Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

Score-based Bayesian networks for the discovery of effective connectivity in fMRI data with the use of the Balloon model

Author: Jorge Díez García-Victoria
Supervisors: Pedro María Larrañaga Múgica & María Concepción Bielza Lozoya

Madrid, July 2021

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial

Título: Score-based Bayesian networks for the discovery of effective connectivity in fMRI data with the use of the Balloon model

July, 2021

Author: Jorge Díez García-Victoria
Supervisors: Pedro María Larrañaga Múgica & María Concepción Bielza Lozoya
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Acknowledgements

It has been some time since I started my studies, and I would have never believed that I would once get to where I am right now. Many people have come in and out of my life in these 6 years, but they have been a crucial part of my life, even if they are not here any longer. But there is one pillar who has always been at mi side and supported me all the way: my family. I may have not been with them physically, but they have given me all the tools and support that I have ever needed. Once again, thanks to everyone.

I would also like to thank Stephen Smith and Mark Woolrich for the C++ and MATLAB source code for the Balloon Model, Rubén Sánchez-Romero for helping me with the MATLAB code and Jeanette Mumford for the FSL filter implementation in MATLAB.

This work has been partially supported by the BBVA Foundation (2019 Call) through the "Score-based nonstationary temporal Bayesian networks. Applications in climate and neuroscience" project.

Resumen

La estimación de la conectividad funcional y efectiva a partir de datos de fMRI es de especial interés, ya que puede ayudar a los investigadores a comprender el flujo de información en el cerebro, desarrollar teorías y ser de gran ayuda en estudios de enfermedades neurodegenaritvas. Por esta razón, existe abundante trabajo en este campo, con un enfoque en el uso de diferentes algoritmos de aprendizaje automático para estimar dicha conectividad. Estudios conocidos han usado datos fMRI simulador mediante el Balloon model, junto con algoritmos de aprendizaje de estructura de redes Bayesianas para estimar las diferentes conectividades. Sin embargo, la naturaleza de los algoritmos utilizados en estos estudios es limitada, y los datos simulados en los estudios siempre se basan en fMRI de estado de reposo.

En este trabajo, probamos algoritmos de aprendizaje de estructura de red bayesiana de estado del arte que no se han utilizado antes en el campo del análisis de datos fMRI, junto con la última versión de la familia de algoritmos que han brindado los mejores resultados en el pasado: los algoritmos LINGAM. También utilizamos el Balloon model para generar nuevos datos basados en el modelo boxcar para estudios fMRI orientados a tareas, que hasta donde sabemos, no se ha realizado en la literatura hasta ahora. Luego, probamos los algoritmos para identificar sus capacidades para discernir correctamente la conectividad funcional y efectiva en diferentes circunstancias, como la duración de la sesión fMRI, y comparamos sus resultados en datos fMRI simulados en estado de reposo y orientados a tareas.

Demostramos que para las mismas circunstancias y algoritmos, los resultados varían si los datos se generan a partir fMRI en estado de reposo u orientados a tareas. También demostramos que incluso si algunos algoritmos ven un aumento en la conectividad correctamente deducida con sesiones de mayor duración, número de sujetos y tiempo de repetición del escaneo, este aumento no es lo suficientemente significativo como para justificar el uso de sesiones de mayor duración, tiempos de repetición, etc. Finalmente, demostramos que DIRECTLINGAM es el mejor algoritmo a utilizar, tanto por su alta detección de conexiones y dirección como por su robustez.

Abstract

The estimation of both functional and effective connectivity from fMRI data is of special interest, since it can help researchers understand the flow of information in the brain, develop theories and be of great aid in studies concerning neurodegenerative diseases. For this reason, there is abundant work in this field, with a focus on the use of different machine learning algorithms to estimate such connectivity. Well-known studies have used simulated fMRI data using the Balloon model, along with many different Bayesian network structure learning algorithms to estimate the different connectivities. However, the nature of the algorithms used in these studies is limited, and the simulated data in the studies is always based on resting-state fMRI.

In this work, we test state-of-the-art Bayesian network structure learning algorithms that have not been used in the field of fMRI data analysis before, along with the latest version of the family of algorithms that have brought the best results in past studies: the LINGAM algorithms. We also use the Balloon model to generate new data based on the boxcar model for task-oriented fMRI studies, which to our knowledge, has not been done in the literature up to now. We then test the state-of-the-art algorithms, to both identify their capabilities in correctly discerning functional and effective connectivity under different circumstances, such as varying session lengths, and compare their results in resting-state and task-oriented simulated fMRI data.

We prove that for the same circumstances, and algorithms, results vary if the data is generated from a resting-state or task-oriented fMRI. We also demonstrate that even if some algorithms do see an increase in correctly inferred connectivity with higher session lengths, number of subjects, and scanning repetition time, this increase is not meaningful enough to warrant the use of higher session lengths, repetition times, etc. Finally, we prove that the DIRECTLINGAM is the best algorithm to use, both for its high detection of both connections and their direction and for its robustness.

Contents

List of Figures

Chapter 1

Introduction

Neuroimaging has gained traction as the main tool to study the brain and as a main tool in healthcare. The individual contributions of many scientists and researches led to the creation of neuroimaging techniques [1] that are used on a common basis in healthcare. One such example is Magnetic Resonance Imaging (MRI), and how it is used to check for possible brain injuries, birth defects, etc.

However, neuroimaging is also widely used to study the brain: its structure, connectivity, function, biology, etc. One specific area that is increasingly gaining interest is the study of functional brain networks. In this specific area of Neuroscience, researchers are interested in discovering the specific brain areas that are activated when performing certain tasks, and the flow of information in these specific areas in relation to such task. The ultimate objective of these studies is to find the relation between the brain and the corresponding behaviour exhibit in certain tasks/circumstances [2].

In neuroimaging, functional magnetic resonance imaging (fMRI) measures the rate of change of the blood-oxygen level dependant response (BOLD) of the brain, which is correlated with neural activity. The BOLD of the brain can be captured in a time series where the responses are localized in small sectors of the brain millimeters wide, known as voxels.

fMRI data can be used to perform functional connectivity analysis, which focuses on discovering the structure of brain connectivity, which consists of locating the statistical relationship between different brain areas and the orientation of such relationships. For example, it is not only of interest knowing which areas of the brain are activated when the brain has to deal with reasoning, but it is also very interesting to know which areas are have causal relationships and how the flow of information progresses from one area to another, which can be defined as the "direction" of the flow of information in the brain. As such, there are two distinct terms when dealing with connectivity studies: functional connectivity, which only studies the statistical patterns between regions of interest, and effective connectivity, which focuses on the causal relationships between such regions.

Brain connectivity is not only relevant for the study of the brain, it is also an approach for the diagnosis of many neuropathologies. Brain connectivity can be used to identify and study the development of schizophrenia [3], autism [4], Alzheimer's disease [5] and even depression [6]. Since the study of brain connectivity has so many uses,

publications in relation to both functional and effective connectivity has skyrocketed in the last decade [7]. As a result, causal network learning has emerged in computer science as a complex and important problem in the research of brain networks and neural connectivity [8].

As a result, there has also been a growing interest in using and creating machine learning and statistical algorithms that are capable of discovering brain connectivity from fMRI data. However, this is a very complex problem, for a series of reasons [9], such as that fMRI does not measure brain activity directly: it measures a change in the BOLD signal, which itself is a result of neural activity. Bayesian networks have been proven to be state of the art in this area [9]. Bayesian networks are probabilistic graphical models, which align perfectly with the nature of our problem, and they offer the possibility of easily observing and understanding the causal relationships between different variables, which again, is precisely the nature of our problem.

On the other hand, there is also another problem that is very relevant in the area of neuroscience: the infeasibility of obtaining large amounts of data. fMRI experiments are very costly, and it is generally a challenge to find enough participants for studies. Furthermore, these studies must be filtered and approved by many different ethical committees. As such, it is a limitation to rely only on observational data.

Many different studies have resorted to using data obtained from simulations, or using simulated data that has been made publicly available, such as the data in Smith's famous work with network modelling methods in fMRI [10], which used the Balloon model to create its data, and made all of the data publicly available. Many further studies have used the Balloon model and/or have used his data.

1.1 Structure of the document

Chapter 2 presents the necessary biological and theoretical background with a literature review that is required to understand the rest of the document and the reasoning behind decisions in further sections. The literature review will cover the biological principles behind fMRI, the Balloon model that will be used to model a human brain in order to simulate fMRI studies and Bayesian networks. Lastly we will have another literature review of Bayesian network structure learning algorithms which will also go over the current state of the art in such area.

Chapter 3 will define the motivation behind our work. We will discuss why we are using Bayesian networks and specific score-based structure learning algorithms with data generated from the fMRI Balloon model. This chapter will also list the objective that this manuscript intends to solve.

Chapter 4 will start by describing the nature of our problem, followed by our proposed solution and the framework used for our solution. Our experimental setup will be described in full detail, along with the methods that were followed in order to apply the different score-based structure learning algorithms to our experimental data. We will also describe how we correspondingly handled and transformed the corresponding data generated by the structure learning algorithms in order to produce our results, and the reasoning behind our election of metrics and score-based algorithms.

Chapter 5 will present the results obtained with our selected score-based algorithms. The results will show the differences between classical approaches, and score-based

algorithms that have different assumptions in regard to the Gaussianity of the data. The results will also show the influence of several parameters in our experimental data generation, such as the number of subjects, fMRI session length, etc.

Chapter 6 will provide an overview and personal interpretation of the results seen in Chapter 5. The objectives and problem statement addressed in Chapter 3 will be directly answered here. A comparison to results in other studies will also be commented, along with a final personal opinion on which algorithms are best suited for functional and effective connectivity analysis.

Chapter 7 will conclude our work along with final remarks on our results. A subsection will also be dedicated to commenting possible lines of future research.

Chapter 2

Literature review

Our literature review has three specific purposes: First, to provide a scientific and biological background in the field of fMRI and Bayesian networks, which will be our choice for the discovery of connectivity from fMRI data. Furthermore, a detailed review of modern and current state of the art algorithms for structure learning algorithms in Bayesian Networks (BNs) will be provided as well, since they will be the algorithms that help us identify the causal relations in the fMRI data, and as such, discover both the functional and effective connectivity. Additionally, we will explain why BNs were chosen as our tool for the discovery of connectivity.

2.1 fMRI

fMRI is a neuroimaging technique which measures the change of the BOLD contrast of the brain. Specifically, fMRI is a type of MRI that creates T2* weighted images, which have low spatial resolution, but can be acquired over time. This gives us the possibility of measuring over time and analyzing changes in cerebral activity over time [11].

The principle behind fMRI comes from as early as the 19th century, where Italian physiologist Angelo Mosso would have a subject lay on a balanced table and would then elicit emotional or intellectual activity from such subject. When this happened, the table would tip at the head, indicating that the blood had flown to the brain in order to meet the corresponding requirements from the neural activity in the brain [12].

On the other hand, the technology behind MRI does not emit any radioactivity and instead relies on strong magnetic fields and radio waves. It is considered a radiology technique, and can be applied in many parts of the human body, because atomic nuclei (which are present in the human body) produce spin polarization when exposed to radio frequencies emitted by MRI [13]. As such, it can clearly distinguish tissue from water and fat, and is widely used in medical diagnosis.

Due to the magnetic properties of our blood, we can see its flow in our brain thanks to fMRI technology. This is because red blood cells contain hemoglobin, which can be detected by the MRI. When neurons fire in our brain, as any other biological process, they require energy to do so, but neurons do not have reserves of sugar or oxygen

[14], so they trigger a haemodynamic response, where blood flows at an elevated rate to provide the necessary energy [15].

As mentioned previously, blood has magnetic properties: It is composed of deoxy-hemoglobin, which is paramagnetic, and oxy-haemoglobin, which in turn is diamagnetic. This means that deoxy-haemoglobin decreases fMRI signal by a large amount, but since oxy-haemoglobin is diamagnetic, it only increases the fMRI signal by a small quantity in comparison. When blood circulates to the brain areas that are under demand, newly oxygenated blood rushes in, which increases the flow of oxy-hemoglobing while the deoxy-hemoglobin is washed out.

The washing away of deoxy-hemoglobin is the main factor in the increase of the BOLD signal, and Ogawa *et al* [16] realized that this change could be picked up by MRI. As we have now seen, we are not technically measuring the BOLD signal, but rather its change. A simple comparison would be the flap of a loose trouser when riding a motorcycle: The flaps of the trouser do not determine the speed at which we are travelling, but it does indicate the change in speed of the motorcycle. The same can be said with how fMRI measures the change in the BOLD signal of the brain.

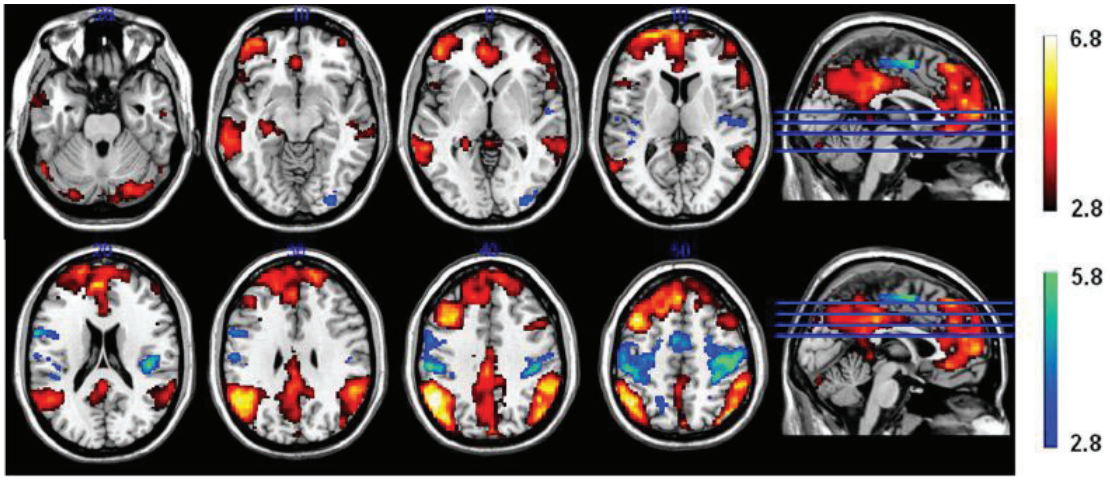

Figure 2.1: *Example of data obtained from a fMRI study.* The white-red color pattern shown in the upper legend indicates higher activation in comparison to control in the specified brain areas, while the green-blue color pattern indicates higher activation in the control group. Data from Kizilirmak *et al* [17].

However, even if the functioning of fMRI and how it is able to capture the blood flow in the brain are understood, the process that ultimately induces the haemodynamic response, and what exactly this response represents, is currently unknown and many different theories still remain unproven [18]. There is extensive research, and thus, profound understanding behind the neurophyisiology of neuronal activity and the fMRI bold response, but it is thought that the neurovascular coupling, which occurs right after the neuronal activity, manipulates an unknown metabolic signal which in turn induces the haemodynamic response. As such, it is unsafe to conclude that BOLD directly represents neural activity, but we can presume some correlation with activity and electrophysiology.

MRI data is obtained via slices of the brain, typically in axial slices. These slices can be acquired in different orders, and together they form a 3D volume of the brain. The

smallest spatial unit is known as a voxel, and it is a cube with a pre-determined size, which is usually in the range of a few millimeters. On the other hand, its temporal resolution is not as high: The temporal resolution of the fMRI is defined by its sampling rate, known as its repetition time (TR). Depending on the technology, the TR can range from a few seconds to hundreds of milliseconds. As technology progresses over time, both the temporal and the spatial resolution improve, and future insights into BOLD and cognitive neuroscience will surely improve our understanding of fMRI and as such, further improve our corresponding resolutions.

However, in order to get data such as the one shown in Figure 2.1, many pre-processing steps must be taken, along with a statistical analysis. Since most fMRI data come in hard to interpret data extensions, many programs have been developed in order to deal with pre-processing, cleaning and interpretation of neuroimaging data. One such program is the widely known FSL [19]. With this program, which can be configured in MATLAB, users can perform all the necessary steps to perform neuroimaging studies: Reconstruction of image data, motion, distortion and slice time correction, normalization, spatial smoothing and statistical analysis. We will not dig deeper into this matter, since it is a complicated task, to the point where neuroimaging is offered as a Masters in Science in many universities throughout the world.

The uses of fMRI are varied. The first and most straightforward is clinical use: One can use fMRI to detect if there is brain activation in certain areas to check for possible injuries and/or treatments for patients in comas [20]. The areas that can are activated in fMRI studies are denominated regions of interest (ROI). The other branch of uses for fMRI could be classified as purely motivated by scientific (cognitive neuroscience) purposes [21]: The localization of specific brain regions that are later associated to higher brain functions, connectivity studies and prediction studies. In our case, we are interested in the study of brain connectivity, which can be divided into two different branches: Functional connectivity, which studies statistical patterns and dependecies between different neuronal events, and effective connectivity, which studies the causal model behind the interactions and how the regions interact [22].

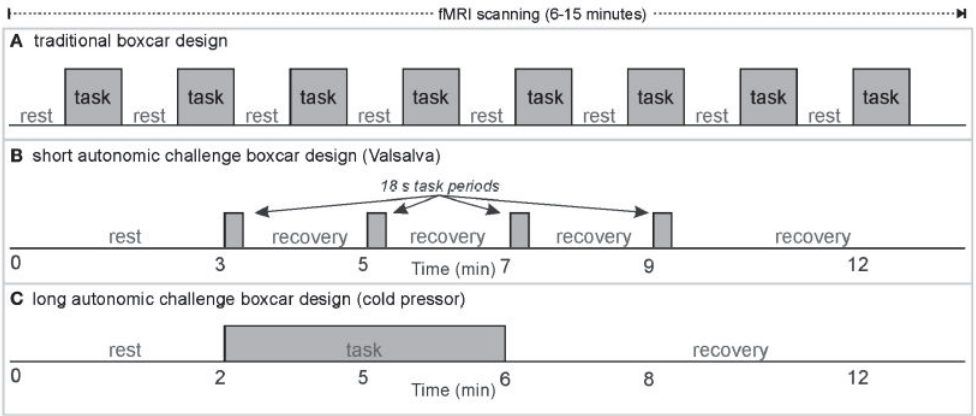

Figure 2.2: *Examples of boxcar fMRI designs.* A, B and C show different ways of setting up boxcar designs for fMRI experiments. In all of them, the grey box corresponds to the task in question, while the remaining space is rest and/or recovery. The scanning session can be longer or shorter than the time shown in the image. Source: [23]

Finally, fMRI studies can be classified into two different groups: Resting-state studies and task-oriented studies. The first consists of introducing subjects to fMRI studies as is, which means that subjects are simply instructed to remain still and motionless, while trying not to think of anything in particular, to avoid introducing noise. The latter, task-oriented studies, are done with the objective of studying specific higher cerebral functions and/or specific brain areas. They are usually done with a box-car model, in which a stimulus is presented for an interval of time, followed by a resting phase [23]. This block is repeated several times, in order to elicit the desired response in the brain (figure 2.2). When bot the "task" and "rest" states are imposed, the difference in brain activity will show which areas were activated during the task.

2.2 Balloon model

As we have seen, the process responsible for the formation of the BOLD signal is very complicated in nature and a result of the interplay between blood flow, oxygen consumption, the contraction of blood capillaries and many other physiological processes. Despite this, researches desired to have a way of simulating BOLD signal, since obtaining data from fMRI studies can be very expensive, time consuming, and require approval from various ethic committees.

Buxton was the first to develop a biomechanical model for the generation of the BOLD signal [24]. He developed a nonlinear haemodynamic model which models the interaction between dynamic changes in blood oxygenation (deoxyhemoglobin concentration) and volume, which are described as nonlinear functions of blood flow and neuronal activity. The main feature behind the Balloon Model is that blood volume can change in such model, and as a consequence, the venouos volume can change, and inflate or deflate, hence the name "balloon model".

To put it in a nutshells Buxton's Balloon Model mimicks the changes of blood flow in the brain when neural activity demands it, and the corresponding blood flow induces changes in deoxyhemoglobin content, which is modelled, along with the corresponding capillaries and venous blood, in order to imitate the BOLD signal. The model is capable of calculating the change in blood flow, and thus, BOLD signal for a given set of parameters and a neural stimulus timeseries.

2.3 Bayesian networks

A BN is a type of probabilistic graphical model which can graphically encode the joint probability distribution of a vector of random variables $X = (X_1, ..., X_n)$ [25]. Bayesian networks are capable of compacting joint probability distributions by the use of conditional independencies found in a given dataset to estimate the model. These independencies are represented via triplets of variables in the dataset, where one variable is shown to be independent of another given the other, for example, we can say that variable X is independent of Z given Y.

This information is directly encoded in the BN, which is composed of a directed acyclic graph (DAG) denoted by $G(V, E)$, where V is the set of vertices (nodes) that correspond to all the different variables in the dataset, while E is the set of directed edges that indicate conditional independencies between the nodes that they connect with their corresponding parents (the predecessors of the node). In all Bayesian networks, each

vertex (node) is conditionally independent of its non-descendants given its parent variables, which is known as the Markov property. Due to these properties, BNs are able to reduce the generally intractable joint probability distribution $P(\boldsymbol{x})$ (where $\boldsymbol{x}$ is an assignment to $\boldsymbol{X}$) into a factorized version that exploits the conditional distributions. As such, we have the following:

$$P(\boldsymbol{x}) = \prod_{i=1}^{n} P(x_i|\boldsymbol{pa}(x_i)) \tag{2.1}$$

where $\boldsymbol{pa}(x_i)$ refers to the parents of x_i in G, as previously described and n is the total amount of variables in the dataset. The process from which the joint probability distribution is reduced to the expression in equation (1) is known as joint probability distribution factorization. It makes model construction easier, since it only stores local distributions at each vertex, it has fewer parameters to assign and allows for very fast inference. In these BNs, the parameters indicate the local conditional distributions for each variable X_i.

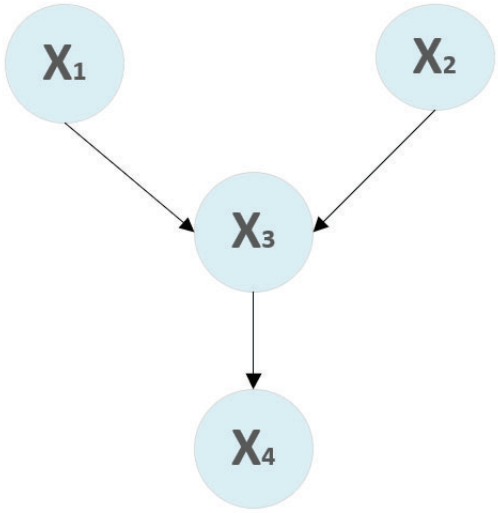

Figure 2.3: *Simple Bayesian network structure with 4 nodes.* In this network, X_4 is dependant on X_3, which is denoted as the conditional probability $P(X_4|X_3)$. Taking into account that X_3 itself is also dependant of X_1 and X_2, we can construct the factorized joint probability distribution using the marginal probabilities of X_1 and X_2 along with the conditional probabilities, such that $P(X_1, X_2, X_3, X_4) = P(X_4|X_3)P(X_3|X_1, X_2)P(X_2)P(X_1)$. Image source: `https://bit.ly/3hcMSW4`

BNs can handle both discrete and continuous data, and in some instances even both types of data simultaneously [26]. If the data is continuous, the data can either be discretized to accommodate the needs of the BN algorithms, or if the BN is able to handle continuous data, the probability mass functions are substituted by density functions.

The advantages of BNs do not only lie in its factorized joint probability distribution. They are also capable of dealing with missing or incomplete data without sacrificing their reliability [27]. They also have the very important option to manually introduce data from experts [28] and this is widely regarded as a very positive aspect. Also, due

to the Bayes theorem, BNs take into account the impact of prior data and thus, are robust to overfitting. Lastly, many BN learning algorithms give the option to introduce background knowledge, which can be expressed as the obligation to include certain statistical dependencies (edges) or veto certain edges.

Another very important characteristic of BNs is that they are *interpretable* machine learning algorithms. When one infers from BNs, or uses them to classify data, he can understand which was the process that led to the result, and how the BN has learned from the data given to it. This is not something that can be done with other machine learning approaches, such as neural networks, which function as a black box, where data is given as input, and the user does not understand the relation between such input and the decision taken in relation to that input. BNs not only are explainable, but they are also *graphical*, making them very user-friendly and offer a greater understanding of the underlying problem due to the easily understood cause-effect relationships shown in its DAG.

We can also perform inference on BNs, which is a powerful tool that gives its users the capability to perform queries on the BN in order to obtain probabilities for a variety of scenarios. With or without evidence, we can perform any kind of reasoning on BNs. For example, in Lauritzen and Spiegalhalter's famous Asia BN [29], we can calculate the probabilities of having different respiratory sicknesses by indicating the characteristics of an individual (has the user travelled to Asia? Is he a smoker? etc.), see figure 2.4.

There are two main forms of performing inference on BNs. The first is exact inference, which is a costly process that calculates the inference directly. Examples of exact inference range from brute-force computation to variable elimination algorithms and message passage algorithms, such as the clustering method devised by Lauritzen and Spiegalhalter [29].

However, exact inference was proven to be a NP-hard problem [30], and many exact inference algorithms were unfeasible in large BNs. As a consequence, approximate inferences was developed, which uses different forms of deterministic and stochastic simulations in order to find approximate answers. A pair of well-known approximate inference algorithms are logic sampling [31] and likelihood weighting [32].

The process responsible for creating the BN from the data is known as learning, and has two distinctive steps: Structure learning and parameter learning. In structure learning, the DAG is learned from the dataset $\mathcal{D}$, and in parameter learning, given the DAG G and the data $\mathcal{D}$, the parameters ϕ are learned, which are the local conditional distribution densities. Overall, the learning process can be summarized as:

$$P(G, \phi|\mathcal{D}) = P(G|\mathcal{D}) \cdot P(\phi|G, \mathcal{D}) \tag{2.2}$$

where the first term corresponds to the learning process, the second term refers to the structure learning, and the third and last term to the parameter learning.

Structure learning algorithms have differing versions on their own: They can either be constraint-based, score-based or hybrid. Constraint-based algorithms test conditional dependence and independence relations between variables to find a G that bests fit such relations. The most common constraint-based algorithm, which is widely used, is Spirts' and Glymour's PC algorithm [33].

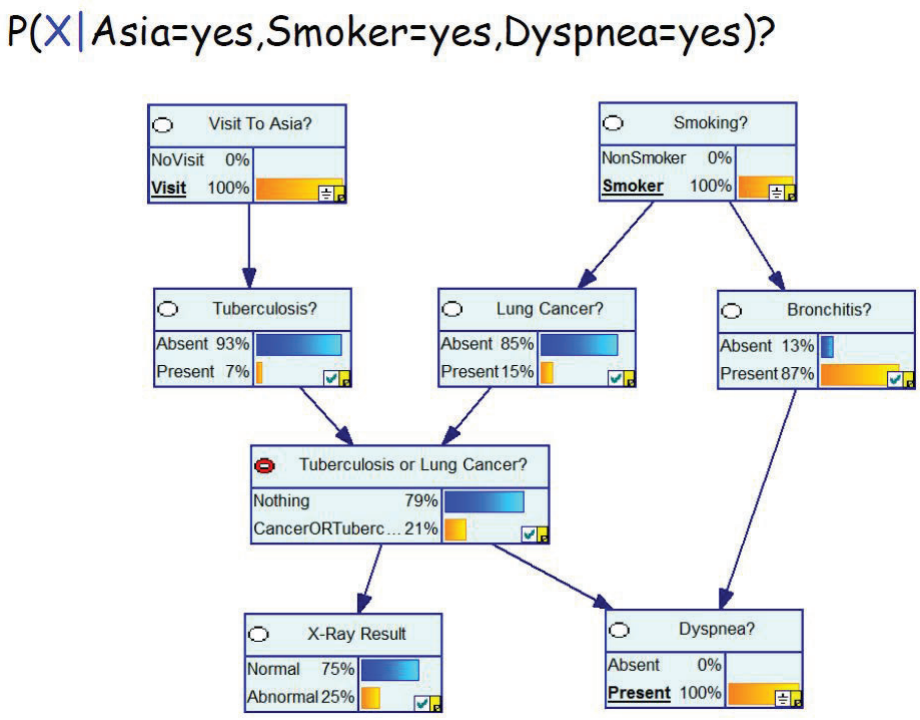

Figure 2.4: *Inference in the Asia BN.* Inference performed on Lauritzen and Spiegalhalter's Asia BN. In this inference, the evidence given is that the subject visited Asia, is a smoker and has dyspnea. The probability of having tubercolosis or lung cancer given the evidence is 21%. Image source: *Bayesian networks* course slide taught by Professors Pedro Larrañaga and Concha Bielza in UPM's Master in Artificial Intelligence

Score-based algorithms search for a G that maximizes a certain score or metric, which indicates the fit of the data into the structure G. In the process of learning, each possible DAG is manipulated in order to maximize the corresponding score metric. Each score-based algorithm differs in the score used and in the search method used to add, delete or modify edges to improve the corresponding score. An example of a score used in score-based algorithms is the Bayesian Information Criterion developed by Schwartz [34] which is defined as: $BIC = kln(N) - 2ln(L)$ where k is the number of parameters in the BN, n is the maximum number of data points and L is the value of the likelihood function. Score-based algorithms are recognized as being more robust than their constraint counterparts [35] because they have the capability to look into past actions and undo them or change them, such as removing previous added arcs or changing their direction.

Hybrid score-based methods are a combination of both constraint and score-based algorithms. They consist of two steps known as restrict and maximise [36]. In the restrict step, all nodes receive a set of candidate parents via conditional independence tests and the maximise step seeks to maximise a scored network subject to the previous condition that all the corresponding nodes must have their parents in the group previously selected in the restrict step. A well-known hybrid method is Tsamardinos' *et al's* max-min hill-climbing learning algorithm [37].

Parameter learning in Bayesian networks is done via estimation, where maximum

likelihood estimation (MLE) is one of the most common ways of doing so. However, other methods can take into account missing data, different types of categorical and continuous data, methods that use gibss sampling or Markov chain Monte Carlo, etc [38]

2.4 Bayesian structure learning

As we previously described, Bayesian structure learning can be divided into constraint-based algorithms, score-based algorithms and hybrid algorithms. In this section we are going to give a more profound overview of state-of-the art score-based, constraint and hybrid algorithms.

Score-based structure learning algorithms are more abundant than both hybrid and constraint-based methods, and more popular in the literature as a result [39]. As such, we will first discuss state of the art in hybrid methods, followed by constraint-based methods. Currently, the most used method [36] is Tsamardinos *et al's* max-min hill-climbing algorithm(MMHC) [37]. MMHC is divided into two different steps: In the first step MMHC learns the skeleton of the BN using Tsamardino *et al's* max-min parents and children [40], which is exclusively a discovery algorithm, and as such only produces the skeleton. The skeleton is then oriented using a classical greedy Bayesian score-based hill climbing search [37]. MMHC has been proven to work remarkable well, competing with many state of the art algorithms, and even outperforming the greedy equivalence search (GES) algorithm [42]. It is also featured in many well-known scientific programming languages, such as *bnlearn* [66].

Constraint-based structure learning depends purely on the corresponding hypothesis test used in the the conditional independence analysis. As previously stated, one of the most widely used approaches is the PC algorithm [33], which starts with a complete undirected graph, which is recursively modified as undirected edges are deleted via conditional independence tests. A final series of conditional independent tests direct the remaining edges. This implies a worst-case scenario of exponential cost in run time, which implies that as the number of variables increases, the PC algorithm becomes computationally intractable. Different versions of the PC algorith have been proposed to make it more robust, such as Kaslich and Bülhman's version [41], which is capable of working with up to one thousand different nodes.

There are many score-based structure learning algorithms worth mentioning, but here we will only focus in the most important ones, that have been proven to be state-of-the-art, and that excel in specific areas. For our first algorithm we will discuss the greedy equivalence search (GES), which was first introduced by Meek [42] in 1997 and later improved by Chickering *et al* [43]. Recent work by Ramsey *et al* brought the fast GES (FGES) [44] a re-organized and parallelized version of the GES algorith, that is able to operate with thousands of variables, and is much faster as well, thanks to its improved Markov blanket search.

The GES algorithm first starts with an empty graph, and edges are added in such a way that its score is maximized. This process is known as the forward stepping search. This step is done continuously until the graph ends in a state where edge additions do not maximize the score anymore. After that, a backward search is performed, where edges are instead removed to maximize the score. The GES algorithm

stops when there are no edge deletions that can increase the corresponding score. In GES and FGES the score used is the BIC.

Both GES and FGES work with partially directed acyclic graphs (PDAG), and as such, return the most probable PDAG. In the original implementations, the algorithm returns two types of connections between variables: Undirected edges, which imply a causal relation that does not have enough information to discern its orientation, and arcs, which contain direction and thus imply direct causation.

However, the GES and FGES algorithms, among most of all the structure-learning algorithms that are capable of dealing with continous data, assume that the data is Gaussian, either explicitly or implicitly. Nonetheless, there exists a family of scored-based structural learning algorithms known as "Linear Non-Gaussian Acyclic Models" (LINGAM) which assume non-Gaussianity of data. The first model of its kind, the classical LINGAM, was created by Shimizu *et al* [45], and later models are based on this first work.

The key feature in LINGAM [45] that distinguishes it from other score-based methods is that instead of using a classical testing of conditional independencies between nodes, it uses independent component analysis (ICA) in order to approximate the connections between nodes. As such, this method is often referred to as ICA-LINGAM, in order to differentiate it from other models/variants of the LINGAM family.

ICA-LINGAM heavily relies on its assumptions: All variables have an external input that has a non-gaussian distribution, the process that generates the data is linear and that there are no unobserved confounders. As a result, ICA-LINGAM does not use a score such as the BIC, but instead uses an ICA algorithm. In the original work [45], the FastICA [46] algorithm is used, and the corresponding mixing matrix that is generated is later used to estimake the corresponding network connections in the DAG.

Other variations of the ICA-LINGAM method have been developed since the first one. One of them is the DIRECTLINGAM [47] which also relies on the non-gaussianity of the data, but does not require any meta-parameters and guarantees convergence with a fixed amount of steps. In the original ICA-LINGAM method, convergence is not assured, while in DIRECTLINGAM, thanks to the use of pairwise regressions and least squares regression, convergence can be guaranteed as long as the model assumptions are met [47] and the corresponding sample size is infinite. DIRECTLINGAM was shown to produce better results with simulated data than the ICA-LINGAM method.

As previously mentioned, there are many more LINGAM methods and variations that also work with the premise of non-Gaussian data. One such example is ParceLINGAM [48], a LINGAM-based method that is robust against violations of latent confounder assumption (there are no unobserved latent confounders). By using their Parcel method, Tashiro *et al* [48] are capable of testing external influences and their independencies and as such able to detect which variables have latent confounders.

However, score-based algorithms that are based on integer linear programming work exceptionally well in general, and thus are considered state of art in their field [39]. The main aspect of integer linear programming is identifying the optimal cutting planes that correctly break cycles in such a way that the optimal solution is finally found [49]. In Cussen's implementation, the solution is measured with the Bayesian Dirichlet score [49]. Integer linear programming has been proven to work

with medium-sized BNs of up to 200 variables [39].

Moreover, learning the structure of a BN is an NP-hard problem, which implies that most structures obtained by score-based structure learning algorithms are approximations. However, interger linear programming approaches guarantee an optimal BN if the corresponding integral solution is found [50]. These kind of methods have also been proven to be state of the art and thus are faster and more exact in comparison to other dynamic programming approaches [50].

A prime example of integer linear programming in score-based learning is Cussens globally optimal BN learning using integer linear programming (GOBNILP) [52]. GOBNILP is offered as a C program and a Python package, albeit the latter has only been available for a few months. GOBNILP is built upon both of the creator's work on integer linear programming and other works such as Studeny's work on matroids in integer linear programming [51]. GOBNILP offers Bayesian structure learning with both discrete and continous data.

Chapter 3

Motivation and objectives

3.1 Motivation

Smith *et al's* work [10] has been one of the most influential in the area of learning effective connectivities in the brain from fMRI data. In their work, they deal with all the issues previously mentioned in Chapter 1, and they were the first to specify that BNs, and the corresponding structure learning score-based algorithms were the most adequate in the field of graphical and statistical methods for the study of brain connectivity.

Following their reasoning and our own, we are also going to use score-based learning algorithms for the study of effective connectivity, and we are also going to introduce an approach with a hybrid method, which we will later discuss. Constraint-based methods will not be explored, since they are not robust, as we have previously mentioned. Furthermore, they are prone to false positives, due to their process of construction with conditional independence tests. This means that in scenarios where the sample size is not big enough, constraint-based methods will perform poorly. This warrants further motivation to avoid constraint-based methods, since in neuroscience the sample size is not usually very big, due to the ethical and replicability limitations mentioned in Chapter 1 and 2.

Our motivation, and thus, the motivation behind this manuscript, is to delve deeper and contribute into the issues presented in works such as Smith *et al's* [10]. In their work, they uses the Balloon model to generate simulated fMRI data, and then applies various structure-based learning algorithms to the data, with the objective of discerning which algorithms are best suited in correctly discovering connectivity and orientation. In their study, they concluded that the existence of connections could be correctly identified with their choice of algorithms, but the proportion of correctly identified directions of such connections were very low.

However, as we have previously mentioned, this area of research is not a simple one, and even they made mistakes. Further studies, such Ramsey *et al* [53] found that Smith *et al* [10] had used a high-pass Butterworth filter using 1/8th of the full frequency range. These kinds of filters are usually used to remove low frequency drift artifacts, but in Smith's study, it was the culprit behind the poor performance in direcionality estimation. Ramsey *et al* [53] proved that the Butterworth filter removed the non-Gaussianity from the fMRI data, which in turn, reduced the effectivity of the

non-Gaussian approaches, such as the ICA-LINGAM algorithm. The cutoff frequency was also set too aggressively, which also further hindered the results.

These were not the only two studies that have been used to guide the motivation of our work. Other studies have focused on the creation of new algorithms made with the explicit purpose of fMRI group analysis, such as the iMAGES algorithm, created by Ramsey *et al* [54]. This algorithm, which is a generalisation of the GES algorithm, differs from all the other presented so far, since it is a score-based structure learning algorithm that is designed to use the concatenated data from all subject fMRI data, and uses the average BIC score for the corresponding Markov equivalence class representative. This algorithm has been proved to work succesfully in biological fMRI data as well [56]

Despite all the studies and advances performed in this area, there are still many aspects that have been overlooked or not studied enough. Even studies as recent as Sanchez-Romero's studies of statistical methods [55] follow the same outline as Smith's study [10], albeit for the use of biological fMRI data.

First of all, of all the studies mentioned up to date, and many others in the literature, use smith's simulated resting-state fMRI data from the Balloon model. However, fMRI studies are not exclusively limited to resting-state scenarios, and it has also been argued by some that resting-state data is not completely reliable, since not only are we not sure of what the subject is thinking while he is being scanned; other external factors such as caffeine and sleep quality can severely alter the results [57]. Smith's, and as a consequence, the rest of the studies have not taken into account the algorithms used in their study with task-oriented fMRI data, such as the boxcar models mentioned in chapter 2. The lack of variability in data is especially relevant in fMRI, where many studies have casted doubts upon the reliability of fMRI [58] [59]. Because of this, studies where fMRI data is involved must be treated with utmost care.

However, other studies that use biological data [56], cannot be completely reliable. In the case of simulations, we can be certain of the ground truth behind the simulated fMRI data, that is, we know the true connections and directions between the regions of interest in the brain, because they were inputted as such into the simulator. On the other hand, biological data has no actual ground truth, and thus, results from algorithms can only be compared against the own scientists' predictions, which have been proven to be invalid in many occations [59]. Therefore, simulated data is best suited for this kind of research, and will be used in our own work.

Furthermore, many of the previously mentioned studies have concluded that the LINGAM structure learning algorithms work best, and perform even better when the session length of the fMRI is increased [53] [10]. Again, we know this from data that only had resting state fMRI, leaving an open question: Does this still hold true for non-resting state simulated fMRI data?. Smith *et al* assure in their work that their results are also relevant for task fMRI data, but do not provide any empirical evidence in doing so. Besides, new LINGAM-based algorithms have been developed since then, and have not been tested on fMRI data yet, such as the DIRECTLINGAM method, which we also wish to test, to determine if there is an improvement in performance.

Still, Smith *et al's* work has been highly insightful, and has given the scientific community very important information as well, such as methods that do not work well

with fMRI data and as such do not warrant further investigation, such as those based purely on Granger causality. Additionally, the only method focused on group-level fMRI, iMAGES, will be included as a benchmark, but since the best-performing algorithms have been single-subject oriented [10] [55], we will also focus on single-subject methods as well.

Despite this, all of the previously mentioned studies have not many other score-based or hybrid algorithms that are widely available and have been proven to yield good results in structure learning, such as the Max-Min Hill Climbing [37], integer lineal programming methods [50], or other classical score-based algorithms such as the tabu search [60]. As a result, these algorithms will be in the list of methods that we intend to study.

Finally, many of these studies, particularly Smith *et al's* original work, use an elevated number of nodes in some of their experiments, with some of them reaching up to 50 nodes (ROIs). If we truly are interested in the applications of algorithms exclusively for fMRI data, it is unreasonable to use such an elevated number of nodes. Most fMRI studies do not surpass more than 10 ROIs, because these regions encompass sizeable chunks of the brain. Furthermore, the brain is usually divided into areas known as Broadmann areas, which divide the cortex into specialized areas for cognitive and behavioural functions [61], and there are only 52 of them. Activation of 50 ROIs would practically mean that the whole brain has been activated.

For these reasons, we will not explore scenarios with many nodes, and limit ourselves to plausible brain activation layouts, that do not surpass more than 10 nodes. We will use Smith's ring-like 5 node structures [10], and we will also create a custom brain network with 8 nodes, that will be based on biological data, with the objective of being as faithful as possible to an empirical scenario.

3.2 Objectives

Taking into account the motivation in section 3.1, the main objectives of this work are as follows:

- Using the Balloon model to create simulated task fMRI data, using the boxcar model described in section 2.1. At the same time, the structure from which the simulated fMRI data will be generated will be based on an empircal structure, that is, following the structure seen in a biological study, where different regions of interest are selected from a task related fMRI.
- Testing BN structure learning algorithms that have not been used before, to the best of our knowledge, in the area of fMRI data. These algorithms will include both score-based methods and hybrid methods.
- Investigating the effect of changing the length of the fMRI scanning session, the number of subjects and the repetition time. This will be done to ascertain whether the changes seen in resting-state simulated fMRI data are the same as in task based simulated fMRI data, such as that LINGAM algorithms benefit from longer scanning sessions. We are also interested in seeing how the effect of changing the different parameters will affect both the algorithms used for the first time in this kind of data.

- Introduce a new simulation in which all of the corresponding parameters, apart from the number of nodes and its structure, will use the current state of the art in fMRI, in order to illustrate the advances in fMRI technology since Smith's study. Since previous studies have also mentioned how the ideal scenario will have as much data as possible, and lengthy scanning sessions, we will have the corresponding parameters emulate such excellent conditions, which are feasible nowadays.

Chapter 4

Methodology and Study

4.1 Problem statement

Our problem can be summarized as the need to test BN score-based and hybrid structure learning algorithms that have not been used in the field of fMRI on both simulated resting-state and task-induced fMRI data. In addition, these algorithms must be tested so that we may see the impact of different scenarios in fMRI studies in their performance, such as the session length, number of subjects and the length of the repetition time of the MRI scanner. However, since it is rare to be able to have such flexibility in fMRI studies, we will have two different scenarios: First, a typical fMRI study with average funding, and second, one with excellent funding, which implies high availability of subjects, state-of-the-art MRI scanners which translates to better MRI scanners with lower repetition time and lengthy scanning sessions, in order to test the most probable practical scenarios.

4.2 Proposed solution and framework

Following the problem statement and our objectives, we propose a two step process: First, we will use the Balloon model, as described in Smith *et al's* work [10], with the pertaining modifications in the corresponding code, and with different network topologies, in order to produce simulated resting-state and task based fMRI data. The second step will consist of feeding the simulated data into the corresponding structure learning algorithms and creating a series of scripts that will allow the analysis of the output from the structure learning algorithms, in order to calculte the number of correctely inferred connections, orientations of such directions, etc.

4.2.1 Simulation data

For our simulated fMRI data we used the same Balloon model specified in Smith *et al's* original work [10]. The C++ code (generated by Mark Woolrich) with the Balloon model was provided upon request by Stephen Smith, along with the corresponding MATLAB script that ran the C++ code with the corresponding parameters.

As in Smith *et al's* work, we had the neural time series based on a Poisson Process with an up and down state, which represented firing and resting states correspondingly. Mean duration of up and down states were left at the original values: 2.5

seconds for up and 10 seconds for down. For all our simulations, all nodes had these Poisson processes as an external input, however, we will later discuss how we included the box-car model into our task-based simulations.

The signals then propagate though the corresponding simulated network via the dynamic causal modeling (DCM) neural network model [62], which models changes of neuronal states of vectors in time with the objective of inferring causal architectures, via:

$$\hat{z} = \sigma \boldsymbol{A} z + \boldsymbol{C} u \tag{4.1}$$

where z is the neural time series, $\hat{z}$ its rate of change and u the external inputs. σ is the neural lag between the nodes, which was set to 50ms instead of the classical 1000 ms, which Smith *et al* also did, in order to emulate the upper limit of neural lags observed in practice [10]. A is the connectivity matrix, which represents the connection strengths between the nodes of the matrix, and the diagonal is set to -1, to indicate that there are no self-connections. Finally, C is the external outputs matrix, which controls the connection strength with which the external inputs are fed into the nodes. The columns in C control the number of external inputs, and the rows represent the corresponding nodes of our simulation topology.

All of the corresponding simulations were realized in a MATLAB script, and then the corresponding neural time series of each node were passed to the nonlinear Balloon model. In order to simulate the haemodynamic response variability, a randomness of 0.5 seconds of standard deviation was put in all the node's model parameters. Lastly noise was added to the final BOLD output, with a standard deviation ranging from 0.1 to 1%, in order to replicate thermal white noise. Further details are given next.

Four different network topologies were created (See figure 4.1), and as it has been specified, with a low count of nodes, in order to represent plausible fMRI data, with the corresponding data signifying different ROIs. The first two network topologies are based on Smith *et al's* main network "S5" [10], which is composed of 5 nodes in a ring-like manner. The last two network topologies are based on Pérez *et al's* study [56]. In their work, they use the GES and iMAGES algorithms on data obtained from a task-based fMRI experiment, where gambling tasks were offered to the participants in a boxcar style. The corresponding data consisted of a time series with eight different regions of interest. We gathered the results from the GES and iMAGES algorithm presented in their work, and constructed a network based on such results, adding the corresponding directions to the edges with no directions.

The two first network topologies have the same number of nodes and connections, but the orientation of the connections is different in the second topology. This is because the first topology is meant for resting-state fMRI and the second topology for a task-based fMRI. In order to avoid possible confounds, we left the first network with the same connectivity matrix (as in Smith *et al's* work), and we changed them for the second network, where we introduced the external stimuli which represents the task in question.

The third and fourth networks are completely alike, except for the task stimulus, which is expressed as an additional connection of the A matrix. In total, there are four different network topologies: two that correspond to resting-state fMRI (first and

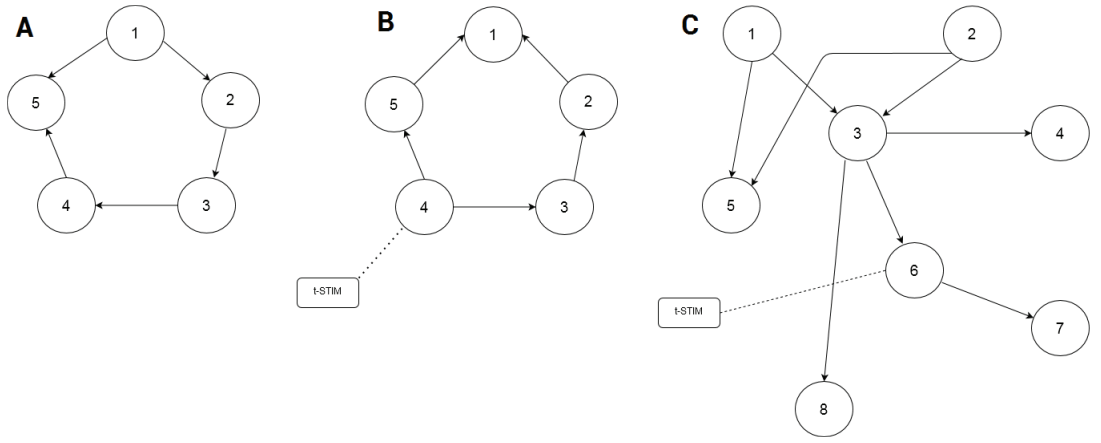

Figure 4.1: *Network topologies used in our simulations*. A) Network topology of the first simulation; a replica of Smith's S5 network from [10]. B) Second network topology, with an external task-based stimulus named t-STIM, following a boxcar based approach and feeds into node 4 with a dotted line. C) Third and fourth network topology. The third network corresponds to all the connections except the t-STIM dotted line, the task-based stimulus. The fourth connection includes the t-STIM stimulus, which again follows a boxcar approach, and feeds into node 6.

third) and two that correspond to task-based fMRI (second and fourth). For our task stimulus, we used a classical boxcar model approach, as specified in section 2.1: We simulated a task that lasted 40 seconds, and a resting state of 20 seconds between tasks. We created a time series with ones and zeroes, where the ones corresponded to the task activity, and the zeroes to rest. The C matrix was then modified: Another column was added, to represent the boxcar stimuli, and a 1 was added in the corresponding row where the stimulus is fed to: to nodes 4 and 6 respectively.

The connection strengths of the A matrix were set to random values with a mean of 0.5 and a standard deviation of 0.1, with the values truncated between a minimum of 0.3 and a maximum of 0.7 in accordance to Sanchez-Romero *et al* [55]. A coding error was detected in the connection strength calculation in Smith *et al's* work [10]: When they set their minimum and maximum values with a range of 0.2 and 0.6 for their connectivity values in the A metric, they summed all the values from -infinity to 0.2 and from 0.6 to infinity, and placed them on 0.2 and 0.6 (see figure 4.2). In order to solve this, we used the MATLAB functions *makedist*, *truncate* and *random* to create a distribution, truncate and sample from it.

Finally, all data was passed through a high-pass filter with a cutoff of 200 seconds ($1/200Hz$). In order to avoid commiting a mistake such as using a filter that removes the non-Gaussianity from the data, like Smith *et al's* use of the Butterworth filter, we used an implementation of the FSL filter on MATLAB, which was provided to us by Sanchez-Romero, and was developed by Jeanette Mumford. This filter was used because Ramsey *et al* proved that it does not impact the non-Gaussianity of the data [53]

All experiments were run on a Ubuntu 20.4 machine, with 16 GB of RAM and a i7-5820K with no additional overclocking. Since the original MATLAB code provided by Smith saved the stimulus files, they had to be deleted after simulation in order to be

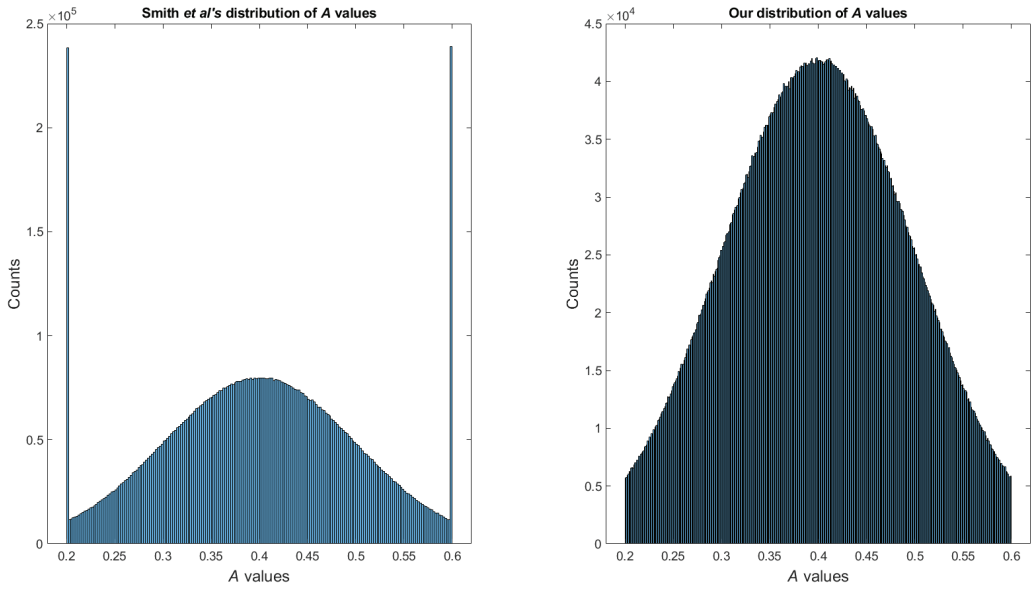

Figure 4.2: *10 million sample comparison between Smith et al's and this work's implementation of the value distribution of the A matrix.* For both methods we drew a total of 10 million random numbers. As previously explained, the distribution of values in Smith *et al's* version is unnatural, since 0.2 and 0.6 contain the sum of all values from those points to the ends of the tails. Please note that we used a minimum and maximum range of 0.2 and 0.6 in order to maintain the same values as in Smith *et al's* study, and that in this work the corresponding values will be 0.3 and 0.7 instead.

able to store all results on the machine, since we did not use a cluster. Final BOLD time series were saved in csv files with no headers.

4.2.2 Tested scenarios

For each of the four different network topologies, we performed a total of eight different experiments, where we varied the repetition time, session length or number of subjects. Our main experiment, which we titled "normal conditions," had a session length of 10 minutes, 60 subjects and a repetition time of 1.2 seconds, to represent advances in fMRI technology since Smith *et al's* study [55]. For the following experiments the parameters were fixed to these values and only one parameter would be changed, in order to study the impact of that parameter exclusively:

- Session length studies: Three scenarios, where the total scanning session had a length of 5, 30 and 60 minutes. The 5 minutes session length was included to validate that results could not only improve (e.g; with 30 and 60 minutes session lengths), but that they could also be liable to worsen if the session length was cut.
- Studies with number of subjects: Also three different scenarios, with 10, 30 and 100 subjects respectively. These tests were done to check if the algorithms benefit from more data, or if they are capable of inferring the structure behind brain activations with little data.

- Studies with different repetition times (TR): Three different scenarios were introduced here as well, which each had a repetition time of 250, 750 and 3000 ms. Again this was made to check if the quality of the MRI scanner has a considerable impact in the performance of the corresponding algorithms.

Finally, another experiment was carried out independently, which we denominated as "excellent conditions". In this experiment, a repetition time of 720 ms, session length of 75 minutes and a subject count of 100 were used, in order to represent a scenario where funding is widely available. Both the number of subjects and the session length are directly correlated with funding: More funds means more spending in publicity, incentives for subjects, and operating costs of MRI machines. On the other hand, even if there are fMRI studies that have TRs from 200-400 ms [63],we decided to use the TR seen in the Human Connectome Project [64]. The overall objective of all these experiments could be summed up as a way to test if it's worth the effort of investing additional funds and/or recruiting more volunteers for fMRI studies which wish to discover the underlying network topology.

Each one of these scenarios was repeated ten times, in order to account for statistical significance. The total runtime used to generate all the corresponding data was approximately 5 days, using the same machine indicated earlier. We would recommend using a cluster if the number of experiments is much greater, and it would be practically unfeasible to do 100 repetitions per scenario without a cluster as well.

4.3 Tested algorithms

Once all the data was generated, we fed it to the corresponding structure learning algorithms. In this section we are going to list which algorithms we used, parameters, and the process that was followed in order to generate the corresponding output data.

- FGES: This algorithm was used as a baseline, in order to compare the results with the other algorithms. We decided that it was a suitable baseline, since it is one of the most known score-based BN structure learning algorithms and it has been used in studies such as Smith *et al's* [10], and we were also interested in whether FGES also brings a performance improvement. We used the command line version of TETRAD [65] in a Python script that we generated which calls the TETRAD program for all the data, subject by subject. We used a penalty discount of 1 (complexion penalty parameter used in the score) and the BIC score. The algorithm returns a .txt file with connections that can be oriented , or have no direction. Total runtime was approximately 17 hours.

- iMAGES: We followed the same rationale explained in FGES. This algorithm is intended as a baseline, and is the only algorithm that processed data in a group level. For this, we centered the subjects' BOLD time series before concatenating them, in order to avoid spurious associations. For centering we used the *stats* package in Python, and transformed the data to zscores with the function *zscore*. The TETRAD command line version was used with a penalty discount of 1, structure prior coefficient of 1 and one edge faithfulness. The algorithm returns a .txt file with connections that can be oriented, or have no direction. Total runtime was approximately 1.5 hours.

- Max-Min Hill Climbing: This is the only hybrid structure learning algorithm

that we will use. Tsamardino *et al's* version was used [37], and we used bnlearn [66], a R package, to execute the algorithm. However, since we already had a framework of Python scripts that already saved and analyzed the corresponding output of the algorithms, so a R function was created, and was later called from a Python script for all subject data. The classical BIC score was used, and the output was a dataframe with the corresponding edges and their orientations between nodes. Because we directly saved the output into Python, we then directly created the corresponding metrics (see section 4.4) without the need to save any additional files. Total runtime was approximately 5 hours.

- Tabu search: A score-based structure learning algorithm developed by Russell *et al* [60]. We used bnlearn's [66] implementation and followed the same steps as in the MMHC algorithm: created a function in R and called it from a Python script, using the BIC score. The metrics were created directly, without the need to save any additional files as well. Total runtime was approximately 5 hours.
- ICA-LINGAM: A score-based structre learning algorithm that works under the assumption that the data is non-gaussian. This was the original LINGAM method, which worked with ICA, developed by Shimizu *et al* [45]. We used the implementation available in the TETRAD command line version with the default parameters. A Python script was used to call the corresponding Java program, which in turn saved the output as a .txt, where the connections between the different nodes are written. Total runtime was approximately 19 hours.
- DIRECTLINGAM: A more recent version of the LINGAM family [47]. A Python package under the name of *lingam* was used, which holds all of the different LINGAM algorithms, including DIRECTLINGAM. The package does not output an external file and instead returns an adjacency matrix within the program. Total runtime was approximately 7 hours.
- Integer Lineal Programming: We used Cussen's and Bartlett's GOBNILP, which is built upon Integer Linear Programming and matroids [51]. We used the Python implementation [52] with the BGe score. However, we ran into two issues with the program. The first one is that the output was a visual graph, so we had to output the plain file version of the graph, which indicates how the corresponding visual library draws the graph, and extract the edge and orientation information from there. Secondly, the program would unexpectedly crash after approximately 240 runs (each run was a subject), but would function normally if the Python environment was closed and opened again, which mean that a possible memory bug is present in the original C version. In order to combat this, a Python script that manually exited after 2 repetitions of the same experiment was created. The user would have to wait a few minutes until the analysis was done, and hit enter again. The script took care of not repeating the analysis on the same data. This took approximately 5 hours.

4.4 Data treatment for measurements

As we have previously mentioned, depending on the type of algorithm, the output of the programs and scripts either left .txt files or directly created the corresponding metrics (true positives, negatives, etc.), which will be discussed in more detail in chapter 5. In order to create the metrics, we compared the corresponding predicted

edges to our ground truth, and counted how many were correctly or incorrectly determined. We did this for both the presence of a connection (functional connectivity), and the correct orientation of the connection(effective connectivity).

For the FGES, iMAGES, and ICA-LINGAM algorithms, separate Python scripts were created, that read all the corresponding .txt files, subject by subject (or group by group in the case of iMAGES), and then generated the corresponding metrics, which were saved as a *pickle* file. The results were then fed through a last step, another Python script which loaded the corresponding pickle metric data, and calculated the mean results over all 10 runs of each experiment, and saved them as .txt files, for ease of plotting later on. This process took approximately 4 hours for all the data.

Chapter 5

Results

For our results, we decided on measuring the "connectivity" and "orientation" of the predicted BNs. A Python script was developed to process the data that was generated by the algorithms that produce output files (FGES, iMAGES and ICA-LINGAM). The rest of the data was also analyzed in the same way, but no additional Python scripts were required.

To generate the results, the presence of a connection was first determined. Once the connection was detected, the ground truth would also be compared, and if there was a connection between the two predicted nodes, independent of its orientation, it would be counted as a hit. After that, the orientation would be checked, and would only register a hit if the orientation was the same as in the ground truth. This method works for both the algorithms that only produce oriented edges, and edges with no direction, such as the GES and iMAGES algorithm.

Consequently, the reported measurements will be the sensitivity and specificity of both the correct identification of connections and orientation. We are therefore essentially measuring the functional and effective connectivity, which again, correspond to the identification of the connection and the orientation of such connections, respectively.

In total, we have 4 measures, and we decided to use the sensitivity and specificty against other classical measurements, such as the accuracy, due to the nature of our problem. We are not only interested in correctly inferring the presence of connections and their orientations, rather we are also interested in knowing if the algorithms overestimate, underestimate, and to know whether the selected positives and negatives are truly relevant. Even if an algorithm performs poorly in detecting the presence of connections, it could still perform well in the detection of negatives, and still be of some value. Other famous studies [10] [55] of probabilistic graphical models in fMRI have also used metrics such as the ones used in this work instead of "only" using the accuracy.

We will now present our results, which will be divided into two different sections. In section 5.1 we will compare the results of the seven different algorithms in our two main scenarios that were described in section 4.2.2: The "normal" and "excellent" conditions. In section 5.2 we will perform a study on the rest of the scenarios, to asses the impact of scanning length, number of subjects and TR in the results. All results will be the mean of the 10 results available per scenario, as indicated above.

5.1 Performance of algorithms in normal and excellent conditions

In this section we will present the results from the seven different algorithms in our two main scenarios. These scenarios will be indicated as 5-node or 8-node in the plots, which signify our network topologies that have 5 and 8 nodes respectively (see section 4.2.1), followed by the words "Boxcar" or "Resting State", which identify if the network topology has an external stimulus or not. The exact disposition of our network structures can be seen in figure 4.1, where 4.1A and 4.1B refer to the *5 node Resting State* and the *5 node Boxcar* plot titles, while the other two: *8-node Resting State* and *8-node Boxcar* correspond to figure 4.1C.

Our previously mentioned two main scenarios refer to the "normal" and "excellent" conditions. The normal conditions refer to a scenario with TR of 1.2 seconds, 60 subjects and session length of 10 minutes, while the excellent conditions have a TR of 720ms, 100 subjects and a session length of 75 minutes.

We will list the results for the normal conditions and excellent conditions separately, and will proved additional commentary on the results as well.

5.1.1 Normal conditions

As stated previously, in this section we are going to display the results obtained for all algorithms for our normal conditions, with all four different topologies. We will first list the first and third topology, which correspond to resting-state (Figure 4.1.A and 4.1.C), and will then list the remaining 2 (Figure 4.1.B and 4.1.C), which correspond to boxcar simulations.

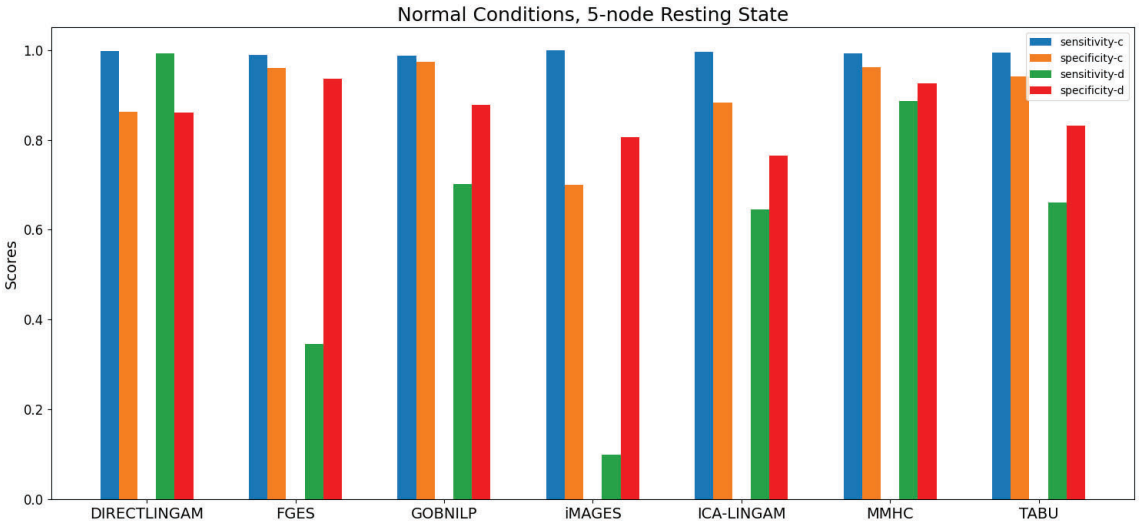

Figure 5.1: *Algorithm performance on the 5-node resting-state topology in normal conditions*. The bar chart shows the sensitivity and specificity scores for both connections and orientations for each distinct algorithm. The connection metrics are specified by ending in "-c", and the orientation values end with a "-d".

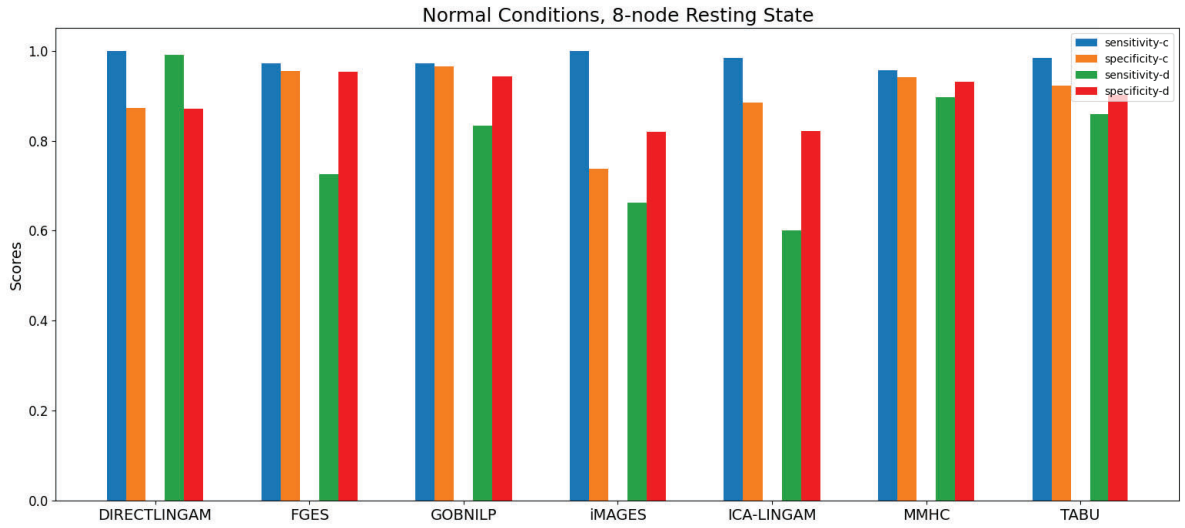

Figure 5.2: *Algorithm performance on the 8-node resting-state topology in normal conditions.* The displayed values follow the same visual guide as described in figure 5.1

The results for this pair of scenarios show many issues. The first one being how the DIRECTLINGAM algorithm obtains very similar and accurate results for both connection and orientation, as well as the MMHC algorithm. Excluding the LINGAM algorithms, an overall improvement can be seen for the rest of the algorithms in the sensitivity of the direction when the structure has 8 nodes, which could mean that these methods do not estimate direction correctly in situations with few nodes. Furthermore, connectivity results are adequate, with all algorithms having a sensitivity of connections higher than 0.9. However, the same can not be said for the iMAGES algorithm, which is the only one to perform poorly in the specificity of the connections in comparison to the rest (figure 5.1). It also obtains discouraging results for direction sensitivity.

On the other hand, the direction results are poor in general, when compared to the connectivity values. DIRECTLINGAM and MMHC obtain good results, with sensitivity and specificity higher than 8, with the sensitivity of DIRECLINGAM approaching a perfect 1, but the rest of the methods do not perform as well. The GOBNILP approach does obtain better results with 8 nodes (figure 5.2), but the previous two do not need 8 nodes to obtain such results. It is also quite surprising to see the FGES and iMAGES methods obtaining such poor quality results in direction, specially with 5 nodes, with iMAGES's direction sensitivity being lower than 0.1.

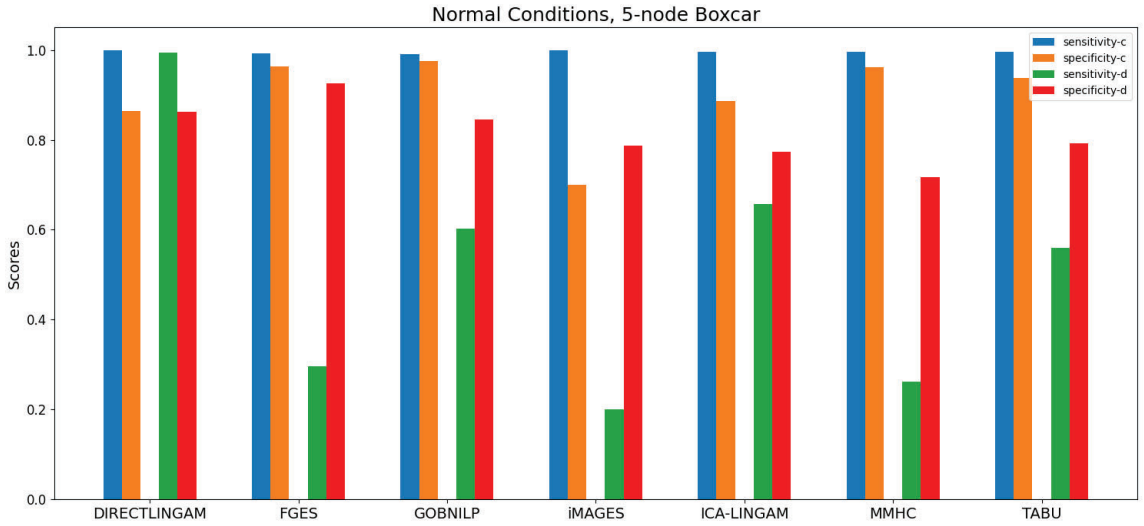

Figure 5.3: *Algorithm performance on the 5-node resting-state boxcar in normal conditions*. The displayed values follow the same visual guide as described in figure 5.1

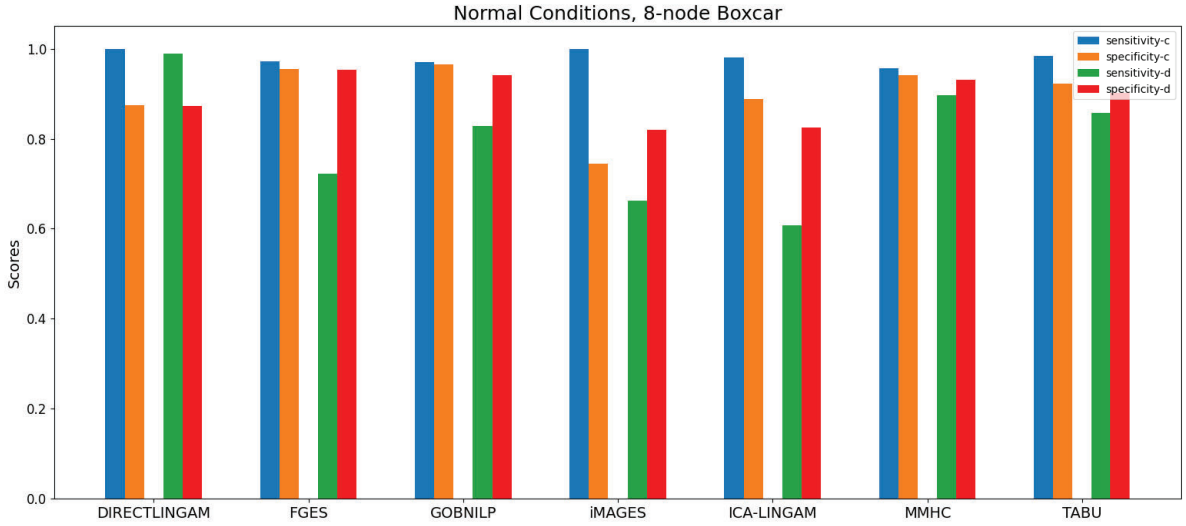

Figure 5.4: *Algorithm performance on the 8-node boxcar topology in normal conditions*. The displayed values follow the same visual guide as in figure 5.1

The results from the boxcar simulations show the same pattern as in the resting-state: results are better when the topology has more nodes, and even show the same improvements when changed from 5 to 8 nodes. However, there are some changes: DIRECTLINGAM is now the only method that works equally well under both topologies. The MMHC algorithm performs very poorly on its direction estimation when there are only 5 nodes (Figure 5.3), but performs well with 8 (Figure 5.4), something that did not happen with the resting-state topologies(MMHC worked well in both scenarios).

Finally, when comparing the resting state and boxcar simulations, the stability of the DIRECTLINGAM method can be appreciated much better. The results are consistent, with very slight variations. Nonetheless, the same cannot be said for the rest of

the algorithms. When comparing the 8-node topologies, they perform similarly on both resting-state and boxcar simulations, with a very slight improvement for some algorithms on the boxcar method, but there is a much bigger difference when the 5-node topologies are compared. The direction results improve significantly for the FGES, GOBNILP, iMAGES and tabu approach, while the connection results remain very similar.

5.1.2 Excellent conditions

For this section we will perform the same tasks as in the previous section, but in this case, we will show the results for the excellent conditions, with the four corresponding topologies. Again, the first two topologies, that pertain to the resting-state topologies will be presented (figure 5.5 and 5.6), followed by the other two that belong to the boxcar topologies (figure 5.7 and 5.8).

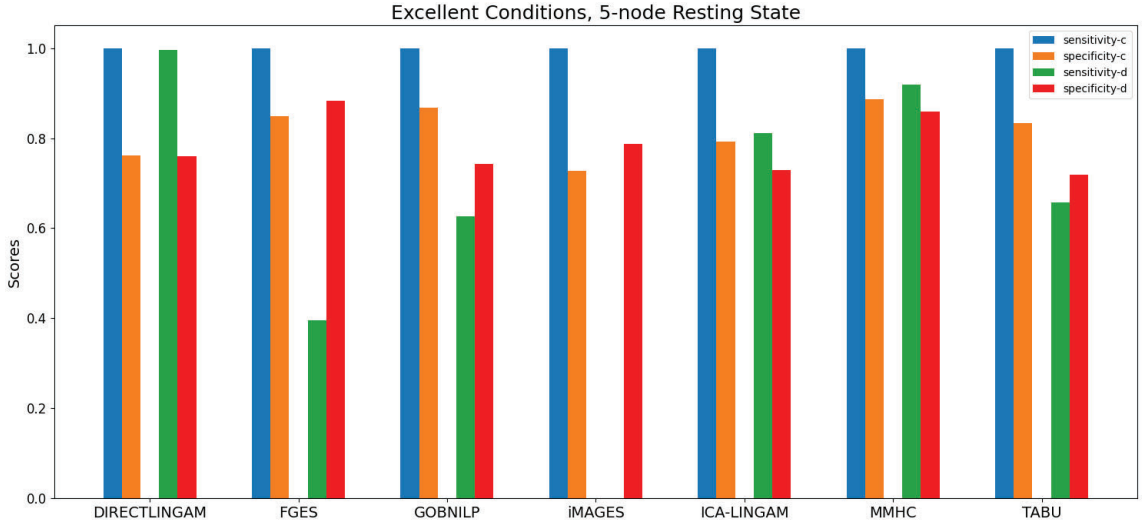

Figure 5.5: *Algorithm performance on the 5-node resting-state topology in excellent conditions*. The displayed values follow the same visual guide as in figure 5.1

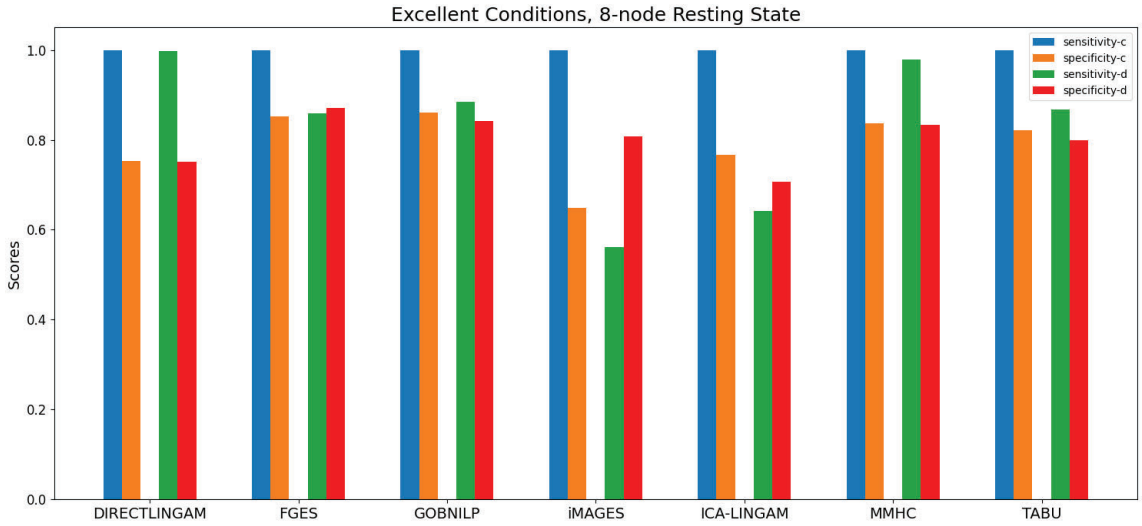

Figure 5.6: *Algorithm performance on the 8-node resting-state topology in excellent conditions*. The displayed values follow the same visual guide as described in figure 5.1

The results for the excellent conditions with resting-state follow practically the same behaviour as in the normal conditions, with the distinction of both of the LINGAM methods, which worsen with the 8-node topology (DIRECTLINGAM worsens, but by a very small factor).The rest of the algorithms significantly improve their sensitivity and specificity for the directions, but with another side-effect: the specificity of the connections drop slightly. The sensitivity of all the algorithms sit at 1, or at near 1, and do not change between the 5 and 8 node topologies, but on the other hand, the specificity is low when compared to previous scenarios, indicating a trade-off between sensitivity and specificity.

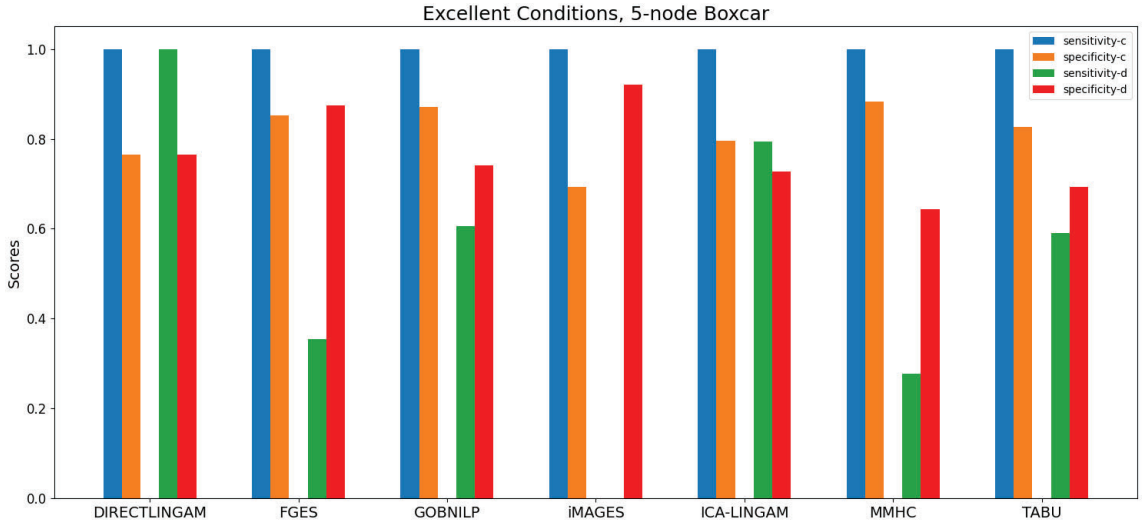

Figure 5.7: *Algorithm performance on the 5-node boxcar topology in excellent conditions*. The displayed values follow the same visual guide as described in figure 5.1

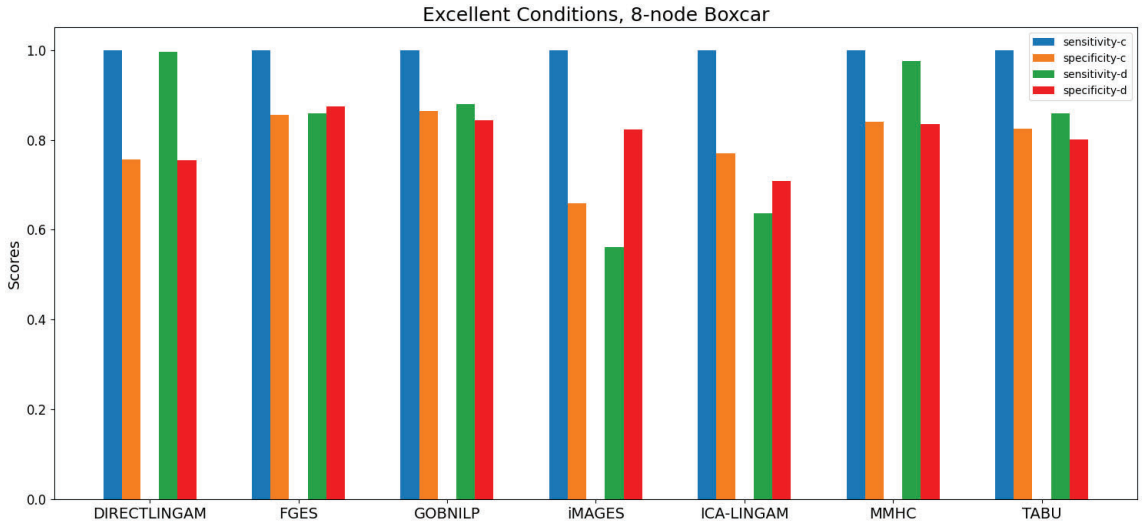

Figure 5.8: *Algorithm performance on the 8-node boxcar topology in excellent conditions*. The displayed values follow the same visual guide as described in figure 5.1

Lastly, we also see a similar pattern for our last two experiments when compared to the normal conditions: the sensitivity and specificity of the direction is better with the 8-node topology, except for the DIRECTLINGAM algorithm. However there is a new component: The 8-node topology has a slightly worse result for the specificity of its connections, even for the DIRECTLINGAM algorithm. The iMAGES and GES algorithm also perform very poorly, with iMAGES boasting a 0 on its direction sensitivity for five nodes. It is also worth noting how the MMHC has notable results for the 8-node network, but just like in the normal conditions, performs very poorly for the 5-node network.

5.2 Full scenario study

As stated previously, the objective of this section is to perform an analysis on all of the scenarios listed in section 4.2.2. For that we will visualize the sensitivity and specificity for both the connection and the direction for all possible combinations of algorithms, topologies and scenarios. With this analysis we will be able to observe the effect that changes in session length, number of subjects and repetition time have on the performance of the algorithms, and the suitability of these different algorithms under all different circumstances.

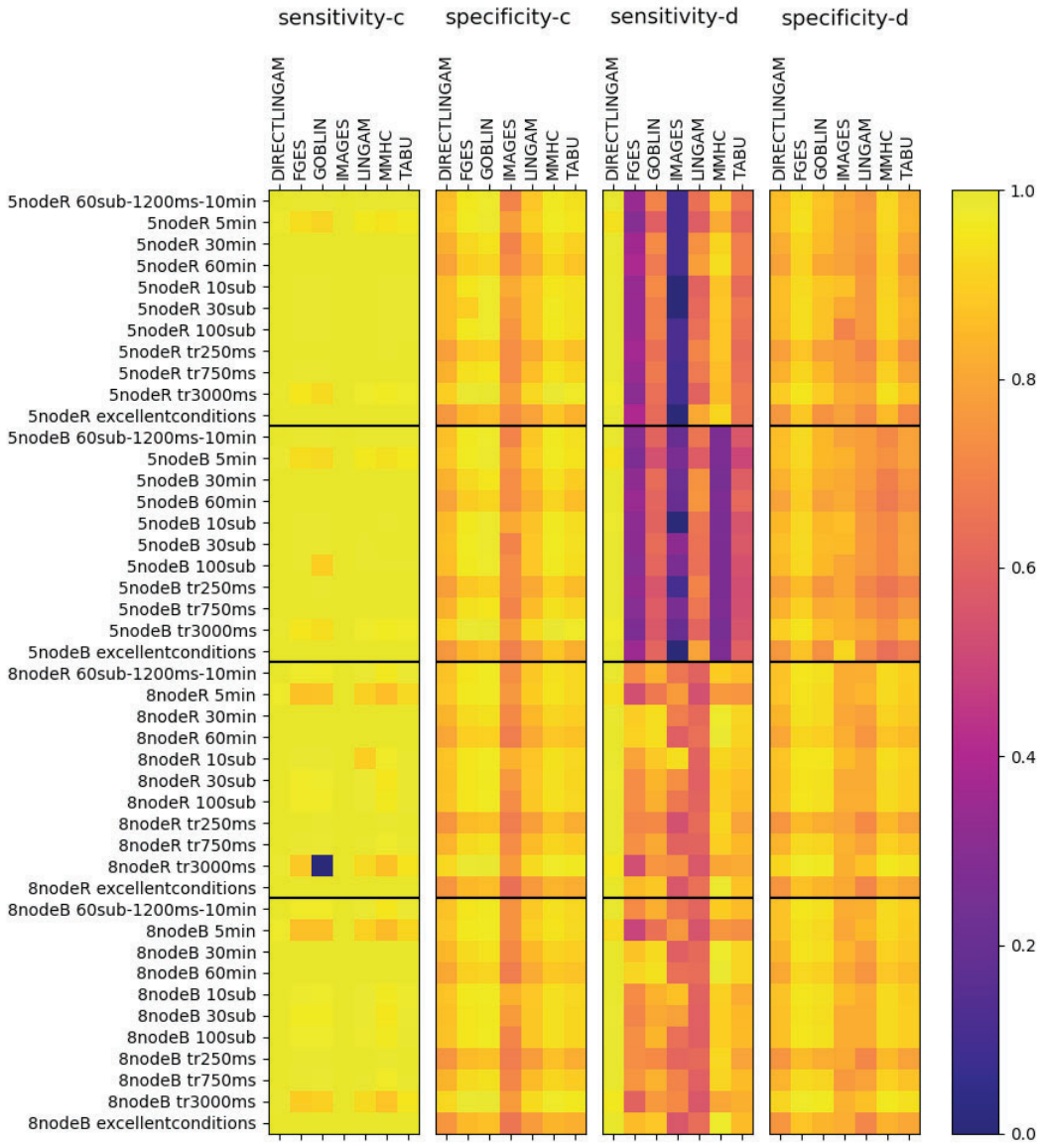

Figure 5.9: *Connectivity and orientation result heatmaps for all possible topologies and scenarios.* Each table displays the results for the sensitivity and specificity of the connection and direction (titles of each table). The x labels denote the combination of the topology and the scenario. The first word is composed of the number of nodes and a letter. 'R' means resting state, and 'B' means boxcar. The second word, which is often hyphenated, refers to the specific scenario. The 5, 30 and 60min refer to the session length, 10,30 and 100sub to the number of subjects and tr250ms, tr750ms and tr3000ms refe rto the different TRs. '60sub-1200ms-10min' is the "normal" conditions, while the excellent conditions is labelled as such. Finally, the black lines separate the four different network topologies: The first 11 rows are based on the network topology with 5 nodes in resting state, and the next 11, separated by black lines, are the networks based on the 5-node boxcar network topology, and so on.

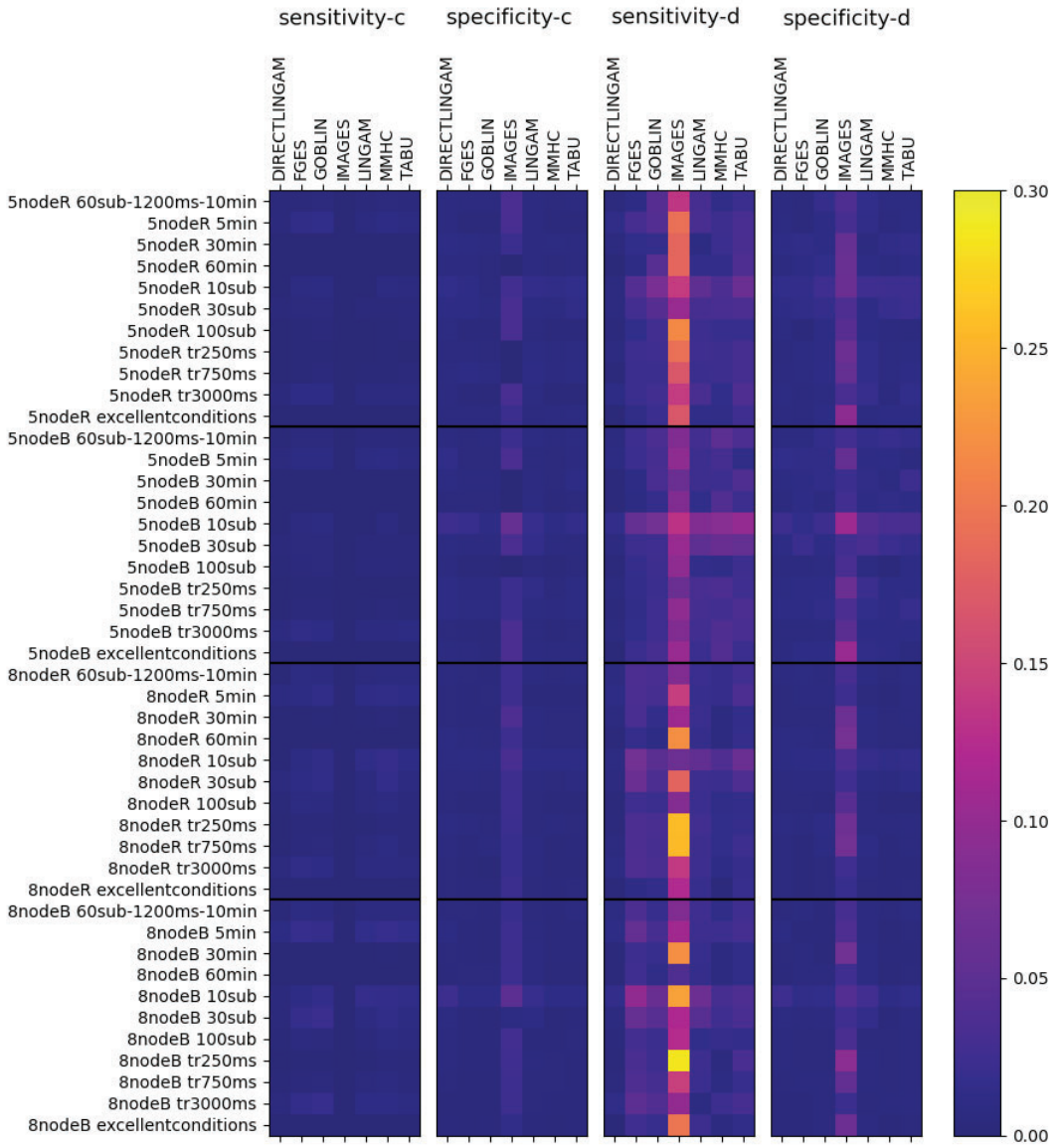

Figure 5.10: *Connectivity and Orientation standard deviations heatmaps for all possible topologies and scenarios*. Each table displays the standard deviations for the sensitivity and specificity of the connection and direction. The data display and labels follow the same indications as in Figure 5.9

The general overview of the results for all possible combinations of algorithms, network topologies and scenarios facilitates a quick comprehension of all algorithms and their performance under different situations. For example, thanks to Figure 5.9, it can be easily seen how the DIRECTLINGAM algorithm is very robust and reliable, since it always provides very similar results for all possible scenarios, and how it also gives very high scores for the sensitivity of both the connections and the directions.

Another interesting property that we can easily observed is how most algorithms perform poorly in the sensitivity of their direction for both of the topologies that have 5 nodes, except for the DIRECTLINGAM algorithm, which performs equally well independent of its topology and scenario. Other algorithms, such as the GOBNILP approach, can be seen performing well where most of the other algorithms fail, but is still inferior to the DIRECTLINGAM approach. The MMHC algorithm also performs reasonably well on most circumstances, and has even better direction specificity than the DIRECTLINGAM approach in some cases, but the poor direction sensitivity results on the boxcar 5-node network does not make it a candidate for the best method, which appears to be DIRECTLINGAM.

The effect of session length seems to have a double impact: the longer the session length, the better the sensitivity results for connection and direction, but it also worsens the results of the specificity for both connection and direction. The effects are not very important, but they exist nonetheless. It would seem that having a longer timeseries benefits the correct identification of negatives, but also creates an overfit of kinds and is not capable of identifying negatives as properly. Of all the algorithms, DIRECTLINGAM shows the least impact, specially in the sensitivity of connections.

On the other hand, the effect of number of subjects per scanning session does not seem to be very significant. The effects are practically nonexistent except for the iMAGES algorithm, which manages to obtain better results with lower amounts of subjects, but we will not take it into consideration, since the algorithm performs very bad in general.

Lastly, the repetition seems to have a slight impact on the results, which seems to be identical in nature with the session length: results for the sensitivity of connections and direction worsen when the TR increases, and the results for the specificity of connections and directions improve when the TR increases. Again, as with the session length: the change is not very noticeable (changes are at most a difference of 0.1), and the effect is reduced on some algorithms, such as DIRECTLINGAM. The conclusion is the same as before, there is a trade-off scenario between sensitivity and specificity.

If we analyze the standard deviations of all 10 runs from all the possible scenarios (Figure 5.10) we can also make some additional remarks. First of all, the iMAGES algorithm is not very reliable, since the standard deviations are uncharacteristically high for the sensitivity and specificity of the directions. On the other hand, the DIRECTLINGAM method is very reliable, since it practically produces a standard deviation of zero for most of the scenarios, including the sensitivity of the direction, where most algorithms have the highest standard deviations.

The standard deviation results also show that the connections are more reliably calculated than the directions, further reinforcing the well known fact that most algorithms perform well in the detectection of connections, but not so well for directions.

Lastly, we can observe that the results with a low count of subjects have higher standard deviations, for both connections and directions, and it is more noticeable for the directions. Even if there was a previous trade-off in sensitivity and specificity, it must be done taking into account that the standard deviations are higher, indicating less reliability of the results. The same cannot be said for the session length and TR, since no apparent worthwhile variation in standard deviation can be appreciated.

Chapter 6

Discussion

Our study has both confirmed some previous findings and brought new ones as well. As expected, the results for connection detection are higher than the ones for orientation, since the key problem lies in orientation. Furthermore, the increased session length and decreased repetition time both improve the sensitivity of direction and connection by a small factor, but they also worsen the specificity of direction and connection as well. This trade-off scenario must be taken into account while also noticing that the effect is not very substantial.

However, one algorithm, DIRECTLINGAM, is not affected as much by these changes as the others, since it presents very high sensitivity and specificity for both connection and direction, and performs equally well independent of the underlying network topology and scenario, which is not true for the other algorithms. It also obtains remarkable results in the sensitivity of the direction, which as we have said before, is the main problem that these methods face. As such, it is safe to assume that DIRECTLINGAM is the best performer out of all the methods presented in this work.

Nevertheless, the results obtained by the Tabu, MMHC and GOBNILP approaches are also quite interesting, with GOBNILP obtaining slightly worse results in comparison. They even obtain better scores overall on the specificity of the connections, and in MMHC's case, outperforms DIRECTLINGAM in specificity of both connection and direction. Still, this is outshadowed by their poor reliability,since they do not perform well with network topologies that have a lower count of nodes, whereas DIRECLINGAM performs equally well for all topologies.

There does appear to be a difference in results between resting-state and boxcar for the same scenario. It seems that when the node count is low, the score-based tabu and hybrid MMHC obtain noticeably worse results in direction, while maintaining the same results in connection. The most noticeable changes can be seen in MMHC, which warrants a further investigation of hybrid-based approaches. A possible explanation is that hybrid and score-based algorithms are not robust enough to deal with the added complexity of an external stimulus, or because when an external stimulus is added, the data becomes even more non-Gaussian, debilitating these algorithms that work upon the premise that the data is Gaussian.

Finally, regarding which algorithms perform better or worse, we can say that based on our data, the iMAGES algorithm performs very poorly. This could be due to a lack of more data, since the standard deviation of iMAGES in the sensitivity and specificity

of its directions is very bad, but since the standard deviation for the connection data is not as bad, it is probably due to the nature of the algorithm. It is also worth mentioning again the iMAGES is the only group-level approach of all the algorithms presented here.

Regarding the comparison of the normal vs excellent scenario, we can conclude that from an economical and logistical point of view, if we only were to apply BN structure learning algorithms, it would not be worth the effort to obtain more expensive machines capable of lower TRs and to extend the session length too long. However, what would be acceptable is to have more subjects, since the reliability of the results improves, as seen in Figure 5.10. As we have seen, there are methods, such as DIRECTLINGAM, capable of obtaining quality results with very little data, so it is to be expected that in the future, better performing algorithms will appear, such a newer versions of the LINGAM family. As such, if one had to decide where to focus additional budget, it would make sense to direct it to subject recruitment.

Chapter 7

Conclusions and future research

The work presented in this Master's Thesis has been a difficult journey, but it ended with very interesting results, which could be summed up as the LINGAM family again outperforms all other algorithms and that task-based simulated fMRI data is of importance, since results can vary for other types of algorithms.

Since we have not found public available task-based simulated fMRI data, we made our data available for public use, in order to encourage more studies like this one. We also published the different scripts that had to be done in order to call the different software packages and programs sequentially. One other problem that we encountered was the generation of the metrics, and had to resort to handcrafted Python scripts, which were also made available. It is quite interesting, and also a surprise, to see how many other libraries dedicated to more popular methods, such as neural networks, are practically a "all-in-one" solution that even output the metrics directly, while in BNs, you have to find your own way out of many problems, such as creating metrics, since the methods are not as popular as others. All the data and code is available at: `https://github.com/Jorge-Diez/Balloon-BN-algorithms`.

Five distinct lines of future research in relation to our work can be identified:

- Study of background knowledge. We have seen that some algorithms perform reasonable well when detecting connections, and others do not when detecting orientations. If we know that specific algorithms work best under specific circumstances, we could generate a knowledge file with a method that has a high specificity of sensitivity for connections, and feed the knowledge into another method that excels at orientation detection.
- Further study of score-based and hybrid approaches. We have seen that these methods seem to work reasonable well, specially the MMHC method, but perform poorly under specific circumstances. A study could be done and exclusively dedicated to score-based and hybrid approaches in order to determine if all perform similarly, and if that is the case, single out the reason for poor performance with boxcar simulations.
- Task-based fMRI studies. We have seen that task-based simulated fMRI data has an impact (depending on the method) on the results. Knowing this, a study done with only task-based simulated fMRI data in order to analyze the best working method for this data alone would be interesting. It could also be merged

with the previous line of work.

- Study with biological data. Once an algorithm has been proved to work remarkably well with adequate statistical significance, it could be introduced to empirical data. In order to do this, due to the limitations of not knowing the inherent network structure with complete exactitude, data with a strong consensus would be needed. For that to happen, neuroscientists with training in neuroimaging and/or access to fMRI machines would be needed.
- Using the generated BN for interpretability and inference. As we stated in our literature review, BNs have the advantage of interpretability and also give users the capability of performing inference. For neuroimaging, we can use BNs to quickly interpret complex data that arises from fMRI and therefore, obtain important information quickly and easily, without the help of complicated additional processes. Inference could also be performed on the resulting BNs to obtain even more valuable data such as the flow of information in the brain and the role that different brain areas play in neurodegenerative diseases. This last line of research requires experienced neuroscientists and/or neuroimaging researchers to validate the corresponding findings.

Bibliography

[1] Dreizen, P. The Nobel prize for MRI: A wonderful discovery and a sad controversy. *The Lancet*, 363(9402), 2004, p.78.

[2] Pessoa, L. Understanding brain networks and brain organization. *Physics of Life Reviews*, 11(3), 2014, pp. 400-435.

[3] Orban, P. et al. Multisite generalizability of schizophrenia diagnosis classification based on functional brain connectivity. *Schizophrenia Research*, 192, 2018, pp. 167-171.

[4] Belmonte, M. Autism and abnormal development of brain connectivity. *Journal of Neuroscience*, 24(42), 2004, pp. 9228-9231.

[5] Zhang, H. et al. Resting brain connectivity: Changes during the Progress of alzheimer disease. *Radiology*, 256(2), 2010, pp. 598-606.

[6] Zeng, L. et al. Identifying major depression using whole-brain functional connectivity: A multivariate pattern analysis. *Brain*, 135(5), 2012, pp. 1498-1507.

[7] Friston KJ. Functional and effective connectivity: A review. *Brain Connectivity*;1(1), 2012. pp. 13-36.

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

[9] Ramsey, J. et al. Six problems for causal inference from fMRI. *NeuroImage*, 49(2), 2010, pp. 1545-1558.

[10] Smith, S. et al. Network modelling methods for FMRI. *NeuroImage*, 54(2), 2011, pp. 875-891.

[11] Huettel, S. et al. An introduction to fMRI, In *Functional Magnetic Resonance Imaging*. 2nd ed. Sunderland, Massachusetts: Sinauer, 2014, pp. 1-31

[12] Mosso A. Applicazione della bilancia allo studio della circolazione sanguigna dell'uomo, vol. XIX, *Atti R Accad Lincei Mem Cl Sci Fis Mat Nat*, 1884, pp. 531-543.

[13] Hoult, D. and Bhakar, B.. NMR signal reception: Virtual photons and coherent spontaneous emission. *Concepts in Magnetic Resonance*, 9(5), 1997, pp. 277-297.

[14] Purves, D., et al. Studying the nervous systems of humans and other animals, In *Neuroscience*. Sunderland, Massachusetts.: Sinauer Associates, Inc. 2012, pp. 1-31.

[15] Huettel, S. et al, From neuronal to hemodynamic activity", In *Functional Magnetic Resonance Imaging*. 2nd ed. Sunderland, Massachusetts: Sinauer. 2014, pp. 159-191.

[16] Ogawa, S. et al . Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, 87(24), 1990, pp. 9868-9872.

[17] Kizilirmak, J. et al. Neural correlates of learning from induced insight: A case for reward-based episodic encoding. *Frontiers in Psychology*, 7, 2016, 1693.

[18] Hillman, E. Coupling mechanism and significance of the BOLD signal: A status report. *Annual Review of Neuroscience*, 37(1), 2014, pp. 161-181.

[19] Jenkinson, M. et al. FSL. *NeuroImage*, 6.2, 2012, pp. 782-790.

[20] Edlow, B. et al. Functional MRI and outcome in traumatic coma. *Current Neurology and Neuroscience Reports*, 13(9), 2013, 375.

[21] Poldrack, R.. The role of fMRI in cognitive neuroscience: Where do we stand?. *Current Opinion in Neurobiology*, 18(2), 2008, pp. 223-227.

[22] Stephan, K. and Friston, K. Analyzing effective connectivity with functional magnetic resonance imaging. *WIREs Cognitive Science*, 1(3), 2010, pp. 446-459.

[23] Macey, P. et al. Functional imaging of autonomic regulation: methods and key findings. *Frontiers in Neuroscience*, 9, 2016

[24] Buxton, R. et al. Dynamics of blood flow and oxygenation changes during brain activation: The balloon model. *Magnetic Resonance in Medicine*, 39(6), 1998 pp. 855-864.

[25] Pearl, J. . *Probabilistic Reasoning in Intelligent Systems*. San Mateo, Calif.: Morgan Kaufmann Publishers. 1998

[26] Cobb B.R. et al. Bayesian network models with discrete and continuous variables, In: Advances in Probabilistic Graphical Models. *Studies in Fuzziness and Soft Computing*, vol 213. Berlin, Heidelberg: Springer, 2007, pp. 81-102.

[27] Liao, W. and Ji, Q. Learning Bayesian network parameters under incomplete data with domain knowledge. *Pattern Recognition*, 42(11), 2009, pp. 3046-3056.

[28] Masegosa, A. and Moral, S. An interactive approach for Bayesian network learning using domain/expert knowledge. *International Journal of Approximate Reasoning*, 54(8), 2013, pp. 1168-1181.

[29] Lauritzen, S. and Spiegelhalter, D. Local computations with probabilities on graphical structures and their application to expert Systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2), 1998, pp. 157-194.

[30] Cooper, F.G (1990). The computational complexity of probabilistic inference using Bayesian belief networks *Artificial Intelligence*, 42 (2-3). 1990, pp. 393-405

[31] Henrion, M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence*, 1988, pp. 149-163.

[32] Fung, R. and Chang, K. Weighing and integrating evidence for stochastic simulation in Bayesian networks. *Uncertainty in Artificial Intelligence*, 1990, pp. 209-219.

[33] Spirtes, P. and Glymour, C. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1). 1991, pp. 62-72.

[34] Schwarz, G. Estimating the dimension of a model. *The Annals of Statistics*, 6(2), 1978.

[35] Koller, D. and Friedman, N. "Structure Learning in Bayesian Networks" in *Probabilistic graphical models*. Cambridge, Mass: The MIT Press. 2012

[36] Scutari, M. et al. Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms. *International Journal of Approximate Reasoning*, 115, 2019. pp.235-253.

[37] Tsamardinos, I. et al. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1), 2006, pp. 31-78.

[38] Koller, D. and Friedman, N. Partially observed data, In *Probabilistic graphical models*. Cambridge, Mass: The MIT Press. 2012.

[39] Scanagatta, M. et al. A survey on Bayesian network structure learning from data. *Progress in Artificial Intelligence*, 8(4), 2019, pp. 425-439.

[40] Tsamardinos, I. et al. Time and sample efficient discovery of Markov blankets and direct causal relations. *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* - KDD 2003, pp. 673-678

[41] Kalisch, M. and Bühlmann, P.: Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*. 8, 2017, pp. 613– 636

[42] Meek, C. Graphical models: Selecting causal and statistical models. 1997

[43] Chickering, D. M. Optimal structure identification with greedy search. *Journal of Machine Learning Research* (3), 2002, pp. 507-554

[44] Ramsey, J. et al. A million variables and more: The fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics* 3(2), 2017, pp. 121-129.

[45] Shimizu, S. et al. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7. 2006, pp. 2003-2030

[46] Hyvarinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3), 1999, pp. 626-634.

[47] Shimizu, S. et al. DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research*, 12, 2011, pp. 1225-1248.

[48] Tashiro, T. et al. ParceLiNGAM: A causal ordering method robust against latent confounders. *Neural Computation*, 26(1), 2014, pp. 57-83.

[49] Cussens, J. Bayesian network learning with cutting planes, In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*. 2011, pp. 153–160.

[50] Jaakkola, T. et al. Learning Bayesian network structure using LP relaxations, In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* 2010, pp. 358-365

[51] Milan, S. How matroids occur in the context of learning Bayesian network structure, In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*. 2015.

[52] Cussens, J. GOBNILP: Learning Bayesian network structure with integer programming, In *Proceedings of the 10th International Conference on Probabilistic Graphical Models*. 2020.

[53] Ramsey, J. et al. Non-Gaussian methods and high-pass filters in the estimation of effective connections. *NeuroImage*, 84, 2014, pp. 986-1006.

[54] Ramsey, J. et al. Six problems for causal inference from fMRI. *NeuroImage*, 49(2), 2010, pp. 1545-1558.

[55] Sanchez-Romero, R. et al. Estimating feedforward and feedback effective connections from fMRI time series: Assessments of statistical methods. *Network Neuroscience*, 3(2), 2019, pp. 274-306.

[56] Perez, C.A. et al. Discovering effective connectivity among brain regions from functional MRI data', *International Journal of Computers in Healthcare*, 1(1), 2010, pp. 86–102

[57] Liu, T. and Falahpour, M. Vigilance effects in resting-state fMRI. *Frontiers in Neuroscience*, 14. 2020, 321

[58] Eklund, A. et al. Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates, In *Proceedings of the National Academy of Sciences*, 113(28), 2016, pp. 7900-7905.

[59] Elliott, M. et al. What is the test-retest reliability of common task-functional MRI measures? New empirical evidence and a meta-analysis. *Psychological Science*, 31(7), 2020, pp. 792-806.

[60] Russell, S.J. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition. 2009.

[61] Hanakawa, T. The role of rostral brodmann Area 6 in mental-operation tasks: an integrative neuroimaging approach. *Cerebral Cortex*, 12(11), 2012, pp. 1157-1170.

[62] Friston, K.J. et al. Dynamic causal modelling. *Neuroimage* 19 (3), 2003, pp. 1273–1302.

[63] Jahanian H. et al. Advantages of short repetition time resting-state functional MRI enabled by simultaneous multi-slice imaging. *Journal of Neuroscience Methods*. 2019, pp. 122-132

[64] Human Connectome Project, *HCP 3T Imaging Protocol Overview* https://www.humanconnectome.org/hcp-protocols-ya-3t-imaging

[65] Scheines, R. et al. The TETRAD Project: Constraint based aids to causal model specification, *Multivariate Behavioral* Research, 33:1, 1998, pp. 65-117

[66] Scutari, M. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3), 2010, pp. 1–22.