Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Causal Inference in Time Series for Identifying Molecular Fingerprints during Sleep

Master Thesis

Ričards Marcinkevičs

August 7, 2019

Department of Mathematics,
Seminar for Statistics, ETH Zürich

Supervisor: Prof. Dr. Joachim M. Buhmann; Advisor: Đorđe Miladinović

Department of Computer Science, Institute for Machine Learning, ETH Zürich

Abstract

The relationship between human sleep and metabolism has not yet been studied systematically and well understood. In this thesis, we investigate the association between sleep stages and exhaled breath mass spectrometry in the framework of Granger causality.

We first introduce a scalable neural network approach for inferring nonlinear Granger causality between continuously- and categorically-valued variables. We test this technique on a wide range of simulated datasets with differing degrees of nonlinearity and demonstrate that, in many settings, it outperforms the conventional linear vector autoregressive model. The datasets, on which validation is performed, include the Lorenz 96 system and rich and realistic simulations of fMRI time series.

By leveraging the developed method and the bootstrapping technique, we then identify Granger causes and effects of sleep phase transitions from breathomics data. Many ions are discovered in the causal analysis of time series; and the results suggest that metabolism and sleep regulate each other mutually. Among these discoveries we find isoprene, a compound the association of which with sleep phases was reported in the literature before [1, 38]. This analysis sheds some light on the relationship between sleep and volatile organic compounds in human breath and opens many venues of future research.

Acknowledgements

I would first like to thank Professor Dr. Joachim M. Buhmann and his Ph.D. student Đorđe Miladinović for their supervision of this thesis. Our weekly meetings with Đorđe were very insightful and critical for the progress of the project. I am very grateful for his supportive guidance and openness to my ideas. It was a pleasure to work on this challenging and captivating interdisciplinary topic with the assistance of such motivated advisor.

My gratitude also goes to all our collaborators from the Zenobi group at the Department of Chemistry and Applied Biosciences of the ETH Zürich, the Brown Lab at the Institute of Pharmacology and Toxicology of the UZH, the USZ and the UKBB. Without them this project would have not been possible. Our meetings improved my understanding of our goals and the broader context. Especially, I would like to thank Nora Nowak who kindly helped me with mass spectrometric data pre-processing and shared her analysis of the results that brought a domain-specific perspective.

Last but not least, I would like to acknowledge all of the teaching staff at the Seminar for Statistics and the Department of Computer Science. Throughout two years of my master's degree I learnt a lot, and the knowledge acquired from them enabled me to produce this thesis, which, I hope, will become a fitting contribution to Zürich Exhalomics project.

Nomenclature

Abbreviations

AUPR	Area under precision-recall curve
AUROC	Area under receiver operating characteristic curve
GC	Granger causality
GC-LSTM	Granger causal long short-term memory
GC-MLP	Granger causal multilayer perceptron
LSTM	Long short-term memory
MLP	Multilayer perceptron
MS	Mass spectrometry
NREM	Non-REM sleep
PSG	Polysomnography
REM	Rapid eye movement sleep
RNN	Recurrent neural network
TRGC	Time-reversed Granger causality
VAR	Vector autoregressive model

Contents

Chapter 1

Introduction

Exhaled breath analysis is an interdisciplinary field of study which has attracted researchers since the times of ancient Greeks, who correctly recognised breath odours as symptoms of diseases and used the sense of smell as a diagnostic tool [5]. Today, *mass spectrometry* (MS), an analytical technique, facilitates an objective and quantitative description of the exhalome, thus, allowing the use of statistical and machine learning models. Breath metabolomics [40], or *breathomics*, is based on the principle that different physiological statuses, e.g. diseases, of the subject may lead to distinct profiles of volatile organic compounds (VOC) within the exhale [40]. This makes breath analysis a potentially powerful non-invasive tool with applications in disease diagnosis and monitoring and even in exploration of complex relationships between metabolites and physiological conditions.

The goal of this thesis is to analyse time course breathomics data and study links between compounds in exhalome and the stages of sleep that are derived from polysomnography (PSG), acquired simultaneously with breath mass spectra. We develop a machine learning model, based on the concept of Granger causality, to recover complex dependencies in these multimodal time series data and infer causal relationships between variables. This model is sufficiently general to represent nonlinearity, non-additive variable interactions and to simultaneously include both continuously- and categorically-valued time series.

1.1 Problem Description

The problem tackled by the thesis can be formalised as follows. We assume that we are given N replicates of multivariate time series retrieved from different experimental units – in this case, single individuals, i.e. subjects. These multivariate time series include:

- A categorically-valued target variable $\{Y_t\}_{t \in \{1,...,T\}} \in \{W, N_1, N_2, N_3, R\}$ which represents the sleep stage across T time steps. Herein, W corresponds to the wakefulness phase, whereas R denotes rapid eye movement (REM) sleep and N_1, N_2, N_3 are non-REM (NREM) sub-phases.

- M continuously-valued time series $\left\{X_t^j\right\}_{t \in \{1,...,T\}}$, where $j = 1, ..., M$ and X_t^j corresponds to the relative intensity of ion j in the mass spectrum of exhaled breath at time step t. Observe, that the mass spectrum at time t is given by vector $\mathbf{MS}_t = \begin{bmatrix} X_t^1 & X_t^2 & \cdots & X_t^M \end{bmatrix}^\top \in \mathbb{R}^M$.

The goal is then to identify metabolites that are causally related to sleep stages, i.e. metabolites that *drive* the sleep stage, denoted by $X^j \longrightarrow Y$, and metabolites that are *driven by* the stage, $Y \longrightarrow X^j$. In general, this knowledge could be helpful for fundamental understanding of metabolic processes that occur during sleep.

1.2 Contributions

The key contribution of this work is the development of the whole pipeline from pre-processing to causal inference for breathomics and sleep stage time series data.

- Inspired by [73], we introduce a regularised neural network model for inferring Granger causality between multiple time series. The architecture of this network differs from the design considered in the literature before [73].

- We investigate the performance of the inference method in presence of both categorically- and continuously-valued time series generated from a variety of linear and nonlinear autoregressive models. Additionally, we perform simulation experiments on the 'real world' breathomics data.

- We apply bootstrapping [18] to quantify uncertainty about causal relationships. This procedure allows identifying relationships that are invariant across all subjects and, thus, can be useful when there exists unwanted variation because of differences between experimental units.

- Last but not least, we use the introduced method for inferring nonlinear Granger causality to discover ions that cause and are caused by different phases of sleep from synchronised time course mass spectrometry and sleep stage data.

1.3 Content

The thesis consists of seven chapters, most of them covering different aspects of the conducted data analysis.

1. **Background** (see Chapter 2): we introduce the reader to the context and theoretical background of this work. First, we briefly review basic principles of mass spectrometric analysis and sleep stage scoring. We explain the concept of Granger causality and discuss conventional approaches to inferring it. Finally, we also provide a short overview of the neural network models, relevant to the the techniques described in further chapters.

2. **Pre-processing & Exploratory Data Analysis** (see Chapter 3): we explain the pre-processing procedures applied to data before to the causal time series analysis. Namely, we discuss mass spectrum and batch normalisation, time series standardisation and denoising. Additionally, we explore the data using dimensionality reduction techniques.

3. **Inferring Granger Causality with Neural Networks** (see Chapter 4): after reviewing related research work, we introduce a nonlinear approach to inferring Granger causality based on feedforward neural networks. We provide implementation details of our technique and discuss a principled way for quantifying uncertainty about inferred causal relationships that uses bootstrapping.

4. **Simulation Experiments** (see Chapter 5): we perform several controlled simulation experiments to compare the performance of neural networks to the conventional linear vector autoregressive model. We consider linear and nonlinear multivariate time series with continuously- and categorically-valued variables. We also examine the empirical performance of time-reversed Granger causality inference.

5. **MS Data Analysis** (see Chapter 6): we use the proposed model alongside with the bootstrap method to infer Granger causes and effects of sleep phases among studied positive and negative ions. In addition, we perform controlled simulation experiments on the breathomics data and validate the model to corroborate the inference results.

6. **Discussion & Conclusions** (see Chapter 7): we reflect on the results of the thesis and finalise it by considering possible directions for further research.

Supplementary materials can be found at the end in four appendices.

Chapter 2

Background

In this thesis we analyse time course mass spectrometry and sleep stage data using a specially tailored machine learning model. Therefore, a basic understanding of mass spectrometry, sleep physiology and the underlying statistical framework is crucial for the comprehension of the whole inference pipeline, from the raw data to the techniques applied. In this chapter, we briefly review these topics to provide the reader with the general context for the rest of the paper.

2.1 Mass Spectrometry

Mass spectrometry [14] is an analytical technique with a wide range of applications, e.g. in metabolomics and proteomics, pollution and food control, reaction physics and kinetics etc. The goal of MS analysis is to quantify relative abundances of ions within a compound. The physical property that it measures are mass-to-charge-ratios.

The process of MS measurements can be roughly described as follows. First, gaseous ions have to be produced from molecules. For that purpose, different ion sources can be used, for instance, electron, chemical or field ionisation [14]. The mass analyser then separates charged particles based on their mass-to-charge ratios into beams; and, finally, the detector measures abundances of ions and transforms them into electric signals that can be transferred to a computer. Figure 2.1, taken from [10], depicts a schematic of a simple mass spectrometer. The final output of the analysis is referred to as *mass spectrum* and is usually presented in the form of a table or a bar plot (see Figure 2.2 for an example) containing mass-to-charge ratios alongside with corresponding relative abundances.

MS analysis has been subject to many technological advancements since the first spectrometer was built by J. J. Thompson in 1912 [14]. Major improve-

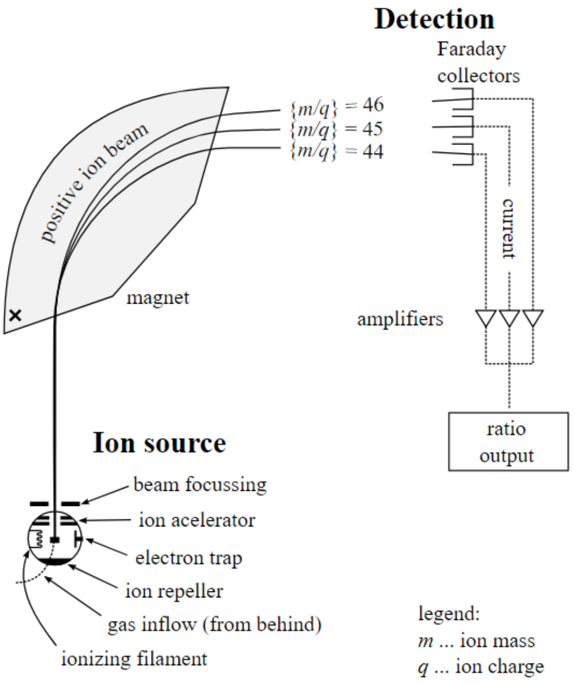


Figure 2.1: A simplified scheme of a mass spectrometer, taken from [10]. A sample is introduced and ionised; ions are propagated in electric and magnetic fields and, consequently, are separated.

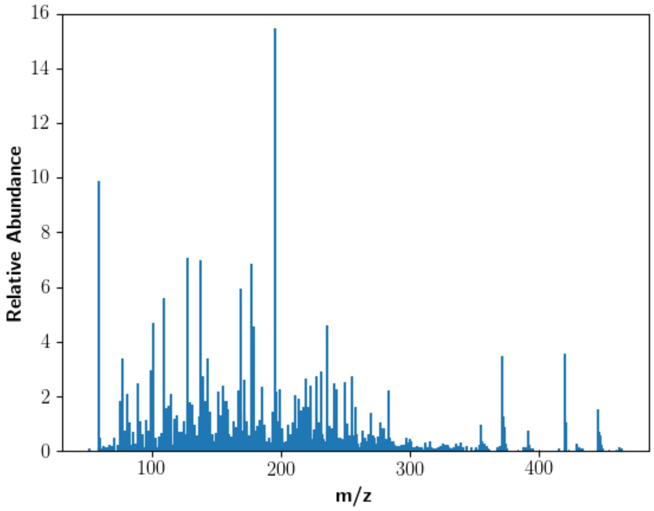


Figure 2.2: A mass spectrum of exhaled breath visualised as a bar diagram. Each bar corresponds to a measured mass-to-charge ratio (often denoted by m/z or m/q), whereas the height of a bar is determined by the relative abundance of the corresponding ion.

ments were made in the resolution and the sensitivity of measurements, new ionisation sources were discovered, a variety of mass analysers and detectors were introduced. Discussing these details is beyond the scope of this work, and we refer the interested reader to [14] for a comprehensive overview of various techniques.

2.2 Sleep Cycle and Sleep Stage Scoring

In our analysis, sleep stage is the target variable. The sleep is one of the less understood and explained physiologic sates. The human brain goes through different phases, namely, wakefulness, rapid eye movement sleep and sleep without rapid eye movements [65], which in turn consists of three sub-phases, often denoted by S1, S2, and S3 [47]. In reality, these stages are not strictly separated, i.e. transitions between them are not abrupt, but smooth and gradual [47].

In order to encourage consistency across laboratories and research groups, standard *sleep stage scoring* procedures were introduced, according to which a phase of sleep can be detected from the observation of body functions [47]. Usually several parameters from polysomnographic recordings are used; namely:

- *Electrooculogram* (EOG) records changes in electrical potential in the vicinity of the eye [34]. The potential results from charge differences between cornea and retina tissues and fluctuates because of eye movements.
- *Electroencephalogram* (EEG) measures electrical potentials on the surface of the head that originate from activity in brain neurons [9]. Compared to other brain imaging techniques, it is non-invasive and has a high time resolution.
- *Electromyogram* (EMG) tracks electrical activity within muscles [52]. Usually it is monitored at multiple sites, for instance, in chin and limbs [47].
- *Electrocardiogram* (ECG or EKG) captures electrical cardiac waves from electrodes placed on body surface [17]. These signals are a product of heart muscle contractions during the cardiac cycle.
- *Respiratory electrodes* measure a range of parameters associated with respiration, such as snore sounds, oronasal airflow, thoracic and abdominal effort [47].

Characteristics of the aforementioned signals differ across sleep phases. During NREM stages overall relaxation in physiological activity can be observed, wheres REM sleep leads to a substantial rise in it [47]. In particular, REM

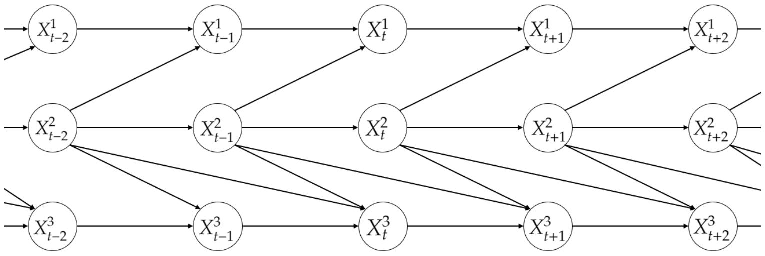


Figure 2.3: Full time graph of a trivariate time series. For every time point $t \in \mathbb{Z}$, X^2_t causally influences X^1_{t+1}, X^3_{t+1} and X^3_{t+2}. Note, that this structure features no instantaneous causal effects.

is described by rapid eye movements, the EEG patterns similar to wakefulness, irregular breathing etc. Based on observations of such specific patterns, formal rules were developed to score each stage from polysomnography. The most commonly used regulations come from Rechtschaffen and Kales method (or *R and K* rules) that is documented in American Academy of Sleep Medicine (AASM) scoring manual [47].

2.3 Time Series Causality

In this thesis we adopt the approach of causal inference, since often the fundamental goal of biological or medical research is to discover causal relationships, rather than mere associations. Causal reasoning allows making statements about changes in outcomes that can occur after interventions on certain variables. Therefore, causality is a more powerful framework than probabilistic reasoning. However, the price of this power is the non-triviality of *causal discovery*, an inverse problem of identifying causal structure from a joint probability distribution [60]. This subsection introduces the reader to causal analysis of time series. The review provided herein closely follows '*Time Series*' chapter by Peters et al. in [60].

Consider multivariate (continuously-valued) time series $\{\mathbf{X}_t\}_{t\in\mathbb{Z}}$ where $\mathbf{X}_t = \begin{bmatrix} X^1_t & X^2_t & \cdots & X^p_t \end{bmatrix}^\top$. Usually we assume that $\{\mathbf{X}_t\}_{t\in\mathbb{Z}}$ is strictly stationary [60]; this assumption is made by many statistical techniques for time series analysis. Causal influences within this series can be then visualised using the *full time graph* [60], an infinite directed acyclic graph (DAG) with nodes corresponding to X^j_t, for $j \in \{1, 2, ..., p\}$ and $t \in \mathbb{Z}$, and directed edges – to cause-effect relationships. An example of such graph for a trivariate time series is shown in Figure 2.3. It is important to note, that, in general, causal effects can be *instantaneous* [60], i.e. the full time graph may contain edges of the form $X^i_t \longrightarrow X^j_t$, for $i \neq j$. Often it is more convenient to summarise the infinite full graph by the corresponding *summary graph* [60], the vertices of which represent variables across whole time. To construct this simplified graph, for all $i \neq j$, if, for some $k \in \{0\} \cup \mathbb{N}$ and some $t \in \mathbb{Z}$, there is edge $X^i_{t-k} \longrightarrow X^j_t$ in the full graph, then $X^i \longrightarrow X^j$ has to be added to the summary graph. For instance, the summary graph corresponding to Figure 2.3 is given by $X^1 \longleftarrow X^2 \longrightarrow X^3$.

Similar to causal reasoning with *i.i.d.* data, we can also consider *interventions* on time series. One way to formalise interventions are *structural causal models* (SCM) [60]. If $\{\mathbf{X}_t\}_{t\in\mathbb{Z}}$ admits an SCM, then, for some $K \geq 0$ and for all $j \in \{1, 2, ..., p\}$:

$$X^j_t := f^j\left(\left(\mathbf{PA}^j_K\right)_{t-K}, \left(\mathbf{PA}^j_{K-1}\right)_{t-K+1}, ..., \left(\mathbf{PA}^j_{\mathbf{0}}\right)_t, N^j_t\right), \tag{2.1}$$

where N^j_t are jointly independent noise, or innovation, terms, and $\mathbf{PA}^j_k$ denotes the set of all variables that influence X^j with lag k. An intervention in such model can be easily represented as a replacement of the appropriate structural assignments (as in Equation 2.1).

A well-studied special case of the SCM is the *vector autoregressive model* (VAR) [46], which assumes that f^j are linear. In VAR of order K, we have:

$$\mathbf{X}_t := \boldsymbol{\nu} + \sum_{k=1}^{K} \mathbf{A}_k \mathbf{X}_{t-k} + \mathbf{N}_t, \tag{2.2}$$

where $\boldsymbol{\nu} \in \mathbb{R}^{p\times 1}$ is a fixed intercept vector, $\mathbf{A}_k \in \mathbb{R}^{p\times p}$ are fixed matrices of coefficients, and $\mathbf{N}_t = \begin{bmatrix} N^1_t & N^2_t & \cdots & N^p_t \end{bmatrix}^\top$ is a zero-mean random vector with $\mathbb{E}\left[\mathbf{N}_t\mathbf{N}_t^\top\right] = \mathbf{\Sigma}_N$ and $\mathbb{E}\left[\mathbf{N}_t\mathbf{N}_s^\top\right] = \mathbf{0}$, for $t \neq s$. This simple, yet practical model has been instrumental in structural analysis of multivariate time series, particularly, in inferring Granger causality.

2.3.1 Granger Causality

One of the most popular approaches to causal time series analysis is Granger causality (GC or G-causality), introduced by C. W. J. Granger in 1969 [29] in the context of econometric models. Since then Granger causality and its extensions have been applied in many domains, including, but not limited to economics [57], climatology [55], neuroscience [63] and metabolomics [16].

According to [19], two properties that a definition of causality should ideally encapsulate are (1) *temporal precedence*: the cause precedes the effect and (2) *physical influence*: intervening on the cause changes the effect. Granger's concept of causality focuses on the former; intuitively, if X is a cause of Y and, thus, temporally precedes it, the past of X should be useful for predicting the future of Y [46]. More formally, Granger causality can be defined as follows [19]. Let us consider stationary time series $\{X_t\}_{t\in\mathbb{Z}}$ and $\{Y_t\}_{t\in\mathbb{Z}}$. Let

$\mathcal{I}^*(t-1)$ be an information set containing all information available in the universe up to time $t-1$, and let $\mathcal{I}^*_{-X}(t-1)$ be the same set as $\mathcal{I}^*(t-1)$, but with values of time series X removed (up to time $t-1$). We say that X *Granger-causes* Y iff

$$Y_t \not\perp\!\!\!\perp \mathcal{I}^*(t-1) | \mathcal{I}^*_{-X}(t-1), \tag{2.3}$$

for all $t \in \mathbb{Z}$. This definition for the bivariate case can be easily extended to multivariate time series by including all of the variables into set $\mathcal{I}^*(t-1)$. Realistically, we might be not able to record all variables. Therefore, usually $\mathcal{I}^*(t-1)$ contains only what was measured. In this case, statements about Granger causality between observed time series are valid only if the considered set of variables is *causally sufficient* [60], i.e. if there exist no unobserved time series $\{Z_t\}_{t\in\mathbb{Z}}$ which is a common cause of two observed variables. In this case, Z_t is a *confounder*, or a *latent variable*.

Granger Causality in VAR

In practice, Granger causality is often inferred by assuming some time series model, for, instance, VAR. It can be shown that in VAR Granger causality can be determined from zero constraints on the coefficients [46].

Recall the model for multivariate time series X_t given by Equation 2.2. We have that X^i does not Granger-cause X^j iff, for all $k \in \{1, 2, ..., K\}$, $(\mathbf{A}_k)_{ji} = 0$. However, normally model coefficients are unknown and have to be estimated from data. There exist tests for zero constraints on VAR coefficients involving statistics with known (asymptotic) reference distributions [46], for example, the F-test can be used.

A natural *exhaustive* procedure to infer the complete structure of the given time series is based on model comparison, carried out for each potential cause-effect pair [3]. Let us consider observing target time series $\{Y_t\}_{t\in\{1,...,T\}}$ and predictor variables $\left\{X^j_t\right\}_{t\in\{1,...,T\}}$, for $j = 1, ..., p-1$. To investigate if some variable X^c drives Y, we need to compare two models [3, 60]:

$$Y_t = \sum_{k=1}^{K} \alpha_k Y_{t-k} + \sum_{\substack{j=1\\ j\neq c}}^{p-1} \sum_{k=1}^{K} \beta_{jk} X^j_{t-k} + N^Y_t \tag{2.4}$$

and

$$Y_t = \sum_{k=1}^{K} \widetilde{\alpha}_k Y_{t-k} + \sum_{j=1}^{p-1} \sum_{k=1}^{K} \widetilde{\beta}_{jk} X^j_{t-k} + \widetilde{N}^Y_t, \tag{2.5}$$

where N^Y_t and $\widetilde{N}^Y_t$ are innovation terms, and α, β, $\widetilde{\alpha}$ and $\widetilde{\beta}$ are fixed unknown coefficients. Note, that the model given by Equation 2.4 is *restricted*, because all coefficients for X^c are set to 0; whereas the model defined by Equation 2.5

is referred to as *full*, since it includes X^c. After fitting these two regression models, a statistical test is conducted to compare their performance [3]. If model 2.5 is significantly better than 2.4, then we reject the null hypothesis and include edge $X^c \longrightarrow Y$ into the summary graph.

Lasso Granger Method

In order to learn the full causal structure of a p-dimensional multivariate time series using the exhaustive method explained above, $\mathcal{O}\left(p^2\right)$ regression models need to be fitted [3], this can be prohibitive for large p. To tackle high-dimensional causal inference problems, the *Lasso Granger method* was proposed [3, 45], which uses Lasso regression [74] for variable selection. In particular, this method utilises an adjusted loss function: the group Lasso penalty [81] is added to the residual sum of squares. In Lasso Granger analysis, we have to regress each variable on the rest [3, 45] only once; thus, in total, only $\mathcal{O}(p)$ regression models have to be fitted.

Let us again consider observing target time series $\{Y_t\}_{t\in\{1,...,T\}}$ and predictor variables $\left\{X_t^j\right\}_{t\in\{1,...,T\}}$, for $j = 1, ..., p-1$. Then, in Lasso Granger, the loss function for the regression of Y on X^j [45] is given by

$$\sum_{i=K+1}^{T}\left(y_i - \sum_{j=1}^{p-1}\sum_{k=1}^{K}\beta_{jk}x_{i-k}^{j} - \sum_{k=1}^{K}\beta_{pk}y_{i-k}\right)^2 + \lambda\sum_{j=1}^{p}\left\|\boldsymbol{\beta}_{\mathcal{G}_j}\right\|_2, \tag{2.6}$$

herein, lower-case letters denote observed values of the corresponding time series; K is the maximum lag at which (auto-)regressive relationships are considered; β_{jk} are fixed unknown coefficients that are estimands; λ is the regularisation parameter that controls the sparsity of the estimated coefficient vector; and $\mathcal{G}_j$ stands for a group of covariates with $\boldsymbol{\beta}_{\mathcal{G}_j} = \begin{bmatrix}\beta_{j1} & \beta_{j2} & \cdots & \beta_{jK}\end{bmatrix}^\top$. Note, that the covariates belonging to the same time series are grouped together. This is crucial for inferring Granger causality, because, if some series X^j does not Granger-cause Y, it is desirable that *all* of the coefficients that belong to X^j are shrunk towards zero during estimation. This kind of behaviour is achieved by introducing a group penalty as shown in Equation 2.6.

An important property of the graphical Lasso method in causal structure learning is its *statistical consistency* [3], which holds under certain assumptions about sparsity and dimensionality (the proof for Granger Lasso is due to [3]). In other words, the output of the method is consistent with the true Granger causality graph with probability 1 when $T, p \to \infty$. However, to achieve the consistency, regularisation parameter λ needs to be chosen appropriately [45], which is non-trivial.

Nonlinear Extensions

Arguably, a substantial limitation of using the vector autoregressive model in Granger causality analysis is the assumption that each time series value can be represented as a linear function of the past values of the target series and its causes. Such representation does not allow for nonlinearities and interactions between covariates.

There has been an extensive body of literature discussing various nonlinear approaches to Granger causality [2, 48, 69, 53, 24, 55, 77, 73]. These methods focus on performing nonlinear multivariate regression with feature spaces of kernel functions [2, 48, 69], random forests (RF) [24, 55] and neural networks, such as multilayer perceptrons (MLP) [53, 73], recurrent neural networks (RNN) [77] and long short-term memory (LSTM) [73].

Limitations of Granger Causality

While the concept of Granger causality is practically compelling and has been applied in many domains, it has some shortcomings and can be misleading in certain cases.

As mentioned before, one of the assumptions of GC analysis is causal sufficiency. In case of observing a causally insufficient set of variables, the results of inference may be misleading [60]. Peters et al. [60] provide an example of a bivariate time series consisting of prices for butter and cheese. The unobserved price for milk drives both of these variables. Due to differences in the production times, tests for Granger causality spuriously infer that the price of butter causes the price of cheese. It is worth mentioning that there exist methods to address this issue, for example, partial Granger causality proposed in [31].

There is also a number of quite artificial examples where Granger causality fails to produce adequate results [60], which will not be discussed herein. It is important to note, that the GC analysis does not consider instantaneous interactions between variables. This may lead to under- and overestimation of causal influence [60]. To alleviate this effect, it could be beneficial to include instantaneous terms into regression models (see Equations 2.4, 2.5 and 2.6) fitted for the inference, however, such analysis can lead to invalid conclusions.

2.4 Neural Networks

Neural networks (NN) provide a powerful framework for solving various problems of machine learning [28]. In recent years *deep learning* has, arguably, become one of the most popular state-of-the-art approaches to both

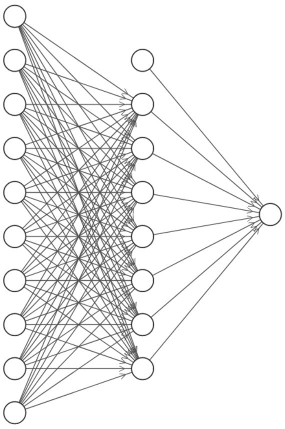


Figure 2.4: The schematic of a two-layer perceptron with ten inputs and one output. The flow of information is indicated by arrows. Note, that bias units are included as well. The graph was generated using NN-SVG tool [41].

supervised and unsupervised learning with applications ranging from playing the game of Go [68] to classifying cancerous lung tissues [78]. In this section we briefly review some models for supervised learning; particularly, those that can be used for time series prediction as 'building blocks' in Granger causality estimation (see Chapter 4). For a comprehensive overview the interested reader is referred to [28] or [6].

2.4.1 Feedforward Neural Networks

The most commonly known architecture are *feedforward neural networks*, or *multilayer perceptrons* (MLP). An MLP approximates some function $\mathbf{y} = f^*(\mathbf{x})$ by $f(\mathbf{x};\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are parameters. $f(\mathbf{x};\boldsymbol{\theta})$ maps inputs $\mathbf{x}$ to outputs $\mathbf{y}$, which can be probabilities of classes (as in classification) or arbitrary continuous values (as in regression) [28].

Feedforward networks can be represented by directed acyclic graphs [28], wherein information flows *forward* from input units to the output without feedback. Figure 2.4 depicts such representation for a simple feedforward architecture. To define feedforward neural networks more rigorously, let us consider a network with M inputs $\mathbf{x} = \begin{bmatrix} x_1 & \cdots & x_M \end{bmatrix}^\top$ and one output. The model can then be described as a series of functional transformations [6]. First, M linear combinations, or *activations*, are computed for the first layer of the network with size $M^{(1)}$ [6]:

$$a_j^{(1)} = \sum_{i=1}^{M} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \tag{2.7}$$

where $j = 1, ..., M^{(1)}$, $w_{ji}^{(1)}$ and $w_{j0}^{(1)}$ are parameters of the first layer, referred to as *weights* and *biases*, respectively. Subsequently, a non-linear *activation function* is applied to every a_j [6]:

$$z_j^{(1)} = h(a_j^{(1)}), \tag{2.8}$$

where $z_j^{(1)}$ are hidden units. Using these hidden unit values, activations for the second layer of size $M^{(2)}$ can be constructed as follows [6]:

$$a_j^{(2)} = \sum_{i=1}^{M^{(1)}} w_{ji}^{(2)} z_i^{(1)} + w_{j0}^{(2)}, \tag{2.9}$$

wherein $w_{ji}^{(2)}$ and $w_{j0}^{(2)}$ are weights and biases of the second layer and $j = 1, ..., M^{(2)}$. Thus, computations of such form are composed until the output layer is reached and the final output is evaluated.

It is important to note, that activation functions can differ for layers, especially, for the output. There exist various activation functions and their use depends on the specifics of the application. The default choice in modern neural networks, recommended in [28], is the *rectified linear unit* (ReLU) $h(x) = \text{ReLU}(x) = \max\{0, x\}$. The choice of the activation function at the output of a network is closely related to the type of the response we want to model. Consider having vector of activations $\mathbf{h}$ from the layer preceding the output. A sensible choice for the normally distributed response, e.g. in regression, is a linear function $\hat{\mathbf{y}} = \mathbf{W}^\top \mathbf{h} + \mathbf{b}$ [28]. On the other hand, for multinomially distributed responses, e.g. in multiclass classification, it is recommended to apply the *softmax* function [28]:

$$\hat{y}_i = \text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}, \tag{2.10}$$

where $\mathbf{z} = \mathbf{W}^\top \mathbf{h} + \mathbf{b}$. Note, that softmax assumes that the network has multiple outputs, which can be seen as probabilities of different categories.

Weights and biases, used by the network, form vector of parameters $\boldsymbol{\theta}$. When fitting, or *training*, a feedforward neural network, we have to find $\boldsymbol{\theta}$ that minimises the error function, specific to the task targeted by the network. For instance, in regression we may consider minimising the *sum-of-squares* error function [6]:

$$E(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i; \boldsymbol{\theta})\|_2^2 \tag{2.11}$$

For multiclass classification with C classes, the *cross-entropy* loss is appropriate [6]:

$$E(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \sum_{c=1}^{C} \{y_{ic} \ln(f(\mathbf{x}_i; \boldsymbol{\theta})_c) + (1 - y_{ic}) \ln(1 - f(\mathbf{x}_i; \boldsymbol{\theta})_c)\}, \tag{2.12}$$

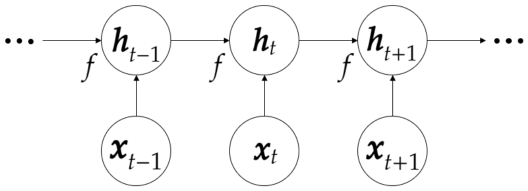


Figure 2.5: A schematic of a recurrent neural network without outputs. As can be seen, input (multivariate) time series values $\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}$ are transformed into hidden features $\mathbf{h}_{t-1}, \mathbf{h}_t, \mathbf{h}_{t+1}$. Note, that $\mathbf{h}_t$ is given by $f(\mathbf{h}_{t-1}, \mathbf{x}_t; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are network's parameters. The diagram is adapted from [28].

where y_{ic} stands for the c-th entry of vector $\mathbf{y}_i$. Minimising such error functions is an optimisation problem that is usually addressed by deriving the gradient of the error function w.r.t. $\boldsymbol{\theta}$ and performing *gradient descent* until convergence to a local optimum where $\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = 0$ [28]. A standard efficient method for computing gradients in multilayer feedforward networks is the *back-propagation* [28]. This algorithm is used as a subroutine in learning to evaluate the gradient based on one training point at a time in stochastic gradient descent (SGD) or on multiple points, forming a *mini-batch*, in mini-batch gradient descent.

An important advantage of feedforward neural networks over conventional statistical techniques for regression and classification is their flexibility in approximating functions. The *universal approximation theorem* provides support for this property. According to this theorem, any continuous function on a closed and bounded subset of $\mathbb{R}^n$ can be approximated arbitrarily well by a feedforward neural network with a linear output and at least one hidden layer with the logistic sigmoid activation function [28].

2.4.2 Recurrent Neural Networks

MLPs map a fixed number of features to their output, however, in sequential data, such as time series, it may be desirable for a neural network model to scale to varying numbers of features. *Recurrent neural networks* (RNN) use *parameter sharing* to facilitate this scalability [28]. This class of networks has become instrumental in forecasting time series [26, 36] and is closely related to the state space models [67].

Figure 2.5, adapted from [28], contains a schematic representation of a simple RNN without output. Let $\mathbf{x}_t$, for $t \in \mathbb{Z}$, denote (multivariate) time series values. An important building block of many recurrent networks are *hidden units*, which can be seen as 'summaries' of the present and the past inputs [28]. Hidden features at time step t are given by

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t; \boldsymbol{\theta}), \tag{2.13}$$

where $\boldsymbol{\theta}$ are parameters. These hidden features can then be mapped, using another transformation, to outputs $\mathbf{y}_t$, which represent, depending on a specific application, forecasts of the future values, probabilities of classes etc. Parameter sharing is possible, because transition function $f(\cdot)$ and parameters $\boldsymbol{\theta}$ can be shared across all time points [28]. As a consequence, the model can easily generalise to time series of varying lengths.

The training of RNNs does not require any specialised algorithms for computing gradients. The standard back-propagation procedure is sufficient and is often referred to as *back-propagation through time* (BPTT). Similarly to MLPs, the error function, evaluated based on the outputs and provided targets, is tightly coupled with what kind of response needs to be modelled.

There exist many variations to the general framework that is laid out in this subsection, including the most powerful state-of-the-art sequential approach – *long short-term memory* (LSTM) RNNs [28]. In this work we are not particularly interested in the predictive power of the most advanced models per se, our goal is to use them as a subroutine in the estimation of Granger causality. Therefore, it is beyond the scope to cover intricate details, which the interested reader can find in [28].

Chapter 3

Pre-processing & Exploratory Data Analysis

In this chapter we describe pre-processing methods for time-course mass-spectrometry data applied prior to estimating Granger causality. In particular, we provide details about normalisation of mass spectra and batches, time series normalisation and denoising. These procedures are crucial in removing undesirable biologically irrelevant artefacts. We also perform some initial exploratory data analysis. To begin with, we explain the experimental setup used for the acquisition of data.

3.1 Experimental Setup and Data

The dataset that is considered in this thesis was collected at the Department of Chemistry and Applied Biosciences of ETH Zürich by the Zenobi research group within Zürich Exhalomics project [83].

In total, 14 subjects participated in the study. Each participant was monitored while sleeping; namely, secondary electrospray ionization (SESI) MS analysis of exhaled breath was performed online alongside with PSG recordings. Figure 3.1 contains a schematic of the data acquisition pipeline.

Measurements were taken two times for most subjects: in negative and positive ion modes, i.e. most subjects slept for two nights. Exhaled breath was sampled and analysed every 10 seconds. In positive mode, ion abundances were recorded for, on average, 2,700 equally spaced time points. In the negative mode, the average length of time series is 2,500 points. Those molecules for which intensities were not higher with the breathing mask put on than without were discarded from the further data analysis, because their signals are not reliable. Thus, the resulting dataset consists of ion abundances for 1,271 ions in the positive mode and 725 in negative, synchronised with sleep

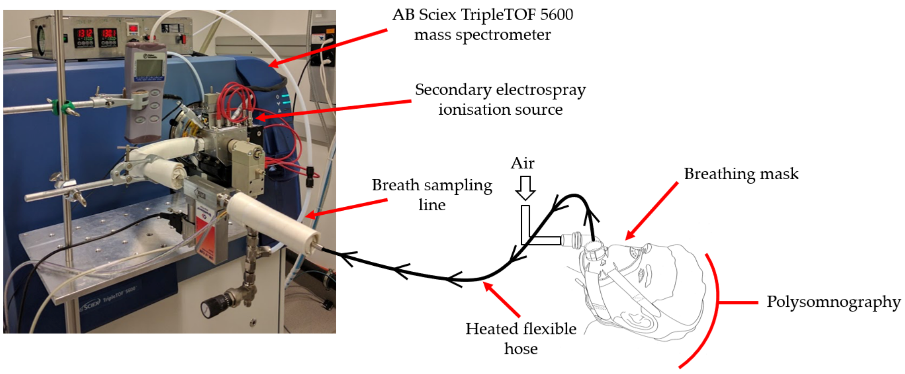


Figure 3.1: A schematic of the data acquisition process. Breath, exhaled by the sleeping subject, is delivered to the ionisation chamber via the heated flexible tube. Subsequently, sampled exhale is analysed by the mass spectrometer. *Image courtesy of Nora Nowak*.

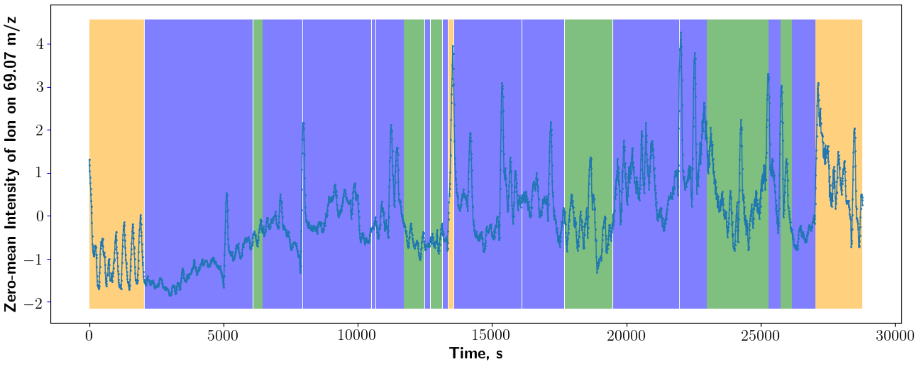


Figure 3.2: Pre-processed relative abundance time series (note, that the time series was re-scaled) for the ion with 69.07 m/z. Synchronised sleep stage labels are indicated by colours: orange segments correspond to wakefulness, NREM stages are shown in blue (N_1, N_2 and N_3), and green segments are aligned with REM sleep.

stage labels. Positive mode data originates from 13 subjects, whereas negative mode data was acquired from 12 subjects. An example of (positive) ion intensity time series alongside with sleep stages is shown in Figure 3.2.

PSG included EOG, ECG, thoracic and abdominal effort, leg movement and pulse oximetry (i.e. blood oxygen saturation) measurements. After obtaining recordings, PSG signals were synchronised with MS time series, and sleep stage scoring was performed manually.

3.2 Pre-processing

Raw mass spectrometry data usually feature unwanted variation that is biologically irrelevant. This variation originates from various sources, such as different weights or volumes of analysed samples [80] or batch effects due to the known structure of the experiment [51]. Procedures that remove technical artefacts are critical in facilitating comparability of sampled mass spectra and time series. These preparation steps can have crucial influence on performance of classification and regression models fitted consequently and on the reproducibility of results. An interesting case study on this issue is provided in [54]. Pre-processing steps performed by us are listed below in the chronological order:

1. Mass spectrum, or sample, normalisation;
2. Smoothing of ion intensity time series;
3. Standardisation of time series.

3.2.1 Sample Normalisation

The general purpose of *sample normalisation* for mass spectrometry data is to remove unwanted variation, which often occurs due to differences in masses and volumes of analysed samples, fluctuations in temperatures and changes in other experimental conditions [62, 51]. Normalisation involves transforming observed raw ion abundances within every mass spectrum to make samples biologically meaningful.

There exist many normalisation techniques, and the choice of an appropriate method needs to be guided by the understanding of underlying assumptions [20]. Herein, we briefly review some of the methods described in the literature. Many of the techniques discussed below are reminiscent of normalisation methods for RNA sequencing (RNA-Seq) data, for example, see [20]. Therefore, we also provide some insights gained from the RNA-Seq literature. Throughout this subsection we assume that N samples of mass spectra with M relative ion abundances are given. For the sake of simplicity, let the intensity of ion j in sample i be denoted by x_{ij}.

Total Ion Count Normalisation

Total ion count (TIC) *normalisation* [15, 62] is a simple and common technique, the idea of which is to normalise every sample w.r.t. the total ion count, the sum of all relative ion abundances within a sample. Values normalised based on TIC are given by

$$\widetilde{x}_{ij} = \frac{x_{ij}}{TIC_i} = \frac{x_{ij}}{\sum_{j=1}^{M} x_{ij}} \tag{3.1}$$

This method, thus, assumes that the total amount of 'expression' across all ions should be the same for all samples [20]. In order to eliminate the influence of outliers on normalising factors TIC_i, ions with highest relative abundances can be ignored in the calculation of the total count [15], thus, yielding a more robust normalisation procedure.

Median Scale Normalisation

Median scale normalisation [62] assumes that sampled mass spectra are similar as much as possible. This method requires choosing reference $\mathbf{x}^* = \begin{bmatrix} x_1^* & \cdots & x_M^* \end{bmatrix}$. It can be, for instance, a randomly selected observed mass spectrum [62]. It is important to note, that the choice of $\mathbf{x}^*$ can strongly affect the outcome of normalisation and, consequently, the results of further analysis. Intensities are normalised based on the reference as follows

$$\widetilde{x}_{ij} = \frac{x_{ij}}{\text{median}_j \left(\frac{x_{ij}}{x_j^*} \right)} \tag{3.2}$$

Thus, the normalising factor is the median of ratios between sample and reference intensities.

Quantile Normalisation

Quantile normalisation [62] enforces identical distributions of ion intensities across all samples. In this procedure, relative abundances within each mass spectrum are ranked and are then mapped to a vector of values in the order given by ranking, i.e. the highest ranking ion is mapped to some value x_1^*, the second highest ranking ion is mapped to x_2^* etc. Each x_j^* is given by the average intensity of ions with rank j. As a result, all corresponding intensity quantiles are the same across all mass spectra.

Internal Standards Normalisation

In some experiments, we might have *a priori* knowledge about concentrations of certain ions or we might be able to insert a compound artificially into each sample. Such control ions with fixed concentrations are referred to as *internal standards* and can be used for sample normalisation [62]. Let us consider having one internal standard – ion with index s. Assuming that abundance x_{is} should be the same across all samples i, normalised intensities are constructed as follows

$$\widetilde{x}_{ij} = \frac{x_{ij}}{x_{is}}, \tag{3.3}$$

alternatively, it is also possible to choose one sample as a reference and normalise w.r.t. the intensity of the standard in the reference sample, as described in [62]. Both versions yield equivalent results.

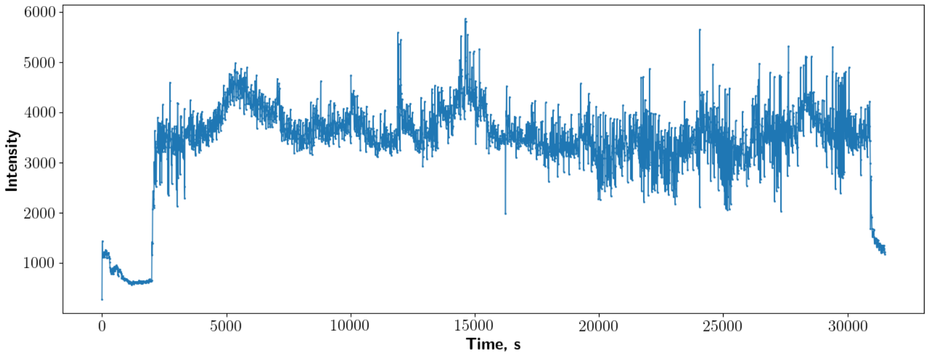


Figure 3.3: Raw intensity time series for the ion with 55.04 m/z (in the positive mode) acquired from one of the subjects. These ions are produced by water vapour, and their relative abundance is expected to stay constant. Nevertheless, the raw time series exhibits some variability. Note, that the end-points of the time series with low intensities correspond to periods when the breathing mask was not attached, therefore, these segments are ignored in normalisation and further analysis.

Generally, this procedure is more robust when it is based on multiple standards [20]. A considerable drawback of this normalisation method is that reference ions might not be available in all experiments, and their insertion into samples may be not affordable.

For the data considered in this thesis, we chose to use internal standards normalisation, because, based on the domain-specific knowledge, we found standards for both, negative and positive, ion modes. Namely, positive mode mass spectra were normalised w.r.t. ions produced by water with the mass-to-charge ratio of 55.04, and negative mode data were normalised to a water-formic acid cluster with 63 m/z. Both of these ions are closely associated with water vapour present in exhaled breath. Since breath is fully saturated with water [8], it is reasonable to assume that the relative abundance of these ions in exhalome should stay constant throughout night. Nevertheless, raw signals for these ions feature a fair amount of variability, Figure 3.3 depicts the raw intensity time series for 55.04 m/z from the positive mode. These fluctuations suggest a need for normalisation.

3.2.2 Time Series Standardisation & Denoising

Next pre-processing step applied to ion intensity time series is *smoothing*. As can be seen from Figure 3.3, ion intensity signals contain high frequency fluctuations. These are, probably, not associated with metabolic changes induced by sleep stages. Therefore, we remove high frequency components by smoothing all time series.

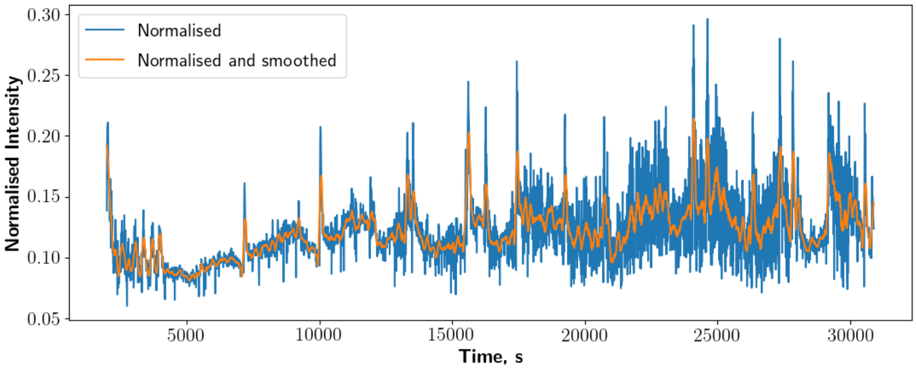


Figure 3.4: Normalised relative abundance time series for ion with 69.07 m/z before and after smoothing, plotted in blue and orange, respectively. Smoothing was performed using Savitzky–Golay filter with the window length of 21 and order 3 polynomials. We used the implementation available in SciPy library [37].

To smooth intensity time series, we apply *Savitzky–Golay* (SG) *filter* [61, 64], a low-pass filter that retains components with lower frequencies. It transforms time series observations by performing least squares polynomial regressions within the range of the moving window of a predefined length. Let us consider degree k polynomial given by $p(n) = \sum_{i=1}^{k} a_i n^i$. During SG smoothing of time series X_t, for each time t, we estimate coefficients a_i by minimising the sum-of-squares $\sum_{l=-m}^{m} (p(l) - x_{t+l})^2 = \sum_{l=-m}^{m} \left(\sum_{i=1}^{k} a_i l^i - x_{t+l} \right)^2$, where $2m+1$ is the width of the filter window [64]. The filtered time series value at time t is given by $\widetilde{x}_t = p(0) = a_0$. This procedure is then repeated for all points in the observed time series by iteratively moving the window. Savitzky and Golay showed that the least squares polynomial smoothing technique described above is equivalent to the discrete convolution of the original signal with a one-dimensional vector of coefficients [64]:

$$\widetilde{x}_t = \sum_{i=-m}^{m} c_i x_{t+i}, \tag{3.4}$$

where c_i are filter's coefficients. Thus, rather than fitting polynomial regressions for every window, coefficients c_i can be computed analytically [61, 64] and used in convolution.

An example of ion abundance time series smoothed with this method is provided in Figure 3.4. The output signal, plotted in orange, is noticeably less noisy than the input, shown in blue. In our pre-processing, we smoothed all normalised intensity time series using the SG filter with the window length of 21 and polynomials of order 3.

Another possible source of unwanted variation within mass spectrometric data are *batch effects* [51], which usually originate from the structure of the experiment. For instance, measurements could be performed with two different devices. As a result, some percentage of variance in the data may be explained by systematic differences between the two groups, or batches. In our case, batch effects are a consequence of biological variation among subjects.

Sample normalisation techniques discussed in the previous subsection do not explicitly account for differences between batches and, thus, usually fail to mitigate them [11]. In this subsection we discuss the *standardisation* [25] of ion abundance time series. The purpose of this transformation is twofold: to prepare variables for regularisation by putting them on comparable scales and to reduce systematic differences between time series from different patients.

Let us consider observing M continuously-valued time series representing normalised ion intensities $\left\{X_t^j\right\}_{t\in\{1,\dots,T\}}$. Standardised observations for ion j are given by

$$\widetilde{x}_t^j = \frac{x_t^j - \bar{x}^j}{\sqrt{\widehat{\sigma}^2\left(x^j\right)}}, \tag{3.5}$$

where $\bar{x}^j = \frac{1}{T}\sum_{i=1}^{T} x_i^j$ is the average of the j-th time series, and $\widehat{\sigma}^2\left(x^j\right) = \frac{1}{T-1}\left(x_i^j - \bar{x}^j\right)^2$ is the standard deviation. As can be seen, $\widetilde{x}_t^j$ are zero-mean and have unit variance for all ions j. Standardisation is performed separately for each replicate of time series, i.e. separately for each subject. Thus, if we assume that subject effects take a form of a fixed shift in all or some of the variables, then re-scaling should be able to remove them.

There exist various departures from the basic formula given in Equation 3.5. For example, it might be sensible to replace the average of the time series with the median and the standard deviation with the interquartile range [51]. This adjustment makes the procedure more robust, since medians and interquartile ranges are less influenced by outliers. Other examples include using various functions of the standard deviation, e.g. the square root or two standard deviations [25, 51]

It is worth mentioning, that the standardisation of mass spectrometric data has been criticised before [51] because of the distributional properties. The procedure is not particularly appropriate for distributions where the mean and the variance are not independent. MS measurements can be roughly described by Poisson process, which is characterised by a dependence between these parameters.

3.3 Exploratory Data Analysis

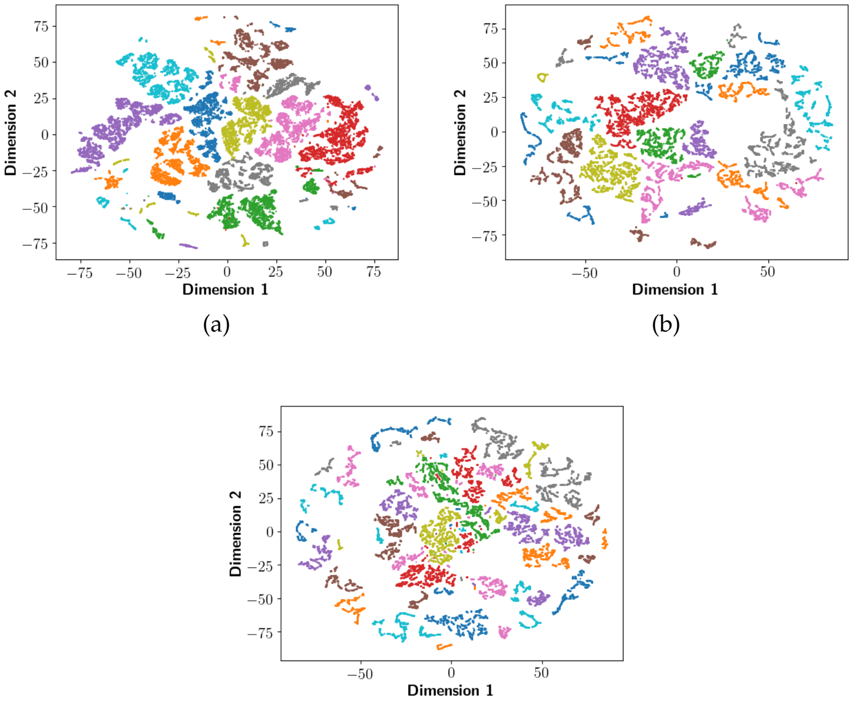
In this subsection we investigate if the MS data contain any systematic structure using dimensionality reduction techniques. In particular, we inspect t-distributed stochastic neighbour embedding (t-SNE) [75] and principal component analysis (PCA) [70] plots. These methods are commonly applied to find non-linear and linear low-dimensional representations for datasets of various origins, while preserving important information, such as clustering, local and global structures. Dimensionality reduction was performed with scikit-learn (v0.21.2) library [58]. Default parameter values were used for t-SNE, in particular, the perplexity was set to 30. Visualisations provided herein completely ignore the time structure by treating data points independently and have merely an exploratory purpose.

3.3.1 Batch Effects

First, we investigate whether raw and pre-processed MS data display any systematic differences between time series acquired from different subjects, i.e. if there are batch effects. Figure 3.5 depicts two-dimensional t-SNE representations of positive mode MS time series obtained from ten subjects (shown in different colours). Note, that plots are provided for raw, normalised and smoothed and fully pre-processed data, see Figures 3.5(a), 3.5(b) and 3.5(c), respectively. t-SNE visualisation of raw data in Figure 3.5(a) clearly shows a clustering that is driven by subjects, since points consistently group by colour. This suggests that batch effects are present in the original data. It appears that after internal standards normalisation and smoothing data points from different subjects are not separated as well anymore (see Figure 3.5(b)). After standardisation, the clustering is even less visible in Figure 3.5(c); as can be seen, points of different colours are intermixed.

Similar observations can be made based on the PCA visualisations of the log-transformed time series provided in Figure A.1 in Appendix A. A substantial amount of variance in the first two principal components could be attributed to batch effects. Note, that the difference between non-standardised and standardised data is even more prominent in these representations than with t-SNE, see figures A.1(b) and A.1(c). We performed the same analysis for the negative ion mode time series. It revealed similar patterns as shown in figures A.2 and A.3 from Appendix A.

In conclusion, two-dimensional t-SNE and PCA visualisations of the MS data suggest that there might be unwanted variability between biological replicates because of differences between subjects. Nevertheless, time series standardisation appears to successfully alleviate batch effects, thus, facilitating meaningful comparison of sleep stages across all subjects.



(a)

(b)

(c)

Figure 3.5: Two-dimensional t-SNE representations of the positive mode MS time series. Each point corresponds to one mass spectrum sampled at some time. Samples are treated completely independently. Figure 3.5(a) was produced from the raw data, 3.5(b) shows data after normalisation and smoothing, finally, 3.5(c) depicts points after all pre-processing steps, including standardisation. Different colours correspond to ten subjects.

3.3.2 Differences between Sleep Stages

Herein, we examine visualisations of the MS data to see if there exist any straightforward systematic differences between mass spectra sampled during different phases of sleep.

To begin with, there is a pronounced imbalance between frequencies of sleep stage labels. In particular, NREM phases (N_1, N_2 and N_3) are more prevalent than wakefulness (W) and REM sleep (R). Tables 3.1 and 3.2 provide frequencies of sleep stage labels alongside with average lengths and frequencies of phases for positive and negative modes, respectively. It is also noteworthy, that positive mode measurements have considerably more time points labelled as wakeful (26.57%) than the negative mode (11.60%). This is because positive mode recordings include two subjects that were awake for unusually long periods of time. For the same reason, the average duration of this phase is two times larger in the positive mode than in negative.

Sleep Stage Label	Label Frequency	Label Percentage, %	Average Duration of Stage, s	Stage Frequency
W	9,523	26.57	935	102
N_1	1,843	5.14	233	79
N_2	14,536	40.56	720	202
N_3	5,980	16.69	564	106
R	3,955	11.04	899	44

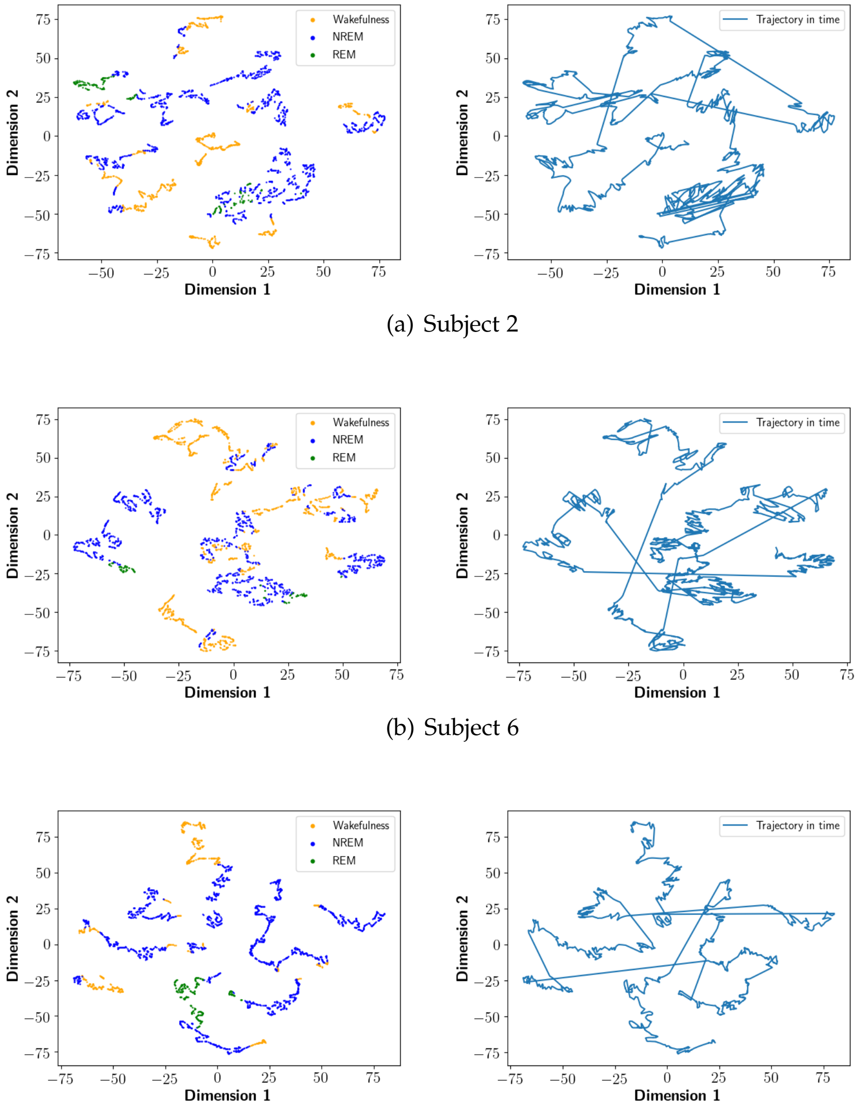
Table 3.1: Distribution of sleep stage labels, stages and their duration in the positive ion mode.

Sleep Stage Label	Label Frequency	Label Percentage, %	Average Duration of Stage, s	Stage Frequency
W	3,480	11.60	529	66
N_1	1,086	3.62	213	51
N_2	13,310	44.37	783	170
N_3	6,652	22.18	739	90
R	5,469	18.23	1,139	48

Table 3.2: Distribution of sleep stage labels, stages and their duration in the negative ion mode.

Figure 3.6 depicts two-dimensional t-SNE representations of pre-processed positive mode MS time series acquired from three different subjects. Note, that points are coloured according to synchronised sleep stage labels. t-SNE representations in these plots do not feature a consistent clustering driven by sleep phases. The same holds true for the rest of subjects that are not considered here. Moreover, negative mode data do not appear to feature systematic differences between sleep stages either, see Figure A.4 in Appendix A. Similar conclusions can be made based on the PCA. First two principal components, shown in figures A.5 and A.6 in Appendix A, do not reveal clear differences between mass spectra during wakefulness, NREM and REM phases.

In addition, we inspect the ordering of data points w.r.t. time. Clusters observed in t-SNE in figures 3.6 and A.4 could be attributed to the time structure. Observe that chronologically consecutive or closely neighbouring points are usually grouped together. This pattern is revealed when connecting consecutive points with line segments, as in the plots on the right side of Figure 3.6. This structure may be associated with a trend component that is present in the time series of many ions. For example, the time series in Figure 3.2 contains a mild trend that is visible in the beginning. Potentially, this may introduce biases when performing sleep stage classification based on ion intensities. It might be easier to correctly classify points from some stage



(a) Subject 2

(b) Subject 6

(c) Subject 10

Figure 3.6: Two-dimensional t-SNE representations of the MS time series in the positive ion mode for three subjects. In plots on the left side, points are coloured according to their sleep stage labels: orange, blue and green colours correspond to wakefulness, NREM and REM phases, respectively. Plots on the right side contain the same t-SNE representations, however, points that are consecutive w.r.t. time are connected by line segments.

due to their contiguity in time, rather than systematic differences between classes.

To sum up, t-SNE and PCA visualisations of MS data show no consistent clustering of data points that is driven by sleep stages. However, t-SNE reveals a grouping of points w.r.t. the time of their acquisition. This might be explained by trend components of ion intensity time series and could bias predictions of sleep stage labels based on mass spectra.

Chapter 4

Inferring Granger Causality with Neural Networks

To tackle the main problem of the thesis, we adopt the approach of Granger causality (see Subsection 2.3.1). As a result, one of the methodological contributions of this thesis is the development of a model for the estimation of multivariate Granger causality based on neural networks (see Section 2.4). Its key advantages are that (i) it can capture nonlinear non-additive interactions; (ii) it provides a principled way of dealing with 'mixed' time series which contain both categorically- and continuously-valued variables; and that (iii) it is scalable. In this chapter, we review related work, introduce our approach and show how we can quantify uncertainty about causal relationships within the proposed model.

Throughout this chapter we consider a multivariate time series consisting of target $\{Y_t\}_{t\in\{1,...,T\}}$ and predictor variables $\left\{X_t^j\right\}_{t\in\{1,...,T\}}$, for $j = 1, ..., p-1$. Herein all time series can be either categorically- or continuously-valued. The problem that we want to tackle is to find the set of all predictors which Granger-cause the response, i.e. identify $S_{in} = \left\{j : X^j \longrightarrow Y\right\}$.

4.1 Related Work

Many 'real world' time series exhibit nonlinear relationships, e.g. in genomics and neuroscience [73]. Therefore, a plenty of nonlinear techniques for identifying Granger-causal interactions were proposed [2, 48, 69, 53, 24, 55, 77, 73], to avoid inconsistencies resulting from model misspecification. Researchers have argued in favour of using neural networks due to their flexibility when compared to other conventional model-based approaches [53]. For example, some nonlinear methods [59, 69] employ generalised additive models (GAM) [32] and, thus, assume the absence of between-variable non-additive interaction terms. In contrast to GAMs, approaches based on neural networks do not make such rigid assumptions [53, 77, 73]. Given the success of neural networks at various predictive tasks, this class of models is a compelling choice for flexibly representing autoregressive relationships in multivariate time series.

4.1.1 Time Series Forecasting with MLP

In order to infer Granger causes of target time series Y_t, techniques described in the literature [53, 77, 73] fit some neural network model to forecast the future of Y based on the past values of X^j. A simple two-layer feedforward network with the continuously-valued output is given by the following equation [21]:

$$\widehat{y}_t = h^o \left\{ w_{10}^{(2)} + \sum_{i=1}^{M^{(1)}} w_{1i}^{(2)} h^{(1)} \left\{ w_{i0}^{(1)} + \sum_{j=1}^{p-1} \sum_{k=1}^{K} w_{i((j-1)K+k)}^{(1)} x_{t-k}^{j} + \sum_{k=1}^{K} w_{i((p-1)K+k)}^{(1)} y_{t-k} \right\} \right\}, \tag{4.1}$$

where $M^{(1)}$ is the size of the hidden layer; and $h^{(1)}$ and h^o are activation functions at hidden and output units, respectively. Note, that forecast $\widehat{y}_t$ is constructed from K past values of all predictors X^j and of the target. Thus, this MLP has pK inputs. To infer Granger causality, we need to identify those predictors that are useful in constructing the forecast. This problem has been addressed in various ways in the literature.

4.1.2 Neural Networks with Non-uniform Embedding

Montalto et al. proposed neural networks with non-uniform embedding (NUE) [53]. In this approach, significant Granger causes are identified using the NUE, a feature selection procedure. Initially, the multilayer perceptron for predicting the future of the target time series contains only one input. It is then iteratively grown by greedily adding lagged predictor components as inputs. The best possible addition is made only if it improves the performance of the model according to the criteria specified in [53]. Once stopping conditions are satisfied, a predictor time series is claimed a significant cause of the target if at least one of its lagged components was added as an input. An important advantage of this technique is that, alongside with causes, it identifies lags at which causal interactions occur. Even though the procedure employs the warm-start approach and resilient back-propagation for training neural networks [53], it is computationally expensive, especially, in a high-dimensional setting, since it requires fitting and comparing many candidate models.

Wang et al. extended the method proposed in [53] by replacing MLPs with long short-term memory networks [77], while still using the NUE for selecting significant causes. As opposed to MLPs, LSTMs do not require specifying the model order, i.e. the maximum lag at which causal interactions occur, because, as explained in Subsection 2.4.2, recurrent neural networks store information about the past of the input sequence in hidden states. Moreover, due to parameter sharing, fitting LSTMs requires estimating much less parameters than for multilayer perceptrons. The approach based on recurrent neural networks was shown to be superior to MLPs in terms of computational cost and performance [77]. Nevertheless, it still involves iterative fitting of prohibitively many augmented models.

4.1.3 Deep Feature Selection

A noteworthy feature selection procedure for feedforward neural networks is deep feature selection (DFS) proposed by Li et al. in [42]. Even though it was originally considered in the context of classification, the DFS can be useful for identifying Granger causes. It is achieved by introducing a weight for each feature [42], which is applied to the corresponding input prior to feeding it into the MLP. The sparsity of these weights is encouraged by adding a penalty term to the loss function. Similarly to the elastic net [82], the penalty consists of L_1 and L_2 norms of the weight vector. Thus, shrinkage of weights towards zero results in selecting some features, while 'knocking out' other.

4.1.4 Componentwise MLPs and LSTMs

Tank et al. proposed a method for estimating Granger causality with MLPs and RNNs, which utilises regularisation in a manner similar to the deep feature selection [73]. Before we explain this approach, we need to introduce some additional notation for the MLP given by Equation 4.1. Observe that this equation can be rewritten in the following matrix form:

$$\widehat{y}_t = h^o \left\{ w_{10}^{(2)} + \sum_{i=1}^{M^{(1)}} w_{1i}^{(2)} \left(h^{(1)} \left\{ \mathbf{w}_0^{(1)} + \sum_{k=1}^{K} \mathbf{W}_k^{(1)} \mathbf{x}_{t-k} \right\} \right)_i \right\}, \tag{4.2}$$

where $\mathbf{x}_{t-k} = \begin{bmatrix} x_{t-k}^1 & x_{t-k}^2 & \cdots & x_{t-k}^{p-1} & y_{t-k} \end{bmatrix}^\top$; $\mathbf{W}_k^{(1)} \in \mathbb{R}^{M^{(1)} \times p}$ is a matrix with the first layer weights corresponding to time series values at lag k; and $\mathbf{w}_0^{(1)} = \begin{bmatrix} w_{10}^{(1)} & w_{20}^{(1)} & \cdots & w_{M^{(1)}0}^{(1)} \end{bmatrix}^\top$ is a vector consisting of biases from the first layer. Note, that herein activation function $h_1(\cdot)$ is applied elementwise to the vector of hidden layer activations.

The intuition behind the method discussed in [73] is that, by introducing the group Lasso penalty on the first layer, weights of non-causal predictors should be shrunk towards zero. The loss function [73], for a continuous response, is given by

$$\sum_{t=K+1}^{T}\left(y_t-\widehat{y}_t\right)^2+\lambda\sum_{j=1}^{p}\left\|\left[\left(\mathbf{W}_1^{(1)}\right)_{:j}\quad\left(\mathbf{W}_2^{(1)}\right)_{:j}\quad\cdots\quad\left(\mathbf{W}_K^{(1)}\right)_{:j}\right]\right\|_F, \tag{4.3}$$

where $\left(\mathbf{W}_k^{(1)}\right)_{:j}$ denotes the j-th column of matrix $\mathbf{W}_k^{(1)}$; and

$$\|\mathbf{A}\|_F=\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}\left|(\mathbf{A})_{ij}\right|^2}$$

is the Frobenius norm of matrix $\mathbf{A}\in\mathbb{R}^{n\times m}$. As can be seen, the penalty term groups the weights of the first layer according to a predictor time series they correspond to. Tank et al. refer to this specialised model as a *componentwise* multilayer perceptron (cMLP). In addition, the authors proposed a componentwise long short-term memory network (cLSTM) that, similarly to cMLP, encourages sparse inputs, but has a recurrent network architecture [73].

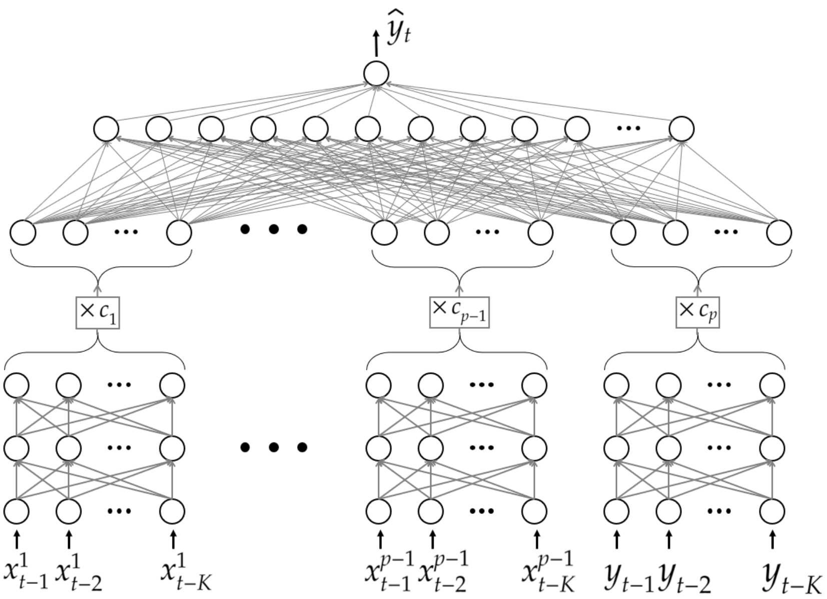
An important advantage of these componentwise networks over the non-uniform embedding, discussed in the previous subsection, is their scalability. Rather than iteratively fitting and comparing augmented models, this method needs to fit only one regularised model to infer all causes of the target series. Therefore, this technique can be particularly helpful when performing causal analysis of high-dimensional time series.

4.2 Model

Inspired by componentwise MLPs [73], we propose another feedforward neural network architecture for identifying Granger causes of the target time series. For the sake of convenience, we will henceforth refer to it as *Granger causal multilayer perceptron* (GC-MLP). We utilise Lasso regularisation to avoid costly comparison between augmented models. In contrast to cMLPs, sparsity is not encouraged in the first layer, but in the middle of the network.

We define the model assuming that response Y and all predictors X^j are continuously-valued. However, this definition can be easily extended to the 'mixed' case, as we will subsequently show. Figure 4.1 depicts the schematic of a GC-MLP with a continuously-valued output. As can be seen, it is a feedforward network consisting of separate sub-networks, one per each predictor, which are then fused together to compute the final output.

The model forecasts value y_t based on K past values of time series $X_t^1, ..., X_t^{p-1}$ and Y_t itself. Herein, we will explain how this forecast is constructed. Let



$\widehat{y}_t$
$\times c_1$
$\times c_{p-1}$
$\times c_p$
x^1_{t-1} x^1_{t-2} x^1_{t-K}
x^{p-1}_{t-1} x^{p-1}_{t-2} x^{p-1}_{t-K}
y_{t-1} y_{t-2} y_{t-K}

Figure 4.1: The schematic of a feedforward neural network for identifying Granger causes of continuously-valued time series Y_t among continuously-valued $X^1_t, X^2_t, ..., X^{p-1}_t$. It consists of sub-networks per each predictor which take lagged time series values as inputs. Subsequently, multidimensional outputs of these sub-networks are weighted, concatenated and fed into another MLP to compute final forecast $\widehat{y}_t$. Note, that biases are omitted.

$\mathbf{x}^j = \begin{bmatrix} x^j_{t-1} & x^j_{t-2} & \cdots & x^j_{t-K} \end{bmatrix}^\top$ be a vector containing K past values of time series X^j_t, for $j = 1, ..., p-1$, likewise, let $\mathbf{y} = \begin{bmatrix} y_{t-1} & y_{t-2} & \cdots & y_{t-K} \end{bmatrix}^\top$. Each $\mathbf{x}^j$ and $\mathbf{y}$ are fed as inputs into sub-networks to compute multidimensional output vectors

$$\mathbf{v}^j = h^{(2)}\left(\mathbf{w}_0^{(2)j} + \mathbf{W}_j^{(2)} h^{(1)}\left(\mathbf{w}_0^{(1)j} + \mathbf{W}_j^{(1)}\mathbf{x}^j\right)\right), \text{ for } j = 1, .., p-1, \tag{4.4}$$

and

$$\mathbf{v}^p = h^{(2)}\left(\mathbf{w}_0^{(2)p} + \mathbf{W}_p^{(2)} h^{(1)}\left(\mathbf{w}_0^{(1)p} + \mathbf{W}_p^{(1)}\mathbf{y}\right)\right), \tag{4.5}$$

where $\mathbf{W}_j^{(1)} \in \mathbb{R}^{M^{(1)}\times K}$ and $\mathbf{W}_j^{(2)} \in \mathbb{R}^{M^{(2)}\times M^{(1)}}$ are matrices with weights of layers 1 and 2, respectively, in the j-th sub-network; $\mathbf{w}_0^{(1)j} \in \mathbb{R}^{M^{(1)}}$ and $\mathbf{w}_0^{(2)j} \in \mathbb{R}^{M^{(2)}}$ are vectors with biases of layers 1 and 2, respectively; and $h^{(1)}(\cdot)$ and $h^{(2)}(\cdot)$ are activation functions that are applied elementwise. Note, that all sub-networks have the same sizes of layers. Subsequently, output vectors $\mathbf{v}^j$ are weighted and concatenated to form one large vector with $pM^{(2)}$ components

$$\mathbf{v} = \begin{bmatrix} c_1\mathbf{v}^1 \\ c_2\mathbf{v}^2 \\ \vdots \\ c_p\mathbf{v}^p \end{bmatrix} \tag{4.6}$$

Note, that $\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \cdots & c_p \end{bmatrix}^\top$ are weights assigned to each variable. Vector $\mathbf{v}$ is then used as an input for the MLP that computes the final forecast

$$\widehat{y}_t = h^o\left(w_{10}^{(4)} + \mathbf{w}^{(4)}h^{(3)}\left(\mathbf{w}_0^{(3)} + \mathbf{W}^{(3)}\mathbf{v}\right)\right), \tag{4.7}$$

where $\mathbf{W}^{(3)} \in \mathbb{R}^{M^{(3)} \times pM^{(2)}}$ is a matrix with weights of layer 3 and $\mathbf{w}^{(4)} \in \mathbb{R}^{1 \times M^{(3)}}$ is a row vector with weights of layer 4; $\mathbf{w}_0^{(3)} \in \mathbb{R}^{M^{(3)}}$ is a vector with biases of layer 3 and $w_{10}^{(4)}$ is the bias of the last layer. When forecasting a continuously-valued time series, we use the identity function $h^o(x) = x$ at the output.

The loss function of GC-MLP is crucial for estimating Granger causality. In particular, it encourages weights $\mathbf{c}$ to be sparse by using an elastic-net-style penalty term [82], like in the deep feature selection [42]. Let $\boldsymbol{\theta}$ denote an ordered set of all parameters. Then, for continuously-valued targets, fitting a GC-MLP requires solving the following optimisation problem:

$$\min_{\boldsymbol{\theta}} \sum_{t=K+1}^{T} (y_t - \widehat{y}_t)^2 + \lambda\left(\alpha\|\mathbf{c}\|_1 + (1-\alpha)\|\mathbf{c}\|_2^2\right), \tag{4.8}$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the trade-off between L_1 and L_2 penalties. Intuitively, we expect that, if time series X_t^j is non-causal, then weight c_j will be shrunk towards zero. In the following chapter (see Chapter 5), we perform several controlled experiments to demonstrate that the neural network behaves as expected in that regard.

4.2.1 Including Categorically-valued Time Series

Now let us consider the case wherein target series Y_t is categorically-valued and takes on values in $\{1, 2, ..., C\}$. This series can be represented using *one-hot encoding*. In particular, rather than having one sub-network for Y_t, as before, we introduce a sub-network for each $Y_t^j = \mathbb{1}_{\{Y_t=j\}}$, for $j = 1, ..., C$, which are treated as separate variables. Another important adjustment is made to the output. To predict probabilities of different classes, the network should have C outputs, and the output activation function should be softmax (see Equation 2.10). Finally, the loss function is changed as well, we use the cross-entropy loss instead of the sum-of-squares:

$$-\sum_{t=K+1}^{T}\sum_{j=1}^{C}\gamma_j\left\{y_t^j\ln\left((\widehat{\mathbf{y}}_t)_j\right)+\left(1-y_t^j\right)\ln\left(1-(\widehat{\mathbf{y}}_t)_j\right)\right\} \\ +\lambda\left(\alpha\|\mathbf{c}\|_1+(1-\alpha)\|\mathbf{c}\|_2^2\right), \tag{4.9}$$

where γ_j is the weight for class j. Weighted loss can be useful when there exists an imbalance between class frequencies [43]. Namely, we can encourage higher sensitivities for infrequent categories by assigning larger weights to them.

Another possible case is when one or several predictors X_t^j are categorically-valued. In this setting, we can use one-hot encoding to represent such predictor time series, in the same way as was explained above for representing a categorically-valued target. Finally, we can also account for having both categorically-valued predictors and the target by implementing all adjustments described in this subsection.

4.2.2 Important Hyperparameters

As can be seen from the description above, the GC-MLPs have several hyperparameters. Some of them might strongly affect the performance of the model at causal inference, namely:

- *Regularisation parameter* λ controls the sparsity of weight vector $\mathbf{c}$ and, thus, intuitively, should be associated with the number of falsely discovered causes. When λ is chosen too large, we expect less false discoveries to be made at the cost of a lower power. On the other hand, with λ too small, we expect the method to make many false discoveries, but have a high power. We investigate the relationship between λ and false discoveries in one of the experiments in Chapter 6.
- *Model order* K determines the maximum lag at which dependencies are considered. Choosing K too small can result in model misspecification, because long-term regressive dependencies might not be captured by the model. However, choosing K too large may lead to overfitting and, consequently, inferring spurious causal relationships.

4.2.3 Implementation Details

To estimate the parameter of interest – weight vector $\mathbf{c}$, we need to minimise the penalised loss function (see Equations 4.8 and 4.9). This optimisation problem is addressed by performing gradient descent, either stochastic or mini-batch. In particular, we employ the implementation of *Adam optimiser* [39] by PyTorch machine learning library [56]. A disadvantage of gradient descent procedures is that obtained parameter estimates do not converge to exact zeros. In practice, when identifying causality, variable weights have to be thresholded. An important hyperparameter of the optimiser is the *learning rate* which determines the size of steps made during optimisation. This parameter alongside with the number of passes through training data, or the number of *epochs*, is crucial for the quality of causal inference.

We implemented the model in Python programming language (version 3.7.1) using PyTorch machine learning library (version 1.0.1) [56]. Code for the custom PyTorch Module that constructs a GC-MLP is provided in Appendix B in Listing B.1. Observe that in this implementation layers 1 and 2 have equal sizes in all sub-networks, and that all hidden units have ReLU activation functions. In order to mitigate overfitting, we allow the application of *dropout* [72] in all layers. This technique prevents co-adaptation of units in neural networks by zeroing out randomly chosen connections.

Last but not least, note, that weight vector **c** is initialised differently from the rest of weights, for which Xavier method is used [27]. Each c_j is drawn independently from $\mathcal{N}\left(\frac{1}{p}, \sigma_c^2\right)$, where p is the number of variables, including the response, and variance σ_c^2 is chosen to be very small. During experimentation, we observed that such initialisation scheme leads to less variability in the results due to differences at the start and faster convergence to the correct causal structure.

4.2.4 LSTMs

Like componentwise MLPs [73], our network architecture can be easily adjusted to leverage long short-term memory networks for inferring Granger causality. Feedforward sub-networks for each predictor variable can be replaced by LSTMs. We will henceforth refer to this architecture as *Granger causal long short-term memory* (GC-LSTM).

A simple LSTM network with input time series value x_t^j is given by equations [73]:

$$\begin{aligned}
\mathbf{f}_t^j &= \sigma\left(\mathbf{w}_j^f x_t^j + \mathbf{U}_j^f \mathbf{h}_{t-1}^j\right), \\
\mathbf{i}_t^j &= \sigma\left(\mathbf{w}_j^{in} x_t^j + \mathbf{U}_j^{in} \mathbf{h}_{t-1}^j\right), \\
\mathbf{o}_t^j &= \sigma\left(\mathbf{w}_t^o x_t^j + \mathbf{U}_j^o \mathbf{h}_{t-1}^j\right), \\
\mathbf{c}_t^j &= \mathbf{f}_t^j \odot \mathbf{c}_{t-1}^j + \mathbf{i}_t^j \odot \sigma\left(\mathbf{w}_j^c x_t^j + \mathbf{U}_j^c \mathbf{h}_{t-1}^j\right), \\
\mathbf{h}_t^j &= \mathbf{o}_t^j \odot \sigma\left(\mathbf{c}_t^j\right),
\end{aligned} \tag{4.10}$$

where $\mathbf{f}_t^j, \mathbf{i}_t^j, \mathbf{o}_t^j$ are forget, input and output gates, respectively; $\mathbf{c}_t^j$ is the cell state; $\mathbf{h}_t^j$ is the hidden state; $\sigma(\cdot)$ is an activation function applied elementwise; and $\odot$ is the elementwise multiplication operator. Note, that we omitted biases to simplify notation. To infer Granger causality, such LSTM is constructed for each predictor time series and the target, i.e. for $j = 1, ..., p$. Subsequently, we apply variable weights to hidden states of all LSTM subnetworks and concatenate them into one vector:

$$\mathbf{v}_t = \begin{bmatrix} c_1\mathbf{h}_t^1 \\ c_2\mathbf{h}_t^2 \\ \vdots \\ c_p\mathbf{h}_t^p \end{bmatrix} \tag{4.11}$$

Finally, vector $\mathbf{v}_t$ is fed into the two-layer perceptron to compute the forecast for y_{t+1} (similarly to Equation 4.7). When training GC-LSTMs, the same penalised objective is used as for the feedforward architecture, see equations 4.8 and 4.9. The implementation of the GC-LSTM model is provided in Listing B.2 in Appendix B. We do not discuss this approach in further chapters and only focus on MLPs.

4.3 Quantifying Uncertainty

As a result of fitting a GC-MLP (or GC-LSTM), we get one point estimate of weight c_j for each predictor variable. If the absolute value of this weight is significantly larger than 0, then we expect X^j to Granger-cause Y. Nevertheless, it would be desirable to obtain a measure of uncertainty about the statement of causality. It is especially important in the presence of biological variability due to possible inter-subject differences. In this section, we explain a procedure that uses independent replicates of time series to construct lower confidence bounds on variable weights in $\mathbf{c}$. It is based on the *bootstrapping* method proposed by B. Efron in [18].

Bootstrapping is a non-parametric technique for estimating the sampling distribution of an arbitrary statistic based on the observed data [18]. The distribution is estimated be evaluating the statistic on data points re-sampled with replacement from the original dataset. Subsequently, *confidence intervals* (CI) can be constructed based on the bootstrapped values. This powerful method is applicable to parameters of regression [22] and classification models and, thus, can be leveraged to estimate the distribution of weights c_j in GC-MLPs. The technique explained further is essentially an adaptation *random-x re-sampling* for regression, as described in [22].

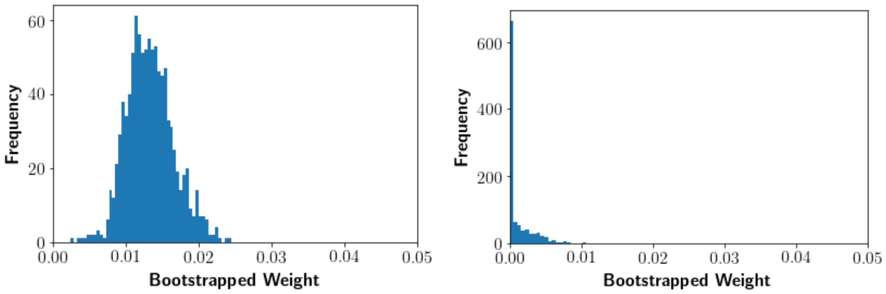
Algorithm 1 contains the pseudocode of the procedure for discovering Granger causes of the given target series. The neural network model is trained B times on re-sampled data. It is important to note, that rather than re-sampling individual data points, we re-sample entire time series replicates. This adjustment is made to the classical method presented in [22], because points within one time series are, possibly, not independent. Moreover, we

Algorithm 1: Bootstrapping procedure for discovering Granger causality.

```
Input: $N$ replicates of target time series $\{Y_t\}_{t\in\{1,...,T\}}$ and predictors
       $\left\{X_t^j\right\}_{t\in\{1,...,T\}}$, $j = 1,..,p-1$; regularisation parameter $\lambda$;
       threshold $c_{th} > 0$; significance level $\alpha \in (0,1)$; number of
       re-samples $B \in \mathbb{N}$.

Output: Set $\widehat{S}_{in}^{\lambda}$ of predictor variables that Granger-cause $Y$.

$\widehat{S}_{in}^{\lambda} \leftarrow \{\}$
// Compute bootstrapped weights
for $b = 1$ to $B$ do
    Sample $N$ replicates $I^b = \{i_1^b, ..., i_N^b\}$ with replacement from
      $I = \{1, ..., N\}$.
    Train the neural network on time series replicates in $I^b$ with
      regularization parameter $\lambda$.
    Retrieve absolute values of weights $c_1^{*b}, ..., c_{p-1}^{*b}$ in $\mathbf{c}$ from the fitted
      model.
end
// Choose causal variables
for $j = 1$ to $p-1$ do
    Compute empirical $\alpha$-quantile of bootstrapped weights for the $j$-th
      variable $q_j := q_{c_j^*}(\alpha)$
    if $q_j \geq c_{th}$ then
        $\widehat{S}_{in}^{\lambda} \leftarrow \widehat{S}_{in}^{\lambda} \cup \{j\}$
end

return $\widehat{S}_{in}^{\lambda}$
```

expect that this re-sampling scheme would allow accounting for biological variability between replicates and help to discover those causal relationships that stay invariant across all subjects.

After B neural networks are fitted and absolute values of bootstrapped variable weights are retrieved, we construct a lower confidence bound for each weight. This bound is given by the left end-point of the $100(1-2\alpha)\%$ bootstrap quantile CI [35] (we use twice the significance level, because we are only interested in the left end-point). This confidence interval is built from empirical quantiles of the bootstrapped statistic [35]. There exist other bootstrap CIs which are, in certain aspects, superior to the quantile interval. However, as opposed to several other methods, quantile CI is appropriate for skewed distributions. Since we consider absolute values of weights, their



(a) Histogram of bootstrapped weights for the ion with 69.07 m/z.

(b) Histogram of bootstrapped weights for the ion with 194.13 m/z.

Figure 4.2: Histograms of $B = 1000$ bootstrapped absolute values of weights for two ions used as predictors for the sleep stage time series. Observe that both histograms are right-skewed. Also note, that in Figure 4.2(a) most values are significantly greater-than zero, whereas in Figure 4.2(b) most weights are clearly shrunk towards zero. It appears that the ion with 69.07 m/z could be causal, whereas the ion with 194.13 m/z is not.

distributions could be skewed. For example, Figure 4.2 depicts histograms of bootstrapped absolute weights for two different ions that were used as predictors for the sleep stage. Note, that the histogram in Figure 4.2(b) is strongly right-skewed.

Finally, causal predictors are chosen based on the constructed lower bounds. A variable is claimed to be a Granger cause of the target if the lower bound for its absolute weight is greater-than-or-equal-to specified threshold c_{th}. When selecting causal variables in a high-dimensional setting, the multiple testing problem occurs [33]. That can lead to the inflation in the number of false discoveries. Therefore, parameters λ, α and c_{th} need to be chosen carefully to ensure an acceptable number of errors.

A significant limitation of the presented procedure is the assumption of having multiple. In many cases, we observe a multivariate time series only once, and, therefore, cannot apply the bootstrap method directly. A pragmatic solution to this problem would be to split the sequence into several (equally long) segments and treat them as independent replicates. This approach is clearly not ideal, because sub-sequences are not fully independent. However, if the observed time series is sufficiently long, this 'trick' appears to be sensible.

4.4 Discovering Effects by Reversing Time

So far, we have treated the problem of discovering a set of predictor time series that cause the given target. We might be interested in considering an inverse problem: given observations of target time series $\{Y_t\}_{t\in\{1,...T\}}$

and predictors $\left\{X_t^j\right\}_{t\in\{1,...,T\}}$, for $j = 1, ..., p-1$, we seek to find set $S_{out} = \left\{j : Y \longrightarrow X^j\right\}$ of predictors that are Granger-caused by Y.

The naïve solution to the problem is straightforward. We have to fit p GC-MLPs for each time series X_t^j as a response and identify if Y_t Granger-causes X_t^j. In the high-dimensional multivariate time series, this approach can be prohibitively costly. A promising solution is to consider time-reversed sequences instead.

Let $\widetilde{Y}_t$ and $\widetilde{X}_t^j$ denote time reverses of series Y_t and X_t^j, respectively. In order to estimate S_{out}, we suggest to train a GC-MLP with $\widetilde{Y}_t$ as a response and $\widetilde{X}_t^j$ as predictors. Intuitively, we expect this network to discover if the future values of X_t^j are useful for predicting the past values of Y_t. We expect that variable weights are shrunk towards zeros for those $\widetilde{X}_t^j$ such that $Y \not\longrightarrow X^j$. Thus, instead of naïvely fitting p models, we potentially need only one GC-MLP to estimate the set of Granger effects.

To our knowledge, there is little literature on the topic of time-reversed Granger causality (TRGC) and its validity. In [79], the authors prove the validity of certain tests for TRGC in a bivariate case for the vector autoregressive model. The time reversal technique in the multivariate case is merely a 'trick' without known theoretical guarantees. We investigate the empirical performance of this method on synthetic data in the next chapter in Section 5.7.

Chapter 5

Simulation Experiments

In this chapter we describe experiments that were performed on artificial datasets. Since the model described in Chapter 4 has no theoretical guarantees for correctly discovering Granger causal relationships, it is of utmost importance to test its performance in a controlled setup. We consider several examples of multivariate time series with varying degrees of complexity of regressive relationships.

Additionally, we compare our neural network technique to linear Granger causality inference with the VAR model, which serves as a baseline method. We use the implementation of the VAR and F-tests for Granger causality available in statsmodels (version 0.10.0) library [66] for Python. To mitigate the multiple comparisons issue and control the false discovery rate (FDR), we apply Benjamini-Hochberg procedure [4] to F-test p-values.

We use areas under the *receiver operating characteristic curve* (AUROC) and the *precision-recall curve* (AUPR) [13] to assess the performance of methods. The latter measure could be more appropriate for causal structure learning algorithms, because the underlying causal graphs are often quite sparse, and AUPR is fairer than AUROC in classification problems with a class imbalance [13]. These metrics were chosen over, for instance, accuracy, because the GC-MLP model evaluates causality via variable weights given by $\mathbf{c}$ that are numerical, rather than binary. For the VAR model, we use F-test p-values for computing AUROC and AUPR.

5.1 Linear Autoregressive Model

In this experiment, we consider a trivariate time series with linear autoregressive relationships and additive Gaussian noise. We generate 100 synthetic datasets with one replicate of the time series 500 steps long. We repeat the experiment for sequences 1,000 and 5,000 points long. Time series are

Method	**Average AUROC($\pm$2SD)**		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	1.000($\pm$0.000)	1.000($\pm$0.000)	1.000($\pm$0.000)
GC-MLP	0.956($\pm$0.179)	0.983($\pm$0.111)	0.998($\pm$0.044)

Table 5.1: Average AUROCs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the linear autoregressive model given by equations 5.1.

Method	**Average AUPR($\pm$2SD)**		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	1.000($\pm$0.000)	1.000($\pm$0.000)	1.000($\pm$0.000)
GC-MLP	0.972($\pm$0.105)	0.989($\pm$0.066)	0.999($\pm$0.026)

Table 5.2: Average AUPRs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the linear autoregressive model given by equations 5.1.

drawn independently from the autoregressive model given by the following equations:

$$\begin{aligned} X_t &= a_1 X_{t-1} + N_{X,t}, \\ Y_t &= a_2 Y_{t-1} + a_3 X_{t-2} + a_4 W_{t-1} + N_{Y,t}, \\ W_t &= a_5 W_{t-1} + a_6 X_{t-1} + N_{W,t}, \end{aligned} \tag{5.1}$$

where $a_i \sim \mathcal{U}\left([-0.8, -0.2] \cup [0.2, 0.8]\right)$ are coefficients sampled independently for each dataset; and $N_{\cdot,t} \sim \mathcal{N}(0, 1)$ are innovation terms. Note, that all causal interactions occur at the lag of at most 2. The correct causal summary graph is given by edges $\{X \longrightarrow Y, X \longrightarrow W, W \longrightarrow Y\}$. GC-MLPs that we fitted had 30 units in layers 1 and 2 (in each sub-network), and 30 units in layer 3. We chose $\lambda = 0.1$, $\alpha = 0.8$ and $K = 5$. The network was trained for one epoch using the SGD with the learning rate of 0.001. VAR models were fitted with the maximum lag of 5.

Tables 5.1 and 5.2 contain average AUROCs and AUPRs, respectively, for VAR and GC-MLP models. VAR is superior to GC-MLP in terms of both ROC and PR curves. It infers the correct causal structure in all 100 synthesised datasets for all time series lengths. Differences in performance measures for $T = 500, 1000$ are statistically significant (paired t-test $p < 0.05$). Nevertheless, the magnitude of differences is not particularly large. In general, these results are not unexpected, because in this experiment the VAR model is appropriate due to the linearity of causal relationships. Given that the VAR requires estimating less parameters and there is no model misspecification, it is sensible that it outperforms neural networks on limited training data. We can also see that the performance of the GC-MLP improves slightly when given longer sequences.

Method	Average AUROC($\pm$2SD)		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	0.550($\pm$0.704)	0.500($\pm$0.725)	0.545($\pm$0.726)
GC-MLP	0.920($\pm$0.545)	1.000($\pm$0.000)	1.000($\pm$0.000)

Table 5.3: Average AUROCs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the nonlinear autoregressive model given by equations 5.2.

Method	Average AUPR($\pm$2SD)		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	0.650($\pm$0.461)	0.630($\pm$0.441)	0.655($\pm$0.465)
GC-MLP	0.960($\pm$0.273)	1.000($\pm$0.000)	1.000($\pm$0.000)

Table 5.4: Average AUPRs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the nonlinear autoregressive model given by equations 5.2.

5.2 Nonlinear Autoregressive Model

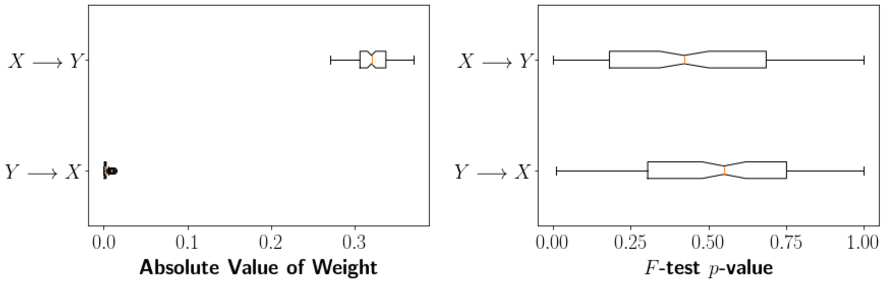
This experiment demonstrates that, in general, model misspecification can lead to inferring an incorrect time series causal structure. We synthesise 100 datasets with sequence lengths of 500, 1,000 and 5,000 from the following bivariate autoregressive model due to [59]:

$$\begin{aligned} X_t &= -0.5X_{t-1} + 0.4N_{X,t}, \\ Y_t &= -0.5Y_{t-1} + (X_{t-1})^2 + 0.4N_{Y,t}, \end{aligned} \tag{5.2}$$

where $N_{\cdot,t} \sim \mathcal{N}(0,1)$ are innovation terms. The summary causal DAG has one edge $X \longrightarrow Y$. We use the same parameters for performing causal time series analysis as in Section 5.1.

Tables 5.3 and 5.4 provide average AUROCs and AUPRs, respectively, for the two inference techniques. The GC-MLP significantly outperforms the VAR for all lengths. For $T = 500$, the neural network approach identifies the correct causal graph in 92% of cases, whereas for sequences 1,000 and 5,000 points long it converges to the true structure in all simulated datasets. F-tests, on the other hand, yield correct conclusions in only, approximately, 30% of datasets. These results suggest that a nonlinear approach to identifying Granger causal relationships can be beneficial when compared to the conventional linear technique.

Let us examine neural network variable weights and adjusted F-test p-values obtained from time series with 5,000 points. Figure 5.1 depicts box plots of absolute values of weights and p-values. The GC-MLP correctly identifies that X drives Y, but Y does not cause X by shrinking most weights for $Y \longrightarrow X$ towards zero, as we would expect. The VAR, on the other hand,



(a) Absolute values of variable weights for GC-MLPs.

(b) F-test p-values from the VAR.

Figure 5.1: Box plots of GC-MLP weights and VAR p-values derived from 100 synthetic datasets with sequences 5,000 time steps long generated from the autoregressive model given by equations 5.2. Weights and p-values are provided for two candidate causal relationships $X \longrightarrow Y$ and $Y \longrightarrow X$.

fails to infer the correct structure, since p-values for $X \longrightarrow Y$ are frequently greater-than p-values for $Y \longrightarrow X$. Ideally, all p-values for tests on $X \longrightarrow Y$ should be less-than p-values for $Y \longrightarrow X$.

5.3 Nonlinear Autoregressive Model with Non-additive Interaction

Herein, we consider an autoregressive bivariate time series model with a non-additive interaction term. This example is due to Peters et al. and is taken from [59]. The time series is defined by the following equations:

$$\begin{aligned} X_t &= 0.2X_{t-1} + 0.9N_{X,t}, \\ Y_t &= -0.5 + \exp\left(-\left(X_{t-1} + X_{t-2}\right)^2\right) + 0.1N_{Y,t}, \end{aligned} \tag{5.3}$$

where $N_{\cdot,t} \sim \mathcal{N}(0,1)$ are innovation terms. The true summary graph is given by edge set $\{X \longrightarrow Y\}$. Observe that past values of X drive Y via non-additive interaction term $\exp\left(-\left(X_{t-1} + X_{t-2}\right)^2\right)$. When training neural networks and fitting the VAR model, the same parameter values were used as in the experiment described in Section 5.1. We generated 100 datasets with a single replicate of the bivariate series 500, 1,000 and 5,000 time points long.

Average AUROCs and AUPRs are shown in tables 5.5 and 5.6, respectively. On these data the performance of the GC-MLP is significantly superior to the VAR in terms of both curves. Neural networks converge to the correct structure in all datasets for time series of all lengths. This experiment demonstrates that the proposed method can account for non-additive interactions.

Method	Average AUROC($\pm$2SD)		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	0.545($\pm$0.712)	0.575($\pm$0.730)	0.540($\pm$0.748)
GC-MLP	1.000($\pm$0.000)	1.000($\pm$0.000)	1.000($\pm$0.000)

Table 5.5: Average AUROCs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the nonlinear autoregressive model given by equations 5.3.

Method	Average AUPR($\pm$2SD)		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	0.650($\pm$0.461)	0.675($\pm$0.479)	0.660($\pm$0.469)
GC-MLP	1.000($\pm$0.000)	1.000($\pm$0.000)	1.000($\pm$0.000)

Table 5.6: Average AUPRs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the nonlinear autoregressive model given by equations 5.3.

5.4 Lorenz 96 Model

A benchmark that is often used to test Granger causality inference techniques is the *Lorenz 96 model* [44] (for example, see [53] and [73]). It is a continuous time dynamical system introduced by E. N. Lorenz in 1996 while studying the physics of atmosphere [44]. The system consists of p variables and is given by differential equations:

$$\frac{dX_t^i}{dt} = \left(X^{i+1} - X^{i-2}\right) X^{i-1} - X^i + F, \tag{5.4}$$

where $X^0 = X^p$, $X^{-1} = X^{p-1}$ and $X^{p+1} = X^1$; and F is a forcing constant that, in combination with p, controls the chaos in the behaviour of the system. For larger values of F, the system tends to behave chaotically [44]. Time series can be obtained by numerical simulation with some fixed sampling rate [73]. The resulting causal structure is sparse (for p large) and features feedback, i.e. it contains two-node cycles. The summary causal graph for $p = 5$ is shown in Figure 5.2.

We simulate the Lorenz 96 system numerically with sampling rate $\Delta t = 0.01$ and obtain time series 500 points long for $p = 20$ variables. We generate data for two values of the forcing constant $F = 10$ and 40. In addition to VAR and GC-MLP techniques, we also consider a random classifier that randomly infers a causal relationship between each predictor and each target with the probability of, approximately, 0.157 (this is the fraction of the number of true causal interactions to the total number of possible interactions). In this experiment we use slightly different parameter values when training GC-MLPs. In particular, we set $\lambda = 0.01$ and use 50 hidden units in the third layer instead of 30.

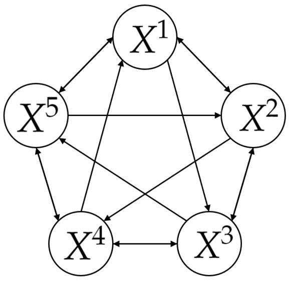


Figure 5.2: The summary causal graph of time series obtained from the Lorenz 96 system with $p = 5$ variables. Double-headed arrows correspond to feedback between two variables.

Method	**Average AUROC(±2SD)**	
	$F = 10$	$F = 40$
Random Classifier	0.500(±0.051)	0.505(±0.055)
VAR	0.922(±0.033)	0.745(±0.064)
GC-MLP	0.975(±0.017)	0.810(±0.072)

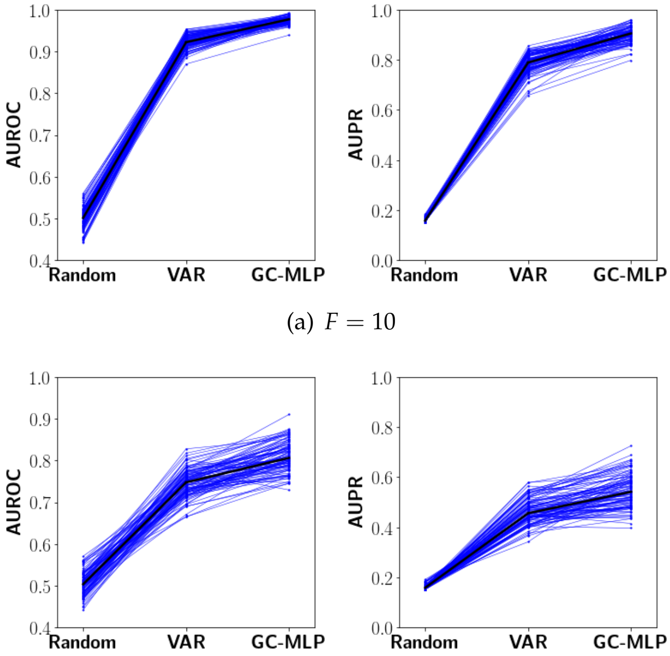
Table 5.7: Average AUROCs and standard deviations for random classifier, VAR and GC-MLP models for time series 500 points long generated from the Lorenz 96 model (see Equation 5.4) with $F = 10$ and 40.

Method	**Average AUPR(±2SD)**	
	$F = 10$	$F = 40$
Random Classifier	0.160(±0.014)	0.162(±0.055)
VAR	0.785(±0.077)	0.464(±0.010)
GC-MLP	0.901(±0.061)	0.546(±0.133)

Table 5.8: Average AUPRs and standard deviations for random classifier, VAR and GC-MLP models for time series 500 points long generated from the Lorenz 96 model (see Equation 5.4) with $F = 10$ and 40.

Tables 5.7 and 5.8 display average AUROCs and AUPRs, respectively, for the three compared methods applied to 100 Lorenz 96 datasets. First, note, that both VAR and GC-MLP techniques considerably outperform the random classifier. Moreover, our method has, on average, noticeably higher areas under ROC and PR curves than the vector autoregressive model, for both values of the forcing constant. These differences in performance are statistically significant (paired t-test $p < 0.001$). It also appears that both VAR and GC-MLP have lower AUROCs and AUPRs for $F = 40$ than for $F = 10$. This decrease is expected, since, as mentioned before, the Lorenz 96 system becomes chaotic and, therefore, less predictable for higher values of the forcing constant.

Figure 5.3 contains parallel coordinates plots with AUROCs and AUPRs of the the three considered methods. Every blue broken line in these plots corresponds to one of the 100 simulated datasets; the Y coordinate of each of



(a) $F = 10$

(b) $F = 40$

Figure 5.3: Parallel coordinates plots of AUPR and AUROC measures for the three methods applied to the Lorenz 96 data. Figure 5.3(a) shows the results for datasets generated with the forcing constant set to 10, whereas Figure 5.3(b) corresponds to $F = 40$. Note, that for the AUROC axis labels start from 0.4, rather than 0. The bold black line is the median.

three points on the line equals-to to the performance measure of one method on the dataset. Plots are provided for both forcing constant values. Observe that broken lines feature a consistent increase in AUROC and AUPR values from the random classifier to the VAR and from the VAR to the GC-MLP. This is also reflected by median broken lines, shown in black. Note, that the increase from the random classifier to the vector autoregressive model is usually the steepest.

To conclude, this experiment shows that the proposed technique can deal with the chaotic Lorenz 96 model. Even though it performs better than the conventional VAR, F-tests for GC are quite resilient to nonlinearity in these data. The results we obtained for the GC-MLP are on par with the average AUROCs reported in [73] for the componentwise multilayer perceptron that was tested in a similar setting.

5.5 Simulated fMRI Data

One possible application of Granger causal time series analysis is the investigation of interactions between different areas of the brain with the help

Simulation	Number of Variables	Sequence Length
1	5	200
2	10	200
3	15	200
5	5	1,200
6	10	1,200
7	5	5,000

Table 5.9: Characteristics of the six fMRI data simulations chosen for the comparison of methods. The number of each simulation is the same as its number in [30].

of functional magnetic resonance imaging (fMRI) [63]. In [71], Smith et al. compare various brain network modelling approaches on simulated fMRI signals. They use a realistic and rich model, based on the dynamic causal modelling, for generating blood oxygenation level dependent (BOLD) time series [71]. A detailed discussion of the model itself is beyond the scope of the thesis, therefore, we refer the interested reader to [71] and [23]. According to the results presented in [71], approaches based on Granger causality perform very poorly at inferring interactions between time series synthesised from this model. Therefore, it is interesting to investigate if the GC-MLP can improve upon conventional VAR F-tests.

In total, Smith et al. produced 28 simulations [71], each with 50 replicates of multivariate time series. Datasets differ based on the numbers of variables, lengths of sequences, connectivity structures, noise levels and other factors. The synthetic data are available in [30]. In this section, we compare the random classifier, VAR and GC-MLP models in terms of ROC and PR curves on six chosen simulations. Table 5.9 contains basic specifications of these simulations. As can be seen, we test techniques on time series with 5, 10 and 15 variables 200, 1,200 and 5,000 time steps long. When training GC-MLPs, we use the same hyperparameter values as in the Lorenz 96 experiment (see Section 5.4), apart from regularisation parameter λ that is set to 0.05. VAR models are fitted with the maximum lag of 5.

Tables 5.10 and 5.11 show the comparison of the three techniques in terms of AUROC and AUPR, respectively. They also contain adjusted paired t-test p-values for the difference in performance measures between the GC-MLP and the VAR. On average, in most simulations neural networks perform better than F-tests. Differences in performance are statistically significant for simulations 2, 3 and 6 w.r.t. AUROC and in simulations 1, 2, 3 and 6 w.r.t. AUPR. Nevertheless, in many cases differences are not extreme. It appears that using GC-MLP is more beneficial in datasets with more variables, namely, in simulations 2, 3, and 6, which have 10 and 15 covariates. Note, that with 5 variables both methods have quite large variances in areas under

Sim.	**Average AUROC($\pm$2SD)**			***t*-test**
	Random	VAR	GC-MLP	***p*-value**
1	0.475($\pm$0.226)	0.549($\pm$0.282)	**0.592($\pm$0.364)**	≥ 0.05
2	0.497($\pm$0.100)	0.566($\pm$0.182)	**0.654($\pm$0.165)**	$< \mathbf{0.0001}$
3	0.501($\pm$0.059)	0.559($\pm$0.178)	**0.656($\pm$0.142)**	$< \mathbf{0.0001}$
5	0.497($\pm$0.235)	0.717($\pm$0.279)	**0.733($\pm$0.278)**	≥ 0.05
6	0.491($\pm$0.103)	0.752($\pm$0.176)	**0.795($\pm$0.171)**	$< \mathbf{0.05}$
7	0.497($\pm$0.196)	**0.799($\pm$0.222)**	0.791($\pm$0.231)	≥ 0.05

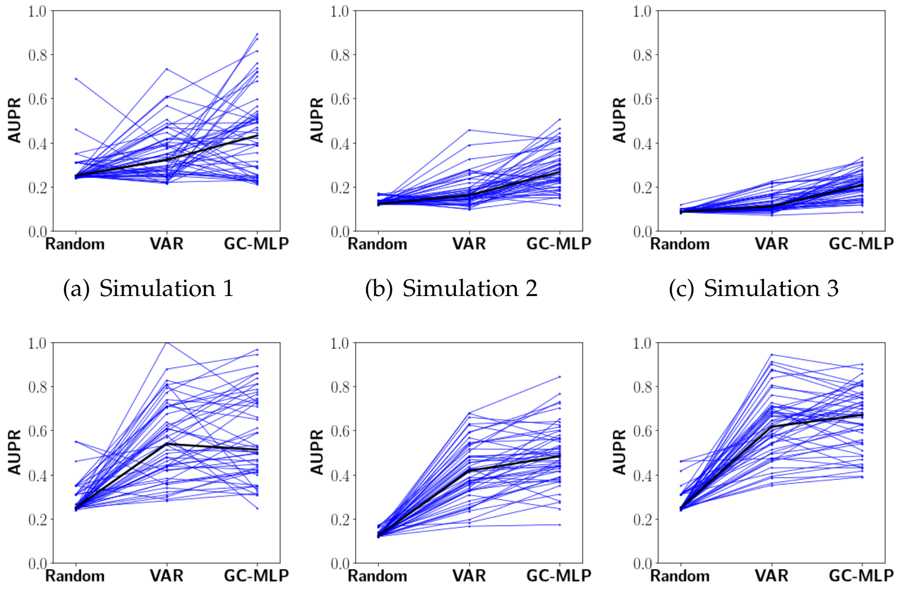
Table 5.10: Comparison of average AUROCs and standard deviations on six fMRI data simulations retrieved from [30]. The last column provides p-values from the paired t-test between AUROCs of the GC-MLP and of the VAR.

Sim.	**Average AUPR($\pm$2SD)**			***t*-test**
	Random	VAR	GC-MLP	***p*-value**
1	0.272($\pm$0.145)	0.349($\pm$0.234)	**0.444($\pm$0.384)**	$< \mathbf{0.05}$
2	0.129($\pm$0.028)	0.182($\pm$0.143)	**0.283($\pm$0.188)**	$< \mathbf{0.0001}$
3	0.089($\pm$0.013)	0.126($\pm$0.081)	**0.208($\pm$0.115)**	$< \mathbf{0.0001}$
5	0.284($\pm$0.143)	0.55($\pm$0.348)	**0.569($\pm$0.39)**	≥ 0.05
6	0.129($\pm$0.030)	0.418($\pm$0.265)	**0.491($\pm$0.272)**	$< \mathbf{0.0001}$
7	0.272($\pm$0.107)	0.621($\pm$0.301)	**0.649($\pm$0.278)**	≥ 0.05

Table 5.11: Comparison of average AUPRs and standard deviations on six fMRI data simulations retrieved from [30]. The last column provides significance of p-values from the paired t-test between AUPRs of the GC-MLP and of the VAR. p-values were adjusted using the Bonferroni correction.

both curves. Figure 5.4 depicts parallel coordinates plots of AUPRs for each simulation. Parallel coordinates with AUROCs are given in Figure C.1 in Appendix C. Observe that in most simulations the median line (plotted in black) features an increase. However, slopes of blue lines from the VAR to the GC-MLP are noticeably less consistent than in Lorenz 96 datasets.

Overall, from this experiment we see that neural networks are non-inferior and even sometimes superior to Granger causality inference with VAR on simulated realistic and rich fMRI time series. Nevertheless, their performance is still quite poor. There might be several reasons that were mentioned in [71]. Namely, BOLD time series are characterised by a low signal-to-noise ratio and by causal interactions with very small lags. Moreover, it is important to note, that several datasets have only 200 points long time series (simulations 1, 2 and 3) and, thus, represent a scenario of very limited training data.



(a) Simulation 1

(b) Simulation 2

(c) Simulation 3

(d) Simulation 5

(e) Simulation 6

(f) Simulation 7

Figure 5.4: Parallel coordinates plots of AUPR measures for the three methods applied to six different simulations [30]. The bold black line corresponds to the median.

5.6 Categorically-valued Time Series

As mentioned before, the proposed model provides a principled way of accounting for categorically-valued time series by replacing the sum-of-squares term in the penalised loss with the cross-entropy and adjusting the output layer. In this section we consider two examples with both continuously- and categorically-valued variables.

5.6.1 Experiment 1

We first examine a trivariate time series given by the following equations:

$$\begin{aligned} X_t &= 0.3X_{t-1} + N_{X,t}, \\ W_t &= -0.6W_{t-1} + 0.25X_{t-1} - 0.5Y_{t-1} + 0.3N_{W,t}, \\ Y_t &= \mathbb{1}_{\left\{\frac{1}{5}\sum_{j=1}^{5} X_{t-j} \geq -\frac{1}{4}\right\}}, \end{aligned} \tag{5.5}$$

where $N_{\cdot,t} \sim \mathcal{N}(0,1)$ are innovation terms. Note, that X_t and W_t are continuously-valued, whereas $Y_t \in \{0,1\}$ is binary-valued. The true summary causal graph has edge set $\{X \longrightarrow W, X \longrightarrow Y, Y \longrightarrow W\}$. In this example all causal interactions are linear. Observe that classes given by Y_t are linearly separable based on the past values of X_t. We generate 100 datasets from this time series model with sequences 500, 1,000 and 5,000 steps long.

Method	**Average AUROC($\pm$2SD)**		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	1.000($\pm$0.000)	1.000($\pm$0.000)	1.000($\pm$0.000)
GC-MLP	0.997($\pm$0.038)	1.000($\pm$0.000)	1.000($\pm$0.000)

Table 5.12: Average AUROCs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the model given by equations 5.5.

Method	**Average AUPR($\pm$2SD)**		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	1.000($\pm$0.000)	1.000($\pm$0.000)	1.000($\pm$0.000)
GC-MLP	0.998($\pm$0.029)	1.000($\pm$0.000)	1.000($\pm$0.000)

Table 5.13: Average AUPRs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the model given by equations 5.5.

When training neural networks, we use the same hyperparameters as in the experiment described in Section 5.4.

Tables 5.12 and 5.13 show average AUROCs and AUPRs, respectively, for the two inference techniques applied to time series of different lengths. The GC-MLP infers the fully correct causal graph in 97 out of 100 datasets for 500 points long sequences and is correct in all cases for 1,000 and 5,000 points long time series. On the other hand, the VAR identifies the correct structure in all datasets for all lengths. Despite the fact that the vector autoregressive model is only appropriate for continuously-valued time series, in this simple example, the misspecification does not adversely affect the inference of causal relationships. At the same time, the GC-MLP performs equally well, while accounting for differences in data types.

5.6.2 Experiment 2

Let us now slightly adjust the autoregressive model defined in equations 5.5 by introducing nonlinearity into the causal relationship between variables X_t and Y_t. The adjusted multivariate time series is given by

$$\begin{aligned} X_t &= 0.3X_{t-1} + N_{X,t}, \\ W_t &= -0.6W_{t-1} + 0.25X_{t-1} - 0.5Y_{t-1} + 0.3N_{W,t}, \\ Y_t &= \mathbb{1}_{\left\{-\frac{1}{4}\leq\frac{1}{5}\sum_{j=1}^{5}X_{t-j}\leq\frac{1}{4}\right\}}, \end{aligned} \tag{5.6}$$

where $N_{\cdot,t} \sim \mathcal{N}(0,1)$ are innovation terms. Note, that the causal summary graph remains the same as in the previous experiment.

The comparison of VAR and GC-MLP techniques on 100 datasets sampled from this time series model is provided in tables 5.14 and 5.15. Neural networks perform clearly better than F-tests in terms of both AUROC and

Method	**Average AUROC($\pm$2SD)**		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	0.820($\pm$0.202)	0.831($\pm$0.220)	0.847($\pm$0.227)
GC-MLP	1.000($\pm$0.000)	1.000($\pm$0.000)	1.000($\pm$0.000)

Table 5.14: Average AUROCs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the model given by equations 5.6.

Method	**Average AUPR($\pm$2SD)**		
	$T = 500$	$T = 1000$	$T = 5000$
VAR	0.873($\pm$0.111)	0.882($\pm$0.118)	0.889($\pm$0.133)
GC-MLP	1.000($\pm$0.000)	1.000($\pm$0.000)	1.000($\pm$0.000)

Table 5.15: Average AUPRs and standard deviations for VAR and GC-MLP models for time series with lengths (T) 500, 1,000 and 5,000 generated from the model given by equations 5.6.

AUPR. In particular, GC-MLPs estimate the correct Granger causal structure in all datasets for all time series lengths, whereas the VAR retrieves the correct graph, on average, in merely 15% of cases. This example demonstrates that the absence of linear separability between classes of a categorical variable can lead to biases when inferring Granger causality with the linear model.

5.7 Reversed Time Analysis

As mentioned before, testing for time-reversed Granger causality can be a promising 'shortcut' when trying to discover a set of Granger effects, rather than causes. Nevertheless, the theoretical properties of TRGC are not well understood, therefore, in this section we examine the empirical performance of neural networks at estimating Granger causality on time-reversed sequences.

We selected several time series models from the experiments described in the previous sections and performed time-reversed causal analysis with the GC-MLP technique. Namely, we look at time series given by equations 5.1, 5.2, 5.3, 5.4 and 5.6. For each model, we generated 100 datasets and considered sequences 500, 1,000 and 5,000 steps long. In the Lorenz 96 system the forcing constant was set to 10. During inference we used the same hyperparameter values as in the corresponding experiments without time reversal.

Results, summarised in tables 5.16 and 5.17, suggest that it is possible to infer correct causal structures by training neural networks on time-reversed sequences. In time series models 5.2, 5.3 and 5.6, given sufficiently long sequences, the GC-MLP manages to retrieve fully correct summary graphs in all 100 datasets. In models 5.2, 5.3, 5.4 and 5.6 we observe improve-

Time Series Model	Average AUROC($\pm$2SD)		
	$T = 500$	$T = 1000$	$T = 5000$
Linear (5.1)	0.890($\pm$0.263)	0.932($\pm$0.209)	0.906($\pm$0.267)
Nonlinear (5.2)	0.650($\pm$0.959)	0.690($\pm$0.930)	1.000($\pm$0.000)
Interaction (5.3)	0.710($\pm$0.912)	0.950($\pm$0.438)	1.000($\pm$0.000)
Lorenz 96 (5.4)	0.948($\pm$0.036)	0.987($\pm$0.011)	0.989($\pm$0.004)
Categorical (5.6)	0.901($\pm$0.223)	0.952($\pm$0.174)	1.000($\pm$0.000)

Table 5.16: Average AUROCs and standard deviations for time-reversed Granger causality inference with GC-MLPs for 5 different time series models. Averages were computed across 100 synthetic datasets for sequences with lengths (T) of 500, 1,000 and 5,000.

Time Series Model	Average AUPR($\pm$2SD)		
	$T = 500$	$T = 1000$	$T = 5000$
Linear (5.1)	0.928($\pm$0.175)	0.955($\pm$0.135)	0.929($\pm$0.209)
Nonlinear (5.2)	0.825($\pm$0.479)	0.845($\pm$0.465)	1.000($\pm$0.000)
Interaction (5.3)	0.855($\pm$0.456)	0.975($\pm$0.219)	1.000($\pm$0.000)
Lorenz 96 (5.4)	0.814($\pm$0.100)	0.944($\pm$0.044)	0.946($\pm$0.021)
Categorical (5.6)	0.939($\pm$0.123)	0.970($\pm$0.103)	1.000($\pm$0.000)

Table 5.17: Average AUPRs and standard deviations for time-reversed Granger causality inference with GC-MLPs for 5 different time series models. Averages were computed across 100 synthetic datasets for sequences with lengths (T) of 500, 1,000 and 5,000.

ments in performance with the increase in the lengths of observed time series. In contrast, for the linear autoregressive model (see Equation 5.1) the time-reversed GC inference with neural networks estimates the correct causal graph in only 59 and 58 datasets out of 100 for sequences 1,000 and 5,000 points long, respectively. A quite high failure rate for such a simple autoregressive model is worrisome, however, it might be attributed to an inappropriate choice of hyperparameters.

To summarise, from this experiment we see that the time-reversed GC inference with MLPs is feasible, but appears to be more challenging and demanding w.r.t. training data than the GC inference without time reversal. Nevertheless, TRGC can still be an advantageous computational 'shortcut' when we are interested in discovering only the set of variables driven by the given target.

Chapter 6

MS Data Analysis

In this chapter we provide the results of causal time series analysis of synchronised mass spectrometry and sleep stage data. We consider Granger causal relationships in two directions: from ion intensities to sleep phases and from sleep phases to ion intensities. In addition, we perform simulation experiments with the MS data to support our choice of hyperparameters and verify that our inference technique behaves as expected. Last but not least, we also assess the predictive performance of trained neural networks using cross-validation.

6.1 Data Processing and Analysis Procedure

Causal analysis of mass spectrometric and sleep stage time series performed in this thesis consists of several steps listed below in the chronological order:

1. Normalisation of mass spectra using internal standards (see Subsection 3.2.1).

2. Ion intensity time series smoothing using Savitzky–Golay filter (see Subsection 3.2.2).

3. Ion intensity time series standardisation (see Subsection 3.2.2).

4. Application of the bootstrapping procedure, given in Algorithm 1, to the pre-processed positive and negative mode data, in order to identify ions that Granger-cause sleep phases. The bootstrapping is run separately for three binary-valued targets derived from the original sleep stage time series: wakefulness-vs.-all, NREM-vs.-all and REM-vs.-all. We consider these three cases, because we are interested in discovering causal relationships characteristic of particular stages of sleep.

5. Analysis in the previous step is repeated on time-reversed positive and negative mode sequences, in order to identify ions that are Granger-

Ion Mode	Number of Discoveries		
	Wake	NREM	REM
Positive	101	49	87
Negative	261	106	126

Table 6.1: Counts of ions identified as Granger causes of different sleep phases for positive and negative modes.

caused by sleep phases (see Section 4.4 for details on time-reversed Granger causality).

6.1.1 Hyperparameters

We train neural networks on each of $B = 1000$ bootstrap re-samples of the data. Each GC-MLP has 100 hidden units in layers 1 and 2 (of every subnetwork) and 200 hidden units in layer 3. The model order is chosen to be $K = 30$, i.e. we consider autoregressive relationships with lags up to 300 seconds. We set regularisation parameter λ to 0.001 and choose $\alpha = 0.8$. We use the penalised weighted cross-entropy, given by Equation 4.9, as the loss function. The weight of 0.9 is assigned to the less prevalent class, whereas the weight of 0.1 is assigned to the more prevalent one. The training is performed for one epoch by gradient descent using the PyTorch [56] implementation of Adam optimiser [39] with mini-batches of 100 data points. For the bootstrapping procedure, we use parameter values $c_{th} = 0.0025$ and $\alpha = 0.05$ (not to be confused with α that controls the trade-off between L_1 and L_2 penalties in the loss function of GC-MLPs).

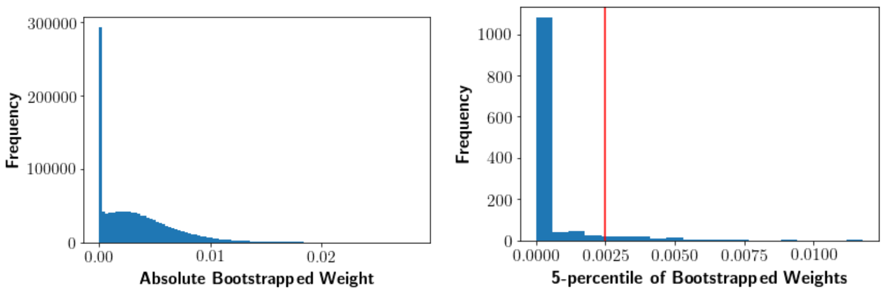
6.2 Results

In this section we summarise and discuss the results of causal time series analysis. As mentioned before, we consider two possible directions of causal relationships for positive and negative mode data.

6.2.1 Ions Driving Sleep Stages

We apply bootstrapping to the original data to find a set of ions that Granger-cause sleep stage transitions. The analysis is conducted separately for three sleep phases: wakefulness, NREM (includes stages N_1, N_2 and N_3) and REM.

Recall that after bootstrapping we obtain 5-percentiles of weights for each ion (denoted by q_j in Algorithm 1). The discovery is claimed if the percentile exceeds specified threshold c_{th}. Numbers of discoveries made by the technique for every sleep stage for both ion modes are shown in Table 6.1. Wakefulness has the largest count of causal ions, in total, 362; it is followed by REM phase with 213 discoveries and NREM stages that are driven



(a) (b)

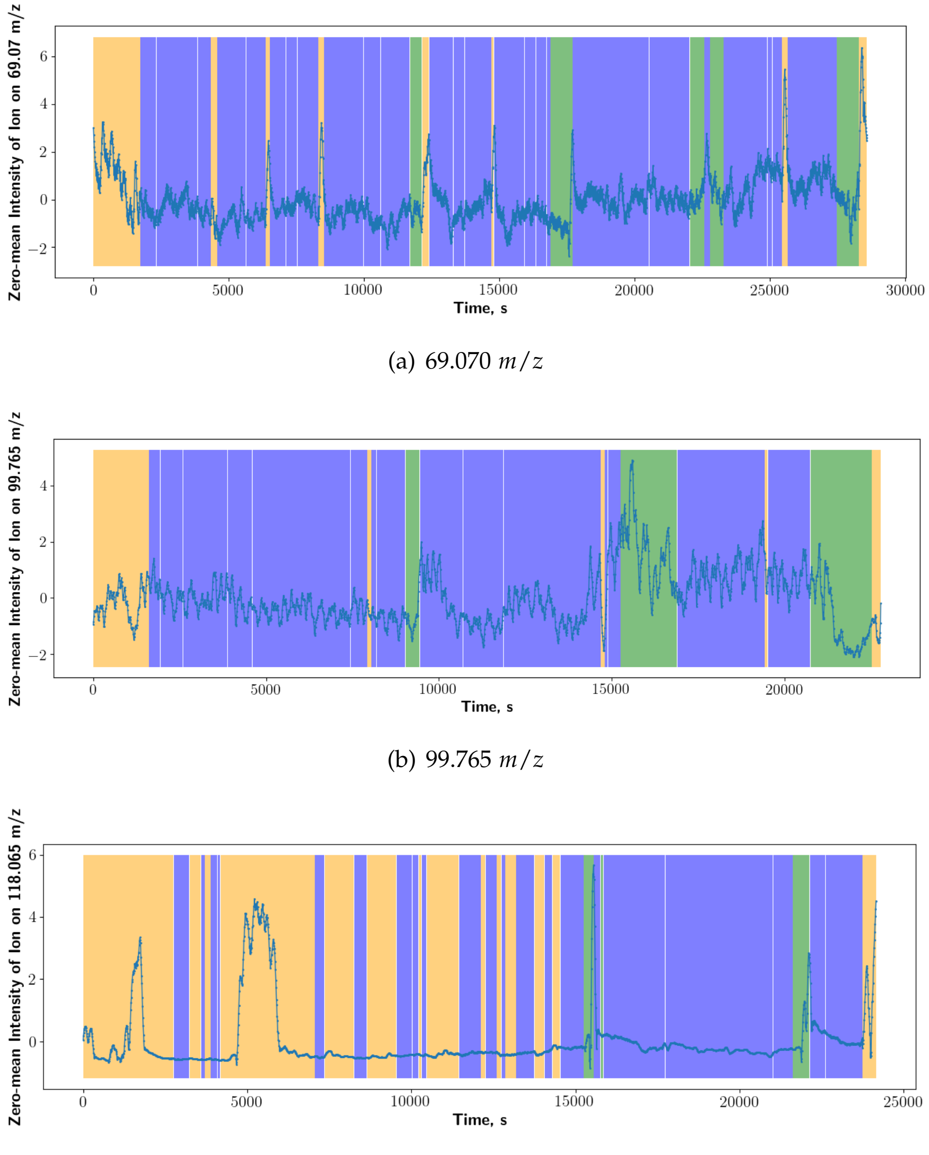
Figure 6.1: Histograms of all bootstrapped absolute variable weights (on the left) and 5-percentiles of bootstrapped weights (on the right) obtained from positive mode data with REM response time series. The red vertical line in the histogram to the right corresponds to $c_{th} = 0.0025$.

by 155 variables. Lists with mass-to-charge ratios of positive and negative causal ions are given in Appendix D.1. Let us examine distributions of absolute values of bootstrapped variable weights and resulting 5-percentiles for REM time series. From histograms plotted in Figures 6.1(a) and 6.1(b) we observe that a large number of bootstrapped variable weights are not shrunk to zero and that weight 5-percentiles of many ions exceed threshold $c_{th} = 0.0025$. Similar weight distributions can be observed for other phases. These distributions clearly suggest that our inference technique identifies some significant associations between the target and considered predictors.

It is interesting that in the positive mode, for all three phases of sleep, we identify the ion with the mass-to-charge ratio of 69.06988 as causal. Moreover, 5-percentiles of bootstrapped weights for this variable are among top five largest percentiles for all phases of sleep. This m/z primarily corresponds to *isoprene* [1], a volatile organic compound the abundance of which in human breath is hypothesised to be associated with leg movements [38]. Peaks in its intensity that can be clearly seen in Figure 3.2 often coincide with leg muscle contractions. Its correlation with sleep stage signals was discussed in the literature before [1, 38].

Some noteworthy patterns which are visible in univariate time series plots of positive and negative mode ions that were identified as causal are shown in Figures 6.2 and 6.3. In fact, all of the ions provided in the plots were discovered to drive all three sleep phases. Observe that in these sequences sleep stages quite consistently coincide with fluctuations of ion intensities. In particular, sharp peaks of ion abundances often occur during phases of wakefulness and REM sleep. Similar patterns can be spotted in many other metabolites discovered in the causal analysis, whereas in some cases the association is not as straightforward.

Since prediction tasks of different sleep phases based on mass spectromet-

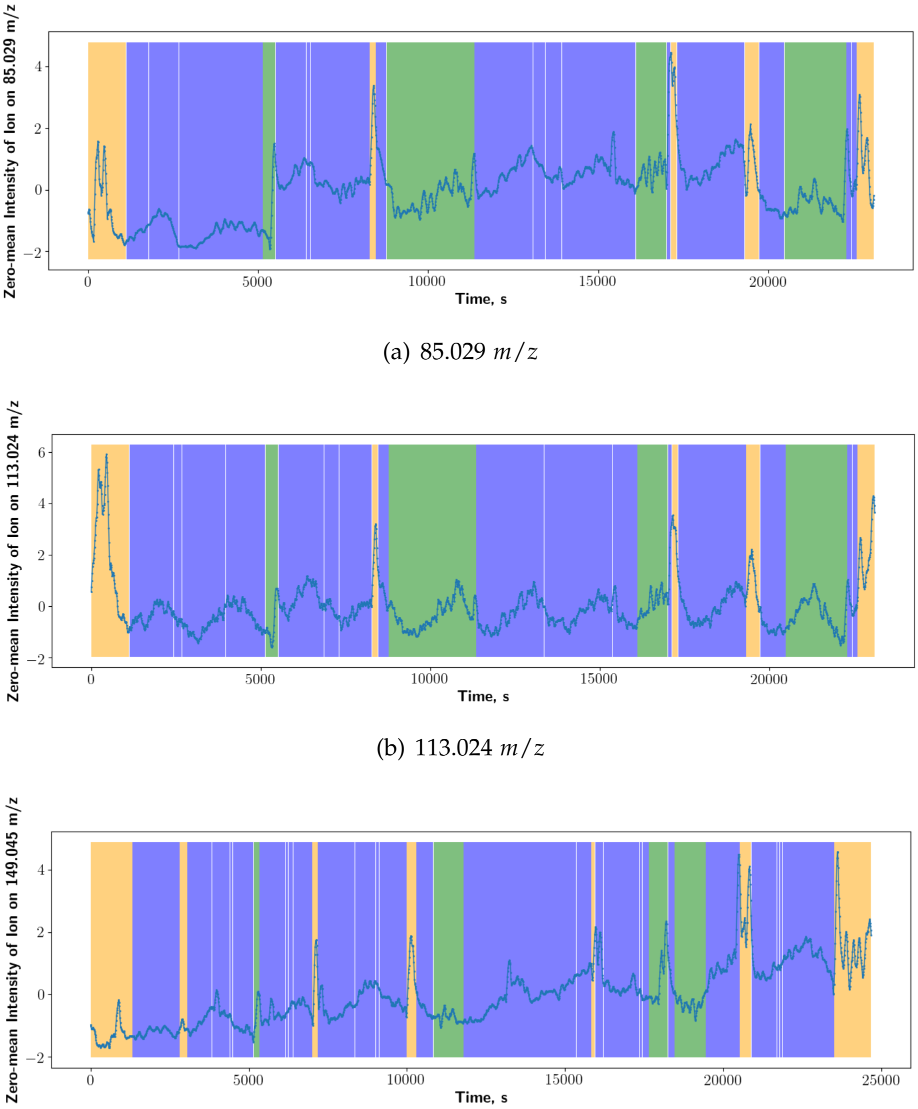


(a) 69.070 m/z

(b) 99.765 m/z

(c) 118.065 m/z

Figure 6.2: Pre-processed relative intensity time series for three positive ions superimposed with synchronised sleep stage labels. The ions were discovered to Granger-cause all stages. Sleep phases are shown in different colours: orange corresponds to wakefulness, blue to NREM, and green to REM. Note, that the time series originate from different subjects.

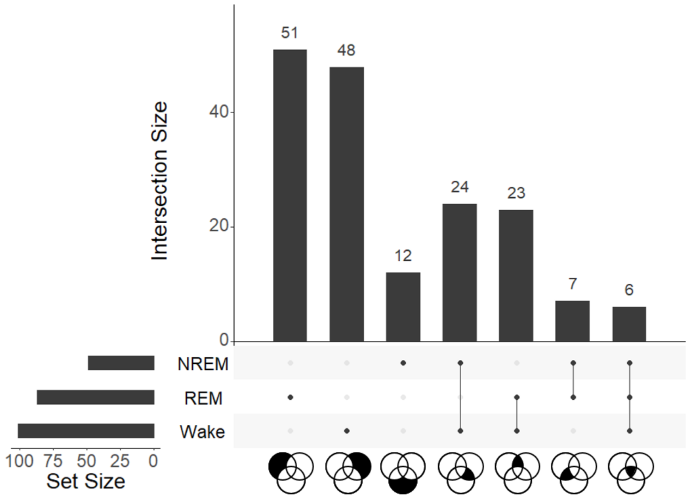


(a) 85.029 m/z

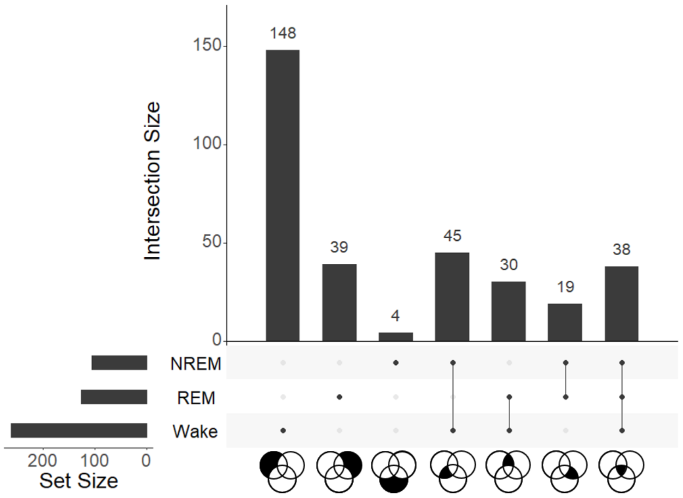
(b) 113.024 m/z

(c) 149.045 m/z

Figure 6.3: Pre-processed relative intensity time series for three negative ions superimposed with synchronised sleep stage labels. The ions were discovered to Granger-cause all stages.



(a) Positive mode



(b) Negative mode

Figure 6.4: Visualisations of intersections between sets of ions that Granger-cause wakefulness, NREM and REM sleep phase time series for positive and negative modes. Every bar corresponds to an intersection that is annotated below. Note, that the corresponding intersections are also shown in Venn diagrams at the bottom. These plots were produced using UpSetR library [12] in R programming language.

ric features should be related, we expect sets of ions identified as Granger-causing different stages of sleep to overlap. Figure 6.4 contains bar plots with sizes of all intersections for both ion modes. Inferred sets of causal ions overlap substantially, all two-way and three-way intersections are not empty in both modes. Observe that NREM sleep phase has the least number of variables discovered to uniquely cause it (only 12 in the positive mode

Intersection	***p*-value**	
	Positive	Negative
Wake ∩ NREM	< 0.01	< 0.0001
NREM ∩ REM	< 0.01	< 0.0001
Wake ∩ REM	< 0.0001	< 0.0001
Wake ∩ NREM ∩ REM	< 0.0001	< 0.0001

Table 6.2: p-values of tests of independence between sets of ions identified to Granger-cause different stages of sleep. p-values were adjusted with the Bonferroni method.

Ion Mode	**Number of Discoveries**		
	Wake	NREM	REM
Positive	119	70	84
Negative	267	113	112

Table 6.3: Counts of ions identified as being Granger-caused by different sleep phases for positive and negative modes.

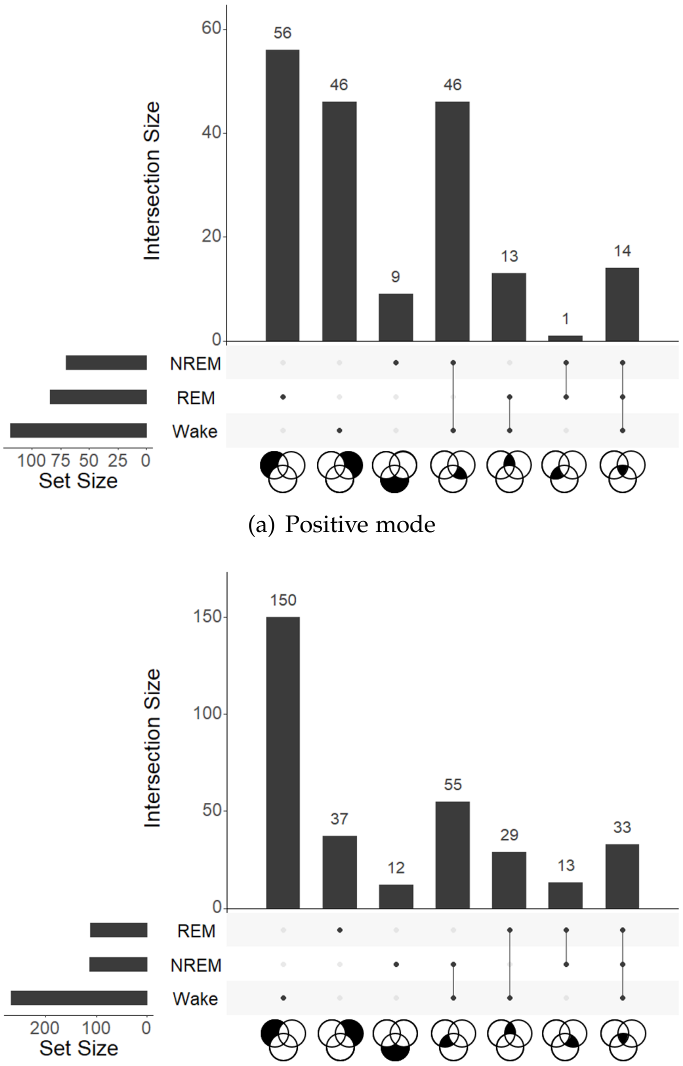
and 4 in negative). Thus, NREM shares most of its causes with the other two phases. Given that stages of NREM can be seen as a gradual transition from wakefulness to deep sleep, it makes sense that many drivers of NREM are the same as of the two other phases.

To see if sizes of intersections differ significantly from sizes that can be obtained by choosing sets of ions independently for each phase, we perform statistical tests. In particular, for two-way intersections, we use the Chi-Square test of independence and, for three-way intersections, we estimate p-values using numerical simulations. Table 6.2 contains adjusted p-values for all intersections of sets of causal ions discovered in both modes. All overlaps between the three sets are statistically significant at level $\alpha = 0.05$. Thus, we can conclude that the three prediction tasks, probably, have a substantial amount of common 'useful' covariates.

6.2.2 Ions Driven by Sleep Stages

To infer Granger causality from sleep phases to ion intensities, we perform bootstrapping and train neural networks on time-reversed sequences. In contrast to the previous subsection, time reversal allows discovering Granger effects of the target time series, rather than its causes.

Table 6.3 contains counts of variables that were identified as being driven by sleep phases for both modes of mass spectrometry. Observe that wakefulness Granger-causes the largest number of ion intensity time series in both modes; in total, it drives 386 ions. NREM and REM were discovered to influence 183 and 196 variables, respectively. Appendix D.2 contains lists of mass-to-charge ratios of ions discovered in the time-reversed analysis from



(a) Positive mode

(b) Negative mode

Figure 6.5: Visualisations of intersections between sets of ions Granger-caused by wakefulness, NREM and REM sleep phase time series for positive and negative modes.

positive and negative mode data.

Similarly to the previous subsection, we look at the intersections of sets of ions identified to be driven by the response. Figure 6.5 depicts sizes of intersections for both modes of mass spectrometry. There are many ions that are shared between the sets. Even the three-way intersections are non-empty. Observe that the overlap between sets for wakefulness and NREM is larger, for both ion modes, than the intersection between wakefulness and REM. This difference make sense, because NREM stages can be seen as a gradual

Intersection	*p*-value	
	Positive	Negative
Wake ∩ NREM	< 0.0001	< 0.0001
NREM ∩ REM	< 0.01	< 0.0001
Wake ∩ REM	< 0.0001	< 0.001
Wake ∩ NREM ∩ REM	< 0.0001	< 0.0001

Table 6.4: *p*-values of tests of independence between sets of ions identified to be Granger-caused by different stages of sleep. *p*-values were adjusted with the Bonferroni method.

Sleep Phase	Number of Causes	Number of Effects	Number of Common Causes & Effects
Wakefulness	101	119	80
NREM	49	70	28
REM	87	84	58

Table 6.5: Numbers of ions identified simultaneously as Granger causes and effects in the positive mode.

Sleep Phase	Number of Causes	Number of Effects	Number of Common Causes & Effects
Wakefulness	261	267	195
NREM	106	113	75
REM	126	112	86

Table 6.6: Numbers of ions identified simultaneously as Granger causes and effects in the negative mode.

transition from wakefulness to deep sleep and, thus, there should be more in common between wakefulness and NREM than wakefulness and REM.

In the same way as before, we test if sizes of set intersections are statistically significant. Adjusted *p*-values (by the Bonferroni correction) are shown in Table 6.4 for positive and negative modes. As can be seen, all intersections are declared significant at level $\alpha = 0.05$. This suggests that the sets of ions Granger-caused by sleep phases were, likely, not chosen independently.

6.2.3 Feedback

Herein we investigate if the inferred causal relationships between ion intensity and sleep stage time series feature any feedback, i.e. we look for ions that simultaneously drive and are driven by phases of sleep. Tables 6.5 and 6.6 contain numbers of common Granger causes and effects of different phases of sleep identified from positive and negative mode data, respectively. Observe that in both modes most of discovered ions drive and are driven by sleep stages. If causal feedback connections are not spurious, these results suggest that metabolism and sleep regulate each other. There exist other possible reasons for causal loops, namely: unobserved confounders

Sleep Phase	Average Balanced Accuracy($\pm$2SD)	
	Positive	Negative
Wake	0.771($\pm$0.146)	0.763($\pm$0.266)
NREM	0.597($\pm$0.155)	0.621($\pm$0.089)
REM	0.676($\pm$0.289)	0.747($\pm$0.174)

Table 6.7: Average balanced accuracy scores and standard deviations for leave-one-subject-out cross-validation of GC-MLPs for different phases of sleep obtained from positive and negative modes.

that influence both ion intensity and sleep stage time series; instant causality; and not sufficiently frequent sampling that prevents the detection of the direction [49].

6.3 Model Validation

A valid question to investigate is whether it is possible to predict future sleep stages solely based on past mass spectrometric profiles. Causal discoveries discussed before would be questionable, if trained neural networks possess no predictive power. Therefore, in this section we validate tour model on reversed-time sequences (similar results were obtained on sequences without time reversal, but we omit them). The only adjustment that we make compared to the setting in the previous sections is that we do not include sleep stage time series as a predictor. We use leave-one-subject-out cross-validation (CV) to see how well GC-MLPs generalise across different subjects. Namely, for each iteration, we leave out one subject and train a neural network on the rest. To evaluate the performance, we employ the balanced accuracy score as implemented in scikit-learn library [58]. This score is more appropriate than the normal accuracy because of imbalances in frequencies of classes.

Average balanced accuracy CV scores are shown in Table 6.7. In all datasets the mean scores are significantly greater-than 0.5 at level $\alpha = 0.05$. Thus, on average, in all prediction tasks GC-MLPs perform better than the reference level. In addition, we examine normal accuracy CV scores, which are provided in Appendix D.3 in Table D.1. As can be seen, the mean scores differ significantly from 0.5 only for wakefulness and REM stages, and not for NREM. Based on both performance metrics, we see that NREM time series appear to be the most difficult to predict from mass spectrometric features. This could be because of the similarity of NREM to the other two stages.

To sum up, the results of cross-validation show that there might be some structure in the data driven by differences between phases of sleep. Nevertheless, the predictive performance of GC-MLPs on these datasets is by far not perfect, especially, for NREM stages, for which the mean accuracy

Sleep Phase	**Average Balanced Accuracy($\pm$2SD)**	
	Linear	Nonlinear
Wake	0.725($\pm$0.163)	**0.771($\pm$0.146)**
NREM	0.586($\pm$0.132)	**0.597($\pm$0.155)**
REM	0.652($\pm$0.362)	**0.676($\pm$0.289)**

Table 6.8: Average balanced accuracy scores and standard deviations for leave-one-subject-out cross-validation of GC-MLPs with linear and nonlinear activation functions for different phases of sleep. The scores were obtained from the positive mode of mass spectrometry.

Sleep Phase	**Average Balanced Accuracy($\pm$2SD)**	
	Linear	Nonlinear
Wake	0.695($\pm$0.278)	**0.763($\pm$0.266)**
NREM	**0.634($\pm$0.084)**	0.621($\pm$0.089)
REM	0.691($\pm$0.207)	**0.747($\pm$0.174)**

Table 6.9: Average balanced accuracy scores and standard deviations for leave-one-subject-out cross-validation of GC-MLPs with linear and nonlinear activation functions for different phases of sleep. The scores were obtained from the negative mode of mass spectrometry.

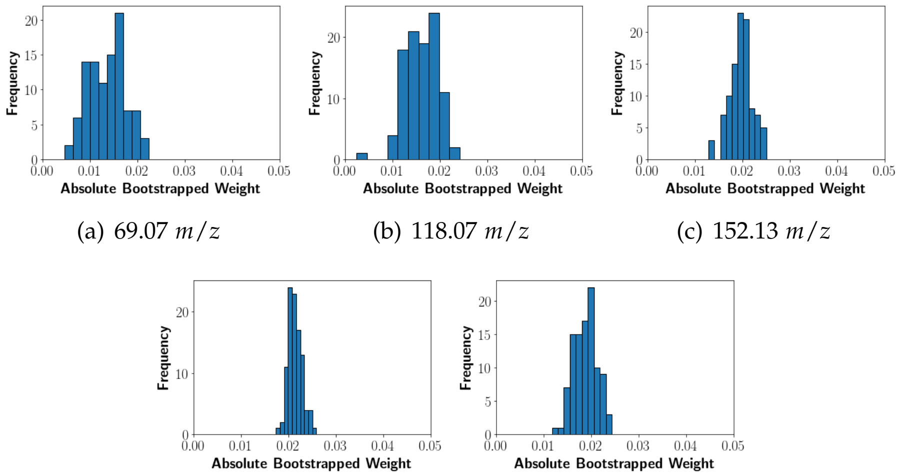
does not differ significantly from the performance level of the random classifier. Therefore,the inference results obtained for this sleep phase should be interpreted with caution.

6.3.1 Nonlinearity

It is interesting to investigate if there is any evidence for nonlinearity in the relationship between ion intensities and stages of sleep. To address this issue, we perform leave-one-subject-out cross-validation for a GC-MLP model with all linear activation functions. We then compare the performance of linear GC-MLPs to the original model with ReLU. Tables 6.8 and 6.9 show average balanced accuracy scores obtained from CV based on positive and negative mode data, respectively. In most cases, the nonlinear version, on average, has a superior balanced accuracy score. However, in neither of datasets the difference in scores is statistically significant. While the GC-MLP with nonlinear activation functions has cross-validations results that are non-inferior to the linear approach, there is no significant evidence for a nonlinearity in the association between sleep stage signals and mass spectrometric profiles.

6.4 Simulation Experiments with MS Data

We perform several simulation experiments, in order to verify that our causal neural network inference technique alongside with bootstrapping behave as expected on the mass spectrometry data in various controlled settings. Namely, we explore the number of false discoveries made in two different



(a) 69.07 m/z

(b) 118.07 m/z

(c) 152.13 m/z

(d) 229.25 m/z

(e) 271.30 m/z

Figure 6.6: Histograms of absolute values of bootstrapped weights for five fixed ions obtained from a simulated dataset with permuted ion intensity time series.

scenarios and investigate the relationship between the number of false discoveries and regularisation parameter λ.

6.4.1 Permuted Ion Intensities

The first experiment we perform is permutation of ion intensity time series. We consider five ions that were originally discovered as Granger-causing the REM sleep stage time series, in particular, we chose ions with mass-to charge ratios 69.06988, 118.06503087, 152.12762686, 229.25221961 and 271.29923335. We generate 10 synthetic datasets wherein we randomly permute all intensity time series except for the sequences of these five ions. Subsequently, we apply the bootstrapping procedure to these datasets with $B = 100$ resamples. We use the configuration of hyperparameters described in Subsection 6.1.1. We expect that none of the variables the time series of which were permuted are identified as causal, whereas the five ions that remain fixed should be.

Figure 6.6 contains histograms of absolute values of bootstrapped weights for the five ions, obtained from one of the simulated datasets. Observe that every of these ions is discovered as causal, because all 5-percentiles are clearly greater-than $c_{th} = 0.0025$. In other nine simulations we observe similar results, i.e. all of the variables that were not permuted are claimed to Granger-cause the REM sleep stage. Moreover, none of the permuted time series in all 10 simulations were discovered to drive the response.

The results of this experiment completely agree with our initial expectations.

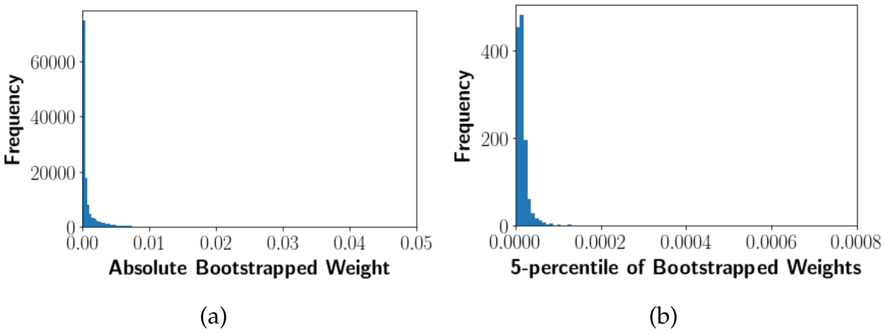


Figure 6.7: Histograms of all bootstrapped absolute variable weights (on the left) and 5-percentiles of bootstrapped weights (on the right) obtained from one dataset with permuted sleep stage labels.

We see that weights of all irrelevant variables are consistently shrunk towards zero, and weights of time series that, possibly, drive the response are not.

6.4.2 Permuted Sleep Stage Labels

Another experiment we perform is with randomly permuting REM sleep stage labels while keeping ion intensity time series untouched. In this setting, we expect our inference technique to identify no variables that are causally related with the permuted target. We run the bootstrapping procedure on 10 different simulated datasets with $B = 100$ re-samples. We use the same hyperparameter values as described in Subsection 6.1.1 and do not perform time reversal.

Figure 6.7(a) depicts the histogram of absolute values of all bootstrapped variable weights obtained from one simulated dataset, and Figure 6.7(b) shows the distribution of 5-percentiles computed for each variable. As expected, most weights are shrunk towards zero. Observe that in this case none of 5-percentiles exceed threshold $c_{th} = 0.0025$. Similar weight distributions were obtained for other nine simulated datasets. In two cases we discovered a causal relationship from permuted sleep stage time series to itself, however, in all datasets no relationships were found from ions to the target. These results are satisfactory, since almost no spurious causal links were inferred.

6.4.3 Synthetic Sleep Stage Labels

The setting considered in the previous experiment might be too optimistic, since permuted sleep stage time series are too unstructured and are very unlikely to be correlated with ion intensities. Therefore, herein we generate

λ	**Numbers of False Discoveries**	**Average Number of False Discoveries**
0.0000	7, 27, 32, 118, 38, 26, 28, 34, 9, 48	36.7
0.0001	43, 21, 73, 29, 70, 14, 12, 60, 17, 14	35.3
0.0010	13, 1, 7, 18, 32, 5, 23, 33, 8, 4	14.4
0.0100	1, 0, 0, 0, 1, 0, 1, 0, 0, 0	0.3

Table 6.10: Numbers of ions falsely identified as Granger-causing the synthetic target time series for different values of the regularisation parameter. Note, that causal inference was performed on 10 datasets.

synthetic target time series which behave like REM sequences. Similarly to the setting above, we expect no causal links to be inferred. We perform bootstrapping on 10 simulated datasets with $B = 100$ re-samples and under the same hyperparameter values as in Subsection 6.1.1. Additionally, we run inference with a range of regularisation parameter values, namely, we look at $\lambda = 0.0, 0.0001, 0.001, 0.01$.

Table 6.10 contains numbers of false discoveries made by the inference technique under different values of λ. Observe that for the largest value of λ almost no false discoveries are made. A decrease in the value of the regularisation parameter seems to lead to more spurious causal relationships being inferred. For $\lambda = 0.001$, the value we use in the causal analysis of MS and sleep stage time series, on average, 14.4 false discoveries are made. Despite the fact that this result is not ideal, larger values of the regularisation parameter could be, in practice, too conservative and, thus, may lead to inferring a causal graph that is much sparser than the true structure.

Overall, this experiment demonstrates how the choice of parameter λ can mitigate spurious inference in a high-dimensional setting. Nevertheless, it needs to be chosen cautiously to avoid a complete loss of power. While the obtained numbers of false discoveries might not transfer to inference results on the 'real world' data, they suggest that some discovered causal relationships could originate from spurious correlations arising due to high dimensionality of the dataset.

Chapter 7

Discussion & Conclusions

In this thesis we presented a framework for causal time series analysis and used it to investigate the relationship between metabolites in human exhalome and sleep stage transitions. This chapter provides a brief discussion of the results of the thesis, summarises our conclusions and lists possible directions for further research.

The dataset with synchronised mass spectrometry and sleep stage labels was acquired at a very high time resolution that has never been considered in the breathomics literature before in a study of such scale. Moreover, there has been little to no systematic discussion of the association between human sleep and volatile organic compounds contained in exhaled breath [1, 38]. Thus, from the point of view of biology and biochemistry, this project is exciting and innovative at least due to its sheer scale and novelty of its research questions.

Granger causality approach that we adopted herein has seen few applications in the analysis of time course MS data [16, 76]. Building on componentwise MLPs and LSTMs, proposed in [73] for nonlinear GC estimation, we introduced our own neural network architecture. In order to account for biological variability between time series replicates and to quantify uncertainty about the inferred causal structure, we leveraged the bootstrap method. In addition, we investigated the use of time-reversed Granger causality for discovering the set of effects of the target sequence. Our method has several advantages over conventional approaches, such as correlation analysis and analysis of variance (ANOVA):

- It can represent non-additive nonlinear dependencies between sleep stage labels and multiple mass spectrometric features;
- It deals with time series in a principled way and can account for time-delayed regressive relationships;

- GC is a directed cause-effect relationship, whereas (cross-)correlation does not focus on precedence in time;
- It does not merely examine marginal relationships, it performs multiple regression.

There are also some substantial disadvantages. The power of this method comes with a range of hyperparameters that need to be tuned and computational costs of training neural networks and bootstrapping. To demonstrate that the introduced inference technique behaves as expected, we tested it on a number of synthetic datasets and also conducted simulation experiments based on the mass spectrometric and sleep stage data. The results were promising and, in general, agreed with our initial expectations.

Using the technique for inferring Granger causality based on neural networks and bootstrapping, we identified quite large sets of positive and negative ions that drive and are driven by wakefulness, NREM and REM stages. Among causal metabolites, we found isoprene, a VOC that was studied before in association with sleep [1, 38]. We observed a substantial overlap between causes and effects of the phases. This could suggest that human sleep and metabolism mutually regulate each other. There could be also other possible explanations for the feedback, such as, confounding, undersampling or presence of instantaneous causality [49]. Sets of ions discovered in this analysis provide a good starting point for a more detailed further investigation of the relationship between metabolism and sleep.

Our findings are corroborated by the results of the cross-validation of neural networks fitted for causal analysis of time series. It appears that it is possible to predict future sleep stage labels based solely on past ion intensities. Nevertheless, the performance of trained neural networks is by far not perfect, especially, for NREM stages. Therefore, inference results for this phase of sleep should be interpreted with caution.

7.1 Limitations

From the methodological perspective, key limitations of this work are associated with assumptions embedded in the definition of Granger causality. In particular, GC analysis can yield spurious conclusions if the set of considered variables is not causally sufficient. Thus, if there exist superior mechanisms that regulate both metabolism and sleep, statements of causality between ion intensities and sleep phases could be meaningless.

An important design limitation of this study is a very moderate number of subjects that the time series were acquired from. Having a larger sample would allow generalising better across the population and would improve the quality of bootstrapping results. It would also provide us with a less biased test error estimate from the cross-validation.

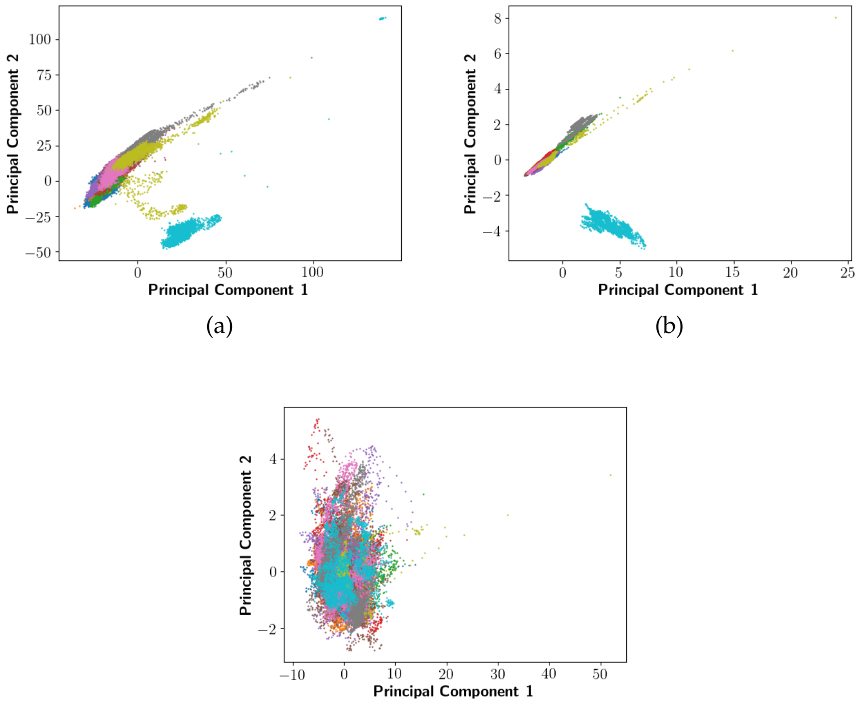
7.2 Further Research

This thesis opens many promising directions for further research with the focus on methodological and biomedical aspects of the project. The following list contains a few topics worthy of further investigation:

- The sizes of layers and depths of sub-networks in the GC-MLP architecture could have influence on the quality of causal inference. It would be interesting to investigate how altering these characteristics can affect the performance of the technique.
- The neural network architecture considered in this thesis does need to be restricted to MLP or LSTM sub-networks. We could examine the use of other modules, for example, dilated or causal convolutions [7].
- As we saw before, the choice of regularisation parameter λ is crucial in controlling the number of false discoveries. Stability selection procedure, proposed in [50], could provide a principled way for selecting the value of λ for GC-MLPs motivated by rigorous error control.
- It could desirable to understand at what time delay causal interactions between sleep stage transitions and metabolism occur. For this purpose, we could perform causal time series analysis under different model orders (K) and study how causal links vary for different horizons.

Appendix A

Visualisations

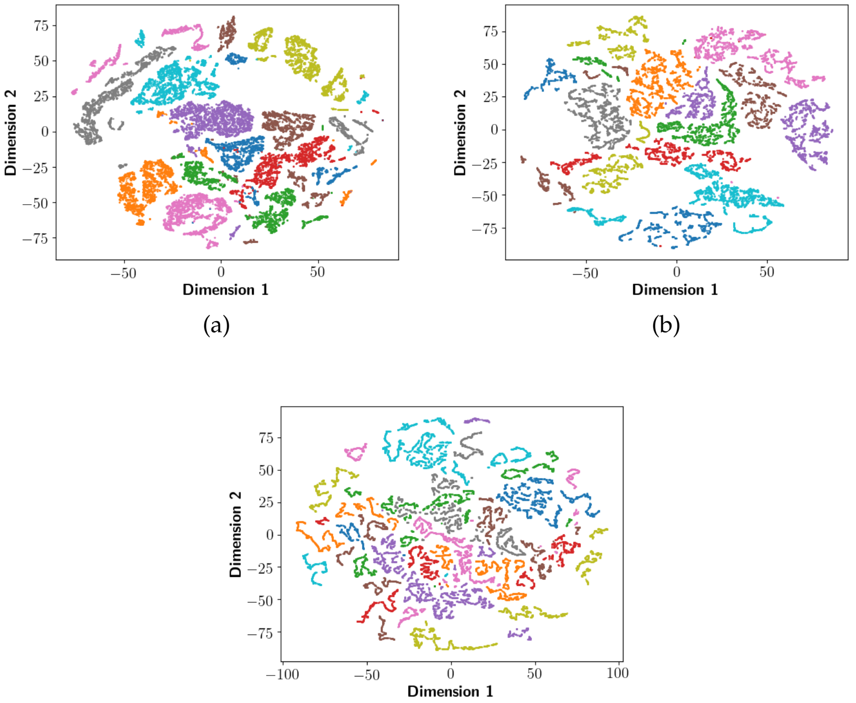


(a)

(b)

(c)

Figure A.1: First two principal components of the PCA of log-transformed positive mode MS time series. Figure A.1(a) is based on the raw data, A.1(b) shows principal components after normalisation and smoothing, and Figure A.1(c) was produced after all pre-processing steps. Data points acquired from ten different subjects are plotted in different colours.

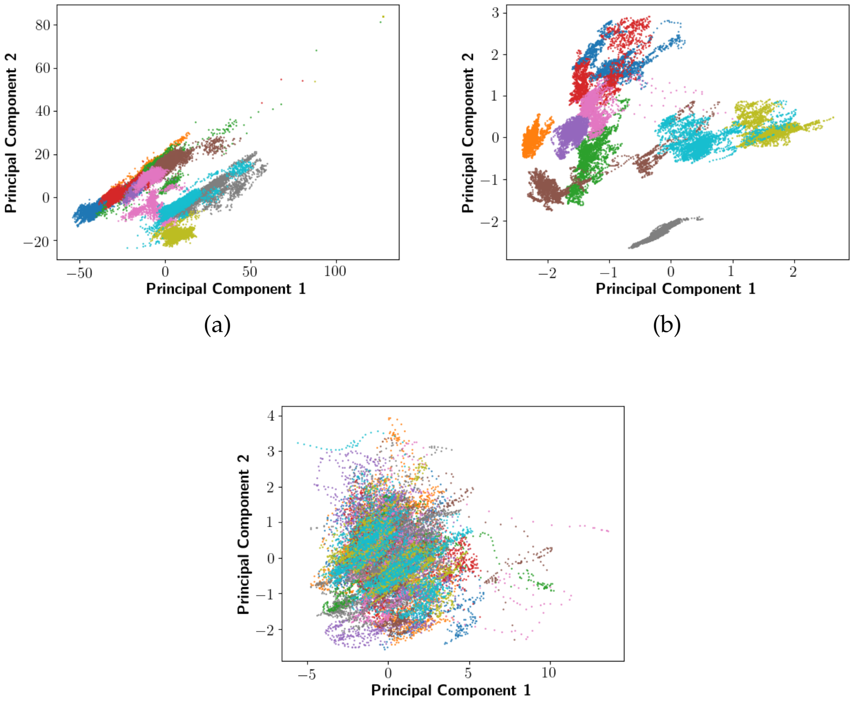


(a)

(b)

(c)

Figure A.2: Two-dimensional t-SNE representations of the negative mode MS time series. Figure A.2(a) was produced from the raw data, A.2(b) shows data after normalisation and smoothing, finally, A.2(c) depicts points after all pre-processing steps, including standardisation. Different colours correspond to ten subjects.

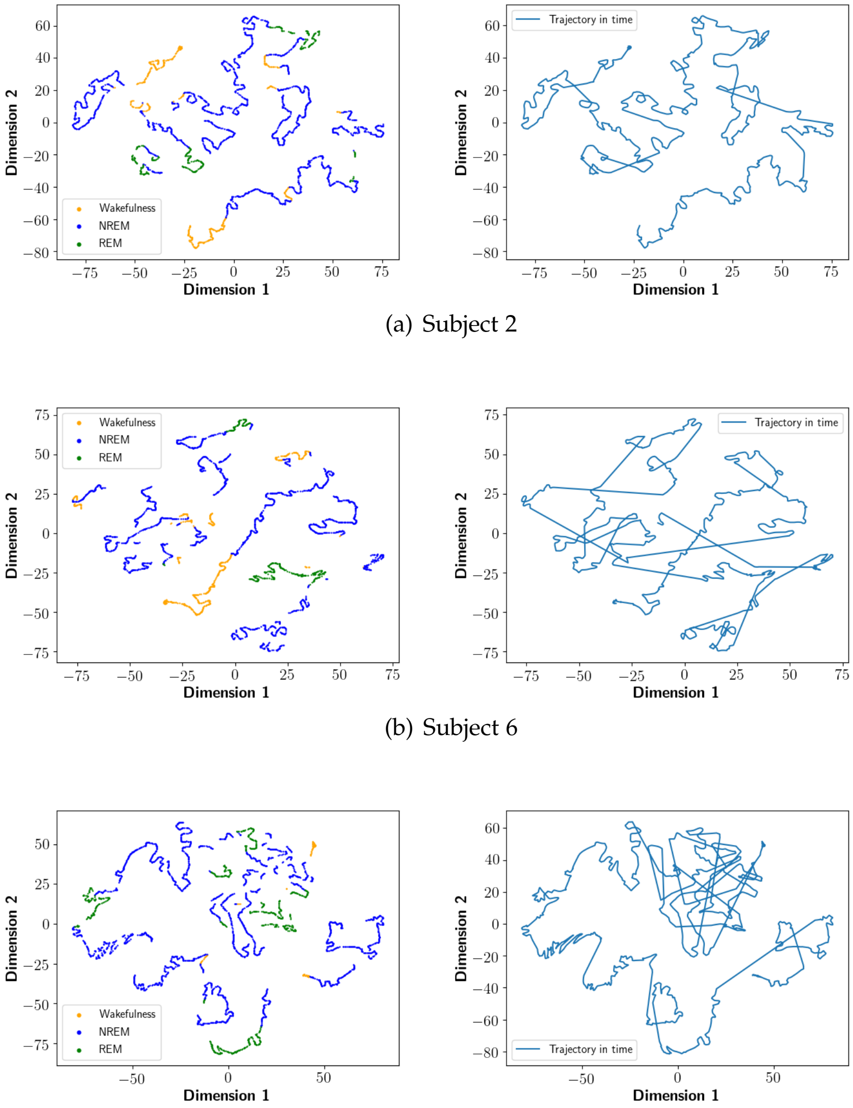


(a)

(b)

(c)

Figure A.3: First two principal components of the PCA of log-transformed negative mode MS time series. A.3(a), A.3(b) and A.3(c) are based on raw, normalised and smoothed and fully pre-processed data, respectively. Data points acquired from ten different subjects are plotted in different colours.

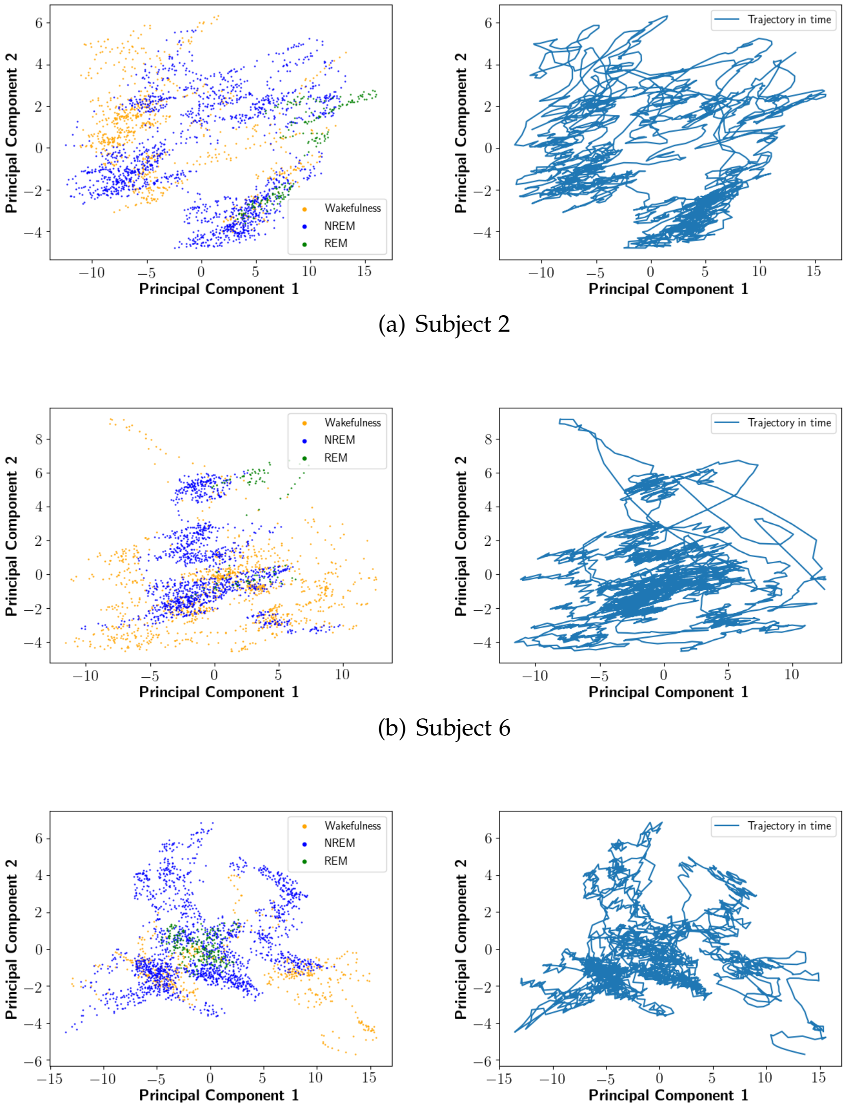


(a) Subject 2

(b) Subject 6

(c) Subject 10

Figure A.4: Two-dimensional t-SNE representations of the MS time series in the negative ion mode for three subjects. In plots on the left side, points are coloured according to their sleep stage labels: orange, blue and green colours correspond to wakefulness, NREM and REM phases, respectively. Plots on the right side contain the same t-SNE representations, however, points that are consecutive w.r.t. time are connected by line segments.

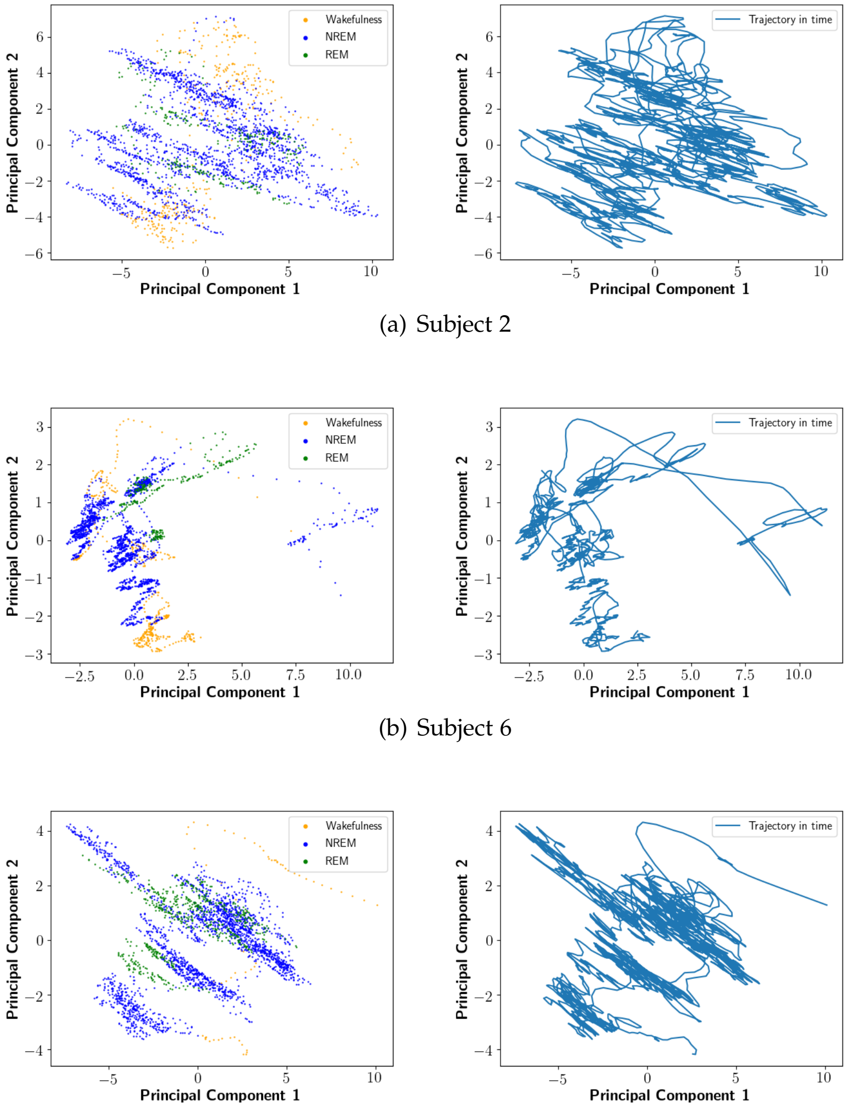


(a) Subject 2

(b) Subject 6

(c) Subject 10

Figure A.5: First two principal components of the log-transformed MS time series in the positive ion mode for three subjects. In plots on the left side, points are coloured according to their sleep stage labels: orange, blue and green colours correspond to wakefulness, NREM and REM phases, respectively. Plots on the right side contain the same PCA representations, however, points that are consecutive w.r.t. time are connected by line segments.



(a) Subject 2

(b) Subject 6

(c) Subject 10

Figure A.6: First two principal components of the log-transformed MS time series in the negative ion mode for three subjects. In plots on the left side, points are coloured according to their sleep stage labels: orange, blue and green colours correspond to wakefulness, NREM and REM phases, respectively. Plots on the right side contain the same PCA representations, however, points that are consecutive w.r.t. time are connected by line segments.

Appendix B

Model

```python
import torch.nn as nn
import numpy as np
import torch
import torch.nn.functional as F
from torch.autograd import Variable


class MLPgc(nn.Module):

    def __init__(self, num_vars, device, lag, hidden_size_1, hidden_size_2,
    num_outputs=1, dp=0.0):
        """
        Initialises an MLPgc module, which represents a neural
        network model for Granger causality estimation.

        :param num_vars: number of variables, including the response.
        :param device: device to be used for calculations, CPU or GPU.
        :param lag: order of considered regressive relationships,
        specifies the horizon in the past of predictors to be
        used to forecast the future of the response.
        :param hidden_size_1: size of layers 1 and 2 in sub-networks.
        :param hidden_size_2: size of layer 3.
        :param num_outputs: number of output units.
        :param dp: dropout rate applied to all layers, to prevent
        the co-adaptation of neurons. Default value 0.0, i.e. no dropout.
        """
        super(MLPgc, self).__init__()

        # Sub-networks
        self.layer1_list = nn.ModuleList()
        self.layer2_list = nn.ModuleList()
        for state in range(num_vars):
            layer1 = nn.Linear(lag, hidden_size_1)
            layer2 = nn.Linear(hidden_size_1, hidden_size_1)
            self.layer1_list.append(layer1)
            self.layer2_list.append(layer2)

        # Initialise weights for each variable
        self.imp_weights = nn.Parameter(torch.Tensor(np.ones((num_vars, )) /
            num_vars + np.random.normal(0, 0.00001,
                (num_vars, ))).float().to(device))

        # Final layers
        self.layer_3 = nn.Linear(hidden_size_1 * num_vars, hidden_size_2)
        self.layer_4 = nn.Linear(hidden_size_2, num_outputs)

        # Initialise the rest of the weights
        self.init_weights()

        # Save parameters
        self.num_vars = num_vars
        self.lag = lag
        self.hidden_size_1 = hidden_size_1
        self.hidden_size_2 = hidden_size_2
        self.dp = dp

    # Initialisation
    def init_weights(self):
        for m in self.modules():
            if isinstance(m, nn.Linear):
                nn.init.xavier_normal_(m.weight.data)
                m.bias.data.fill_(0.1)
            elif isinstance(m, nn.BatchNorm1d):
                m.weight.data.fill_(1)
                m.bias.data.zero_()

    # Forward propagation
    def forward(self, inputs):
        # Dimensions of inputs need to be [batch size, lag * num_vars]
        aggregated = None

        # Propagate in sub–networks
        for i in range(self.num_vars):
            layer_1 = self.layer1_list[i]
            layer_2 = self.layer2_list[i]
            inp = inputs[:, (self.lag * i):(self.lag * (i + 1))]
            tmp = F.dropout(F.relu(layer_2(F.dropout(F.relu(layer_1(inp)),
                p=self.dp, training=True))), p=self.dp, training=True)
            if i == 0:
                aggregated = self.imp_weights[i] * tmp
            else:
                aggregated = torch.cat((aggregated, self.imp_weights[i] *
                    tmp), dim=1)

        # Final two layers
        pred = self.layer_4(F.dropout(F.relu(self.layer_3(aggregated)),
            p=self.dp, training=True))

        return pred
```

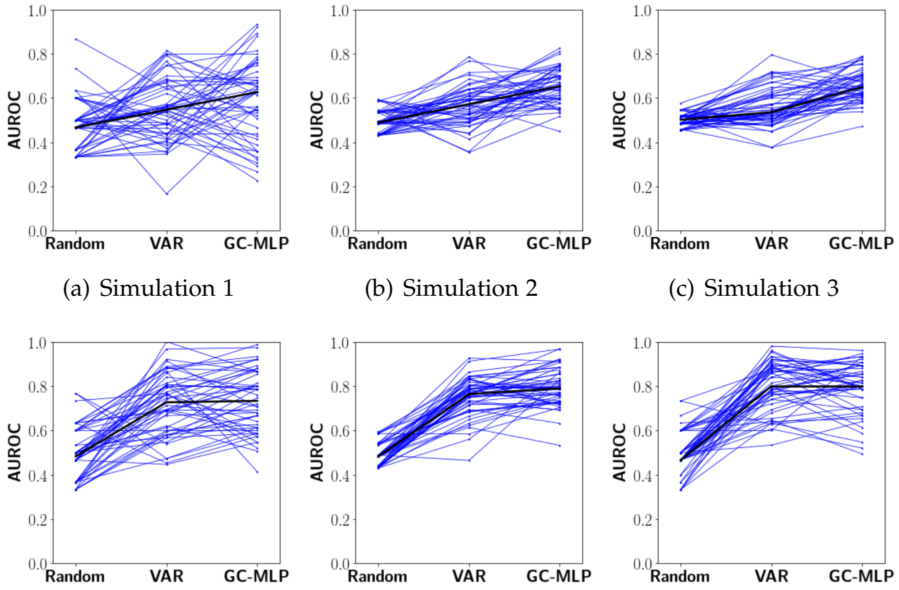
Listing B.1: Python implementation of the GC-MLP model for Granger causality estimation. The implementation is based on PyTorch machine learning library [56].

```python
import torch.nn as nn
import numpy as np
import torch
import torch.nn.functional as F
from torch.autograd import Variable


class LSTMgc(nn.Module):
    def __init__(self, num_vars, device, lag_max, hidden_size_lstm,
    hidden_size_mlp, num_outputs=1):
        """
        Initialises an LSTMgc module, which represents a neural network
        model for Granger causality estimation.

        :param num_vars: number of variables, including the response.
        :param device: device to be used for calculations, CPU or GPU.
        :param lag_max: input size for nn.LSTMCell module.
        :param hidden_size_lstm: size of hidden states in LSTMs.
        :param hidden_size_mlp: size of hidden layer in MLP.
        :param num_outputs: number of output units.
        """
        super(LSTMgc, self).__init__()

        # LSTMs
        self.lstm_cell_list = nn.ModuleList()
        for state in range(num_vars):
            self.lstm_cell_list.append(nn.LSTMCell(lag_max,
                hidden_size_lstm))

        # MLP for prediction
        self.pred_mlp_l1 = nn.Linear(hidden_size_lstm * num_vars,
            hidden_size_mlp)
        self.pred_mlp_l2 = nn.Linear(hidden_size_mlp, num_outputs)

        # Initialise weights for each variable
        self.imp_weights = nn.Parameter(torch.Tensor(np.ones((num_vars,)) /
            num_vars + np.random.normal(0, 0.00001, (num_vars,))))

        # Initialise weights
        self.init_weights()

        # Save parameters
        self.num_vars = num_vars
        self.lag = lag_max
        self.hidden_size_lstm = hidden_size_lstm
        self.hidden_size_mlp = hidden_size_mlp

        # Initialise LSTM states
        self.lstm_state_list = []
        for state in range(num_vars):
            self.lstm_state_list.append((Variable(torch.zeros(1,
                self.hidden_size_lstm).float()).to(device),
                    Variable(torch.zeros(1,
                        self.hidden_size_lstm).float()).to(device)))

    def init_weights(self):
        for m in self.modules():
            if isinstance(m, nn.Linear):
                nn.init.xavier_normal_(m.weight.data)
                m.bias.data.fill_(0.1)
            elif isinstance(m, nn.BatchNorm1d):
                m.weight.data.fill_(1); m.bias.data.zero_()

    def forward(self, inputs):
        # Input shape: [batch size, number of variables, sequence length,
        # variable dimension]

        # Concatenate LSTM hidden state vectors into one large vector,
        # which will be then used for prediction
        aggregated = []
        cnt = 0
        for state, (lstm_cell, lstm_state) in enumerate(zip(
        self.lstm_cell_list, self.lstm_state_list)):
            lstm_state = lstm_cell(
                inputs[:, state, :, :].view(inputs.shape[0], -1),
                    lstm_state)
            aggregated.append(lstm_state[1] * self.imp_weights[cnt])
            cnt += 1
        aggregated = torch.cat(aggregated, dim=1)

        # Calculate predictions
        pred = F.relu(self.pred_mlp_l1(aggregated))
        pred = self.pred_mlp_l2(pred)

        return pred
```

Listing B.2: Python implementation of the GC-LSTM model for Granger causality estimation. The implementation is based on PyTorch machine learning library [56].

Appendix C

Simulation Results



(a) Simulation 1

(b) Simulation 2

(c) Simulation 3

(d) Simulation 5

(e) Simulation 6

(f) Simulation 7

Figure C.1: Parallel coordinates plots of AUROC measures for the three methods applied to six different simulations [30]. The bold black line corresponds to the median.

Appendix D

Inference Results

D.1 Granger Causes

D.1.1 Positive Mode

The following lists contain mass-to-charge ratios of positive mode ions that were identified to Granger-cause different phases of sleep.

Wakefulness:

51.042273890874, 69.06988, 71.0491586058439, 73.0496017576813, 80.04948, 81.000451019127, 83.049295208478, 91.0562446158381, 92.057831342461, 99.0106501617184, 99.765153114363, 100.051208017493, 107.033774588533, 109.066966646674, 110.070452019458, 111.064836924014, 113.02638526304, 115.07534318436, 115.111683663421, 118.065030872907, 119.070142824166, 120.109894849693, 121.085905849639, 122.059996572494, 122.083076823652, 124.096458264414, 125.080828445829, 126.714564987143, 134.059756333431, 134.081076536013, 135.065505847178, 136.096555496559, 137.132555066338, 138.135894194824, 139.148013244918, 144.138294898655, 146.117649715453, 149.04458941481, 150.055075530994, 152.091436674958, 152.127626858015, 160.097102449594, 160.133312491838, 161.153673377034, 164.091922490765, 166.143758659002, 167.127755829265, 170.072603305602, 172.133360736814, 174.112685543623, 174.149065234836, 178.107204990319, 180.159248953508, 186.148775679665, 188.128030119628, 188.164299449449, 194.138421619266, 195.122331073215, 207.174200634127, 208.177689288382, 212.200766246293, 213.203776351641, 216.196599466429, 218.138524862507, 218.174364242184, 223.169263788512, 235.205328167707, 236.073658417814, 236.208786913016, 245.137972460505, 248.257940371348, 252.232518795873, 260.197508749868, 260.258938002409, 262.27396290581, 271.299233346565, 277.08910105104, 278.171019911663, 281.320174394414, 289.273973747315, 303.020342640102, 328.010893017507, 328.081080397556, 330.007958332551, 332.118769144975, 357.04792008628, 358.048375144379, 359.027081201626, 360.027496761511, 361.024972690575, 372.318168958412, 388.126379584427, 393.288721654869, 399.345066579663, 422.264730648397, 430.887197376519, 432.067592372852, 432.884373607377, 433.064686182754, 462.148298871756, 463.147599445635

NREM:

60.0437209383538, 65.0231132388371, 69.06988, 73.0496017576813, 79.0391621821154, 84.0567768618971, 89.0409515280725, 93.0546681401631, 94.0406148759303, 99.0804113603317, 99.765153114363, 100.06126818984, 105.05451238676, 107.033774588533, 113.02638526304, 115.07534318436, 116.078346209518, 118.065030872907, 119.070142824166, 120.109894849693, 124.096458264414, 133.122217391582, 134.059756333431, 134.081076536013, 135.065505847178, 139.148013244918, 144.138294898655, 144.146044958792, 146.117649715453, 150.055075530994, 152.127626858015, 160.133312491838, 161.153673377034, 164.11533246238, 166.049988894489, 166.143758659002, 170.072603305602, 176.606831820195, 180.159248953508, 194.138421619266, 194.56051283579, 212.200766246293, 213.203776351641, 215.127609323884, 236.073658417814, 289.273973747315, 388.126379584427, 462.148298871756, 463.132738683395

REM:

58.8684561381039, 59.0491, 60.0437209383538, 60.0517223565092, 65.0231132388371, 69.06988, 71.0491586058439, 78.0314191853689, 79.0391621821154, 79.0755636353184, 81.000451019127, 84.0567768618971, 89.0233212424452, 89.0409515280725, 99.0106501617184, 99.765153114363, 100.051208017493, 100.075788438583, 101.078905585793, 107.033774588533, 109.049306373307, 109.066966646674, 111.064836924014, 116.11468667392, 118.065030872907, 118.086241128592, 119.08554300472, 119.106713252802, 120.109894849693, 122.031806265552, 126.714564987143, 135.044035645059, 136.096555496559, 137.132555066338, 138.135894194824, 144.146044958792, 152.127626858015, 159.138101066512, 160.141162500899, 172.133360736814, 173.153663296183, 174.156785169171, 186.148775679665, 189.184800338201, 197.22638639714, 215.127609323884, 215.200417954294, 219.047289332663, 219.174277195235, 220.05056265899, 220.178510567241, 221.043296629624, 229.252219609162, 235.04204001725, 235.205328167707, 236.175357284987, 236.208786913016, 238.060996593023, 241.073804536059, 242.074524088926, 242.150253301397, 243.210012317901, 243.231712093644, 244.234651750364, 246.181131744937, 260.197508749868, 262.27396290581, 271.299233346565, 274.213374039433, 275.148637299659, 278.171019911663, 279.144330301785, 293.097720472189, 313.113784230095, 314.114409156757, 316.093449118511, 317.090293524496, 328.081080397556, 355.282902301772, 361.024972690575, 383.329153643144, 389.110097790571, 399.345066579663, 430.887197376519, 432.884373607377, 447.347205329621, 453.467186347803

D.1.2 Negative Mode

The following lists contain mass-to-charge ratios of negative mode ions that were identified to Granger-cause different phases of sleep.

Wakefulness:

60.8067402701685, 60.9936302748408, 64.01160035029, 69.0347104758678,
73.0293105757328, 75.9800906495023, 77.0242106756053, 79.002960725074,
79.9989707499743, 82.9714108242853, 84.992640874816, 85.0289708757243,
86.032160900804, 86.738600918465, 87.0082509252063, 88.711320967783,
88.747680968692, 88.9876209746905, 89.011360975284, 89.0241009756025,
90.0271810006795, 91.0274110256853, 94.998041124951, 95.9516611487915,
96.9594111739853, 97.0288111757203, 97.0651811766295, 98.0364112009103,
99.044321226108, 101.059911276498, 102.063291301582, 103.002951325074,
103.039331325983, 104.009811350245, 105.018581375465, 106.026161400654,
107.01521142538, 107.034271425857, 113.023661575592, 115.998291649957,
116.034941650874, 116.04278165107, 117.043891676097, 117.054821676371,
119.034191725855, 120.041921751048, 122.02201180055, 123.987851849696,
124.983131874578, 124.999391874985, 128.03441195086, 129.05481197637,
130.058182001455, 130.997682024942, 132.041252051031, 133.049772076244,
134.057432101436, 134.992632124816, 135.029092125727, 135.065412126635,
136.99038217476, 140.034302250858, 140.98440227461, 143.070442326761,
143.992952349824, 144.110272352757, 146.023442400586, 147.029142425729,
147.042212426055, 147.065532426638, 148.036952450924, 148.073152451829,
149.044822476121, 149.059432476486, 150.001682500042, 150.052622501316,
150.987642524691, 151.060042526501, 151.98278254957, 152.01918255048,
153.028932575723, 159.101672727542, 160.036602750915, 161.044572776114,
162.049622801241, 163.060152826504, 164.032972850824, 164.063572851589,
165.039542875989, 165.064372876609, 166.047022901176, 166.999422924986,
167.055122926378, 168.996252974906, 175.023933125598, 175.060313126508,
176.065743151644, 177.039533175988, 177.075793176895, 178.046733201168,
178.996593224915, 179.055273226382, 180.028443250711, 180.058653251466,
181.034293275857, 181.07081327677, 182.020033300501, 182.043103301078,
182.074433301861, 183.049893326247, 184.057903351448, 185.153943378849,
192.998693574967, 193.034653575866, 193.049793576245, 194.030773600769,
194.042703601068, 195.050493626262, 196.023313650583, 196.05838365146,
197.066053676651, 198.038693700967, 198.06919370173, 199.097143727429,
199.169853729246, 200.173353754334, 201.076263776907, 204.135903853398,
206.008883900222, 207.050443926261, 209.029693975742, 210.037644000941,
211.024584025615, 211.04560402614, 211.0599840265, 212.042294051057,
212.053244051331, 213.055164076379, 213.076124076903, 214.056764101419,
215.091954127299, 217.143974178599, 218.150774203769, 220.058364251459,
220.178324254458, 224.040704351018, 226.032584400815, 226.056074401402,
228.047964451199, 229.107474477687, 230.151304503783, 231.086654527166,

231.159284528982, 232.093084552327, 232.167294554182, 233.065904576648, 233.079364576984, 235.045174626129, 235.060674626517, 237.0240046756, 237.041164676029, 239.075904726898, 239.164644729116, 240.084514752113, 241.091724777293, 242.05121480128, 243.050544826264, 244.167074854177, 245.10278487757, 245.175404879385, 246.146254903656, 246.182654904566, 247.08198492705, 257.239065180977, 261.097835277446, 262.105325302633, 263.076945326924, 263.113395327835, 264.084145352104, 266.063885401597, 267.195765429894, 270.214905505373, 273.170375579259, 274.177725604443, 275.185765629644, 276.084565652114, 277.092755677319, 278.099885702497, 279.072025726801, 279.108605727715, 280.079955751999, 281.248445781211, 282.059795801495, 283.264295831607, 284.267695856692, 285.065345876634, 285.207015880175, 285.270845881771, 286.214635905366, 287.150475928762, 291.108946027724, 294.096276102407, 295.067706126693, 295.103006127575, 295.228086130702, 299.186666229667, 300.193396254835, 300.262886256572, 301.238736280968, 302.242316306058, 303.145626328641, 303.216966330424, 303.244536331113, 304.151826353796, 305.16120637903, 309.083346477084, 311.223096530577, 315.253356631334, 316.187776654694, 316.261676656542, 317.124446678111, 317.233116680828, 317.265406681635, 318.204566705114, 319.139896728497, 320.14760675369, 321.155146778879, 326.085626902141, 326.158216903955, 327.253756931344, 329.269546981739, 330.272917006823, 331.139857028496, 331.246997031175, 331.275407031885, 332.219807055495, 333.225577080639, 334.198857104971, 335.135027128376, 336.141907153548, 344.219777355494, 346.2360074059, 347.207517430188, 348.215017455375, 349.150917478773, 350.228847505721, 351.130227528256, 352.138467553462, 352.211037555276, 353.145387578635, 354.153537603838, 357.192837679821, 359.208617730215, 360.252547756314, 362.15878780397, 362.231257805781, 375.239678130992, 376.247048156176, 391.19799852995, 394.185498604637, 406.185508904638

NREM:

73.0293105757328, 75.0086306252158, 75.9800906495023, 76.0120506503013, 77.0242106756053, 79.002960725074, 79.9989707499743, 80.9744107743603, 84.0085508502138, 84.992640874816, 85.0289708757243, 86.738600918465, 87.0082509252063, 88.711320967783, 88.747680968692, 89.0241009756025, 90.0271810006795, 90.720561018014, 91.003561025089, 91.0274110256853, 91.7208110430203, 92.0065510501638, 93.0786110769653, 93.9902610997565, 94.998041124951, 95.9516611487915, 96.9594111739853, 101.059911276498, 102.018931300473, 103.002951325074, 103.039331325983, 104.009811350245, 105.018581375465, 106.026161400654, 106.997871424947, 107.034271425857, 113.023661575592, 116.009871650247, 116.04278165107, 117.043891676097, 119.034191725855, 120.041921751048, 122.980061824502, 124.00041185001, 128.03441195086, 132.041252051031, 133.049772076244, 134.057432101436, 135.029092125727, 136.037252150931, 136.99038217476, 137.023892175597, 147.065532426638, 148.036952450924, 149.044822476121, 150.052622501316, 150.987642524691, 151.060042526501, 152.01918255048, 160.024152750604, 161.044572776114, 164.032972850824, 165.039542875989, 167.055122926378, 177.039533175988, 178.013463200337, 178.046733201168, 178.982163224554, 179.055273226382, 180.028443250711, 180.058653251466, 181.07081327677, 182.043103301078, 182.074433301861, 183.049893326247, 193.034653575866, 194.042703601068, 195.050493626262, 196.05838365146, 197.066053676651, 198.038693700967, 198.06919370173, 207.050443926261, 210.037644000941, 211.04560402614, 212.053244051331, 213.055164076379, 228.047964451199, 241.054564776364, 242.05121480128, 244.167074854177, 261.097835277446, 275.185765629644, 280.079955751999, 293.088336077208, 294.096276102407, 295.067706126693, 295.103006127575, 309.083346477084, 310.090956502274, 317.233116680828, 318.204566705114, 326.085626902141, 326.158216903955, 332.219807055495, 333.225577080639

REM:

72.993040574826, 75.0086306252158, 75.9800906495023, 76.0120506503013, 79.9989707499743, 80.9744107743603, 82.9714108242853, 85.0289708757243, 86.032160900804, 86.738600918465, 87.0082509252063, 88.747680968692, 89.011360975284, 89.0241009756025, 90.0271810006795, 90.720561018014, 91.003561025089, 91.0274110256853, 91.7208110430203, 92.0065510501638, 92.9742210743555, 93.0786110769653, 94.998041124951, 102.018931300473, 103.039331325983, 106.026161400654, 106.997871424947, 107.01521142538, 107.034271425857, 111.01880152547, 113.023661575592, 115.998291649957, 116.009871650247, 116.034941650874, 117.043891676097, 121.013441775336, 122.980061824502, 123.987851849696, 124.00041185001, 124.983131874578, 124.999391874985, 125.010761875269, 125.99558189989, 127.002541925064, 128.03441195086, 133.013442075336, 135.029092125727, 135.065412126635, 136.037252150931, 139.997822249946, 140.98440227461, 140.992752274819, 143.992952349824, 147.029142425729, 149.044822476121, 150.001682500042, 150.052622501316, 151.060042526501, 152.01918255048, 160.024152750604, 162.018442800461, 164.007622850191, 165.039542875989, 167.007442925186, 169.064962976624, 170.068123001703, 177.039533175988, 178.013463200337, 178.982163224554, 178.996593224915, 180.058653251466, 181.07081327677, 182.074433301861, 189.03961347599, 190.013733500343, 190.997713524943, 191.106853527671, 193.034653575866, 195.050493626262, 195.064663626617, 196.05838365146, 197.066053676651, 198.06919370173, 199.024973725624, 202.063033801576, 205.013503875338, 206.008883900222, 209.029693975742, 210.037644000941, 212.053244051331, 219.174984229375, 220.178324254458, 221.065944276649, 225.112494377812, 226.056074401402, 226.068094401702, 227.091714427293, 227.164564429114, 229.143674478592, 230.151304503783, 231.159284528982, 232.056864551422, 232.167294554182, 239.164644729116, 241.054564776364, 241.107434777686, 241.180464779512, 242.05121480128, 244.167074854177, 245.175404879385, 255.159985129, 255.232565130814, 256.235965155899, 267.159385428985, 269.174985479375, 273.170375579259, 275.185765629644, 281.248445781211, 283.264295831607, 284.267695856692, 285.270845881771, 295.103006127575, 310.090956502274, 326.158216903955, 352.211037555276, 353.218517580463

D.2 Granger Effects

D.2.1 Positive Mode

The following lists contain mass-to-charge ratios of positive mode ions that were identified to be Granger-caused by different phases of sleep.

Wakefulness:

51.042273890874, 65.0231132388371, 69.06988, 71.0491586058439, 73.0496017576813, 75.0442286410383, 76.047528295076, 78.0314191853689, 79.0755636353184, 80.04948, 81.000451019127, 83.049295208478, 84.0567768618971, 89.0233212424452, 90.0266276072504, 91.0562446158381, 92.057831342461, 93.0546681401631, 99.0106501617184, 99.0440707359613, 100.051208017493, 100.06126818984, 107.033774588533, 109.066966646674, 110.021001269491, 110.070452019458, 111.064836924014, 117.054658510567, 118.065030872907, 119.106713252802, 120.109894849693, 121.085905849639, 124.096458264414, 125.080828445829, 128.032858072216, 134.059756333431, 134.081076536013, 135.065505847178, 136.096555496559, 137.132555066338, 138.135894194824, 139.148013244918, 144.138294898655, 146.117649715453, 148.039502970709, 148.096923348219, 149.04458941481, 150.026885365465, 152.016576294588, 152.091436674958, 152.127626858015, 160.097102449594, 160.133312491838, 163.060093400642, 164.091922490765, 166.143758659002, 167.127755829265, 170.072603305602, 172.133360736814, 174.112685543623, 174.149065234836, 178.107204990319, 180.159248953508, 186.148775679665, 186.155575561056, 188.128030119628, 188.164299449449, 190.107252660669, 192.122972838713, 194.138421619266, 200.128653626368, 201.184890599689, 204.122996506883, 212.200766246293, 213.203776351641, 216.196599466429, 218.138524862507, 223.169263788512, 228.167543674968, 231.122266187116, 232.190343241402, 236.073658417814, 236.112627983513, 243.231712093644, 245.137972460505, 248.257940371348, 252.232518795873, 260.197508749868, 261.242255898871, 262.27396290581, 266.248259677328, 271.299233346565, 274.213374039433, 275.17055690175, 278.171019911663, 279.144330301785, 281.320174394414, 289.273973747315, 303.020342640102, 304.248313159125, 314.114409156757, 316.32093101381, 328.010893017507, 330.007958332551, 332.118769144975, 357.04792008628, 358.048375144379, 359.027081201626, 360.027496761511, 361.024972690575, 372.318168958412, 388.126379584427, 393.288721654869, 399.345066579663, 422.264730648397, 430.887197376519, 432.884373607377, 462.148298871756, 463.147599445635.

NREM:

69.06988, 71.0491586058439, 73.0496017576813, 75.0442286410383, 76.047528295076, 78.0314191853689, 79.0213514685799, 79.0755636353184, 83.049295208478, 84.0567768618971, 89.0233212424452, 90.0266276072504, 93.0546681401631, 99.0106501617184, 100.051208017493, 100.06126818984, 107.033774588533, 109.066966646674, 110.021001269491, 110.070452019458, 114.0913, 115.094643438855, 118.065030872907, 119.106713252802, 120.109894849693, 121.028485209804, 122.031806265552, 124.096458264414, 125.080828445829, 134.059756333431, 134.081076536013, 135.065505847178, 136.096555496559, 144.138294898655, 146.117649715453, 148.096923348219, 152.016576294588, 152.091436674958, 152.127626858015, 160.097102449594, 160.133312491838, 164.091922490765, 166.143758659002, 167.127755829265, 170.072603305602, 174.112685543623, 174.149065234836, 176.606831820195, 180.159248953508, 186.148775679665, 188.164299449449, 194.56051283579, 212.200766246293, 213.203776351641, 215.127609323884, 216.196599466429, 228.167543674968, 236.073658417814, 237.075857332203, 258.100623696766, 275.17055690175, 281.320174394414, 289.273973747315, 303.020342640102, 357.04792008628, 359.027081201626, 360.027496761511, 361.024972690575, 388.126379584427, 462.148298871756.

REM:

58.8684561381039, 59.0491, 60.0437209383538, 60.0517223565092, 63.0436720381458, 69.06988, 71.0491586058439, 77.0600125463731, 79.0391621821154, 84.0567768618971, 89.0409515280725, 99.0106501617184, 100.051208017493, 107.033774588533, 107.070075170623, 108.761861910675, 109.049306373307, 111.064836924014, 113.02638526304, 118.065030872907, 118.086241128592, 119.08554300472, 119.106713252802, 120.109894849693, 126.714564987143, 135.044035645059, 137.132555066338, 138.135894194824, 149.071519582609, 152.127626858015, 159.138101066512, 160.141162500899, 172.133360736814, 173.153663296183, 174.156785169171, 186.148775679665, 198.11308732925, 212.200766246293, 213.203776351641, 215.127609323884, 215.200417954294, 216.098681262428, 216.131500659933, 218.174364242184, 219.047289332663, 219.174277195235, 220.05056265899, 220.178510567241, 221.043296629624, 229.252219609162, 235.04204001725, 235.205328167707, 236.175357284987, 236.208786913016, 237.057647532128, 238.060996593023, 239.054285905404, 243.210012317901, 243.231712093644, 244.234651750364, 245.247161338585, 246.250461032766, 251.185019885526, 259.242120249625, 260.197508749868, 262.27396290581, 263.276169977271, 266.248259677328, 271.299233346565, 272.245827787126, 273.257740673641, 274.213374039433, 278.171019911663, 295.211730451883, 313.113784230095, 327.252651295341, 333.285985283392, 355.282902301772, 357.29848131253, 361.024972690575, 383.329153643144, 430.887197376519, 432.884373607377, 453.467186347803.

D.2.2 Negative Mode

The following lists contain mass-to-charge ratios of negative mode ions that were identified to be Granger-caused by different phases of sleep.

Wakefulness:

60.8067402701685, 60.9936302748408, 62.9972603249315, 69.0347104758678, 71.0502605262565, 73.0293105757328, 75.9800906495023, 77.0242106756053, 79.9989707499743, 81.0342207758555, 82.9714108242853, 83.0497708262443, 84.020720850518, 84.992640874816, 85.0289708757243, 86.032160900804, 86.738600918465, 87.0082509252063, 88.711320967783, 88.747680968692, 88.9876209746905, 89.011360975284, 89.0241009756025, 90.0271810006795, 90.992841024821, 91.0274110256853, 93.9902610997565, 95.9516611487915, 96.9594111739853, 96.9687011742175, 97.0288111757203, 97.0651811766295, 98.0364112009103, 98.9657412241435, 99.044321226108, 100.015531250388, 101.059911276498, 102.063291301582, 103.039331325983, 105.018581375465, 106.026161400654, 107.01521142538, 113.023661575592, 114.027321600683, 115.998291649957, 116.009871650247, 116.034941650874, 116.04278165107, 118.997741724944, 119.034191725855, 120.041921751048, 123.987851849696, 124.983131874578, 124.999391874985, 128.03441195086, 129.018431975461, 129.05481197637, 130.058182001455, 130.997682024942, 132.029502050738, 132.041252051031, 133.049772076244, 134.057432101436, 135.029092125727, 135.065412126635, 136.99038217476, 140.98440227461, 143.070442326761, 143.992952349824, 146.023442400586, 147.029142425729, 147.042212426055, 148.024362450609, 148.036952450924, 148.073152451829, 149.044822476121, 150.001682500042, 150.052622501316, 150.987642524691, 151.060042526501, 152.01918255048, 153.028932575723, 157.08600267715, 158.089492702237, 159.101672727542, 161.008222775206, 161.044572776114, 162.049622801241, 163.023752825594, 163.060152826504, 164.063572851589, 165.039542875989, 165.053892876347, 165.064372876609, 166.047022901176, 167.007442925186, 167.055122926378, 168.996252974906, 169.024832975621, 170.992663024817, 172.023963050599, 175.060313126508, 176.065743151644, 177.039533175988, 178.013463200337, 178.996593224915, 179.055273226382, 180.058653251466, 181.034293275857, 181.07081327677, 182.020033300501, 182.043103301078, 182.074433301861, 183.049893326247, 184.057903351448, 187.038283425957, 189.0760034769, 190.013733500343, 192.050523551263, 193.034653575866, 194.030773600769, 194.042703601068, 195.050493626262, 196.05838365146, 197.066053676651, 198.06919370173, 199.097143727429, 199.169853729246, 203.091973827299, 206.008883900222, 207.050443926261, 209.029693975742, 210.037644000941, 211.04560402614, 212.042294051057, 212.053244051331, 213.055164076379, 213.185464079637, 214.056764101419, 215.091954127299, 218.150774203769, 219.174984229375, 220.178324254458, 224.040704351018, 225.112494377812, 225.18560437964, 226.032584400815, 226.056074401402, 227.028284425707, 227.038774425969, 228.047964451199, 229.050314476258, 229.107474477687, 229.143674478592, 230.051874501297, 230.151304503783, 231.086654527166, 231.159284528982, 232.093084552327, 232.167294554182, 233.065904576648, 233.079364576984, 237.0240046756, 239.040024726001, 239.164644729116, 240.084514752113, 241.054564776364, 241.091724777293, 242.05121480128, 243.050544826264, 244.167074854177, 245.10278487757, 245.175404879385, 246.146254903656, 246.182654904566, 255.232565130814, 256.235965155899, 257.175665179392, 257.239065180977, 259.191225229781, 261.097835277446, 262.105325302633, 263.076945326924, 263.113395327835, 269.211435480286, 271.191155529779, 271.226975530674, 272.230365555759, 274.177725604443, 275.185765629644, 275.21761563044, 276.084565652114, 277.092755677319, 278.099885702497, 279.072025726801, 280.079955751999, 281.248445781211, 283.191315829783, 283.264295831607, 284.267695856692, 285.207015880175, 285.270845881771, 286.214635905366, 287.150475928762, 287.222445930561, 294.096276102407, 295.067706126693, 295.103006127575, 295.228086130702, 297.243826181096, 298.247406206185, 299.186666229667, 299.259476231487, 300.229656255741, 300.262886256572, 301.202166280054, 301.238736280968, 302.209556305239, 302.242316306058, 303.145626328641, 303.216966330424, 303.244536331113, 309.083346477084, 310.090956502274, 311.223096530577, 314.246246606156, 315.253356631334, 316.261676656542, 317.124446678111, 317.265406681635, 318.204566705114, 319.139896728497, 320.14760675369, 321.155146778879, 326.085626902141, 327.253756931344, 329.232866980822, 329.269546981739, 330.272917006823, 331.275407031885, 332.219807055495, 333.119267077982, 333.225577080639, 335.135027128376, 343.212937330323, 344.219777355494, 344.252337356308, 345.264907381623, 346.2360074059, 346.267567406689, 347.171397429285, 347.207517430188, 347.242657431066, 348.215017455375, 349.150917478773, 350.228847505721, 357.192837679821, 360.252547756314, 361.258147781454, 362.15878780397, 362.231257805781, 362.261437806536, 375.203528130088, 375.239678130992, 376.247048156176, 394.185498604637, 406.185508904638.

NREM:

69.0347104758678, 73.0293105757328, 75.9800906495023, 81.0342207758555, 84.992640874816, 85.0289708757243, 86.032160900804, 87.0082509252063, 88.747680968692, 89.0241009756025, 90.0271810006795, 90.720561018014, 91.003561025089, 91.0274110256853, 91.7208110430203, 92.0065510501638, 93.0786110769653, 93.9902610997565, 95.9516611487915, 98.0364112009103, 101.059911276498, 102.018931300473, 105.018581375465, 106.026161400654, 107.034271425857, 111.044341526109, 113.023661575592, 114.67959161699, 115.039351625984, 116.009871650247, 117.018521675463, 117.043891676097, 119.034191725855, 120.041921751048, 124.00041185001, 128.03441195086, 129.05481197637, 130.058182001455, 132.041252051031, 133.013442075336, 133.049772076244, 134.057432101436, 135.029092125727, 135.065412126635, 136.037252150931, 136.99038217476, 137.023892175597, 146.023442400586, 147.029142425729, 149.044822476121, 150.052622501316, 151.060042526501, 152.01918255048, 157.08600267715, 159.101672727542, 161.044572776114, 163.060152826504, 164.032972850824, 164.063572851589, 165.039542875989, 166.047022901176, 167.055122926378, 175.060313126508, 177.039533175988, 178.046733201168, 179.055273226382, 180.028443250711, 180.058653251466, 181.034293275857, 181.059973276499, 181.07081327677, 182.043103301078, 182.074433301861, 183.049893326247, 190.08319350208, 192.050523551263, 193.034653575866, 194.042703601068, 195.050493626262, 196.05838365146, 197.066053676651, 198.038693700967, 198.06919370173, 207.050443926261, 209.029693975742, 210.037644000941, 211.04560402614, 212.053244051331, 213.055164076379, 226.032584400815, 241.091724777293, 242.05121480128, 269.211435480286, 271.226975530674, 272.230365555759, 280.079955751999, 286.214635905366, 294.096276102407, 295.067706126693, 295.103006127575, 303.216966330424, 309.083346477084, 314.246246606156, 317.233116680828, 318.204566705114, 326.085626902141, 329.232866980822, 330.240337006008, 331.246997031175, 332.219807055495, 333.225577080639, 346.2360074059, 348.215017455375.

REM:

50.0024300000607, 72.993040574826, 75.9800906495023, 79.9989707499743, 80.9744107743603, 85.0289708757243, 86.738600918465, 87.0082509252063, 88.747680968692, 89.011360975284, 89.0241009756025, 90.0271810006795, 90.720561018014, 91.003561025089, 91.0274110256853, 91.7208110430203, 92.0065510501638, 92.9742210743555, 93.0786110769653, 93.9902610997565, 94.998041124951, 102.018931300473, 106.997871424947, 107.034271425857, 113.023661575592, 117.043891676097, 120.041921751048, 121.028631775716, 122.980061824502, 123.987851849696, 124.00041185001, 124.983131874578, 125.010761875269, 127.002541925064, 129.018431975461, 129.05481197637, 131.059452026486, 133.013442075336, 135.029092125727, 135.065412126635, 136.037252150931, 139.997822249946, 143.070442326761, 149.023692475592, 149.044822476121, 150.001682500042, 150.052622501316, 151.02438252561, 151.060042526501, 152.01918255048, 157.01321267533, 161.080852777021, 162.018442800461, 165.039542875989, 167.007442925186, 169.064962976624, 170.068123001703, 177.039533175988, 178.013463200337, 178.996593224915, 179.055273226382, 180.058653251466, 181.059973276499, 181.07081327677, 182.074433301861, 189.03961347599, 190.013733500343, 191.106853527671, 193.034653575866, 194.042703601068, 195.050493626262, 196.05838365146, 197.066053676651, 198.038693700967, 198.06919370173, 202.063033801576, 203.091973827299, 206.008883900222, 209.029693975742, 210.037644000941, 212.053244051331, 213.055164076379, 221.065944276649, 223.081694327042, 225.076144376904, 225.112494377812, 226.068094401702, 230.151304503783, 231.159284528982, 237.041164676029, 239.164644729116, 241.107434777686, 243.050544826264, 244.167074854177, 245.175404879385, 255.159985129, 255.232565130814, 256.235965155899, 267.159385428985, 269.174985479375, 273.170375579259, 275.185765629644, 281.248445781211, 283.264295831607, 284.267695856692, 285.270845881771, 289.129635978241, 301.130036278251, 326.158216903955, 347.171397429285, 352.211037555276, 368.20641795516.

D.3 CV Results

Sleep Phase	**Average Accuracy(±2SD)**	
	Positive	Negative
Wake	**0.719(±0.133)**	**0.839(±0.148)**
NREM	0.532(±0.336)	0.521(±0.099)
REM	**0.792(±0.164)**	**0.759(±0.176)**

Table D.1: Average accuracy scores and standard deviations for leave-one-subject-out cross-validation of GC-MLPs for different phases of sleep obtained from positive and negative modes. Cells shown in bold correspond to mean accuracy scores that are significantly greater-than 0.5 at level $\alpha = 0.05$ (t-test p-values were adjusted using the Bonferroni method).

Bibliography

[1] A. Amann, G. Poupart, S. Telser, M. Ledochowski, A. Schmid, and S. Mechtcheriakov. Applications of breath gas analysis in medicine. *International Journal of Mass Spectrometry*, 239(2):227–233, 2004.

[2] N. Ancona, D. Marinazzo, and S. Stramaglia. Radial basis function approach to nonlinear Granger causality of time series. *Physical Review E*, 70(5), 2004.

[3] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical Granger methods. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 66–75, 2007.

[4] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Methodological)*, 57(1):289–300, 1995.

[5] L. R. Bijland, M. K. Bomers, and Y. M. Smulders. Smelling the diagnosis: a review on the use of scent in diagnosing disease. *The Netherlands Journal of Medicine*, 71(6):300–307, 2013.

[6] C. M. Bishop. Neural networks. In M. Jordan, J. Kleinberg, and B. Schölkopf, editors, *Pattern Recognition and Machine Learning*, chapter 5, pages 225–290. Springer, 2006.

[7] A. Borovykh, S. Bohte, and C. W. Oosterlee. Conditional time series forecasting with convolutional neural networks, 2017. arXiv:1703.04691.

[8] R. F. Burton. Respiration. In *Physiology by Numbers: An Encouragement to Quantitative Thinking*, chapter 5, pages 65–91. Cambridge University Press, 2000.

[9] A. J. Casson, M. Abdulaal, M. Dulabh, S. Kohli, S. Krachunov, and E. Trimble. Electroencephalogram. In *Seamless Healthcare Monitoring*, pages 45–81. Springer, 2017.

[10] Wikimedia Commons. Schematic of a typical mass spectrometer. `https://commons.wikimedia.org/wiki/File:Mass_Spectrometer_Schematic.svg`, 2008. Accessed: 22-05-2019.

[11] A. Conesa, P. Madrigal, S. Tarazona, D. Gomez-Cabrero, A. Cervera, A. McPherson, M. W. Szcześniak, D. J. Gaffney, L. L. Elo, X. Zhang, and A. Mortazavi. A survey of best practices for RNA-seq data analysis. *Genome Biology*, 17(1), January 2016.

[12] J. R. Conway, A. Lex, and N. Gehlenborg. UpSetR: An R package for the visualization of intersecting sets and their properties. *bioRxiv*, 2017.

[13] J. Davis and M. Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240. ACM, 2006.

[14] E. de Hoffmann and V. Stroobant. *Mass Spectrometry: Principles and Applications*. Wiley-Interscience, 2007.

[15] S.-O. Deininger, D. S. Cornett, R. Paape, M. Becker, C. Pineau, S. Rauser, A. Walch, and E. Wolski. Normalization in MALDI-TOF imaging datasets of proteins: practical considerations. *Analytical and Bioanalytical Chemistry*, 401(1):167–181, 2011.

[16] H. Doerfler, D. Lyon, T. Nägele, X. Sun, L. Fragner, F. Hadacek, V. Egelhofer, and W. Weckwerth. Granger causality in integrated GC–MS and LC–MS metabolomics data reveals the interface of primary and secondary metabolism. *Metabolomics*, 9(3):564–574, 2013.

[17] A. Dupre, S. Vincent, and P. A. Iaizzo. Basic ECG theory, recordings, and interpretation. In *Handbook of Cardiac Anatomy, Physiology, and Devices*, pages 191–201. Humana Press, 2005.

[18] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[19] M. Eichler. Causal inference in time series analysis. In C. Berzuini, A.P. Dawid, and L. Bernardinelli, editors, *Causality: Statistical Perspectives and Applications*, pages 327–354. Wiley, United States, 2012.

[20] C. Evans, J. Hardin, and D. M. Stoebel. Selecting between-sample RNA-seq normalization methods from the perspective of their assumptions. *Briefings in Bioinformatics*, 19(5):776–792, 2017.

[21] J. Faraway and C. Chatfield. Time series forecasting with neural networks: a comparative study using the air line data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47(2):231–250, 1998.

[22] J. Fox. Bootstrapping regression models. In *An R and S-PLUS Companion to Applied Regression: A Web Appendix to the Book*, 2002. http://statweb.stanford.edu/~tibs/sta305files/FoxOnBootingRegInR.pdf. Accessed: 26-06-2019.

[23] K. J. Friston, L. Harrison, and W. Penny. Dynamic causal modelling. *NeuroImage*, 19(4):1273–1302, 2003.

[24] M. S. Furqan and M. Y. Siyal. Random forest Granger causality for detection of effective brain connectivity using high-dimensional data. *Journal of Integrative Neuroscience*, 15(01):55–66, 2016.

[25] A. Gelman. Scaling regression inputs by dividing by two standard deviations. *Statistics in Medicine*, 27(15):2865–2873, 2008.

[26] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12:2451–2471, 1999.

[27] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256. PMLR, 2010.

[28] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. Accessed: 05-06-2019.

[29] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, August 1969.

[30] FMRIB Analysis Group. NetSim – evaluation of network modelling methods for FMRI. https://www.fmrib.ox.ac.uk/datasets/netsim/. Accessed: 01-07-2019.

[31] S. Guo, A. K. Seth, K. M. Kendrick, C. Zhou, and J. Feng. Partial Granger causality — eliminating exogenous inputs and latent variables. *Journal of Neuroscience Methods*, 172(1):79–93, 2008.

[32] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1:297–310, 1986.

[33] T. Hastie, R. Tibshirani, and J. Friedman. High-dimensional problems: $p \gg N$. In *The Elements of Statistical Learning*, chapter 18, pages 649–698. Springer, 2009.

[34] W. Heide, E. Koenig, P. Trillenberg, D. Kompf, and D. S. Zee. Electrooculography: technical standards and applications. The International Federation of Clinical Neurophysiology. *Electroencephalography and Clinical Neurophysiology. Supplement*, 52:223–240, 1999.

[35] N. E. Helwig. Bootstrap confidence intervals. University lecture: http://users.stat.umn.edu/~helwig/notes/bootci-Notes.pdf, 2017. Accessed: 26-06-2019.

[36] S. L. Ho, M. Xie, and T. N. Goh. A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction. *Computers & Industrial Engineering*, 42(2-4):371–375, 2002.

[37] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python. http://www.scipy.org/, 2001–. Accessed: 08-07-2019.

[38] J. King, A. Kupferthaler, B. Frauscher, H. Hackner, K. Unterkofler, G. Teschl, H. Hinterhuber, A. Amann, and B. Högl. Measurement of endogenous acetone and isoprene in exhaled breath during sleep. *Physiological Measurement*, 33(3):413–428, 2012.

[39] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. arXiv:1412.6980.

[40] O. Lawal, W. M. Ahmed, T. M. E. Nijsen, R. Goodacre, and S. J. Fowler. Exhaled breath analysis: a review of 'breath-taking' methods for offline analysis. *Metabolomics*, 13(10), 2017.

[41] A. Lenail. NN-SVG. http://alexlenail.me/NN-SVG/. Accessed: 06-06-2019.

[42] Y. Li, C.-Y. Chen, and W. W. Wasserman. Deep feature selection: Theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016.

[43] C. X. Ling and V. S. Sheng. Cost-sensitive learning and the class imbalance problem. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*, pages 231–235. Springer, 2010.

[44] E. N. Lorenz. Predictability: a problem partly solved. In *Seminar on Predictability*, volume 1, pages 1–18, Shinfield Park, Reading, 1995. ECMWF, ECMWF.

[45] A. C. Lozano, N. Abe, Y. Liu, and S. Rosset. Grouped graphical Granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12):i110–118, 2009.

[46] H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 2007.

[47] R. K. Malhotra and A. Y. Avidan. Sleep stages and scoring technique. In *Atlas of Sleep Medicine*, pages 77–99. Elsevier, 2014.

[48] D. Marinazzo, M. Pellicoro, and S. Stramaglia. Kernel method for nonlinear Granger causality. *Physical Review Letters*, 100(14), 2008.

[49] M. Maziarz. A review of the Granger-causality fallacy. *The Journal of Philosophical Economics: Reflections on Economic and Social Issues*, 8(2):86–105, 2015.

[50] N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[51] B. J. A. Mertens. Transformation, normalization, and batch effect in the analysis of mass spectrometry data for omics studies. In *Statistical Analysis of Proteomics, Metabolomics, and Lipidomics Data Using Mass Spectrometry*, pages 1–21. Springer, 2016.

[52] K. R. Mills. The basics of electromyography. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(suppl_2):ii32–ii35, 2005.

[53] A. Montalto, S. Stramaglia, L. Faes, G. Tessitore, R. Prevete, and D. Marinazzo. Neural networks with non-uniform embedding and explicit validation phase to assess Granger causality. *Neural Networks*, 71:159–171, 2015.

[54] S. Monti. The effect of data preprocessing on SESI-MS breath analysis studies. Master's thesis, Seminar for Statistics, ETH Zürich, 2018.

[55] C. Papagiannopoulou, D. G. Miralles, S. Decubber, M. Demuzere, N. E. C. Verhoest, W. A. Dorigo, and W. Waegeman. A non-linear Granger-causality framework to investigate climate–vegetation dynamics. *Geoscientific Model Development*, 10(5):1945–1960, 2017.

[56] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[57] S. Paul and R. N. Bhattacharya. Causality between energy consumption and economic growth in India: a note on conflicting results. *Energy Economics*, 26(6):977–983, 2004.

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[59] J. Peters, D. Janzing, and B. Schölkopf. Causal inference on time series using restricted structural equation models. In *Advances in Neural Information Processing Systems 26*, pages 154–162, 2013.

[60] J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference – Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, MA, USA, 2017.

[61] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. Statistical description of data. In *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing*, chapter 14, pages 603–649. Cambridge University Press, 2 edition, 1992.

[62] M. R. N. Ranjbar, Y. Zhao, M. G. Tadesse, Y. Wang, and H. W. Ressom. Evaluation of normalization methods for analysis of LC-MS data. In *2012 IEEE International Conference on Bioinformatics and Biomedicine Workshops*. IEEE, 2012.

[63] A. Roebroeck, E. Formisano, and R. Goebel. Mapping directed influence over the brain using Granger causality and fMRI. *NeuroImage*, 25(1):230–242, 2005.

[64] R. W. Schafer. What is a Savitzky-Golay filter? [Lecture Notes]. *IEEE Signal Processing Magazine*, 28(4):111–117, 2011.

[65] L. Schneider. Anatomy and physiology of normal sleep. In *Sleep and Neurologic Disease*, pages 1–28. Elsevier, 2017.

[66] S. Seabold and J. Perktold. Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[67] R. H. Shumway and D. S. Stoffer. State space models. In *Time Series Analysis and Its Applications*, chapter 6, pages 287–381. Springer, 4 edition, 2017.

[68] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[69] V. Sindhwani, H. Q. Minh, and A. Lozano. Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and Granger causality. In *Proceedings of the Twenty-Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pages 586–595, 2013.

[70] L. I. Smith. A tutorial on principal components analysis. Technical report, Cornell University, 2002.

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

[72] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[73] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox. Neural Granger causality for nonlinear time series, 2018. arXiv:1802.05842.

[74] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[75] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[76] L. Wang, X. Sun, J. Weiszmann, and W. Weckwerth. System-level and Granger network analysis of integrated proteomic and metabolomic dynamics identifies key points of grape berry development at the interface of primary and secondary metabolism. *Frontiers in Plant Science*, 8:1066, 2017.

[77] Y. Wang, K. Lin, Y. Qi, Q. Lian, S. Feng, Z. Wu, and G. Pan. Estimating brain connectivity with varying-length time lags using a recurrent neural network. *IEEE Transactions on Biomedical Engineering*, 65(9):1953–1963, 2018.

[78] J. W. Wei, L. J. Tafe, Y. A. Linnik, L. J. Vaickus, N. Tomita, and S. Hassanpour. Pathologist-level classification of histologic patterns on resected

lung adenocarcinoma slides with deep neural networks. *Scientific Reports*, 9(1), 2019.

[79] I. Winkler, D. Panknin, D. Bartz, K. Müller, and S. Haufe. Validity of time reversal for testing Granger causality. *IEEE Transactions on Signal Processing*, 64(11):2746–2760, 2016.

[80] Y. Wu and L. Li. Sample normalization methods in quantitative metabolomics. *Journal of Chromatography A*, 1430:80–95, 2016.

[81] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society (Series B)*, 68(1):49–67, 2006.

[82] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[83] Universität Zürich. Zurich exhalomics. https://www.hochschulmedizin.uzh.ch/en/projekte/zurich-exhalomics.html, 2018. Accessed: 26-05-2019.