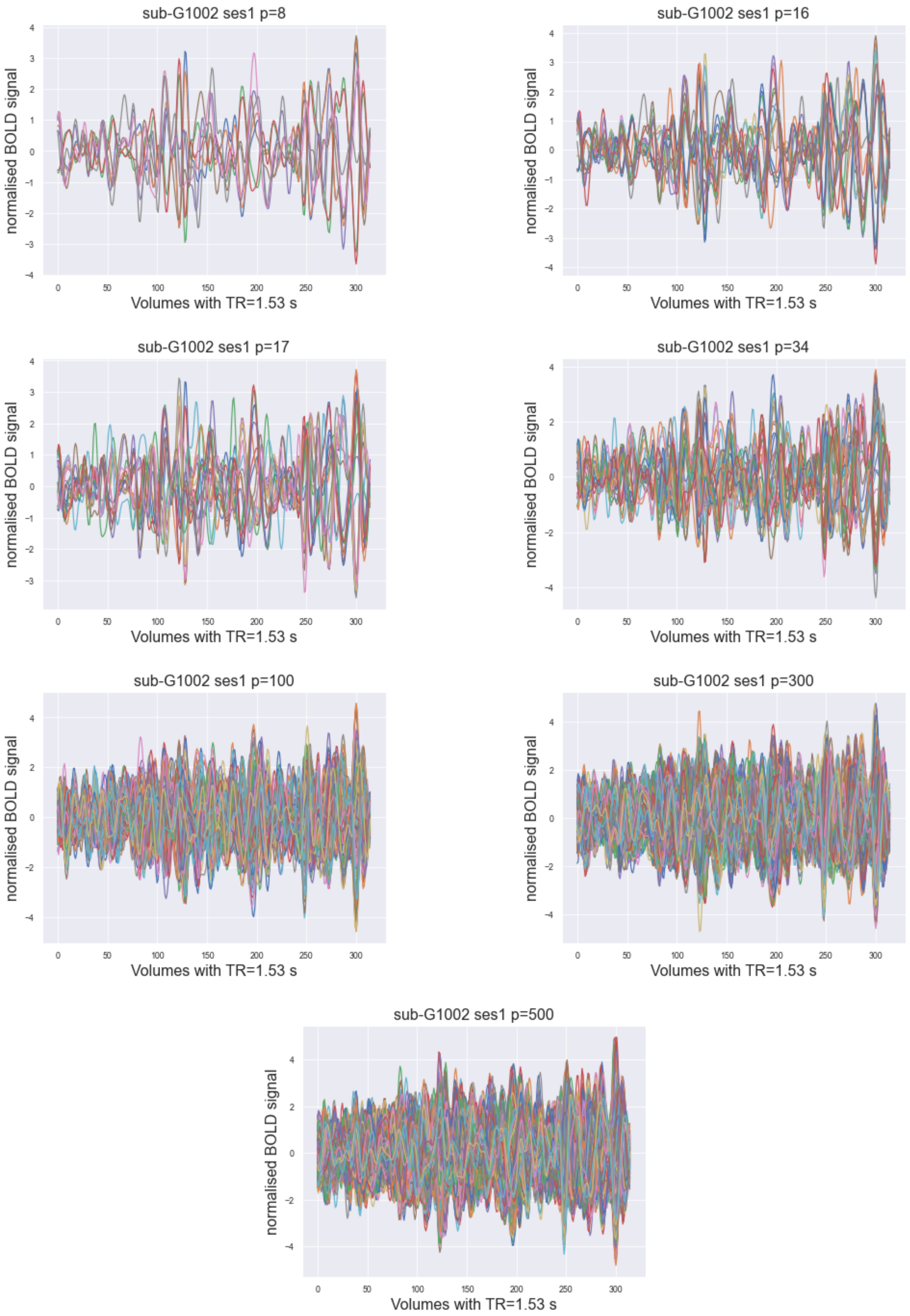


Figure 3.2: There are seven parcellation schemes of p nodes or brain regions. Each node holds a representative time series corresponding to the first PC of the voxelwise, normalised BOLD signals within the node (see Section 2.1). The time series consist of $T = 315$ observations or equivalently volumes, with a TR of 1.53s. Data is supplied by a control subject's first session: sub-G1002.

3.2 Analysis of connectivity measures

In Section 2.2 we presented three different approaches to estimating functional connectivity. When analyzing FC, the strength, directness, and directionality of edges in a brain network should all be taken into account; see Section 1.2.2. The FC approaches differ in terms of which network features are considered. The functional connectivity between node i and node j is represented by a weighted edge, w_{ij} belonging to $\mathbf{W} \in \mathbb{R}^{pxp}$. Recall that for connectivity methods yielding symmetric weight matrices, such that $w_{ij} = w_{ji}$, we examine the lower triangular part of $\mathbf{W}$. For the DCM, which yields bidirectional edges, the diagonal elements of $\mathbf{W}$, or equivalently the self-connections, are omitted. The input of all implemented methods is the time series data, $\mathbf{X} \in \mathbb{R}^{nxp}$, presented in Section 2.1 and visualised in Figure 3.2.

3.2.1 Pearson correlation

The simplest of the FC measures considered in this work is the sample correlation matrix as defined in Section 2.2.1. The lower triangular weight matrices are presented in 3.3 for all parcellation schemes of a representative control subject. Each parcellation scheme's node order, $j = 1, \ldots, p$, matches the parcel or network order specified in Appendix B.2. For the smallest parcellation schemes of $p = 8$ and $p = 16$, the node order is also specified via the labels in Figure 3.1. Brain regions close to each other in terms of spatial location hold neighbouring indexes. Observe that there is an increased correlation between adjacent brain regions or nodes.

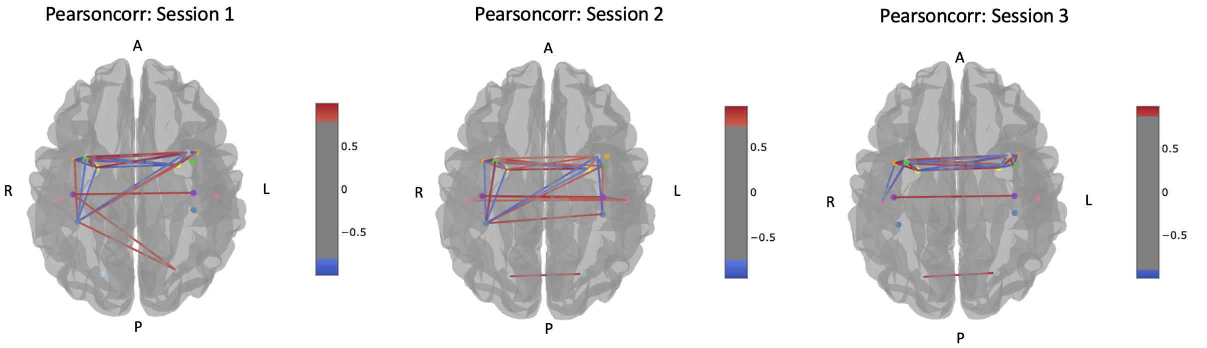


Figure 3.4: The brain networks of $p = 16$ nodes from session 1 (left), session 2 (centre) and session 3 (right). The edges belong to the correlation matrices in Figure 3.3. The 25% absolute strongest weighted edges are visible. The coordinates are as follows: L-R(left-right hemispheres), P-A(posterior-anterior axis) and I-S (inferior-superior axis). The nodes are coloured according to the labels in Figure 3.1. The locations of the nodes correspond to the centre of gravity overall enclosed voxels. The parcellation scheme for $p = 16$ nodes can cover separate brain areas. This is because the Kong-Schaefer parcellation joins nodes based on functionality rather than location. Consequently, nodes are crammed together in the centre of the brain. Coordinates are extracted using FMRIB Software Library (FSL) software, and the plots are visualised using Nipype, see Appendix C.1. Data is supplied by sub-G1002.

The weighted edges in each of the plots in Figure 3.3 indicate a presence of both correlations and anticorrelations between nodes corresponding to different brain regions. We defined functional connectivity (FC) in Section 1.2 as the statistical dependency between measured activity from brain signals [5]. A strong correlation between two nodes signifies a strong FC. Consequently, there is a synchrony of neural activity in the two corresponding brain regions. The neurophysiological understanding of the role of anticorrelation is still unclear. Because these negative correlations may be important according to a previous study on the human brain's functional organization [61], these are kept for analysis of trt reliability.

The spatial organization of the FC network of $p = 16$ nodes from three sessions of a control subject is visualised in Figure 3.4. The 25% strongest weights in absolute value are showcased as edges. The locations of the nodes signify the mass centre of the contained voxels. Equivalent nodes across hemispheres are coloured and labelled identically owing to homogeneous function; see Figure 3.1. Observe that the strongest connections are between related nodes of each hemisphere. Figure 3.5 shows the distribution of all weights of session 1, 2 and 3 for the same control as visualised in Figure 3.3. There are some differences between the topology of the three networks depicted in Figure 3.4. Figure 3.5 confirms this heterogeneity between sessions, as several weights from identical node pairs of distinct sessions do not match in value.

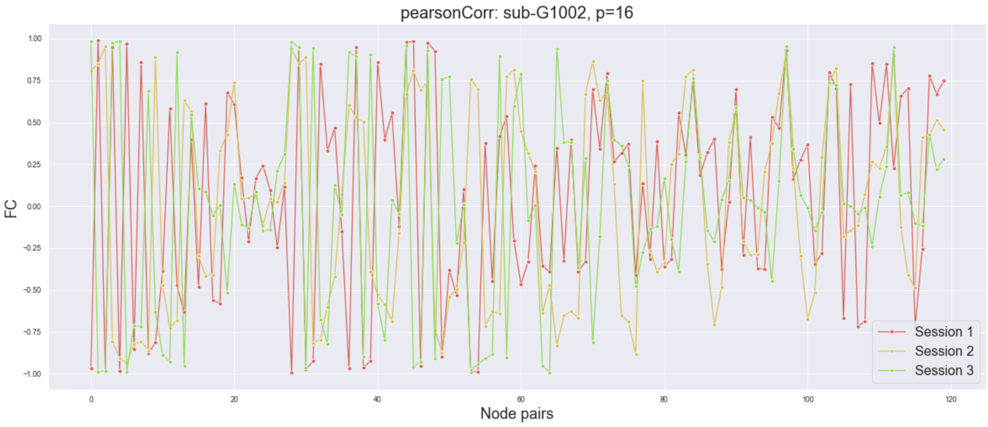


Figure 3.5: The correlation weights from the lower triangular of $\mathbf{W}$ of session 1, 2 and 3 of sub-G1002. The networks consist of $p = 16$ nodes. There are $((p-1)\times p)/2 = 120$ unique node pairs.

3.2.2 Partial correlation with Graphical lasso

Partial correlation, as defined in Section 2.2.2, regresses all variables when measuring the pairwise correlation between time series. The FC method accounts for both strength and directness of the edges in a brain network. Figure 3.3 of the correlation matrix proves that $\mathbf{X}$ suffers from multicollinearity; several nodes are highly correlated. When some predictors in a regression model are correlated, estimating the coefficients may be challenging. A solution to this problem is penalizing the parameters using lasso. The partial correlation coefficient is defined from a scaling of Θ, see (2.9). The precision matrix is estimated by maximizing the regularized log-likelihood function in (2.12) with respect to the Θ. The optimization method is called GLasso and is implemented in R, see Appendix C.1. Furthermore, a 5-fold-cross validation (CV) with loss function, the negative log-likelihood in (2.11), is performed to find the optimal tuning parameter λ. Figure B.2 in Appendix B.3 presents the CV error for all choices of λ for each parcellation scheme.

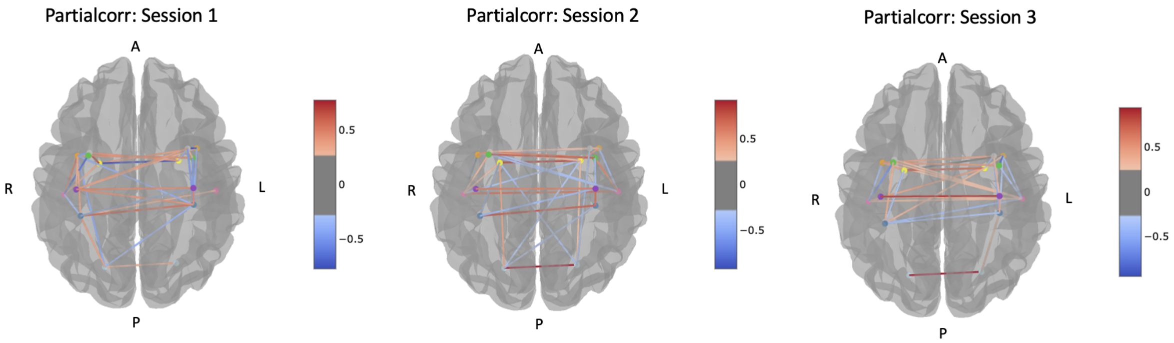


Figure 3.7: The brain networks of $p = 16$ nodes from session 1 (left), session 2 (centre) and session 3 (right). The edges belong to the partial correlation matrices presented in Figure 3.6. The 25% absolute strongest weighted edges are visible. The coordinates are as follows: L-R(left-right hemispheres), P-A(posterior-antieroir axis) and I-S (inferior-superior axis). The nodes are coloured according to the labels in Figure 3.1. The locations of the nodes correspond to the centre of gravity overall enclosed voxels. The parcellation scheme for $p = 16$ nodes can cover separate brain areas. This is because the Kong-Schaefer parcellation joins nodes based on functionality rather than location. Consequently, nodes are crammed together in the centre of the brain. Coordinates are extracted using FSL software, and the plots are visualised using Nipype, see Appendix C.1. Data is supplied by sub-G1002.

The lower triangular partial correlation matrices are showcased in Figure 3.6 for each choice of p. Notice that the degree of sparsity is higher for the precision matrices compared to the sample correlation matrices presented in Figure 3.3. Increased sparsity is a result of the lasso penalty on the size of the coefficients, defined in (A.27). Sampling variations in fMRI often lead to FC estimates that are never exactly zero, even for two conditionally independent variables. Weak edges are removed when sparsity is increased. A sparse network is, therefore, usually preferred for interpretability. The high degree of sparsity among the partial correlation coefficients is also visible in Figure 3.8, especially when compared with Figure 3.5. Most of the node pairs in Figure 3.8 contain edges with weights close to zero. Across sessions, there are some identical node pairs with edges of considerable weights of opposite signs. Such major, temporal changes in individual edges indicate poor reliability. An example of one such node pair is the close-to 30th node pair. However, the overall variance between sessions, examined by looking at the distance between the three lines, is significantly lower in Figure 3.8 compared to Figure 3.5.

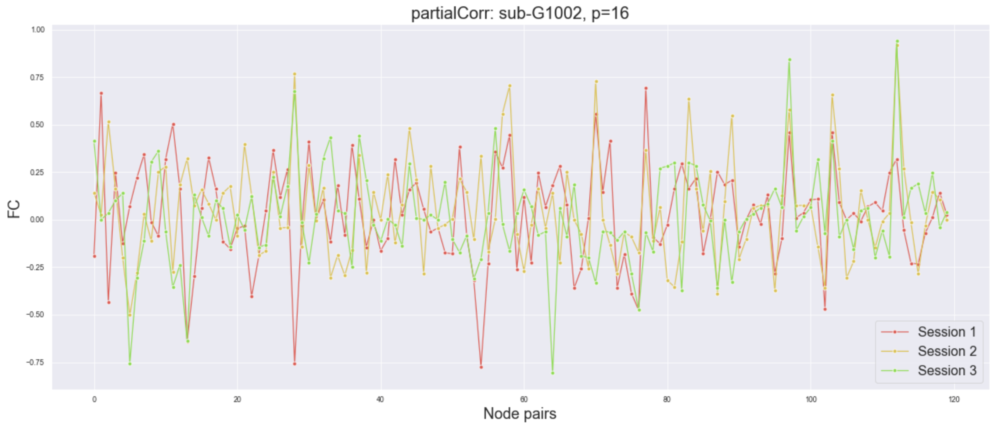


Figure 3.8: The partial correlation weights from the lower triangular of $\mathbf{W}$ of session 1, 2 and 3 of sub-G1002. The networks consist of $p = 16$ nodes. There are $((p-1)\times p)/2 = 120$ unique node pairs.

Figure 3.7 illustrates the spatial organization of the FC network of $p = 16$ nodes for three sessions. Similarly as for Figure 3.4, only the 25% absolute strongest weights are showcased. The locations of the nodes signify the mass centre of the contained voxels. Equivalent nodes across hemispheres are coloured and labelled identically owing to homogeneous function; see Figure 3.1. Once again, we observe that the strongest partial correlations are interhemispheric (similar to in Figure 3.4 presented in the last section). The network topology from the partial correlation coefficients looks similar across sessions. This observation is in agreement with the reduced variance of weights between sessions depicted in Figure 3.8 compared to Figure 3.5.

3.2.3 Spectral Dynamical Causal Model (spDCM)

The last of the FC methods assessed in this work is the spectral spDCM, described in Section 2.2.3. Unlike the two former models, the spDCM tries to account for the underlying biophysical processes relating neural activity to the measured oxygen level of the blood flow. As a result, the method yields a network of edges containing strength, directness and directionality. However, in large data settings, the spDCM tends to be computationally expensive compared to the two formerly introduced stationary models. More concretely, the inversion scheme, trying to maximize the negative free energy $\tilde{F}(q,\theta)$ in (2.27), is computationally demanding due to a large number of free parameters contained. Prior distributions of the parameters included in θ must be specified for the EM algorithm to have initial values. For the hemodynamic parameters in θ we utilized the default priors implemented in Statistical Parametric Mapping (SPM) [62]. A $p\times p$ matrix of ones was used as the prior for the intrinsic connectivity matrix, $\mathbf{A}$ in (2.20) to account for all potential connections. The spDCM is restricted to run on the three smallest parcellation schemes examined in this study. That is for $p = 8, 16$ and 17 nodes.

Figure 3.9 presents heatmaps of the final posterior estimate of the intrinsic coupling matrix, **A** for the three smallest parcellations. Notice that only the off-diagonal elements are included. In comparison to the two correlation-based measures in Figures 3.3 and 3.6, the scale of magnitude of the weights in Figure 3.9 has decreased. The weights in **A** are in units of Hertz, defining the rates of change between regions [55]. Recall that the resulting weights of the spDCM represent the effective connectivity between nodes; see Section 1.2.2. Notice that the sparsity of the coupling matrices increases significantly with more nodes.

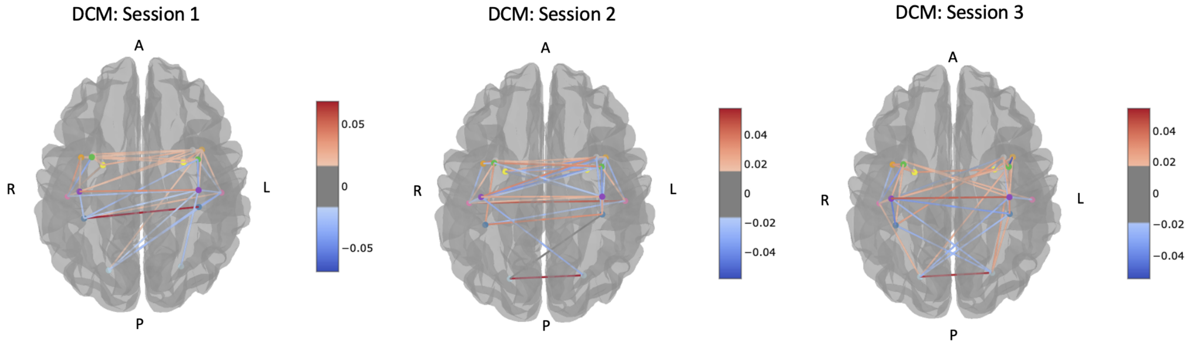


Figure 3.10: The brain networks of $p = 16$ nodes from session 1 (left), session 2 (centre) and session 3 (right). The edges belong to the intrinsic coupling matrices presented in Figure 3.9. The 25% absolute strongest weighted edges are visible. The coordinates are as follows: L-R(left-right hemispheres), P-A(posterior-antieroir axis) and I-S (inferior-superior axis). The nodes are coloured according to the labels in Figure 3.1. The locations of the nodes correspond to the centre of gravity overall enclosed voxels. The parcellation scheme for $p = 16$ nodes can cover separate brain areas. This is because the Kong-Schaefer parcellation joins nodes based on functionality rather than location. Consequently, nodes are crammed together in the centre of the brain. Coordinates are extracted using FSL software, and the plots are visualised using Nipype, see Appendix C.1. Data is supplied by sub-G1002.

Figure 3.10 visualises the 25% absolute strongest weights, as well as the spatial organization of the 16 nodes network. The node locations represent the mass center of all enclosed voxels. Due to homogeneous function, equivalent nodes across hemispheres are coloured and labelled identically, as seen in Figure 3.1. Similarly as for Figures 3.4 and 3.7, the inter-hemispheric connections dominate in Figure 3.10. Figure 3.11 showcases the weights of the edges between all pairs of nodes. Recall that the spDCM results in twice as many connections as the two former methods due to the edges being bidirectional. From both of Figures 3.10 and 3.11 we observe that the majority of weights are relatively consistent across sessions. In contrast to the partial correlation coefficients in Figure 3.8, the most extreme weights from the spDCM do not have contradictory signs.

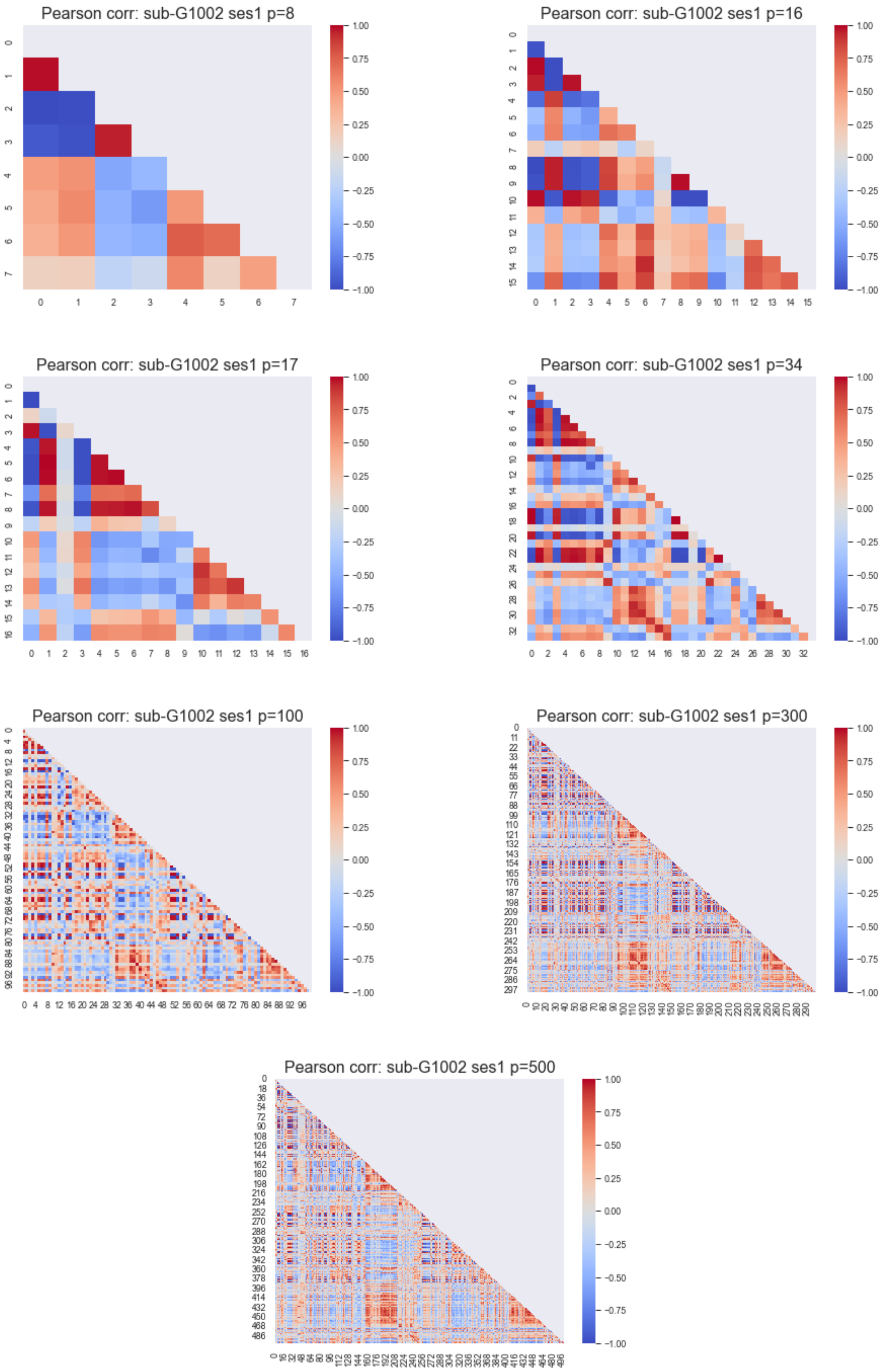


Figure 3.3: Lower triangular weight matrices from Pearson correlation. Data is supplied by sub-G1002, session 1. The colour bar corresponds to the strength of Pearson correlation coefficients.

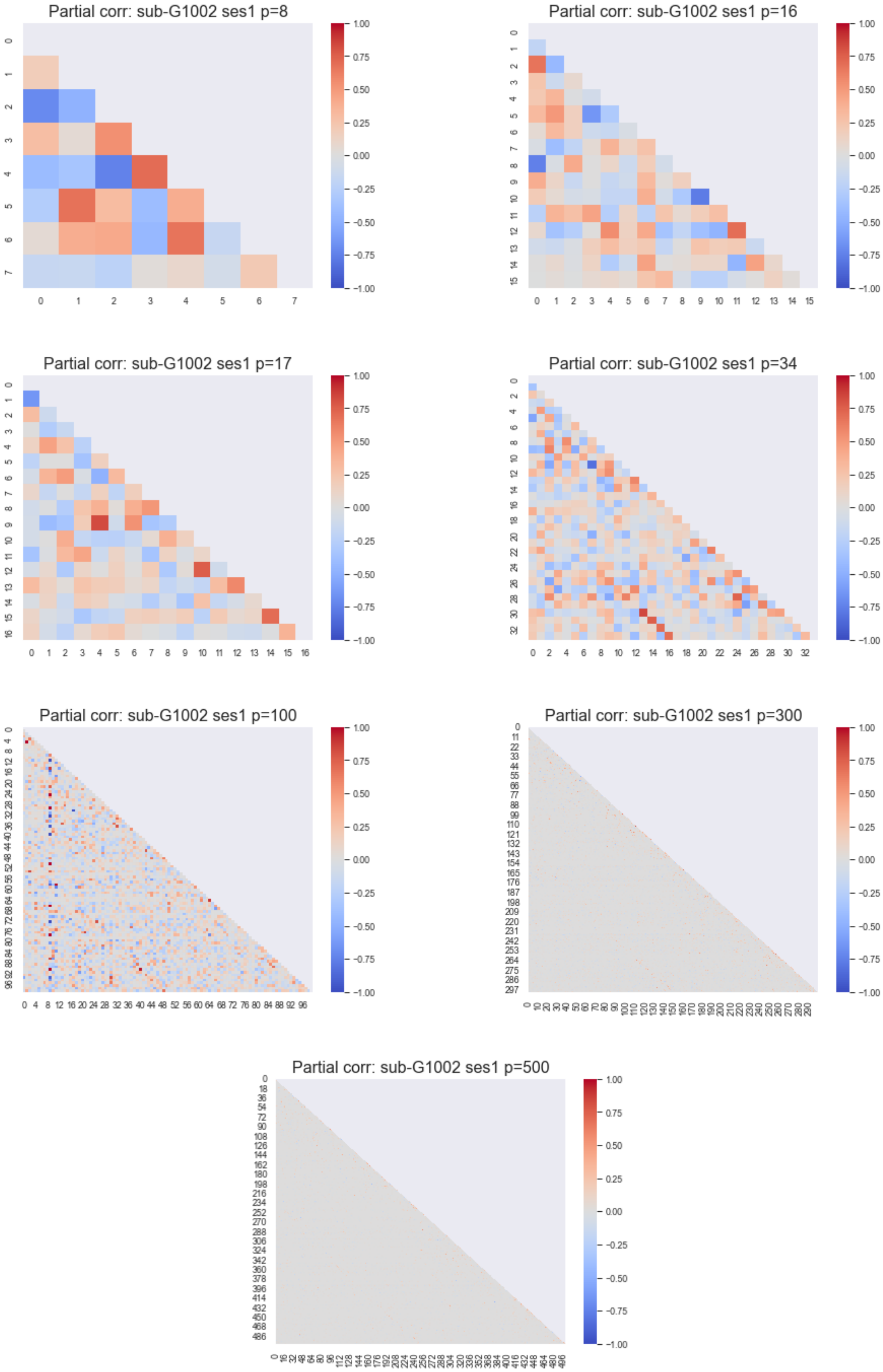


Figure 3.6: Lower triangular weight matrices from Partial correlation. Data is supplied by sub-G1002, session 1. The colour bar corresponds to the strength of the partial correlation coefficients.

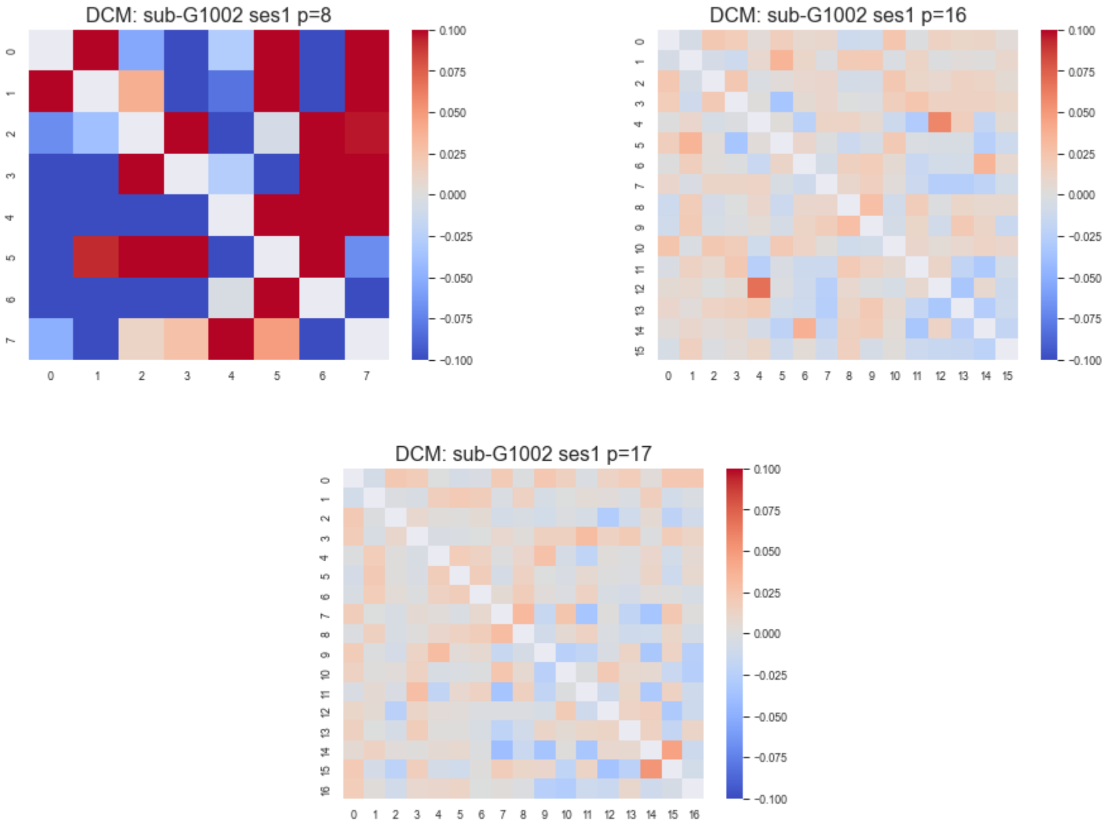


Figure 3.9: The weight matrices from DCM for $p = 8, 16, 17$ ($p > 17$ not available). Data supplied by sub-G1002 session 1. The colour bar corresponds to the strength of the weight coefficients.

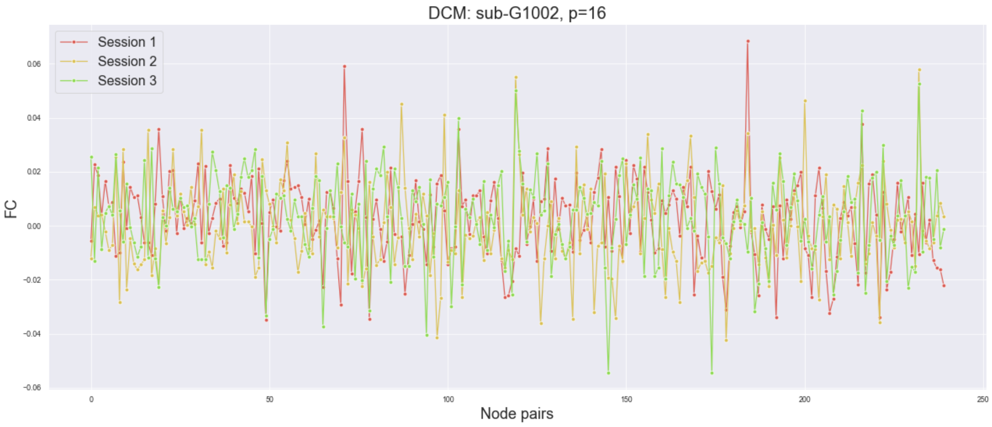


Figure 3.11: The intrinsic coupling weights from the off-diagonal elements (that is, entries of a matrix that are not in the diagonal) of $\mathbf{W}$ from session 1, 2 and 3 of sub-G1002. The networks consist of $p = 16$ nodes. There are $(p-1) \times p = 240$ unique node pairs.

3.3 Analysis of Test-Retest reliability

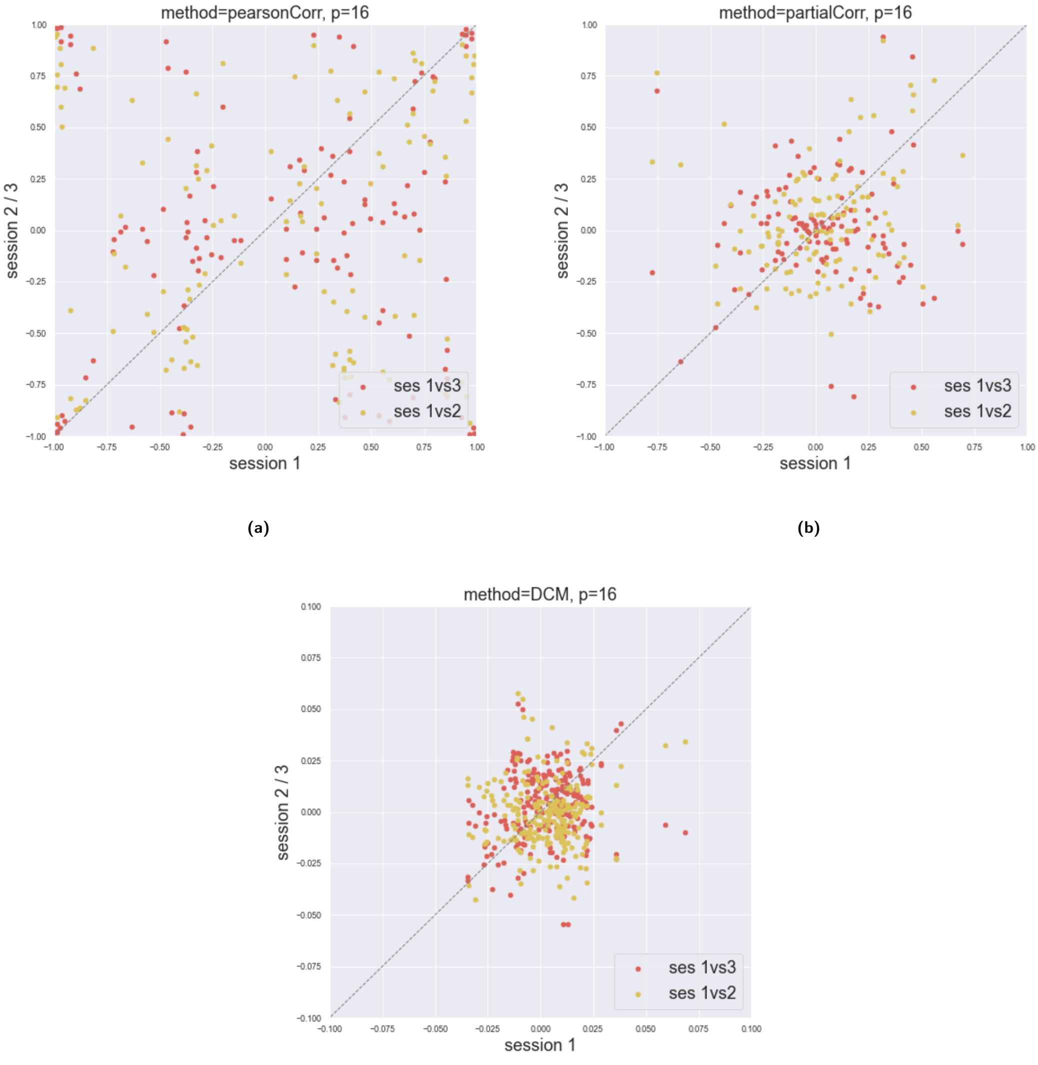
The Test-Retest reliability of a method refers to its ability to give consistent results under repeated measurements given similar conditions. The same scanning protocols are used across participants and sessions to secure similar settings. Since we expect the functional network topology of a healthy brain to stay unchanged for three months, the estimated reliability scores give insight into the precision of our chosen analysis methods and the rest of the pipeline presented in Chapter 2.

p	Method	r(ses1, ses2)	r(ses1, ses3)
8	Pearson correlation	0.18	0.16
16		0.18	0.16
17		0.14	0.28
34		0.14	0.26
100		0.28	0.18
300		0.24	0.24
500		0.25	0.25
8	Partial correlation	0.16	0.29
16		0.17	0.14
17		0.11	0.20
34		0.23	0.23
100		0.10	0.08
300		0.23	0.21
500		0.25	0.20
8	DCM	0.15	0.03
16		0.06	0.15
17		0.19	0.26

Table 3.1: The estimated reliability scores between sessions 1 and 2 (1-month interval) and sessions 1 and 3 (3-month interval) for all methods and parcellations combinations. The columns are: the number of nodes p, the chosen method, and the estimated average-across subjects reliability scores, r(ses1,ses2) and r(ses1,ses3).

There are three repeated measurements, referred to as session s for $s = 1,2,3$, and three control subjects, referred to as sub-G1002, sub-G1014, sub-G1018 (see Table 2.1). Sessions 1 and 2 are acquired one month apart, while for sessions 1 and 3, the time interval equals three months. Earlier, we mentioned that the measured brain networks must be reliable to observe functional rewiring after a stroke. The trt reliability score is reflected by the correlation coefficient between two vectorized weight matrices of different sessions as described in Section 2.3.1. Recall that the Pearson correlation coefficient reflects the degree of a linear relationship between two variables; see Section 2.2.1. The bivariate relationship between weights from sessions 1 and 2 and sessions 1 and 3 is depicted in the Figures 3.12a, 3.12b and 3.12c. When plotted against each other, two variables are linearly correlated if all observations are distributed around the diagonal line passing through the origin with a slope of 1. In comparison to the weight distributions in Figures 3.12b and 3.12c, the weights in Figure 3.12a are more dissimilar. That is, there are more outliers or equivalently extreme values. Outliers are observed more frequent as the sample's variance increases. This discovery is accompanied by Figure 3.8, which shows that multiple weighted edges differ across sessions.

Let $r(\text{ses } k, \text{ses } l), k = 1, l = 2, 3$, quantify the net similarity between the two weight matrices $\mathbf{W}^{(k)}$ and $\mathbf{W}^{(l)}$. Figure 3.13 presents the reliability score of the three methods for different parcellation schemes. The standard deviation across subjects is visualised as error bars around the means. The average score across subjects is also showcased in Table 3.1 for all methods and parcellation combinations. Figures 3.14a, 3.14b and 3.14c further examine the variability between all subject‘s reliability score. Individual controls' scores are visualised separately for each FC method. When there are fewer nodes, the between-subject variance of $r(\text{ses } k, \text{ses } l), k = 1, l = 2, 3$ is higher. The length of the error bars in Figure 3.13, as well as the distance between the lines in Figures 3.14a, 3.14b and 3.14c demonstrate this. Next, compare the performance of the different FC methods in terms of trt reliability. Notice that not only the between-subject variance converges as the number of nodes rises, the trt reliability for the two correlation-based methods also approached closer; see Table 3.1.

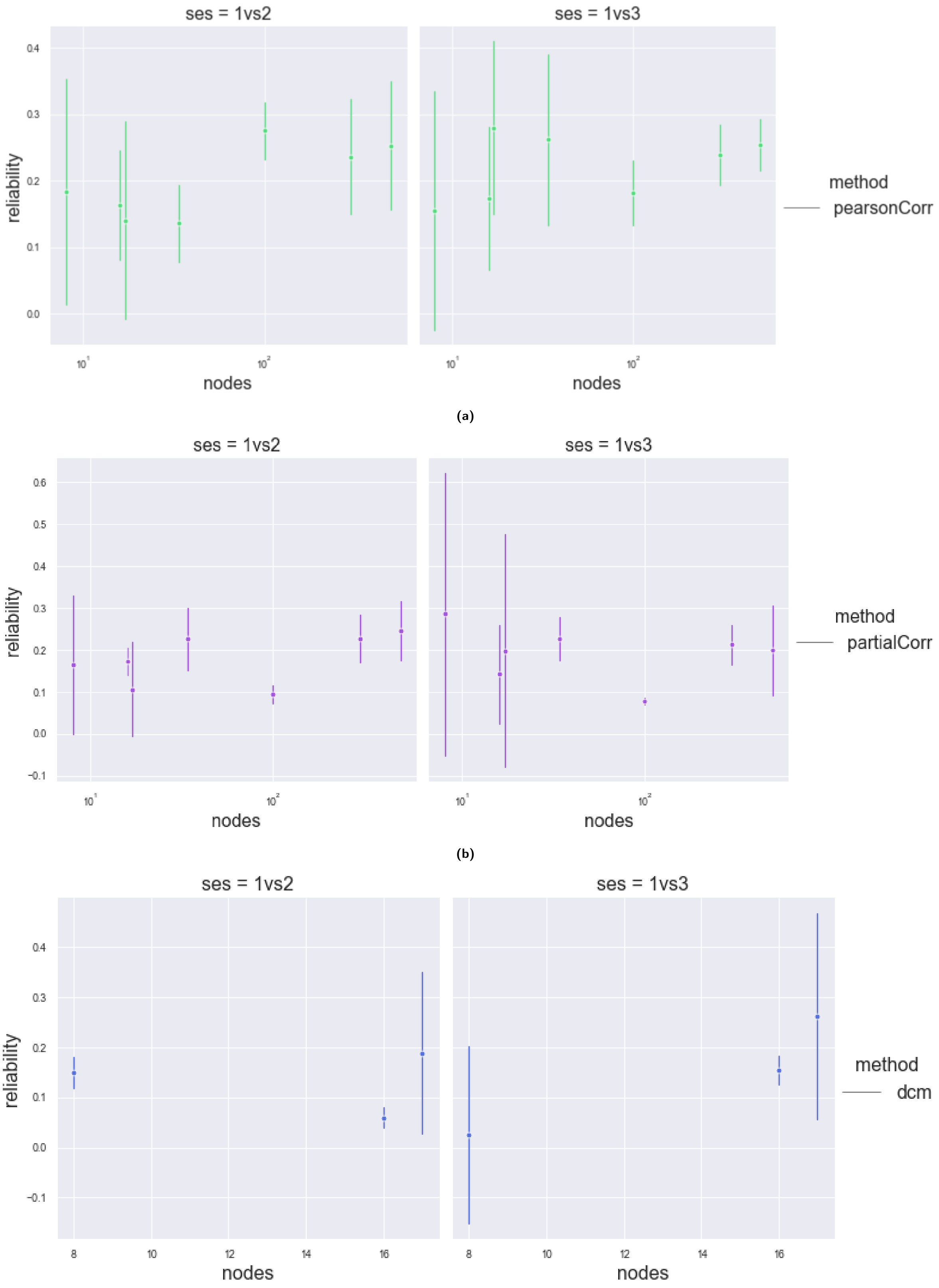


(a)

(b)

(c)

Figure 3.12: Scatter plots of the weights from session 1 vs. session 2 (interval: 1 month) and session 1 vs. session 3 (interval: 3 months) for each FC method studied in this work. The data belongs to one control subject. All networks consist of 16 nodes with (a) weights from Pearson correlation, (b) weights from Partial correlation and (c) weights from spDCM. The dotted, diagonal line resembles a straight line, cutting through the origin with a slope of 1.

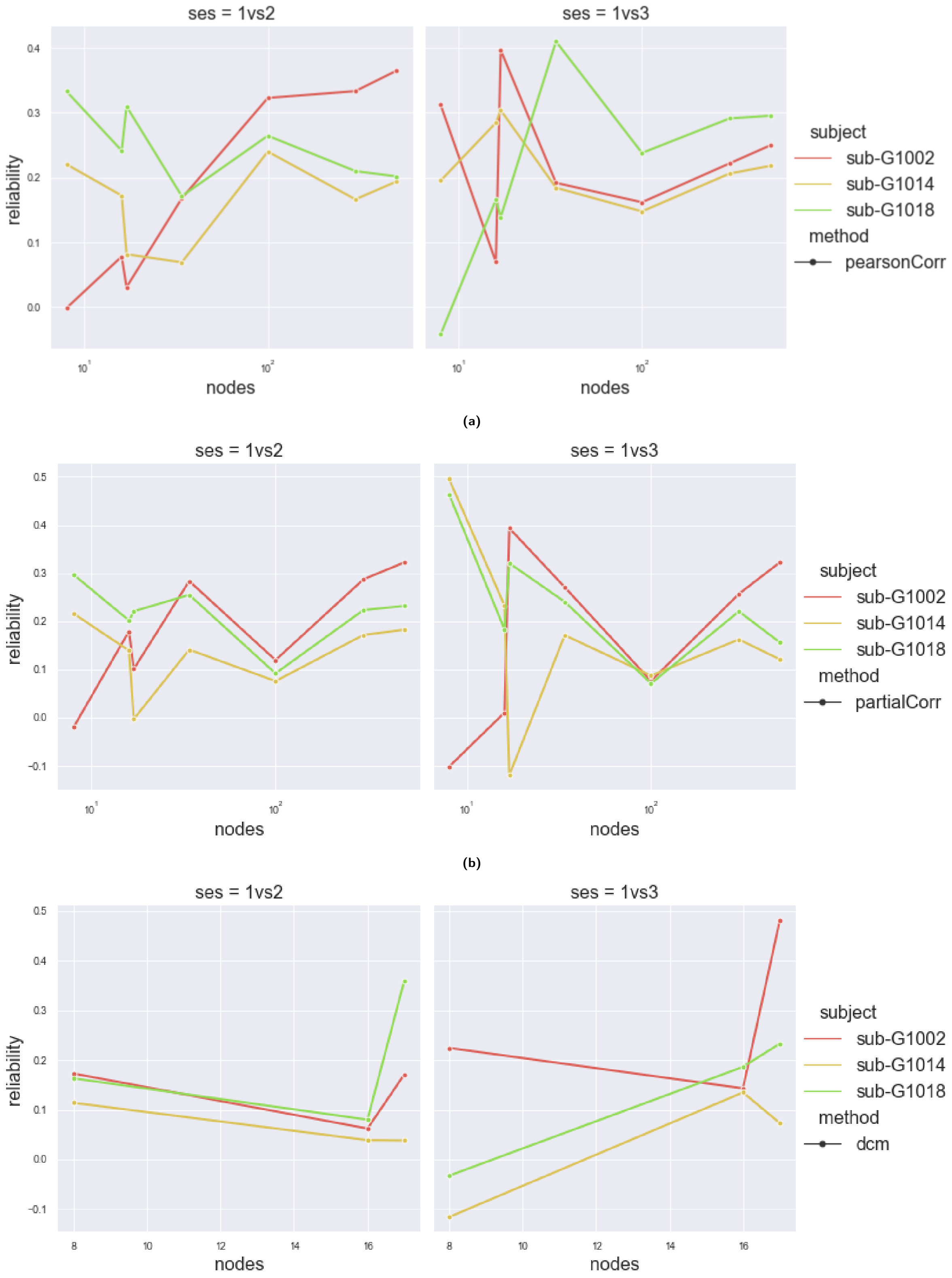


(a)

(b)

(c)

Figure 3.13: The mean across subjects Test-Retest (trt) reliability of sessions 1 and 2 (left) and sessions 1 and 3 (right) as a function of different number of nodes. The standard deviation across subjects is indicated by an error bar around the mean. Each row corresponds to a separate FC method: Pearson correlation (a), Partial correlation (b), and spDCM (c).



(a)

(b)

(c)

Figure 3.14: The Test-Retest reliability between sessions 1 and 2 (left) and sessions 1 and 3 (right) as a function of different number of nodes, and for each FC method: Pearson correlation (a), Partial correlation (b), and spDCM (c). The reliability scores from individual control subjects are visualised as separate lines.

3.4 Power Analysis

In the last section, the Test-Retest reliability scores, quantifying the net similarity of the entire FC networks between sessions, were presented. Now, encompass the individual edges and how they differ across sessions of the same subjects. We conduct edgewise power analyses on parcellations p and method choice combinations.

Let $d_i^{(\text{sub } g)}$ denote the weight difference between the two first sessions for control subject g, as specified in (2.33). Since we have no available stroke data related to this study, we instead utilize data from control subject‘s session 1 and session 2 for conducting the power tests defined in Section 2.3.2. The hypothesis in (2.35) assume that $d_1^{(\text{sub } g)}, \dots, d_e^{(\text{sub } g)}$ are drawn from a normal distribution with zero mean and unknown variance σ^2. The validity of this assumption can be examined by looking at Figure 3.15. The distribution of weight differences between sessions 1 and 2 for all individuals is shown in Figure 3.15a, demonstrating that this assumption is valid for each of the FC techniques in this study.

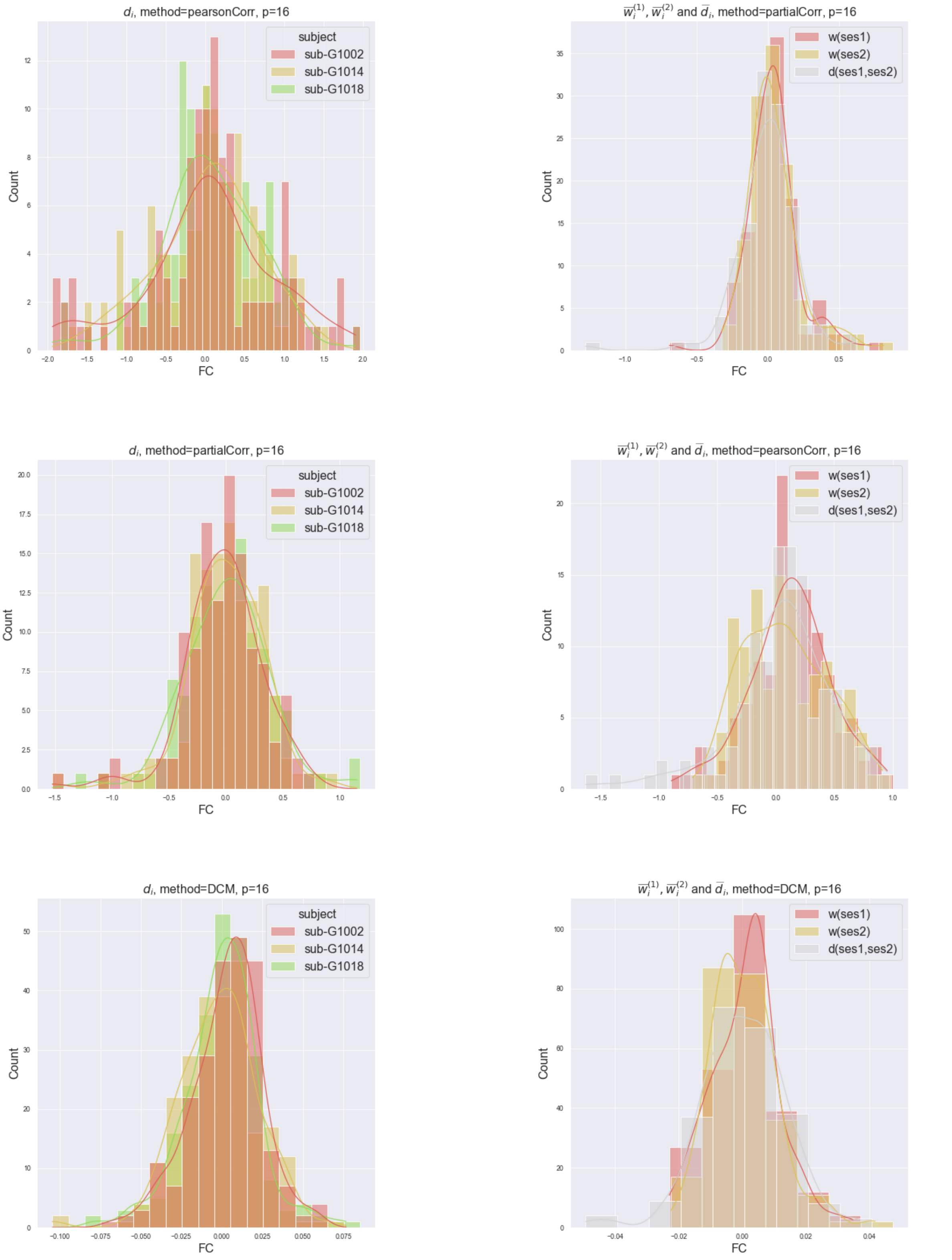
Notice that the changes in temporal weights, represented by $d_i, i = 1, \dots, e$ for all edges, are uniformly distributed around zero. Notice that the distribution of d_i in the first row of Figure 3.15a has a heavy tail compared to the other FC approaches. In other words, the probability of getting a very large d_i is higher for the weights retrieved from the Pearson correlation. This observation is consistent with the findings in Section 3.2. Figure 3.15b showcases the mean weights from the two first sessions and the mean differences across subjects. Recall that during three months, we assume the functional brain network of controls to be constant. As a result, the distribution of $\bar{d}_i^{(c)}$ should have a higher frequency of observations equal to zero than the distribution of the averaged weights, denoted $\bar{w}_i$. The distribution of weights and weight differences in Figure 3.15b have a mean zero. When comparing the two distributions: $\bar{d}_i^{(c)}$ to have more outliers than $\bar{w}_i$. This conflicts with our assumption that the functional connections of controls are constant for three months.

Based on the test in (2.35), we compute how many patients must be averaged over for the expected temporal change in FC following a stroke to be significant. Recall that the standard deviation, σ, and effect size, $\delta(q)$, is estimated based on data from control participants, as stated in Section 2.3.2. The predicted sample size, $n^{(\text{s})}$, is determined iteratively using (2.39) for all parcellation schemes p and methods combinations. Based on the 90% quantile, that is $q = 0.9$, of the sorted weight list, Table 3.2 provides a summary of $n^{(\text{s})}$ for $\alpha = 0.05$, $\beta = 0.2$. For all FC methods, notice that the estimated standard deviations, $\hat{\sigma} = s$, and estimated effect size $\delta(q)$, both drop significantly as the number of nodes p increases. Still, the sample size $n^{(\text{s})}$ rises along with the number of nodes due to the Bonferroni correction resulting in a small α_i, defined in (2.38). More samples are needed when we accept fewer Type I-errors.

Due to the high amount of sparsity present in the precision matrix from (2.12), the effect size $\delta(q)$, which depends on the magnitude of the weights, shrinks to zero. As a result, the estimated sample size of the network of $p = 500$ nodes with edges equal to the partial correlation coefficients defined in (2.9), diverges. This is a disadvantage of utilizing a median-based method to estimate $\delta(q)$. The predicted sample sizes for different selections of significance level α, power $1-\beta$, and quantiles q impacting the resultant effect sizes $\delta(q)$ are shown in Figures 3.16a, 3.16b and 3.16c. An increase in either α or β, meaning that we tolerate a higher probability of Type I and Type II-errors, leads to reduced estimated sample size. Note that the magnitude of the effect size δ significantly impacts the predicted number of stroke patients. Our results suggest that the amount of change in temporal functionality of a brain network following a stroke, as opposed to healthy control, is the most significant variable in the hypotheses test in (2.35).

p	Method	$n^{(s)}$	$\delta(q)$	$\hat{\sigma}$
8	Pearson correlation	21	0.80	0.77
16		24	0.80	0.70
17		21	0.80	0.71
34		23	0.72	0.63
100		26	0.59	0.46
300		32	0.55	0.43
500		33	0.53	0.41
8	Partial correlation	18	0.5	0.43
16		26	0.36	0.36
17		25	0.37	0.37
34		31	0.29	0.29
100		43	0.21	0.23
300		264	0.02	0.02
500		∞	0	0.03
8	DCM	15	0.29	0.21
16		34	0.02	0.02
17		27	0.02	0.02

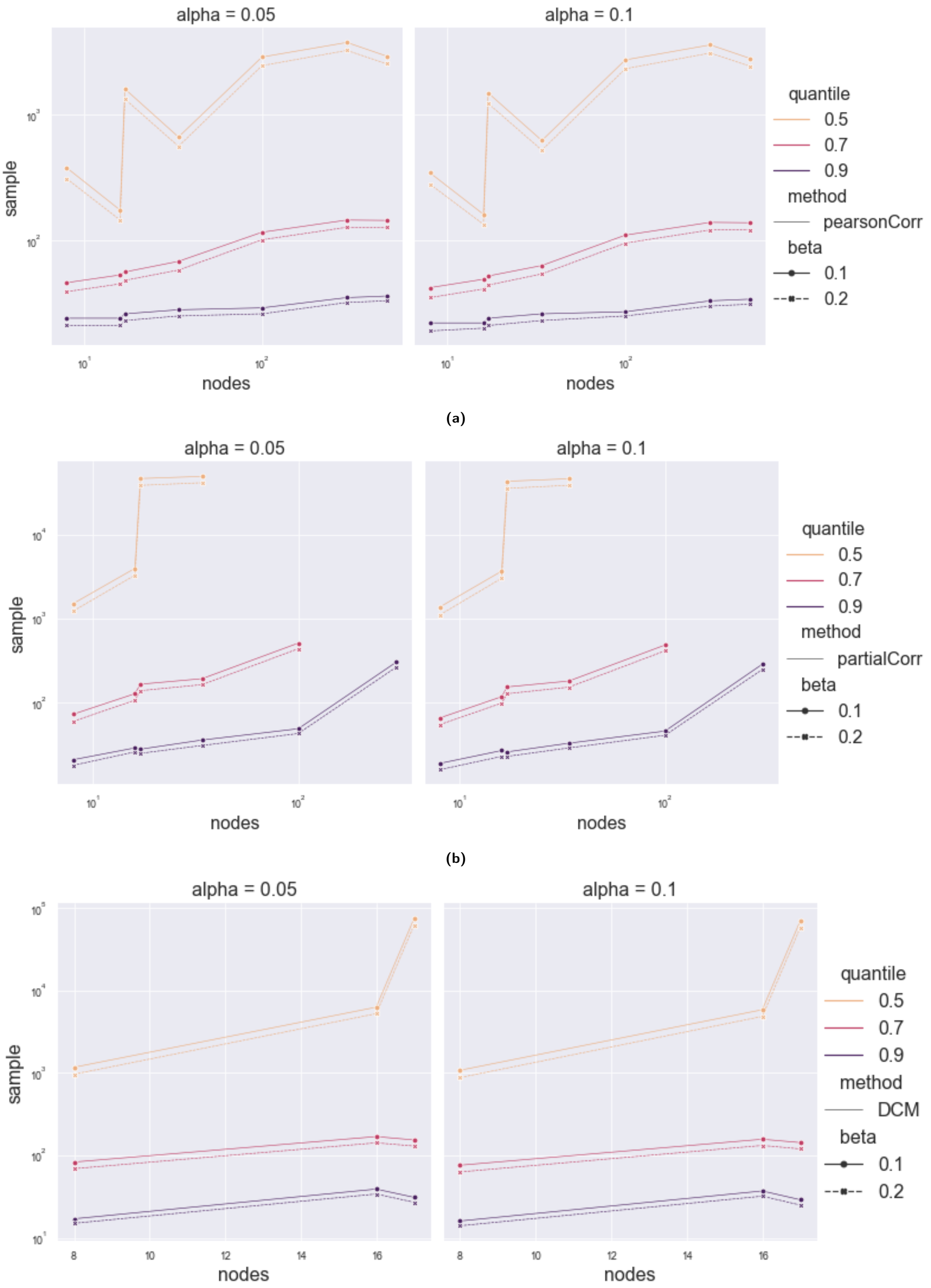
Table 3.2: The estimated sample sizes for all methods and parcellations combinations with $\alpha = 0.05$, $\beta = 0.20$ and $q = 0.90$. The columns are: the number of nodes p, the estimated sample size of stroke patients $n^{(s)}$, the effect size $\delta(q)$ and the estimated standard deviation $s = \hat{\sigma}$. See (A.45).



(a)

(b)

Figure 3.15: The weights from networks of $p = 16$ nodes. Each row corresponds to one FC method. The leftmost plot (a) displays the distribution of differences $d_i = w_i^{(1)} - w_i^{(2)}$ for all control subjects. Bin size is 0.1. The rightmost plot (b) gives the distribution of mean weights, $\bar{w}^{(\text{ses 1})}$ and $\bar{w}^{(\text{ses 2})}$, together with the mean distances $\bar{d}$ across subjects. Bin size is 0.01. The curves in (a) and (b) are the kernel density estimates of the distributions.



(a)

(b)

(c)

Figure 3.16: The estimated sample sizes for each of the three FC methods as a function of the number of nodes, p. The resulting $n^{(s)}$ from different choices of significance level α, power $1-\beta$ and quantiles q (which is affecting the effect size $\delta(q)$) are showcased.

Chapter 4

Discussion

The objectives of the present study were to investigate (1) the required sample sizes of stroke subjects necessary to detect significant time-dependent changes in the functional brain network using rs fMRI;(2) the trt reliability of three FC methods and seven parcellation schemes. We present some more noteworthy findings and their implications for the related study at St.Olav Hospital. Then limitations that have impacted the outcomes are discussed. Several exciting mathematical possibilities for the current research are addressed in connection to future progress. A conclusion for this thesis is provided at the end of this chapter.

4.1 Findings

The conducted power analyses in Section 3.4 suggest that there is a need for a large number of stroke patients in order to find significant temporal FC changes after stroke. As expected, the sample sizes increase with smaller effect sizes. Consequently, the study relies on either enrolling enough volunteers or observing significant differences between the two group‘s temporal connectivity changes. Keep in mind that it is very challenging to recruit volunteering stroke patients that meet the clinical trial requirements mentioned in Section 2.1 [14]. The predicted sample sizes of all parcellation and methods combinations are presented in Table 3.2 for a significance level equal to 0.05 and statistical power of 80%. The estimated effect size is set to the 90th percentile of weights from all controls in this study, meaning that the expected change in FC post-stroke equals the 10% strongest connection observed in controls. As a result, a single connection in stroke patients' functional brain network must change less than the 10% strongest connection in a healthy brain network to reject the null hypothesis in (2.35). Considering similar studies, such substantial FC changes in stroke patients might be unusual. Branscheidt et al. [4] collected rs fMRI data from 19 subcortical stroke patients and

11 controls. The subjects finished up to five sessions during a year. In contrast to this study, Branscheidt et al. examined only motor and somatosensory areas of the cortex, known to be relevant for hand movement. Despite substantial behavioural improvement among patients in the first year after stroke, they found no evidence for longitudinal post-stroke cortical connectivity changes. There were no observed changes in FC neither across time nor across the two groups.

We have studied three FC methods: two that are correlation-based and stationary, and one that provides bidirectional edges by accounting for the BOLD signal's time dependency. The latter refers to effective connectivity, a subset of FC. Recognise that the notion of directed effective connectivity does not imply two nodes are causally related [63]. Smith et al. [64] validate and compare different FC methods using the DCM to yield simulated fMRI data, see (2.22). The method's performance is determined by its sensitivity to recognising appropriate network connections. Their findings suggest that correlation-based methods, such as Pearson and Partial correlation, outperform non-stationary methods. Due to mathematical and computational feasibility, Smith et al. did not evaluate the Dynamical causal model. Also, the spectral DCM [56], implemented in this work, was not yet introduced when the research article by Smith et al. was published. We have measured method performance using a trt reliability score. Our findings suggest that the DCM, which can be regarded as a lag-based method, performs no worse than the correlation-based methods. However, unlike Smith and coauthors, we focus on the precision of a method rather than its accuracy. Also, simulated data is not comparable to observed BOLD signals from fMRI sessions owing to simulations' inability to recreate the proper form of noise. Out of the three methods, partial correlation and spDCM yield weights with significantly lower variance between sessions in comparison to the Pearson correlation (see Figures 3.5,3.8 and 3.11). This finding is interesting as the latter method is the one most frequently used in FC analysis. Yet, it is worth remembering that the DCM holds a possible advantage because it produces bidirectional edges.

We have also examined how different parcellation schemes influence the resulting trt reliability of each FC method. There are no fMRI studies have assessed the effect of this variable on the trt reliability, as far as known. We studied seven alternative parcellation schemes, each dividing the human cortex into functionally separate brain areas called nodes. Table 3.1 shows a weak rising trend in the control group mean reliability scores as the number of nodes increases. Furthermore, the between-subject variances of the reliability scores, as shown in Figure 3.13, reduce significantly when the number of nodes increases from tens to hundreds. This finding implies that the results are more consistent for more nodes. As mentioned in Chapter 1, one could argue that the individual neurons or voxels represent single nodes more realistically. However, the large volumes of time series data to be investigated may limit the available connectivity analysis methods. Due to an exponential expansion of free parameters as the number of nodes increases, the spDCM cannot analyse more than the 17 time series. As a result, we incorporated the smaller parcellation schemes in the current study to include this model. Still, one can question whether a network of 8 nodes covering the entire cortex can capture the activity caused by approximately 86 billion neurons.

4.2 Limitations

Regarding anatomy, no brains are the same due to our genetic material and unique life experiences [65]. Furthermore, we know that stroke may alter the brain in various ways. As a result, the recovery process differs greatly amongst patients [3]. Resting-state fMRI data should ideally be interpreted for individual patients when studying FC networks of the human brain in health and disease [66]. Examination of single patients is usually insufficient in fMRI research due to poor trt reliability. Previous studies of functional rewiring post-stroke consider the averaged connectivity patterns within each group: controls and patients [4] [3]. Because the observed variation of identical edges across sessions is considerable, as shown in Figure 3.15, we are obliged to use the mean across patients, despite the drawbacks we have discussed. The independence assumption of $d_i = w_i^{\text{ses1}} - w_i^{\text{ses2}}$ is another weakness with the power analyses described in Section 3.4. We have considered $d_i^{\text{sub}g}$ to be independent of $d_j^{\text{sub}g}$ for the same subject, and $d_i^{\text{sub}g}$ to be independent of $d_i^{\text{sub}k}$ from another subjects. These assumptions are analogous to w_i being independent of w_j within and across subjects. Since most of the nodes in a brain network are connected via edges whose existence is dependent on one another, this is a weak assumption. Despite this, the assumptions were required to evaluate the distribution of individual weight changes among participants, as used in (2.35), with just three controls. Additionally, the lack of stroke data affects the estimated effect size, $\delta(q)$, which is based on the range of magnitude in the weights of a control subject network. However, the lack of enough subject data is probably the major limitation of the conducted power analyses in this thesis.

The methods all perform poorly in terms of the trt reliability scores presented in Table 3.1 for both 1 and 3 months Test-Retest intervals. As previously indicated, certain of the selected FC method‘s and reliability measure‘s assumptions regarding the data utilized are not being met. One such example is the assumption of independence between sample points of the time-dependent BOLD signal, which as addressed in Section 2.2.1 can result in "nonsense correlations" [42]. A presence of autocorrelation in the response signal not only leads to these false-positive connections but may also bias the effect on precision on the estimated connectivity pattern [67]. These miscalculated edges will shift from session to session, possibly resulting in reduced overall consistency of the methods.

It is crucial to keep in mind the low signal-to-noise ratio of rs-fMRI data [68] [66] when addressing the reliability of our methods. The observed BOLD signal contains thermal noise from the scanner and physiological noise from the individual subjects, owing to head movement and respiratory activity. Additionally, physiological effects such as level of stress [69] and recent sleep quality [70] act as possible confounders that may disturb fMRI measurements. Recall that trt reliability quantifies how similar the within-subject FC network is under identical conditions, such as the same scanner, time of day, and analytic procedures. Although we strive to account for both forms of noise during data preprocessing, see Appendix B.1, perfectly identical settings between sessions are hard to achieve. We cannot be sure how these factors affect the measured response signal, which is an indirect measure of brain activity. However, we must consider the likelihood that these confounders lead to biased estimations of functional connectivity [68]. According to a recent study [71], extending the scan durations of rs fMRI scans by up to double the existing BOLD time series can improve reliability. The recording for this study were set to 8 minutes such that the total MR sequence, consisting of additional acquisitions (MRI structural scans, tasks-dependent fMRI and diffusion Imaging) not exceeded an hour.

4.3 Future Work

The only time-dependent among the thee FC methods studied in this work is the spDCM, which was limited to the parcellation schemes of maximum 17 nodes. Therefore, examining another time-varying FC method that manages to evaluate a brain network of hundreds of nodes is of interest. Linearized variants of the Dynamical causal model, such as the more recent regression DCM [72] is one such option. Considering the low variance and high sparsity created by the GLasso estimated precision matrix, another possibility is to evaluate this procedure more closely. The tuning parameter estimation, performed through CV, is not optimized for any of the parcellation schemes, see Figure B.2. Wang et al. state that k-fold CV with the negative log-likelihood as the choice of loss function tends to select overly dense graphs [73]. This finding may suggest that cross validation is not the optimal penalty selection method for our data or that even a regularisation when estimating the inverse covariance matrix is unnecessary for some cases. More efficient approaches exist for tuning parameter selection [73]. In addition, a possible prospect is to extend the model to account for the time dependence of the BOLD signal, using cross-correlation before partial correlating time series of node-pairs.

How to handle negative FC estimations is an important aspect of defining edges. As mentioned earlier, there remains little consensus for handling or interpreting these connections. A fraction of FC studies delete the negative weights prior to analysis [74]. We have examined the resulting trt reliability of all parcellation schemes and methods combinations for all negative weights set to zero. Table B.3, as well as Figures B.3 and B.4 prove that the reliability is slightly improved when negative weights are set to zero. Other connectivity studies have treated the negative weights as positive weights and performed separate analyses on the two resulting connectivity graphs [75]. Prior to the power analyses conducted in Section 3.4, we define d_i as the difference between w_i from sessions 1 and 2, and then average d_i across subjects, see (2.34). It would be advantageous to treat the weights with different signs separately before subtracting weights to prevent signal cancellation owing to opposite-signed weights.

A last prospect of study involves the choice of brain regions. Similar studies have encompassed only motor, and sensory, related to the expected damage post-stroke [4] [3]. This kind of brain parcellation is referred to as region of interest (ROI) analysis, as the nodes correspond to brain regions relevant to what is studied. There are several advantages with ROI-based parcellations. Firstly, the computational burden which confines the spDCM is reduced as there are fewer nodes to consider. Furthermore, the nodes represented by ROIs usually span smaller brain areas than the nodes of a whole-brain parcellation. If only a few functional connections are affected by rewiring following a stroke, we may be more likely to notice them when "zooming" in on relevant brain areas instead of encompassing the entire brain simultaneously. On the other hand, analysis of pairwise interactions between all brain areas concurrently, such as in a whole-brain parcellation, might provide a better understanding of the network dynamics. Nonetheless, this relies on consistent models of the functional brain networks, which, as discussed in this chapter, is challenging to achieve. Therefore, considering areas in the Somamotor-area of the brain while studying motor recovery in stroke patients is an optimistic possibility for further work. Additionally, utilizing task-dependent fMRI where the participants perform motor tasks during the data acquisition might improve the reliability between sessions and among subjects [76].

4.4 Conclusion

The main objective of this thesis was to investigate whether we can detect neuroplasticity after stroke using rs fMRI data. We examined three sessions of rs fMRI scans performed over three months on healthy elderly participants. To be convinced that possible, observed changes in FC after stroke are caused by motor recovery; we analysed the trt reliability of various FC approaches and parcellation schemes. We also conducted power tests to predict how many patients should be recruited for a stroke rehabilitation study at St.Olav Hospital to detect significant changes in the functional brain network post-stroke. Knowing whether or not a research's experimental design should be adjusted to yield useful results might save time and resources. The sample size predictions were based on the observed temporal FC changes across controls. Due to the small sample size, fMRI noise, and violations of model assumptions, the measured brain networks could not provide a reliable conclusion on the exact number of subjects to be recruited. However, the number is likely not feasible for the stroke rehabilitation study – which will not be able to recruit a large number of stroke patients, particularly those who have had similar strokes. Our findings, therefore, contribute valuable insight into prospects for this study. Furthermore, the reliability analysis and sample size calculations were incorporated into a statistical framework, for which one can easily include new methods and data quality. This framework helps to address the issue of reproducibility in neuroimaging studies. Further work is suggested to implement the framework using new FC measures, motor-task fMRI with longer scan durations and more subjects.

Appendix A

Additional statistical theory

A.1 Principal Component Analysis (PCA)

The following section provides details regarding the dimensionality reduction tool Principal component analysis (PCA), following the review article by Jolliffe and Cadima [77]. Consider a data matrix $\mathbf{Y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y}_v) \in \mathbb{R}^{T\times v}$. The columns of $\mathbf{Y}$, each of length T, are centered to have zero mean. The first PC, denoted $\boldsymbol{x}_1$ equals the normalized linear combination of the columns of $\mathbf{Y}$ such that variance is maximised. Define $\boldsymbol{x}_1$ as follows

$$\boldsymbol{x_1} = \phi_{11}\boldsymbol{y_1} + \phi_{21}\boldsymbol{y_2} + \ldots + \phi_{v1}\boldsymbol{y_v} = \sum_{j=1}^{v} \phi_{j1}\boldsymbol{y_j} = \mathbf{Y}\boldsymbol{\phi}_1, \tag{A.1}$$

where $\boldsymbol{\phi}_1 = (\phi_{11}, \phi_{21} \ldots \phi_{v1})^T \in \mathbb{R}^{v\times 1}$ is a vector of constants, called the *loadings* of the first PC. Next, estimate the loading vector such that the variance of $\boldsymbol{x}_1$ is maximised,

$$\max_{\phi_{11},\phi_{21}\cdots\phi_{p1}} \mathrm{Var}(\mathbf{Y}\boldsymbol{\phi}_1) = \max_{\phi_{11},\phi_{21}\cdots\phi_{p1}} \frac{1}{T}\sum_{i=1}^{T}(\boldsymbol{\phi_1}^T\boldsymbol{y}_i - \boldsymbol{\phi}_1^T\bar{\boldsymbol{y}})^2 = \max_{\phi_{11},\phi_{21}\cdots\phi_{p1}} \boldsymbol{\phi}_1^T\hat{\Sigma}\boldsymbol{\phi}_1, \tag{A.2}$$

where $\hat{\Sigma} \in \mathbb{R}^{v\times v}$ resembles the sample covariance matrix of $\mathbf{Y}$. For the problem to have a well-defined solution, impose the following constraint on the loading vectors: $\sum\limits_{j=1}^{v} \phi_{j1}^2 = 1$, or equivalently $\boldsymbol{\phi_1^T}\boldsymbol{\phi_1} = 1$. A Lagrange multiplier, denoted by $\lambda_1 \in \mathbb{R}^1$, is introduced for the maximizaton problem (A.2) to enforce this constraint

$$\max_{\phi_{11},\phi_{21}\cdots\phi_{p1}} \boldsymbol{\phi}_1^T\hat{\Sigma}\boldsymbol{\phi}_1 + \lambda_1(1 - \phi_1^T\phi_1). \tag{A.3}$$

Now, set the derivative of the above expression, with respect to $\boldsymbol{\phi}_1$, equal to zero. Then the maximization problem in (A.3) is equivalent to solving an eigenvalue-decomposition of the covariance matrix defined as follows:

$$\hat{\mathbf{\Sigma}}\boldsymbol{\phi}_1 = \lambda_1\boldsymbol{\phi}_1. \tag{A.4}$$

The above equation concludes that $\boldsymbol{\phi}_1$ is the first eigenvector of $\hat{\Sigma}$. That is the eigenvector with the largest eigenvalue, in this case λ_1. Since the first eigenvector of a covariance matrix yields the direction of the axes where there is most variance, the variance must be maximum when $\boldsymbol{\phi}_1$ equal to the first eigenvector, also known as the first PC [78].

A.2 Generalized Linear Model

A.2.1 Linear Regression Model

Regression aims to explain the effect of an independent variable $\boldsymbol{x}$, called a *covariate*, on a dependent variable of interest $\boldsymbol{y}$, also called the *response* variable [59]. The relationship between the two continuous variables $(y_i, x_i), i = 1, \ldots, n$ can be represented through a simple linear regression model as follows

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, \ldots, n. \tag{A.5}$$

The regression coefficients β_0 and β_1 represent the intercept and slope of a regression line, respectively. These coefficients are estimated from the data $(y_i, x_i), i = 1, \ldots, n$, denoted as $\hat{\beta}_0$, $\hat{\beta}_1$, to yield an estimated linear regression line $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i, i = 1, \ldots, n$. The residual errors $\epsilon_1, \ldots, \epsilon_n$, are assumed independent and identically normally distributed with zero mean and constant variance σ^2 across all errors ϵ_i,

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2). \tag{A.6}$$

It follows that the observations of the response variable y_i follow a normal distribution, conditionally independent on the covariate, with mean $\mu_i = E[y_i \mid x_i] = \beta_0 + \beta_1 x_i, i = 1, \ldots, n$ and variance σ^2,

$$y_i \sim \mathcal{N}(\mu_i, \sigma^2). \tag{A.7}$$

Estimations of the regression coefficients β_0, β_1 in (A.5) is necessary for inferring the regression line $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i, i = 1, \ldots, n$. The estimates of β_0, β_1 are denoted as $\hat{\beta}_0, \hat{\beta}_1$ The simplest approach is called the method of least squares (LS), as it minimizes the sum of the squared deviations as follows

$$\hat{\beta}_{LS} = \min_{\beta} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2. \tag{A.8}$$

Using simple calculus, the estimated slope coefficient becomes

$$\hat{\beta} = \frac{\sum\limits_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum\limits_{i=1}^{n}(x_i - \bar{x})^2}, \tag{A.9}$$

where $\bar{x} = \frac{1}{n}\sum\limits_{i=1}^{n} x_i$ is the mean of the observations.

Now consider p independent continuous variables $\boldsymbol{x_1},\dots,\boldsymbol{x_p}$ each of n observations, and one response variable $\boldsymbol{y} = (y_1,\dots,y_n)^T$. We have a multiple linear regression model,

$$y_i = \boldsymbol{\beta}^T\boldsymbol{x}_i + \epsilon_i, i = 1,\dots,n, \tag{A.10}$$

where $\boldsymbol{x}_i = (1, x_{i1},\dots,x_{ip})$ is the ith observation of each covariate augmented with a 1 to allow for a constant term in the regression equation, and $\boldsymbol{\beta} = (\beta_0,\beta_1,\dots,\beta_p)^T$ the coefficient vector. Now, let $\mathbf{X}$ contain $p+1$ column vectors, the first being a vector of ones and the rest are the covariate vectors $\boldsymbol{x_j} = (x_{1j},\dots,x_{nj})$ for $j = 1,\dots,p$. $\mathbf{X}$ is called the *design matrix*, of dimension $n \times (p+1)$. The regression model in (A.10) can be written in matrix form as follows:

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{A.11}$$

with $\boldsymbol{\epsilon} = (\epsilon_1,\dots,\epsilon_n)^T$, called the error vector.

A.2.2 GLM framework

It is possible to extend the model in (A.11) to allow for non-normal response and error distribution. The response variable $\boldsymbol{y}$ is restricted to be written on the form of an *univariate exponential family*:

$$f(y_i \mid \theta_i) = \exp\Big(\frac{y_i\theta_i - b(\theta_i)}{\phi}\Big)w + c(y_i,\phi,w_i), \tag{A.12}$$

where θ_i is the natural parameter, ϕ the dispersion parameter and w_i the weight parameter. The linear model in equation (A.10) is a special case of Generalized linear model (GLM) where the response function y_i is gaussian and a member of the exponential family with $\phi = \sigma^2$, $b(\theta_i) = \frac{1}{2}\mu_i^2$ and $\theta_i = \mu_i$.

The structural component of a GLM ensures linearity. The response function is allowed to be nonlinear. This is done through a one-to-one and twice differentiable *response function*, h connecting the linear predictor $\eta_i = \boldsymbol{x_i^T\beta}$ with the mean of the response $E(y_i) = \mu_i$,

$$\mu_i = h(\eta_i) = h(\boldsymbol{x_i^T\beta}). \tag{A.13}$$

We often use the inverse of this function called the *link function*,

$$\eta_i = \boldsymbol{x_i^T\beta} = g(\mu_i). \tag{A.14}$$

A.3 Maximum Likelihood Estimation

The method of least square (A.8) yields estimates of the unknown regression coefficients $\boldsymbol{\beta}$ in a linear regression model, such as (A.11). For a GLM the parameters $\boldsymbol{\beta}$ are estimated by maximizing the log-likelihood function of the data. That is, we want to estimate $\hat{\boldsymbol{\beta}}$ such that the probability of observing the data we already have observed is at it‘s most likely [59].

Consider a sample $\boldsymbol{y} = (y_1,\ldots,y_n)$ from a population with probability density function $f(y_i \mid \boldsymbol{\theta})$ belonging to the exponential family. The likelihood function equals the joint density of the observations y_i as a function of a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$

$$L(\boldsymbol{\theta} \mid \boldsymbol{y}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\theta}), \tag{A.15}$$

The above equation (A.15) assumes $y_1,\ldots y_n$ to be to be *independent and identical distributed.* The Maximum Likelihood Estimation (MLE) of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}}_{MLE} = \max_{\theta} L(\boldsymbol{\theta} \mid \boldsymbol{y}). \tag{A.16}$$

Usually, it is more convenient to maximize the log-likelihood function. Since the logarithm is a strictly increasing function, maximizing $\log L(\boldsymbol{\theta} \mid \boldsymbol{y})$ is equivalent to (A.16).

The parameter vector $\boldsymbol{\theta}$ represents the mean of the response μ_i, which through the link function (A.13) is related to the linear predictor η_i. The linear predictor is further a function of the regression parameters $\boldsymbol{\beta}$ through (A.14). Therefore, the likelihood function in (A.15) is a function of $\boldsymbol{\beta}$.

A.3.1 The Fisher Scoring algorithm

The MLE from (A.16) is usually estimated by setting the derivate of $\log L(\boldsymbol{\theta})$, called the *score function*, to zero and solving for $\hat{\boldsymbol{\theta}}$:

$$s(\hat{\boldsymbol{\theta}}) = \frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0}. \tag{A.17}$$

For a nonlinear response function, the problem in (A.17) yields nonlinear equations that cannot be solved analytically. An approximate solution can be found through a first-order Taylor expansion of the score function $s(\hat{\boldsymbol{\theta}})$ around an arbitrary point $\boldsymbol{\theta}_0$

$$s(\hat{\boldsymbol{\theta}}) = s(\boldsymbol{\theta_0}) + \frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta_0}} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta_0}) + \mathcal{O}(\boldsymbol{\theta}^2). \tag{A.18}$$

The negative of the first order derivative of the score function is called the *Observed Fisher information*

,

$$H(\boldsymbol{\theta}) = -\frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial l^2(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta^T}} \tag{A.19}$$

Rearrange the terms in (A.18) to solve for $\boldsymbol{\theta}$, and we obtain an iterative algorithm called *Newton Raphson,*

$$\hat{\boldsymbol{\theta}}^{t+1} = \hat{\boldsymbol{\theta}}^t + H(\hat{\boldsymbol{\theta}}^t)^{-1} s(\hat{\boldsymbol{\theta}}^t) \tag{A.20}$$

The algorithm is constantly finding a new tangent line of the score function evaluated at the current estimate of $\hat{\boldsymbol{\theta}}$ until the line approximately does not change anymore. A similar optimization technique called the *Fisher Scoring* can be derived by replacing the observed Fisher information matrix in (A.20) with the *Expected Fisher information matrix,*

$$\mathcal{I}(\boldsymbol{\theta}) = E\Big(-\frac{\partial l^2(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta^T}}\Big). \tag{A.21}$$

This gives the iterative algorithm called *Fisher Scoring*

$$\hat{\boldsymbol{\theta}}^{t+1} = \hat{\boldsymbol{\theta}}^t + \mathcal{I}(\hat{\boldsymbol{\theta}}^t)^{-1} s(\hat{\boldsymbol{\theta}}^t) \tag{A.22}$$

The Expected Fisher information matrix might be simpler to compute compared to $H(\boldsymbol{\theta})$, and the Fisher Scoring is therefore often preferred [59].

A.3.2 Multivariate Gaussian MLE

Consider the vector $\mathbf{Y} = (y_1, \ldots, y_p)^T$ drawn from a normal distribution $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with probability density function

$$f(\boldsymbol{y}_i \mid \boldsymbol{\mu}_i, \Sigma) = \frac{1}{\sqrt{(2\pi)^p \mid \Sigma \mid}} \exp\Big(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu})\Big), \tag{A.23}$$

where $\boldsymbol{y}_i$ and $\boldsymbol{\mu}$ are vectors in $\mathbb{R}^p$. Insert the probability density function of y_i in the log-likelihood function (A.15),

$$\begin{aligned}
\log L(\boldsymbol{\mu}, \Sigma \mid \boldsymbol{y}) &= -\frac{np}{2}\log 2\pi - \frac{n}{2}\log(\det \Sigma) - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}) \\
&= -\frac{np}{2}\log 2\pi - \frac{n}{2}\log(\det \Sigma) - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T \Sigma^{-1}(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) - n(\bar{\boldsymbol{y}} - \boldsymbol{\mu})^T \Sigma (\bar{\boldsymbol{y}} - \boldsymbol{\mu}) \\
&= -\frac{np}{2}\log 2\pi + \frac{n}{2}\log(\det \Sigma^{-1}) - \frac{1}{2}\operatorname{tr} \underbrace{\sum_{i=1}^{n}(\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T}_{\hat{\Sigma}_{ML}} \Sigma^{-1} - n(\bar{\boldsymbol{y}} - \boldsymbol{\mu})^T \Sigma (\bar{\boldsymbol{y}} - \boldsymbol{\mu})
\end{aligned} \tag{A.24}$$

The above derivation exploits the trace tricks, $\boldsymbol{x}^T \mathbf{A} \boldsymbol{x} = \operatorname{tr}(\boldsymbol{x}^T \mathbf{A} \boldsymbol{x}) = \operatorname{tr}(\boldsymbol{x}\boldsymbol{x}^T \mathbf{A})$. One can easily notice that the third term of the last equation is maximised for $\boldsymbol{\mu} = \bar{\boldsymbol{y}}$, which is the MLE of the mean. The MLE of the covariance matrix is equals the sample covariance matrix [35].

A.4 LASSO Regularization

Consider the linear regression model in (A.11). The regression parameters $\boldsymbol{\beta}$ can be estimated using the method of least squares

$$\min_{\boldsymbol{\beta}} \frac{1}{2n}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}). \tag{A.25}$$

According to the book *Statistical Learning with Sparsity* by Hastie et al. [46], there are two reasons why one should consider the regularization of coefficients $\boldsymbol{\beta}$ in the problem above. The first is an improvement of prediction accuracy by introducing bias and reducing the variance of the predicted values of $\boldsymbol{y}$. The second reason is to increase interpretability by identifying a smaller subset of predictors that exhibit the strongest effects, shrinking the rest of the coefficients to zero. The rest of this section is based on the same book.

Regularization imposes a penalty on the size of the coefficients,

$$\min_{\boldsymbol{\beta}} \frac{1}{2n}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1 \tag{A.26}$$

where λ is a tuning parameter controlling the amount of shrinkage and $\| \|_1$ denotes the ℓ_1 norm given by:

$$\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\boldsymbol{\beta}|^2 . \tag{A.27}$$

The method called lasso shrinks small coefficients to zero, doing model selection on the regression parameters. We can rewrite the lasso problem in the *Lagranian* form

$$\frac{1}{2n}\min_{\boldsymbol{\beta}}\|\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1 . \tag{A.28}$$

The optimization problem in (A.28) is a convex, quadratic program that can be solved by a numerical computational method such as *coordinate descent.*

Consider a response variable $\boldsymbol{y} \in \mathbb{R}^n$ with p covariates $\mathbf{X} = (\boldsymbol{x}_1, \ldots \boldsymbol{x}_p)^T$ and regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$. Coordinate descent solves the lasso problem in (A.28) for the jth regression coefficient β_j with all other coefficients, $\beta_k, k \neq j$, fixed. It cycles through all covariates in a fixed order, that is $j = 1, 2, \ldots, p, 1, 2, \ldots, p, \ldots$, and updates β_j on the jth step. This update is repeated until satisfied convergence. The objective in (A.28) can be written as

$$\frac{1}{2n}\sum_{i=1}^{n}(y_i-\sum_{k\neq j}x_{ik}\beta_k-x_{ij}\beta_j)^2+\lambda\sum_{k\neq j}|\beta_k|+\lambda|\beta_j|, \tag{A.29}$$

assuming that both y_i and x_{ij} have been standardized and the intercept can be omitted. The solution for each β_j, expressed in terms of the partial residuals $r_i^{(j)} = y_i - \sum_{k\neq j} x_{ik}\hat{\beta}_k$ is

$$\begin{aligned}\hat{\beta}_j &= \mathrm{S}_\lambda\Big(\frac{1}{n}\langle \boldsymbol{x}_j, r^{(j)}\rangle\Big) = \mathrm{S}_\lambda\Big(\frac{1}{n}\langle \boldsymbol{x}_j, y_i - \boldsymbol{x}_{-j}\hat{\boldsymbol{\beta}}_{-j}\rangle\Big)\\ S_\lambda(x) &= \mathrm{sign}(x)(|x| - \lambda)_+,\end{aligned} \tag{A.30}$$

$S_\lambda(x)$ is called the soft-thresholding function whose output is zero for $|x| \geq \lambda$ and $(|x| - \lambda)$ multiplied with the sign function of the input x else. The sign function is a piecewise function defined as follows:

$$\mathrm{sign}(x) = \begin{cases} -1 & x < 0, \\ 0 & x = 0, \\ 1 & x > 0. \end{cases} \tag{A.31}$$

The algorithm minimizes (A.29) along each coordinate j at a time, updating the j^{th} coefficient $\hat{\beta}_j$ according to (A.30) keeping the other coefficients, $\hat{\beta}_{-j}$ fixed.

The regularization parameter λ can be selected using k-fold *cross-validation* with the Gaussian log-likelihood in (A.24) as a score function. The observations are divided into k groups of approximately equal size; $k-1$ of them are training sets used to estimate the coefficients, and the last fold is the validation set where (A.24) is validated, inserting the coefficient estimates from the training set. This scheme is repeated k times so that each fold is treated as the validation set. The CV-estimate is computed by averaging over the score functions of each iteration [34],

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \log L(\boldsymbol{\beta}_{\text{train}}, \Sigma \mid \mathbf{X}). \tag{A.32}$$

This procedure is usually run for a list possible choices of λ, and the optimal parameter is that which gives the smallest average test error of (A.32).

A.5 Variational Bayesian inference

A.5.1 Bayesian Inference

The DCM relies on Bayesian statistics, treating the observed data as fixed functions and the parameters as random variables. Consider a sample $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^T$ drawn from a probability distribution $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ with $\boldsymbol{\theta}$ considered the random variable. The Bayesian model consists of [59]:

The prior distribution, $\pi(\boldsymbol{\theta})$ contributes to prior beliefs about the parameter before data is observed.

The posterior distribution, $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ updates the prior after observing the data $\boldsymbol{y}$.

The evidence $p(\boldsymbol{y})$, which is the marginal distribution of the observed data, is independent of any unknown parameters.

The aim of Bayesian inference is estimating the posterior distribution, obtained from *Bayes theorem*:

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{p(\boldsymbol{y})} \tag{A.33}$$

where the marginal distribution $p(\boldsymbol{y})$ can be written as $\int_{\boldsymbol{\theta}} p(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$. Notice that this quantity is constant with respect to θ, and can therefore be treated as a normalizing constant ensuring that $p(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$ integrates up to one. Rewrite Bayes theorem as a proportional relation: $\pi(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$.

The maximum likelihood procedure was described in Appendix A.2 to achieve an estimator for the unknown parameter $\boldsymbol{\beta}$. The same technique can be performed in Bayesian inference,

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{ML} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\Big(p(\boldsymbol{y} \mid \boldsymbol{\theta})\Big) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\Big(\sum_i \log p(y_i \mid \boldsymbol{\theta})\Big).\end{aligned} \tag{A.34}$$

The above equation does not include any prior information when deriving the estimator. An alternative estimator can be found by maximizing the posterior distribution function $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ with respect to $\boldsymbol{\theta}$, leading to the Maximum a Posterior (MAP):

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{MAP} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\Big(\pi(\boldsymbol{\theta} \mid \boldsymbol{y})\Big) &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\Big(p(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})\Big) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\Big(\sum_i \log p(y_i \mid \boldsymbol{\theta}) + \log \pi(\boldsymbol{\theta})\Big).\end{aligned} \tag{A.35}$$

The marginal likelihood distribution $p(\boldsymbol{y})$ from (A.33) is independent of $\boldsymbol{\theta}$, and therefore omitted in the maximization.

A.5.2 The Expectation-Maximization (EM) algorithm

Estimating the posterior distribution can be analytically unfeasible due to the integral in (A.33). The integral can be approximated in two ways: stochastic sampling or variational inference, bypassing the computation using a few assumptions [79]. A generalized version of the Expectation maximization (EM) algorithm, an iterative procedure for inferring point estimators when dealing with latent variables, is an example of the latter approximative method. The rest of this section is based on the book *Pattern Recognition and Machine Learning* by Bishop [78].

Now consider a sample of continuous observed data $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^T$, some unobserved (latent) data $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)^T$ and a vector of unknown parameters $\boldsymbol{\theta}$. The EM algorithm seeks to find the MLE (or MAP) of $\boldsymbol{\theta}$ (A.34) by iteratively maximizing the expectation of the marginal distribution of the observed values, $p(\boldsymbol{y} \mid \boldsymbol{\theta})$, with respect the ith estimate of the conditional distribution of the $\boldsymbol{z}$ given $\boldsymbol{y}$, denoted as $q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)}) = p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\theta}^{(i)})$. We can see that the likelihood function

$$L(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{z}) = \int p(\boldsymbol{y}, \boldsymbol{z}) d\boldsymbol{z} = \int p(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{\theta}) p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\theta}) d\boldsymbol{z}, \tag{A.36}$$

is not possible to solve directly without $\boldsymbol{z}$ being known. Therefore it is necessary to approximate $p(\boldsymbol{z} \mid \boldsymbol{\theta})$ for the ith guess of $\boldsymbol{\theta}$. The *expectation* of (A.36), with respect to the current estimate $q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)})$, of $\boldsymbol{\theta}^{(i)}$ is:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(i)}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\theta}^{(i)})}[\log L(\boldsymbol{\theta} \mid \boldsymbol{y},\boldsymbol{z})] = \int q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)}) \log L(\boldsymbol{\theta} \mid \boldsymbol{y},\boldsymbol{z}) d\boldsymbol{z}. \tag{A.37}$$

Next, update the parameter estimate from $\boldsymbol{\theta}^{(i)}$ to $\boldsymbol{\theta}^{(i+1)}$ through a *maximization* of (A.37). This procedure, called the EM algorithm, is iterated until the parameter converges or $q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)})$ equals the true distribution $p(\boldsymbol{z} \mid \boldsymbol{\theta})$.

In cases where the maximization of (A.37) remains intractable, seek $\boldsymbol{\theta}$ such that a lower bound, $F(q,\boldsymbol{\theta})$ on (A.36) increases. The lower bound of the likelihood function is derived by applying Jensen‘s inequality for concave functions, $\mathbb{E}[f(\boldsymbol{x})] \leq f(\mathbb{E}[\boldsymbol{x}])$, as follows:

$$\begin{aligned}
\log L(\boldsymbol{\theta} \mid \boldsymbol{y},\boldsymbol{z}) &= \log \int p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{\theta}) d\boldsymbol{z} \\
&= \log \int p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{\theta}) \frac{q(\boldsymbol{z} \mid \boldsymbol{\theta})}{q(\boldsymbol{z} \mid \boldsymbol{\theta})} d\boldsymbol{z} \\
&= \log \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\theta})} \left[\frac{p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{\theta})}{q(\boldsymbol{z} \mid \boldsymbol{\theta})} \right] \\
&\geq \underbrace{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\theta})} \left[\log \left(\frac{p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{\theta})}{q(\boldsymbol{z} \mid \boldsymbol{\theta})} \right) \right]}_{-\tilde{F}(q,\boldsymbol{\theta})}.
\end{aligned} \tag{A.38}$$

The function $\tilde{F}(q,\boldsymbol{\theta})$ is called *the variational free energy* from statistical physics and with opposite sign it equals *the Evidence lower bound* [79]. We can now summarize the generalized EM algorithm as a "maximization-maximization" algorithm in 2

Algorithm 2: Maximization-maximization

```
E-step: $q^{(i)} \leftarrow \underset{\boldsymbol{q}}{\operatorname{argmax}}\left(\tilde{F}(q,\boldsymbol{\theta}^{(t)})\right)$
M-step: $\boldsymbol{\theta}^{(i+1)} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\left(\tilde{F}(q^{(i)},\boldsymbol{\theta})\right)$
```

We can in fact show that the M-step of the ordinary and generalized EM algorithm both maximize the same quantity by re-organizing (A.38) as follows

$$\begin{aligned}
\tilde{F}(q,\boldsymbol{\theta}) &= \int q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)}) \log \frac{p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{\theta})}{q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)})} d\boldsymbol{z} \\
&= \int q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)}) \log p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{\theta}) d\boldsymbol{z} - \int q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)}) \log p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{\theta}) d\boldsymbol{z} \\
&= \underbrace{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\theta}^{(i)})}[\log p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{\theta})]}_{Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i)})} - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\theta}^{(i)})}[\log q(\boldsymbol{z} \mid \boldsymbol{\theta}^{(i)})].
\end{aligned} \tag{A.39}$$

The last term above is the negative entropy of the approximate distribution q which is independent of $\boldsymbol{\theta}$. Thus a maximization of $\tilde{F}(q,\boldsymbol{\theta})$ implies a maximization of the expectation of the log likelihood,

$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(i)})$.

A.6 Hypothesis testing

Statistical hypothesis testing is an inference method that determines which of two complementary hypotheses is true based on a sample from a population [80]. When formulating hypotheses, we commonly use a summary statistic such as the population mean. Let μ be the population mean of a sample $x_1, \ldots, x_n$, and specify the hypotheses

$$\begin{aligned} \mathrm{H}_0 &: \mu = 0 \\ \mathrm{H}_1 &: \mu > 0. \end{aligned} \tag{A.40}$$

The *null hypothesis* claims that the population mean is zero, whereas the *alternative hypothesis* states the opposite. The test is one-tailed, which means that we consider only one side of the population distribution.

When evaluating a hypothesis test there are two possible outcomes that can lead to incorrect findings. These are referred to as Type I-and type-II-errors [80],

$$\begin{aligned} P(\text{Type I-error}) &= P(\text{Reject } \mathrm{H}_0 \mid \mathrm{H}_0 \text{ True}) = \alpha \\ P(\text{Type II-error}) &= P(\text{Not reject } \mathrm{H}_0 \mid \mathrm{H}_1 \text{ True}) = \beta. \end{aligned} \tag{A.41}$$

The risk of rejecting the null hypothesis when it is true is represented by α, whereas the risk of not rejecting the null hypothesis when the alternative is true is represented by β. Statistical power refers to the opposite scenario of a Type-II error. When the population mean is different from zero, this is the chance of correctly rejecting H_0.

Assume the population is normally distributed $X \sim \mathcal{N}(\mu, \sigma)$ with unknown variance and define the following test statistic

$$T = \frac{\bar{x} - 0}{\sqrt{\frac{s^2}{n}}} \sim t_{n-1}, \tag{A.42}$$

which is t-distributed with $n-1$ degrees of freedom. The sample mean is the average of all observations, $\bar{x} = \sum_{i=1}^{n} x_i$, while the sample variance is $s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$. Set a decision rule for when to reject the null hypothesis in (A.40). That is, define a critical value c such that if $T \geq c$, the null hypothesis is rejected. The critical value is determined by the chosen significance level defined in (A.41). For the test statistic in (A.42) we reject H_0 if $T \geq t_{\alpha, n-1}$.

A.6.1 Power analysis

Sometimes we wish to know how many observations are required to avoid a Type-I or Type-II error. It is then necessary to define the likelihood for both types of errors, α and β. Additionally, we must select the size of the effect we believe can cause the population mean to differ from zero. To illustrate, let $\bar{x}$ denote the mean weight change during one month of a population receiving a diet. We want to find the required number of people to be included in the sample for us to observe a mean weight increase of δ due to the diet. We can estimate the minimum sample size n necessary to achieve a power of $1-\beta$ with significance level α and effect size δ as follows

$$\begin{aligned} P(\text{Reject H}_0 \mid \bar{x}_i = \delta) &= 1-\beta \\ P\Big(\frac{\mid \bar{x}_i - 0 \mid}{\sqrt{s^2/n}} > t_{\alpha,n-1} \mid \bar{x}_i = \delta\Big) &= 1-\beta \\ P\Big(\mid \bar{x}_i \mid > t_{\alpha,n-1}\sqrt{s^2/n} \mid \bar{x}_i = \delta\Big) &= 1-\beta \\ P\Big(\frac{\mid \bar{x}_i \mid - \delta}{\sqrt{s^2/n}} > \frac{t_{\alpha,n-1}\sqrt{s^2/n} - \delta}{\sqrt{s^2/n}} \mid \bar{x}_i = \delta\Big) &= 1-\beta. \end{aligned} \tag{A.43}$$

In the last equation above, the expression is standardized according to δ. Now, the right side of the inequality above equals the $1-\beta$ quantile of the t-distribution

$$\frac{t_{\alpha,n-1}\sqrt{s^2/n} - \delta}{\sqrt{s^2/n}} = t_{1-\beta,n-1}, \tag{A.44}$$

which can be solved for n as follows

$$n = \frac{s^2}{\delta^2}(t_{\beta,n-1} + t_{\alpha,n-1})^2. \tag{A.45}$$

Appendix B

In depth data processing and analysis

B.1 Preprocessesing details

Preprocessing of the rs fMRI data was performed by my co-supervisor, Riccardo Iandolo, and is described in detail in the following section

Anatomical images:

The 7T MP2RAGE UNI and INV2 images were firstly defaced with the software pydeface [81]. Then, the INV2 image was corrected for intensity non-uniformity using SPM [62]. UNI image was multiplied with a bias-corrected and scaled version of the INV2 as described here [82]. This step is needed to get rid of the noisy salt and pepper pattern in the background of UNI image (this process is also "called" as mpragesing, as described here [83][84]). Finally, after bias field correction, brain extraction has been performed with SPM. The resulting brain mask was adjusted with AFNI 3d mask-tool.

Preprocessing of functional images:

Preprocessing of resting state data has been performed with fMRIPrep version 21.0.2 [85]. The above already skull-stripped UNI structural image has been provided as input (the command has been launched with the option -skull-strip-t1w skip).

A B0-nonuniformity map (or fieldmap) was estimated based on two (or more) echo-planar imaging (EPI) references with 'topup' as part of FSL [86]. For each of the BOLD runs found per subject the following preprocessing was performed. First, a reference volume and its skull-stripped version were generated using a custom methodology of fMRIPrep. Head-motion parameters with respect to the BOLD reference (transformation matrices, and 6 corresponding rotation and translation parameters) are estimated before any spatiotemporal filtering using 'mcflirt'. The estimated fieldmap was then aligned with rigid-registration to the target EPI (echo-planar imaging) reference run. The field coefficients were mapped on to the reference EPI using the transform. BOLD runs were slice-time corrected to 0.742s (0.5 of slice acquisition range 0s-1.49s) using "3dTshift" from AFNI [87]. The BOLD reference was then co-registered to the T1w reference using "mri-coreg" (FreeSurfer) followed by "flirt" with the boundary-based registration cost-function. Co-registration was configured with six degrees of freedom. Several confounding time series were calculated based on the preprocessed BOLD: framewise displacement (FD), DVARS and three region-wise global signals. FD was computed using two formulations following Power (absolute sum of relative motions) and Jenkinson (relative root mean square displacement between affines). FD and DVARS are calculated for each functional run, both using their implementations in Nipype.

The three global signals are extracted within the CSF, the WM, and the whole-brain masks. Additionally, a set of physiological regressors were extracted to allow for component-based noise correction CompCor. Principal components are estimated after high-pass filtering the preprocessed BOLD time-series (using a discrete cosine filter with 128s cut-off) for the two CompCor variants: temporal (tCompCor) and anatomical (aCompCor). CompCor components are then calculated from the top 2% variable voxels within the brain mask. For aCompCor, three probabilistic masks (CSF, WM and combined CSF+WM) are generated in anatomical space. The implementation differs from that of Behzadi et al. [88] in that instead of eroding the masks by 2 pixels on BOLD space, the aCompCor masks are subtracted a mask of pixels that likely contain a volume fraction of GM. This mask is obtained by thresholding the corresponding partial volume map at 0.05, and it ensures components are not extracted from voxels containing a minimal fraction of GM. Finally, these masks are resampled into BOLD space and binarized by thresholding at 0.99 (as in the original implementation). Components are also calculated separately within the WM and CSF masks. For each CompCor decomposition, the k components with the largest singular values are retained, such that the retained components' time series are sufficient to explain 50 percent of variance across the nuisance mask (CSF, WM, combined, or temporal). The remaining components are dropped from consideration.

The head-motion estimates calculated in the correction step were also placed within the corresponding confounds file. The confound time series derived from head motion estimates and global signals were expanded with the inclusion of temporal derivatives and quadratic terms for each. Frames that exceeded a threshold of 0.5 mm FD or 1.5 standardised DVARS were annotated as motion outliers. The time series were resampled into several standard spaces, correspondingly generating the following spatially-normalized, preprocessed BOLD runs: MNI152NLin6Asym, MNI152NLin2009cAsym. One reference volume and its skull-stripped version were generated using a custom methodology of fMRIPrep. Automatic removal of motion artifacts using independent component analysis, ICA-AROMA was performed on the preprocessed BOLD on Montreal Neurological Institute (MNI) space time series after removal of the first three non-steady state volumes and spatial smoothing with an isotropic, Gaussian kernel of 6mm FWHM (full-width half-maximum). "Non-aggresively" denoised runs were produced after such smoothing. Additionally, the "aggressive" noise-regressors were collected and placed in the corresponding confounds file. The ICA was performed as exploratory in order to identify the potential presence of unwanted artifacts. All resamplings can be performed with a single interpolation step by composing all the pertinent transformations (i.e. head-motion transform matrices, susceptibility distortion correction when available, and co-registrations to anatomical and output spaces). Gridded (volumetric) resamplings were performed using "antsApplyTransforms" (ANTs) [89], configured with Lanczos interpolation to minimize the smoothing effects of other kernels. Non-gridded (surface) resamplings were performed using "mri-vol2surf" FreeSurfer.

Other anatomical steps were performed using fMRIPrep. That involves a brain tissue segmentation of cerebrospinal fluid (CSF), white-matter (WM) and gray-matter (GM) performed on the brain-extracted T1w using ‘fast‘. Volume-based spatial normalization to two standard spaces, which are (MNI152NLin6Asym, MNI152NLin2009cAsym), was performed through nonlinear registration with ‘antsRegistration‘, using brain-extracted versions of both T1w reference and the T1w template. The following templates were selected for spatial normalization: FSL’s MNI ICBM 152 non-linear 6th Generation Asymmetric Average Brain Stereotaxic Registration Model, ICBM 152 Nonlinear Asymmetrical template version 2009c*.

Postprocessing of functional images

All postprocessing steps have been performed on functional data registered to MNI152NLin6Asym with 1 mm resolution voxel size. First, the brain mask obtained with fMRIPrep has been applied to the resting state data. Second, the following confounds has been regressed out using the command "fsl-regfilt": CSF, WM, six motion parameters and framewise displacement. Then, the time series were band-pass filtered $[0.01, 0.08]$ Hz using the AFNI command 3dTproject.

B.2 Parcellation details

The parcellation of preprocessed BOLD signals received in NIfTI-1 format was conducted by the author of this master’s thesis. The process, which was briefly discussed in Chapter 2, is described in further detail in the following section.

Following the Kong-Schaefer atlas [32], we have included seven distinct parcellation techniques in

Number of nodes (p)	Description	Combined across hemispheres
8	based on 500-parcels.	Yes
16	based on 500-parcels-	No
17	based on 500-parcels.	Yes
34	based on 500-parcels.	No
100	based on 100-parcels.	No
300	based on 300-parcels.	No
500	based on 500-parcels.	No

Table B.1: This study examined seven different parcellation systems derived from Kong-Schaefer2021 atlas [32]. The leftmost column explains how many nodes are used to cover the entire cortex. The centre column gives a description of which of the parcellation scheme the nodes are extracted from. The left and right hemishere's respective networks can be integrated into a single node based on their functional similarity. The rightmost column shows whether the nodes comprise networks from both hemispheres or from each hemisphere separately.

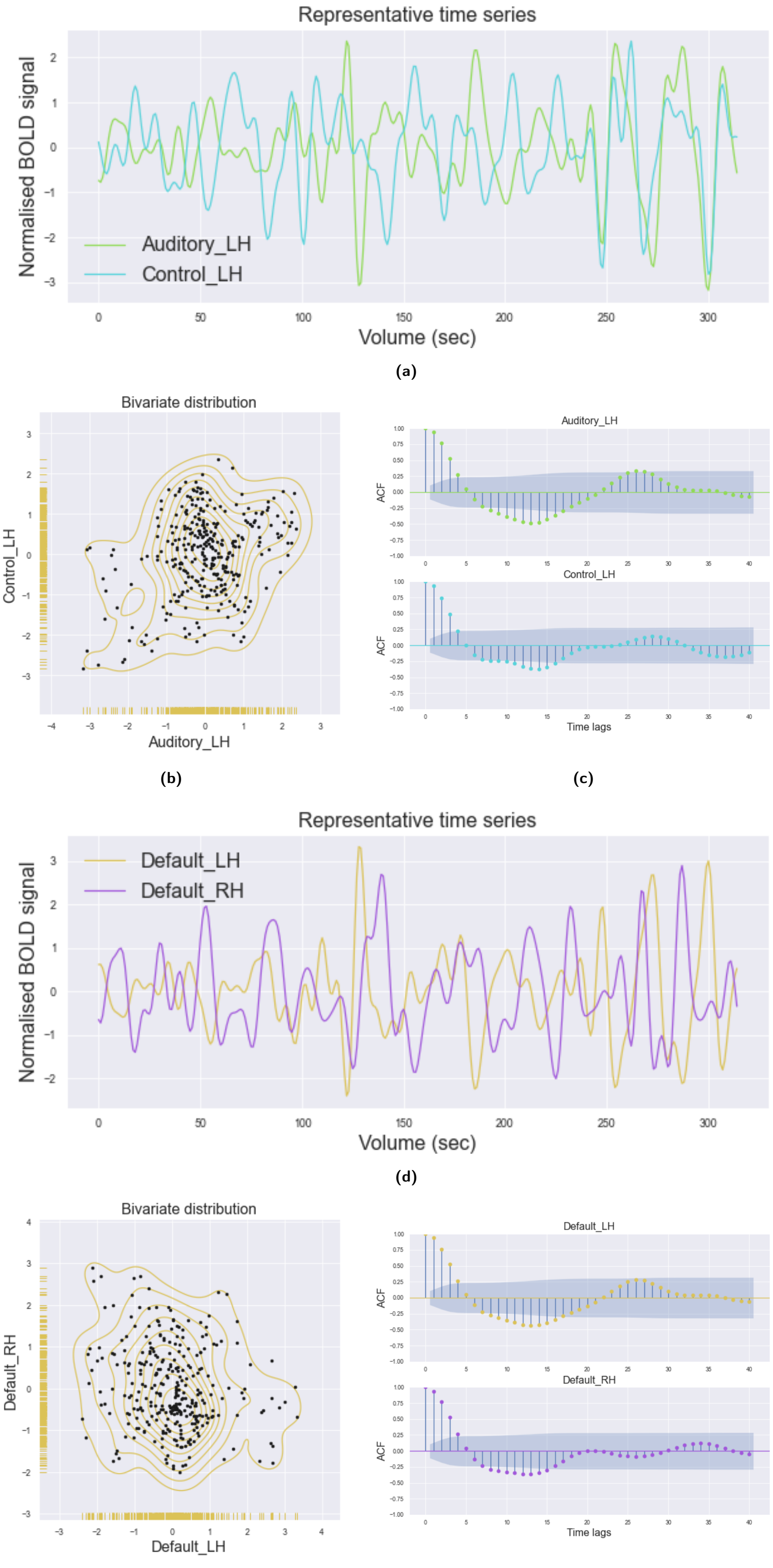
this work. The atlas, based on a volumetric coordinate space, contains instructions for ten parcellations of the full cortex with various degrees of granularity from $p = 100, 200, \ldots 1000$ parcels. The instructions specify which voxels should be grouped to form parcels, each represented by *masks*. The masks are made up of binary numbers where any voxels within the mask are assigned a value one, and any voxels outside the mask are assigned a vlue of zero. Kong et al. also match each of the ten parcellations to Yeo 17 network parcellation [33]. Thus, the parcels can be further clustered into 17 global networks per hemisphere, which implies a total of 34 nodes. Table B.1 gives an overview of all parcellation schemes utilized in this work. All schemes cover the entire cortex.

The parcellation schemes of 500, 300 and 100 nodes correspond to the three original ten schemes by Kong-Schaefer. The Yeo 17 network parcellation identifies the 34 nodes with a base in the 500-parcellation. That is, the 34 masks are created from merging predefined masks of the 500 scheme according to [32]. Some of the 17 networks of each hemisphere are functionally comparable. These networks are aggregated according to Table B.2, yielding 8 networks per hemisphere or 16 nodes. Figure 3.1 provides an overview of the location of the 8 networks in each hemisphere. The 17 nodes are created by combining two equivalent masks from the 17 networks of left and right hemispheres. As a result, the network masks of "LH-DefaultA" in the left hemisphere (LH) and "RH-DefaultA" in the right hemisphere (RH) are concatenated into "DefaultA". Equivalently, the 16 nodes are created by integrating two equivalent masks from the 8 networks of the left and right hemisphere: "LH-Default" and "RH-Default" become "Default". The Bash script in Appendix C.2 show how the masks are created, and BOLD time series are extracted using FSL.

17 Network	**8 Network**
Default A Default B Default C	Default
Language	Language
Control A Control B Control C	Control
SalVenAttn A SalVenAttn B	SalVenAttn
DorsAttn A DorsAttn B	DorsAttn
Auditory	Auditory
Somamotor A Somamotor B	Somamotor
Visual A Visual B Visual C	Visual

Table B.2: The 17 Yeo networks of each hemisphere in left column are combined to yield 8 networks per hemisphere in right column. Networks of similar function, thus similar labels, are merged.

B.3 Additional Results



(a)

(b)

(c)

(d)

(e)

(f)

Figure B.1: Visualisations of the relationship between different pairs of BOLD time series, related to Section 2.2.1. The BOLD signals correspond to the first PCs within two nodes of the $p = 16$ parcellation scheme. The nodes are labeled according to the networks defined in Table B.2 and the associated hemisphere, LH or RH. There are three different visualisations. (a) and (d) give the normalised BOLD time series with $T = 315$ observations, and a sample rate of 1.53s. (b) and (e) show the bivariate distributions (Seaborn, C.1) which are both bell shaped. The amount of contours are analogous to the sampling density. The individual observations are visualised as ticks along the axes. (c) and (f) show the Autocorrelation function with the 40 first lags of each vector. The lags within the shaded region, the 95% confidence interval, are assumed to have no significant correlation with the previous value. There is evidence for significant autocorrelation between the first time points in all variables.

p	Method	r(ses1, ses2)	r(ses1, ses3)
8	Pearson correlation	0.22	0.23
16		0.23	0.31
17		0.24	0.35
34		0.22	0.34
100		0.33	0.31
300		0.32	0.33
500		0.35	0.34
8	Partial correlation	0.28	0.30
16		0.30	0.28
17		0.24	0.29
34		0.33	0.33
100		0.15	0.15
300		0.34	0.33
500		0.36	0.29
8	DCM	0.11	0.07
16		0.17	0.19
17		0.19	0.23

Table B.3: The estimated reliability scores between sessions 1 and 2 (1 month interval) and sessions 1 and 3 (3 month interval), for all methods and parcellations combinations when negative weights are set to zero. The columns are: the number of nodes p, the chosen method, and the estimated average-across subjects reliability scores, r(ses1, ses2) and r(ses1, ses3).

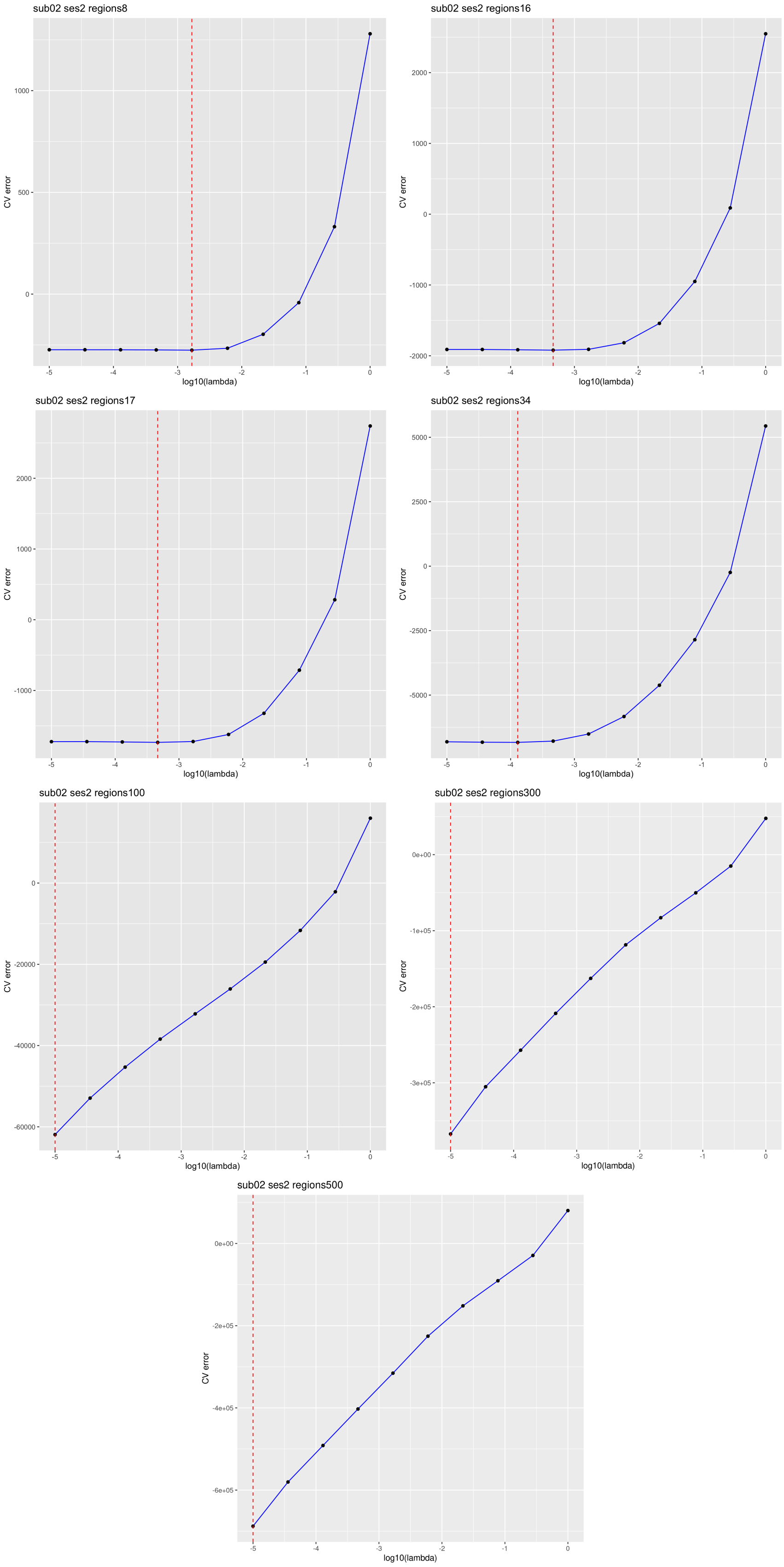
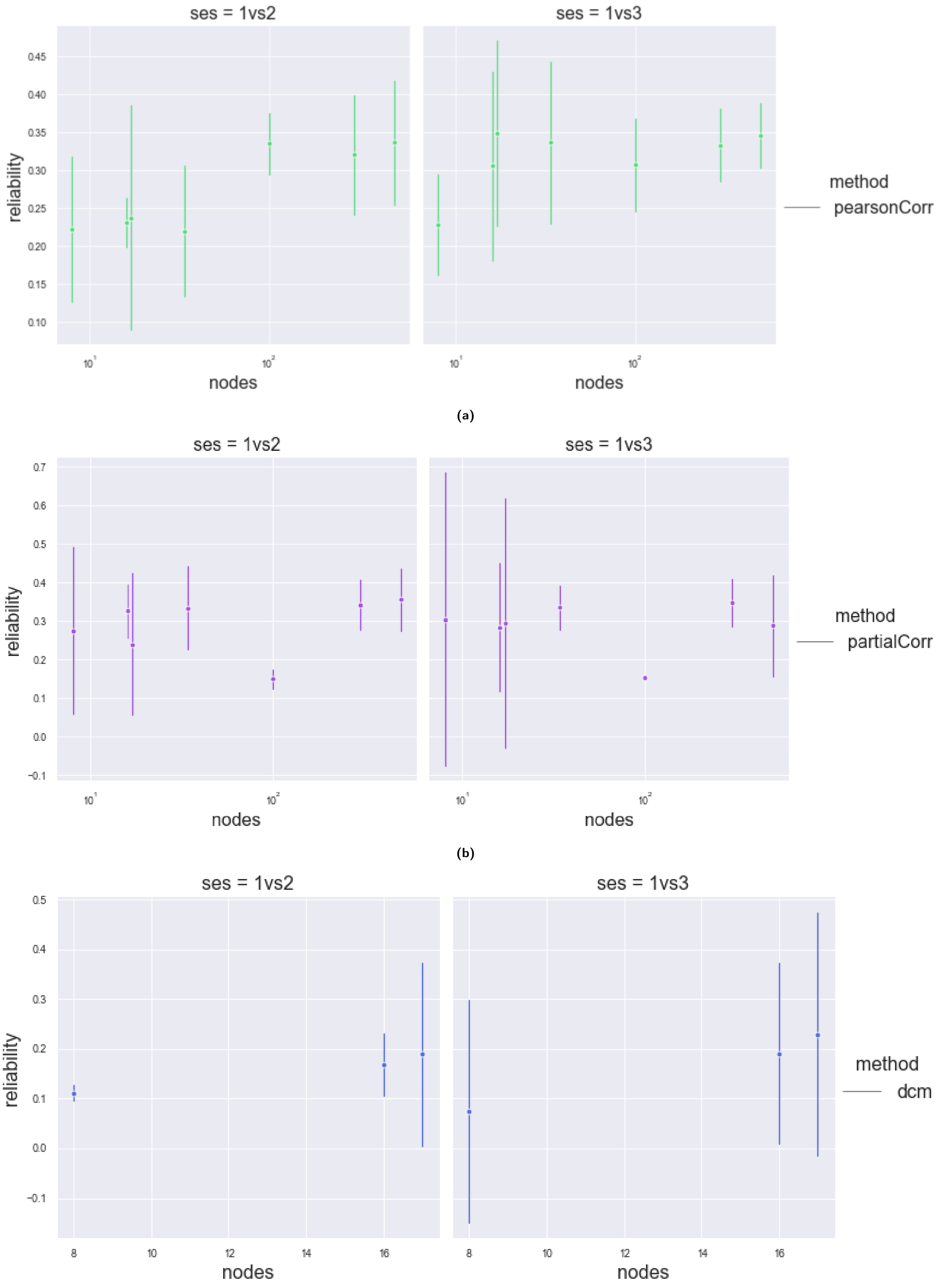


Figure B.2: The cross validation error for different choices of the tuning parameter λ of the GLasso model in (2.13). The optimal λ, which is presented in log10-scale, minimizes the CV error which equals the negative log-likelihood function of Θ in (2.12).The CV procedure is showcased for all parcellation schemes of one subject's session 1. For some of the cases, the selected parameter lays on the boundary. This is due to the scheme not being able to fit (2.12) using smaller values of λ.

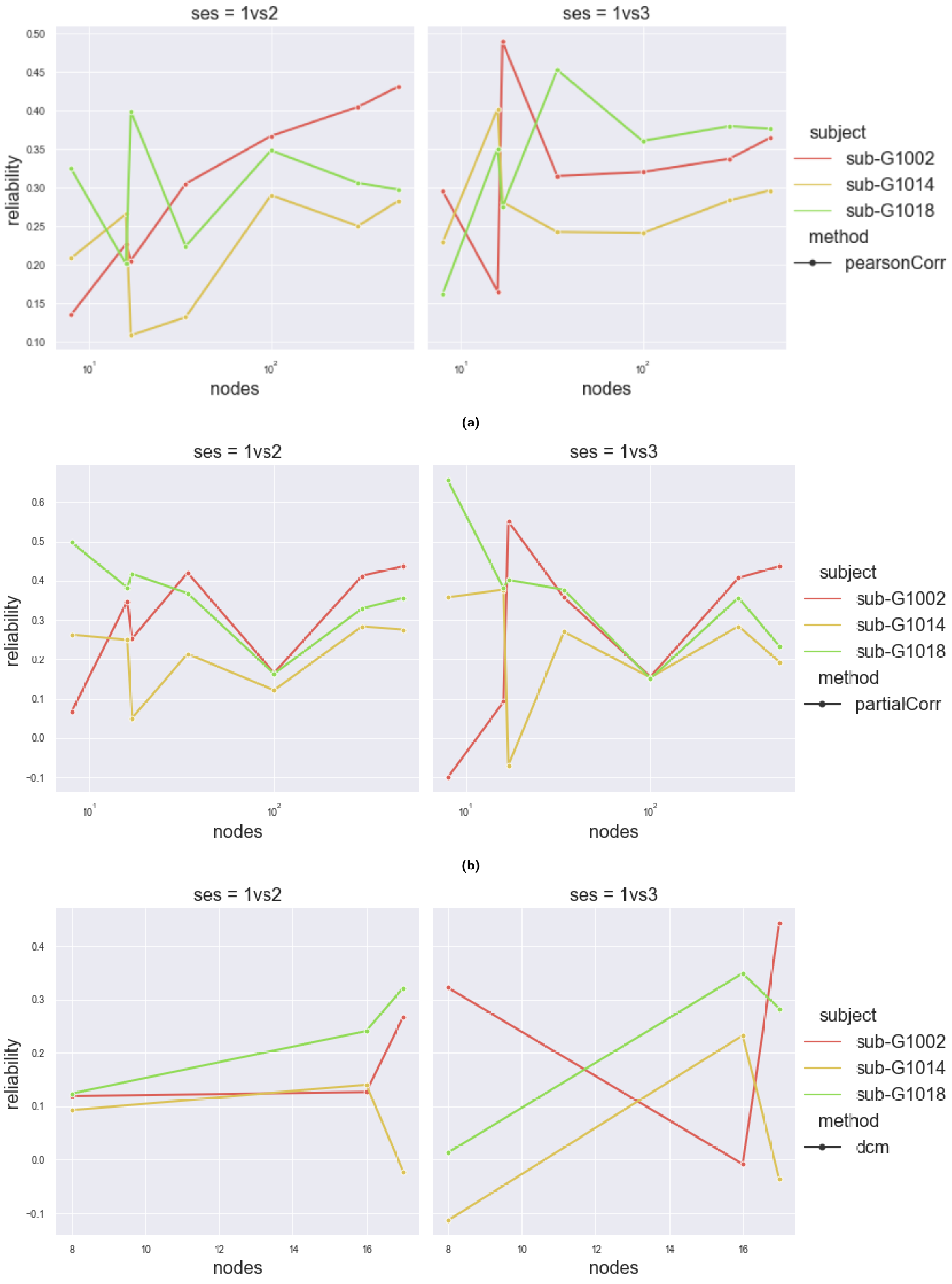


(a)

(b)

(c)

Figure B.3: The mean across subjects Test-Retest (trt) reliability of sessions 1 and 2 (left) and sessions 1 and 3 (right) as a function of different number of nodes when all negative weights are set to zero. The standard deviation across subjects is indicated by an error bar around the mean. Each row corresponds to a separate FC method: Pearson correlation (a), Partial correlation (b), and DCM (c).



(a)

(b)

(c)

Figure B.4: The Test-Retest reliability between sessions 1 and 2 (left) and sessions 1 and 3 (right) as a function of different number of nodes, and for each FC method: Pearson correlation (a), Partial correlation (b), and spDCM (c). The reliability scores from individual control subjects are visualised as separate lines.

Appendix C

Tools and source code

C.1 Tools

C.1.1 FSL (FMRIB Software Library)

The FMRIB Software Library (FSL) is a software library for analysing brain imaging data, such as fMRI data. The program was written mainly by members of the Analysis Group at the Oxford Centre for Functional MRI of the Brain (FMRIB) [86]. For the preprocessed data received for this thesis, FSL was used to create masks according to the Kong-Schaefer atlas and extract .txt-files of time series data. A set of command-line functions from FSLUTILS, allowing analysis of NIfTI-1 files, were applied to our data to perform these procedures. Also, the image viewer, FSLeyes [60], was utilised for the same purpose.

C.1.2 SPM (Statistical Parametric Mapping)

Statistical Parametric Mapping (SPM) refers to the software written by the Welcome Department of Imaging Neuroscience at University College London [62]. The program is a collection of MATLAB functions used to organise and interpret fMRI data. SPM12 is the current version, applied in this thesis to run DCM for cross-spectral densities.

C.1.3 CVGlasso (Lasso Penalized Precision Matrix Estimation)

The R library CVGlasso is a wrapper around the 'glasso' package that Tibshirani wrote, one of the three authors that introduced the Graphical lasso algorithm [47]. The 'glasso' package performs estimation of sparse inverse covariance matrices using lasso penalty. The CVglasso package extends these capabilities by including a cross validation procedure for estimating the hyperparameter [90]. The CV plots in Section 3.2 are plotted using the R package ’ggplot2’.

C.1.4 Nilearn

Nilearn provides statistics for neuroimaging in Python [91]. We have used the library to produce images of the brain network (see Figures 3.4 3.7 3.10).

C.1.5 Seaborn and matplotlib

Seaborn is data visualization library [92] in Python based on matplotlib [93]. All plotted results except the CV plots and the brain network images in Section 3.2 utilize these libraries.

C.1.6 Markov

Markov is a cluster at the Department of Mathematical Sciences at NTNU. The cluster has been extensively used to run the data extraction, the spDCM in Matlab and the GLasso algorithm in R.

C.2 Bash scripts

C.2.1 Creating masks for parcellation

```bash
resolution="1mm"
parcels="500"

# path of output folder
out34="./Masks_${resolution}/Networks34"
out17="./Masks_${resolution}/Networks17"
out16="./Masks_${resolution}/Networks16"
out8="./Masks_${resolution}/Networks8"
out="./Masks_${resolution}/Parcels${parcels}"

# path of masks
out="./Masks_${resolution}/Parcels${parcels}"

# path to atlas
atlas="./Parcellations/Parcellations_Kong2022_17network_order/MNI/Schaefer2018_${parcels}Parcels_Kong2022_17Networks_order_FSLMNI152_${resolution}.nii.gz"

# left hemisphere
lowleft=('1' '17' '37' '54' '66' '82' '97' '107' '122' '136' '155' '167' '182' '198' '213' '228' '246');
```

```bash
upleft=(’16’ ’36’ ’53’ ’65’ ’81’ ’96’ ’106’ ’121’ ’135’ ’154’ ’166’ ’181’ ’197’ ’212’ ’227’ ’245’ ’250’);

# right hemisphere
lowright=(’251’ ’271’ ’282’ ’296’ ’305’ ’319’ ’336’ ’353’ ’370’ ’393’ ’406’ ’421’ ’431’ ’451’ ’463’ ’480’ ’497’);
upright=(’270’ ’281’ ’295’ ’304’ ’318’ ’335’ ’352’ ’369’ ’392’ ’405’ ’420’ ’430’ ’450’ ’462’ ’479’ ’496’ ’500’);

# network names
netnames=(’DefaultA’ ’DefaultB’ ’DefaultC’ ’Language’ ’ContA’ ’ContB’ ’ContC’ ’SalVentAttnA’ ’SalVentAttnB’ ’DorsAttnA’ ’DorsAttnB’ ’Audio’ ’SomMotA’ ’SomMotB’ ’VisA’ ’VisB’ ’VisC’);

# creating masks for 500 parcels, do the same for 100 and 300
for i in {1..500}
do
    # Define masks for each parcel
    fslmaths ${atlas} -thr ${i} -uthr ${i} -bin ${out}/parcel${i}_${resolution}.nii.gz
done

# creating masks for 34 networks
echo 34 networks
for i in {0..16}
do
    echo ${lowleft[$i]} - ${upleft[$i]} ${lowright[$i]} - ${upright[$i]} ${netnames[$i]}
    fslmaths ${atlas} -thr ${lowleft[$i]} -uthr ${upleft[$i]} -bin ${out34}/LH_${netnames[${i}]}_${resolution}
    fslmaths ${atlas} -thr ${lowright[$i]} -uthr ${upright[$i]} -bin ${out34}/RH_${netnames[${i}]}_${resolution}
done


# creating masks for 17 networks using fslmaths -add
echo 17 networks
for names in ${netnames[*]}
do
    fslmaths ${out34}/LH_${names}_${resolution} -add ${out34}/RH_${names}_${resolution} ${out17}/both_${names}_${resolution}
done
```

```

#creating masks for 16 networks using fslmaths -add
echo 16 networks
#left hemisphere
fslmaths ${out34}/LH_DefaultA_${resolution}.nii.gz -add ${out34}/LH_DefaultB_${resolution}.nii.gz -add ${out34}/LH_DefaultC_${resolution}.nii.gz ${out16}/LH_Default_${resolution}.nii.gz
cp ${out34}/LH_Language_${resolution}.nii.gz ${out16}/LH_Language_${resolution}.nii.gz # there is only one language network
fslmaths ${out34}/LH_ContA_${resolution}.nii.gz -add ${out34}/LH_ContB_${resolution}.nii.gz -add ${out34}/LH_ContC_${resolution}.nii.gz ${out16}/LH_Cont_${resolution}.nii.gz
fslmaths ${out34}/LH_SalVentAttnA_${resolution}.nii.gz -add ${out34}/LH_SalVentAttnB_${resolution}.nii.gz ${out16}/LH_SalVentAttn_${resolution}.nii.gz
fslmaths ${out34}/LH_DorsAttnA_${resolution}.nii.gz -add ${out34}/LH_DorsAttnB_${resolution}.nii.gz ${out16}/LH_DorsAttn_${resolution}.nii.gz
cp ${out34}/LH_Audio_${resolution}.nii.gz ${out16}/LH_Audio_${resolution}.nii.gz # there is only one audio network
fslmaths ${out34}/LH_SomMotA_${resolution}.nii.gz -add ${out34}/LH_SomMotB_${resolution}.nii.gz ${out16}/LH_SomMot_${resolution}.nii.gz
fslmaths ${out34}/LH_VisA_${resolution}.nii.gz -add ${out34}/LH_VisB_${resolution}.nii.gz -add ${out34}/LH_VisC_${resolution}.nii.gz ${out16}/LH_Vis_${resolution}.nii.gz
#right hemisphere
fslmaths ${out34}/RH_DefaultA_${resolution}.nii.gz -add ${out34}/RH_DefaultB_${resolution}.nii.gz -add ${out34}/RH_DefaultC_${resolution}.nii.gz ${out16}/RH_Default_${resolution}.nii.gz
cp ${out34}/RH_Language_${resolution}.nii.gz ${out16}/RH_Language_${resolution}.nii.gz # there is only one language network
fslmaths ${out34}/RH_ContA_${resolution}.nii.gz -add ${out34}/RH_ContB_${resolution}.nii.gz -add ${out34}/RH_ContC_${resolution}.nii.gz ${out16}/RH_Cont_${resolution}.nii.gz
fslmaths ${out34}/RH_SalVentAttnA_${resolution}.nii.gz -add ${out34}/RH_SalVentAttnB_${resolution}.nii.gz ${out16}/RH_SalVentAttn_${resolution}.nii.gz
fslmaths ${out34}/RH_DorsAttnA_${resolution}.nii.gz -add ${out34}/RH_DorsAttnB_${resolution}.nii.gz ${out16}/RH_DorsAttn_${resolution}.nii.gz
cp ${out34}/RH_Audio_${resolution}.nii.gz ${out16}/RH_Audio_${resolution}.nii.gz # there is only one audio network
fslmaths ${out34}/RH_SomMotA_${resolution}.nii.gz -add ${out34}/RH_SomMotB_${resolution}.nii.gz ${out16}/RH_SomMot_${resolution}.nii.gz
fslmaths ${out34}/RH_VisA_${resolution}.nii.gz -add ${out34}/RH_VisB_${resolution}.nii.gz -add ${out34}/RH_VisC_${resolution}.nii.gz ${out16}/RH_Vis_${resolution}.nii.gz


# creating masks for 8 networks using fslmaths -add
echo 8 networks
fslmaths ${out17}/both_defaultA_${resolution}.nii.gz -add ${out17}/both_defaultB_${resolution}.nii.gz -add ${out17}/both_defaultC_${resolution}.nii.gz ${out8}/DefaultTot_${resolution}.nii.gz
cp ${out17}/both_language_${resolution}.nii.gz ${out8}/LanguageTot_${resolution}.nii.gz # there is only one language network
fslmaths ${out17}/both_ContA_${resolution}.nii.gz -add ${out17}/both_ContB_${resolution}.nii.gz -add ${out17}/both_ContC_${resolution}.nii.gz ${out8}/ContTot_${resolution}.nii.gz
fslmaths ${out17}/both_SalVentAttnA_${resolution}.nii.gz -add ${out17}/both_SalVentAttnB_${resolution}.nii.gz ${out8}/SalVentAttnTot_${resolution}.nii.gz
fslmaths ${out17}/both_DorsAttnA_${resolution}.nii.gz -add ${out17}/both_DorsAttnB_${resolution}.nii.gz ${out8}/DorsAttnTot_${resolution}.nii.gz
cp ${out17}/both_Audio_${resolution}.nii.gz ${out8}/AudioTot_${resolution}.nii.gz # there is only one audio network
fslmaths ${out17}/both_SomMotA_${resolution}.nii.gz -add ${out17}/both_SomMotB_${resolution}.nii.gz ${out8}/SomMotTot_${resolution}.nii.gz
fslmaths ${out17}/both_VisA_${resolution}.nii.gz -add ${out17}/both_VisB_${resolution}.nii.gz -add ${out17}/both_VisC_${resolution}.nii.gz ${out8}/VisTot_${resolution}.nii.gz
```

List of Code C.1: Bash script to create the masks for all parcellations in Table B.1, using FSL commands.

C.2.2 Extracting time series data from masks

```
resolution="1mm"
parcels="500"

# path for txt files
txt="./txt_files_${resolution}"

#path to input data
input="./subjects/${subject}/${subject}_rest_1mm_denoised_filteredBP_nodemeaned.nii.gz"

```

```
# path of masks
out="./Masks_${resolution}/Parcels${parcels}"

# all subject-session scans to extract data from
subjects=(sub-G1002s1 sub-G1002s2 sub-G1002s3 sub-G1014s1 sub-G1014s2 sub-G1014s3 sub-G1018s1 sub-G1018s2)

# extract txt-file with 1PC from all subjects with more than one session
for subject in ${subjects[@]}
do

for i in {1..500}
do
    fslmeants -i ${input}  -m ${out}/parcel${i}_${resolution}.nii.gz --eig --transpose >> ${txt}/${subject}_regions${parcels}.txt
    fslstats ${out}/parcel${i}_${resolution}.nii.gz -c >> file_name
done
done
```

List of Code C.2: Bash script used to extract time series from the preprocessed NIfTI-1-files of all available sessions of the subjects in Table 2.1 using the masks from the script above. The above script is an example of the extraction of time series for the $p = 500$ parcellation.

C.2.3 Extracting coordinates

```
resolution="1mm"
nets="16"

# path for txt files
txt="../Coordinates_${resolution}"

# path to mask
mask="../Masks_${resolution}/Networks${nets}"

netnames=(’LH_Default’ ’LH_Language’ ’LH_Cont’ ’LH_SalVentAttn’ ’LH_DorsAttn’ ’LH_Audio’ ’LH_SomMot’ ’LH_Vis’ ’RH_Default’ ’RH_Language’ ’RH_Cont’ ’RH_SalVentAttn’ ’RH_DorsAttn’ ’RH_Audio’ ’RH_SomMot’ ’RH_Vis’);

#loop through all networks
for names in ${netnames[*]}
  do
    fslstats ${mask}/${names}_${resolution}.nii.gz -c >> ${txt}/coordinates_regions${nets}.txt
```

```
 done
```

List of Code C.3: Bash script to extract centre of gravity coordinates from the $p = 16$ masks/nodes. The coordinates are defined in the MNI space in mm.

C.3 R code

```

library(R.matlab)
library(CVglasso)
library(ggplot2)

subs = c("02", "14", "18")
sess = c("1", "2", "3")
regs <- c("8", "16", "17", "34", "100", "300", "100", "500")

for (sub in subs){
for (ses in sess){
for (reg in regs){

input <- sprintf("timeseries/sub-G10%ss%s_regions%s.mat",sub,ses,reg)
data <- readMat(input)

t = 1e-3
lambdas <- exp(seq(log(1e-5), log(1), length = 10))

g <- CVglasso(data$Y,K=5,tol=t, lam=lambdas, cores=2)

#CV plot
ggplot( mapping=aes(x=log10(g$Lambdas),y=g$AVG.error)) +
geom_line(color="blue") +
geom_point() +
geom_vline(xintercept=g$Tuning[1], linetype="dashed", color = "red")+xlab("log10(lambda)") +
ylab("CV error") +
ggtitle(sprintf("sub%s ses%s regions%s",sub,ses,reg))

#save files
path.out <- sprintf("glasso_results/sub-G10%s",sub)
filename.txt <- sprintf("glasso_sub-G10%ss%s_regions%s.txt",sub,ses,reg)
filename.png <- sprintf("cv_sub-G10%ss%s_regions%s.png",sub,ses,reg)
ggsave(file.path(path.out,filename.png))
write.table(g$Omega, file=file.path(path.out,filename.txt), append = FALSE, sep = " ", dec = ".", row.names = FALSE, col.names = FALSE)
}}
```

List of Code C.4: Implementation of the GLasso algorithm from CVglasso in R. The methods returns precision matrix which is imported into the Python script in C.5.

C.4 MATLAB code

```
clear DCM

subs = ["02", "14", "18"]
sess = ["1", "2", "3"]
regs <- [8,16,17]

for sub in subs
  for ses in sess:
    for reg in regs:

      name = sprintf("sub-G10%ss%s_regions%d.mat",sub,ses,reg)
      input = fullfile("../mat-files/",name)
      output = fullfile("dcm_results", "DCM_"+name)

      ROI = load(input);
      DCM.Y.y = ROI.Y; % BOLD time series for n ROIs
      DCM.Y.dt = 1.53;  % Interscan interval, TR
      DCM.n = reg;     % Number of nodes
      DCM.v = 315;    % Number of time points

      % specify priors
      DCM.a = ones(DCM.n,DCM.n);
      DCM.b = zeros(DCM.n,DCM.n,0);
      DCM.c = zeros(DCM.n,0);
      DCM.d = zeros(DCM.n,DCM.n,0);
      DCM.U.u = zeros(DCM.v,1);

      DCM = spm_dcm_fmri_csd(DCM);
      save(output)
```

List of Code C.5: The spDCM is implemented in SPM12, in MATLAB. The intrinsic connectivity matrix $\mathbf{A}$ is returned and imported into the Python script in C.5.

C.5 Python code

```python
import numpy as np
import scipy.stats as sc
import math
from numpy import matlib
from scipy import io
import pandas as pd
import glob
import re
from itertools import groupby
from itertools import product

# %%%%%%%%%%%%%%%%%%%%%%%% Methods %%%%%%%%%%%%%%%%%%%%%%%%


#Remove diagonal of connectivity matrix
def removeDiag(mat):
    '''
    given a matrix (FC), remove diagonal elements and
    return matrix with Nan in diagonal.
    '''
    mask = np.zeros_like(mat, dtype=bool) # nullmatrise med samme shape som mat
    mask[np.diag_indices_from(mat)] = True
    mat[mask] = np.nan
    mat_pos = np.where(mat<0, 0, mat)
    return mat

def partialCorr(P):
    nvars = np.shape(P)[0]
    #square root of the diagonal elements to be used in the denominator
    mydiag = np.sqrt(np.abs(np.diag(P)))
    parCorr = (-P/np.matlib.repmat(mydiag,nvars,1).T)/np.matlib.repmat(mydiag,nvars,1)+2*np.eye(nvars)
    return parCorr


#Pairwise correlation
def pearsonCorr(data):
    '''
    given a nxp matrix with p time series of length n, return the
    pearson correlation matrix and remove diagnoal elements
    '''
    corr = removeDiag(np.corrcoef(data,rowvar=False))
    # corr_pos = np.where(corr>0,corr,0) #remove negative weights
    return corr

#Vectorize symmetric connectivity matrix
def vectorize(mat, symmetric):
    '''
    given a weight matrix, return upper triangular
    elements as an array if symmetric. If assymetric only ignore diagonal elements.
    '''
    if symmetric:
        # zero out everything above and including the diagonal
        tri = np.tril(mat,k=-1) #-1 is below main diagonal
        vec = []
        for row in range(1, tri.shape[0]):
            for col in range(row):
                vec.append(tri[row, col])
        output = np.asarray(vec)
    else:
        vect = np.ndarray.flatten(mat)
        output = vect[~np.isnan(vect)]

    return output

# number of nodes to index in Regions
def nodesToIndex(nodes):
    '''
    Give the number of nodes, return the correct index for the Region class
    '''
    if nodes == 8:
        return 0
    if nodes == 16:
        return 1
    if nodes == 17:
        return 2
    if nodes == 34:
        return 3
    if nodes == 100:
        return 4
    if nodes == 300:
        return 5
    if nodes == 500:
```

```python
        return 6

def powerAnalysis(alpha_orig, beta, quantile, df):
    '''
    Given significance level (alpha), 1-power (beta),
    quantile of weights to decide effect size (q),
    and dataframe from prePowerAnalysis. Returns sample size
    '''
    # assume edges are iid and subjects are iid
    sigma = np.std(df['d'])

    # find delta: quantile of all session1 weights
    q = quantile
    w_sorted = np.sort(df['w1'])
    delta = w_sorted[int(q*np.shape(w_sorted)[0])]

    # tests = nr. of edges
    nr_tests = np.shape(df['d']/3)[0] #divide by 3 subjects

    # bon ferroni correction
    alpha = alpha_orig/nr_tests

    # initial sample size estimate
    z_alpha = sc.norm.ppf(alpha) #one sided
    z_beta = sc.norm.ppf(beta)
    n0 = (z_alpha + z_beta)**2*(sigma/delta)**2

    #iteratively estimate sample size
    nprev = 0
    n = n0
    while abs(n-nprev) > 0.01:
        nprev = n
        t_alpha = sc.t.ppf(q=alpha, df = nprev-1)
        t_beta = sc.t.ppf(q=beta, df= nprev-1)
        n = (t_alpha + t_beta)**2*(sigma/delta)**2

    # if n is inf
    if np.isinf(n):
        n = np.nan
    else:
        n = int(math.ceil(n))

    return n, round(sigma, 2), round(delta, 2)


# %%%%%%%%%%%%%%%%%%%%%%%% Define classes %%%%%%%%%%%%%%%%%%%%%%%%

class Subject:
  def __init__(self, sub_nr, ses1, ses2, ses3=None):
    self.sub_nr = sub_nr
    self.ses1 = ses1
    self.ses2 = ses2
    self.ses3 = ses3

  # assume same number of regions in both sessions
  def reliabilityAnalysis(self):
    '''
    Computes a reliability score using Pearson Correlation
    for session1 vs session 2 and session 2 vs session 3 over all regions for both FC-methods.
    '''
    # list of dictionaries containing: nodes (int), reliability (float), method (string), ses (string)
    data = []

    # loop through all regions, i is regional index
    for i in range(len(self.ses1.getRegions())):
      num_nodes = self.ses1.getRegions()[i].num_nodes

      # FC MEASURE 1
      data.append({"nodes" : num_nodes,"reliability" : np.corrcoef(self.ses1.getRegions()[i].pearsoncorr_vect,self.ses2.getRegions()[i].pearsoncorr_vect)[0,1], "method" : "pearsonCorr",  "ses" : "1vs2" })
      data.append({"nodes" : num_nodes,"reliability" : np.corrcoef(self.ses1.getRegions()[i].pearsoncorr_vect,self.ses3.getRegions()[i].pearsoncorr_vect)[0,1], "method" : "pearsonCorr",  "ses" : "1vs3" })

      # FC MEASURE 2
      data.append({"nodes" : num_nodes,"reliability" : np.corrcoef(self.ses1.getRegions()[i].partialcorr_vect,self.ses2.getRegions()[i].partialcorr_vect)[0,1],"method" : "partialCorr", "ses":"1vs2" })
      data.append({"nodes" : num_nodes,"reliability" : np.corrcoef(self.ses1.getRegions()[i].partialcorr_vect,self.ses3.getRegions()[i].partialcorr_vect)[0,1],"method" : "partialCorr", "ses":"1vs3" })

      #FC MEASURE 3
      if self.ses1.getRegions()[i].num_nodes not in [500,300,100,34]:
        data.append({"nodes" : num_nodes, "reliability" : np.corrcoef(self.ses1.getRegions()[i].dcm_vect,self.ses2.getRegions()[i].dcm_vect)[0,1], "method" : "dcm", "ses": "1vs2" })
        data.append({"nodes" : num_nodes, "reliability" : np.corrcoef(self.ses1.getRegions()[i].dcm_vect,self.ses3.getRegions()[i].dcm_vect)[0,1], "method" : "dcm", "ses": "1vs3" })

    # make dataframe of numpy list
    self.reliability = pd.DataFrame(data)

  def getReliability(self):
    return self.reliability


class Session:
  def __init__(self,ses_nr, sub_nr, regions=None):
    self.ses_nr = ses_nr
    self.sub_nr = sub_nr
    self.regions = regions

  def getRegions(self):
    return self.regions

  # run pearsonCorr for all regions
  def runPearsonCorr(self):
    for region in self.regions:
      #print("Run measures for region ", region.num_nodes)
      region.updatePearsonCorr()

  # get path to partialCorr for all regions
  def runPartialCorr(self):
    for region in self.regions:
      reg = region.num_nodes
      ses = self.ses_nr
      sub = self.sub_nr
      if sub==2:
        sub= "02"
      path = f"glasso_results/sub-G10{sub}/glasso_sub-G10{sub}s{ses}_regions{reg}.txt"
      region.updatePartialCorr(path)

  # get path to DCM for p=8,16,17
  def runDCM(self):
    for region in self.regions:
      reg = region.num_nodes
      ses = self.ses_nr
      sub = self.sub_nr
      if reg not in [500,300,100,34]:
        if sub==2:
          sub= "02"
        path = f"DCM_results/sub-G10{sub}/DCM_sub-G10{sub}s{ses}_regions{reg}.mat"
        region.updateDCM(path)

  # save .mat file of time series data, to be used for input in R and matlab
  def save_mat(self):
    for region in self.regions:
      reg = region.num_nodes
      ses = self.ses_nr
      sub = self.sub_nr
      if sub==2:
        sub= "02"
      path = f"timeseries/sub-G10{sub}s{ses}_regions{reg}.mat"
      io.savemat(path, {"Y" : region.data.tolist()})


class Region:
  def __init__(self,path,num_nodes):
    self.path = path
    self.num_nodes = num_nodes
    self.extractData()

    self.pearsoncorr = None
    self.pearsoncorr_vect = None
    self.partialcorr = None
    self.partialcorr_vect = None
    self.dcm = None
    self.dcm_vect = None


  # Extract data from txt-files for each Region
  def extractData(self):
    self.data = extractData(self.path,self.num_nodes)

  def updatePearsonCorr(self):
    if self.pearsoncorr is None:
      self.pearsoncorr = pearsonCorr(self.data)
      self.pearsoncorr_vect = vectorize(self.pearsoncorr,True)

  def updatePartialCorr(self, path):
    if self.partialcorr is None:
      self.partialcorr = extractGlasso(path,self.num_nodes)
      self.partialcorr_vect = vectorize(self.partialcorr,True)

  def updateDCM(self, path):
    if self.dcm is None:
      self.dcm = extractPosterior(path)
      self.dcm_vect = vectorize(self.dcm,False)


# %%%%%%%%%%%%%%%%%%% Functions outside the classes %%%%%%%%%%%%%%%%%%%

# help function for subjectFactory
def key_funcs(group,dir):
  return lambda x: re.search(dir+'/sub-G10(.*)s(.*)_regions(.*).txt',x).group(group)


# Construct the objects with subjects, sessions and regions
def subjectFactory(dir):
  '''
  given a path to directory containing txt-files on the form: sub-G1002s1_regions300.txt,
  return a list of initialized Subject objects
  '''
  pathList = glob.glob(dir+"/sub-G10*.txt")

  # initialize key functions for each category
  subject_func = key_funcs(1,dir)
  session_func = key_funcs(2,dir)
  region_func = key_funcs(3,dir)
  subjects = []

```

```python
  # iterate through each region for each session for each subject
 for i, sub in groupby(sorted(pathList, key = subject_func),subject_func):
  sessions = [None, None, None]
  for j, ses in groupby(sorted(sub, key = session_func),session_func):
   regions = [None, None, None, None, None, None, None]
   for region in list(ses):
    k = region_func(region)
    if k =="8":
     regions[0] = Region(region,int(k))
    if k =="16":
     regions[1] = Region(region,int(k))
    if k =="17":
     regions[2] = Region(region,int(k))
    if k =="34":
     regions[3] = Region(region,int(k))
    if k == "100":
     regions[4] = Region(region,int(k))
    if k == "300":
     regions[5] = Region(region,int(k))
    if k == "500":
     regions[6] = Region(region,int(k))
   if j == "1":
    sessions[0] = Session(int(j),int(i),regions)
   if j == "2":
    sessions[1] = Session(int(j),int(i),regions)
   if j == "3":
    sessions[2] = Session(int(j),int(i),regions)
  subjects.append(Subject(int(i),*sessions))

 return subjects


# extract parcelData from txt.file
def extractData(path,regions,timepoints=315):
 '''
 given path to txt.file of subject data, number of time points and number of regions
 return an np.array of the time series as column vectors
 '''
 all_parcels = np.zeros((timepoints,regions))
 col = 0

```

```
  with open(path) as f:
    for line in f:
      parcel_data = line.split() # now every 2. list is []
      if parcel_data != []:
        better_data = [float(num) for num in parcel_data]
        all_parcels[:,col] = np.transpose(better_data)
        col +=1
  return all_parcels

# extract data from .mat file
def extractPosterior(path):
  '''
  given a path to .mat file of intrinsic coupling matrix A,
  return a numpy array
  '''
  data = io.loadmat(path)
  DCM = data['DCM']
  Ep = DCM['Ep']
  Ep = Ep[0]
  Ep = Ep[0]
  Ep = Ep[0]
  Ep = Ep[0]
  Ep = Ep[0]
  return removeDiag(Ep)

# extract data from .mat file
def extractGlasso(path,regions):
  '''
  given a path to .mat file of connectivity weights from precision matrix of size pxp,
  return a numpy array and remove diagonal elements
  '''
  i = 0
  mat = np.zeros((regions,regions))
  with open(path) as f:
    for line in f:
      row = line.split()
      new_row = []
      for n in row: #add each elem in row to a vector
        new_row.append(float(n))
      mat[i,:] = new_row
      i+=1
  return removeDiag(partialCorr(mat))


# get coordinates from txt.files from FSL
def getCoords(file):
  '''
  given a path to txt-file containing Kong-Schaefer coordinates for a given parcellation,
  return a numpy array of coordinates
  '''
  allcords = []
  #read all line
  with open(file) as f:
      lines = f.readlines()
      for line in lines:
        line = line.split()
        r = float(line[0])
        a = float(line[1])
        s = float(line[2])
        coord = [r,a,s]
        allcords.append(coord)
  return np.array(allcords)

# prepare power analysis
def prePowerAnalysis(method, nodes, sublist):
    '''
    specify method, number of nodes and list of all subject objects, return a dataframe of
    weights and differences for all subjects
    '''
    data  = []
    n = nodesToIndex(nodes)

    #loop through all subjects
    for s in sublist:
        snr = "02" if s.sub_nr == 2 else str(s.sub_nr)

        if method == "pearsonCorr":
            w1, w2,w3 = s.ses1.regions[n].pearsoncorr_vect,s.ses2.regions[n].pearsoncorr_vect,s.ses3.regions[n].pearsoncorr_vect

        if method == "partialCorr":
            w1,w2,w3 =  s.ses1.regions[n].partialcorr_vect, s.ses2.regions[n].partialcorr_vect,s.ses3.regions[n].partialcorr_vect

        if method == "DCM":
            w1,w2,w3 = s.ses1.regions[n].dcm_vect, s.ses2.regions[n].dcm_vect,s.ses3.regions[n].dcm_vect

        nlist = np.arange(np.shape(w1)[0])
        data.append({'subject' : "sub-G10"+snr, 'node-pair' : nlist ,  "w1" : w1,"w2" : w2, "w3" : w3, "d" : np.subtract(w1,w2)})
        # d refers to the distance between ses1 and ses2



    df = pd.DataFrame(data)
    df = df.explode(['node-pair','w1','w2','w3','d'], ignore_index=True)
    return df

# run power analysis
def runPowerAnalysis(alphas, betas, quantiles, sublist, all_nodes):
    '''
    specify parameters for conducting power tests, a list all subject objects and a list of all parcellation schemes,
    return a dataframe of estimated sample size for all combinations
    '''
    sample_sizes = []
    all_methods = ("pearsonCorr", "partialCorr", "DCM")
    for it in list(product(alphas,betas,quantiles, all_methods, all_nodes)):
        alpha, beta, q, method, nodes = it[0], it[1], it[2], it[3], it[4]
        if not (method == "DCM" and nodes > 17):
            df = prePowerAnalysis(method, nodes, sublist)
            n, sigma, delta = powerAnalysis(alpha, beta, q, df)
            sample_sizes.append({"nodes" : nodes, "sample" : n, "method" : method, "alpha":alpha, "beta":beta, "quantile":q, "effectsize" : delta, "std" : sigma })

    df_sample_sizes = pd.DataFrame(sample_sizes)
    df_sample_sizes['log(sample)'] = np.log10(df_sample_sizes['sample'])
    return df_sample_sizes
```

List of Code C.6: The Python script is used to perform the primary analysis of this thesis. That involves running the connectivity measures to yield weight matrices, calculating reliability scores and conducting power analyses. Classes and functions that are utilized to present the results in Chapter 3, except the code from plotting, are included in the code.

Bibliography

[1] C. T. Butts. ‘Revisiting the foundations of network analysis’. In: *Science* 325.5939 (2009), pp. 414–416.

[2] T. H. Murphy and D. Corbett. ‘Plasticity during stroke recovery: From synapse to behaviour’. In: *Nature Reviews Neuroscience* 10.12 (2009), pp. 861–872.

[3] J. Lee, A. Lee, H. Kim, W. H. Chang and Y. H. Kim. ‘Differences in motor network dynamics during recovery between supra- and infra-tentorial ischemic strokes’. In: *Human Brain Mapping* 39.12 (2018), pp. 4976–4986.

[4] M. Branscheidt et al. ‘No evidence for motor-recovery-related cortical connectivity changes after stroke using resting-state fMRI’. In: *Journal of neurophysiology* 127.3 (2022), pp. 637–650.

[5] K. J. Friston. ‘Functional and Effective Connectivity: A Review’. In: *Brain Connectivity* 1.1 (2011), pp. 13–36.

[6] M. L. Stanley, M. N. Moussa, B. M. Paolini, R. G. Lyday, J. H. Burdette and P. J. Laurienti. ‘Defining nodes in complex brain networks’. In: *Frontiers in Computational Neuroscience* 0.NOV (2013), p. 169.

[7] S. B. Eickhoff, B. T. Yeo and S. Genon. *Imaging-based parcellations of the human brain.* 2018. URL: https://www.nature.com/articles/s41583-018-0071-7.

[8] H. E. Wang, C. G. Bénar, P. P. Quilichini, K. J. Friston, V. K. Jirsa and C. Bernard. ‘A systematic framework for functional connectivity measures’. In: *Frontiers in Neuroscience* 8.DEC (2014), p. 405.

[9] R. Botvinik-Nezer et al. ‘Variability in the analysis of a single neuroimaging dataset by many teams’. In: *Nature* 582.7810 (2020), pp. 84–88.

[10] M. L. Elliott, A. R. Knodt, D. Ireland, M. L. Morris, R. Poulton, S. Ramrakha, M. L. Sison, T. E. Moffitt, A. Caspi and A. R. Hariri. ‘What Is the Test-Retest Reliability of Common Task-Functional MRI Measures? New Empirical Evidence and a Meta-Analysis’. In: *Psychological Science* 31.7 (2020), pp. 792–806.

[11] K. S. Button, J. P. Ioannidis, C. Mokrysz, B. A. Nosek, J. Flint, E. S. Robinson and M. R. Munafò. 'Power failure: Why small sample size undermines the reliability of neuroscience'. In: *Nature Reviews Neuroscience* 14.5 (2013), pp. 365–376.

[12] D. Szucs and J. P. Ioannidis. 'Sample size evolution in neuroimaging research: An evaluation of highly-cited studies (1990–2012) and of latest practices (2017–2018) in high-impact journals'. In: *NeuroImage* 221 (2020), p. 117164.

[13] R. A. Poldrack, C. I. Baker, J. Durnez, K. J. Gorgolewski, P. M. Matthews, M. R. Munafò, T. E. Nichols, J. B. Poline, E. Vul and T. Yarkoni. 'Scanning the horizon: Towards transparent and reproducible neuroimaging research'. In: *Nature Reviews Neuroscience* 18.2 (2017), pp. 115–126.

[14] Norwegian University of Science and Technology. *Rehabilitation of Stroke Patients for Maximal Neurological Restoration*. 2021. URL: https://clinicaltrials.gov/ct2/show/NCT05086055 (visited on 27/12/2021).

[15] A. Woodruff. *What is a neuron?* 2018. URL: https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron (visited on 04/10/2021).

[16] *Action potentials and synapses*. 2016. URL: https://qbi.uq.edu.au/brain-basics/brain/brain-physiology/action-potentials-and-synapses (visited on 13/10/2021).

[17] F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, W. J. Filho, R. Lent and S. Herculano-Houzel. 'Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain'. In: *Journal of Comparative Neurology* 513.5 (2009), pp. 532–541.

[18] H. C. Eric. *Brain facts and figures*. 2006. URL: https://faculty.washington.edu/chudler/facts.html#neuron (visited on 26/12/2021).

[19] N. K. Logothetis, J. Pauls, M. Augath, T. Trinath and A. Oeltermann. 'Neurophysiological investigation of the basis of the fMRI signal'. In: *Nature* 412.6843 (2001), pp. 150–157.

[20] 'Anatomical Terminology'. PhD thesis. 1920, p. 560.

[21] *Structural MRI Imaging*. 2019. URL: http://fmri.ucsd.edu/Howto/3T/structure.html (visited on 25/12/2021).

[22] M. A. Lindquist. 'The Statistical Analysis of fMRI Data'. In: *Statistical Science* 23.4 (2008), pp. 439–464.

[23] *What Is FMRI?* 2019. URL: http://fmri.ucsd.edu/Research/whatisfmri.html (visited on 25/12/2021).

[24] D. J. Heeger and D. Ress. 'What does fMRI tell us about neuronal activity?' In: *Nature Reviews Neuroscience* 3.2 (2002), pp. 142–151.

[25] G. Rees, K. Friston and C. Koch. 'A direct quantitative relationship between the functional properties of human and macaque V5'. In: *Nature Neuroscience* 3.7 (2000), pp. 716–723.

[26] D. J. Heeger, A. C. Huk, W. S. Geisler and D. G. Albrecht. ‘Spikes versus BOLD: What does neuroimaging tell us about neuronal activity?’ In: *Nature Neuroscience* 3.7 (2000), pp. 631–633.

[27] A. T. Reid et al. ‘Advancing functional connectivity research from association to causation’. In: *Nature Neuroscience* 22.11 (2019), pp. 1751–1760.

[28] B. Horwitz. ‘The elusive concept of brain connectivity’. In: *NeuroImage* 19.2 Pt 1 (2003), pp. 466–470.

[29] World Stroke Organization. *Learn about stroke*. 2021. URL: https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters/learn-about-stroke (visited on 16/02/2022).

[30] World Health Organization (WHO). *Stroke, Cerebrovascular accident*. 2018. URL: http://www.emro.who.int/health-topics/stroke-cerebrovascular-accident/index.html (visited on 16/02/2022).

[31] D. M. Guimarães, B. Valério-Gomes and R. Lent. ‘Neuroplasticity: The Brain Changes Over Time!’ In: *Frontiers for Young Minds* 8 (2020).

[32] R. Kong et al. ‘Individual-Specific Areal-Level Parcellations Improve Functional Connectivity Prediction of Behavior’. In: *Cerebral Cortex* 31.10 (2021), pp. 4477–4500.

[33] B. T. Thomas Yeo et al. ‘The organization of the human cerebral cortex estimated by intrinsic functional connectivity’. In: *Journal of Neurophysiology* 106.3 (2011), pp. 1125–1165.

[34] G. James, D. Witten, T. Hastie and R. Tibshirani. *An Introduction to Statistical Learning (2nd Edition)*. Vol. 102. Springer Texts in Statistics. New York, NY: Springer US, 2021, p. 618. arXiv: arXiv:1011.1669v3.

[35] W. K. Härdle and L. Simar. *Applied multivariate statistical analysis (4th edition)*. Springer Berlin Heidelberg, 2015, pp. 1–580.

[36] J. Lee Rodgers and A. Nicewander. ‘Thirteen ways to look at the correlation coefficient’. In: *American Statistician* 42.1 (1988), pp. 59–66.

[37] M. D. Nefzger and J. Drasgow. ‘The needless assumption of normality in Pearson’s r.’ In: *American Psychologist* 12.10 (1957), pp. 623–625.

[38] A. Binder. ‘Considerations of the place of assumptions in correlational analysis.’ In: *American Psychologist* 14.8 (1959), pp. 504–510.

[39] P. Schober and L. A. Schwarte. ‘Correlation coefficients: Appropriate use and interpretation’. In: *Anesthesia and Analgesia* 126.5 (2018), pp. 1763–1768.

[40] F. J. Anscombe. ‘Graphs in statistical analysis’. In: *Key Topics in Surgical Research and Methodology* 27.1 (2010), pp. 441–475.

[41] ‘Time Series Regression and Exploratory Data Analysis’. In: *Time Series Analysis and Its Applications*. Springer, New York, NY, 2006, pp. 48–83.

[42] K. D. Harris. ‘Nonsense correlations in neuroscience’. In: *bioRxiv* (2020), p. 2020.11.29.402719.

[43] P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data*. Springer Series in Statistics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[44] S. L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

[45] T. et. all. Hastie. *The Elements of Statistical Learning*. Vol. 27. 2. Springer, 2009, pp. 83–85.

[46] T. Hastie, R. Tibshirani and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. 2015, pp. 1–337.

[47] J. Friedman, T. Hastie and R. Tibshirani. 'Sparse inverse covariance estimation with the graphical lasso'. In: *Biostatistics* 9.3 (2008), pp. 432–441.

[48] K. J. Friston, L. Harrison and W. Penny. 'Dynamic causal modelling'. In: *NeuroImage* 19.4 (2003), pp. 1273–1302.

[49] A. Razi, M. L. Seghier, Y. Zhou, P. McColgan, P. Zeidman, H. J. Park, O. Sporns, G. Rees and K. J. Friston. 'Large-scale DCMs for resting-state fMRI'. In: *Network Neuroscience* 1.3 (2017), pp. 222–241.

[50] R. B. Buxton, E. C. Wong and L. R. Frank. 'Dynamics of blood flow and oxygenation changes during brain activation: The balloon model'. In: *Magnetic Resonance in Medicine* 39.6 (1998), pp. 855–864.

[51] L. Held and D. Sabanés Bové. *Applied statistical inference: Likelihood and bayes*. Vol. 9783642378. Springer-Verlag Berlin Heidelberg, 2014, pp. 1–376.

[52] K. Friston, J. Mattout, N. Trujillo-Barreto, J. Ashburner and W. Penny. 'Variational free energy and the Laplace approximation'. In: *NeuroImage* 34.1 (2007), pp. 220–234.

[53] W. D. Penny. 'Comparing Dynamical Causal Models using AIC, BIC and Free Energy'. In: *NeuroImage* (2011).

[54] P. Zeidman, A. Jafarian, N. Corbin, M. L. Seghier, A. Razi, C. J. Price and K. J. Friston. 'A guide to group effective connectivity analysis, part 1: First level analysis with DCM for fMRI'. In: *NeuroImage* 200 (2019), pp. 174–190. arXiv: 1902.10597.

[55] K. Friston, R. Moran and A. K. Seth. 'Analysing connectivity with Granger causality and dynamic causal modelling'. In: *Current Opinion in Neurobiology* 23.2 (2013), pp. 172–178.

[56] K. J. Friston, J. Kahan, B. Biswal and A. Razi. 'A DCM for resting state fMRI'. In: *NeuroImage* 94 (2014), pp. 396–407.

[57] G. M. Sullivan and R. Feinn. 'Using Effect Size—or Why the P Value Is Not Enough'. In: *Journal of Graduate Medical Education* 4.3 (2012), pp. 279–282.

[58] J. Suckling et al. 'Are power calculations useful? A multicentre neuroimaging study'. In: *Human Brain Mapping* 35.8 (2014), pp. 3569–3577.

[59] L. Fahrmeir, T. Kneib, S. Lang and B. Marx. *Regression: Models, methods and applications*. Vol. 9783642343. Springer-Verlag Berlin Heidelberg, 2013, pp. 1–698.

[60] P. McCarthy. *FSLeyes (1.4.0)*. 2022. URL: https://zenodo.org/record/6511596.

[61] 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'. In: *Proceedings of the National Academy of Sciences of the United States of America* 102.27 (2005), pp. 9673–9678.

[62] W. Penny, K. Friston, J. Ashburner, S. Kiebel and T. Nichols. *Statistical Parametric Mapping (SPM)*. 2007. URL: https://www.fil.ion.ucl.ac.uk/spm/software/spm12/.

[63] N. J. Foti and E. B. Fox. 'Statistical model-based approaches for functional connectivity analysis of neuroimaging data'. In: *Current Opinion in Neurobiology* 55 (2019), pp. 48–54.

[64] 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'. In: *NeuroImage* 54.2 (2011), pp. 875–891.

[65] S. A. Valizadeh, F. Liem, S. Mérillat, J. Hänggi and L. Jäncke. 'Identification of individual subjects on the basis of their brain anatomical features'. In: *Scientific Reports* 8.1 (2018).

[66] J. Dubois and R. Adolphs. 'Building a Science of Individual Differences from fMRI'. In: *Trends in Cognitive Sciences* 20.6 (2016), pp. 425–443.

[67] M. M. Monti. 'Statistical analysis of fMRI time-series: A critical review of the GLM approach'. In: *Frontiers in Human Neuroscience* 0.MARCH (2011), p. 28.

[68] J. Teeuw, H. E. Hulshoff Pol, D. I. Boomsma and R. M. Brouwer. 'Reliability modelling of resting-state functional connectivity'. In: *NeuroImage* 231 (2021), p. 117842.

[69] E. H. Keulers, P. Stiers, N. A. Nicolson and J. Jolles. 'The association between cortisol and the BOLD response in male adolescents undergoing fMRI'. In: *Brain Research* 1598 (2015), pp. 1–11.

[70] S. S. Yoo, N. Gujar, P. Hu, F. A. Jolesz and M. P. Walker. 'The human emotional brain without sleep - a prefrontal amygdala disconnect'. In: *Current Biology* 17.20 (2007), R877–R878.

[71] M. L. Elliott et al. 'General functional connectivity: Shared features of resting-state and task fMRI drive reliable and heritable individual differences in functional brain networks'. In: *NeuroImage* 189 (2019), pp. 516–532.

[72] S. Frässle, E. I. Lomakina, A. Razi, K. J. Friston, J. M. Buhmann and K. E. Stephan. 'Regression DCM for fMRI'. In: *NeuroImage* 155 (2017), pp. 406–421.

[73] Y. Wang, J. Kang, P. B. Kemmer and Y. Guo. 'An efficient and reliable statistical method for estimating functional connectivity in large scale brain networks using partial correlation'. In: *Frontiers in Neuroscience* 10.MAR (2016), p. 123.

[74] M. N. Hallquist and F. G. Hillary. 'Graph theory approaches to functional network organization in brain disorders: A critique for a brave new small-world'. In: *Network Neuroscience* 3.1 (2018), pp. 1–26.

[75] J. C. Griffis, N. V. Metcalf, M. Corbetta and G. L. Shulman. 'Damage to the shortest structural paths between brain regions is associated with disruptions of resting-state functional connectivity after stroke'. In: *NeuroImage* 210 (2020), p. 116589.

[76] L. M. Shah, J. A. Cramer, M. A. Ferguson, R. M. Birn and J. S. Anderson. 'Reliability and reproducibility of individual differences in functional connectivity acquired during task and resting state'. In: *Brain and Behavior* 6.5 (2016), e00456.

[77] I. T. Jollife and J. Cadima. 'Principal component analysis: A review and recent developments'. In: *Phil. Trans. R. Soc. A.* 374.2065 (2016).

[78] C. M. Bishop. *Pattern recognition and machine learning*. Springer New York, NY, p. 738.

[79] L. P. Cinelli, M. A. Marins, E. A. Barros da Silva and S. L. Netto. *Variational Methods for Machine Learning with Applications to Deep Networks*. Springer International Publishing, 2021.

[80] G. Casella and R Berger. 'Statistical Inference. 2nd Edition'. In: *Duxbury Press*. Pacific Groove: Duxbury Press, 2001.

[81] O. F. Gulban, D. Nielson, R. Poldrack, J. Lee, C. Gorgolewski, Vanessasaurus and S. Ghosh. 'pydeface: v2.0.0'. In: (2019).

[82] U. S. Choi, H. Kawaguchi, Y. Matsuoka, T. Kober and I. Kida. 'Brain tissue segmentation based on MP2RAGE multi-contrast images in 7 T MRI'. In: *PLoS ONE* 14.2 (2017), e0210803.

[83] Sriranga Kashyap. *Scripts to prepare MP2RAGE for Freesurfer*. URL: https://github.com/srikash/presurfer.

[84] S. Kashyap, D. Ivanov, M. Havlicek, L. Huber, B. A. Poser and K. Uludağ. 'Sub-millimetre resolution laminar fMRI using Arterial Spin Labelling in humans at 7 T'. In: *PLoS ONE* 16.4 April (2021), e0250504.

[85] O. Esteban et al. 'fMRIPrep: a robust preprocessing pipeline for functional MRI'. In: *Nature Methods* 16.1 (2019), pp. 111–116.

[86] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich and S. M. Smith. 'FSL'. In: *NeuroImage* 62.2 (2012), pp. 782–790.

[87] *Analysis of Functional NeuroImages (AFNI)*. URL: https://afni.nimh.nih.gov/ (visited on 03/06/2022).

[88] Y. Behzadi, K. Restom, J. Liau and T. T. Liu. ‘A component based noise correction method (CompCor) for BOLD and perfusion based fMRI’. In: *NeuroImage* 37.1 (2007), pp. 90–101.

[89] B. B. Avants, N. J. Tustison and H. J. Johnson. *ANTs*. 2009. URL: http://stnava.github.io/ANTs/.

[90] M. Galloway. *CVglasso: Lasso Penalized Precision Matrix Estimation*. 2018. URL: https://cran.r-project.org/package=CVglasso.

[91] A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Mueller, J. Kossaifi, A. Gramfort, B. Thirion and G. Varoquaux. ‘Machine learning for neuroimaging with scikit-learn’. In: *Frontiers in Neuroinformatics* 8.FEB (2014), p. 14. arXiv: 1412.3919.

[92] M. Waskom. *seaborn: statistical data visualization*. 2021. URL: https://doi.org/10.21105/joss.03021.

[93] J. D. Hunter. *Matplotlib: A 2D graphics environment*. 2007.

NTNU
Norwegian University of
Science and Technology