ACIT5930

MASTER'S THESIS phase III

in

Applied Computer and Information Technology (ACIT)

May 2021

Mathematical Modeling and Scientific Computing

A mutual information approach on fNIRS functional connectivity network

Sergio Alejandro Sotres Romero

Department of Computer Science
Faculty of Technology, Art and Design

Preface

This work is intended to explore fNIRS data using statistical and information theory tools for brain activity. The project was completed during an uncertain moment in modern history given the presence of a global pandemic. I want to dedicate this work to M. Edit Romero Hernández, Sergio A. Sotres Hernández for giving me the support on this new project of my life. I also want to thank Jared R. Romero Hernández and Julieta E. Sotres R. for all the support and encouragement you have provided me over the years. A very special mention to Vilde A. S. for giving me all the support and patience to complete this project over the last few months.

I also want to thank my supervisors Pedro G. Lind and Peyman Mirtaheri for introducing me to the topic and for the valuable advice provided over the last year.

Sergio Alejandro Sotres Romero
Oslo, Norway, May 2021

Abstract

The functional near-infrared spectroscopy (fNIRS), as a brain imaging modality, is a versatile technique for understanding brain activity processes at the level of the brain cortex. The use of this technology facilities the understanding of brain metabolism, oxygenation, and its related brain activity parameters when participants perform dynamical tasks. In this thesis, we apply different methods to extract the functional connectivity network of the brain, employing data generated by this technology. This functional connectivity network is a measure that qualitatively informs the interconnection of different regions of the brain. To perform such a task, we calculated the Pearson correlation coefficients and the mutual information between pairs of signals from fNIRS data, to determine the strength of shared information among them. We construct weighted networks that display the more correlated regions and compare these methods to unsupervised learning techniques such as PCA, ICA, and dendrograms. Additionally, we include an implementation where we explore nonlinear dependencies of fNIRS data using mutual information.

From this analysis, we observed that a mutual information approach based on binning techniques allows quantifying more general correlations than using the Pearson coefficient but is highly susceptible to bias. The method also provides more relevant information compared to the PCA and ICA, since with the last one, we can observe the dependencies of signals but in a disorderly manner. The resulted bias is been reflected in lower values that are more visible when doing a threshold examination (5.1,5.7). A deeper analysis in this regard to bias reduction needs further exploration in future work. Additionally, the calculation of a coefficient (referred to in the thesis as Λ) that distinguishes the type of dependence between random variables resulted to be a useful method for fNIRS data. Such a coefficient indicates a clear way to quantify linear and nonlinear dependencies by using mutual information, but with the incapability of reflecting the specific type of behavior involved.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is often said that there are two mysteries in nature that win the title for being the most challenging and mesmerizing to unravel: the human mind and the universe itself. When one drives the attention to the study of the mind, it is often possible to particularize the problem on the biological system: the brain. With the technological advances of today, computers are becoming more powerful and more capable of performing more complicated tasks compared to the ones at the beginning of the millennium. Progress in material and computer sciences has allowed researchers to explore different areas of knowledge by using computational and technological tools that can provide a better understanding of the behavior of biological phenomena. Simultaneously, the fast-growing technological advances have opened the door for computers to be powerful enough to receive, record, and process different types of signals emanating from the brain and translate physical signals into a digital language more efficiently. This task allows the creation of an interface between the brain and the computer in a communication system that could, for example, enable the brain to control external devices around it. Such interfaces are usually named *brain-machine interface* (BMI) or to be more precise for this thesis, *brain-computer interface* (BCI). This field of study explores the technology responsible to collect information from the brain and the *translation* of it into data that can be posteriorly analyzed.

In that particular field, there is one technology that concerns this thesis known as *functional near-infrared spectroscopy* (fNIRS). The use of fNIRS has been gaining terrain over recent years due to its practical usage and versatility for conducting experiments as neuroimaging concerns. This technology compares the amount of oxygenated and non-oxygenated blood regions of the brain after a task performed by the individual of study. The use of fNIRS offers certain advantages over other techniques for neuroimaging (see Chapter 2) such as functional magnetic resonance (fMRI), electroencephalography (EEG), and magnetoencephalography (MEG); and is because of that reason we are motivated in using it to collect data from the brain. Some features that make fNIRS particularly interesting are its non-invasiveness, its portability, and its tolerance to body movements (although it has its disadvantages as well). Consequently, it enables the possibility to be suitable for a wide range of applications and also, a more flexible use for data collection inside or outside of a laboratory. The latter brings some attention particularly to the studies involving data collection from cognitive neuroscience studies and different analyses on activated regions from brain activity.

Given the fact that the brain is highly interconnected and since the brain activates certain regions while performing a task, an immediate question is: How to quantify the amount of *shared* information between different areas of the brain? Or in other words, how to quantitatively evaluate the relationship between a region in the brain and its neighboring regions by looking at the fNIRS data? To answer that, it is proposed that statistical and information theory measures are good candidates to be used for extracting the connectivity network in the brain. When it comes to correlation measures, one can think of two types depending on the nature of the information at our disposal. These two notions are what we call in this thesis linear and non-linear measures. Once such measures are applied, another question to explore is: How good is fNIRS data for extracting the functional connectivity network between regions of the brain?

The main focus of this thesis is to answer the pair of questions presented above by using the fNIRS technology to extract information from regions of the brain while a participant executes a specific task. We later analyze it with statistical tools and build functional connectivity networks for the oxygenated and non-oxygenated data. To draw a connectivity network we use statistical weights (more specifically, a family of weights) that measure types of correlation between the activated channels measured with fNIRS data. The goal of this thesis is to compare the accuracy of these measures with the actual physiological hypothesis. The data used in this thesis was collected from several activated channels localized in the prefrontal cortex that have collected oxygenized and non-oxygenized hemoglobin lectures as a function of time.

In Chapter 2, we start by discussing the main features of fNIRS technology and what it consists of. Later on, we give a presentation about the fundamentals of multivariable analysis, introducing the concept of correlations and statistical tools and how they apply to sets of data. We introduce concepts from information theory to study more general approaches of correlations and discuss some unsupervised learning techniques. In Chapter 3 we describe the methodology to follow, and some approaches for analyzing fNIRS data and measuring nonlinear dependencies. In Chapter we describe a synthetic data generation procedure to test the correlation measures and in Chapter we apply the framework to a particular fNIRS data sets. Lastly, in Chapter 6 we discuss the results obtained and conclude the thesis.

Chapter 2

Fundamental concepts

The emergent technology of neuroimaging has had a very crucial impact on our understanding of brain activity, the functioning of the brain, and its health. In that regard, neuroimaging is surrounded by many constraints due to the inaccessibility to the brain and the complexity of such an organ during the execution of tasks. Technologies such as functional Magnetic Resonance Imaging (fMRI), magnetoencephalography (MEG), electroencephalography (EEG), Positron Emission Tomography (PET), and *functional Near-Infrared Spectroscopy* (fNIRS); have taken part in a big growth of a way to monitor brain activity in a wide range of applications, going from just capturing images on brain structures to obtaining information about cognitive activities and sensorimotor systems. In particular, fNIRS technology has demonstrated to be one of the most successful areas to investigate brain activation and neuroimaging given the fact that it is a suitable option for analyzing body movements due to its portability and low sensitivity to movements, unlike other options that use stationary scanners. This non-invasive neuroimage device provides lots of opportunities in exploring regions of the brain from a modern perspective. In fact, in the present work, data collected employing an fNIRS device will be used for the implementation of a functional connectivity network [36].

In this chapter, we will present a brief overview of technologies in biomedical engineering dedicated to exploring the brain. In particular, we will focus on fNIRS and study the basics of how it works. Because the goal of this thesis is to implement a functional connectivity network from brain data, the discussion will be followed by an introduction to mathematical concepts involving data analysis. In that section, we will cover the concept of correlation and covariance followed by a discussion of some nonlinear correlation measures from information theory in addition to a method that uses mutual information to calculate nonlinearity quantitatively. We end this chapter by mentioning some common unsupervised learning techniques that will be used later on in the text to explain certain features of the data.

2.1 Technologies used in neuroscience

In biomedical engineering, there are many tools and techniques to study the human brain [39]. In order to have a deeper understanding of how it works, it is necessary to look inside such an organ [28]. That task is today realized by brain imaging methods that act in an entirely non-invasive way. In this section, we will describe some of the most common brain imaging technologies and have a comparison between them by looking at the advantages and disadvantages of each. The list of available technologies for studying human cognition includes: (f)MRI, PET, EEG, MEG and fNIRS. The first two rely on neurovascular coupling, the third and fourth detect the electromagnetic activity of the brain and the last one relies on infrared spectroscopy of light. These technologies are typically compared based on the temporal and spatial resolution of each, but before going into detail there, it is better to have a short explanation of some of them.

2.1.1 EEG

Electroencephalography (EEG) is a technique that measures the electrical activity of the brain through electrodes that are placed on the scalp. Such electrodes measure brain activity and its changes, as a response to some stimuli [29]. The electrodes used in EEG detect only electrical changes of a large number of neurons that respond to a signal at the same time. The information from the electrodes is then amplified and received by a computer where data is analyzed. A complication from this technology is that due to the spatial resolution, it is difficult to know how deep the signal is produced [30].

2.1.2 MRI

Magnetic resonance imaging (MRI) is a complex imaging technique that uses strong magnetic fields to interact with the protons of the hydrogen atoms in the body. Given the fact that the organic tissue consist of a high degree of water [23], this technology uses this feature for obtaining images of organs. The strong magnetic field aligns the protons in a certain direction, but when a radio pulse is emitted, it interacts with the protons flipping them in their orientation. As the protons go back to the alignment, there is an energy release detected by the MRI machine [20] [13] and analyzed by a computer to create the image of the tissue.

2.1.3 fMRI and fNIRS

Functional MRI (fMRI) [6] works in a similar way as described above, but the main difference with MRI is that the intention here is focused on determining the changes in the flow of oxygenated blood [13].

On the other hand, functional Near-Infrared Spectroscopy (fNIRS) [2] is similar to fMRI in the sense that it relies on the blood oxygen level-dependent signal that happens when the neurons activate and consume oxygen. The differences with fMRI are that for this technique the presence of the magnetic is crucial whereas in fNIRS only optical properties are considered [4]. To be more specific, in fMRI the deoxygenated hemoglobin affects more the magnetic field compared to the oxygenated hemoglobin therefore, the ratio of these two quantities is analyzed to measure brain activity. fNIRS, on the other hand, takes advantage of the different absorption spectra between oxygenated and deoxygenated hemoglobin.

Because the utility of a particular neuroimaging technique can be assessed in a variety of ways [34], it is important to discuss why is it relevant to focus on fNIRS for the purpose of this thesis. This is done in the next section. The following table shows a brief comparison between different neuroimaging technologies that are used nowadays.

	fNIRS	fMRI	EEG/MEG	PET
Signal	**HbO2/HbR**	BOLD (HbR)	Electromagnetic	Glucose metabolism
Spatial resolution	**2–3** cm	0.3 mm voxels	5–9 cm	4 mm
Penetration depth	**Brain cortex**	Whole head	Brain cortex/deep	Whole head
Sampling rates	**1-200 Hz**	1–3 Hz	>1000 Hz	<0.1 Hz
Range of tasks	**Enormous**	Limited	Limited	Limited
Motion	**Very good**	Limited	Limited	Limited
Participants	**Everyone**	Limited	Everyone	Limited
Sounds	**Silent**	Very noisy	Silent	Silent
Portability	**Yes**	None	Yes	None
Cost	**Low**	High	Low /high	High

Table 2.1: *Comparison between neuroimaging methods. In this table, the information of strengths and weaknesses of the neuroimaging methods are displayed [4] [15].*

As seen from Table 2.1, fNIRS systems have many advantages when it comes to comfort, cost, and portability. The main feature of this technology is that the optical components do not interfere with electromagnetic fields allowing the researchers to gather a more complete set of information from individuals.

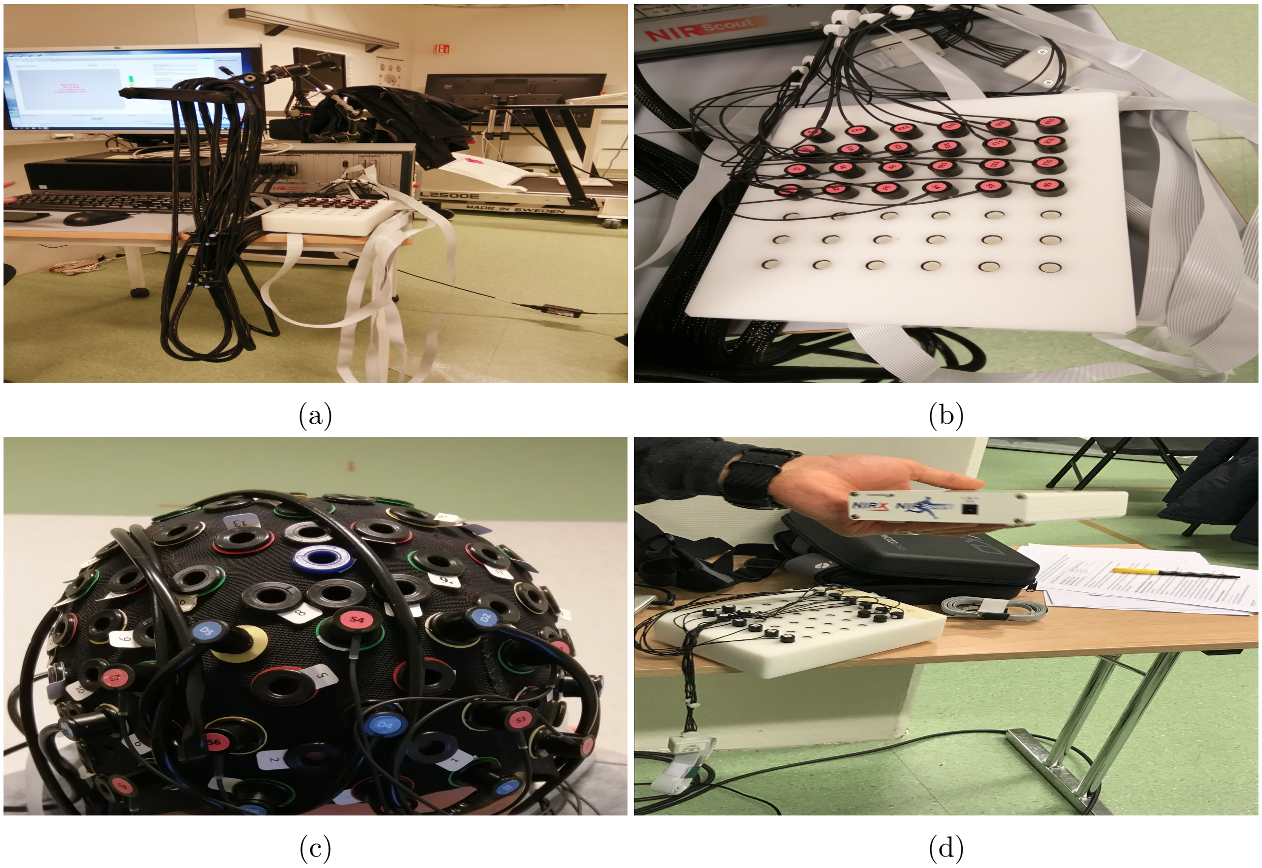
Due to the wide range of advantages, this technology is used in various institutions and research centers such as universities. In particular, this fact allows for generating a wide collection of data in different experiments from which it is possible to extract information from the connectivity of the brain. It should be noted that this last is the main objective of this thesis topic, so the use of this technology will be discussed in more detail in the next section.

2.2 The fNIRS technology

To collect information from brain activity, fNIRS needs measurements that can compare physical quantities from the regions of interest [12] [9]. The measurements in this case, are typically performed by transmitting infrared light onto the head of an individual and compare it to the one that is received. fNIRS consist of a set of electrodes that emit infrared light that is shot onto the scalp [26] (Figure 2.1). The light goes through several different organic layers that have different optical properties [12]. Along its path, the light is absorbed and scattered not only due to the equipment, but also because irregularities and composition of organic tissue is involved [26].

When the light goes through a material, the photons [1] can be absorbed, transmitted, or reflected as a result of interaction with the barrier. In the human body, it is well-known [12] [5] that one of the most infrared absorbing chromophore substance is *hemoglobin.* Such

[1] Elementary blocks of light.



(a) (b)

(c) (d)

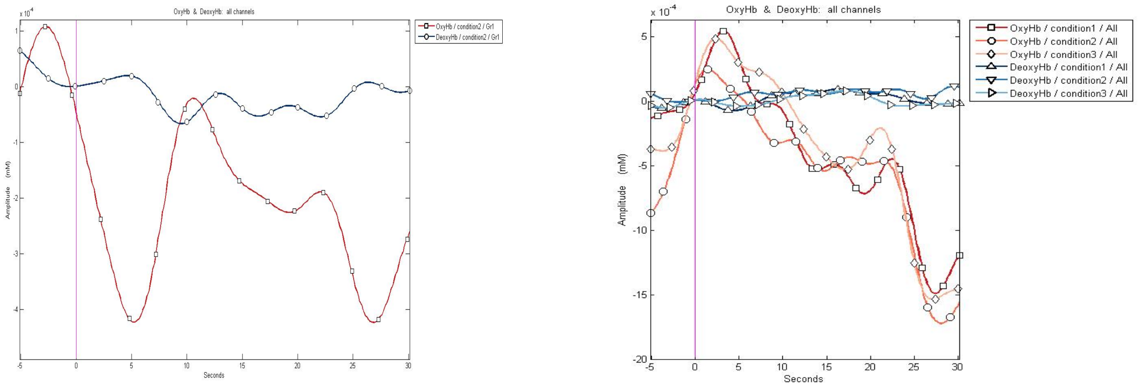
Figure 2.1: *Here we can see the setup of the fNIRS device used for the data collection of this thesis. In a) the setting presented is connected to the NIRscout device. In b), c) we see the electrodes and the cap used and in d) the NIRX sport device is shown.*

a large molecule (or protein to be more specific) is responsible for providing oxygen to the bloodstream. Because hemoglobin is the oxygen carrier in the body, the amount of oxygen that it contains has a noticeable feature when it comes to absorption. Specifically for infrared light. Oxygenated hemoglobin (oxyhemoglobin, HbO_2) and deoxyhemoglobin HbR absorb near-infrared (NIR) light between 650-900 nm [2]. With this in mind, it is possible to use these features to make use of spectroscopical measurements and tools to localize oxygenated areas in the brain. When a brain area is active and involved in a certain task, the brain requires a supply of glucose and oxygen resulting in an increase of blood flow. Such an increase is proportional to the increase in HbO_2 and simultaneously a decrease in HbR concentrations. The differences in concentrations are measured by the estimation of light *attenuation* with fNIRS.

As mentioned above, the NIR light also suffers a process of scattering which is more frequent than absorption contributing to light attenuation. This means that the more photons scattered, the longer the traveled path and the greater the probability of being absorbed [26]. These issues are considered and treated with fNIRS to detect brain activity.

[2] According to [12], HbO_2 absorption is higher for wavelengths in the range of $\lambda > 800$ nm, while HbR absorption coefficient is in the range of $\lambda < 800$ nm.

It is worth mentioning that recent works have been done using this technology. Investigations about brain activity for people with amputated limbs and finger tapping [36] from Norwegian institutions (Figure 2.2) are the base of this thesis. Because in such works they have used fNIRS technology for collecting the data, it is interesting to explore correlation measures to implement the connectivity network from regions of the brain for such data set [36].



(a)

(b)

Figure 2.2: *Example of data collected from fNIRS technology. In the plot displayed in a), it is possible to see the amplitude of the signals received from HbO_2 and HbR during a time interval. In b), it is possible to see the comparison between 3 different conditions. These plots were provided from a set of measures at OsloMet taken for a study of patients with Multiple sclerosis (MS). For further references check: [10, 1, 37, 33] and [36].*

2.3 Fundamentals of multivariate data analysis

The use of fNIRS technology and its relevance to this thesis, have been discussed in the previous sections. However, the core of this project is related to signal analysis and the implementation of techniques to measure the correlation between them. To begin the discussion regarding correlation measures, it is worth introducing basic concepts such as probability density, linear and nonlinear correlation measures, as well as certain basic generalities in stochastic processes [21]. This last concept is attributed to the need of adopting a formalism that quantifies the temporal evolution of the signals since, as is well known, the experimental measurements are parameterized with the time associated with the duration of the signal.

2.3.1 Probability density function

The Bayesian interpretation of *probability* is a measurement that quantifies the likelihood of an event to happen. This approach introduces the notion of information and the *uncertainty* of an event to occur based on that information. Probability has as its basis a concept known as *random variables.* When an experiment is performed, the object of interest is some function of the outcome as opposed to the actual outcome itself. Let us think about tossing a coin to exemplify this matter. In this case, one is interested in the number of heads and not the actual head/tail sequence that results. These quantities of interest, or, more formally, these real-valued functions defined on the sample space, are known as *random variables.*

From probability theory, it is known that a *random variable* is a quantity that can take on numerical values with certain probabilities [24], [35]. However, it is possible to consider another type of random variables whose set of possible values is either finite or countably infinite and in some cases, uncountable.

Let's consider X to be a random variable. X is considered to be a *continuous* random variable if there exists a non negative function $f \in \mathbb{R}$, with the property that for any set $B \in \mathbb{R}$

$$P(X \in B) = \int_B f(x)dx. \tag{2.1}$$

The function f is called the *probability density function* of X and it contains the information of the probabilities for the set once the function has been integrated. In other words, what eq. (2.1) is saying is that the probability of X being in B, is calculated by taking the integral of the probability density over the set B. And because X must have a particular value, $f(x)$ must satisfy

$$P\left[X \in (-\infty, \infty)\right] = \int_{-\infty}^{\infty} f(x)dx = 1. \tag{2.2}$$

Eq. (2.2) means that the probability of measuring any value, is one. Now, if $B = [a, b] \in \mathbb{R}$, the probability of finding X in the interval $[a, b]$ is

$$0 \leq P(X \in [a, b]) = \int_a^b f(x)dx \leq 1. \tag{2.3}$$

Notice that (2.3) takes values in the real line, but because $[a, b]$ is a subset of $\mathbb{R}$,

$$P\left[X \in [a, b]\right] \in [0, 1].$$

With the concept of *random variable* introduced, an interesting question to ask could be that given a pair of random variables X and Y, to what extend having some knowledge of X helps predict Y? (or vice versa). In other words, we are asking about the degree to which two random variables are *correlated* [24].

2.3.2 Correlation and covariance

The larger the correlation between variables, the more information we know about a variable helping to predict the other. It is possible to exemplify qualitatively, what correlation means. If it happens to perceive an increase in X that corresponds to an increase in Y (on average) then, we can call this association as a *positive* correlation. On the other hand, we say a *negative* correlation occurs when an increase in X corresponds to a decrease in Y (on average). Having said this, it is important to clarify that the concept of *correlation* does not necessarily imply *causation* [24].

To understand the general way in which random variables are correlated, it useful to introduce the correlation coefficient r that will be defined below. By using the central limit theorem, it is possible to approximate real-life variables to normal distributions. In this direction, we can consider a random variable X normally distributed with $\mu = 0$ (to simplify the argument) and standard deviation σ_x. Now, if another random variable Y is partially determined (in a linear way) by X and partially determined by another random variable Z with standard deviation σ_z and $\mu_z = 0$ (independent of X); it is possible to quantify the dependence of Y on X and Z by expressing

$$Y = mX + Z \tag{2.4}$$

where m is a real scalar.

In (2.4) mX and Z have standard deviation $m\sigma_x$ and σ_z respectively. And because we are considering the mean (or expectation value) of Y to be $\mu_y = m\mu_x + \mu_z = 0$ then $\sigma_y = \sqrt{m^2\sigma_x^2 + \sigma_z^2}$.

All these quantities define the *correlation coefficient* $\boldsymbol{r}$, a scalar that provides a way of measuring the degree of correlation between variables. For given values of m and σ_z in a linear dependence attributed to X as stated above, the correlation coefficient is expressed as

$$r \equiv \frac{m\sigma_x}{\sigma_y} = \frac{m\sigma_x}{\sqrt{m^2\sigma_x^2 + \sigma_z^2}}. \tag{2.5}$$

The meaning of the equality in (2.5) is that r^2 is the ratio of the variance of Y that can be attributed to X.

Because the analysis in this thesis corresponds to a collection of signals (or more generally, a collection of data points) to determine correlations (and therefore r), it is important to introduce the concept of *covariance.* In this sense, just as the expected value and the variance of a single random variable give information about the random variable of interest, the same happens with the covariance but between two random variables.

The *covariance* between X and Y, denoted by $\mathrm{Cov}(X, Y)$ is defined as

$$\mathrm{Cov}(X, Y) \equiv E[(X - E[X])(Y - E[Y])] = E[(X - \mu_x)(Y - \mu_y)], \tag{2.6}$$

where $E[X]$ is the expectation value of the random variable X. In case the X is a discrete random variable, $E[X]$ is expressed as

$$E[X] = \sum_x xP(X = x). \tag{2.7}$$

In the case X is a continuous random variable with probability density function $f(x)$ then

$$E[X] = \int_{-\infty}^{\infty} xf(x)dx. \tag{2.8}$$

Taking this into consideration we can extend more generally the idea that for any real-valued function $g(x)$, eq. (2.9) represents the expectation value of such function given a probability density $f(x)$.

$$E[g(x)] = \int_{-\infty}^{\infty} g(x)f(x)dx. \tag{2.9}$$

Having defined the covariance, it is possible to see that eq. (2.5) can be written in terms of it. Given the fact that the assumption of X and Z are independent variables, it is possible to rewrite r as

$$\begin{aligned} r = \frac{m\sigma_x}{\sigma_y} = \frac{m\sigma_x^2 + 0}{\sigma_x\sigma_y} = \frac{mE(X^2) + E(XZ)}{\sigma_x\sigma_y} = \frac{E[X(mX+Z)]}{\sigma_x\sigma_y} \\ = \frac{\operatorname{Cov}(X, mX+Z)}{\sigma_x\sigma_y} = \frac{\operatorname{Cov}(X,Y)}{\sigma_x\sigma_y} \end{aligned} \tag{2.10}$$

$$\therefore \boxed{r = \frac{\operatorname{Cov}(X,Y)}{\sigma_x\sigma_y}}. \tag{2.11}$$

The reader must notice that eq. (2.11) holds regardless of the distribution. The advantage of (2.11) is that it does not contain m. Meaning that it has an advantage of use when one wants to study the correlation by using a set of data points (x_i, y_i) instead of a specific distribution.

Traditional texts like [25] and [35] denote the *correlation* of two random variables in terms of the variance instead of the *standard deviation* as long as the variance is positive. This coefficient is also called **Pearson coefficient** and this is the notation that will be followed in this thesis for the sake of clarity to the reader.

$$\rho(X,Y) = \frac{\operatorname{Cov}(X,Y)}{\sqrt{\operatorname{Var}(X)\operatorname{Var}(Y)}}, \tag{2.12}$$

where

$$\begin{cases} -1 \leq \rho(X,Y) \leq 1, \\ \operatorname{Var}(X) = E[X^2] - (E[X])^2, \\ \sigma_x \equiv \sqrt{\operatorname{Var}(X)}. \end{cases}$$

Again, this coefficient measures the linearity between X and Y so a positive value of $\rho(X,Y)$ means that Y tends to increase as X does, while a negative value represents an increase of one variable when the other decreases. If $\rho(X,Y) = 0$ X and Y are said to be *uncorrelated.*

Because the measurement of brain activity under the use of the fNIRS technology has multiple signal collection channels, it is worth mentioning the generalization of correlation measures when having a larger number of random variables. This is the intention of the next subsection.

2.3.3 Multivariable case

If we consider a more general case where an array of random variables is written as a vector $\vec{x} = (X_1, \cdots, X_d)$ in a d-dimensional space then, its **covariance matrix** is defined as

$$\mathrm{Cov}[\vec{x}] \equiv \mathbb{E}[(\vec{x} - \mathbb{E}[\vec{x}])(\vec{x} - \mathbb{E}[\vec{x}])^T],$$

where the entries of the matrix are

$$\mathrm{Cov}[\vec{x}] = \begin{bmatrix} \mathrm{Var}[X_1] & \mathrm{Cov}[X_1, X_2] & \cdots & \mathrm{Cov}[X_1, X_d] \\ \mathrm{Cov}[X_2, X_1] & \mathrm{Var}[X_2] & \cdots & \mathrm{Cov}[X_2, X_d] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[X_d, X_1] & \mathrm{Cov}[X_d, X_2] & \cdots & \mathrm{Var}[X_d] \end{bmatrix}. \tag{2.13}$$

This matrix is symmetric and positive definite, the entries of the covariance matrix codify the variance and covariance of the random variables pairwise. Given the definition of the latter quantities, it is easy to see that such entries can take values in $[0, \infty)$. Because of that reason, we can define the corresponding **correlation matrix** of $\vec{x}$ as a normalized measure with a finite upper bound

$$\mathrm{Corr}[\vec{x}] = \begin{bmatrix} \rho[X_1, X_1] & \rho[X_1, X_2] & \cdots & \rho[X_1, X_d] \\ \vdots & \vdots & \ddots & \vdots \\ \rho[X_d, X_1] & \rho[X_d, X_2] & \cdots & \rho[X_d, X_d] \end{bmatrix}. \tag{2.14}$$

Now, up to this point, the importance of quantifying the signal correlation linearly has been emphasized, but also, it is desired to know non-linear features between signals. To do this, we will make use of information theory concepts to describe nonlinear measures.

2.4 Information theory measures for nonlinear correlations

Information theory studies the representation of a certain type of data compactly together with its transmission and storage in a way that data is more susceptible to errors. Therefore, when dealing with collecting information on a system, it is often possible to *separate* the blocks of information in regions to analyze their properties separately. These different sets of information will then contain measures of the phenomena or the problem involved. In this particular case, we consider signals from brain activity as discussed above. In data analysis is important to figure out a way of measuring correlations between those so-called elementary blocks of information in different manners. In many practical cases, probabilistic models offer a good approach to solve this question just as, for example, decoding signals from noisy channels [25]. To extract as much information as possible from a set of measurements, linear and nonlinear correlation measures are considered in this thesis. Linear correlation measures were introduced in the previous section for continuous and discrete random variables. It is because of this reason that the notion of nonlinear measures needs to be introduced in this section for a more complete evaluation of the information between random variables.

Therefore it is precise to introduce the key concept of classical information theory: the *Shannon entropy*[27].

2.4.1 Shannon entropy

Let's consider again a discrete random variable X. The *Shannon entropy* of X measures in a quantitative way, how much information is gain, on average, when the value of X is learned. In other words, we can understand the Shannon entropy by considering that the entropy of X measures the amount of *uncertainty* about the variable X before we can get to know the value [27], [7]. The previous definitions are complementary to each other in the sense that it is possible to think of the entropy as a measure of our uncertainty *before* knowing the value of X, or as a measure of how much information we have gained *after* we learn the value of X.

Now, because the information content of a random variable shouldn't depend on the labels attached to the values that may be taken by the random variable, the entropy of a random variable is defined to be a function of a probability distribution, $p_1, ..., p_n$. Then the Shannon entropy associated with this probability distribution is:

$$H[X] \equiv H(p_1, ..., p_n) \equiv -\sum_k p(X=k) \log_2 p(X=k). \tag{2.15}$$

Notice that here we are using the log base 2, meaning that the units of measure for the entropy are *bits* as corresponds to classical information. We will keep this convention throughout the entire text to refer logarithms base 2 as $'\log'$ and $'\ln'$ indicating natural logarithms [3]. To justify this definition of entropy, eq.(2.15) *quantifies the necessary resources to store information* [27]. These minimal physical resources produced by the source can, at a later time, reconstruct the information via $H[X]$. More specifically, in the context of this thesis, the source which produces the information are the signals received from the electrodes connected to a participant's head. Most of real information sources consider strings of independent, identically distributed random variables X_i for modeling reality. We will use this for our future analysis along with extra considerations for modeling real data.

2.4.2 Relative entropy

Another entropy measure that is useful to introduce, is the *relative entropy* (or Kullback-Leibler divergence). This quantity measure the closeness (distance or *dissimilarity*) of two probability distributions, $p(x)$ and $q(x)$, over the same index set x [27]. Having this in mind, the relative entropy is defined as:

$$H(p(x)||q(x)) \equiv \sum_x p(x) \log \frac{p(x)}{q(x)} \equiv -H(X) - \sum_x p(x) \log q(x). \tag{2.16}$$

An important feature from this measure is that $H(p(x)||q(x))$ is strictly non-negative. That means $H(p(x)||q(x)) \geq 0$ with the equality achieved $\iff$ $p(x) = q(x) \forall x$. Eq. (2.16)

[3] For natural logarithm the units of entropy units are known as *nats*.

is often useful to define other entropy measures since they can be thought as special cases of this one.

2.4.3 Conditional entropy and mutual information

With the previous definitions stated above, one question that one could address is that if X and Y are two random variables, How is the information of X related to the information of Y? One could think that computing the correlation coefficient might be enough, but it happens to be a very limited measure of dependence due to its linearity profile. To answer the latter question, two concepts are needed: *conditional entropy* and the *mutual information* (We will pay special attention to this last one since its treatment is the core of this thesis). But before introducing them, it is convenient to define first the *joint entropy* of two random variables X and Y as

$$H(X,Y) = -\sum_{x,y} p(x,y) \log p(x,y), \tag{2.17}$$

where eq. (2.17) measures the total uncertainty of the pair of variables (X,Y) and $p(x,y)$ represents the *joint probability* mass function. An advantage of this definition is that it can be extended in any vector representation.

In the case where a variable is known say Y, then there are $H(Y)$ bits of information acquired from the pair (X,Y). Therefore, the remaining uncertainty of the pair (X,Y), is associated with the remaining lack of knowledge about the other variable X, regardless if Y is already known. For this reason, the entropy of X *conditional* on knowing Y is

$$H(X|Y) \equiv H(X,Y) - H(Y), \tag{2.18}$$

where $H(X|Y)$ in eq.(2.18) is known as *conditional entropy.*

This quantity brings to the light that both X and Y can have information in common. A natural question at this point is about a way to define a measure in which we can know the amount of information that one random contains about another. To answer that, let's suppose that we add the information content of X, $H(X)$, to the information in Y; the resulting common information between X and Y will be counted twice in the sum, while the information that is *not* common, will be counted just once. By subtracting off the joint information of the pair (X,Y) and $H(X,Y)$, we can define the *mutual information* of X and Y as

$$I(X,Y) = \sum_{x}\sum_{y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{2.19}$$

or alternatively,

$$I(X,Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y). \tag{2.20}$$

The signals that will be analyzed in this project contain information that is parameterized over time. For this reason, stochastic description of our system is a reasonable way to analyze data. Along this thesis we will discuss and apply this quantity to extract nonlinear features from data, since it is a different way of extracting information from a pair of random variables without any requirement of linearity. In the next section we will mention more about the continuous case.

Another immediate concept relevant in information theory is the *conditional mutual information* which is defined as the reduction of the uncertainty of the random variable X due to the knowledge of Y when Z is given [7]. We can express the *conditional mutual information* as

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z) = \sum_z p(z) \sum_x \sum_y p(x,y|z) \log\left(\frac{p(x,y|z)}{p(x|z)p(y|z)}\right). \quad (2.21)$$

2.4.4 Differential entropy measures

In the previous subsection we discussed some well known definitions of information measures for discrete random variables. In this section, we will introduce the concept of *differential entropy*, which considers the case of continuous random variables for the definition of entropy measures. A continuous approach allow us analyzing certain features of the implementation of a mutual information matrix for data based on a quantization theorem [4] .

In this case, let's consider X to be a random variable with a *cumulative distribution function (cdf)* $F(x) \equiv Pr(X \leq x)$. When the function $F(x)$ happen to be continuous, then we say X is a continuous random variable. As defined in (2.8), we define the *probability density function (pdf)* as $f(x) = F'(x)$ when such derivative exists. That being said, let's define the continuous entropy measures.

Definition (Support set)**.** We define the *support set* of X the set where $f(x) > 0$.

Definition (Differential entropy)**.** The differential entropy of a continuous random variable X with (*pdf*) $f(x)$ is

$$h(X) = -\int_S f(x) \log f(x) dx \quad (2.22)$$

where S is the support set of X, where $h(X)$ depends only on the *pdf*.

For the case of a set $X_1, \ldots, X_n$ of random variables with *pdf* $f(x_1, \ldots, x_n)$,

$$h(X_1, \ldots, X_n) = -\int f(x^n) \log f(x^n) dx^n \quad (2.23)$$

In a similar fashion, it is possible to extend the previous definition to several variables and therefore the continuous version of the entropy measures previously described [7].

Definition (Conditional differential entropy)**.** For a pair of continuous random variables X, Y that have a joint density function $f(x, y)$, the conditional differential entropy is

$$h(X|Y) = -\int f(x,y) \log[f(x|y)] dxdy = h(X,Y) - h(Y) \quad (2.24)$$

[4] Extra remarks are also mentioned in Appendix A.

Definition (Relative entropy)**.** In the case of 2 *pdf*'s $f(x)$ and $g(x)$ for a random variable X, we can define the relative entropy as

$$D(f||g) = \int f \log \frac{f(x)}{g(x)} dx \tag{2.25}$$

where $D(f||g) < \infty$ if the support of f is contained in the support of g.

Definition (Mutual information)**.** For the case of 2 continuous random variables with joint density $f(x, y)$. the mutual information is defines as

$$I(X, Y) = \int f(x, y) \log \frac{f(x, y)}{f(x)f(y)} dxdy \tag{2.26}$$

Definition (General version mutual information)**.** Let $\mathcal{X}$ be the smallest interval that contains all the values of the random variable X (i.e. the *range* of X). A partition $\mathcal{P}$ of $\mathcal{X}$ is a finite collection of disjoint sets P_i that cover $\mathcal{X}$ such that $\mathcal{X} = \bigcup_i P_i$. If X and Y are random variables with partitions $\mathcal{P}$, $\mathcal{Q}$, then the mutual information is given by

$$I(X; Y) = \sup_{\mathcal{P}, \mathcal{Q}} I([X]_{\mathcal{P}}; [Y]_{\mathcal{Q}}) \tag{2.27}$$

where the supremum is taken over all finite partitions $\mathcal{P}$ and $\mathcal{Q}$ [7].

It exist a relation between the definitions of discrete entropy measures and the ones from the differential entropy fashion. The demonstration assumes a similar approach as the construction of the Riemann integral. This method is known in the literature as *quantization* of a continuous random variable and can be summarized in the following theorem

Theorem 2.4.1. *Let X be a continuous random variable with a pdf $f(x)$ that is Riemann integrable. If we divide the range of X into bins of length Δ and consider X^{Δ} as the quantized random variable defined by $X^{\Delta} = x_i$ if $X \in [i\Delta, (i+1)\Delta]$, then*

$$H(X^{\Delta}) + \log \Delta \to h(f) = h(X) \tag{2.28}$$

when $\Delta \to 0$.

The previous result will come handy when we examine the bias of the histogram-based estimation of the mutual information matrix. Such examination will be discussed more thoroughly in the following chapters of this text, when we apply entropy measures to measure non linearity in the data.

2.5 Non-linearity using mutual information

Previously introduced in sections above, we saw that Pearson correlation coefficient (2.12) is one of the most used measures that reflects only the linear dependence between the two random variables. As we recall from its definition, such coefficient does not provide any nonlinear dependence since such dependence is more subtle and requires other type of analysis. To exemplify this subtlety, we can think of the correlation dependencies for the known

Anscombe's Quartet. Such quartet is known for having four data sets with almost identical statistical features but with different distributions when plotted (See Apendix C, Table C.1).

In Figure (C.1) we can see the distributions of the data sets from Table (C.1) and notice that they all have the same mean (μ), standard deviations, Pearson correlation and coefficient of determination (R^2). Moreover, the data sets have pretty much the same linear fit following the line equation $y = 0.5x + 3.00$. As mentioned before, nonlinear dependence is not reflected by any of these measures. On the other hand, by calculating the mutual information for each set, it is evident that the value obtained is different for most of the cases. This is an indication that mutual information is a useful measure for examine nonlinear dependence.

If we recall from the previous section, the concept of mutual information was introduced as a measure of the reduction of uncertainty between random variables by knowing another. In this context, mutual information can be interpreted as a measure of total dependence. The latter is evident from eq. (2.19) and (2.20) when 2 random variables have a joint entropy less than the sum of their individual Shannon entropies because otherwise, such random variables would be independent form each other.

As we will see in the next chapter, the way mutual information is calculated in this thesis is based on probabilities by using a binning method for the computation of the joint probability distribution between a pair of random variables. Typically the binning method and nearest neighbor measurements are the most common methods for calculating mutual information.

An important reminder to the reader is that the binning method described in this thesis is completely non-parametric. Meaning that in our mutual information computation for a given pair of random variables, there is no inherent parameter along the computation that influence the outcome. This point is crucial because it allows us to apply the framework described in [38] for an *estimation of the linear component of mutual information.* The goal of this estimation can be understood in the following way: After calculating the mutual information from the original data, we would like to remove the linear component of dependence given by the Pearson correlation. Then, recalculate the mutual information on the new data set and lastly, compare the mutual information from both data sets. The technique can be described in four steps as follows:

- *Pearson coefficient and least-squares regression*

  In this step, we take advantage on the Pearson coefficient and the least-squares regression, to extract the linear features of the random variables. If we consider the pair (X, Y) as the relevant random variables, it is possible to plot them and obtain a regression function for Y given X [5]. In particular, if we designate $\hat{Y}$ as the fitted values of Y, after we conduct a linear regression fit, we can calculate the difference between the original dependent variable Y and the fitted values as

  $$z_i = Y_i - \hat{Y}_i \tag{2.29}$$

  where z_i are the residuals and $i \in 1, \ldots, N$ [6]. Notice that the linear regression effectuated to Y is justified due to the fact that we want to extract only the linear behavior

[5] Here we are considering Y as the dependent variable and X as the independent random variable.

[6] We are using this label $i \in 1, \ldots, N$ to match the notation in Chapter 4.

of the data. In this way, considering the difference between Y and $\hat{Y}$ will leave the residuals z_i with other dependencies that are strictly nonlinear. For the pair (X, Y), the Pearson coefficient informs us about the linear correlation between the variables, and the coefficient of determination R^2 informs us about how precise the linear fit was. In this way, we can have a better understanding of the dependencies in z_i.

- *Mutual information calculation using a binning method*

  The next step is to calculate the mutual information $I(X, Y)$. For this, we do the computation following the description in Chapter 4 where first we compute the 2D histogram to create the joint probability matrix for each entry and then use expression (2.19) to obtain the result [7].

- *Analysis of residuals*

  As mentioned before, the residuals obtained in eq.(2.29) represent a quantitative way of nonlinear dependence between the random variables X and Y because its magnitude indicate the distance of the point Y_i from the ideal linear behaviour $\hat{Y_i}$. Given the way the residuals are defined, this suggests that the relationship between them and the independent random variable X should have a Pearson correlation equal to zero ($\rho(X, z) = 0$) because in principle, only nonlinear dependencies survive with z [8].

  To compare the mutual information between X and z and the pair X and Y it is important to perform such difference at the level of the joint entropy between each pair of variables [38]. In this sense, what we want to do is to perform a mutual information comparison between 2 pair of variables (X, Y) and (X, Y') where Y and Y' have the same marginal probability density function (*pdf*) and therefore, the same Shannon entropy $H(Y) = H(Y')$. Following this idea, it is immediate to see that this ensures the difference of the mutual information to be exclusively present at the level of the joint entropies ($H(X, Y)$ and $H(X, Y')$). Notice that if the Shannon entropies $H(X)$ and $H(z)$ are different then, it will be unclear whether the difference between $I(X, Y)$ and $I(X, z)$ is due to $H(z)$ or $H(X, z)$ (eq.2.20). To calculate Y', Smith's technique [38] uses the van der Waerden normal transform (also known as quantile normalization) to map the cumulative distribution function (*cdf*) [9] onto the *cdf* of a normal distribution. We use this technique to map the (*cdf*) of the residuals and match the *pdf* to the dependent variable X to achieve the same entropy. This will ensure the reduction of the mutual information at the level of the joint entropies $H(X, Y)$ and $H(X, Y')$. The quantile transform of Y' is

  $$Y' = F_y^{-1}(G(z)) \tag{2.30}$$

  where F_y is the *cdf* of the dependent variable Y, F_y^{-1} is the quantile function [10] of Y

[7] The amount of bins used in the computation are given by Sturge's rule: $\log_2 N + 1$, where N represents the time steps of our signals.

[8] In practice we will obtain that this correlation is sufficiently small ($\sim 10^{-10}$) so the claim holds for our numerical analysis.

[9] The cumulative distribution function (cdf) of a discrete random variable X is $F_X(x) \equiv P(X \leq x)$. For the continuous case we make use of the *pdf* f_X so that $F_X(x) = \int_{-\infty}^{x} f_X(t)dt$.

[10] Also known as *inverse cumulative distribution function*.

and $G(z)$ is the *cdf* of the residuals. In this way, Y and Y' have the same *pdf* and therefore, the same Shannon entropy. To perform this quantile transform in Python, first we calculate the *cdf* of the residuals using a uniform distribution. Next, we make use of the `sklearn.preprocessing` package and the `QuantileTransformer` library to transform the data using a normal distribution and concatenate the information to obtain the array Y' needed for the calculation of $I(X, Y')$.

- *Mutual information of X and Y'*

  The last step is to calculate $I'(X, Y')$, which is the mutual information of X and Y' but with the linear dependence subtracted. Notice that an important property of the mutual information comes up to light. Due to the property that Shannon entropy is non increased under functions $H(X) \geq H(g(X))$ for $g(X)$ and arbitrary function, then $I(X, Y) \geq I(h(X), k(Y))$ for $h(X), k(Y)$ arbitrary functions. Therefore, we can conclude that $I \geq I'$ reaching the equality iff $h(X)$ and $k(Y)$ are invertible. Thus, in the case where linear dependence is more manifest, this will result on $I' \sim 0$. On the other hand, if the linear dependence is very small (or dominated by non linear behaviour), then $I \sim I'$. This allows to define a coefficient (Λ) [38] that informs about the global proportion of the linear dependence between X and Y

  $$\Lambda = 1 - \frac{I'}{I} \tag{2.31}$$

  Based on the definition in eq. (2.31), $\Lambda \in [0, 1]$ and when $\Lambda = 0$ it means that the random variables have completely nonlinear dependence between them when $I > 0$. In the other extreme case where $\Lambda = 1$, then we can ensure that the relationship is entirely linear and that the Pearson coefficient is enough to describe the correlation.

  In this fashion, the Λ coefficient improves the understanding of what portion of the total dependence between a pair of random variables is linear. Since a direct correspondence between the mutual information and the Pearson correlation is not yet resolved other that for normal distributions (eq. A.3), this calculation of Λ indirectly answers such correspondence.

To conclude this section, let's just introduce some useful statistical learning techniques that will be used for analysing synthetic and fNIRS data. For the most part of the text, we will discuss *unsupervised learning* techniques applied to time series to observe how much information they share after undergone a mixture in a certain fashion (synthetically or directly from the experiment).

2.6 Topics in unsupervised learning

Unsupervised learning is a category of techniques that approach problems where there is no associated response measurement after conducting an observation [16]. In this sense, the intention is to infer probabilistic properties of a random vector, by providing accurate answers of the observation without the help of a supervisor [11]. In the analysis of fNIRS data, there are some of these techniques that allow to extract certain features of the data that are relevant for a pre-processing analysis. In this thesis we will consider two relevant algorithms: *principal* and *independent component analysis*.

2.6.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a statistical technique widely used to examine interrelations among a set of variables in order to identify a certain structure of those variables. The method uses *orthogonal linear transformations* to express correlated variables into linearly uncorrelated variables named as *principal components* [17] [11]. It converts high dimensional data (as the one used in this report) to low dimension data, scaling the features where most of the information of the dataset is captured. Such features are the directions where the highest variance of the data occurs. The method projects the dataset onto a different subspace where the data is represented well enough.

This is done by finding and ranking all the eigenvalues and eigenvectors of a covariance matrix. This is useful because high-dimensional data (with p features) may have nearly all their variation in a small number of dimensions k, i.e. in the subspace spanned by the eigenvectors of the covariance matrix that have the k largest eigenvalues. If we project the original data into this subspace, we can have a dimension reduction (from p to k) with hopefully little loss of information [11].

2.6.2 Independent Component Analysis (ICA)

Independent component analysis (ICA) is a technique for estimating independent source signals from a set of recordings in which the source signals were mixed together in unknown ratios. An important feature this technique has is the assumption of *statistical independence* and non-Gaussianity behaviour on the source signals. Often when one performs a statistical analysis a common assumption is the normally distributed behavior that a particular data set has. This so called Gaussian feature is important for techniques as PCA as described in the section above. However, there are many measurements that aren't necessarily normally distributed. An example of the latter are the electrical signals from different brain regions or measurements from brain activity like the ones concerning this thesis [25], [22]. When non-Gaussianity is assumed, ICA could distinguish individual signals after they are submitted to a mixture process involving multiple sources.

An important remark is that unlike PCA, ICA does not maintain a hierarchy for the components that result of the process. That means the method itself does not consider a magnitude order with each component.

To understand ICA, one can think of the so-called *cocktail party problem* which is an example of *blind source separation* (BSS) problem [25] in which we do not know anything about the source of the signals that one is registering. In this case, if we consider that in a reunion each person has a microphone X_j that collects a mixture of signals form independent sources S_l (in this case people at the party), ICA is able to distinguish the voice of each source from the linear combinations of their voices exploiting the independence and non-Gaussianity of the sources itself [11] Figure 2.3.

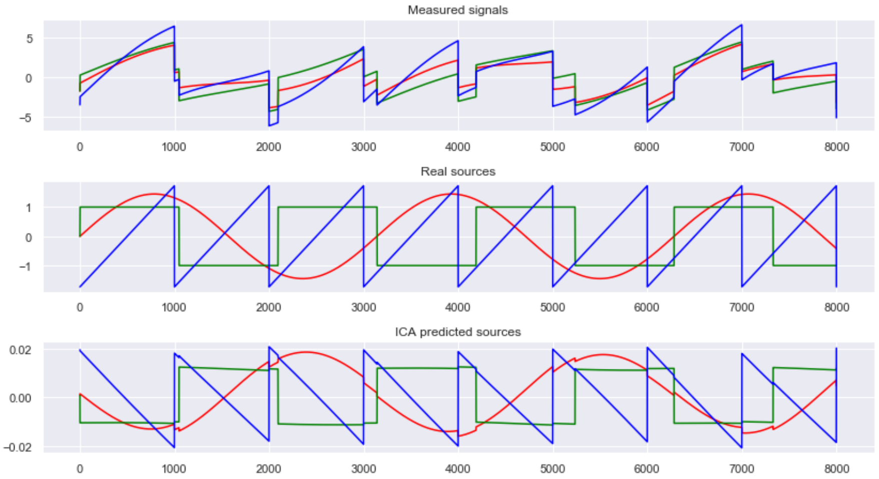


Figure 2.3: *ICA on artificial time-series data. Here we are observing 3 source signals measured at* 8000 *uniformly spaced time points. The upper panel shows the signals collected by the microphones and the lower panels show the real sources form each individual speaker and the independent component solutions.*

If we consider an observed vector $\vec{x} = (X_1, \dots X_n)^T$ whose n components are linear combinations of independent elements of a random vector $\vec{s} = (S_1, \dots S_n)^T$ given by

$$\vec{x} = \mathbb{A}\vec{s} \tag{2.32}$$

where $\mathbb{A}$ is a $n \times n$ mixing matrix, then the purpose of ICA is to find an unmixing matrix $\mathbb{W}$ such that it will retrieve a vector $\vec{y}$ that is the best approximation to $\vec{s}$ (i.e. $\vec{y} = \mathbb{W}\vec{x} \approx \vec{s}$). Notice that in ICA, the time index t is not considered and it is assumed that each mixture X_j and each independent component S_k are random variables. The vector-matrix form written in (2.32) provides an advantage for brain signals since $\vec{x}$ is the fNIRS signal matrix (from the observations registered in the experiment), $\mathbb{A}$ is the basis matrix spanning a subset of the observation space [18] and $\vec{s}$ is the random vector with component signals $S_1 \dots S_n$. The matrix $\mathbb{A}$ is a square full rank matrix with inverse $\mathbb{W}$.

To proceed with the use of ICA some assumptions must be considered:

- As we have said before, *statistical independence* between every source S_i form the sources vector $\vec{s}$ must be assumed. This feature opens the possibility for different ways to measure independence that result in slightly different unmixing matrices [22]
- The mixing signal matrix $\mathbb{A}$ must be full rank
- The only source of stochasticity is the source vector $\vec{s}$
- The data is centered (i.e it has zero mean). Depending on the algorithm, the observation vector $\vec{x}$ must be whitened i.e., it must be linearly transformed so the correlation matrix retrieves $\mathbb{E}[\vec{x}\vec{x}^T] = \mathbb{I}$
- The source signals S_i must not have a Gaussian probability density function (pdf).

There are several prescriptions for introducing measures of non-Gaussianity (i.e. objective functions for ICA estimation). To mention some, we can name the `Infomax` (or maximizing mutual information), the *minimization of mutual information and maximization of non-Gaussanity.* According to [14], these methods are equivalent to a *maximum likelihood estimation.* One of the most common ways to estimate square mixing matrices is the `fastICA` algorithm introduced in Ref.[14]. This method is the one used in this thesis given its efficiency and performance [22].

The `fastICA` algorithm given by [14], [22]is the following:

- Choose an initial weight vector $\mathbf{w_i}$
- Let $\mathbf{w_i}^+ = \mathbb{E}(g'(\mathbf{w_i^T x}))\mathbf{w_i} - \mathbb{E}(\mathbf{x}g(\mathbf{w_i^T x}))$
- $\dfrac{\mathbf{w_i^+}}{\|\mathbf{w_i^+}\|}$
- For $i = 1$ go to step 7. Else, continue with step 5
- $\mathbf{w_i^+} = \mathbf{w_i} - \sum_{j=1}^{i-1} \mathbf{w_i^T w_j w_j}$
- $\dfrac{\mathbf{w_i^+}}{\|\mathbf{w_i^+}\|}$
- If not converged, go back to step 2. else go back to step 1 with $i = i + 1$ until all components are extracted.

where the $\mathbf{w_i}$'s are the column-vector of the matrix $\mathbb{W}$ and $\mathbf{w_i^+}$is the variable used to calculate $\mathbf{w_i}$ in the iteration loop, $g(.)$ is a non-quadratic function usually $g(.) = Tanh(.)$ and $g'(.)$ its derivative. The algorithm above is later used in Chapter 5 when we perform ICA to the fNIRS data. To apply PCA and ICA algorithms to our data, we will make use of `sklearn.decomposition` routine for computing it in Python.

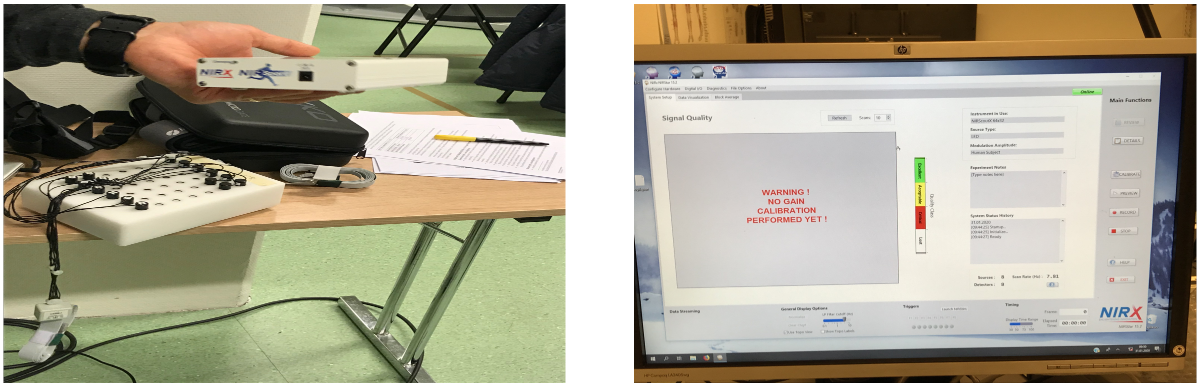
Chapter 3

State of the art and methodology

In this chapter, we discuss the standard approach to fNIRS. We start by describing the equipment and software used to collect the data. Subsequently, we present the usual fNIRS data acquisition followed by the typical pre-processing of such data. Lastly, we introduce the methodology and the type of data that will be used for the analysis.

3.1 fNIRS technology: equipment, data, and software

For this thesis, we will be using the data sets generated by the fNIRS equipment at Oslo Metropolitan University (Figures 2.1 and 3.1). This data set is similar to the one in [36], where the brain activity lectures were collected using a portable continuous-wave system named NIRSport, a device that works with wavelengths of 760nm and 850nm. The portability of this device allows the possibility of broader physical tasks, for example, experiments that involve measurements to people with lower limb amputation (LLA).



WARNING !
NO GAIN
CALIBRATION
PERFORMED YET !

(a) NIRSport device

(b) Interface of the software

Figure 3.1: *On the left-hand side, we can see the NIRSport device used to collect the data. In (b), one can see an image of the interface used for the data collection.*

The data that we will be using in this thesis consists of a collection of lectures taken from the prefrontal cortex of a subject doing a series of physical tasks. In particular, work with data sets from LLA individuals that perform three physical tasks: walk along an 8-figure path, walk with a trial and glasses of water, and finally, walking on an unstable terrain. Such measurements inform about the concentrations of oxygenated (oxyHb) and deoxygenated hemoglobin (deoxyHb), that are later use to create functional connectivity networks and explore the types of correlation using the measures described in Chapter 2. All these lectures were collected from the fNIRS equipment described above (Figure 3.1), and then saved in a `.txt`-file.

Later on, we analyze these files to extract statistical features by using Python 3 as the main programming language for this thesis. In particular, we will be using Python libraries and other resources (such as the Jupyter notebooks) to generate synthetic data, the networks, and other relevant diagrams to be presented in more detail in Chapters 4 and 5.

3.2 Acquiring fNIRS data with a NIRx machine

When collecting data from brain activity, it is important to understand how the data acquisition equipment works [1]. Is because of this reason that we provide a summary of data collection for fNIRS experiments. Moreover, we describe the traditional setting for a new study and the machine's calibration for our dataset. Specifically, we focus our attention on the NIRx equipment. Hardware used for this thesis.

3.2.1 Hardware and software for fNIRS acquisition

The machine used to collect the data is the NIRSport I imaging system (Figure 3.1b), and operates with 32 sources (colored in red) and 32 detectors (colored in blue). Figure 2.1 shows the setting of the machine and the location of the electrodes placed on a subject's head. The detector tips used are frequently referred to as the *standard detector tips*, and each of them is labeled with a number. Moreover, the device uses delicate fiber-optic wires that provide efficient signal transmission from the electrodes to the machine.

When using the NIRSport I equipment, the information containing the lectures of brain activity is later passed to a software named NIRStar for further analysis and processing of data. This software typically runs on a Windows PC (see Figure 3.1a)[2]. As illustrated in Figure 2.1c, we can see attached optodes to the participant's head by using textile polyester caps adjustable to different sizes (typically $54, 56, 58$ and 60cm diameter). The caps contain holes specifically placed according to international standard head locations, where an array of detectors and light emitters are placed.

[1] The standard analysis of fNIRS data is by using the nirsLAB v201706 software (`https://www.nitrc.org/projects/fnirs_downstate/`).

[2] As a side note to the reader, it is worth mentioning the use of another software for analyzing fNIRS data, named Homer 3. This software uses a MATLAB environment and works as an alternative to the NIRStar option for Mac OS users. Homer 3 was only used to visualize the physical location of the optodes and the virtual channels as dispalyed in Figure (5.2).

3.2.2 Setting for a new study: program montage

To conduct a new data collection using the device described above, the first step is to arrange the cap's configuration also known as *probe montage.* The latter is performed by knowing the number of sources and detectors available per person (i.e., according to the individual's head size). In Figure 2.1c we can see an example of the probes' montage. Notice that sources and detectors need to alternate so a standard spacing between each one of them is maintained. The connection between source-detector is what we will define as a *virtual channel* [3]. In practice, there are many standard layouts that one can use to measure activity from different parts of the brain such as the prefrontal, premotoric, motoric or visual cortex; but for our experimental data, we will focus the attention only on the prefrontal cortex region.

To set a new experiment, it is necessary to tell the acquisition software what the montage is going to be. This can be seen in the manual [40] for further details since it is a standardized procedure and its description is beyond the scope of this thesis. The NIRStar software can create a new montage that includes all the relevant files the experiment needs. At this point, the participant's head must also be measured to ensure that the size of the cap and the electrodes are located correctly along with the head [4].

3.2.3 Calibration and data collection

Before beginning an experiment, the system must be calibrated. The latter is done by opening the NIRStar software and ensuring that the montage is correct. The software has different labels and bottoms to choose the right calibration option. At this stage, it is possible to see the channels (source-detector numbers) activation by colors. The coloration denotes the quality of the signal that is coming through the electrodes. There are 4 labels: *white, red, yellow,* and *green* labeling from the worst to the best-received signal respectively. When the software detects low-quality signals, adjustments need to be made. These include checking the connection of the electrodes and removing the grommet cap and push out some excess hair to avoid light cut out in the electrodes. This procedure can be done many times until the software receives good quality signals. After mapping the regions with good calibration, it is possible to *record* the data and has a visualization of it [40]. An important note is that warming up the machine before the measurements will reduce the signal drifting. Once the experiment is completed, the recording is terminated causing the NIRStar to automatically save the data. This will create a folder with the format of *"DATE-NUMBER"*, where the measurements are saved. The data sets explored in Chapter 5, were collected based on these settings.

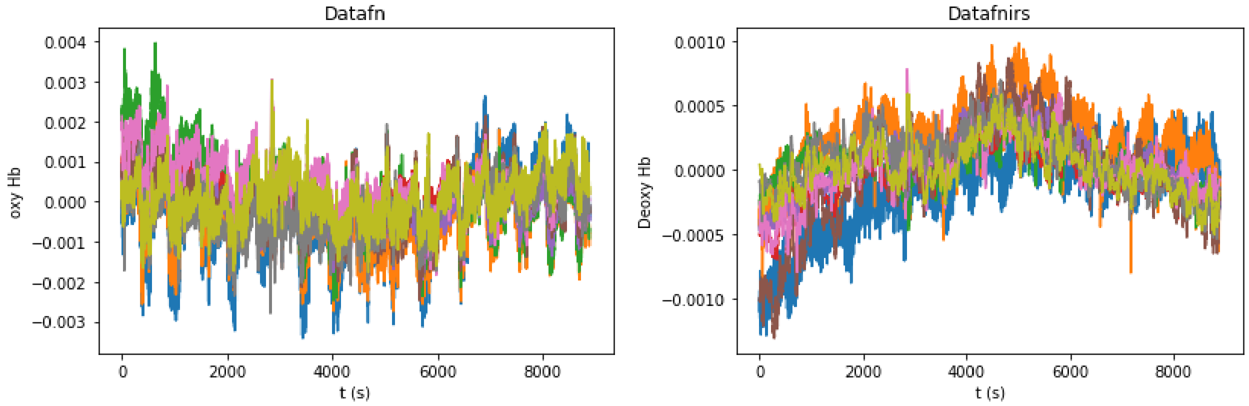
3.3 Standard pre-processing for raw fNIRS data

Once the files are saved after the experiment, a pre-processing treatment is performed to the raw fNIRS signals to extract relevant features from the experiment. This raw data

[3] We will use this nomenclature in the rest of the text for a pair source-detector. As we will see in Chapter 4, we will name *signals* to the analog of these virtual channels for synthetic data.

[4] The optic fiber cables connected to the NIRSport I device are back over and behind the cap, so the participant's face is clear. The stabilization of the cables is important to avoid loose connections.

measures light intensity, and the amplitude of the signals is measured in voltage units (V). A visualization of the raw data can be obtained directly from the NIRStar software Figure 3.2.



(a) (b)

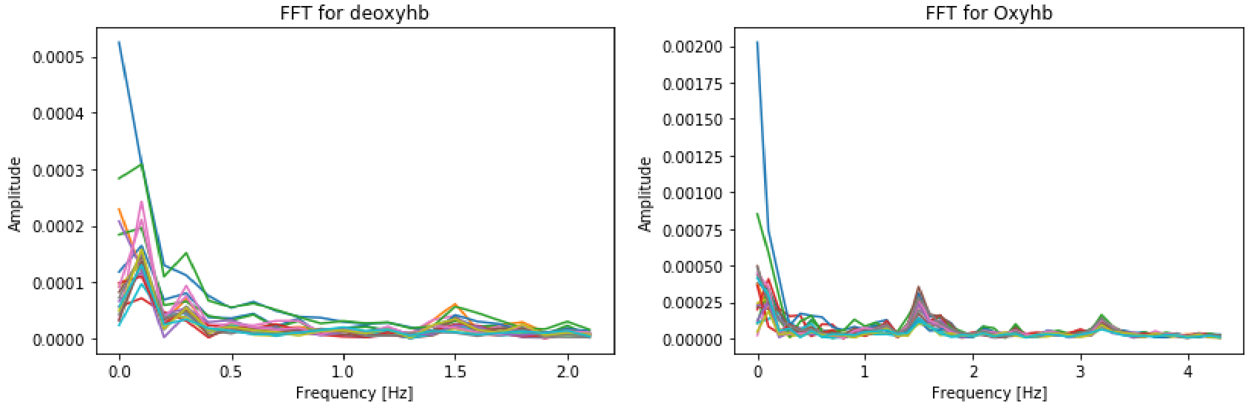
Figure 3.2: *Time series of the fNIRS data from the files available. In this case the time series are describing lectures of brain activity for people with amputated limbs. The plot displayed in a) is the time series for Oxy. Hb (HbO_2) from the* `.txt` *file. The plot in b) is the time series for deoxy. Hb from the* `.txt` *file.*

The file with the signals is divided into columns each of which corresponds to one virtual channel lecture (i.e. a source-detector pair). The columns correspond to the time series in a certain acquisition frequency measured in Hertz (Hz). It is important to mention that when looking at an fNIRS signal, it is not just brain activity the one that is being recorded. The signals contain noises from the externals sources as well as from other factors inherited from the experiment itself. For instance, cortical signal is a small percentage of the total raw signal that has been recorded. Many times a way to observe noise on data that comes directly from the participant's physiology is by performing a fast Fourier transformation (FFT). Figure 3.3 [5].

3.3.1 Physiological noise

One special type of noise the raw data contains is the one coming from *physiological contributions*, endemic to the human body. For instance, cardiac signal, respiration, and Mayer waves [31] are the most common contributions one can encounter. The first two have an impact that depends on the physical condition and age of the participant and can synchronize with repeated stimuli presentations from the experiment. The Mayer waves are fluctuation in blood pressure that occurs at a certain frequency, approx 0.1 Hz; and are related to the position of the participant. They can vary when the participant is sitting, standing, or laying down [19].

[5] Normally, the peaks observed in Figure 3.3 are acknowledged as physiological noise such as cardiac pulse, respiration and other factors. However, for the purposes of this report, we are just illustrating the FFT of our relevant data sets.



(a) (b)

Figure 3.3: *Fast Fourier Transformation (FFT) of the signals from our data sets. Here we have the FFT's for the 19 virtual channels. The bumps observed in the plots are the physiological noises caused by cardiac pulse, respiration, etc. a) is for the deoxygenated Hb, and b) is for the Oxygenated Hb.*

The removal of the physiological noise from the data is carried out in two stages. The first one is during the data collection. Several experiments use the General Linear Model (GLM) approach [31], so in this case, it is useful to use multiple presentations of the task and space the stimuli in intervals of time. Also, the use of shorter channel regressions can help to reduce physiological noise. Depending on the experiment, some people also use other peripherical physiological measures like respiration monitoring and heart rate monitoring to reduce these contributions so that one condition (or stimuli) won't reliably obtain changes in breathing and heart rate.

The second is during the pre-processing stage. Here, there are different algorithms for addressing the reduction of data contamination. To mention some of the most used we can name Frequency filtering, PCA [6], and Adaptive and Kalman filtering. Frequency filtering is useful because one can specify which of these frequencies one wants to keep in the signal and which ones should be removed. For most GLM stimuli the range of work is 0.1-0.3 Hz [31].

For physiological noise, the PCA filter creates new dimensions for the data to fall on according to the variance (Chapter 2 and [17]). The method identifies those dimensions which are responsible for the most variation in the data and from there one can choose to discard the variability due to those dimensions. The utility of this analysis for fNIRS data is that some oscillations like Mayer waves can be removed if we know the variation of the signal they create.

3.3.2 Non-physiological noise

The non-physiological contributions are derived from the measurement itself. This kind can be reduced due to controlled environment conditions when the experiment is been conducted

[6] This one used later in the text for analyzing correlations.

but cannot be completely mitigated. For this kind of noise, we can name some techniques to reduce such contributions. At the level of data collection,

- The machine drift can be reduced by using LEDs and by turning on the machine 15 min before the collection of data.
- Motion artifacts: reduce unnecessary participant motion, stabilize the wires properly.
- Measurement noise: ensure good contact with the electrode to the scalp of the participant, block the ambient light, etc.

During the pre-processing stage, we can use again, frequency filtering for the machine drift, PCA, Spline, correlation-based signal improvement (CBSI) for motion artifacts, and manual removal of motion stimuli. An important thing to notice is that since the motion artifacts can have a strong contribution to the data collection, a PCA filter is one of the most used tools for dealing with this in the pre-processing pipeline. Another technique commonly used but not as popular as PCA is the CBSI. The latter is based on the knowledge that oxygenated Hb and deoxygenated Hb are strongly anti-correlated during cortical activation [3] yet, they have a correlated motion artifact spike or discontinuity. In plain terms, this method recreates the *true* signal by forcing a negative correlation [7]. It assumes the negative correlation under a certain assumption. This procedure is done by the software but it could be problematic in case there are many motion artifacts.

The pre-processing stage is crucial for extracting relevant features of the brain and as we have seen, it involves different techniques that allow scientists to remove undesired noises from data. The set of tools to analyze fNIRS data has grown rapidly in recent years that standard techniques are easy and, in some cases, automatically implemented by the software itself. The study described in this thesis intends to explore other ways of analysis based on linear and nonlinear correlation measures to have a different perspective of how interconnected the brain is. This discussion brings more into context the kind of results that we will be analyzing more thoroughly later in Chapter 5 and opens the door to discuss how to distinguish linearity dependencies in our fNIRS data. In particular, in the next section we will introduce a method that involves mutual information to answer this inquiry.

3.4 Overview of the methodology

We divide the methodology into four important steps that will allow us understand the purpose of this thesis. First, we will generate multivariable synthetic signals sets where we test some of the correlation measures discussed in Chapter 2, namely linear correlations using covariance matrix and nonlinear by using mutual information. Later on, we will study the behaviour between numerical and analytical approaches, and test their response when increasing the number of parameters involved in the computation. After that, we will see what information can be validated with the synthetic data so it can be applied to the framework of fNIRS data. At this stage we will also compare the results with other statistical

[7] In this thesis we will use another approach for this by the use of independent component analysis (ICA) because it intrinsically uses the mutual information.

learning techniques. Finally, we will interpret the results via functional connectivity networks similar to Figure 3.4, and assure positive direct relations between functional connectivity network and the data of the participants. The stages are:

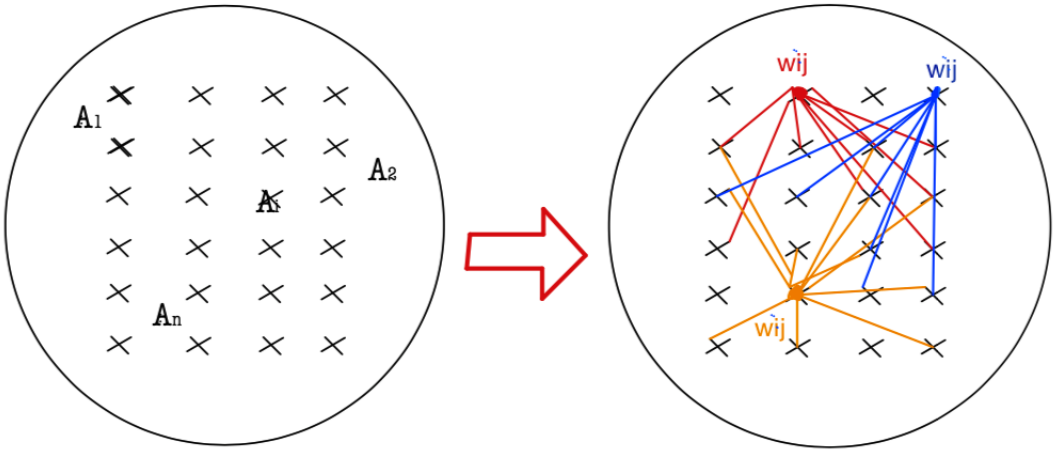


Figure 3.4: *A framework of the Activation Channels. In this figure, we show the Activation channels from the subjects as the mapping area of the activation channels that measures brain activity. The activation channel denoted as A_i represents the measure of Oxygenated and deoxygenated blood in a lecture after the activity has been performed. Once the signals are collected, the idea is to model the activation channels via correlations (linear and non-linear) to map different regions of the brain and to draw a connectivity network as in the representation in the RHS. Here, w_{ij} represents the correlation families of interest.*

1. **Set of synthetic multivariable signals with an imposed correlation matrix**

   We begin by generating synthetic data signals from coupled stochastic processes to compare numerical and analytical correlation matrices, based on an entry-wise matrix norm of the difference. This tool will allow us to explore how the percentage error between the two matrices behave when we increase the number of time-steps and the number of signals involved in a simulation. Notice that for a set of *n-signals*, determining a correlation measure between them is not an easy task since such labor cannot be done in general for any kind of stochastic process. This step will be implemented by writing a Python code where it is possible to calculate correlation matrices from an array of numbers. In Python, we can work with arrays of numbers. We can have vectors and matrices describing the signals and calculate the *correlation* between them.

2. **Implementation of mutual information for synthetic signals**

   The next stage is to implement the mutual information using a binning method. More specifically, we create a joint probability distribution between a pair of random variables using a 2D histogram from which we will compute the mutual and normalized mutual information. As in the previous step, we will test these measures by exploring the behavior of the matrix norm when the number of signals and time-steps are increased. Finally, we compare the binning method described above with the analytical counterpart i.e., when we use normal distributions and compare with the continuous case.

3. **Application of the computational framework to fNIRS data**

   With the code prepared for synthetic data, it is possible to simulate the signals from a region around an individual's head with *n-points*. After completing the step above and achieving reasonable results from the synthetic data set, we proceed to use the real data to test the analytical methods and techniques. The latter has measurements from oxygenated and deoxygenated blood responses from an experiment previously done by OsloMet. The collection of this data consists of brain activity signals using fNIRS technology from subjects with a lower limb amputation (LLA) walking in an 8 figure path, along an unstable terrain, and with a trail of glasses of water. The idea here is to extract information on how the brain is responding to a task and to create a functional connectivity network with the information collected. The main work at this stage is to classify important information and visualize it in different ways.

4. **Interpretation of the extracted functional connectivity networks**

   For this stage, we study the results obtained from applying the different correlation techniques to the fNIRS data set and compare the results with unsupervised learning techniques such as PCA, ICA, and hierarchical clusters. In addition to that, we also explore the meaning of the visualization tools that have a big role in the data analysis in creating the connectivity network.

To end this chapter let's just recapitulate a little. So far, we have described the methodology for measuring brain activity using fNIRS data specifying relevant features for the data sets concerning this thesis. Also, we have introduced a recent method to use nonlinear measures to quantify the linear dependencies by pairs of random variables. The next step is to consolidate these ideas more explicitly when we describe the treatment of the data in the next chapters.

Chapter 4

Synthetic data for correlations and mutual information

In this chapter, we explain the logic behind the generation of synthetic data using stochastic processes. The assumptions and hypotheses for the analysis and creation of the code are described here. By implementing and using a synthetic data set we can observe the behavior of the linear and nonlinear correlation measures and test their reliability.

Once we have implemented and examined the measures for the synthetic data set, we proceed and apply the formalism to fNIRS data in a way we can interpret the outcome both qualitative and quantitative. This last part will help to understand the discussion section in Chapter 5.

4.1 Generating signals from coupled stochastic processes

From the previous chapter, we know that in an fNIRS experiment, the relevant information involves lectures of oxygenated and deoxygenated hemoglobin from the brain when a subject performs a specific activity. Such measurements are collected in files, as time series for each of the brain signals involved in the experiment. The files that we will consider in the analysis of this thesis are measurements of light amplitude (for oxy Hb and deoxy Hb) collected at a specific interval of time, with a fixed number of *virtual channels* (a pair of source-detector) given by the experiment. Therefore, the purpose of generating synthetic data is to simulate a possible experiment where we can analyze the linear and nonlinear correlation measures verify them, and apply them to the real data set. For the nonlinear correlation measures, the idea is to examine its behavior when certain parameters are changed. Namely, the length of the time series and the number of signals involved.

The advantage of this procedure is that the `.txt` data files (Figure 3.2) from the experiment, are basically a matrix with lectures of each *virtual channel* therefore, we use this feature to describe our *signals* in the form of an *arrays* in our Python code.

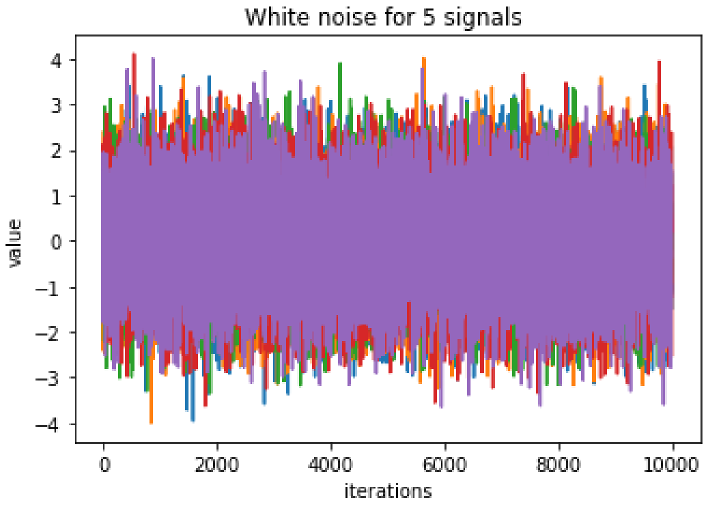


Figure 4.1: *White noise for 5 signals with* $\mu_{\vec{w}^{(i)}} = 0$ *and* $\sigma_{\vec{w}^{(i)}} = 1$ *for* $i = 1, 2, 3, 4, 5$. *Here we can see the time series generated by 5 Wiener processes.*

4.1.1 The use of stochastic processes

The idea of how we generate synthetic data is the following: because the set of *signals* recorded from the virtual channels on the cap contain the lectures of oxygenated (or deoxy-Hb) hemoglobin from a region of the brain (Figure 3.4) at a certain time-step, we can consider this discretization of information as arrays in the code. In particular, we can create our testing signals using the increments of Wiener processes. We begin by generating M vectors $\vec{w}^{(m)}$ $(m \in 1, \ldots, M)$ with entries randomly generated according to a normal distribution $N(0, 1)$. These $\vec{w}^{(m)}$ are all uncorrelated series of Wiener increments and all of them have dimension equal to N (i.e., dim $\vec{w}^{(i)} = N \ \forall i$).

To prepare a set of vectors $\vec{X}^{(m)}$ $(m \in 1, \ldots, M)$ correlated with each other, we make use of an auxiliary matrix $\mathbb{B}$.The entries of this matrix (written as B_{ln}) are also randomly generated by a normal distribution such that

$$\vec{X}^{(m)} = \mathbb{B}\vec{w}^{(m)} \tag{4.1}$$

with $m \in 1, \ldots, M$ and the time steps represented by the N entries of $\vec{X}$ [1]. See Figures 4.2 and 4.1.

Given the way we are constructing these $X^{(m)}$'s, we can calculate the correlation matrix as in eq.(2.14). Notice that this matrix size is $M \times M$ because we are considering M-signals. This particular correlation matrix of the M *signals* is what we will refer to as the *numerical correlation matrix*. The *analytical (or true) correlation matrix* of the process is actually inherited from $\mathbb{B}$ itself. To exhibit this last statement more clearly, we need to make use of the expressions (2.6) and (2.11) to justify this claim.

[1] To make the notation consistent to the one presented in Chapter 2, we will remove the arrow of the vectors to have a cleaner approach to the random variable description presented earlier in this thesis. The reader should keep in mind that each $\vec{X}^{(m)} \equiv X^{(m)}$, where the super-index (m) is the label of the M vectors involved $(m \in 1, \ldots, M)$ and that each of which has N entries (dim $X^{(i)} = N \ \forall i$).

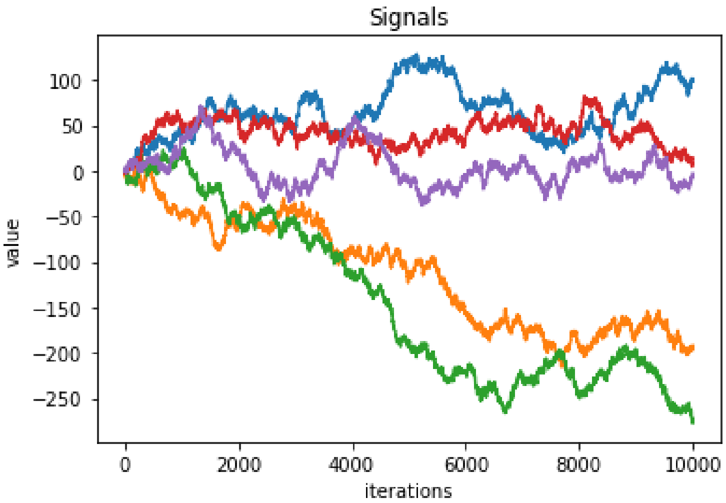


Figure 4.2: *In this figure each plot represents a synthetic channel described by a vector* $\vec{X}^{(i)}$ *from* $i = 1, 2, 3, 4, 5$.

From Figure 4.2 we see the $X^{(m)}$ *signals* having different expected values after being generated according to (4.1). By shifting the expected value of the *signals* to zero (i.e. $E[X^{(m)}] = 0\ \forall m \in M$ (Figure 4.3)) and standardize the $X^{(m)}$ with $\sigma_m = 1$, we see that the correlation between $X^{(i)}$ and $X^{(j)}$ for $i \neq j$ is

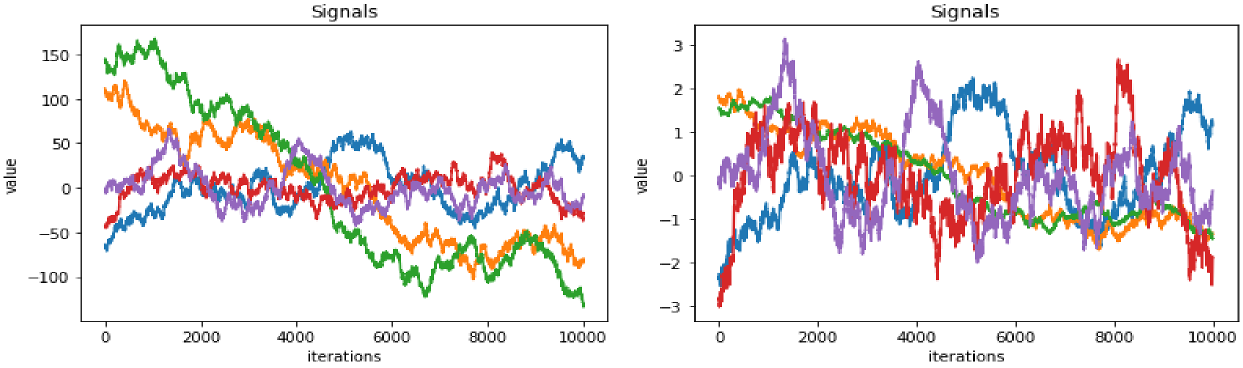
$$\mathrm{Cov}(X^{(i)}, X^{(j)}) = E[(X^{(i)} - \underbrace{E[X^{(i)}]}_{=0})(X^{(j)} - \underbrace{E[X^{(j)}]}_{=0})] = E[X^{(i)}, X^{(j)}] = E\left[\sum_k \sum_l B_{ik} w_k B_{jl} w_l\right]$$
$$= \sum_k \sum_l B_{ik} B_{jl} \underbrace{E[w_k, w_l]}_{\delta_{k,l}} = \sum_k B_{ik} B_{jk} = \sum_k B_{ik} B^T_{kj}.$$

From the computation above we see that $E[w_k, w_l] = \delta_{k,l}$ because by construction we are using that the $w^{(k)}$ are independent and uncorrelated vectors for $k \neq l$. Thus, the covariance matrix of the process is given by $\mathbb{B}\mathbb{B}^{\mathbb{T}}$, and therefore, the *correlation matrix* is given by

$$\rho(X^{(i)}, X^{(j)}) = \frac{\mathrm{Cov}(X^{(i)}, X^{(j)})}{\sqrt{\mathrm{Cov}(X^{(i)}, X^{(i)})\mathrm{Cov}(X^{(j)}, X^{(j)})}} = \frac{(\mathbb{B}\mathbb{B}^{\mathbb{T}})_{ij}}{\sqrt{(\mathbb{B}\mathbb{B}^{\mathbb{T}})_{ii}(\mathbb{B}\mathbb{B}^{\mathbb{T}})_{jj}}} \tag{4.2}$$

Where $\rho(X^{(i)}, X^{(j)})$ are the entries of the *analytical* correlation matrix at the i column and j row. Notice that the expression in (4.2) is different to what we will refer to as the *numerical correlation matrix*. The latter is obtained by applying eq. (2.14) to the vectors $X^{(m)}$ in (4.1) directly [2].

[2] Another way to generate *synthetic channels* (Figure 4.1) is by using *white noise* signals (*Wiener increments*). With these, we can build the $\vec{w}^{(i)}$ with $\mu_{\vec{w}^{(i)}} \approx 0$ and $\sigma_{\vec{w}^{(i)}} \approx 1$ directly, and use the analysis described above to generate the $\vec{X}^{(m)}$.



(a)

(b)

Figure 4.3: *Time-series from the 5 signals in Figure 4.2 when (a) the mean is shifted to zero and (b) when we shift the signals and normalize by the standard deviation ($\sigma_i = 1$).*

Since the correlation matrix can be directly estimated from the data eq.(2.14), the difference between such estimation and the true value $\mathbb{B}\mathbb{B}^{\mathbb{T}}$ can be computed. To evaluate how different the 2 matrices are, we consider the *Frobenius norm* [3] of their difference:

$$\|\text{Corr}[\vec{x}]_{an} - \text{Corr}[\vec{x}]_{num}\|_F \equiv \|\text{Corr}[\vec{x}] - \widehat{\text{Corr}}[\vec{x}]\|_F \coloneqq \|\mathbb{A}\|_F, \tag{4.3}$$

where here, $\vec{x} = (X^{(1)}, \ldots, X^{(M)})$ as the multivariable case in Chapter 2, $\widehat{\text{Corr}}$ is the numerical correlation matrix, and where the Frobenius norm (B) of a matrix $\mathbb{A}$ is defined by

$$\|\mathbb{A}\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} |a_{ij}|^2}, \tag{4.4}$$

with a_{ij} the entries of such matrix.

Henceforth, to analyze how different those two matrices are we consider the relative magnitude of this difference defined by

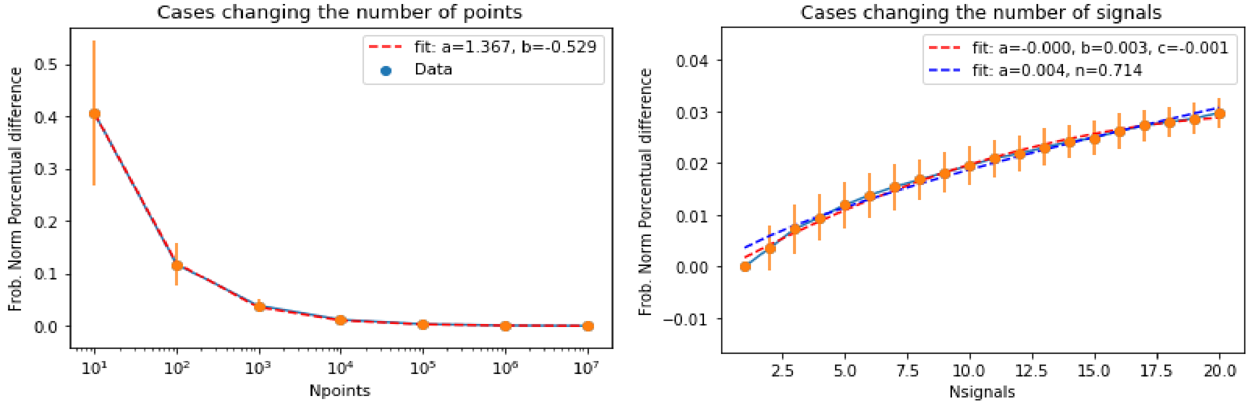
$$\epsilon = \frac{\|\text{Corr}[\vec{x}] - \widehat{\text{Corr}}[\vec{x}]\|_F}{\|\text{Corr}[\vec{x}]\|_F}. \tag{4.5}$$

This scalar ϵ quantifies how different the *analytical* and the *numerical* correlation matrices are. To have a better insight of how the Frobenius norm behaves when we change the number of time steps (N) and the number of channels (or signals M), we can do the following:

1. Fix the number of *signals* and compute the Frobenius norm for different time-steps N. Typically by going from 10^1 up to 10^7 time-steps in increases of one order of magnitude. For this case, we ran 50 simulations to examine the behaviour of the norm when there is an increase on the number of time-steps. The plot generated after the simulation has points center in the average of values after the simulations and uncertainty given by the standard deviation (Figure 4.4 *a*).

[3] See Appendix B for the justification of this norm.

2. Fix the time-steps (10^4) and increase the number of *signals* from 2 to 20. For this case, we run 10^3 simulations. The plot in this case is generated in the same way as in the paragraph above, taking the average and the standard deviation after the simulations (Figure 4.4*b*).



(a) (b)

Figure 4.4: *Behaviour of the percentage difference for the Frobenius norm, when the parameters N and M are changed. The plot displayed in a) reflects the behaviour of the percentage difference after performing 50 simulations. Here, the number of signals is fixed to 5, and the number of points go from 10^1 to 10^7. As we can see from the figure, the data follows a power law behaviour $y(x) = a * x^b$ with parameters $a = -1.367$ and $b = -0.529$. The plot in b) is the percentage difference after performing 1000 simulations. Here, the number of time-steps are fixed to 10^4, and number of signals increase form 2 to 20. In this case, we observe that the data also fits well to a power law behaviour $y(x) = a * x^k$ with parameters $a = 0.004$ and $k = 0.714$ (blue dashed line).*

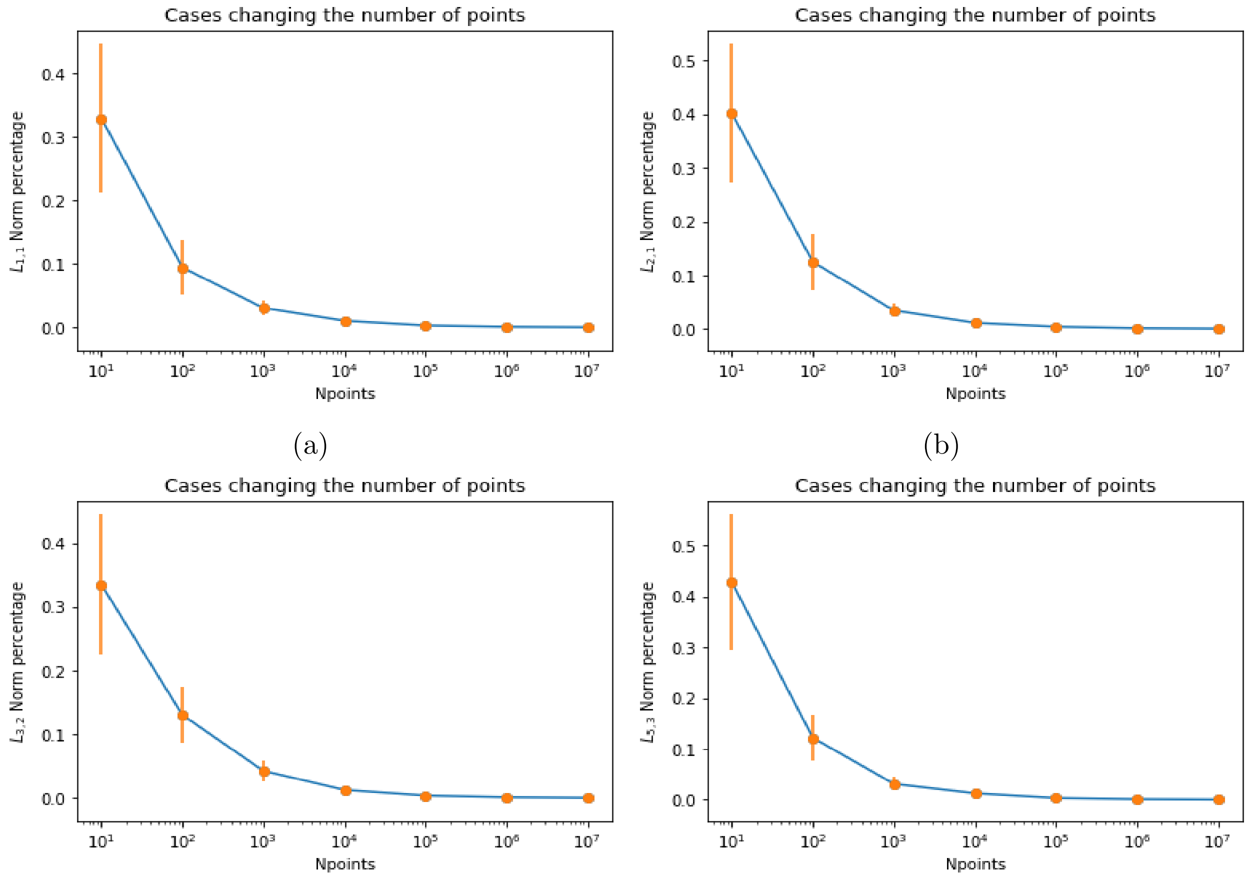
From the figures shown in Figure 4.4, we can see that the Frobenius norm is one way of calculating the error between the *true* and the *numerical* correlation matrices. In that figure we can appreciate the Frobenius norm $\| \cdot \|_F \to 0$ following a power law behaviour when $N \to \infty$. Notice that in Figure 4.4*a*) the error bars from the uncertainty also decrease when the number of time-steps increase. This means that the more time-steps we have in a signal, the more accurate the *numerical* implementation is. On the other hand, Figure 4.4*b*) shows that when we increase the number of *signals*, the percentage difference between the numerical and analytical cases, increase in a power-law fashion. The latter can be appreciated by the curve fitting with a blue dash line where we have a power-law fit. In contrast, we can also observe a second order polynomial curve fitting with a red dashed line respectively. An important remark at this point is that we cannot forget that the correlation matrices described above in eq. (2.14), are merely *linear* correlation measures, and a similar comparison for nonlinear measures is also relevant particularly when we consider the mutual information defined in eq. (2.17).

It is also possible to introduce other types of matrix norms for the difference between the *analytical* and *numerical* correlation matrices. The generalization of the Frobenius norm is the $L_{p,q}$ norm defined as

$$\|\mathbb{R}\|_{p,q} = \left(\sum_{j=1}^{n}\left(\sum_{i=1}^{m}|r_{ij}|^{p}\right)^{\frac{q}{p}}\right)^{\frac{1}{q}}. \tag{4.6}$$

where the r_{ij} are the entries of the matrix $\mathbb{R}$.

For $p = q = 2$ the $L_{p,q}$ norm retrieves the Frobenius norm as in eq.(4.4). An interesting case is when $p = 2$ and $q = 1$ because this norm could act as an error function widely used for robust data analysis given that the error for each point is not squared. In Figure 4.5 we can see the behavior of different $L(p, q)$ norms when the number of time steps is increased. A question that arises at this point is if there is a way to optimize p and q for which the $L(p, q)$ norm is minimum for this particular type of data.



(a)

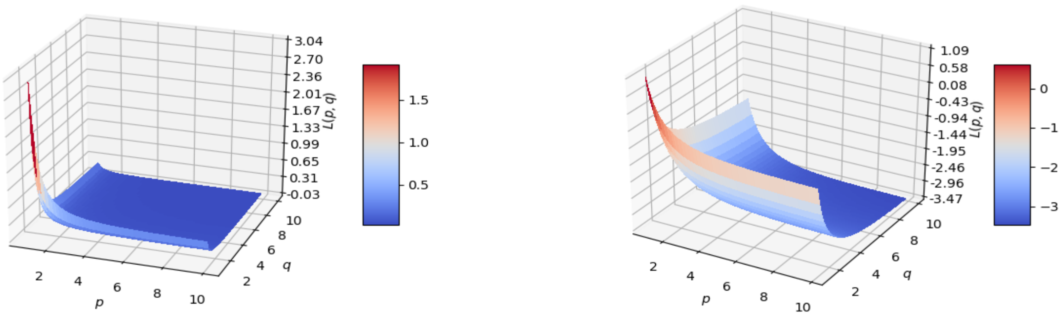
(b)

(c)

(d)

Figure 4.5: *Behaviour of the $L(p, q)$-norms. In these figures we see the behaviours of the $L(p, q)$-norms for 5 signals after 20 simulations when: (a) $p = q = 1$, (b) $p = 2, q = 1$, (c) $p = 3, q = 2$ and (d) $p = 5, q = 3$. In these figures we still obtain a power-law as in Figure 4.4.*

Given that eq.(4.4) and eq.(4.6) are entry-wise matrix norms that treat a $m \times n$ matrix as a vector of size $m \cdot n$, the parameters' choice p and q in eq.(4.6) must not affect the behaviour of the matrix difference because geometrically speaking, this represents a mere distance between 2 vectors. In addition to that, the *equivalent norm theorem* in (B.0.1) can be used to refute a more privileged matrix norm since the ones we considered in this thesis are from a more general one as in eq.(4.6). The norm election re-scales the value obtained after the computation but the parameters do not give any information of how good the norm is. That depends on the type of data to analyze. To see this argument more clearly, we can plot a surface with different values for p and q. In Figure 4.6, we can observe that if we increase the values of p and q, the surface that represents the $L(p, q)$ norm of the difference between the *analytical* and *numerical* correlation matrices, decreases monotonically. This means that by increasing the parameters of such norm we will obtain a similar power-law behaviour. It is because of this reason that in future analysis of this thesis we will only consider the Frobenius norm due to a more recognizable geometric meaning to the Euclidean space.



(a)

(b)

Figure 4.6: *Behaviour of the $L(p, q)$-norms for different values of its parameters. The surface follows the prescription given in* (4.6) *when we compare the analytical and the numerical correlation matrix with synthetic data. In* (a) *we observe that as we increase the values of p and q the norm decreases monotonically showing that there is not a predilect value for p or q. In* (b) *we display the natural logarithm of the $L(p, q)$ norm to observe the behaviour better.*

4.2 Implementation of mutual information to coupled stochastic processes

As we have previously mentioned in Chapter 2, one way to measure nonlinear correlations between a pair of signals is by computing the mutual information from their probability distributions. This means that first, we need to determine the probability distributions of the $\vec{X}^{(i)}$ and then use eq. (2.19) to compute the mutual information. Equation (2.19) contains the information about the *joint probability distribution* and the *marginal probabilities* from the pair of random variables involved.

To obtain such information from our $X^{(i)}$'s, we can obtain the *normalized* 2D histogram between a pair of signals $X^{(i)}$, $X^{(j)}$ (for $i, j \in 1, \dots M$). By giving a specific number of bins, we can determine the joint probability distribution matrix and the marginal probabilities (Figure 4.7).

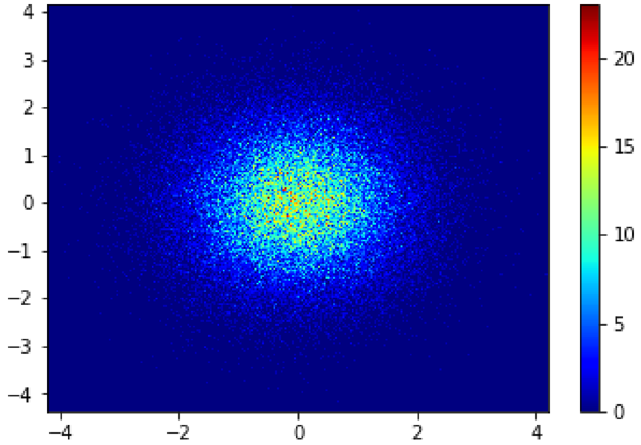
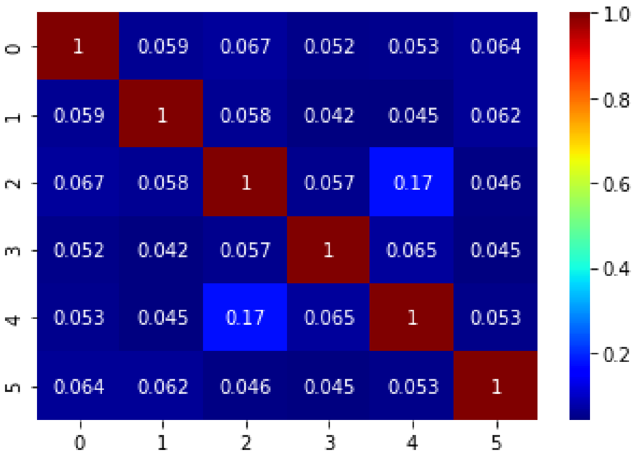


Figure 4.7: *2D Histograms for the joint probability distribution between two signals $X^{(i)}$ and $X^{(j)}$. In this case $i \neq j$. In this figure we observe the join probability matrices represented as a heat map. The heat map of the joint prob. matrix showed here corresponds to $N = 10^5$. The election of number of bins in the cases displayed are $\sqrt{N}$.*

The reader should recall that the $X^{(m)}$ are generated following the prescription in eq.(4.1). From the joint probability matrix in Figure 4.7, it is possible to obtain the marginal probabilities $P(x_i)$, $P(x_j)$ to use them as indicated in eq.(2.19) since those, are the sum or the values per column and per row of the joint probability matrix. The joint probabilities are simply the entries on the matrix in Figure 4.7. With these quantities, it is possible to compute the mutual information matrix between the generated *signals*.

In Figure 4.8 we can see a mutual information matrix for a collection of 6 signals. Each matrix entry represents the computation of the *normalized* mutual information of a pair $X^{(i)}$ and $X^{(j)}$. As we can see, the matrix is symmetric and follows the properties from eq.(2.19) entry-wise. The reason for introducing such matrix is because the *normalized mutual information matrix* (Figure 4.8) allows us having a more direct comparison to the correlation matrix, based on a nonlinear approach. The normalization choice decided for this thesis consist on taking each entry of the mutual information matrix divided by the square



	0	1	2	3	4	5
0	1	0.059	0.067	0.052	0.053	0.064
1	0.059	1	0.058	0.042	0.045	0.062
2	0.067	0.058	1	0.057	0.17	0.046
3	0.052	0.042	0.057	1	0.065	0.045
4	0.053	0.045	0.17	0.065	1	0.053
5	0.064	0.062	0.046	0.045	0.053	1

Figure 4.8: *Normalized mutual information matrix. In this case, $m \in [1, \ldots, 6]$ and the amount of time-steps is $N = 10^5$. We notice that the mutual information matrix is symmetric as in (2.19). The terms in the diagonal represent the normalized Shannon Entropies.*

root of the product of the adjacent diagonal terms of the matrix [4]. In other words

$$\tilde{I}_{ij} = \frac{I_{ij}}{\sqrt{I_{ii} * I_{jj}}} \tag{4.7}$$

At this point, we can repeat a similar analysis for this mutual information matrix as we did for the correlation matrix in the previous section. Since we do not have an *analytical* and *numerical* comparison for mutual information matrices, we will examine the behavior of the Frobenius norm when we increase the number of signals (M) and the number of time steps (N).

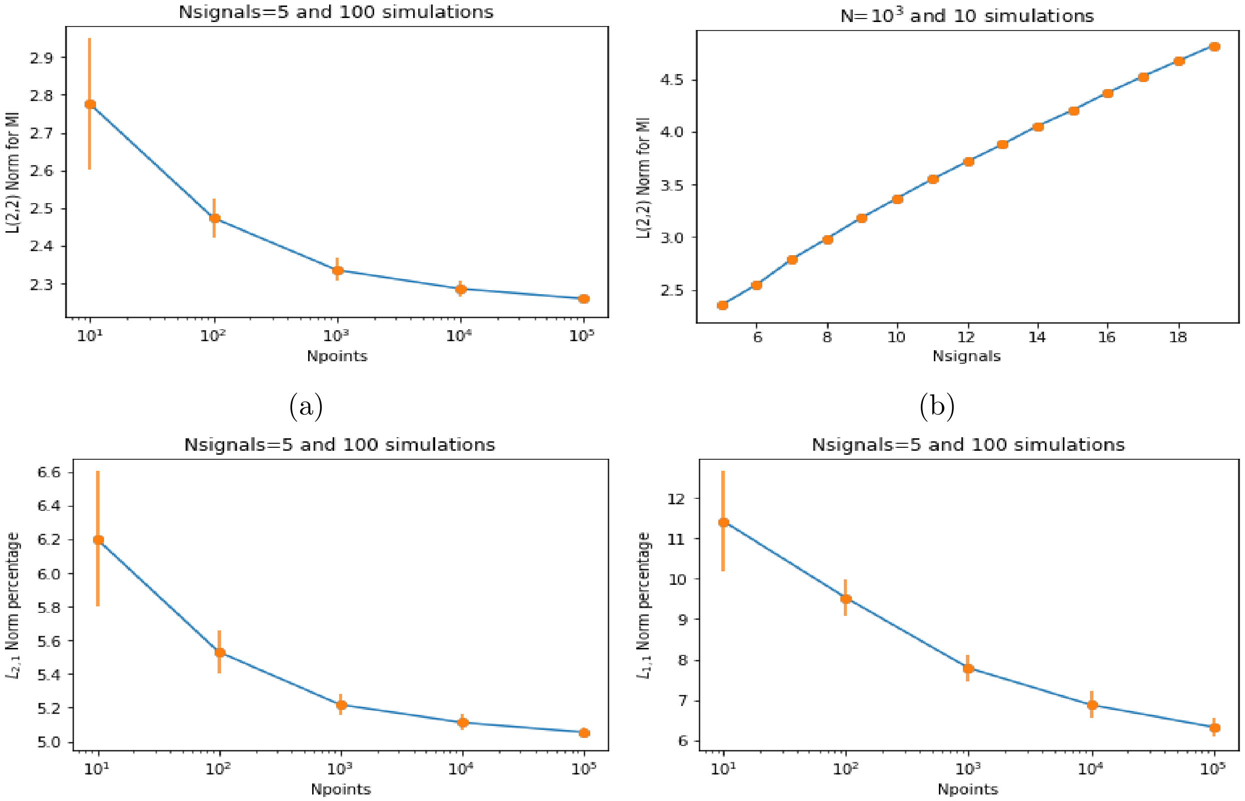
For this case, we run some simulations to analyze the behaviour for the Frobenius norm.

- After having the routine for the mutual information matrix, we can fix the number of signals and change N. In this case, we run 100 simulations for when N goes from 10^1 to 10^5 as in the Figure 4.9.

- We can see the other simulation fixing $N = 10^3$ and increasing the amount of signals from 5 to 19 . In this other case, we run 10 simulations due to the computational time.

In Figure 4.9, we observe a similar behavior for the different $L(p, q)$ norms applied to the normalized mutual information matrices to the ones obtained for the correlation matrices.

[4] This normalization election is not unique. The normalization election used in this thesis is chosen in such way so that the diagonal terms also take part in the calculation and to have the diagonal terms equal to 1. Since $I(X, Y) \in [0, \infty)$, other prescriptions could also be used to normalize the mutual information entry-wise. For instance, one of the most immediate manners of normalizing the mutual information is to take the measure: $\tilde{I}_{ij}(X, Y) = 1 - e^{-I_{ij}(X,Y)}$, where $\tilde{I} \to 0$ when $I \to 0$ and $\tilde{I} \to 1$ when $I \to \infty$. The advantage of this normalization is that it treats high values of the mutual information in a finite manner unlike eq.(4.7) where divergences could occur if the denominator is small enough. Another normalized measure commonly used [8] is $\tilde{I}(X, Y) = \dfrac{2 \cdot I(X, Y)}{H(X) + H(Y)}$.



(a)

(b)

(c)

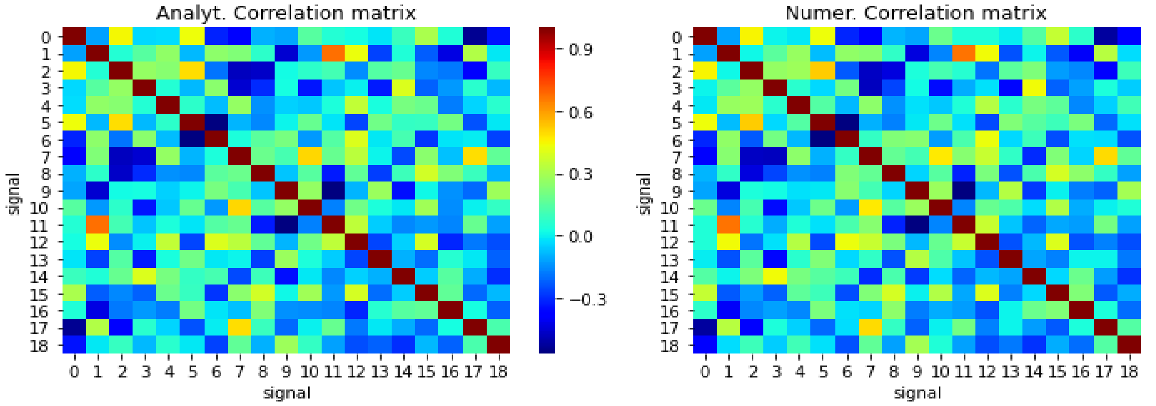
(d)

Figure 4.9: *Analysis using the different norms for the normalized mutual information matrix. In (a) we can see the plot of the $L(2,2)$ norm when we increase the number of time-steps. A power-law behavior is similar to the one obtained in (4.4). The plot in (b) shows a similar behavior as in Figure 4.4 (b) when the number of signals is increased from 5 to 19. In (c) we see a similar behaviour for the $L(2,1)$ after 100 simulations. In (d) we see the behaviour for the $L(1,1)$ norm after 100 simulations. As expected, the matrix norm election has the same effect for the mutual information.*

The points plotted in that figure represent the mean values after each simulation and the error bars correspond to their respective standard deviations. It is clear that when one increases the number of time-steps, the$L(p, q)$ norms decrease as a power-law. This is similar to the analysis in Figure 4.5 but with the difference that in 4.9 we can see that the parameters p and q make the *elbow* of the curve in a slightly different way. With this we can confirm once again that the Frobenius norm is good enough for evaluating a percentage error between matrices.

4.3 Features of the mutual information matrix for synthetic data

In this section we will observe the behaviour of the *mutual information* matrix for synthetic data. In order to do so, we generate 19 signals with 10^3 time-steps each. We chose the value of this parameters this way to simulate a similar scenario for our real data set. As we have previously seen, from eq.(4.5) and in previous sections of this chapter, the percentage error between the *numerical* and *analytical* correlation matrices is expected to be low for 10^3 time-steps. In Figure 4.10 one can see both correlation matrices displayed for the case of synthetic signals. In this particular case, the percentage error (ϵ) between them is 8.62%.



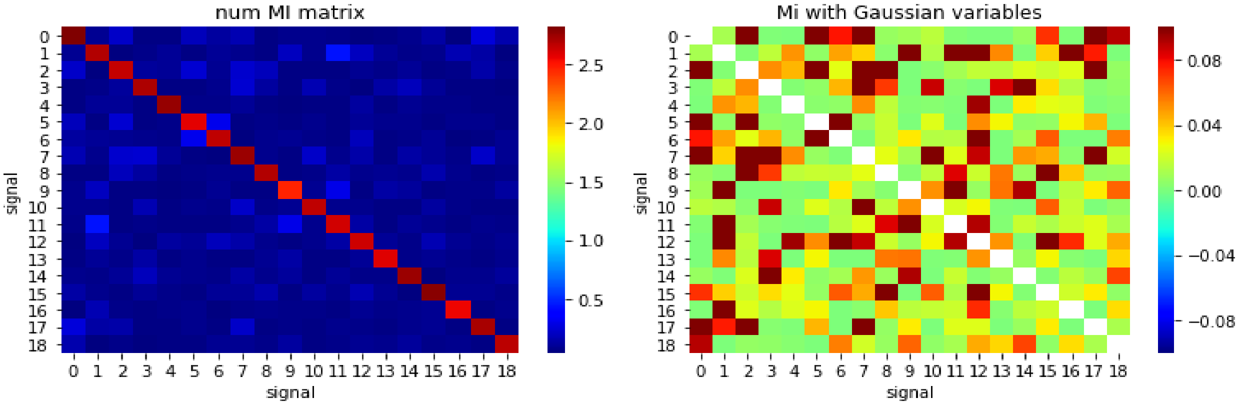
(a)

(b)

Figure 4.10: *Comparison between analytical and numerical correlation matrices for synthetic data. As we can see from the heatplots, the matrices are very similar. Here the percentage error is around 8.62%*

For the case of the mutual information matrix, as we have previously described in section 4.2, first, we need to compute the joint probability matrix and from there, use eq.(2.19) to estimate the mutual information. The way of estimate the joint probability distribution and therefore, the *joint* entropy as in eq.(2.20), is by computing a 2D histogram per pair of variables. This method involves an arbitrary election of the number of bins involved in the computation which induces a bias and increases the percentage error. Due to the amount of time-steps involved we consider Sturge's formula for the number of bins in the 2D histogram (i.e., $k = \log_2(N) + 1$) since the amount of steps is large enough to obtain a decent result for the calculation of the mutual information.

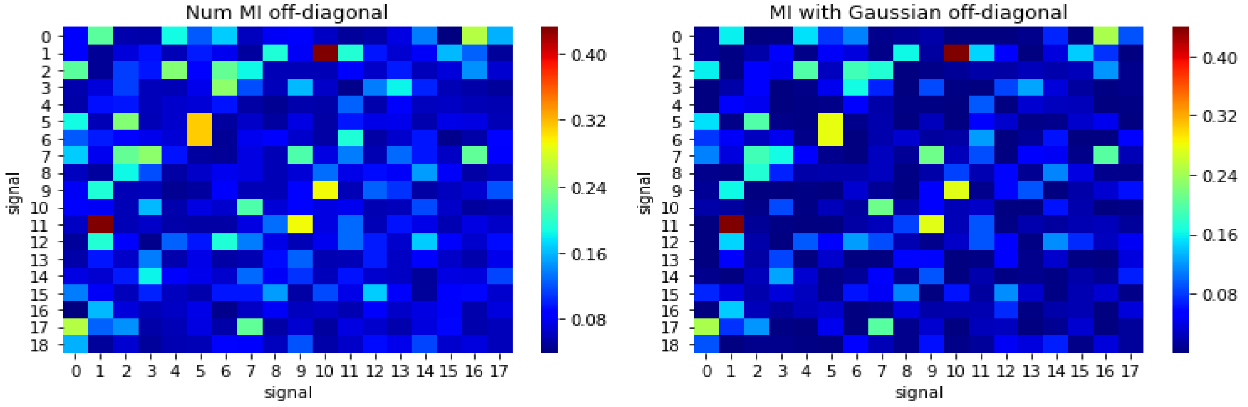
Contrary to the correlation matrix case, for the mutual information matrix there is not such direct analog between *numerical* and *analytical* mutual information matrices. However, it is possible to compute a similar comparison by calculating eq.(A.3) with the correlation coefficients directly from the analytical correlation matrix. The reason of this is because the only well-known case between mutual information and correlation coefficient is given by Gaussian variables as described in Appendix A. In this case, such comparison can provide us with information about how close is the data to a Gaussian behaviour.



(a) (b)

Figure 4.11: *Comparison between mutual information matrices for synthetic data. In* (*a*) *is the M.I. matrix generated by the joint probability distributions using 2D histograms (using Sturge's formula), whereas* (*b*) *is the application of expression (A.3) using the coefficients from the analytical correlation matrix.*

As we can see from Figure 4.11, it is possible to see certain similarities between the mutual information matrices. The matrix generated by the joint probability distributions in (a), shows that the off-diagonal contributions are much smaller than the diagonal terms. This mean that the Shannon entropy of each individual signal is considerable larger than the mutual information per pairs of signals. It is also possible to see some off-diagonal entries where the coefficients are larger but still close to zero. For the implementation with Gaussian variables in (b), it is possible to see a similar setting. The off-diagonal terms represent regions with small values close to zero but there are many that have a larger positive influence. We notice in this case that the contribution compared to the diagonal terms is very low as in (a). One feature that it is possible to notice is that in both heatmaps there are certain off-diagonal regions that seem to have a qualitative similar *footprint*. The latter indicates that even after calculating the 2D histogram for the joint probabilities, the data might still have a trace of Gaussian behaviour. Based on the observation from the heatmaps, it is also noticeable that the values between them are quite different in magnitude with the biggest difference displayed in its diagonal terms. Since $I(X,Y) \geq 0$, $I(X,Y)$ can in principle take large values just as the diagonal terms in Figure 4.11(b) as a direct consequence of the maximum linear correlation along the diagonal. Given that the diagonal terms in both matrices don't allow us making a clear comparison on the overall contribution for different signals, we can remove such terms and calculate the percentage error for the off-diagonal terms (Figure 4.12).



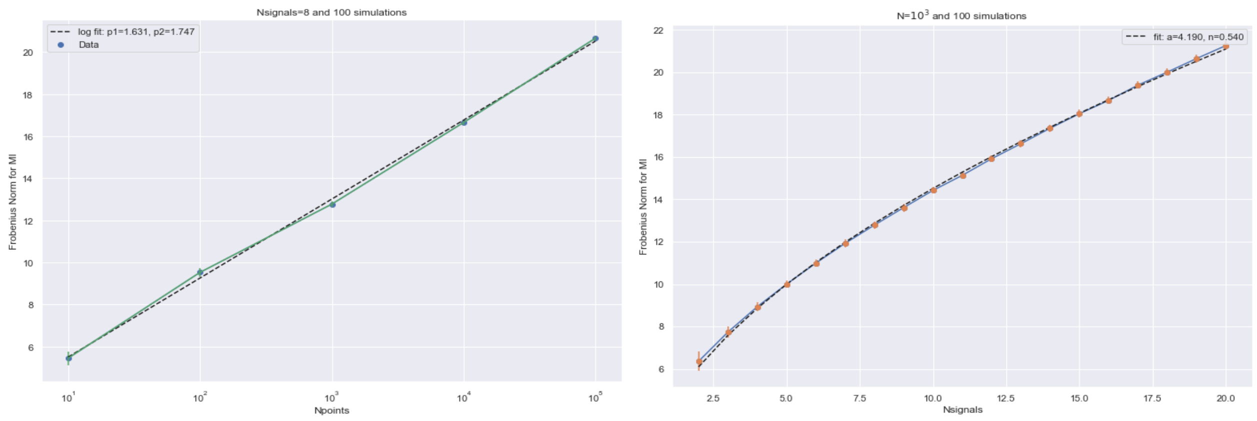
(a) (b)

Figure 4.12: *Comparison between mutual information matrices for the off-diagonal terms using synthetic data. In* (*a*) *is the M.I. matrix generated by the joint probability distributions using 2D histograms whereas* (*b*) *is the application of expression A.3. Here we display the heatmaps exclusively for the off-diagonal terms in Figure 4.11, obtaining a percentage error of 68.26%.*

In Figure 4.12 we observe a qualitative indication that the inherent Gaussian behaviour is still there even after using the binning election previously mentioned. In the ideal case that the binning method collected all the information needed to compute the mutual information for each pair of signals, we would've expect Figure 4.11(a) to be much close to (b), since the signals were created using white noise as described in section 4.1 above. However, we can see that the number of bins selected have a bias involved that increases its deviation from the ideal case. As for the off-diagonal terms in Figure 4.12, although a clear pattern is obtained displaying Gaussian behaviour, the percentage error is 68.26%. This is a significant increase compared to the one for the correlation matrices is attributed to the election of the amount of bins needed at the joint entropy level. This percentage error using of Sturge's formula retrieves a much lower error compared to the square-root bin choice in Figure 4.7.

Given that the mutual information matrix for the synthetic signals is fundamentally important to the analysis of this thesis, an immediate question to ask is: What happens to the mutual information when the time steps and the amount of signals vary?. In Figure 4.13 we observe the behaviour of the Frobenius norm for the mutual information matrix as a function of the *number of points* and the *number of signals*. In this case, we observe that when we fix the *number of signals* to 8 and run 100 simulations, the Frobenius norm increases in a logarithmic manner. In particular, the curve for Figure 4.13 (a) that fits better the simulation points is $y(x) = 1.631 * \ln(x) + 1.747$ with $R^2 = 0.998$. Notice that this Figures are different from the ones in Figure 4.9 because in this case, we are not considering the normalized mutual information matrix where we have an upper bound for the mutual information.

For case (b) in Figure (4.13), we observe that as we increase the number of available signals



(a)

(b)

Figure 4.13: *Frobenius norm behaviour for the mutual-information matrix as a function of a) number of points (time-steps) and b) number of signals. In these cases we observe the behaviour after 100 simulations. For the first case (LHS), we encounter a logarithmic increase of the Frobenius norm when there is an increase on the number of points. In b) (RHS) we can see that the curve that fits the data more accurately is the power law black dashed line.*

in the simulation, the black dashed line fits the data more accurately than the quadratic regression. In this case, we observe again the power law is the best description for the simulation following the equation $y(x) = 4.190 * x^{0.540}$ with $R^2 = 0.9994$ [5].

In the previous sections of this chapter, we have introduced a set of steps and tests for working with synthetic data to compute the correlations and mutual information between a set of signals involved in an experiment. We tested these tools to observe the behavior of the matrix norm and the comparison between *numerical* and *analytical* cases, where we encountered that the binning method has a relevant role for obtaining a precise result. In the following chapter we will apply these tools the real fNIRS data sets in more in detail.

[5] For this case, we also noticed that a parabolic fit is also accurate but to a lower degree than the power-law fit previously mentioned. In particular, we encounter that the best parabolic fit follows the equation $y(x) = -0.016x^2 + 1.158x + 4.451$ with a $R^2 = 0.99907$. Although, both fits have a large R^2, a power-law behavior is better at an early stage when few signals are present in the simulation, in addition to one less coefficient involved in the fit.

Chapter 5

Applied framework to fNIRS data

As showed in the previous chapter, we have visualized the signals' contributions using linear and nonlinear correlation measures. Namely, the Pearson coefficient and mutual information. In both cases, it is clear to notice that some signals maintain strong dependencies to others with the autocorrelations being cases of maximum correlation. Additionally, based on the comparisons between the analytical and numerical cases, a clear correlation pattern (or $footprint$) for both types of measures is maintained (Fig. 4.10 and 4.12). It is not surprising that such footprint exists given the fact that we are examining correlations of time series, but what is interesting is the interpretation of the coefficients as well as the collective behavior of all the signals involved.

In this chapter, we focus our attention on fNIRS data and apply the framework to analyze the functional connectivity networks based on linear and nonlinear correlations. To do so, we examine the data sets more carefully. Since our data set has not been pre-processed, we need to distinguish the appearance of possible patterns and try to quantify them for a posterior global description of the data. In particular, we want to see if using information measures for nonlinear correlations, we could obtain a new perspective of how the regions of the brain are interconnected to each other.

5.1 Comparison between correlation and mutual information matrices for fNIRS signals

After justifying the routines for the correlation, mutual information, and the behavior of the $L(p,q)$ norm for *signals* generated by Wiener processes; we would like to compare the corresponding matrices for our particular data set (Figure 3.2). The data sets contain measures from an individual in control group, performing physical activities such as walking in a 8-figure pattern, walking along an unstable terrain and walking with glasses of water while maintaining the attention to the glass. The brain activity lectures from such experiments are saved in `.txt` files for $oxy-Hb$ and $deoxy-Hb$. After tabulating the data, it was observed that from the 20 channels recorded in the experiment, channel 12 did not have numerical values (both for the oxy-Hb file and for the deoxy-Hb file). This may be the consequence of measurement errors. Therefore, such channel was removed from both data sets, making the analysis consisting of 19 channels ($M = 19$) with 8891 readings (*time-steps*) for the oxy-Hb file and the deoxy-Hb file (Table 5.1).

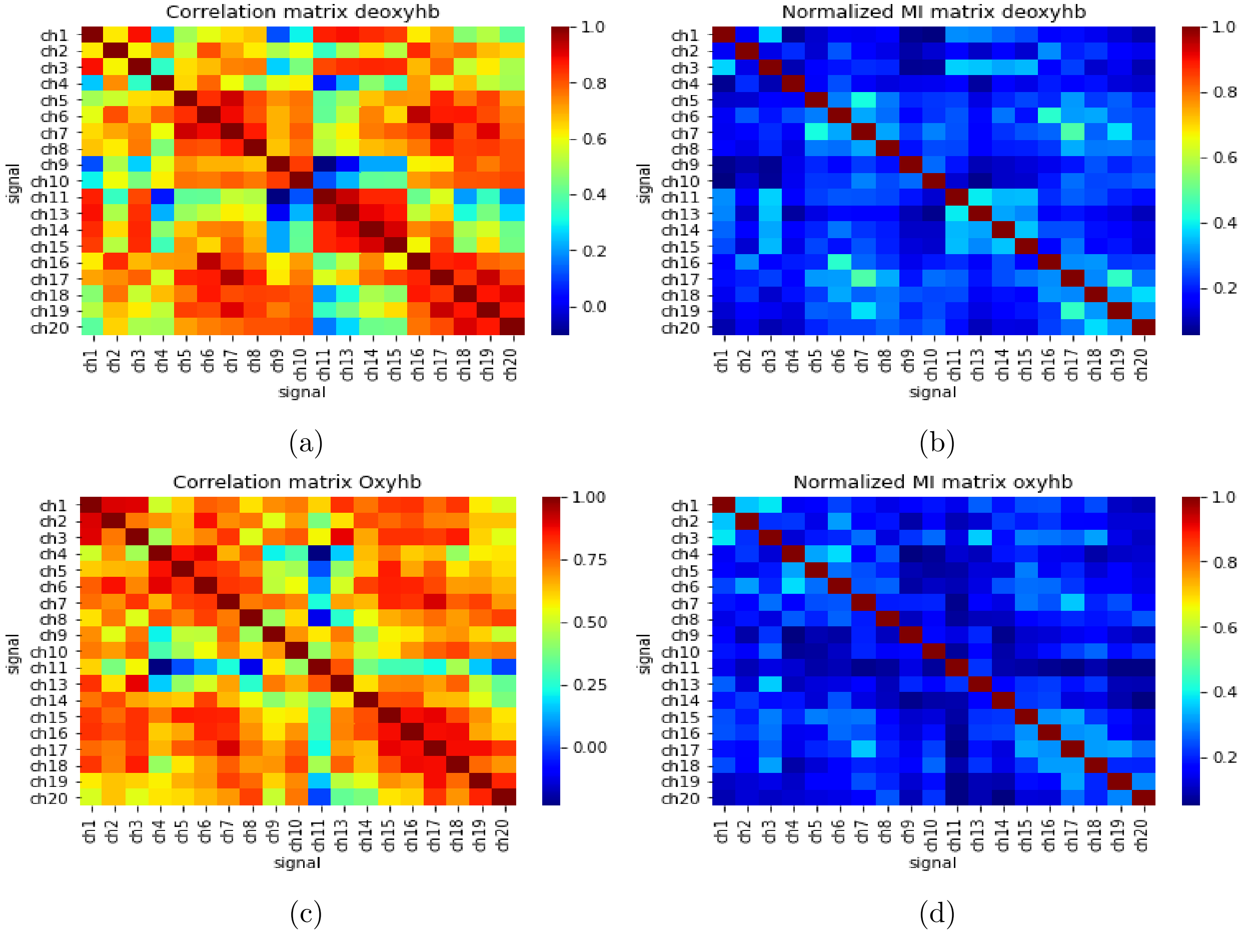
ch1	ch2	...	ch11	ch13	...	ch19	ch20
-0.000942	-0.000305	...	-0.001724	-0.001717	...	-0.000270	-0.000070
-0.001108	-0.000434	...	-0.001754	-0.001822	...	-0.000265	-0.000071
-0.000954	-0.000360	...	-0.001764	-0.001853	...	-0.000250	-0.000069
-0.000994	-0.000310	...	-0.001785	-0.002018	...	-0.000293	-0.000066
-0.001006	-0.000275	...	-0.001813	-0.001920	...	-0.000295	-0.000084

Table 5.1: *Table of the different measurements for deoxy-Hb. Here, the columns represent the channels in the cap, while each row represents the measurements obtained in each interval of time. Note that channel 12 was removed from deoxy-Hb and oxy-Hb files due to the absence of numerical values. Here, 5 out of the 8891 measurements are shown for the deoxy-Hb file. For the oxy-Hb there is a similar setting.*

As we can see from Table 5.1, each column represent a time series which we can treat as a discrete random variable, in this sense, it is possible to calculate the Pearson coefficient and the normalized mutual information pairwise and visualize the comparison between the correlation and the *normalized* mutual information matrices for the deoxy-Hb and oxy-Hb data sets (Figure 5.1).

In Figure 5.1 it is clear to see that the correlation matrices for both cases show regions with high correlation. The red zones reveal *clusters* where certain signals are more correlated to other neighboring signals. We can also see in the four heatmaps that the diagonal terms are the ones with the highest correlations and the largest information gains. The heatmaps on the LHS [1] of Figure 5.1 make evident that a different pattern is followed for oxy-Hb and deoxy-Hb data sets. We can see for the deoxy-Hb case a very well marked region where signals have strong correlation, whereas in the oxy-Hb case, the correlations are strong but in a more scattered fashion. For the normalized mutual information matrices (RHS), we see a slightly different behavior. Although we obtain the highest information gains in the diagonal, we observe that for different signals, the normalized mutual information is generally lower to its correlation counterpart. We notice that certain highly correlated regions have also higher mutual information contributions and that a very well marked footprint is maintained for those signals. The latter is an indication that nonlinear correlations are still relevant during the experiment and that those are easy to observe after examining the normalized mutual information. The important comment that needs to be added is that the pattern between the signals for both types of linear and nonlinear measures is somehow maintained for the oxy-Hb and deoxy-Hb cases, but that the data sets do not retrieve the same pattern. Such patter is more manifest by using Sturge's rule for the amount of bins needed in the 2D histogram when calculating the normalized mutual information. Therefore, this binning rule is the one followed along the rest of the text.

[1] Left hand side (LHS) and right hand side (RHS).



(a)

(b)

(c)

(d)

Figure 5.1: *Comparison between the correlation and normalized mutual information matrices for the 19 signals in each data set.* (*a*) *and* (*b*) *correspond to the deoxygenated Hb, and* (*c*), (*d*) *are for the Oxygenated Hb.*

5.2 Topology for the virtual channels

In Figure 3.4 we sketched a representation of the possible network we might obtain from analyzing possible correlations from experimental data. As been mentioned before, for the purposes of this thesis we only consider measurements on the prefrontal cortex of the brain. Is because of this reason that a visualization of the physical location of the optodes is important. In this way, the heatmaps' representation from Figure 5.1 can have a graphical interpretation about the brain activity. In particular, this region covers the forehead of the individual with a setup sketched in Figure 5.2. In this figure it is possible to observe the optodes' physical distribution along the subject's prefrontal cortex. The numbers colored in *red* represent the location of the *light sources* whereas the *blue* ones indicate the location of the *light detectors* respectively. An important remark to consider is that the *gray* edges adjacent to each node in Figure 5.2, represent the *virtual channel* between a source-detector pair. Such *virtual channels* are precisely what we have been referring in this text as *signals*. The reader can easily verify that from the experiment that there are 20 of them. However, as mentioned before, the virtual channel number 12 was compromised leaving our analysis with 19 signals, corresponding with the amount of signals in Section 4.3. From this point on, when we refer to *channels* we mean *virtual channels*.

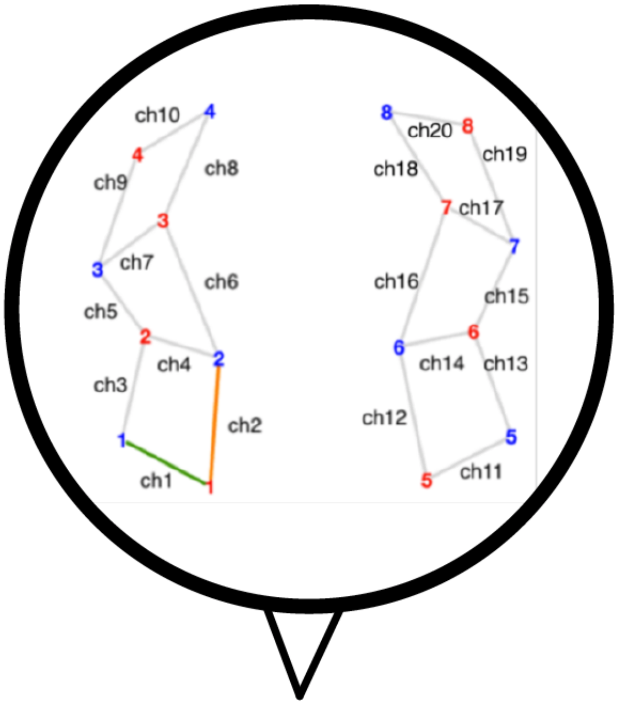
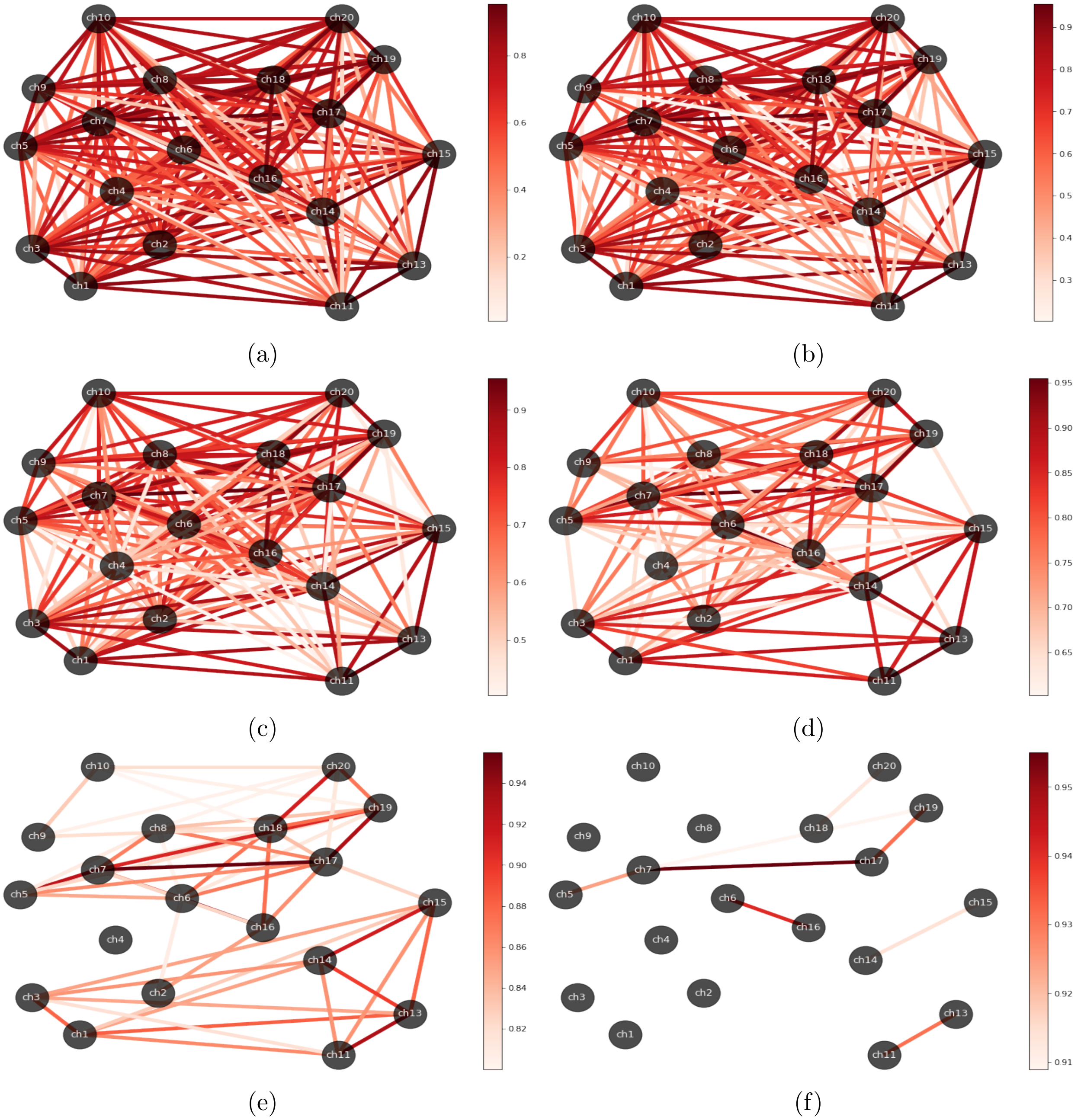


Figure 5.2: *GUI for the optodes generated by Homer3. Here, we are representing the topology of the optodes placed on the individual's forehead. The red and blue numbers correspond to the light sources and the detectors respectively. The gray edges correspond to what is typically know as virtual channels. These virtual channels are what we are referring as signals and as we can see, our configuration involves 20 of them following the same order as in Figure 5.1.*

Figure 5.2 give us a better idea of the activated regions of the brain after the subject performs a task. With this figure the reader can notice that there is now a correspondence between the heatmaps in Figure 5.1 and the physical position during the experiment. However, since the virtual channel is a link between the light source and the detector, the resolution of the measurement is compromised since there is a region where such hemodynamic response could happen. Such contribution is considered to be somewhere along the connecting line between a pair nodes. Therefore, for our purposes we consider the main contribution located at the center of each edge of Figure 5.2, which facilitates the analysis when drawing a network.

5.2.1 Networks based on correlation over a certain threshold



(a)

(b)

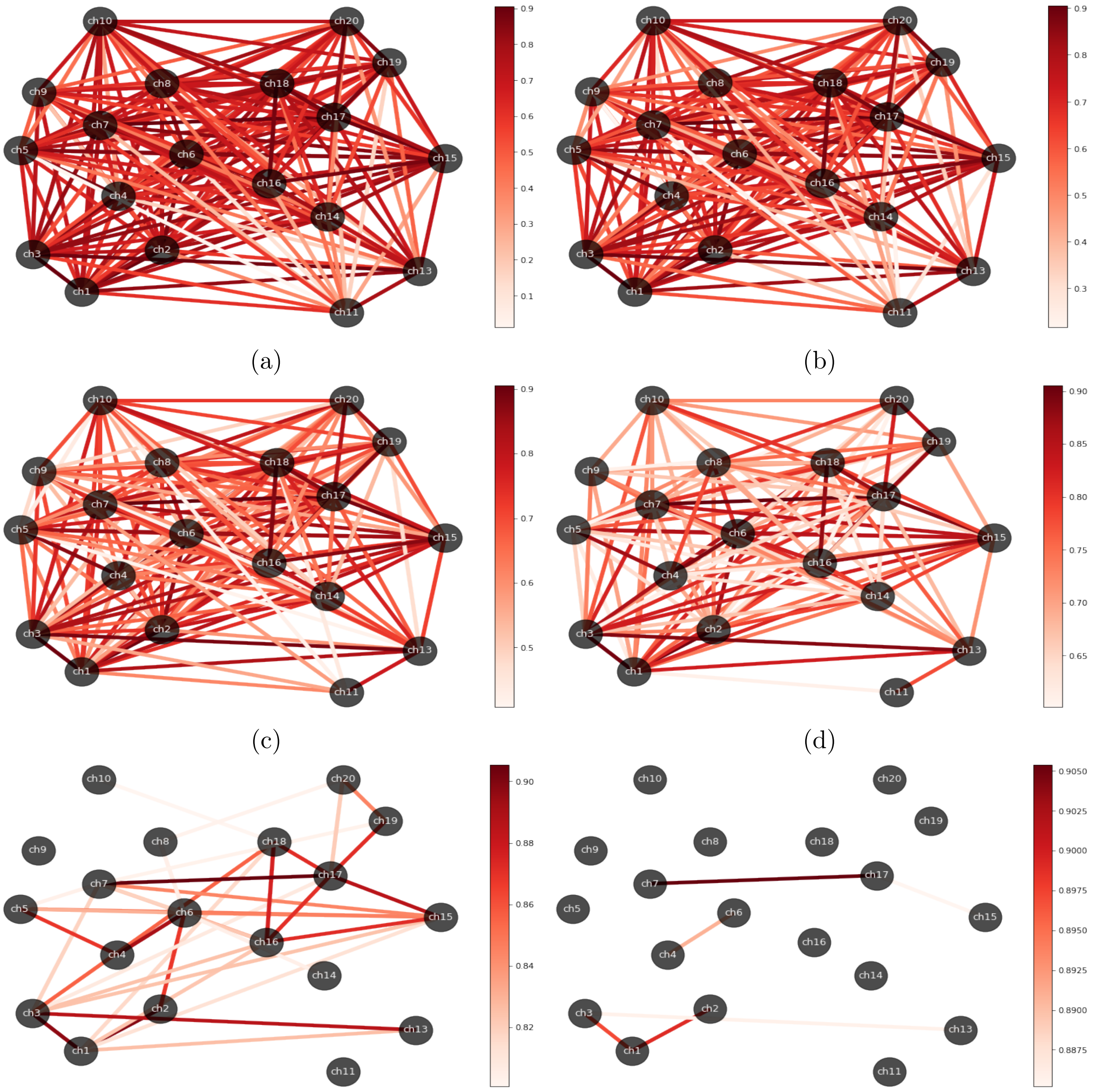
(c)

(d)

(e)

(f)

Figure 5.3: *Undirected weighted networks for the deoxy-Hb based on correlation threshold. In (a) we are keeping the edges of the network in case there is a correlation factor > 0 between each node. For the subsequent cases the threshold is increased by 0.2 except for (f) where the threshold is set to be > 0.9. In this figure we are avoiding self-correlations for the nodes (corresponding to a self-loop) since the correlation factor is equal to 1. The thresholds for these networks are set to be larger than 0, 0.2, 0.4, 0.6, 0.8 and 0.9 respectively.*

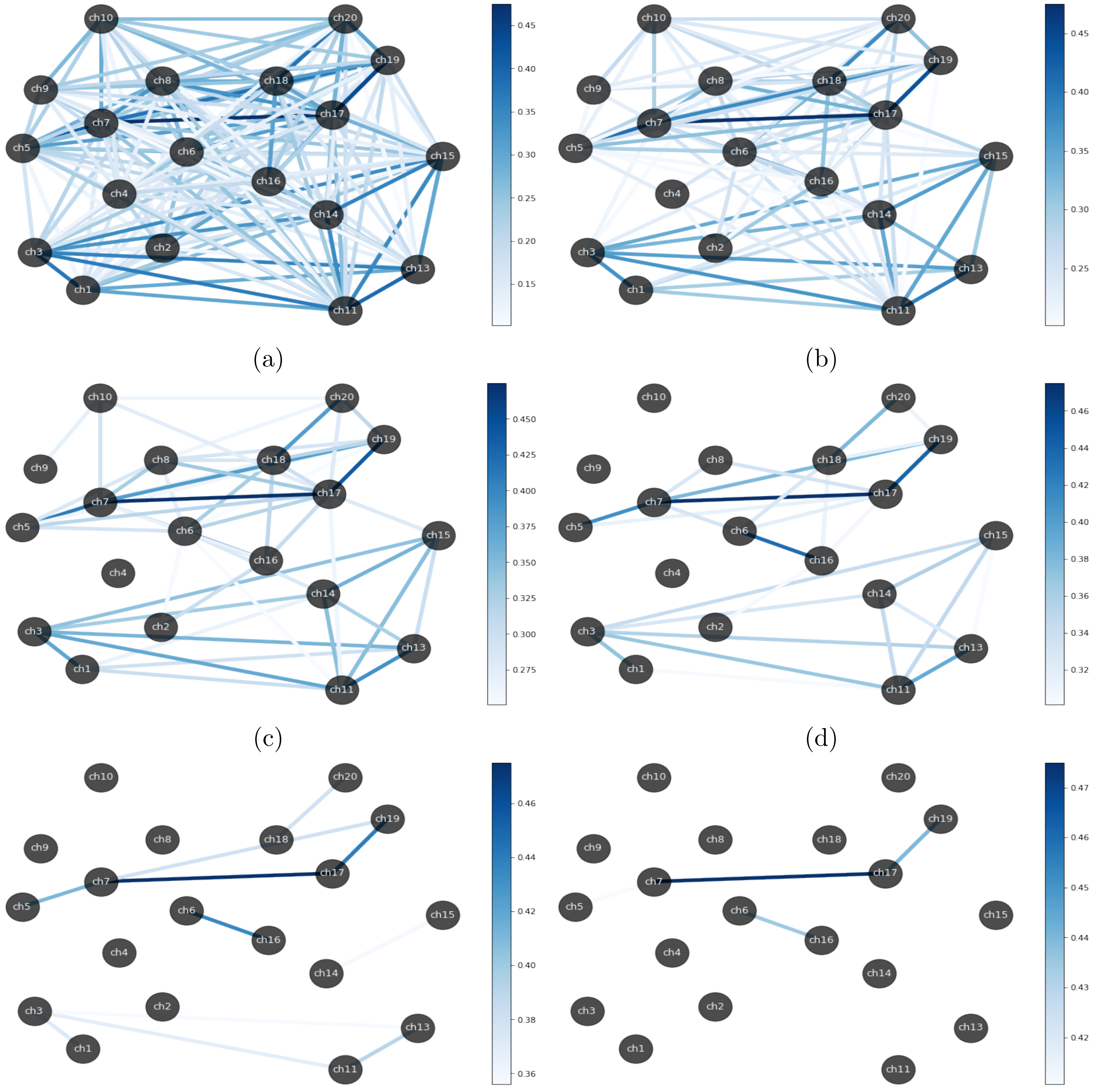


(a) (b)

(c) (d)

(e) (f)

Figure 5.4: *Undirected weighted networks for the Oxy-Hb based on correlation threshold. In this figure we show the networks with a correlation threshold from 0 to 0.8 with an increase step of 0.2 just as in Figure (5.3). The f) case corresponds to the threshold set at 0.9. The thresholds for these networks are set to be larger than 0, 0.2, 0.4, 0.6, 0.8 and 0.9 respectively.*



(a)

(b)

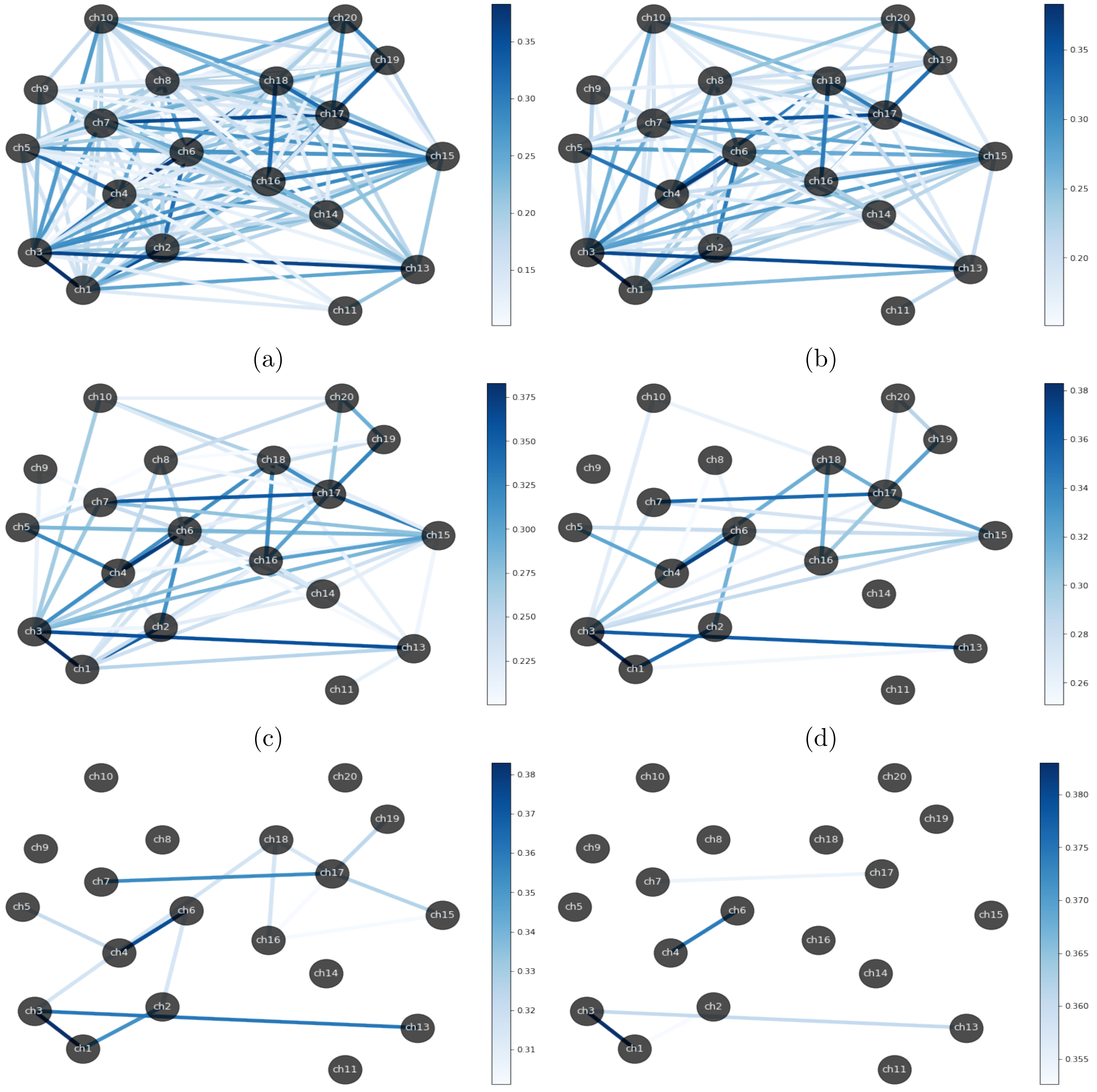
(c)

(d)

(e)

(f)

Figure 5.5: *Networks for the deoxy-Hb based on normalized mutual information threshold. In a) we are keeping the edges of the network for a threshold > 0.1 between each node. For the subsequent cases the correlation factor is increased by 0.05 until case f) where the threshold follows a correlation factor > 0.45. The thresholds for these networks are set to be larger than 0.1, 0.2, 0.25, 0.3, 0.35 and 0.45 respectively.*



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.6: *Networks for the oxy-Hb based on normalized mutual information threshold. In a) we are keeping the edges of the network for a threshold > 0.1 between each node. For the subsequent cases the correlation factor is increased by 0.05 until case f) where the threshold follows a correlation factor > 0.35. The thresholds for these networks are set to be larger than 0.1, 0.15, 0.2, 0.25, 0.3 and 0.35 respectively.*

The visual representation of the heatmaps between the different *channels* allows detecting which virtual channels have more influence on each other. This matrix visualization provides an overall picture of a more general interaction between regions of the brain but lacks specif details given its physical localization. There is another way to exhibit the latter by drawing the networks where we can keep the correlation manifest over a certain threshold. The latter means it is possible to draw the graph where each *signal (virtual channel)* (a node in a more general weighted graph) and the links between them, represent the existence of correlation. What we intend to explore here is the progression of a functional connectivity network when a correlation threshold is imposed (Figure 5.3).

If we consider the $signals'$ as the nodes of an undirected weighted network where the edges between them are drawn with the respective weights indicated by the entries of the matrices given in Figure 5.1, then, we can observe the functional connectivity of such figure in a different fashion (Figs. 5.3,5.5, 5.4 and 5.6). Furthermore, if we now set a threshold and observe the networks' formation between the *signals*, we can actually see the evolution of the connectivity between them when the threshold is increased. In Figure 5.3 we present the generated networks for the deoxy-Hb case, when different correlation thresholds are imposed.

In Figure 5.3(a) we keep all edges of the graph that have a positive correlation between the respective pair of nodes, and the corresponding weight of each edge follows a color bar scale. As expected, most of the links are drawn in red due to the few cases where the correlation between the signals is ~ 0. Figure 5.3$b-e$ presents the cases where the correlation threshold is increased by 0.2, meaning that in (b) we are only drawing the edges where the correlation between nodes is > 0.2 and so on. In Figure 5.3 cases (e) and (f) we notice than the number of edges drawn is reduced because only few nodes preserve high correlations. An important remark is that in Figures 5.3, 5.4 we are not considering self-correlations at all. In Figure 5.4 we present the same analysis for the Oxy-Hb. The reader can notice that the nodes' distribution are not symmetric. The underlying reason is to avoid confusion with overlapped edges in the diagrams and preserve the clarity of the diagrams.

For the deoxy-Hb and oxy-Hb sets in Figures 5.3 and 5.4, we see that the lower the threshold the more dense the network is. Also, the higher the threshold the more sparse it becomes. We also notice that the most correlated virtual channels in oxy-Hb do not always correspond to the most correlated ones in the deoxy-Hb case, as initially observed in Figure 5.1. The exception occurs for the pair between channel 7 and channel 17, where the correlations are constantly high in both cases. By having these correlation thresholds it is possible to detect certain clusters within the network that are more representative and that indicates a larger region of influence along the individual's brain cortex. For example,in Figure 5.3 $(a) - (c)$ it is not possible to see a clear distinction from the more correlated signals, but we can see that as the threshold increases, the density of each node decreases considerably form case to case. In cases from Figure 5.3$(d)-(f)$ we can observe a more sparse networks being formed. Notice that the network in case (f) suggests that the participant is a right handed individual.

For the oxy-Hb case, we observe a similar correlation influence between Ch7 and Ch17 as in deoxy-Hb, but a weaker correlation with the rest of the channels. The (f) cases in both Figures 5.3 and 5.4 are different given that for oxy-Hb network we encounter only few pairs of strongly correlated virtual channels in the left and right side, whereas for the deoxy-Hb we observe a clear activation mostly on the right hand side of the network. In both Figures

5.3 and 5.4 we observe that the strongest correlations are present along signals from opposite sites (Figure 5.2), suggesting that the activated region manages to bridge both sides of the subject's head during the activity.

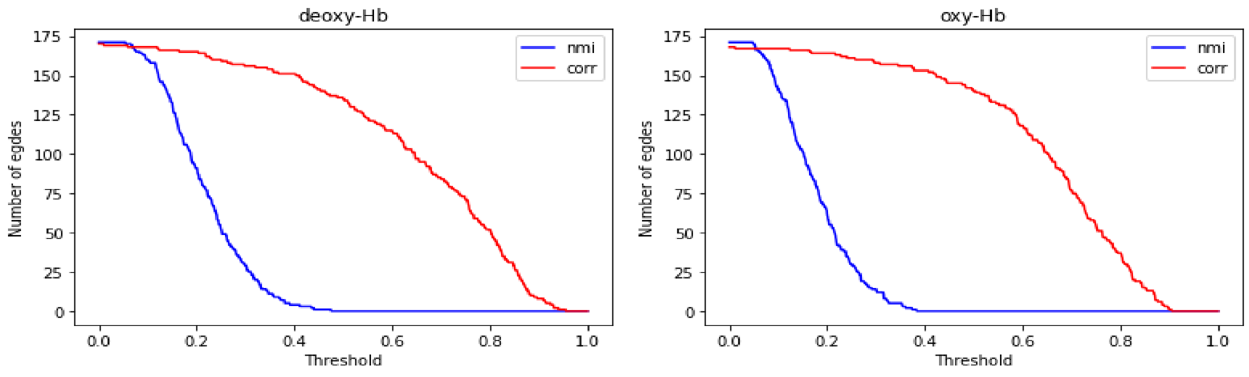
One can repeat this visualization for the mutual information matrices on the fNIRS data (i.e. for Figures 5.1 b and 5.1d). If we consider again the *normalized* mutual information matrices as in Figure 5.1, what one obtains is that the networks become sparse much faster than the ones in Figures 5.3 and 5.4. For the case of deoxy-Hb, this behaviour can be seen more clearly in Figure 5.5. There, the initial threshold starts at 0.1 (case (a)) for the normalized mutual information up to a threshold equal to 0.45 (Figure 5.5f). We can see that although the progression of these networks as we increase the threshold becomes less dense much faster than in the correlation case, it is possible to see two disconnected sub networks in (e) and (f) that suggest nonlinear nature.

For the oxy-Hb with the normalized mutual information case, we have a similar thing. In this case we see that the upper threshold is obtained at 0.35 with the similar increasing on the number of steps as for the deoxy-Hb case (Figure 5.6). We have to point out that for the oxy-Hb networks in Figures 5.4 and 5.6, we obtain different observations. For instance, Fig. 5.4f displays the pair chan7-17 to be the most correlated whereas Fig. 5.6f the heaviest edges correspond to the pairs between chan3-1 and chan4-6. These results indicate that perhaps there are extra contributions perceived by the normalized mutual information that are not entirely captured by the Pearson coefficient.

To observe the behaviour of the number of edges in the network as we increase the threshold, we can take a look at Figure 5.7. In this figure, we notice that we are only displaying coefficients that are strictly positive for comparison purposes between the Pearson coefficient and the normalized mutual information. Pearson coefficient $\rho \in [-1, 1]$ as we have previously discussed whereas the normalized mutual information is ≥ 0, meaning that in Figure 5.7, we are not displaying the negative correlated channels. The latter is due to the very few negatively correlated pair of channels in the data. In Figure 5.7a, we observe that the red plot starts a little below the blue one when we allow the threshold to be larger than zero. This is because for the deoxy-Hb case, the only negative correlation pair obtained corresponds to $\rho_{ch9-ch11} = -0.1038$. In contrast, we notice that the gap between the red and blue plots in Figure 5.7b is more notorious, and the reason is because for the oxy-Hb, there are three pairs of channels that are negatively correlated, namely: $\rho_{ch4-ch11} = -0.236$, $\rho_{ch8-ch11} = -0.117$ and $\rho_{ch11-ch20} = -0.001$.

The plots in Figure 5.7 indicate the number of edges that have a coefficient larger than the value indicated in the x-axis. In other words, for $x = 0$ we observe that we obtain the maximum number of edges. Meaning that we are counting all edges in the network with a threshold larger than zero. Naturally, as we increase the threshold the number of edges that are larger than such value decreases.

The number of existing edges for the Pearson coefficient (*red*) and for the normalized mutual information (*blue*) evolve differently as expected from the analysis of the networks depicted above. For both cases, we notice a concave shape for normalized mutual information and a convex shape for correlation. This shape for the normalized mutual information exhibits that the bits of information between channels are less intense compared to their Pearson coefficient, indicating an inherent nonlinear correlation between the virtual channels involved.



(a) (b)

Figure 5.7: *Amount of present edges in the correlation and normalized mutual information networks as a function of the threshold. The red plot corresponds to the correlation case, whereas the blue is for the normalized mutual information. That the number of counts of the red plots is initially lower than in the blue counterparts because in these figures we are only considering positive defined values for comparison purposes.*

5.2.2 Tree-structured graphs for hierarchical clustering

In the previous subsection, we exhibited the generated networks based on Pearson correlation and normalized mutual information thresholds for deoxy-Hb and oxy-Hb data sets. It was clear that when the threshold was higher, the formation sub networks between nodes appeared, revealing strong influences among them. To explore this behavior more clearly, we can observe the hierarchical cluster (or *dendrogram*) for the Pearson correlation and the normalized mutual information. Figures 5.8 and 5.9. There, we can see the formation of clusters following a *bottom-up* approach.

The hierarchical clusters in Figure 5.8 uses the information of the matrices in Figure 5.1, and arranges it by the similarity of its coefficients. We use the `seaborn.clustermap` function in Python to compute the clusters and the dendrograms. The procedure consist on figuring out which entry (Pearson or normalized mutual information coefficient channel i) is most similar to another. After forming the first cluster based on the similarity of the entries, we continue doing this comparison until we figure out which two signals are most similar pairwise. Once this has been achieved, we merge them into a bigger cluster. This procedure is repeated until we don't have any more clusters to compare. Figure 5.8 is accompanied by its corresponding dendrograms which indicate both the similarity and the order in which the clusters were formed.

The method used for determining the similarity in our dendrograms is the Euclidean distance. The linkage method involved here is Ward's method. This method of cluster comparison exploits the Euclidean distance in order to minimize the variance between them. Since each cell in the heatmaps has a numerical value, what the Euclidean distance method does is to calculate the distance between each similar cluster. In this case, since the matrix is symmetric, the dendrograms are mirrored by the diagonal. Such Euclidiean distance can be understood as $\sqrt{\sum_i (a_i - b_i)^2}$ where $a_i - b_i$ is the difference between the samples involved (i.e., the entries of the matrices in Figure 5.1). In our case, those values are the numeric

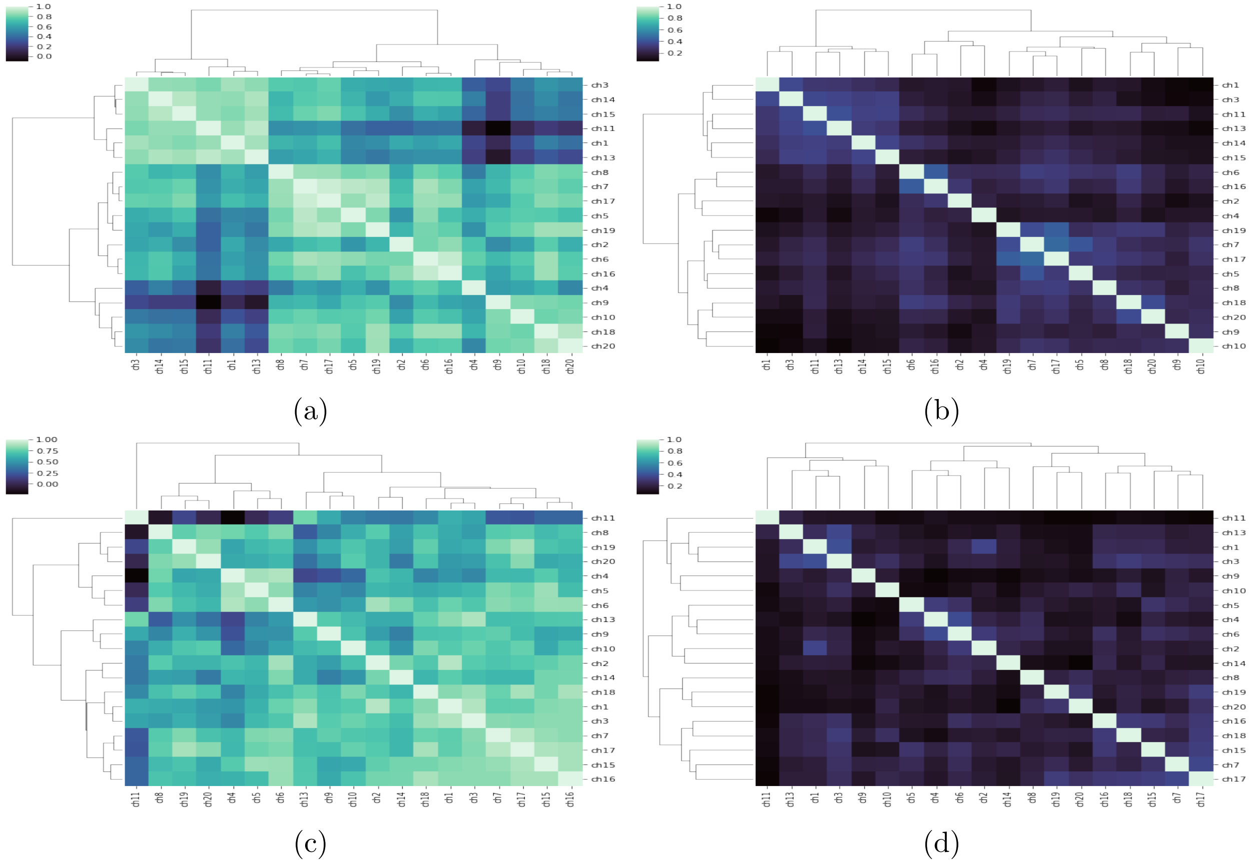
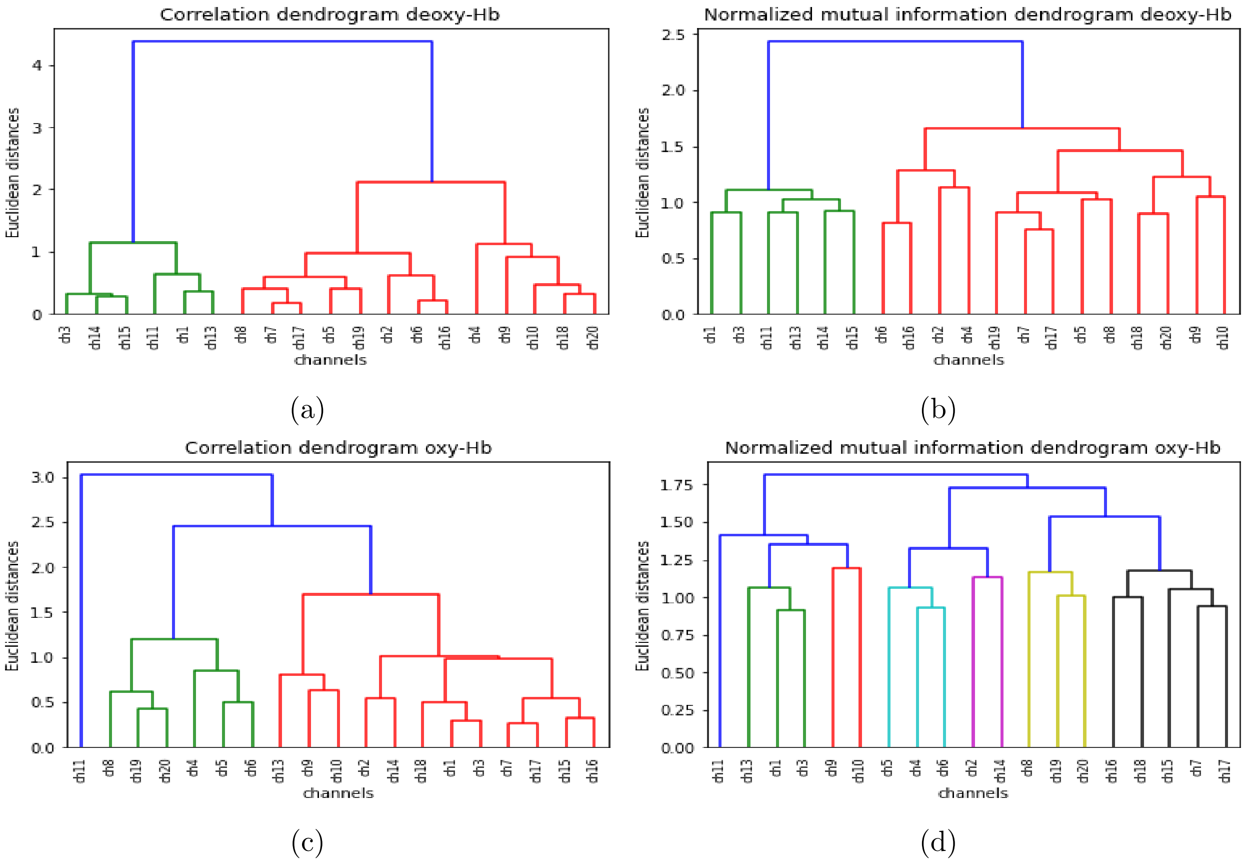


Figure 5.8: *Heatmaps with dendrograms showing the hierarchical clustering of the channels. These dendrograms are implemented by a Euclidean metric and with a linkage set up following Ward's criteria. In the LHS of the figure we observe the dendrograms for the correlations matrices for deoxy-Hb* (*a*) *and Oxy-Hb* (*c*)*. In the RHS we observe the dendrograms for the normalized mutual information for the deoxy-Hb* (*b*) *and oxy-Hb* (*d*) *.*

values from the correlation (or normalized mutual information) of the virtual channels. A better picture for the height of the branches in the dendrograms can be seen in Figure 5.9. There, the height show which cluster is more similar. The shortest branches indicates which cluster is first formed based on the similarity of its coefficients.

For deoxy-Hb case, we see that the clusters are formed faster (or equivalently, in a shorter Euclidean distance) that in the normalized mutual information counterpart where the clusters are mostly formed when such distance is close to 1. Similarly for the oxy-Hb case. The dendrograms for deoxy-Hb case reflect the formation two big structures (green and red) when the Euclidean distance is large enough. The green and red clusters for deoxy-Hb contain the same channels in both cases but the formation of pairs are not constant in each case. We observe that the strongest connection obtained by the networks is between ch7-ch17 resulting on the formation of the earliest cluster in both correlation and normalized mutual information dendrograms.

For the oxy-Hb, we observe a notorious difference in regards to the formation of the dendrograms. In this case, the correlation dendrogram forms three major clusters whereas, for



(a)

(b)

(c)

(d)

Figure 5.9: *Dendrograms for the correlation and normalized mutual information matrices between fNIRS channels.*

the normalized mutual information counterpart, we observe seven. This difference exhibits the distinction between the deoxy-Hb measures because with the mutual information we can see that more specific responses are captured compared with the linear correlation. The latter is justified by the number of sub-clusters depicted in different colors in Figure 5.9(c) and (d). In Figure 5.9(c) we observe that the first cluster correspond to the pair between ch7-17 whereas in Figure 5.9(d) the first formed cluster corresponds to the pair ch1-3 matching the last sub networks from the previous section.

The dendrograms for the normalized mutual information cases reach just above the half of the Euclidean distance of their correlation counterparts. This is because the values of the normalized mutual information are considerably smaller than the ones in the correlation case explaining why the threshold in Figures 5.5 and 5.6 is lower than in Figures 5.3 and 5.4 respectively. Moreover, the fact that we obtain dendrograms for normalized mutual information suggest that the nature of dependencies codify somehow the non-linearity between each channel. The reason to make this claim is because as seen in Figure 5.1, an existing footprint between linear an nonlinear measures is evident. If such footprint had an overall influence on all entries of the matrices, then we would expect very similar dendrograms but with scaled Euclidean distances, preserving a similar cluster formation.

5.3 PCA for fNIRS data

PCA can qualitatively explain such relationship by finding a list of *principal axes* in the data and using those axes to describe the data sets [41], [17]. Moreover, from Figure 3.2 one can see that the data sets do not have tall peaks where well-defined features of the brain activity are displayed. It is for this reason that many of the features of NIRS spectra have highly correlated readings [32], [2] and therefore, need to be treated statistically.

The procedure we use to apply PCA to the fNIRS data sets is described as follows:

From the data, it is possible to obtain information about the means and the *standard deviations* of each channel. The latter will inform if a normalization (or standarization) needs to be performed to get rid of any bias towards the features that could lead to false results. The datasets used in this project required a standardization for the variables. This means that before finding the principal components, a scaling of the variables was performed so they have zero mean and standard deviation equal to 1. This was done using the `scale`() function from `sklearn` (Table 5.2).

ch1	ch2	…	ch11	ch13	…	ch19	ch20
-2.435881	-2.006836	…	-2.128628	-2.428613	…	-1.182455	-0.557294
-2.926276	-2.491063	…	-2.182049	-2.605588	…	-1.161627	-0.5624051
-2.471257	-2.214142	…	-2.199027	-2.658068	…	-1.108564	-0.555368
-2.588064	-2.025180	…	-2.236887	-2.937227	…	-1.268245	-0.542556
-2.623596	-1.894624	…	-2.286736	-2.771390	…	-1.273498	-0.622475

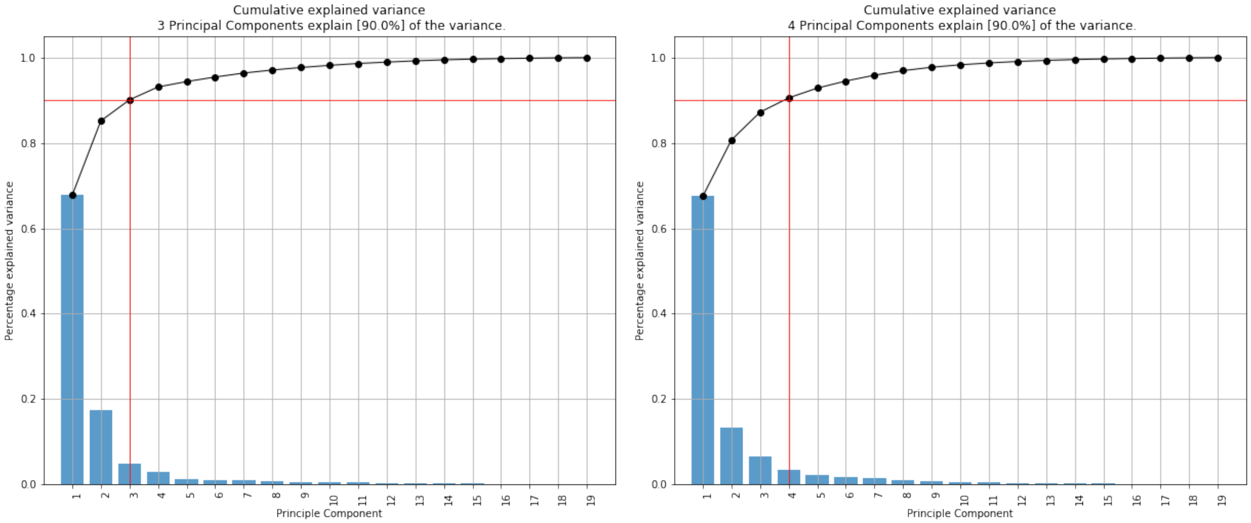
Table 5.2: *Table of the different measurements for deoxy-Hb after the standarization of the data to zero mean and variance equal to 1.*

The `PCA`() function provides with information about the *explained variance* and the *explained variance ratio*. The first describes the amount of variance of each component and the second, the percentage of variance explained (PVE) by each of the components (Figure 5.10). This will help determine the minimum amount of principal components for high percentage of variance of each data set.

In Figure 5.10, one can see the cumulative explained variance for both data sets. There, we plot the expected variance ratio for the 19 principal components presented as color bars, and the cumulative proportion of explained variance with the black line. Each principal component explains some of the existent variations in the data. For deoxy-Hb in Figure 5.10 (a), one can see that the first 3 principal components account for the 90.11% of the total variance while for the oxy-Hb the first 3 principal components account for the 87.22% of the total variance. For the oxy-Hb in Figure 5.10 (b) 90.56% of the total variance is achieved by the first 4 principal components.

Because the first three principal components represent the majority of the variance for both cases, Figure 5.11 shows the scatter plots of the first 2 and 3 principal components in an attempt to achieve dimensional reduction.

In Figure 5.12 it is possible to see the influence on each of the principal components by channel. For the deoxy-Hb on the left-hand side, the correlation between the channels and the principal components is shown. There, one can see that the 1st principal component



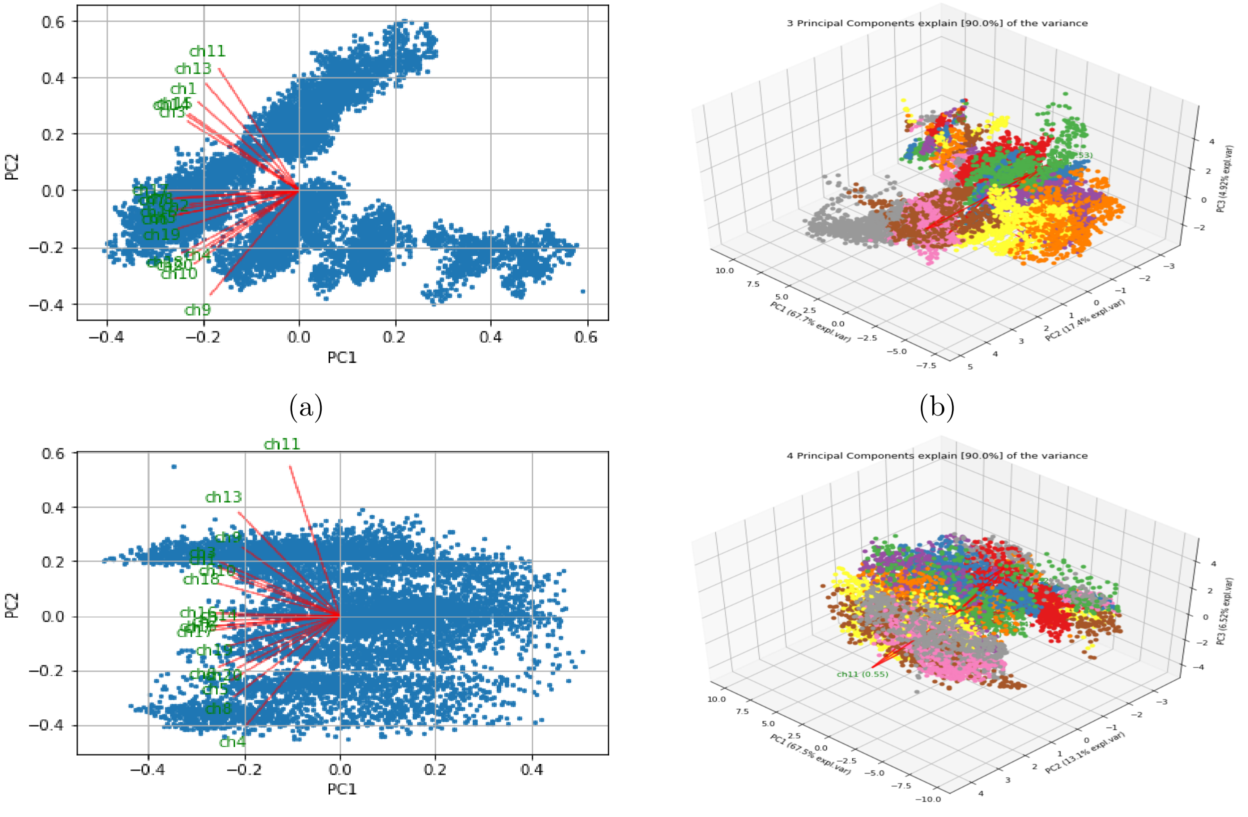
(a) (b)

Figure 5.10: *Plots of the explained variance ratio. The plots in* (*a*) *correspond to the deoxy-Hb dataset (LHS). There it is possible to see that only 3 principal components explain the 90% of the variance. In b*) *we see the case corresponding to the oxy-Hb (RHS). For this case, the first 4 principal components explain the 90% of the variance.*

has a uniform correlation with the 19 channels. For the 2nd principal component, we see that some channels are highly correlated to that component while some others are almost uncorrelated. Similar behavior is shown on the right-hand side for the oxy-Hb and its first four principal components. These figures show that most of the channels in both cases are mostly correlated to the first 3 principal components. If we recall from the correlation networks in Figs. 5.3 and 5.4 chan7 and 17 had the strongest correlation. However, this information is not visible by the principal component contributions from figure 5.12 since what we observe is that all that information might be already contained in the components with the largest variance.

In Figures 5.11 *a*) and *c*) and in Figure 5.13 one can see the plots of the first 2 principal components. In Fig. 5.11, we can see the principal component scores and the loading vectors while in 5.13 is just the loadings. From Figure 5.13, certain channels are more correlated to others because some loadings are located close to each other and far from others. It is possible to see that for both data sets there is a clear division between the channel correlations. All channels are well aligned along with the two principal components where most of the variance is accumulated. It is possible to observe that for some channels, the correlation is maintained due to the proximity of the loadings for the oxy-Hb and deoxy-Hb cases simultaneously. This indicates that such channels have a bigger presence in the experiment corresponding to very active and relevant regions of the brain.

After performing the PCA and create the biplots for the two principal components, it is not clear to see a neat formation of clusters that can give specific information about how correlated a pair of channels are. We can only give this information based on the distance the loadings have to each other as indicated in Figure 5.13. Although the method is convenient to qualitatively describe the correlation between each channel, it certainly does not provide any insight into possible nonlinear dependence.



(a)

(b)

(c)

(d)

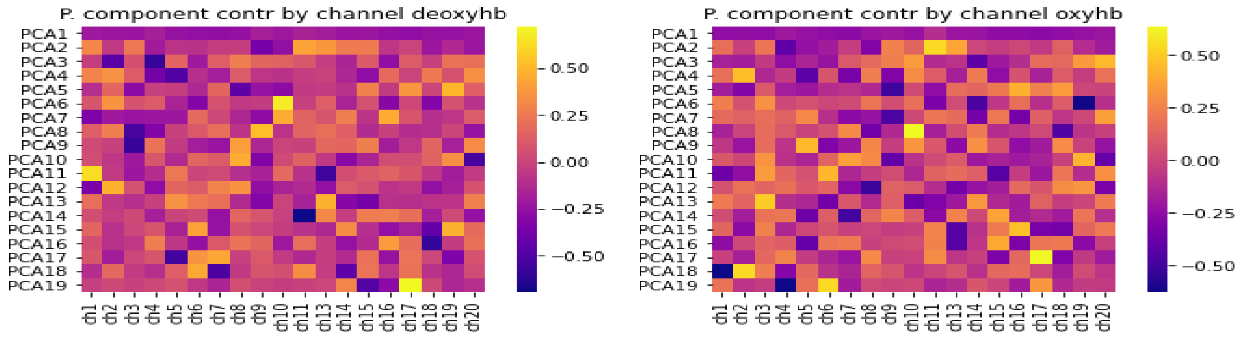
Figure 5.11: *Scatter plots of the first 2 and first 3 principal components for the deoxy-Hb and oxy-Hb datasets respectively. In* (*a*) *one can see the 2D biplot, and in* (*b*) *the 3D biplot for deoxy-Hb data. The subplots* (*c*), (*d*) *correspond to the oxy-Hb dataset.*

5.4 ICA for fNIRS data

We can also apply an ICA to the data sets in order to observe if those independent components can say anything about nonlinear behaviour. Figure 5.14 show the independent components for the deoxy-Hb and the oxy-Hb respectively after applying the `fast ICA` routine, as mentioned in Chapter 2. We observe from those figures that the independent components aren't ordered as in the PCA case and that the method itself forgets about the labeling of each channel, loosing the track of the physical location of such components.

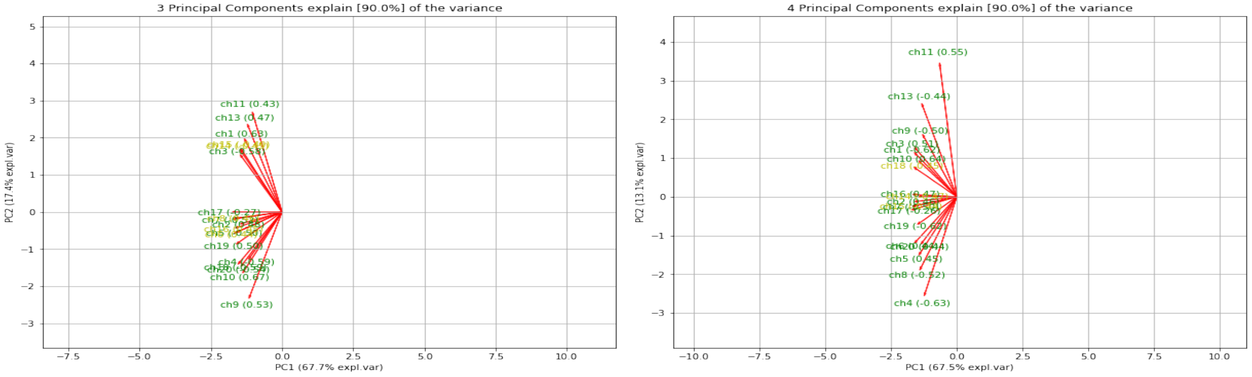
An interesting observation is that when we observe individually the oxy-Hb time series (Figure 5.14 (*b*)), the blue plots corresponding to the measurements form the experiment seem to have a very strong patter described by a clear pulse for most of those cases. This pulse is no longer observed once ICA is performed.

These independent components have zero linear correlation to one another. This means



(a) (b)

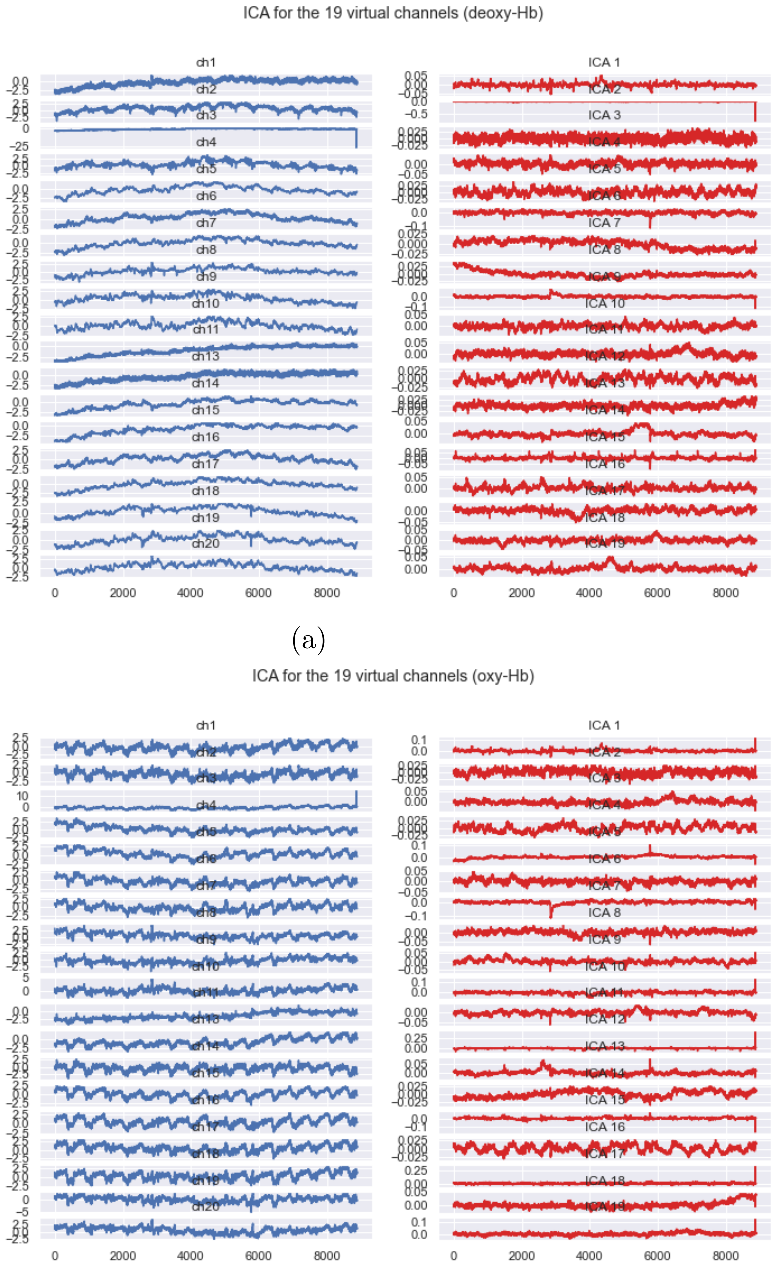
Figure 5.12: *Contribution of each principal component on each channel for fNIRS data. In (a) we observe the heatmap for the deoxy-Hb and in (b) we have the case for the oxy-Hb. The first principal component for both cases corresponds to the first row of each heatmap while the last principal component is placed at the bottom of each sub figure.*



(a) (b)

Figure 5.13: *Plots of the loadings for the deoxy-Hb (a) and oxy-Hb (b) without the scatter plot of the first two principal components.*

that its correlation matrix has all entries equal to zero except for the diagonal terms. However, if we calculate the normalized mutual information matrix for these independent components, we obtain non-zero contributions that reflect a more complicated share of information between pairs of channels. In Figure 5.15, we plot the normalized mutual information matrices for the independent components for both data sets. Since the diagonal term of the original matrix consist of only ones, it is more practical focusing the attention to the off-diagonal terms as in Figure 5.15. From this figure, we observe that the independent components maintain a strict positive normalized mutual information which reflects non-linearity. A thing certainly not captured by the correlation matrix. These contributions can also be attributed to the fact that the independent components are signals measured in a spatial region that might be susceptible to some error during the experiment. It is not clear from this analysis which factor is the one that make these contributions exists, but what we can observe here is the mutual information measure is still capturing contributions that are strictly nonlinear in a very useful manner.



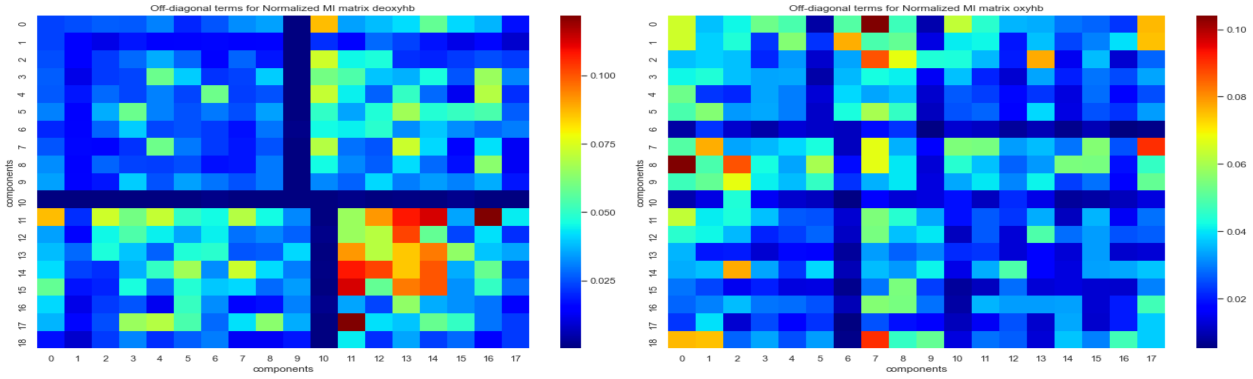
(a)

(b)

Figure 5.14: *ICA for the* 19 *virtual channels for (a) the deoxy-Hb and (b) the oxy-Hb case.*

5.5 Λ coefficient for fNIRS data

In Section 4.4 we presented the correlation and normalized mutual information matrices for fNIRS data. From Figure 5.1 it is possible to see that the normalized mutual information coefficient is significantly smaller than the Pearson coefficient when comparing a pair of signals in both oxy-Hb and deoxy-Hb cases. Clearly one can see that the pattern is somehow conserved for both types of matrices suggesting that nonlinear dependence are relevant



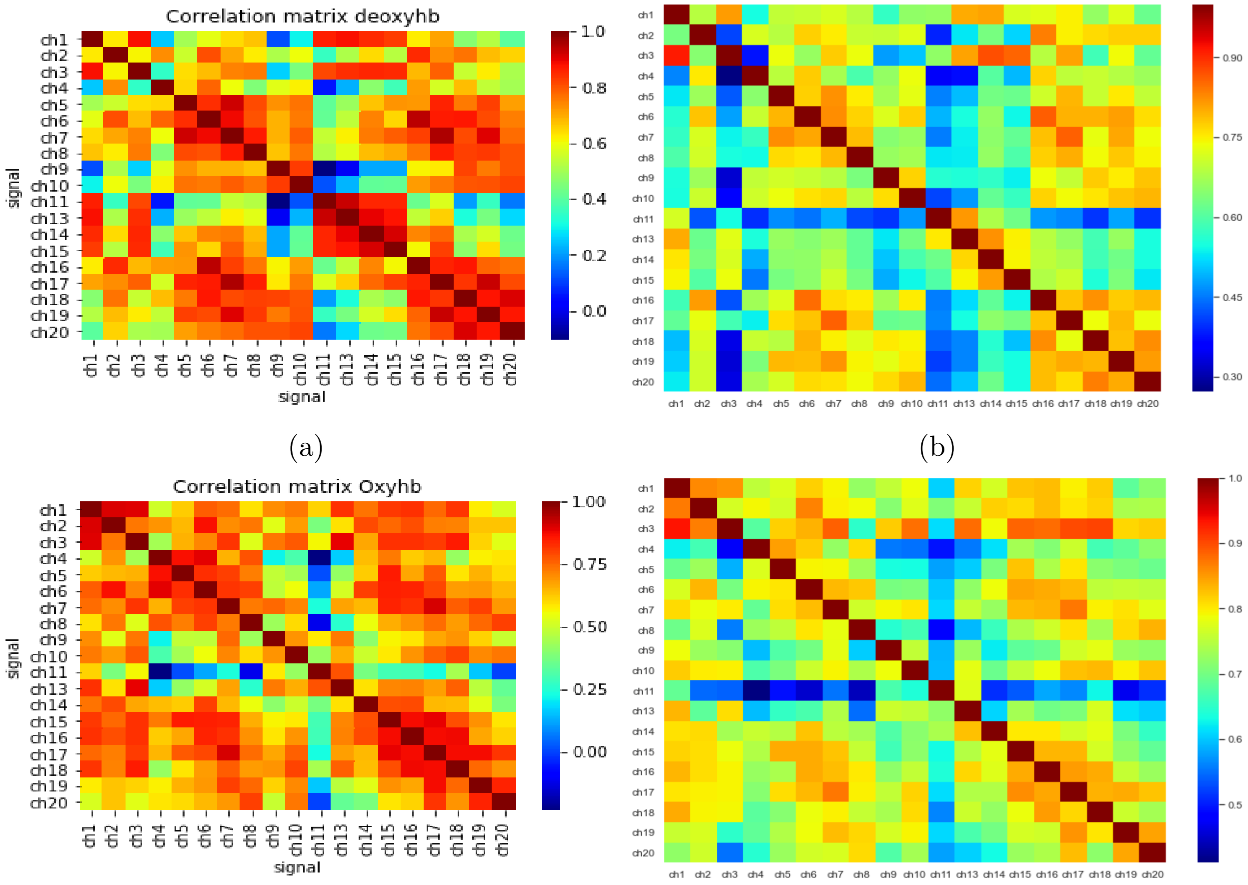
(a) (b)

Figure 5.15: *Normalized mutual information for the independent components. A) represent the case for deoxy-Hb, and b) for oxy-Hb case.*

when computing calculating the mutual information. What we do in this section is to measure nonlinear dependence using the mutual information to separate linear and nonlinear components of dependence according to Smith's method [38].

With the Λ coefficient justified in Chapter 3, we want to obtain the coefficient for each pair of signals in our fNIRS data. Figure 5.1 compares the correlation and normalized mutual information matrices yet, it shows only a qualitative way of making evident the nonlinear dependencies between signals. By computing the Λ-matrices for oxy-Hb and deoxy-Hb, we can have a better notion of how much linear dependency the fNIRS data sets have. The entries of these matrices correspond to the computation of the Λ coefficient by pair of signals according to eq. (2.31). These matrices will help understanding much better Figure 5.1 because the Λ-matrices manifestly distinguish the linear dependence of a pair of signals (Figure 5.16).

Figure 5.16 contains relevant information about the kind of correlation each signal has to another. In Figure 5.16(b) we observe that for the deoxy-Hb case, the signals with more linear dependence are colored in orange and red. Notice that the diagonal terms have $\Lambda = 1$ as expected since those terms correspond to a complete linear dependence. In such sub-figure (b), it is possible to observe the formation of certain cluster of orange regions. Such orange clusters have similar locations to the red ones in the correlation matrix counterpart. This is an indication that the most correlated zones in the deoxy-Hb data set have linear dependence. In particular, it is possible to see that most of the linear dependence between the signals concentrate in five clusters. An important remark is that the location of these five relevant clusters is exhibited in Figure 5.16(b) and in the normalized mutual information in Figure 5.1, with the difference that in Fig. 5.16(b) we know the *degree of linearity dependence* in a quantitative precise fashion. Another remarkable feature of this analysis is that the color bar in Fig.5.16(b) has a lower limit of $\Lambda_{ij} = 0.3$. This means that the technique is able to capture the linearity dependencies between the signals but with a lower threshold in which non-linearity is still present. The matrix in Fig. 5.16(b) is not perfectly diagonal compared to the mutual information and correlation counterparts due to the analysis in Step 3 from the method itself. The main reason can be explained due to the alternation of dependent



(a) (b)

(c) (d)

Figure 5.16: *Comparison between the correlation matrices and the* Λ *matrices for deoxy-Hb and Oxy-Hb data. The heatmaps in the top panel belong to the deoxy-Hb* $(a), (b)$ *set whereas the pair in the lower panel correspond to the oxy-Hb set* $(c), (d)$.

and independent variable X and Y in Step 3, added to the fact that the bias induced due to the number of bins might also increase the asymmetry in the matrix. These details are the ones responsible for the non-diagonal computation of the Λ matrix. In regards to the deoxy-Hb the pair Ch7-17 has a highly linear dependence which means that the networks above describe mostly linear contributions in the more activated regions.

For the oxy-Hb case, we achieve similar results but in a different regions. The correlation matrix in this case contain more orange and red regions spread along the different signals without a clear distinction of clusters (as in deoxy-Hb case). Nevertheless, when computing the normalized mutual information matrix the pattern is somehow reproduced but with less intensity. The Λ matrix for this case, maintain the highly correlated patterns as expected, indicating a larger value of Λ. This heatmap also has a lower threshold of around 0.4 indicating that nonlinear dependence is still manifest. In this case the pair Ch1-3 that has stronger presence in the networks result in a high linear dependence. We also observe the same issues regarding the diagonalization of this matrix as in the deoxy-Hb case.

The Λ coefficient method allows identifying the degree of linear dependence but it does not provide information about the kind of non linear dependence. It is important to mention that although a value of $\Lambda \sim 1$ implies linearity, it does not determine the strength of the linear correlation [38]. In Figure (5.16) we observe that regardless of the correlation coefficient, Λ can be high when it accounts most of the liner dependence. Likewise, it is also possible to obtain large correlation coefficients with low Λ indicating nonlinear dependence.

Chapter 6

Conclusions and future directions

The main aim of this study was to examine different methods that can provide a relevant interpretation of fNIRS data for studying the connectivity of the brain. We do this examination by using an information theory measure, mutual information, and compare the results with linear correlations and unsupervised learning techniques such as PCA, ICA, and dendrograms. We obtained that the creation of a functional connectivity network based on correlations and mutual information is an interesting approach to examine the nature of the dependencies between regions of the brain. However, these approaches to brain data need to explore with more specific experiments to verify their effectiveness and reliability. Compared to the unsupervised learning techniques we noticed that a mutual information approach might be more informative to explore the specific nature between regions of the brain, given its sensitivity of capturing linear and nonlinear phenomena. We observed that PCA is a good pre-processing data technique but it does not express nonlinear information. ICA on the other hand can be used to explore nonlinear dependencies after isolating the component paying the price of the location interpretation of the optodes. The use of dendrograms allows explaining the cluster formation given by the functional connectivity networks as a complementary visualization of the possible sub-networks formed based on a threshold.

The implementation presented in this project was tested for synthetic data and later applied to fNIRS data sets of a control group. As we have seen, the Pearson correlation coefficient is strictly a linear correlation measure that, despite being widely used for quantitative analysis of interrelation among variables, has several limitations when describing general data sets. One of these is the incapability of the coefficient to reflect other dependencies within the data that are not strictly linear. To explore non-linearity, we calculated the mutual information between time series under the consideration that the time series collected from the experiment are discrete random variables.

The implementation for obtaining the mutual information between these sets of data is based on a binning formula (Sturge's rule) that allows observing nonlinear features that are not directly seen in the correlation matrix. Although this is a straightforward approach for calculating mutual information, we encountered that the method is susceptible to certain bias due to the binning process. An example of this was when we compared the mutual information values between the binning method and the analytical result under the assumption of Gaussian variables. Here, we obtained a percentage error of 68.26% that we think can be reduced by a low-bias estimation method. Since there are explicit analytical results for the calculation of mutual information in the case of Gaussian variables, more general approaches for reducing the bias of the mutual information for a finite data set are still needed and represent an open question within the realm of information theory. The effect of the bias in our analysis might have been the factor that made the coefficients of the normalized mutual information to be lower than the Pearson correlation.

Based on the results from Chapter 5, we can conclude that mutual information is a great candidate to explore nonlinear dependencies on a data set because its formulation does not require linear dependencies as the ones assumed in the Pearson coefficient. On the other hand, mutual information does not uncover the specific type of dependencies between a pair of random variables. In other words, the measure is sensible enough to capture linear and nonlinear dependencies but without a clear separation of the degree of linearity or nonlinearity involved. By using a binning method, we observed the Frobenius norm of the mutual information matrix increases logarithmically as we increase the number of time steps when fixing the number of signals. This could be useful for more robust data sets with more measurements involved. Additionally, we also obtained that when the number of time steps of the signals is fixed, a power-law behavior describes the increase of the Frobenius norm as the number of signals increases. This is assumed to be a consequence of the definition itself of the mutual information for the discrete case, but also due to the bias of the joint distribution. A question that needs further exploration is if the reduction of the bias for the joint probability distribution might change the type of behavior of the Frobenius norm in a way that is possible to maintain the norm small enough as we increase the time-steps or the number of signals.

On a network model as the ones shown in Figures 5.4 and 5.6, we observe that for the oxygenated hemoglobin the Pearson coefficient indicates a strong correlation between most of the virtual channels involved during the experiment. This indicates that most of the mapped regions of the brain are activated during the measurement process with the strongest presence along the bridge between channel 7 and channel 17. Such feature is also exhibited by the mutual information but with the difference that other connections are also relevant, suggesting a more complex biological process endeavor. For the deoxygenated case Figures 5.3 and 5.5, we indicate that the participant might be right-handed since most of the connections with stronger correlations and mutual information are displayed on the left-hand side. The latter is not observed for the oxy-Hb networks which indicate that oxygenated and deoxygenated hemoglobin have a different physiological interpretation as expected. The network models provide an advantage over a simple matrix calculation given the possibility of localizing the nodes geometrically. However, the more dense the network is (i.e. the more optodes are present in the experiment) the more difficult it is to discern specific connections between activated regions.

Concerning the unsupervised techniques, we obtained that PCA allows determining only linear contributions without providing relevant information beyond that. In this sense, the calculation of the correlation and the normalized mutual information matrices turns out to be a better approach since we can maintain the explicit label of channels and therefore have knowledge about the topology of the optodes in future analysis. In contrast, ICA results to be a more suitable technique compared to PCA. Although ICA also loses the information about the physical distribution of the optodes, it enables observing pure nonlinear dependencies between the independent components. This feature is fundamental to examine more complicated connections and interactions between different brain regions.

The information of these data sets about the linear dependence displayed in Figure 5.16, is a useful method to verify the linear dependence of a pair of signals. However, in the case where non-linearity is dominant, it is not possible to know what kind of relation the pair of variables have. Although Λ is a coefficient that, to a certain degree informs about linear dependence using mutual information, it does not inform about the type of nonlinear dependencies present in the data. To determine if this calculation of Λ matrix is telling relevant information and therefore be useful in a more systematic way, more tests on fNIRS data sets are needed to verify if the method describes in this thesis, will provide better insights about brain activity that are not evident at the level of the correlation matrix and the mutual information matrix. We noticed from Figure 5.16 that most of the zones with higher correlation and mutual information coefficients reflect a high value of Λ meaning that most of the dependencies are linear. This was expected because these zones are evident when seeing the footprint in Figure 5.1. We notice that in the Λ matrices, channel 11 reflects for oxy and deoxy case, a low coefficient suggesting that in that channel there might be more nonlinear behavior. This suggestion is supported by the fact that channel 11 was the one with negative correlations in both cases.

Additionally, since the bias correction for the mutual information has a big influence on our results, more explorations on the way of estimating entropy measures are needed to compare and validate more systematically the results of this thesis. K-nearest neighbor is an alternative method recently used to study mutual information with a low bias and can perhaps provide more accurate results than the ones obtained here. Lastly, it would be interesting to explore other information theory measures such as conditional mutual information and possibly, more sophisticated tools like entanglement on network generation.

Appendix A

Information theory remarks

In this appendix we present some interesting results from differential entropy. As we will see, there is a special case where there is a direct connection between correlation coefficient and mutual information.

Entropy of a normal distribution

For a *continuous* random variable with a normal distribution, $X \sim N(0, \sigma^2)$, we know that the probability density function (*pdf*) is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

therefore,

$$h(X) = -\int_{-\infty}^{\infty} f(x) \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)\right) dx = -\int_{-\infty}^{\infty} f(x) \left[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{x^2}{2\sigma^2}\underbrace{\log(e)}_{1}\right] dx$$

$$= -\int_{-\infty}^{\infty} f(x) \left[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{x^2}{2\sigma^2}\right] dx = \int_{-\infty}^{\infty} f(x) \left[\frac{1}{2}\log(2\pi\sigma^2)\right] dx + \underbrace{\int_{-\infty}^{\infty} f(x) \left(\frac{x^2}{2\sigma^2}\right) dx}_{\frac{1}{2}}$$

$$\therefore \boxed{h(X) = \frac{1}{2}\log(2\pi e\sigma^2)} \tag{A.1}$$

which is a concave function of σ^2.

For a multivariate normal distribution $\vec{x} \sim N(0, \mathbb{K})$ if $\vec{x} = (X_1, \ldots, X_d)^T$ is a d-dimensional Gaussian vector with zero mean and covariance matrix $\mathbb{K}$, then the *pdf* is

$$f(\vec{x}) = \frac{1}{(\sqrt{2\pi})^d |\mathbb{K}|^{1/2}} \exp\left(-\frac{1}{2}\vec{x}^T \mathbb{K}^{-1} \vec{x}\right)$$

Therefore

$$h(\vec{x}) = -\int f(\vec{x}) \left[-\frac{1}{2}\log\left((2\pi)^d|\mathbb{K}|\right) - \frac{\log(e)}{2}\left(\vec{x}^T\mathbb{K}^{-1}\vec{x}\right)\right] d\vec{x}$$
$$= \frac{1}{2}\log\left((2\pi)^d|\mathbb{K}|\right) + \frac{d\log(e)}{2}$$

$$\therefore \boxed{h(\vec{x}) = \frac{1}{2}\log\left((2\pi e)^d|\mathbb{K}|\right)} \tag{A.2}$$

Mutual information between correlated Gaussian variables

If we consider that the two Gaussian variables have a correlation coefficient ρ, let $(X,Y)^T$ be a zero-mean Gaussian random vector with covariance matrix $\mathbb{K} = \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}$, where ρ is given as in 2.11. In this case, $h(X) = h(Y)$ given by A.1 and $h(X,Y)$ obtained from A.2. Therefore in this case:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) = \log(2\pi e\sigma^2) - \frac{1}{2}\log\left((2\pi e)^2|\mathbb{K}|\right)$$
$$= \log(2\pi e\sigma^2) - \frac{1}{2}\log\left((2\pi e)^2\sigma^4(1-\rho^2)\right) =$$

$$\therefore \boxed{I(X,Y) = -\frac{1}{2}\log\left(1-\rho^2\right)} \tag{A.3}$$

Appendix B

Matrix norms

In this appendix we will introduce some definitions that will help justifying the Frobenius norm used in the implementation.

For matrices in $\mathbb{C}^{i,j}$ on a field $\mathbb{C}$, we can define define the following:

Definition (Matrix norm)**.** A matrix norm is a function $\|\cdot\| : \mathbb{C}^{m,n} \to \mathbb{C}$ such that if $\forall$ matrices $A, B \in \mathbb{C}^{m,n}$ and all scalars $\alpha \in \mathbb{C}$ the following conditions are satisfied

- $\|A\| \geq 0$ with the equality iff $A = 0$
- $\|\alpha A\| = |\alpha| \|A\|$
- $\|A + B\| \leq \|A\| + \|B\|$

In other words, a matrix norm is a vector norm on the finite dimensional vector space of $m \times n$ matrices.

Theorem B.0.1 (Equivalent norms)**.** *All matrix norms are equivalent. If $\|\cdot\|$ and $\|\cdot\|'$ are 2 matrix norms on $\mathbb{C}^{m,n}$, then there exist $\mu, M > 0$ such that $\mu\|A\| \leq \|A\|' \leq M\|A\|$ $\forall A \in \mathbb{C}^{m,n}$. Additionally, a matrix norm is a continuous function $\|\cdot\| : \mathbb{C}^{m,n} \to \mathbb{R}$.*

Definition (Submultiplicative matrix property)**.** For square matrices $A, B \in \mathbb{C}^{m,m}$ and a matrix norm we say the norm is submultiplicative if

$$\|AB\| \leq \|A\|\|B\|$$

.

Definition (Consistent Matrix norm)**.** We say a matrix norm is *consistent* if is a submultiplicative matrix norm defined $\forall m, n \in N$

Definition (Subordinate matrix norm)**.** We say a matrix norm $\|\cdot\|$ on $\mathbb{C}^{m,n}$ is *subordinate* to a vector norm $\|\|_\alpha$ on $\mathbb{C}$ and $\|\|_\beta$ on $\mathbb{C}^m$ if

$$\|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\|_\alpha,$$

$\forall$ $A \in \mathbb{C}^{m,n}$ and $\mathbf{x} \in \mathbb{C}^n$.

Definition (Frobenius matrix norm)**.** The frobenius norm is a consistent matrix norm subordinate to the Eucledian vector norm.

Appendix C

Anscombe's Quartet

Introduced in Chapter 2, here we explicitly present the Anscombe's Quartet. This quartet consist of data sets that have very similar statistical properties but display different graphic behavior. By using this quartet, it is clear to see that not all the information related to the dependencies between the data sets can be given by linear measures. As Figure (C.1) shows, mutual information is a good candidate to explore non linearity.

x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Table C.1: *Anscombe's Quartet. In this table we present four data sets with similar statistical properties but with different graphs displayed in Figure (C.1)*

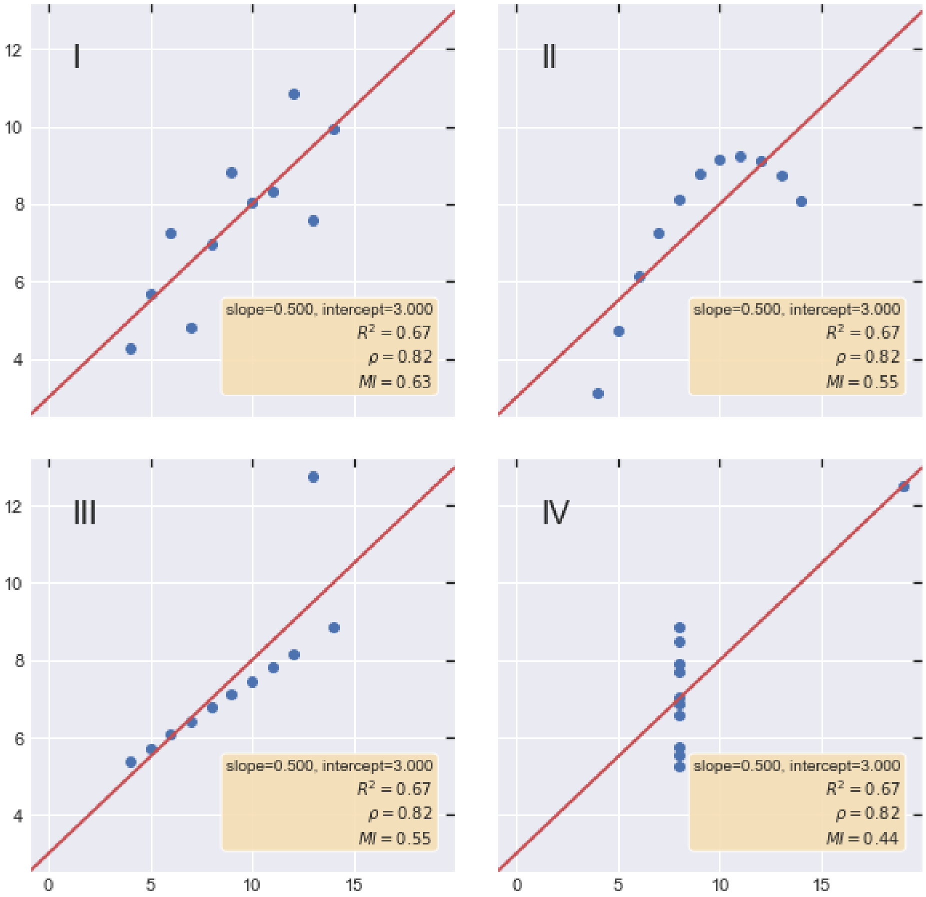


Figure C.1: *Statistical features and mutual information for Anscombe's quartet. In this case the joint probability used 3 bins for its computation due to the number of points in each data set. We noticed that although we have similar statistical features for each data set, the mutual information is generally different in each case.*

Appendix D

Code

```
from pylab import *
import numpy as np
import pandas as pd
import scipy
import pylab as pl
import matplotlib
from matplotlib.ticker import ScalarFormatter, FormatStrFormatter
import sys
from numpy import *
import networkx as nx
import seaborn as sn
############Load data
Datafnirs= loadtxt('NIRSdeoxyhb.txt') #use NIRSoxyhb.txt for oxyHb case
#here is just a visualization of the dataset using Pandas
dfi = pd.DataFrame(Datafnirs ,columns=['ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10','ch11','ch12','ch13','ch14','ch15','ch16','ch17','ch18','ch19','ch20'])
#removing the NAN
array1 = Datafnirs
nan_array = np.isnan(array1)
not_nan_array = ~ nan_array
array2 = array1[not_nan_array]
Datafnirs2=np.reshape(array2, (len(Datafnirs),len(array1[0])-1))
# here I have eliminated channel 12 that was compromised
dfi2 = pd.DataFrame(Datafnirs2 ,columns=['ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10', 'ch11','ch13','ch14','ch15','ch16','ch17','ch18','ch19','ch20'])
# calculate the correlation matrix
corr = dif2.corr()
# plot the heatmap for correlation matrix
sn.heatmap(corr,xticklabels=corr.columns, yticklabels=corr.columns,cmap=plt.cm.jet)
plt.title('Correlation matrix deoxyhb')
plt.xlabel('signal')
plt.ylabel('signal')
```

Listing D.1: Routine for correlation matrices

```
datadeoxy=Datafnirs2.transpose()
```

```
#here I create the joint prob matrices for each pair of signals
veam=[]
for i in range(len(datadeoxy)):
   # veam2=[]
    for j in range(len(datadeoxy)):
        nxbins, nybins = (np.log2(len(datadeoxy[0]))+1,np.log2(len(datadeoxy[0]))+1)
        #with this we calculate all joint
        outp, xedges, yedges = np.histogram2d(datadeoxy[i],datadeoxy[j],bins=(nxbins,nybins))
        outp /= np.sum(outp)
        veam.append(outp)
########################################################
#here we define the function for the mutual information
#this function eats the joint probability matrices from the array: veam
def mutinfo2(jopmat):
    #jpmatr=np.array([x1,x2])
    #marginals of X:
    #returns an array with all the marginal prob of X
    Pmx=np.sum(jopmat, axis=0)
    #marginals of Y:
    #returns an array with all the marginal prob of Y
    Pmy=np.sum(jopmat,axis=1)

    mutualinf=0
    #for i in range(len(pru)):
    for i in range(len(jopmat)):
        for j in range(len(Pmx)):
            if jopmat[i][j]==0.:
                pass
            else:

                mutualinf+=jopmat[i][j]*np.log2(jopmat[i][j]/(Pmx[j]*Pmy[i]))
    return mutualinf
###########################################################
qpd=[]
for i in range(len(veam)):
    ora=mutinfo2(veam[i])
    qpd.append(ora)
#print(qpd)
MImat=np.reshape(qpd,(len(datadeoxy),len(datadeoxy)))
#print(MImat)
###########################################################
ax = sn.heatmap(MImat, cmap=plt.cm.jet)
plt.title(’Numer MI matrix’)
plt.xlabel(’signal’)
plt.ylabel(’signal’)
##########################################################
#To normalize the matrix
initmat=(len(datadeoxy),len(datadeoxy))
normMI=np.zeros(initmat)
for ii in range(len(datadeoxy)):
    for jj in range(len(datadeoxy)):  ### a_ij = a_ij/sqrt(a_ii * a_jj)
```

```
        normMI[ii,jj]=MImat[ii,jj]/(np.sqrt(MImat[ii,ii]*MImat[jj,jj]))
#print(normMI)
ax = sn.heatmap(normMI,  cmap=plt.cm.jet)
plt.title('Normalized MI matrix deoxyhb')
plt.xlabel('signal')
plt.ylabel('signal')
```

Listing D.2: Routine for mutual information matrices

```
MI = pd.DataFrame(normMI, columns=['ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10', 'ch11','ch13','ch14','ch15','ch16','ch17','ch18','ch19','ch20'])
MI.index=['ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10', 'ch11','ch13','ch14','ch15','ch16','ch17','ch18','ch19','ch20']
##########################################################
#### Transform it in a links data frame (3 columns only):
links = MI.stack().reset_index()
links.columns = ['var1', 'var2', 'value']
###########################################################
plt.figure(figsize=(12,8))
G=nx.from_pandas_edgelist(links, 'var1', 'var2', edge_attr='value') #add attributes of the weight
widths = nx.get_edge_attributes(G, 'value')
nodelist = G.nodes()
#### Transform it in a links data frame (3 columns only):
links = MI.stack().reset_index()
links.columns = ['var1', 'var2', 'value']
#### Keep only correlation over a threshold and remove self correlation (cor(A,A)=1)
links_filtered=links.loc[ (links['value'] > 0) & (links['var1'] != links['var2']) ]
#### Build the graph
G=nx.from_pandas_edgelist(links_filtered, 'var1', 'var2', edge_attr='value')
##### Plot the network:
plt.figure(figsize=(12,8))
#nodes
nx.draw_networkx_nodes(G,posit,
                       nodelist=nodelist,
                       node_size=1500,
                       node_color='black',
                       alpha=0.7)
#node labels
nx.draw_networkx_labels(G, posit,
                        labels=dict(zip(nodelist,nodelist)),
                        font_color='white')
###edges
mcl = nx.draw_networkx_edges(
    G, posit, edge_cmap=cm.Blues, width=5,
    edge_color=[G[u][v]['value'] for u, v in G.edges])
labels = nx.get_edge_attributes(G,'value')
plt.box(False)
plt.colorbar(mcl)
plt.show()
```

Listing D.3: Routine for creating the networks. In this case we present the routine for normalized mutual information

```
### Replace 'MI' instead of 'corr' for mutual information
sn.clustermap(corr, metric="euclidean",  method="ward", cmap="mako")
# Show the graph
plt.show()

import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(corr,metric="euclidean", method  =
    "ward"))
plt.title('Correlation Dendrogram deoxy-Hb')
plt.xlabel('channels')
plt.ylabel('Euclidean distances')
plt.show()
```

Listing D.4: Dendrograms

```
# Here we have the data standarized ready to apply PCA
#for that, we use the scale() function form sklearn
from sklearn.preprocessing import StandardScaler
y = StandardScaler().fit_transform(dfi2)
y = pd.DataFrame(y, columns=['ch1','ch2','ch3','ch4','ch5','ch6','ch7','
   ch8','ch9','ch10',
'ch11','ch13','ch14','ch15','ch16','ch17','ch18','ch19','ch20'])
####Get the PCA components
from sklearn.decomposition import PCA
pcamodel = PCA(n_components=19)
pca = pcamodel.fit_transform(y)
ax = sn.heatmap(pcamodel.components_,
                 cmap='plasma',#'YlGnBu',
                 yticklabels=[ "PCA"+str(y) for y in range(1,pcamodel.
   n_components_+1)],
                 xticklabels=list(y.columns),
                 cbar_kws={"orientation": "vertical"})
ax.set_aspect("equal")
plt.title('P. component contr by channel deoxyhb')
# Plot cumulative explained variance
from pca import pca
# Initialize up to the number of component that explains 90% of the
   variance.
model = pca(n_components=0.9)
# Fit transform
results = model.fit_transform(y)
#plot figure
fig, ax = model.plot()
```

Listing D.5: Routine for PCA

```
#Standardize the data
dfi2 /= dfi2.std(axis=0)
# Compute ICA
```

```python
ica = FastICA(n_components=19)
S_ = ica.fit_transform(dfi2)  # Reconstruct signals
A_ = ica.mixing_  # Get estimated mixing matrix
#Convert to Pandas frame
ind_ca= pd.DataFrame(S_ ,columns=['ICA1','ICA2','ICA3','ICA4','ICA5','ICA6','ICA7','ICA8','ICA9','ICA10', 'ICA11','ICA12','ICA13','ICA14','ICA15','ICA16','ICA17','ICA18','ICA19'])
#Plot channels and ICA's
ka=len(Datafnirs2.T)
fig, axs = plt.subplots(ka,2)
for i in range(ka):
    axs[i,0].plot(Datafnirs2.T[i])
    axs[i,0].set_title('ch%d' %(i+1))

    axs[i,1].plot(S_.T[i], 'tab:red')
    axs[i,1].set_title('ICA %d' %(i+1))
fig.tight_layout()
```

Listing D.6: Routine for ICA

```python
def lambda_(x,y):
    ###Define the function
    def recta(x, a, b):
        return (a*x)+b
    #Fit for the parameters a,b,c of the function func
    popt, pcov=curve_fit(recta,x,y)
    #this is the fit data
    fit_data=recta(x,*popt)
    #The residuals
    res=y-recta(x,*popt)
    #r^2
    r2=r2_score(y,fit_data)
    #####Calculate the cdf for the residuals:
    G_z = scipy.stats.uniform.cdf(res) # calculate the cdf using uniform distribution
    ###### we can plot the cdf
    #sn.lineplot(res, G_z)
    #plt.show()
    ####### Now we do the quantile transform Y'=F_{y}^{-1}(G(z))
    from sklearn.preprocessing import QuantileTransformer
    ##### reshape data to have rows and columns
    data = G_z.reshape((len(G_z),1))
    ###### quantile transform the raw data
    quantile = QuantileTransformer(n_quantiles=len(data), output_distribution='normal') #using normal distrib
    data_trans = quantile.fit_transform(data)
    ###### concatenate the infromation to use it as Y' as an array in the MI matrix
    y_p=np.concatenate(data_trans)
    lam=1-(mutinf_mat(x,y_p)[0][1]/mutinf_mat(x,y)[0][1])
    return lam
#Create a list for the matrix
bv=[]
for g in range(len(lista2)):
    m=[]
    for i in lista2[g]:
        ag=lambda_(i[0],i[1])
        m.append(ag)
    bv.append(m)
#Plot the matrix as heatmap
sn.heatmap(bv,xticklabels=corr.columns, yticklabels=corr.columns,cmap=plt.cm.jet)
```

Listing D.7: Routine for Λ coefficients for fNIRS data

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from sklearn.metrics import r2_score
import seaborn as sn
import random
import math

x = [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5]
y1 = [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68]
y2 = [9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74]
y3 = [7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73]
x4 = [8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8]
y4 = [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89]


datasets = {
    ’I’: (x, y1),
    ’II’: (x, y2),
    ’III’: (x, y3),
    ’IV’: (x4, y4)
}
#create the frames
fig, axs = plt.subplots(2, 2, sharex=True, sharey=True, figsize=(10, 10),
                        gridspec_kw={’wspace’: 0.1, ’hspace’: 0.1})
#loop for plotting the data points according to the dictionary
for ax, (label, (x, y)) in zip(axs.flat, datasets.items()):
    ax.text(0.1, 0.9, label, fontsize=20, transform=ax.transAxes, va=’top’)
    ax.tick_params(direction=’in’, top=True, right=True)
    ax.plot(x, y, ’o’)
    #########Fit for the parameters a,b,c of the function func
    # linear regression
    p1, p0 = np.polyfit(x, y, deg=1)  # slope, intercept
    ax.axline(xy1=(0, p0), slope=p1, color=’r’, lw=2)

    textstr=’\n’.join((’slope=%5.3f, intercept=%5.3f’ %tuple(popt) ,
    r’$R^2=%.2f$’ % (r2, ),
    r’$\rho=%.2f$’ %(np.corrcoef(x,y)[0][1],),
                  r’$MI=%.2f$’ %(mutinf_mat(x,y)[0][1],)))

    props = dict(boxstyle=’round’, facecolor=’wheat’, alpha=0.9)
    ax.text(0.95, 0.07, textstr, transform=ax.transAxes, fontsize=10,
            horizontalalignment='right', bbox=props)
```

Listing D.8: Routine for Figure (C.1)

Bibliography

[1] Arild Berg et al. "Health promoting experiences in urban green space: A case study of a co-design toolkit based on feedbacks from fNIRS, IoT and game probes". In: *13th EAI International Conference on Pervasive Computing Technologies for Healthcare-Demos and Posters*. European Alliance for Innovation (EAI). 2019.

[2] David A Boas et al. *Twenty years of functional near-infrared spectroscopy: introduction for the special issue*. 2014.

[3] Shannon Burns. *fNIRS Bootcamp*. URL: `https://www.youtube.com/channel/UCDYOrRBdlTx7E_QAimgX2SQ/playlists` (visited on 07/24/2019).

[4] Shannon Burns and Matthew Lieberman. "The Use of fNIRS for Unique Contributions to Social and Affective Neuroscience". In: (July 2019). DOI: `10.31234/osf.io/kygbm`.

[5] Shannon M Burns et al. "A functional near infrared spectroscopy (fNIRS) replication of the sunscreen persuasion paradigm". In: *Social cognitive and affective neuroscience* 13.6 (2018), pp. 628–636.

[6] Richard B. Buxton. *Introduction to Functional Magnetic Resonance Imaging: Principles and Techniques*. 2nd ed. Cambridge University Press, 2009. DOI: `10.1017/CBO9780511605505`.

[7] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006. ISBN: 0471241954.

[8] Leon Danon et al. "Comparing community structure identification". In: *Journal of statistical mechanics: Theory and experiment* 2005.09 (2005), P09008.

[9] DT Delpy and M Cope. "Quantification in tissue near–infrared spectroscopy". In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 352.1354 (1997), pp. 649–659.

[10] Terje Gjovaag, Peyman Mirtaheri, and Inger Marie Starholm. "Carbohydrate and fat oxidation in persons with lower limb amputation during walking with different speeds". In: *Prosthetics and orthotics international* 42.3 (2018), pp. 304–310.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[12] Fabian Herold et al. "Applications of functional near-infrared spectroscopy (fNIRS) neuroimaging in Exercise–Cognition science: a systematic, Methodology-Focused review". In: *Journal of clinical medicine* 7.12 (2018), p. 466.

[13] Elizabeth MC Hillman. "Coupling mechanism and significance of the BOLD signal: a status report". In: *Annual review of neuroscience* 37 (2014), pp. 161–181.

[14] Aapo Hyvärinen and Erkki Oja. "Independent component analysis: algorithms and applications". In: *Neural networks* 13.4-5 (2000), pp. 411–430.

[15] Farzin Irani et al. "Functional Near Infrared Spectroscopy (fNIRS): An Emerging Neuroimaging Technology with Important Applications for the Study of Brain Disorders". In: *The Clinical Neuropsychologist* 21.1 (2007). PMID: 17366276, pp. 9–37. DOI: `10.1080/13854040600910018`. URL: `https://doi.org/10.1080/13854040600910018`.

[16] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[17] Gareth James et al. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL: `https://faculty.marshall.usc.edu/gareth-james/ISL/`.

[18] Yun Jiao et al. "Independent component analysis of event-related functional near-infrared spectroscopy (fNIRS)". In: *2008 International Conference on BioMedical Engineering and Informatics*. Vol. 2. IEEE. 2008, pp. 440–444.

[19] Claude Julien. "The enigma of Mayer waves: facts and models". In: *Cardiovascular research* 70.1 (2006), pp. 12–21.

[20] Bernd André Jung and Matthias Weigel. "Spin echo magnetic resonance imaging". In: *Journal of Magnetic Resonance Imaging* 37.4 (2013), pp. 805–817. DOI: `10.1002/jmri.24068`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/jmri.24068`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/jmri.24068`.

[21] Donald E. Knuth. "Fundamental Algorithms". In: Addison-Wesley, 1973. Chap. 1.2.

[22] Dominic Langlois, Sylvain Chartier, and Dominique Gosselin. "An introduction to independent component analysis: InfoMax and FastICA algorithms". In: *Tutorials in Quantitative Methods for Psychology* 6.1 (2010), pp. 31–38.

[23] Andrew F Mills et al. "Principles of quantitative MR imaging with illustrated review of applicable modular pulse diagrams". In: *RadioGraphics* 37.7 (2017), pp. 2083–2105.

[24] David J Morin. *Probability: For the Enthusiastic Beginner*. Createspace Independent Publishing Platform, 2016.

[25] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[26] Noman Naseer and Keum-Shik Hong. "fNIRS-based brain-computer interfaces: a review". In: *Frontiers in human neuroscience* 9 (2015), p. 3.

[27] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: `10.1017/CBO9780511976667`.

[28] Amy R Nippert, Kyle R Biesecker, and Eric A Newman. "Mechanisms mediating functional hyperemia in the brain". In: *The Neuroscientist* 24.1 (2018), pp. 73–83.

[29] Soheyl Noachtar and Jan Rémi. "The role of EEG in epilepsy: a critical review". In: *Epilepsy & Behavior* 15.1 (2009), pp. 22–33.

[30] E. Oja, S. Harmeling, and L. Almeida. "Independent component analysis and beyond". In: *Signal Processing* 84.2 (Feb. 2004), pp. 215–216.

[31] Paola Pinti et al. "Current Status and Issues Regarding Pre-processing of fNIRS Neuroimaging Data: An Investigation of Diverse Signal Filtering Methods Within a General Linear Model Framework". In: *Frontiers in Human Neuroscience* 12 (2019), p. 505. ISSN: 1662-5161. DOI: `10.3389/fnhum.2018.00505`. URL: `https://www.frontiersin.org/article/10.3389/fnhum.2018.00505`.

[32] Paola Pinti et al. "Current status and issues regarding pre-processing of fNIRS neuroimaging data: an investigation of diverse signal filtering methods within a general linear model framework". In: *Frontiers in human neuroscience* 12 (2019), p. 505.

[33] Marco A Pinto-Orellana et al. "A hemodynamic decomposition model for detecting cognitive load using functional near-infrared spectroscopy". In: *arXiv preprint arXiv:2001.08579* (2020).

[34] Valentina Quaresima and Marco Ferrari. "Functional near-infrared spectroscopy (fNIRS) for assessing cerebral cortex function during human behavior in natural/social situations: a concise review". In: *Organizational Research Methods* 22.1 (2019), pp. 46–68.

[35] Sheldon Ross. *A first course in probability.* Pearson, 2014.

[36] Jette Schack et al. "Increased prefrontal cortical activation during challenging walking conditions in persons with lower limb amputation–an fNIRS observational study". In: *Physiotherapy Theory and Practice* (2020), pp. 1–11.

[37] Habib Sherkat, Terje Gjvaag, and Peyman Mirtaheri. "Experimental investigation on the light transmission of a textile-based over-cap used in functional near-infrared spectroscopy". In: *European Conference on Biomedical Optics.* Optical Society of America. 2019, 11074_70.

[38] Reginald Smith. "A mutual information approach to calculating nonlinearity". In: *Stat* 4.1 (2015), pp. 291–303.

[39] Stephen M Smith and Thomas E Nichols. "Statistical challenges in "big data" human neuroimaging". In: *Neuron* 97.2 (2018), pp. 263–268.

[40] NIRx Medical Technologies. *nirsLAB Manual Release Notes.* URL: `https://nirx.de/downloads/nirsLAB/nirsLAB_AllChapters.pdf` (visited on 09/30/2020).

[41] Jake VanderPlas. *Python data science handbook: Essential tools for working with data.* " O'Reilly Media, Inc.", 2016.