Statistical Machine Learning in Biomedical Engineering

January, 2024

Autor: Alba González Cebrián

Director: D. Alberto J. Ferrer Riquelme

A mis rockstars:
mi padre, mi madre y mi hermana.

Resumen

Esta tesis, desarrollada bajo una beca de formación de personal investigador de la Universitat Politècnica de València, tiene como objetivo proponer y aplicar metodologías de *Statistical Machine Learning* en contextos de Ingeniería Biomédica. Este concepto pretende aunar el uso de modelos de aprendizaje automático junto con la búsqueda de comprensión e interpretabilidad clásica del razonamiento estadístico, dando lugar a soluciones tecnológicas de problemas biomédicos que no pasen únicamente por el objetivo de optimizar el desempeño predictivo de los modelos. Para ello, se han dibujado dos objetivos principales que vertebran además el documento: proponer metodologías novedosas dentro del paraguas del *Statistical Machine Learning*, y aplicar soluciones a problemas biomédicos reales manteniendo esta filosofía en mente. Estos objetivos se han materializado en contribuciones metodológicas para la simulación de valores atípicos y la imputación de datos faltantes en presencia de datos atípicos, y en contribuciones aplicadas a casos reales para la mejora de procesos de atención médica, la mejora en el diagnóstico y pronóstico de enfermedades, y la estandarización de procedimientos de medición en entornos biotecnológicos. Dichas contribuciones se han artículado en capítulos correspondientes a las dos partes principales ya mencionadas. Finalmente, las conclusiones y líneas futuras cierran el documento, recalcando los mensajes principales de las contribuciones de la tesis doctoral en general, y sentando además las bases para líneas futuras que se han dibujado a consecuencia del trabajo realizado a lo largo del doctorado.

Resum

Aquesta tesi, desenvolupada sota una beca de formació de personal investigador de la Universitat Politècnica de València, té com a objectiu proposar i aplicar metodologies de *Statistical Machine Learning* en contextos d'Enginyeria Biomèdica. Aquest concepte pretén unir l'ús de models d'aprenentatge automàtic juntament amb la cerca de comprensió i interpretació clàssica del raonament estadístic, donant lloc a solucions tecnològiques de problemes biomèdics que no passen únicament per l'objectiu d'optimitzar el rendiment predictiu dels models. Per a això, s'han dibuixat dos objectius principals que vertebren a més el document: proposar metodologies noves dins del paraigua del Statistical Machine Learning, i aplicar solucions a problemes biomèdics reals mantenint aquesta filosofia en ment. Aquests objectius s'han materialitzat en contribucions metodològiques per a la simulació de valors atípics i la imputació de dades mancants en presència de valors atípics, i en contribucions aplicades a casos reals per a la millora de processos d'atenció mèdica, la millora en el diagnòstic i pronòstic de malalties, i l'estandardització de procediments de mesurament en entorns biotecnològics. Aquestes contribucions s'han articulat en capítols corresponents a les dues parts principals ja esmentades. Finalment, les conclusions i línies futures tanquen el document, recalant els missatges principals de les contribucions, de la tesi doctoral en general, i assentant a més les bases per a línies futures que s'han dibuixat com a consequència del treball realitzat al llarg del doctorat.

Abstract

This thesis, developed under a research personnel formation grant from the Universitat Politècnica de València, aims to propose and apply methodologies of Statistical Machine Learning in Biomedical Engineering contexts. This concept seeks to combine machine learning models with the classic understanding and interpretability of statistical reasoning, resulting in technological solutions for biomedical problems that go beyond solely optimizing the predictive performance of models. To achieve this, two main objectives have been outlined, which also structure the document: proposing novel methodologies within the umbrella of Statistical Machine Learning and applying solutions to real biomedical problems while keeping this philosophy in mind. These objectives have materialized into methodological contributions for simulating outliers and imputing missing data in the presence of outliers and applied contributions to real cases for improving healthcare processes, enhancing disease diagnosis and prognosis, and standardizing measurement procedures in biotechnological environments. These contributions are articulated in chapters corresponding to the aforementioned two main parts. Finally, the conclusions and future lines of research conclude the document, reiterating the main messages of the contributions and the overall doctoral thesis and laying the groundwork for future lines of inquiry stemming from the work conducted throughout the doctorate.

Acknowledgements

Esta tesis clama iniciar con un incommensurable agradecimiento hacia las personas que han inspirado – incluso convenientemente forzado – un crecimiento profesional y personal que en mí era necesario. Pese al inminente fracaso que supone intentar abarcar lo incommensurable con meras palabras, trataré de no quedar en ridículo ante mis innecesariamente intensos, pero terriblemente ciertos, sentimientos de agradecimiento.

Gracias, Alberto, no sólo por ser el vivo ejemplo del rigor y del perfeccionismo, sino también por tu paciencia, por tus correcciones, por tu tiempo y por haberme dado la que considero la mejor oportunidad de mi vida. Quién me iba a decir, en la primera clase de Estadística, que el profesor que contó las firmas de asistencia, iba a terminar haciéndome saltar lágrimas de admiración, pero especialmente, de aprecio. Gracias a Francisco, Abel, Raffaele y Sonia, por vuestra colaboración en forma de revisiones, consejos e ideas. Thank you so much as well to all my colleagues from Dublin, for welcoming and supporting me during the latter stages of my thesis, specially to Horacio – gracias por tu apoyo y tu confianza. A todos vosotros, junto al grupo GIEM, y a todos los profesionales de la educación e investigación que me han animado a seguir este camino: gracias por haber tomado en serio mis sueños incluso cuando yo no he sabido hacerlo.

Gracias a todos mis colegas de doctorado, resaltando a Dani, por ser un ejemplo de trabajo inigualable; a Pedro, por tu apoyo incondicional en todas mis aventuras y por los paseos y cenas de verano; y a Sergio, por el diligente cuidado de mi mesa en el despacho. No obstante, tengo un agradecimiento

especial a dos compañeros. Gracias, Giulia, colega diurna y cómplice nocturna de confesiones, aventuras, miedos y errores, pero sobretodo de éxitos; eres hoy una de mis mejores amigas. Y, especialmente, gracias a Joan, por ser un apoyo infalible, haciéndome sentir válida, escuchada y entendida todos y cada uno de los días del doctorado, y sobretodo por saber encontrarme la risa entre cafés. Hacer el doctorado ha merecido la pena aunque sea solo por haberos conocido a ambos. Os quiero y os admiro.

Gracias a mis amigas y amigos por guardarme siempre el sitio y por ser mi vía de escape y mi apoyo cuando lo he necesitado. Gracias a Paula y Ana por los cafés planificados con semanas de antelación y las cervezas improvisadas en una hora. Gracias a mis amigos de Abengibre por ser la familia escogida y mi religión. Gracias a mis amigas de la UPV por hacer del estudio un hábito del que no quiero desprenderme nunca, sois ejemplares. Gracias a todos los amigos que hice en iGEM, una experiencia definitiva en mi apuesta por la investigación. Gracias, especialmente, a Manu y a Dani por ser increíblemente pacientes, cariñosos, sinceros y por creer en mí desde siempre. Gracias a Ana y Hind por ser mis hermanas mayores cuando más lo necesitaba, espero que estéis orgullosas de mí. Gracias a mis amigos de València, me siento la persona más afortunada del mundo por ese huequecito entre personas tan maravillosas. Y gracias a mis amigos de Dublín, que han sido una sorpresa de la que todavía no doy crédito. Gracias especialmente a Héctor, por quererme en toda mi profundidad, por tu confianza y por cada segundo que hemos pasado juntos, con mención especial a los balconeos con reflexiones de vida tan estériles como ciertas. E incontestablemente, gracias a Ana Cruz por reencontarnos siempre, estemos dónde estemos y sea cómo sea, por seguir queriéndonos y creyendo la una en la otra en todos nuestros momentos; en resumen, gracias por ser mi mejor amiga. Es un honor teneros en mi vida.

Gracias, finalmente, a mis más antiguas amistades: mi familia. Gracias por la paciencia que habéis tenido conmigo en mis días mudos de deprimente atropello melancólico y en mis huidas hacia delante por la puerta de atrás. Incluyo en este grupo a Cristina, ya que con tu trabajo has puesto el mundo a mis pies cuando sentía que lo tenía totalmente encima. Papá y mamá, gracias por enseñarme a vivir volando libre. Y a Ángela, gracias por tu fe en mí; siempre has sido la luz de todos mis pozos. Ya solo me queda decir que espero que el trabajo esté a vuestra altura. Os quiero.

Contents

List of Figures

List of Tables

Part I

Prologue

Chapter 1

Justification, Objectives and Contributions

1.1 SynBioFactory and SynBioControl projects

The present thesis has been developed with funding from a research personnel formation (FPI) grant subprogram I from the Universitat Politècnica de València (PAID17-I) from 2018 to 2021. At the beginning of 2018, this grant was related for several months to the project *Synthetic biology for bioproduction enhancement: Design, optimisation, monitoring and control* (acronym *SynBioFactory*, code DPI2014-55276-C5-1-R-AR). Then it continued with the project *Design, characterisation and optimal tuning of synthetic biocircuits for bioproduction with control of the metabolic load* (acronym *SynBioControl*, code DPI2017-82896-C2-1-R-ARR), which lasted between 2018 and 2021.

The team involved in SynBioFactory integrated five research groups from academia and a biotech company, bringing in the expertise from two systems and control engineering groups specialised in bioprocess modelling and control, and systems and synthetic biology (GCSC-UPV, ML-UdG), a chemical engineering group with experience in bioprocess optimisation and model-building in systems biology (IIM-CSIC); two applied statistics groups with expertise in multivariate statistical tools for modelling, monitoring, process scaling-up, mega-data analysis in −omics, and meta-heuristic optimisation and decision making (GIEM-UPV, UPCT-UMU), and BiopolisS.L., a leader Spanish biotech company providing R&D and production services for the agrifood, pharmaceutical, chemical and energy sectors.

The technical purpose of the SynBioFactory project was to apply Synthetic Biology (SynBio) for bioproduction enhancement, emphasising engineering design methods' role in exploiting optimisation, monitoring and feedback control. Its broader goal was to help SynBio to become an engineering discipline emphasising engineering principles and methodology in designing, constructing and characterising biological systems used for genetic engineering research. Within this framework, SynBioFactory targeted two practical problems from the bioprocess industry that aim to understand and drive the microorganism to the states maximising yield and productivity:

- Goal 1: Develop efficient production systems for protein synthesis and expression, emphasising control of protein expression variability and host-circuit interaction.
- Goal 2: Rational design and optimisation of synthetic pathways for synthesising commodities, emphasising methods and circuits to drive metabolic fluxes to maximise yield and productivity and manage the metabolic burden.

Moreover, two other goals were related to the relevance of methodological aspects concerning the availability of biological parts (biobricks), biological devices, and software tools to decouple design from implementation. Therefore, in transversal to the goals above, the following ones expressed their materialisation into applied contributions:

- Goal 3: Fostering Synthetic Biology to become engineering by making designing more systematic (standardised), modular, predictable, robust, scalable, and efficient.
- Goal 4: Implement software methods and biobricks on an open-source, public-access basis.

Methods from mathematical optimisation, systems engineering and control, and multivariate statistics were used to achieve these goals. These methods were the tools that, coupled with metabolic engineering and DNA synthesis and assembly, allowed the proposal of proper solutions to the described challenges and goals.

This research line continued with the SynBioControl project, which integrated two groups the SB2CLab (former GCSC-UPV), the GIEM-UPV group, the IIM-CSIC group and the company Biopolis S.L. as well.

The SynBioControl project tackled the integration of more complex synthetic genetic circuits to produce proteins and metabolites of industrial importance. Increasing the complexity of the synthetic genetic circuits also meant increasing the metabolic and genetic load of the cell, resulting in altered dynamics caused by the interplay of shared resources. Targeting these phenomena required, as part of SynBioControl, to design and implement feedback control mechanisms of protein and metabolite expression, considering the effects of metabolic and genetic load. Two methodological goals tackle this general goal:

- Goal 1: Development of structural design methods, analysis and robust parametric tuning of synthetic control genetic circuits through multiobjective optimization.
- Goal 2: Development of data analytics methods and grey models with application to scaling up from the laboratory to the pre-industrial bioreactor.

Additionally, two objectives will be considered transversal to the previous ones:

- Goal 3: Contribution to the conversion of Synthetic Biology into engineering, making the modelling and design process more systematic (standardized), modular, predictable, and robust, with an emphasis on the development of methodologies that can be applied effectively in the practical environment of a standard industrial biotechnology laboratory.
- Goal 4: Implement open source software tools and devices or biological parts (*biobricks*) of public access, facilitating the dissemination of Synthetic Biology as an engineering area.

1.2 Objectives of this thesis

This section describes the objectives of this thesis. The core objectives of this thesis are:

1. Propose, implement and deploy new methodologies for Statistical Machine Learning, and
2. Apply existing and novel Statistical Machine Learning techniques to real Biomedical Engineering problems.

At this point, I felt the need to add a personal commentary addressing the leap from the Biotechnological field to the Biomedical Engineering one. Despite the biotechnological focus of SynBioFactory and SynBioControl projects, many other areas share the need to use mathematical tools for studying biological and medical problems. Given my academic background and interest in Biomedical Engineering, and under the agreement of my supervisor and all researchers involved in the abovementioned projects, I found room for setting Biomedical Engineering as the applied context for the methodological proposals described throughout this thesis, embedding the contributions for Synthetic Biology as part of this broad discipline.

The following points link the core objectives of the thesis with the previous projects, specifically concerning the use of model reduction for analysing biological systems, the proposal of solutions for the systematisation and standardisation of Synthetic Biology, and applying these methods to real datasets, providing tools for data understanding.

Objective 1: Propose, implement and deploy new methodologies for statistical machine learning

Regarding Objective 1, two main contributions are related to different points of applying statistical machine learning. Chapter 5 contains a proposal to simulate outliers using the framework of a Principal Component Analysis (PCA, more in Section 3.3.1) model, i.e., associating outliers with large values of the Squared Prediction Error (SPE) and the Hotelling's T^2. Using this pair of statistics to describe observations provides meaningful criteria to define outliers, controlling not only the type of simulated outliers but also how far these outliers will be from the reference data set.

Chapter 6 will propose a new algorithm to impute missing data when matrices might present as well cellwise, single rowwise or grouped rowwise outliers. This work has two main contributions. The main one is the proposal of a new algorithm that imputes missing data, detects and corrects cellwise outliers, detects rowwise outliers and imputes if they are forming a cluster. This algorithm, named RadarTSR, introduces conceptual novelty by splitting the rowwise outliers category into single rowwise outliers (not creating a cluster) and grouped rowwise outliers (forming a cluster). This distinction is the scaffold for a step that extends the existing work on dealing with missing data and outliers, whose furthest point reached was the missing data imputation, the detection of rowwise outliers and the detection and correction of cellwise outliers. Moreover, to simulate the single rowwise outliers, the framework from Chapter 5 was used.

Objective 2: Apply existing and novel statistical machine learning techniques to real biomedical engineering problems

The content related to this second objective is structured in different chapters, each one of them associated with a different context of biomedical engineering, defined by a problem and a provided solution employing statistical machine learning approaches.

Chapter 7 proposes the use of latent variable-based multivariate statistical techniques, such as Partial Least Squares regression (PLS, Section 3.4.1), in the Six Sigma statistical toolkit for healthcare processes improvement, illustrating their implementation into the DMAIC (Define, Measure, Analyze, Improve and Control) phases of a Six Sigma project carried out in an Outpatient Pharmaceutical Care Unit in the Department of Pharmacy at Hospital Universitario Doctor Peset in Valencia (Spain). This unit provides prescription drugs and

pharmaceutical care services to outpatients. The outcomes of the multivariate Six Sigma approach will be compared with the conclusions obtained by classical Six Sigma statistical tools, such as the ANalysis Of Variance (ANOVA).

Chapter 8 uses PLS-DA approaches to classify individuals in the healthy control or case groups of Myalgic Encephalomyelitis/Chronic Fatigue Syndrome (ME/CFS), also determining which variables could be used as potential biomarkers, holding the most discriminant power between these two classes of participants. In this context, the performance of Partial Least Squares for Discriminant Analysis (PLS-DA, Section 3.4.1) models is assessed in terms of performance, including a comparison with other machine learning techniques (Support Vector Machines, Random Forest and Linear Discriminant Analysis) and their interpretability. This chapter illustrates how this characteristic of latent variable-based models, including both capabilities of prediction and interpretation, directly affects the applicability of predictive models, reducing the number of necessary variables and providing insight into the physiopathology underlying different health conditions.

Chapter 9 contains a work whose goal was to predict the severity level of the disease in a COVID-19 patient admitted to the hospital as early and accurately as possible. This meant identifying the contributing factors of mortality, and developing an easy-to-use score that could quickly assess the mortality risk using only information recorded during the hospitalization event. Multiple machine learning algorithms were developed to predict mortality, and the information about these classifiers' performance was used to determine the most influential factors in predicting mortality. This information was used afterwards to define a mortality score that could be easily calculated using minimal mortality predictors while yielding accurate patient severity status estimates.

Chapter 10 is directly related to the SynBioFactory and SynBioControl projects, and it proposes a model to transform fluorescence measurements, expressed in arbitrary units and relative to the plate reader setting, to units of calibrant concentration, which are absolute, comparable, and independent of the measurement device setup. To achieve this independence from the device parameters (namely from the gain parameter), we propose a correction of the fluorescence readings by using a gain-effect model. To address the problem of the arbitrariness of units, we used already established protocols with calibrants that can be used to produce precise estimates of molecules equivalent to fluorescein (MEFL) from fluorescence measurements, deriving fluorescein concentration values. The resulting unit calibration model enabled users of fluorescence plate readers to bring experimental measurements into a common

gain-independent domain, which means a step towards the standardization of data and the comparability of results obtained from different plate readers.

1.3 Contributions

1.3.1 Articles in peer-reviewed journals

González-Cebrián, A., Arteaga, F., Folch-Fortuny, A. & Ferrer, A. How to Simulate Outliers with the Desired Properties. *Chemometrics And Intelligent Laboratory Systems.* **212** (2021), `https://doi.org/10.1016/j.chemolab.2021.104301`.

González-Cebrián, A., Almenar-Pérez, E., Xu, J., Yu, T., Huang, W., Giménez-Orenga, K., Hutchinson, S., Lodge, T., Nathanson, L., Morten, K., Ferrer, A. & Oltra, E. Diagnosis of Myalgic Encephalomyelitis/Chronic Fatigue Syndrome With Partial Least Squares Discriminant Analysis: Relevance of Blood Extracellular Vesicles. *Frontiers In Medicine.* **9** (2022), `https://www.frontiersin.org/articles/10.3389/fmed.2022.842991`.

González-Cebrián, A., Hermenegildo, M., Climente, M. & Ferrer, A. Multivariate Six Sigma: A case study in an outpatient pharmaceutical care unit. *Quality Engineering.* **34** (2022), `https://doi.org/10.1080/08982112.2022.2042018`.

González-Cebrián, A., Borràs-Ferrís, J., Ordovás-Baines, J.P., Hermenegildo-Caudevilla, M., Climente-Marti, M., Tarazona, S., Vitale, R., Palací-López, D., Sierra-Sánchez, J.F., Saez de la Fuente, J. & Ferrer, A. Machine-learning-derived predictive score for early estimation of COVID-19 mortality risk in hospitalized patients. *PLOS ONE.* **17** (2022), `https://doi.org/10.1371/journal.pone.0274171`.

González-Cebrián, A., Borràs-FerrFerrís, J., Boada, Y., Vignoni, A., Ferrer, A. & Picó, J. PLATERO: A Calibration Protocol for Plate Reader Green Fluorescence Measurements. *Frontiers in Bioengineering and Biotechnology.* **11** (2023), `https://doi.org/10.3389/fbioe.2023.1104445`.

González-Cebrián, Folch-Fortuny, A., Arteaga F. & Ferrer, A. RadarTSR: A New Algorithm for Cellwise and Rowwise Outlier Detection and Missing Data Imputation. *Chemometrics And Intelligent Laboratory Systems.* **247** (2023), `https://doi.org/10.1016/j.chemolab.2023.105047`

1.3.2 Conference contributions

González-Cebrián, A., Arteaga, F., Folch-Fortuny, A. & Ferrer, A. Adaptation of Trimmed Scores Regression to deal with outliers in model building. *19th Annual Conference Of The European Network For Business And Industrial Statistics (ENBIS 2022).* **19** 66-66 (2019)

González-Cebrián, A., Arteaga, F., Folch-Fortuny, A. & Ferrer, A. Adaptation of Trimmed Scores Regression to deal with outliers in model building. *16th Scandinavian Symposium On Chemometrics (SSC16).* **16** 45-46 (2019)

González-Cebrián, A., Arteaga, F., Folch-Fortuny, A. & Ferrer, A. RadarTSR: Robust Adaptation for Datasets with Anomalous Rows and cells of Trimmed Scores Regression. *XXXIX Congreso Nacional de Estadística e Investigación Operativa.* **39** 169 (2022)

Borràs-Ferrís, J. González-Cebrián, A., Martínez-Minaya, J., Palací-López, D., Ferrer, A. Statistical Machine Learning for defining the Design Space. *22nd Annual Conference Of The European Network For Business And Industrial Statistics (ENBIS 2022).* **22** (2022)

1.3.3 Software

González-Cebrián CRAN - Package SCOUTer. (CRAN,2020), `https://cran.r-project.org/package=SCOUTer`.

González-Cebrián PLATERO: A Plate Reader Calibration Protocol to work with different instrument gains and asses measurement uncertainty. (GitHub,2020), `https://github.com/sb2cl/PLATERO.git`.

González-Cebrián RadarTSR: Robust Adaptation for Anomalous Rows and cells of Trimmed Scores Regression. (GitHub,2020), `https://github.com/albagc/RadarTSR-matlab-master.git`.

Chapter 2

On biomedical engineering

2.1 Introduction

This chapter will introduce the scientific domain of Biomedical Engineering (BME). First, a general description of the discipline will be provided, followed by a critical commentary on its future concerning the Industry 4.0 and Healthcare 4.0 paradigms. Afterwards, two main sections will address in more detail two branches of BME related to the contributions of this thesis: Synthetic Biology and Biomedical Informatics.

> *Biomedical engineering (BME) is an interdisciplinary area of science that applies engineering principles and tools to understand, modify or control biological systems.*

At its beginnings in the 1950s and 1960s, BME's main concern was the security of the use of medical devices. Since then, the area evolved by reaching many other scenarios where engineering methodologies are applied to solve problems defined by medical or biological constraints and requirements [1], [2]. This has led to new areas whose etymology reflects this symbiotic essence: Bioinstrumentation, Bioinformatics, Biomaterials, Biomechanics, Genetic Engineering, Clinical Engineering, Bionanotechnology, etc.

Whereas areas of BME, either with a biological or medical application, are boundlessly branching over time, they are all connected to the engineering and mathematical core of BME. This methodological trunk has remained a channel of innovation from Information and Technology (IT) disciplines. For instance, the increase in computational power materialised in better equipment and tools for medical imaging, an unseen abundance of data, and automated biomedical data analysis systems. On this last matter, the role of Machine Learning (ML) models and Artificial Intelligence (AI) has become crucial, but these concepts are explored in more depth in Chapter 3.

This interdisciplinary effort on deploying fast, automated, and data-based solutions puts BME as a particular case within the context of Industry 4.0, a term coined at the end of 2011 by a German government article defining the technological strategy for 2020 [3]. This strategic plan embraced the ideas of the Fourth Industrial Revolution, characterised by accelerated changes in technology, industries, and societal patterns fostered by the increasing interconnection and automation of processes.

Undoubtedly, the Industry 4.0 scenario has impacted BME, resulting in the fourth "major update" of healthcare: Medicine 4.0 or Healthcare 4.0. Similar to Industry 4.0, Healthcare 4.0 is characterised by using wearable, intelligent sensors and medical devices integrated with cloud computing, big data analysis, AI, and decision support techniques [4].

A key feature of the 4.0 paradigm is the transcendence beyond the traditional monogamy between a dataset, a process, and a specific scientific domain. Today's outlook is characterised by continuous permeation between the physical, digital, and biological worlds: the same dataset can be shared and fed to processes and analyses for various purposes, or datasets might be merged to expand the frontiers of certain studies. Three main concepts articulate such an ecosystem [5]:

1. the Internet of Things, as an infrastructure of interconnected devices generating a continuous streamline of data;
2. the Cloud, as the warehouse for this massively produced data; and
3. the Big Data technologies, as the tools to extract value from the large volumes of heterogeneous data.

This thesis is mainly related to the third item, proposing tools to extract information from biomedical data, often called the "smart" feature of Healthcare 4.0. This characteristic is brought by using AI and ML models and pursuing individualised and patient-centred healthcare management. However, as good as it seems, this paradigm presents several technological and social challenges further discussed in this thesis.

The effects of the 4.0 paradigm come across all levels of granularity within BME applications, reflected in the different scales at which contributions presented in this thesis affect. The cell scale considers models and tools for controlling compounds using cells as bioreactors. Systems Biology studies this first level. The tissue level, although not included in the contributions of this thesis, has the collective of cells as its reference unit. The next levels are the individual level, considering the interaction with each patient for a specific clinical casuistic, and the organisational level, which includes institutions for healthcare and public health policymaking.

The rest of this chapter will describe the two main branches of BME related to the contributions included in Part III of this thesis. On the one hand, Chapter 10 is devoted to process understanding at the cell scale in the context

of biological processes, a part of Systems Biology, described in Section 2.2. The rest of the applied contributions in the thesis could be included as examples of research in the field of Biomedical Informatics, described in Section 2.3, both at an individual level (Chapters 8 and 9) and at an organisational level (Chapter 7).

2.2 Systems biology

In 1957, Francis Crick stated one of the basic notions about living organisms, coined "the central dogma of molecular biology" [6], [7]. This lemma described the flow of genetic expression by claiming that: "the transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein may be possible, but transfer from protein to protein, or from protein to nucleic acid is impossible. Information means the precise sequence determination, either of bases in the nucleic acid or amino acid residues in the protein". This statement suggests a unidirectional path of information codified in genes (Deoxyribonucleic acid, DNA) which is transcribed to Ribonucleic acid (RNA) and then translated to proteins. Over time, the described pathway has been modified with additional flows, such as the replication of RNA molecules or the reverse transcription of RNA-yielding DNA.

The ultimate goal of Systems Biology is to analyse, describe, predict and control such biological information across all its different layers of expression; to do so, it uses mathematical modelling and statistics [8]. It pursues understanding all layers in the studied biological processes within organisms, integrating information related to different omics sciences: genomics, transcriptomics, proteomics, metabolomics and fluxomics.

The first two omic sciences refer to different chain levels described by the central dogma of molecular biology:

- *Genomics* aims to identify an organism's genome, establishing the link between the structure and function of genes. A frequent type of data to infer information about genetic expression is fluorescence data. Chapter 10 uses fluorescence data emitted by fluorescein, a fluorescent compound with a spectral pattern similar to the one of Green Fluorescent Protein (GFP, [9]). The use of GFP is widespread, especially in Synthetic Biology, a discipline described in more depth in Section 2.2.1.
- The genetic information codified as DNA is transcribed to RNA. There are different types of RNA depending on their function (messenger RNAs,

transfer RNA, etc.), but they are generally referred to as "transcripts". *Transcriptomics* looks at this information by studying these RNA molecules produced by the transcription of those genes. This perspective helps to infer the processes in which genes will be involved via their assembled transcripts. Chapter 8 contains an analysis of the presence of micro RNA (miRNA) sequence searching for potential biomarkers but also to infer the network of interactions between genes. Section 2.2.2 describes the essays to quantify miRNA expression and conduct network analysis in more depth.

Beyond the translation of RNA into a protein, three different omics are studying the structure and function of biochemical compounds and phenomena. *Proteomics* examines the structure and function of proteins and their interactions. More generally, *metabolomics* identifies metabolites, i.e., biochemical molecules and compounds involved in biochemical reactions that will use or produce them. The chains of these reactions resulting in a flux of metabolites are studied by *fluxomics*. Section 2.2.3 gives the basic information about Raman spectroscopy, a technique widely used to identify the presence of specific metabolites from the chemical fingerprint of biological samples.

The following subsections will address in more depth three types of data related to different omic levels that will be used in some Chapters of this thesis. Some context about the goal and experiments associated with acquiring these data will also be provided.

2.2.1 Quantifying genetic expression in Synthetic Biology

Bridging industry and biology, Synthetic Biology is a scientific area that involves redesigning organisms genetically to dote them with new abilities to carry out valuable purposes. These tasks include the production of compounds of interest, such as a medicine or fuel, or gaining a unique ability, such as acting as sensors for certain biochemicals or specific environmental conditions [10], [11]. This industrially oriented biotechnology is also named "white biotechnology", referring to the industrial use of microorganisms as the resources to produce several biochemical compounds. This fusion between industry and biology, which resonates with the Industry 4.0 paradigm, has its epitome in a concept envisioning cells as small factories: the *biofactories*.

Like in any other manufacturing industry, the success of biofactories as standard methods of production is intrinsically determined by the prosperity of Synthetic Biology in the construction of new organisms with the necessary

functions to adapt to production demands. Yet, when classical schemes of industrial production for measuring, monitoring, modelling, and control have to be applied to microorganisms, technical challenges are presented.

Enhancing Synthetic Biology for its transition from a trial-and-error process to an engineering discipline embracing more formal methods requires standards. These facilitate the Design-Build-Test-Learn (DBTL) lifecycle by enabling the integration of inherently different tools and techniques into coherent workflows. The DBTL cycle requires a complete description of the components in a biological system, data to describe the system function and interconnections, and computational models to predict the impact of environmental parameters on the system's behaviour. In this context, data standards describing genetic constructs and their mathematical models foster information sharing, which is crucial to overcoming characterization and reproducibility issues across laboratories.

A common source of information in Synthetic Biology is fluorescent signals emitted by cells through the expression of fluorescent reporters. This information is commonly used for quantifying gene expression levels and a wide range of other biological properties. In these settings, a measure of the light emitted by a specific fluorescent molecule, e.g. the Green Fluorescent Protein (GFP), is used to estimate the amount of GFP molecules expressed by the cell. Thus, by linking the expression of a gene of interest to that of GFP, fluorescence measurement can be used to measure the expression level of the first one indirectly.

Currently, there are two main devices for measuring fluorescence: flow cytometers and plate readers, with the latter being the most affordable option. However, these measurement techniques still face some challenges, mainly related to the lack of standard frameworks to normalize the fluorescence data, remove any effects associated with the measuring setup, and retrieve information about the genetic activity.

Improvements in this matter can be divided into two main categories. On the one hand, some studies focused more on autofluorescence correction by normalization of fluorescence measurements with fluorescence-free cell cultures grown simultaneously in parallel with the fluorescence-loaded ones [12]. On the other hand, other strategies try to use mathematical expressions to normalize the data. Most of these approaches acknowledge autofluorescence's effect, the gain used for the measurements, and the plate reader. Some examples are the FlopR software by [13] and the software FlowCal from [14].

Nevertheless, despite the attempts to address autofluorescence and normalization, these approaches present limitations. Approaches based on parallel fluorescence-free cell cultures [12] require more resources, limiting their scalability and practical implementation across diverse experimental setups. Similarly, solutions based on mathematical expressions utilized for data normalization [13], [14] encounter hurdles in accommodating the dynamic complexities inherent in varied biological systems and, while acknowledging autofluorescence and instrument-related gains, struggle to encapsulate the nuanced interplay of factors affecting the variability of fluorescence measurements.

Given this casuistic, it seems reasonable to leverage statistical tools to elevate the state-of-the-art and proffer novel innovations. Such questions are further discussed in Chapter 10, which includes a calibration protocol to fit a normalization model based on a unified mathematical framework and using a set of statistical tools that provide validation for the underlying mathematical assumptions, a quantification of the uncertainty within the predictions, and an assessment on the plate reader‘s measuring quality.

2.2.2 Quantification of miRNAs

Around 90% of genomic information in eukaryotes is transcribed to RNA, from which 98% approximately will not be translated to a protein. This vast amount of RNA sequences is named non-coding RNAs (ncRNAs). Among ncRNAs, microRNAs (miRNAs) have received considerable attention [15].

Discovered in 1993, fragments of miRNAs are single-stranded chains of approximately 22 nucleotides whose main mission is to inhibit gene expression. They have been reported in a wide range of living organisms. Besides acting intracellularly, they can mediate cell-cell communication via extracellular vesicles (EVs) secreted by most cell types in the extracellular space and different biological fluids. Moreover, dysfunction or dysregulation of miRNAs was reported in several types of diseases [16], [17]. For all these reasons, miRNAs have earned attention as biomarkers for diseases.

The different systems to quantify the concentration of miRNAs in fluids can be divided into direct and indirect measurements. Direct measurements quantify the miRNA counts in a sample, whereas indirect methods rely on previous RNA extraction and/or amplification before measuring the concentration of miRNAs in the samples. After quantifying the presence of miRNAs, inferring the network of elements involved in a particular process enables the obtention of a functional picture where nodes in the network represent the entities related

to the detected miRNAs (genes or transcripts), and they are linked according to the relationships inferred from the analysis of the measurements. Different techniques can calculate relationships, entropy, and mutual information metrics [18].

Among these technologies, data from indirect miRNA measurements are used in Chapter 8. The technology used for miRNA measurement is the NanoString's nCounter. This indirect method relies on a one-to-one relationship between each miRNA sequence and different miRTags, sequences of oligonucleotides specific for each miRNA strand. After coupling the miRNA and the miRTag, a code complex complementary to the hybridized miRTag and miRNA sequence pairs with it. Each code complex has a unique barcode at its end, formed by a combination of six positions and four different fluorophores. When the complexes are stabilized on a cartridge, the quantification of each one of the barcodes takes place. This assay relies only upon the extraction of the RNA and can include up to 800 different miRNA sequences. However, it has presented lower sensitivity than other technologies, and its turnaround time of approximately two days can prevent its use for real-time clinical diagnosis.

Despite being promising, the approval of miRNA quantifications as diagnostic tests is far from being established since they present several limitations regarding the high variability of their levels within the organism, the effects of the sample preparation and storage, variability in the efficiency of the RNA purification, and the lack of consensus in the choice of reference molecules for the posterior normalization of the analyzed data. For this reason, using data analysis tools that enable the comparison between the information inferred via miRNA quantification and more traditional tools is especially interesting. The benefits of such analysis would be bidirectional: on the one hand, miRNA results would be validated against well-established conventional means, and on the other hand, this might also validate already existing tools but more affordable than miRNA technologies.

2.2.3 Identification of metabolites by Raman spectroscopy

Raman spectroscopy is based on the phenomenon of Raman scattering. This occurs when a beam of photons interacts with the electrons of a sample. Suppose this interaction changes the polarizability of the sample's molecules concerning their previous stage. In that case, the Raman effect takes place, and the incident photons are then returned with a scattering proportional to the polarizability change. This produces a spectrum with different levels of energy associated with the frequencies of the captured photons, whose energy will vary

according to the sample's composition. Although this is the basic principle for Raman spectroscopy, this technique has dozens of variations.

The use of Raman spectroscopy in metabolomics to study the composition of samples is well established. The returned chemical fingerprints enable the identification of different compounds present in biochemical structures. Specifically, Raman spectroscopy has been widely used to study the composition of biological and clinical samples, given its lack of interference with water molecules. Moreover, Raman spectroscopy experiments are easy to apply: they come at a low cost, with a high speed of analysis, and provide broad information about the chemical composition and the studied structures.

Thus, models integrating the information acquired from affordable and non-invasive tests, such as Raman spectroscopy, with data from other omic levels can be especially interesting in carrying out translational research and suggesting more accessible sources of disease biomarkers. It has gained more interest in the last years, given its performance without the need for biopsies and more invasive tests. Namely, Raman spectroscopy has shown in previous works its utility in detecting composition differences in patients' EVs [19], [20].

In this thesis, Raman spectroscopy data appears in Chapter 8 to characterize the composition of Extracellular Vesicles (EVs), exemplifying an application of Raman spectroscopy in the context of biomarkers research from body fluids and secretions.

2.3 Biomedical Informatics

Biomedical informatics started being used in the 1970s to refer to any generic interaction between computers and medical attention. From then on, computer programs were developed targeting specific medical fields, reaching the decade of the 1980s with the emergence of computational biology. The growing assertion about biomedical data's power culminated in the 90s with the formal apparition of global interdisciplinary projects such as the Human Genome Project [21]. In parallel, this pathway was nurtured by improvements in computation science, which yielded smaller hardware devices with increasing processing power, easing their implementation in medical devices.

This brief but rapid history took us to the present time, with an outlook that is well known: data generation is more abundant than ever, and individuals are more self-aware based on their data (e.g., steps counters, estimated caloric consumption, etc.). This engagement was expected to expand with a com-

pound annual rate of 36% growth in the volume of healthcare data produced between 2018 and 2025 [22]. Moreover, this growth might be underestimated, considering the significant role of data in managing the COVID-19 pandemic.

This systematic implementation of the IoT has several implications involving all actors across health systems, with data generation reaching the user level, probably being its most transcendent feature. This scenario has enabled the integration of all its actors (patients, hospitals, research institutions, and policymakers). As a result, new products embracing this new paradigm were born during the last few years. The entities embodying most clearly this integration of information across all levels of healthcare are probably Clinical Decision Support Systems (CDSS).

A CDSS is intended to improve healthcare delivery by enhancing medical decisions with targeted clinical knowledge, patient information, and other health information. This concept mainly undertakes patient-centred approaches focused on providing a personalized healthcare service aligned with the Healthcare and Medicine 4.0 ideal.

The following sections will provide more information about CDSS. First, Section 2.3.1 describes the types of CDSS depending on the goal of their application. Secondly, Section 2.3.2 delves into the barriers faced by CDSS implementation in healthcare environments and ends with a critical comment focused on the technical solutions that could be provided to overcome some of the obstacles and weaknesses discussed in the section.

2.3.1 Types of Clinical Decision Support Systems

There are several criteria to classify CDSS. In terms of their components, Greenes describes CDSS as an integration of five parts: the method of computation, the knowledge needed as an input, the information model that imposes how the data is provided to the CDSS, the type of recommendation offered, and how the process interacts with the application environment [23]. Hence, different kinds of CDSS can be defined by articulating other choices for each one of the components mentioned above.

Berner provides another interesting definition: most CDSS consist of three parts [24]:

- The knowledge base is the information used to build links between pieces of data, i.e., if–the rules, associated risks between symptoms and diagnoses, or treatment incompatibilities.

- The inference engine is the logical or mathematical syntax that combines the rules with data from a certain patient.
- The communication mechanism is the interface enabling the input of patients' data and the visualization of CDSS' outcome.

According to clinicians' active or passive role in the use of CDSS, Berner distinguished between knowledge-based or non-knowledge-based CDSS [24]. The first ones are more permeable to clinicians' voices, e.g., suggesting a list of potential diagnoses or expecting the user to filter and override systems recommendations according to their judgment. On the contrary, the latter is based mainly on ML techniques to detect a specific pattern or condition from the patient's data. These systems have always been controversial because of the unjustifiable decision-making process that cannot explain the use of specific data and the relationships established to yield the obtained outcomes.

Another classification of CDSS can attend to their functionality. The application of CDSS in hospital environments can assist with managing patients on research/treatment protocols, logistics, preventive care, and healthcare process improvement. Sutton distinguishes six different functions attributed to CDSS [25]:

- Improve patient safety by reducing medication/prescribing errors and adverse events.
- Enhance clinical management by controlling the adherence to clinical guidelines and including alert systems with treatment follow-up reminders.
- Cost containment by reducing redundant tests, saving time by automating tedious steps, and suggesting more affordable treatment options.
- Administrative functions such as automating repetitive tasks like selecting standard medical codes or tracking service performance.
- Diagnostics support systems suggest diagnoses based on patient data, automating the interpretation of test results. A subcategory of these systems concerns directly analyzing data from medical images and laboratory tests.
- Patient-facing decision support aims to provide the CDSS outcome directly to the patient, mainly done via extensions of a commercial EHR service or standalone web-based or mobile-based applications.

Interestingly, merging the three first functionalities (improve patient safety while reducing errors and costs) results in CDSS resembling the process improvement philosophy propelled by the Lean Six Sigma methodology [26], [27]. The Lean Six Sigma methodology uses the Lean Manufacturing principles and the Six Sigma statistical tools to understand and improve a particular process. Although these techniques have been widely and traditionally applied in manufacturing and industrial environments, they can also improve healthcare processes. For instance, addressing protocol inaccuracies can improve patient adherence to hospital visits and treatments, reducing the variability of clinicians' agendas and ultimately resulting in a better healthcare service. Chapter 7 contains a case study of clinical management to shorten waiting time and to reduce the variability in the attention time for outpatients of a hospital's pharmacy unit.

The second type of CDSS receiving more attention in this thesis is CDSS for clinical diagnosis, also known as diagnostic decision support systems (DDSS), mentioned as the 5th functionality in [28]. Given the known incidence of diagnostic errors, particularly in primary care [29], DDSS can be helpful in laboratory testing and interpretation to improve diagnosis [30]. Over the last few years, the explosion of data generated resulted in DDSS using non-knowledge-based techniques like certain machine learning models. This philosophy might pave the way for a more accurate diagnosis. Still, it presents some drawbacks, and, unfortunately, DDSS has not had as much influence as other types of CDSS for reasons further discussed in Section 2.3.2. Chapters 8 and 9 present different real biomedical problems in which DDSS were applied following an approach that tried to overcome some of the barriers commented on in the next section.

2.3.2 Challenges of clinical decision support systems

As it has been exposed in this section, CDSS might be the most symbolic materialization of Medicine 4.0 paradigm. However, their use and implementation are far from clear, and there is an ongoing discussion about technical and ethical challenges brought by CDSS. This section briefly describes the different dimensions of these controversies: ethical, technological and financial.

First, there is scepticism about the ethical and professional consequences of CDSS implantation. This would be a long-term consequence of humans' over-reliance on CDSS, resulting in trespassing the decision-making to the CDSS and complete dependence on CDSS [31]. This "de-skilling" case raises a frequent ethical discussion about the responsibility of decision-making. However,

the doubts of such debate are based on the myth of machines' ethical responsibility, as CDSS decisions should always be double-checked and approved by humans [32].

Another long-term scenario involves the "carryover effect", which occurs when CDSS are used as educational tools and stop being required after completing the training [33]. The "carryover effect" could be solved by a continuous update of CDSS that should be part of proper maintenance, an often neglected part of their life cycle. Yet, this need leads to further questions (e.g., the update frequency, the interaction with other applications or operating systems, etc.) of a more pragmatic nature that also relate to the technical challenges described in the next paragraph.

Secondly, on a more technical side, as with any other computer service, the transportability and interoperability of CDSS can be hampered by complex programming and the intrinsic diversity of clinical data sources [34]. Besides, CDSS must also update their knowledge bases, as constituted by the datasets used to train the systems and by the approved medical guidelines. The difficulty of keeping CDSS up to date with the fluid medical knowledge is well known, and this part of maintenance is even more obscured if there is no way of identifying the algorithmic rules behind CDSS. As Musen, Middleton, and Greenes point out, many CDSS are embedded directly by vendors in their products, resulting in a wide range of approaches that are difficult to compare, inspect, formalize, and share [35].

The relation between these technical challenges and the ethical ones previously described seems clear, which leads to the question of how materially possible it is to implement CDSS if they already present considerable technical and ethical issues.

The previous question pinpoints the third and last dimension contemplated in this brief critical comment: the economic barriers. The truth is that even assuming a good integration of CDSS within the medical environment in all the aspects above, financial viability remains a struggle. Implementation costs to set up and integrate new systems can be substantial, and even if this barrier is overcome, ongoing costs should also be considered. Even cost-benefit assessments about CDSS implementations are limited, given the dependence on social and technological factors. This yields mixed, sparse conclusions that require systematic approaches to evaluate the economic effects of CDSS [36]–[40].

Thus, as has been exposed in this section, the implantation of CDSS faces many obstacles. The culture of personalized and fast services, spurred by the enthusiastic use of ML, AI, and Deep Learning tools, has met the complexity of factors involved in health. Although the technological progress seen in the last decades and described in Section 2.1 might explain the predominant role of accuracy in the design of CDSS, it also seems so far such a vision almost exclusively based on accuracy is not enough to push for the complete implantation of CDSSs in our healthcare systems.

Perhaps the next steps towards a more realistic and sustainable use of CDSS require looking at health as something beyond an individual service that focuses on individualized but opaque approaches. This last point backslides to the importance of interpretability in models used for CDSS. The exclusive focus on accuracy often implies hermetic decision-making rules that do not provide global knowledge, which can limit CDSS implementation, as already mentioned. Besides, health also comprehends collective dynamics and phenomena, with repercussions as crucial as the ones we could see recently with the COVID-19 pandemic when epidemiologists had to look beyond individual-level factors in understanding the virus dynamics. In the last instance, knowledge is derived from interpreting the information, guiding the ultimate decision-making. Nonetheless, the described technical barriers can inspire new solutions based on ML. Retaking Berner's definition of CDSSs, systems more permeable to clinicians' knowledge and completing the relationship between data – information – knowledge [41], [42], might be a good inspiration for proposing ML tools in BME.

The core motivation of this thesis is to pursue biomedical and healthcare knowledge that still leads to "smart" solutions encouraged by the 4.0 philosophy without jeopardizing the personalized component of the healthcare attention process focused on accuracy. Yet, this benefit can hardly be seen the other way around, i.e., the implantation of opaque CDSS does not allow the inference of factors with collective meaning. This pinpoints a second notion motivating this thesis: ML models' opacity can conflict with the intrinsic nature of research and science, whose ultimate goal is to understand reality better.

All the discussion maintained throughout this chapter, and more incisively along this section, falls upon the need for a critical gaze on the tools used to achieve the Healthcare 4.0 paradigm. Yet, this should not be read as a call-back for the exclusive use of rudimentary statistical tools but as an appeal for their integration along with ML approaches that can successfully account for complex interdependencies of variables, nonlinearities, and the heterogeneous nature of data massively recorded. This synergy between ML techniques and

traditional statistical thinking is embodied by the term Statistical Machine Learning (SML), which is addressed more deeply in Chapter 3.

Chapter 3

On statistical machine learning

3.1 Introduction

This chapter starts with an Introduction defining the concept of Machine Learning (ML), bringing in the debate around the "prediction culture" and linking it with the previous problem on the use of ML within BME, culminating with the definition of statistical machine learning as the direction guiding the research within this thesis. The next sections describe techniques and concepts that will be methodologically relevant throughout the rest of the chapters.

In 1959, Arthur Samuel (1901 - 1990) coined the term Machine Learning (ML) as [43]:

> *Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.*

The goal of obtaining knowledge from data was initially pursued by Statistics, a discipline based on inductive reasoning, materialized and formalized in the problem of statistical inference. Over the beginning of the past century, starting from separate applications in different fields, Statistics consolidated as a discipline on its own after the cohesion brought by the formulation of general mathematical concepts by personalities such as Karl Pearson or Ronald Fisher. Although the interpretative character of Statistics brought light to many fields dominated by purely theoretical abstractions, such as economics, physics, or agronomy, classical statistics faced certain limitations in dealing with increasingly complex problems.

From then on, the trajectory of Statistics starts curving towards being regarded as a computer science approach [44]. The advances in computational capacity led to new applications of mathematical concepts previously addressed by statistics. Some examples are resampling methods such as bootstrap and jackknife, the implementation of Expectation - Maximization algorithms to find the maximum likelihood estimators, or Monte Carlo Markov Chain (MCMC) approaches for parameters' inference.

The increase in computational power also democratized the use of sensors, and the massive acquisition of data gradually replaced the traditional experimental design planned before any data gathering. As a result, many of the assumptions required to apply statistical methods developed over the 20th century were not met under this new paradigm. This development of computational power

and the limitations encountered by classical statistical tools led to a different generation of tools for data science: Machine Learning techniques.

Machine Learning models are tools based on algorithms using a certain strategy to optimize a given target function: minimizing the estimation error, maximizing the sensitivity of a classification problem, etc. These models are all tuned by numerous parameters and hyperparameters that give them almost boundless freedom to adjust the problem represented by the data properly.

This turned into a rise of a "prediction culture" within research, a term coined by Leo Breiman, who, in his essay "Statistical modelling: the two cultures", compares the culture of data models to the culture of algorithmic modelling, i.e., of ML models, standing himself as an advocate for the second one [45].

In his article, Breiman exposes the limitations that he found when classical model-based statistical thinking was applied to real problems with real data. As a response, he got equally interesting replies from Professor David Cox and Brad Efron, previously mentioned in this section. Cox acknowledges the utility of empirical approaches based on prediction accuracy to certain problems but adds:

> *"Prediction (is) always hazardous without some understanding ...]. Formal models are useful and often almost, if not quite, essential for incisive thinking. [...] Professor Breiman takes a rather defeatist attitude towards attempts to formulate underlying processes; is this not to reject the base of much scientific progress?".*

This scepticism on Breiman's view is perhaps more constructively shaped in Efron's response, who takes Breiman's perspective not as a rejection of classical statistics but as a call for their update:

> "*New methods always look better than old ones. [...] Complicated methods are harder to criticize than simple ones, [...] One of the best things statisticians do is clarify the inferential basis of a proposed new methodology. I believe the hardest part of this work remains to be done. Papers like Leo's are a call for more analysis and theory, not less. [...] I believe that the current interest in statistical prediction will eventually invigorate traditional inference, not eliminate it. [...] The whole point of science is to open up black boxes, understand their insides, and build better boxes for the purposes of (human)mankind.*"

Yet, the most inspiring part might be, paradoxically, the last piece of words of Breiman's paper, written by himself as his concluding remark responding to the previous comments from other authors:

> "*The danger is that if we define the boundaries of our field in terms of familiar tools and familiar problems, we will fail to grasp the new opportunities.*"

Ironically, the success of Breiman's vision meant familiarizing data scientists with the prediction culture brought by ML tools. Following Breiman's advice, it might be time to reassess this moment of data science research with a critical gaze.

The statistical framework offered something aforementioned by Cox and Efron that mere Machine Learning lacks: statistical thinking. Currently used algorithms are usually black boxes that can hardly be generalized, as they are paired with the data used to fit them. Their opacity prevents the explanation of their decisions, their proper use in slightly different scenarios, or the justification of their service when they have public and social consequences. As Efron also put it in his essay, "We are back at the beginning [...], with lots of ingenious ad hoc algorithms, but no systematic framework for comparing them". It seems timely to propose updates that merge statistical thinking with the new methodological corpus provided by machine learning, which is critical in BME research for the reasons exposed in the following paragraphs.

As exposed in Chapter 2, medical data generation is expected to expand in volume and variety in the upcoming years. From a technical side, such heterogeneous data are often incomplete or unbalanced, which may affect the use of advanced technology for its analysis. Coming across this "dirty" data to obtain a clean and usable data set can need domain knowledge, which can be difficult to integrate quantitatively and qualitatively without tools providing an exploratory analysis of the original data. Yet, the assessment of results derived from such exploratory analysis often requires the choice of techniques that are permeable to the biomedical knowledge base of researchers and clinicians.

Moreover, even if models can be obtained from curated datasets, insufficient details or transparency can lead to low confidence in their outcomes and implementation. On the other side of the spectrum, models with too much detail about a specific situation might be difficult to scale up, and extrapolating slightly different conditions can become unfeasible. Thus, even the most ac-

curate prediction models should account for this trade-off between process complexity and the feasibility of the proposed methodology.

From a social perspective, "smart" approaches based on AI and ML models bring value because of their accurate and individualised predictions. However, the black-box models at their cores opaque the decision-making process of the resulting algorithms. Such an approach trades might not be necessarily in the best interest of research on understanding biology's or health's nature. Yet, the complexity of factors involved requires new approaches quite different from traditional causal inference, needing to account for interdependencies of variables, nonlinearities, and the heterogeneous nature of data massively recorded. This leads to the technical problem described in the previous paragraph, restarting the circular path defined by the complex relationship between "smart" automation and the extraction of global knowledge. Besides, using opaque decision-making systems also affects the implantation of CDSS, as pointed out previously in Chapter 2.

This complexity dominating the Industry 4.0 and Healthcare 4.0 eras raises challenges but also sets the perfect breeding ground for innovations which the exposed critical thinking should nurture. Recommendations include using tools to identify causal effects and analytical approaches that can ground theoretical assumptions. All of this highlights the use of the appropriate statistical methods. Specifying precisely the causal questions related to a certain issue brings a better understanding of the data needed to implement workflows as a response, to decide the best analytical approach for decision-making, and also provides ground to validate the assumptions involved.

In summary, understanding bioprocesses and health at all levels requires integrating many different types of evidence, rigorous quantitative analysis of observational studies, and systems modelling. The arrival of Industry 4.0 and Healthcare 4.0 paradigms already broke the barrier to accessing increasing amounts of linked data. Now, it is the turn of the research community to offer new methodologies and technical solutions to tackle this exciting challenge.

This thesis aims to be a small step toward finding this balance between hypothesis-driven approaches and emerging data-driven algorithms. The concept of Statistical Machine Learning aims to embody such balance. With that philosophy in mind, different methodologies have been applied to propose both new algorithms for data science research (Part II) and new models for BME problems (Part III), keeping a translational approach in mind. More methodological context is provided in this Chapter, with the following sections describing some of the techniques applied throughout this thesis.

3.2 Notation

This thesis will represent scalar values as italic capital letters (e.g., N) and indices as lower-case letters (e.g., i). When an index is related to a particular scalar, the same letter will be used for both (e.g., $n = 1, \ldots, N$).

Column vectors are represented as bold lower-case letters (e.g., $\boldsymbol{x}$) and row vectors as their transpose (e.g., $\boldsymbol{x}^\top$). When the elements within a vector are listed, they will be expressed between brackets (e.g., $\boldsymbol{x}^\top = [x_1, \ldots, x_K]$. The same notation without commas will be used if a vector is a concatenation of vectors (e.g., $\boldsymbol{x}^\top = [\boldsymbol{y}^\top \boldsymbol{z}^\top]$).

Matrices will appear as bold capital letters (e.g., $\boldsymbol{X}$) and are often illustrated as squares and rectangles. Rows will represent observations or individuals, whereas columns will represent variables. The same notation as in vectors will be used for matrices (also applicable to rows and columns of a matrix). When possible, the same letter will be used for the dimension of a matrix and the index of one of its elements (e.g., the kth column of the matrix $\boldsymbol{X}$ with K columns, will be referred to as $\boldsymbol{x}_k$).

Either Latin or Greek characters will be used to represent scalars, vectors, or matrices. When a vector or matrix has the same value for all its entries, it will be expressed as a bold number with the dimensions (e.g., $\mathbf{0}_N^\top$ is a row vector with N zeros).

The mathematical operator $\times$ will be used to denote the size of a matrix. The mathematical operator $\cdot$ will denote products between scalars, vectors, or matrices. The mathematical operators $\langle$ and $\rangle$ indicate the inner product. The mathematical operator $\odot$ denotes the Hadamard or element-wise product.

In sections related to missing data (Section 3.5), matrices and vectors will be treated as a partition between observed (denoted by *) and missing values (denoted by $^\#$), i.e.: $\boldsymbol{X} = \left[\boldsymbol{X}^* \boldsymbol{X}^\#\right]$. For instance, $\boldsymbol{T}^* = \boldsymbol{X}^* \boldsymbol{P}^*$ would be the scores obtained by projecting the observed variables onto the PCA model.

Finally, in the context of machine learning (ML) models, it's pivotal to distinguish between Model Building (MB) and Model Exploitation (ME) contexts. MB refers to the phase where the model is trained or fitted using available data. This process involves learning the underlying patterns or relationships within the data, adjusting model parameters to minimize errors, and establishing a predictive or descriptive framework. On the other hand, ME occurs after the model has been constructed and validated; it involves deploying the trained model to make predictions or projections on new or unseen data. In

summary, ME utilizes the model fitted during the MB phase. Understanding this distinction is crucial as it illuminates the dual roles of ML models: one in constructing the model (MB) and the other in applying that model to new data (ME).

This chapter provides the mathematical framework of ML models used in the rest of the chapters of this thesis. The division will be based on the most common division of ML models, which is based on the mathematical nature of the problem they are supposed to solve as *unsupervised* or *supervised*. An assessment of their interpretability will be provided in their corresponding sections.

3.3 Unsupervised machine learning techniques

When the modelling task uses only the information about inputs, without labels or response variables within the dataset, it uses unsupervised learning. With these algorithms, the target function to be optimized cannot be the accuracy of the response prediction.

Probably, the best-known type of unsupervised learning is clustering. The goal of clustering methods is to find similarities in the training data. The underlying assumption of clustering is that assigned clusters could be, in the last instance, a classification of the observations not explicitly declared in the data. The output of unsupervised models is a certain structure of the data.

A typical case is the use of unsupervised models as exploratory tools. A cost function can be minimized to assess the resulting structure's goodness. For instance, the distance between observations and the representative parameters of its assigned cluster is minimized. Moreover, it is essential to guarantee that the extracted patterns represent the data and that no over-fitting is artificially generating them.

3.3.1 Principal component analysis

Let $\boldsymbol{X}$ be a matrix with N observations on K variables. After some pre-processing, such as mean-centring and unit variance scaling, a Principal Component Analysis (PCA) model is estimated ([46]). This is done by compressing the high-dimensional $\boldsymbol{X}$ matrix into a low-dimensional subspace of dimension A (with $A \leq rank(\boldsymbol{X})$).

Mathematically, PCA is based on the bilinear decomposition of $\boldsymbol{X}$ as in Equation 3.1.

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\top} + \boldsymbol{E} \tag{3.1}$$

where $\boldsymbol{T}$ is an $N \times A$ matrix of *scores* and $\boldsymbol{P}$ is a $K \times A$ matrix of *loadings*. The A columns of the loading matrix $\boldsymbol{P}$ are the A *loading* vectors $\boldsymbol{p}$.

The *score* matrix $\boldsymbol{T}$ can be considered as a collection of row vectors $\boldsymbol{\tau}^{\top}$ (scores of an observation) or column vectors $\boldsymbol{t}$ (latent variables, with $\boldsymbol{t}_a = \boldsymbol{X}\boldsymbol{p}_a$ and $a = 1, 2, \ldots, A$). The score matrix can be obtained as $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{P}$, that is, as the projection of the $\boldsymbol{X}$ matrix on the $A-$dimensional space of the PCA model (i.e., columns of $\boldsymbol{P}$ matrix). Analogously, given an observation $\boldsymbol{x}^{\top}$ of the original $K-$dimensional space, its projection $\boldsymbol{\tau}$ onto the subspace of the model can be obtained using the projection matrix $\boldsymbol{P}$ as well by $\boldsymbol{\tau} = \boldsymbol{P}^{\top}\boldsymbol{x}$.

From the scores matrix, one can recall the explained part of $\boldsymbol{X}$ by the PCA model as $\hat{\boldsymbol{X}} = \boldsymbol{T}\boldsymbol{P}^{\top}$. This notation can be used as well for individual observations, where $\hat{\boldsymbol{x}} = \boldsymbol{P}\boldsymbol{\tau}$. The original observation can be decomposed into a part explained (i.e., predicted) by the model (signal or $\hat{\boldsymbol{x}}$) and an error term not considered in any of the A latent variables (noise or $\boldsymbol{e}$). Thus, for a given observation we have $\boldsymbol{x} = \boldsymbol{P}\boldsymbol{\tau} + \boldsymbol{e}$ and then $\boldsymbol{e} = \left(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^{\top}\right)\boldsymbol{x}$.

3.3.2 Robust principal component analysis

In real datasets, finding outliers within the matrix used for PCA Model Building (PCA-MB) is frequent, which becomes a threat for those methods purely based on least-squares (LS) estimators. The property used more frequently to quantify the resistance of an estimator to the presence of outliers is its breakdown value [47].

The breakdown value of an estimator (Equation 3.2) is given by the minimum number of M observations among a set of N that must be replaced by arbitrary values so that the estimator T applied to the altered sample $\boldsymbol{X}'$, yields a value beyond all bounds:

$$\epsilon_N^*(T; \boldsymbol{X}) = \frac{1}{N}\left\{M \in 1, \ldots, N : \quad sup_M D(T(\boldsymbol{X}); T(\boldsymbol{X}')) = \infty\right\} \tag{3.2}$$

where $\boldsymbol{X}$ is the original clean set, $\boldsymbol{X}'$ is the new set with M altered observations, T is a given estimator and D is a distance metric.

Another attribute defining the robustness of an estimator is its influence function, which measures the effect of infinitesimal contamination at a point $\boldsymbol{x}$, following a clean distribution F, on the estimator T:

$$IF(\boldsymbol{x}, T, F) = lim_{\varepsilon \to 0} \frac{T(F_\varepsilon) - T(F)}{\varepsilon} \tag{3.3}$$

where ε is the fraction of contamination and $F_\varepsilon = (1-\varepsilon)F + \varepsilon\Delta_{\boldsymbol{x}}$, with $\Delta_{\boldsymbol{x}}$ having all its mass in $\boldsymbol{x}$. Robust estimators should show a bounded IF with small values. Another useful property is the efficiency of an estimator, which compares the goodness of its values for non-contaminated data to the estimates yielded by a classical estimator.

For instance, the classical PCA model from Section 3.3.1 has a 0% breakdown value, i.e., even a single outlying point could break the PCA model. Understandably, such a low breakdown value for classical PCA inspired the proposal of robust adaptations of the PCA model.

Robust PCA models propose different strategies for obtaining a PCA model clean from the influence of outliers. In broad lines, there are three types of robust PCA models according to the strategy they follow. There are robust PCA models based on robust estimators, robust PCA models based on projection pursuit approaches, and robust PCA models that merge both approaches and are named hybrid robust PCA methods. The following sections briefly describe each group, diving more deeply into robust steps included in the MacroPCA algorithm from Hubert et al. [48], used later in Chapter 6.

Robust Estimators

One possibility to robustify the PCA model is to obtain it from a robust estimation of the mean and covariance matrix. One of the first attempts was the M-estimators proposed by Huber [49]. These M-estimators are defined as the solutions $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ to the system of equations:

$$\begin{cases} n^{-1}\sum_{i=1}^{n} u_1\left[\{(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^\top \hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})\}^{1/2}\right](\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}) = \mathbf{0} \\ n^{-1}\sum_{i=1}^{n} u_2\left[\{(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^\top \hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})\}\right](\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^\top = \hat{\boldsymbol{\Sigma}} \end{cases} \tag{3.4}$$

where $u_1(s)$ and $u_2(s)$ are nonnegative, nonincreasing and continuous functions defined for $s \geq 0$. The key is to find such functions $u_1(s)$ and $u_2(s)$ that

act as weights of the observations in the computation of the location and scatter parameters. When these weights are equal to 1 for all the observations (i.e., observations have full weight regardless of their Mahalanobis distance), Equations 3.4 are the classical least squares estimators.

Some weaknesses of M-estimators include a potential lack of robustness for matrices with small samples of high dimensionality [50], and their suboptimality if observations are outlying only for one principal component, but with normal values along the rest of directions [51]. This later criticism resulted in Campbell's proposal of iteratively computing principal components little influenced by outliers, sequentially estimating each eigenvector from $\hat{\boldsymbol{\Sigma}}$ until extracting all A principal components or until reaching a specified proportion of explained variance. The final robust estimate for the covariance matrix is reconstructed from the spectral decomposition using the fitted eigenvectors and eigenvalues.

Whereas the methods mentioned above kept a modified version of the least sum of squares estimation based on their reweighting, the Minimum Covariance Determinant (MCD) algorithm [52] aimed to minimize the covariance determinant by finding the set of h least outlying observations:

1. The sample mean $\hat{\boldsymbol{\mu}}_0$ of the h observations yielding the MCD is calculated.
2. The covariance matrix yielded by the same set of h observations is calculated and multiplied by a consistency factor $\hat{\boldsymbol{\Sigma}}_0$.
3. A reweighting step is applied to improve the efficiency of the estimators based on observations' Mahalanobis distance to the hyperellipsoid defined by the raw MCD estimates $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\Sigma}}_0$, yielding the final estimates as:

$$\hat{\boldsymbol{\mu}}_{MCD} = \frac{\sum_{i=1}^{N} W(d_i^2)\boldsymbol{x}_i^\top}{\sum_{i=1}^{N} W(d_i^2)} \tag{3.5}$$

$$\hat{\boldsymbol{\Sigma}}_{MCD} = c_1 \frac{1}{n} \sum_{i=1}^{N} W(d_i^2)(\boldsymbol{x}_i^\top - \hat{\boldsymbol{\mu}}_{MCD})(\boldsymbol{x}_i^\top - \hat{\boldsymbol{\mu}}_{MCD})^\top \tag{3.6}$$

with $d_i = \sqrt{(\boldsymbol{x}_i^\top - \hat{\boldsymbol{\mu}}_0)^\top \hat{\boldsymbol{\Sigma}}_0^{-1} (\boldsymbol{x}_i^\top - \hat{\boldsymbol{\mu}}_0)}$ where $W(d^2)$ is a weight function and the constant c_1 is another consistency factor.

However, finding the subset of the h least outlying observations implied a search among $\binom{N}{h}$ combinations, which was too time-consuming. This lim-

itation motivated the proposal of the Fast-MCD algorithm years later [53]. This new proposal reduced the number of subsets of size h considered for the computation of the MCD while keeping the robust properties of the MCD estimator.

However, none of these robust methods could resist many outliers and they required the inversion of the estimated covariance matrix at some point in their implementation, which could be problematic with high-dimensional datasets.

Projection Pursuit

In contrast to approaches from Section *Robust Estimators*, Projection Pursuit (PP) approaches define the robust PCA model fitting as an optimization problem other than minimizing the sum of squared residuals. The pioneering work by Li and Chen [54] obtained the eigenvectors of the covariance matrix, i.e., the loadings of the PCA model, as the directions with maximal dispersion of the scores, using as a measure of dispersion a robust scale estimator $\hat{\boldsymbol{s}}_R$ (yielded by applying the s_R function to a matrix or vector):

$$v_{s_R,a} = argmax_{||\boldsymbol{v}||=1,\boldsymbol{v}\perp\boldsymbol{v}_{s_R,1},\ldots,\boldsymbol{v}\perp\boldsymbol{v}_{s_R,k-1}} s_R(\boldsymbol{v}^\top \boldsymbol{x}_1, \ldots, \boldsymbol{v}^\top \boldsymbol{x}_n) \tag{3.7}$$

The associated eigenvalues are calculated as the robust variances of the projections, and the covariance matrix could be obtained from the spectral decomposition using the obtained A eigenvectors and eigenvalues.

The main drawback of this method was the computationally intensive optimization of Equation 3.7. This limitation fostered the proposal of more PP approaches still based on the concept from Equation 3.7, but varying either the robust scale metric (s_R) or the searching method to obtain the set of directions $\boldsymbol{v}$, considered as potential eigenvectors.

The algorithm of Croux and Ruiz-Gazen(C–R algorithm) [55] proposed a new PP algorithm based on the work from Li and Chen, but using the L_1-median as the robust location estimate μ_R and the M-estimator of scale as s_R. However, given the problematic use of M-estimators in high-dimensional data sets where $N < K$, authors suggested using the MAD (Median Absolute Deviation) or the Q estimator as robust scale measures in such matrices [56].

The key component of the C–R algorithm was the search for new direction among the set $\boldsymbol{V}^{(a)}$ (Equation 3.8), instead of considering all potential solutions.

$$\boldsymbol{V}^{(a)}(\boldsymbol{X}) = \frac{\boldsymbol{x}_i^a}{||\boldsymbol{x}_i^a||}; 1 \leq i \leq n \tag{3.8}$$

Therefore vectors in $\boldsymbol{V}^{(a)}$ are actually the directions of the data points deflated by their reconstructed versions using the already obtained $a-1$ eigenvectors:

$$\boldsymbol{x}_i^a = \boldsymbol{x}_i^{a-1} - \boldsymbol{\tau}_i^{a-1}\hat{\boldsymbol{v}}_{a-1}^{s_R} \forall i = 1, \ldots, N \tag{3.9}$$

This algorithm, usually referred to as the C–R algorithm, was based on the idea that it might be simpler to inspect the observations to find a fairly close direction to the ath eigenvector indicating the true maximum dispersion of the data. However, despite improving the previous results from PP approaches, the C–R algorithm still lacks numerical stability and was quite time-consuming for high-dimensional datasets.

The work from Hubert, Rousseeuw, and Verboven [57], used the C–R algorithm as a base to propose the RAPCA algorithm, which stands for *Reflection-based Algorithm for PCA*. The RAPCA method kept the same robust estimators and the stepwise approach from [56] but included two new steps. It included the Reflection step (R–step) to solve the numerical issues presented by the C–R algorithm in high dimensions. Nonetheless, since the R–step took more computation time, the RAPCA algorithm included a previous kernel transformation of $\boldsymbol{X}$ to speed up the computations when $K > N$.

The first step of the RAPCA algorithm is to map the original data onto the subspace spanned by the N observations. This subspace is the kernel approach for obtaining the eigenvalue decomposition of $(\boldsymbol{X} - \boldsymbol{1}_N (\hat{\boldsymbol{\mu}}^C)^\top)(\boldsymbol{X} - \boldsymbol{1}_N (\hat{\boldsymbol{\mu}}^C)^\top)^\top$, where $\hat{\boldsymbol{\mu}}^C$ is the classical mean vector of the initial matrix $\boldsymbol{X}$. This decomposition is performed on a $N \times N$ matrix instead of on a $K \times K$ matrix, which is especially helpful for cases where $K >> N$. All the obtained eigenvectors are stored in the loading matrix $\tilde{\boldsymbol{P}}$ to project the data.

$$\boldsymbol{X} - \boldsymbol{1}_N \left(\hat{\boldsymbol{\mu}}^C\right)^\top = \tilde{\boldsymbol{T}}\tilde{\boldsymbol{P}}^\top \tag{3.10}$$

where $\tilde{\boldsymbol{T}}$ is a matrix of $N \times R$ dimensions, $\tilde{\boldsymbol{P}}$ is a matrix of $K \times R$ dimensions, $\hat{\boldsymbol{\mu}}^C$ is the classical mean vector and $R = rank(\boldsymbol{X} - \boldsymbol{1}_N (\hat{\boldsymbol{\mu}}^C)^\top) \leq (N-1)$. It is important to notice that the goal of this step is to accelerate posterior calculations, not to search for the latent dimension of the dataset, and thereby,

it is critical to use all yielded eigenvalues, not only the first ones related to the biggest eigenvalues.

Secondly, the RAPCA algorithm introduces the reflection step (R-step) used for the stepwise search of the principal components' directions. The input of this step is the centered scores matrix $\tilde{\boldsymbol{T}} - (\hat{\boldsymbol{\mu}}_T^R)^\top = \boldsymbol{X}^{(1)}$, with $\hat{\boldsymbol{\mu}}_T^R$ being a robust location estimator of $\tilde{\boldsymbol{T}}$, namely the median:

1. The first eigenvector $\hat{\boldsymbol{v}}_1$ is obtained as in the C–R algorithm.
2. The data is transformed by a reflection transformation U_1 such that:

$$U_1(\tilde{\boldsymbol{v}}_1) = \boldsymbol{e}_1 = (1, 0, \ldots, 0)^\top \in \mathbb{R}^R \tag{3.11}$$

   A vector with the direction mapping $\hat{\tilde{\boldsymbol{v}}}_1$ onto $\boldsymbol{e}_1$ and with unitary module, is obtained as:

$$\boldsymbol{n}_1 = \frac{\boldsymbol{e}_1 - \tilde{\boldsymbol{v}}_1}{||\boldsymbol{e}_1 - \tilde{\boldsymbol{v}}_1||} \tag{3.12}$$

   The reflection operation is applied on each observation $\boldsymbol{x}_i^{(1)}$ as:

$$\tilde{\boldsymbol{x}}_i^{(2)} = U_1(\boldsymbol{x}_i^{(1)}) = \boldsymbol{x}_i^{(1)} - 2\langle \boldsymbol{x}_i^{(1)} \boldsymbol{n}_1 \rangle \boldsymbol{n}_1 \tag{3.13}$$

   which yields a vector $\tilde{\boldsymbol{x}}_i^{(2)}$ of the same module as $\boldsymbol{x}_i^{(1)}$ and aligned in the direction $\boldsymbol{n}_1$.
3. The data points $\boldsymbol{x}_i^{(1)}$ are projected directly onto the orthogonal complement of $\tilde{\boldsymbol{x}}_i^{(2)}$ by omitting its first coordinate, yielding $\tilde{\boldsymbol{x}}_i^{(2)} = (x_{i2}^{(2)}, \ldots, \tilde{x}_{iR}^{(2)})^\top \in \mathbb{R}^{R-1}$
4. The second eigenvector $\hat{\tilde{\boldsymbol{v}}}_2$ is found applying the C–R strategy to the set $\tilde{\boldsymbol{X}}^{(2)}$. The vector $\hat{\tilde{\boldsymbol{v}}}_2$ is backtransformed to $\mathbb{R}^R$ by applying the inverse reflection.
5. The steps are repeated until A eigenvectors are found. This yields the final matrix $\tilde{\tilde{\boldsymbol{P}}} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_A)$ of $R \times A$ dimensions. The final loadings matrix of the PCA model is obtained by its back-projection onto the loadings matrix of the kernel transformation:

$$\boldsymbol{P} = \tilde{\boldsymbol{P}}\tilde{\tilde{\boldsymbol{P}}} \tag{3.14}$$

And similarly, with the location estimator:

$$\hat{\boldsymbol{\mu}}^R = \hat{\boldsymbol{\mu}}^C + \hat{\boldsymbol{\mu}}_T^R \tilde{\boldsymbol{P}}^\top \tag{3.15}$$

The final PCs computed by the RAPCA algorithm are expressed in the following equation:

$$\boldsymbol{T} = \left(\left(\boldsymbol{X} - \mathbf{1}_N \hat{\boldsymbol{\mu}}^C\right)\tilde{\boldsymbol{P}} - \mathbf{1}_N \hat{\boldsymbol{\mu}}_T^R\right)\tilde{\tilde{\boldsymbol{P}}} = \left(\boldsymbol{X} - \mathbf{1}_N \hat{\boldsymbol{\mu}}^c - \mathbf{1}_N \hat{\boldsymbol{\mu}}_T^R \tilde{\boldsymbol{P}}^\top\right)\tilde{\boldsymbol{P}}\tilde{\tilde{\boldsymbol{P}}} = \left(\boldsymbol{X} - \mathbf{1}_N \left(\hat{\boldsymbol{\mu}}^R\right)^\top\right)\boldsymbol{P} \tag{3.16}$$

This line of work started with the RAPCA algorithm and was then followed by the same research group with a series of new hybrid algorithms for robust PCA, explained in the following section.

Hybrid approaches

A third line of work merges both strategies previously mentioned. Hybrid approaches usually start from a PP step used for initial dimension reduction. Then, robust covariance estimators can be applied once this reduction is obtained (with $N > K$).

After the proposal of the RAPCA algorithm [57], Hubert, Rousseeuw, and Vanden-Branden [58] proposed the ROBPCA algorithm. This algorithm combined an initial PP part for initial dimension reduction with the estimation of the final latent subspace using the MCD estimator. The ROBPCA method follows three main stages:

1. The reduction of the data space to the subspace spanned by the N observations is applied with the same philosophy as in RAPCA [57]. The reconstructed matrix using the R_0 resulting eigenvectors and eigenvalues is used for the subsequent steps.
2. The h least outlying observations are searched as those observations yielding the smallest value of Equation 3.17, where B contains all directions

through two data points and if $\binom{n}{2} > 250$, a random set of 250 directions is selected.

$$outl(\boldsymbol{x}_i) = max_{\boldsymbol{v} \in B} \frac{|\boldsymbol{x}_i^\top \boldsymbol{v} - t_{MCD}(\boldsymbol{x}_i^\top \boldsymbol{v})|}{s_{MCD}(\boldsymbol{x}_i^\top \boldsymbol{v})} \tag{3.17}$$

This outlyingness measure is a version of the originally proposed Stahel–Donoho affine–invariant outlyingness measures (Stahel 1981, Donoho 1982). In case of having null variance for a given projection, $s_{MCD}(\boldsymbol{x}_i^\top \boldsymbol{v})$, that is an "exact fit" case, and it means that the direction $\boldsymbol{v}$ of projection is completely orthogonal to the hyperplane $H_{\boldsymbol{v}}$ containing h observations. In this case, the ROBPCA algorithm applies the reflection step from the RAPCA algorithm [57].

Once the set H_0 of h least outlying observations is found, their mean ($\hat{\boldsymbol{\mu}}^{(1)}$) and covariance matrix ($\hat{\boldsymbol{\Sigma}}^{(0)}$) are obtained. The latent dimension A_0 is set according to the eigenvalues of the spectral decomposition of $\hat{\boldsymbol{\Sigma}}^{(0)}$, yielding the matrix $\boldsymbol{P}^{(*)}$ of dimensions $R_0 \times A_0$ and the scores are obtained as $\boldsymbol{T}^{(*)} = (\boldsymbol{X} - \boldsymbol{1}_N \hat{\boldsymbol{\mu}}^\top)\boldsymbol{P}^{(*)}$.

3. The FAST-MCD algorithm from Rousseeuw and Van Driessen [53] is applied after applying some C-steps to observations $\boldsymbol{x}_i^{(*)}$ with $i \in H_0$. This might yield another h-subset H_1 of least outlying observations, with a location estimator $\hat{\boldsymbol{\mu}}^{(3)}$ and a covariance matrix $\hat{\boldsymbol{\Sigma}}^{(1)}$ yielding a lower determinant than those from H_0. The FAST-MCD algorithm is applied on this subset H_1, yielding the $(\hat{\boldsymbol{\mu}}^{(3)}, \hat{\boldsymbol{\Sigma}}^{(2)})$ estimates. The set $(\hat{\boldsymbol{\mu}}^{(2)}, \hat{\boldsymbol{\Sigma}}^{(1)})$ or $(\hat{\boldsymbol{\mu}}^{(3)}, \hat{\boldsymbol{\Sigma}}^{(2)})$ corresponding to the lower covariance matrix determinant, will be the one used in further steps.

   A reweighted mean and covariance matrix are obtained, yielding $(\hat{\boldsymbol{\mu}}^{(5)}, \hat{\boldsymbol{\Sigma}}^{(4)})$. The spectral decomposition of $\hat{\boldsymbol{\Sigma}}^{(4)}$ yields the loadings matrix $\boldsymbol{P}^{(2)}$. The final loading matrix and the final location estimator are back-projected to the original space yielding the final location vector, covariance matrix, and loadings matrix.

This algorithm outperformed its precedent, purely PP version RAPCA (see Section 3.3.2 [57]), showing lower errors in estimating the eigenvalues. Nonetheless, ROBPCA was not prepared to deal with the existence of a different type of outlier, cell-wise outliers. Its next adaptation came years later as the MacroPCA algorithm [48], which applied ROBPCA after a previous step exe-

cuting the Detect Deviating Cells (DDC) algorithm [59] to deal with outlying cells (see Section 3.6.2).

3.4 Supervised machine learning techniques

Supervised learning aims to fit an ML model that successfully reproduces an already-existent classification or forecast system (made by humans or other machines). This type of learning focuses on relating the inputs to the outputs of the data set, driven by target functions that measure the goodness of predictive relationships.

In these cases, models are susceptible to the training data set. Biases and imbalances can be reproduced by supervised models, including them in the future classification with new individuals. Thus, over-fitting is a problem found for these models as well. This can be especially critical if the training data set contains unrecognized or misclassified outliers, given that the resulting model will also fit these observations, misinterpreting the correct generalization rules.

3.4.1 Partial Least Squares Regression

Partial Least Squares Regression (PLS) [60], [61] pursues finding a subspace in the N input space $\boldsymbol{X}$ and a subspace in the N output space $\boldsymbol{Y}$ to maximize the covariance of the resulting LVs after projection,

Whereas Multiple Linear Regression (MLR) or ML techniques focus on modeling the relationship between inputs ($\boldsymbol{X}$) and outputs ($\boldsymbol{Y}$), PLS links both spaces through the LVs providing not only a model for this relationship but also a model for $\boldsymbol{X}$, what offers unique properties. LVs computed in the PLS model represent the main driving forces linking the input to the output space. The following equations describe the PLS regression model structure:

$$\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}^{*} \tag{3.18}$$

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\top} + \boldsymbol{E} \tag{3.19}$$

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{Q}^{\top} + \boldsymbol{F} \tag{3.20}$$

Columns of $\boldsymbol{T}$ are the PLS score vectors, conforming to a matrix of dimensions $N \times A$, where A is the number of LVs of the model. These are estimated as a linear combination of the original variables with the corresponding weight vectors from $\boldsymbol{W}^{*}$ (Equation 3.18).

As commented, PLS models the relation between $\boldsymbol{X}$ and $\boldsymbol{Y}$ through their projection onto the latent subspace of dimension equal to the number of LVs. This is the reason why PLS scores, $\boldsymbol{T}$, are simultaneously good summaries of $\boldsymbol{X}$ according to $\boldsymbol{P}$ (Equation 3.19) and good predictors of $\boldsymbol{Y}$ according to $\boldsymbol{Q}$ (Equation 3.20). Besides, the number of selected LVs is related to the effect of the dimensionality reduction. The bigger the reduction, the fewer LVs (A), and the information not represented by these A LVs is stored in the error terms $\boldsymbol{E}$ (for inputs) and $\boldsymbol{F}$ (for outputs).

Consequently, $\boldsymbol{E}$ and $\boldsymbol{F}$ become key indicators of the PLS model goodness of fit: the smaller the sum of squares of $\boldsymbol{F}$ is, the better the model is for the prediction, and the smaller the sums of squares of $\boldsymbol{E}$ is, the better the model explains the $\boldsymbol{X}$-space. Usually, the number of latent variables is selected in such a way that $\boldsymbol{E}$ and $\boldsymbol{F}$ matrices can be considered to contain nothing but noise, keeping the meaningful information (signal) stored in the A PLS latent variables.

For a given observation, to evaluate the model performance, projecting an observation, $\boldsymbol{x}$, onto it, the Hotelling-T^2 in the latent space, T^2, and the Squared Prediction Error (SPE), are calculated:

$$\boldsymbol{\tau} = \boldsymbol{W}^{*\top}\boldsymbol{x} \tag{3.21}$$

$$\boldsymbol{T}^2 = \boldsymbol{\tau}^\top \boldsymbol{\Lambda}^{-1}\boldsymbol{\tau} \tag{3.22}$$

$$SPE = (\boldsymbol{x} - \boldsymbol{P}\boldsymbol{\tau})^\top (\boldsymbol{x} - \boldsymbol{P}\boldsymbol{\tau}) = \boldsymbol{e}^\top \boldsymbol{e} \tag{3.23}$$

Where $\boldsymbol{e}$ is the residual vector associated with the observation, $\boldsymbol{\Lambda}^{-1}$ the diagonal matrix containing the inverse of the A variances of the scores associated with the LVs, $\boldsymbol{\tau}$ the vector of scores corresponding to the projection of the n-th observation $\boldsymbol{x}$ onto the latent subspace of the PLS model, and the T^2 and the SPE hold the same meaning as for the PCA model (see Section 3.3.1).

The PLS model can be expressed as well as a function of the input variables by substituting Equation 3.18) into Equation 3.20):

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}^*\boldsymbol{Q}^\top + \boldsymbol{F} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{F} \tag{3.24}$$

where matrix $\boldsymbol{B}$ contains the PLS regression coefficients stored by columns.

The Variable Importance to Prediction coefficient (Equation 3.25) informs about the influence of the predictor variable $\boldsymbol{x}_k$ in a PLS model of A PCs.

$$VIP_k = \sqrt{\frac{\sum_{a=1}^{A} (SSY_a \cdot w_{ak}^2 \cdot K)}{SSY \cdot A}} \tag{3.25}$$

SSY_a is the sum of squares of explained variance for the a-th component, and SSY is the sum of squares of the explained variance by the model with A components.

All PLS model parameters can be calculated sequentially using the NIPALS algorithm [60], which also handles missing data. This makes the PLS an attractive tool for analyzing complex databases. Moreover, when the response variable is categorical, there is an adaptation of PLS that can be used for discriminant and classification purposes. This version is named PLS-Discriminant Analysis (PLS-DA) [62].

Along this thesis, PLS coefficients are often expressed by their associated jackknife intervals. Jackknife [63], [64] is an approach to compute confidence intervals for a given estimate without making assumptions about the parameter's distribution but about the parameter's bias.

Let $\boldsymbol{x}$ be a vector with N values of a certain variable X, and θ the parameter of interest about X distribution. The term $\hat{\theta}_{(i)}$ refers to the value of the parameter θ obtained when the element x_i is removed from $\boldsymbol{x}$, and $\hat{\theta}_{(.)}$ is the average of all $\hat{\theta}_{(i)}$ values. The pseudovalues of the estimator are:

$$\tilde{\theta}_i = N\hat{\theta} - (N-1)\,\hat{\theta}_{(i)} \tag{3.26}$$

where $\hat{\theta}$ is the value of the parameter estimated from the N observations of the sample $\boldsymbol{x}$. Therefore, a jackknife estimation systematically removes each observation from the data matrix, computes the estimate of interest, and provides confidence intervals for that estimate, considering the standard deviation between each iteration's estimate and the global estimate with all the observations. The confidence intervals are calculated using the single-point estimate of the parameter (Equation 3.27) and its estimated variance (Equation 3.28).

$$\hat{\theta}_{JACK} = \frac{1}{N}\sum_{i=1}^{N} \tilde{\theta}_i \tag{3.27}$$

$$\hat{\sigma}^2_{\theta,JACK} = \frac{1}{N-1}\sum_{i=1}^{N} \left(\tilde{\theta}_i - \hat{\theta}_{JACK}\right)^2 \tag{3.28}$$

Using Equations 3.27 and 3.28, a confidence interval for the parameter can be provided as:

$$CI(\theta)_{100(1-\alpha)\%} = \left[\hat{\theta}_{JACK} - t_{\alpha/2,N-1}\frac{\hat{\sigma}_{\theta,JACK}}{\sqrt{N}}; \hat{\theta}_{JACK} + t_{1-\alpha/2,N-1}\frac{\hat{\sigma}_{\theta,JACK}}{\sqrt{N}}\right] \tag{3.29}$$

where $t_{\alpha,N-1}$ is the value of the normal distribution with $N-1$ degrees of freedom for which $P(t > t_{\alpha,N-1}) > \alpha$.

PLSDA

Partial Least Squares for Discriminant Analysis (PLS-DA) is the direct extension of PLS, developed for classification problems [62]. First of all, the response matrix $\boldsymbol{Y}$ must be one-hot encoded. This encoding consists of having, for each observation, a dummy L-dimensional row vector with "1" on the columns of the class the observation belongs to, and "0" otherwise in the rest of $L-1$ columns, with L being the number of categories. Then, matrix $\boldsymbol{X}$ is regressed via PLS on $\boldsymbol{Y}$.

What the PLS equations (Equations 3.18 to 3.20) will obtain for new observations will actually be a class prediction that can be interpreted in a non-strict way as the probability, given the input values $\boldsymbol{x}^\top$, of belonging to each one of the L classes. The assignation to return a categorical output respecting the original nature of the classification problem can be carried out according to different rules: assigning the label with the highest probability a posteriori, assigning the label provided that it is above a certain threshold, etc.

3.4.2 kernel Partial Least Squares Regression

Section 3.4.1 exemplified how to adapt the PLS framework to deal with categorical response variables, overcoming the limitation of assuming a continuous response. In an analogous way, kernel PLS focuses on another limitation, the assumption of that latent variables should be linear combinations of the original features. To bypass this limitation, a prior kernel transformation is applied to the data expanding the dimensionality of the original feature space. The kernel transformation is given by:

$$K(\boldsymbol{x}_i^\top; \boldsymbol{x}_j^\top) = \langle x_i^\top; x_j^\top \rangle \tag{3.30}$$

where $\boldsymbol{x}_i^\top$ and $\boldsymbol{x}_j^\top$ are two row vectors of the original data matrix to which a specific mapping function is applied, while $\langle$ and $\rangle$ denote the inner product.

If one applies this transformation to every possible couple of vectors constituting a generic array, $\boldsymbol{X}$, with dimensions $N \times K$, it will be converted into a squared symmetric $N \times N$ kernel matrix, $\boldsymbol{K}$, whose elements constitute dissimilarity or distance measurements between two different observations.

When dealing with kernel-based approaches, it is not needed to know the mapping function a *priori*. There are many generic kernel functions one can resort to for obtaining $\boldsymbol{K}$ and all of them exhibit two fundamental properties: i) they allow the original data to be projected onto a higher-dimensional space, the feature space; ii) they provide a way to calculate the inner product between observations in such a feature space.

The former permits to describe in a linear way possible non-linear relationships in $\boldsymbol{X}$. The latter makes all the algorithms of classical multivariate linear methodologies, which are based on the calculation of the inner product matrix of $\boldsymbol{X}$ (e.g. PCA, PLS and PLSDA), suitable for being applied in this higher-dimensional feature space.

Among the available kernels, in this thesis it will be particularly used the Gaussian on, also known as the radial basis function kernel.

$$exp\left(-\frac{||\boldsymbol{x}_i\boldsymbol{x}_j||^2}{2\sigma}\right) \tag{3.31}$$

Therefore, once the kernel matrix, $\boldsymbol{K}$, has been computed, a classical bilinear technique can be applied to it generating the classical model. Although the use of a previous kernel distorts in a certain way the interpretability of the model, it is possible to retrieve the inner connections between responses and predictors by a pseudosampling approach [65].

3.4.3 Random Forest

Random Forests (RF, [66]) is a type of ensemble model based on CART (Classification And Regression Trees, [67]) and on bagging. The CART algorithm is a machine learning technique that emulates a sequential binary decision-making process by fitting classification or regression trees. At the initial or root node, all observations remain together. Then, the CART algorithm will

generate internal nodes, splitting the data set into two new nodes. Internal nodes generate the partition, asking for specific information on the predictors.

In each node, the variable selection and the threshold to split the data are made according to the Gini index, which measures the node's entropy (also called heterogeneity or impurity). Gini's entropy index of a given note t is calculated by summing the probability p_i of each item being chosen times the probability $1 - p_i$ of a mistake in categorizing that item.

$$I(t) = \sum_{i=1}^{N} p_i \left(1 - p_i\right) \tag{3.32}$$

Thus, it reaches the zero value when all cases in the node fall into a single target category.

The CART algorithm stops growing the tree when a leaf node is reached. Leave nodes can be defined by meeting certain algorithm hyperparameters, such as having reached the maximum number of tree partitions or if the number of observations in that node is below a certain number.

One of the main advantages of the CART algorithm is that it can model non-linear problems and still be interpretable. However, complex issues usually resulted in long and complicated decision trees, which were difficult to interpret. Moreover, the CART algorithm required a phase named "pruning" to reduce the number of partitions of the decision tree to avoid overfitting.

Random Forests overcame these limitations by implementing resampling techniques that allowed the use of the Law of Large Numbers. A set of decision trees defines a Random Forest model fitted utilizing a subset of observations and a subset of features (Equation 3.33).

$$\left\{h\left(\boldsymbol{x}, \Theta_k\right), k = 1, \ldots\right\} \tag{3.33}$$

For each tree, bootstrapping [68] is done to sample with replacement N observations from the initial data set, with an independent and identically distributed probability of each observation being selected. Moreover, a random selection of variables is also performed for each tree. Then, for a new observation, the RF obtains a prediction from each one of its trees and assigns the most voted response.

In the original work proposing RF, it is proved that the combination of enough trees ensures RF convergence and prevents over-fitting problems. Moreover, the bootstrapping implementation also enables an unbiased prediction error estimation. Observations not used for a subset of trees are known as out-of-bag observations. Analogously, for each observation, there will be a subset of trees that did not use it for their fitting, i.e., that observation remained out-of-bag for such trees. This allows using these trees to obtain a set of predictions for the same observation.

This *out-of-bag* estimates can present certain advantages to computing the uncertainty within the predictions, which remain biased in other resampling schemes such as cross-validation. Moreover, when the number of cases is reduced, the out-of-bag estimation would allow the prediction error computation as in an external test set without performing the classical partition within the training and test set.

Regarding interpretability, RF cannot return a single decision-making structure as the CART algorithm does. However, the out-of-bag procedure also enables estimating metrics about the importance of each variable, known as *Variable Importance for the Prediction* (VIP).

To compute variables' VIP, a given tree's out-of-bag observations are permuted on a single predictor $\boldsymbol{x}_j$. These corrupted observations on the $\boldsymbol{x}_j$ variable are then run down the tree, obtaining a particular decision. This procedure is repeated for each one of the K variables. When some variable is corrupted, the outcomes obtained for each observation are compared to the observations' true labels (or responses). The increase in the misclassification rate when a variable is corrupted is compared to the out-of-bag misclassification rate. This increase, expressed as a percentage, provides a measure of the importance of the variable.

While the out-of-bag estimation allows computing prediction errors and estimating variable importance without the need for a separate test set, an essential aspect to note is that within the process of estimating VIP coefficients, a single predictor $\boldsymbol{x}_j$ is permuted among a tree's out-of-bag observations. Therefore, it's important to acknowledge that corrupting individual variables in this manner creates observations that may disrupt the correlation structure of the dataset, essentially generating new instances that significantly differ from those in the training dataset. This introduces a potential limitation to the method as it attempts to predict observations that could essentially be outliers or anomalous data points, altering the typical data structure.

3.4.4 Evaluation of supervised models

This section introduces some notions used throughout the thesis, namely in Chapter 6 from Part II, but especially Part III, to compare across ML models. The terms upon which this comparison will be established will be divided into two main types: those describing model performance according to the traditional evaluation for supervised models and those reporting about the interpretability of the models in terms of the inferred relationship between input and output variables.

Comparison of models' performance

The following lines describe metrics used to evaluate the performance of supervised models trained for classification problems. The term TP refers to True Positives, FP to False Positives, TN to True Negatives, and FN to False Negatives:

- The recall (Sensitivity or True Positive Rate) measures the proportion of positive instances correctly identified by the model. It is calculated as the ratio of true positives to the sum of true positives and false negatives. A high recall indicates that the model is good at capturing positive cases, reducing the number of false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{3.34}$$

- The specificity (True Negative Rate) measures the proportion of negative instances correctly identified by the model. It is calculated as the ratio of true negatives to the sum of true and false positives. High specificity means that the model is effective at identifying negative cases, reducing the number of false positives.

$$Specificity = \frac{TN}{TN + FP} \tag{3.35}$$

- The precision (Positive Predictive Value) measures the proportion of positive predictions made by the model that are correct. It is calculated as the ratio of true positives to the sum of true and false positives. A high precision indicates that the model is making few false positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{3.36}$$

- The Area Under the Curve (AUC, Equation 3.37) refers to the area under the Receiver Operating Characteristic (ROC) Curve. The ROC curve is a graphical representation of the model's performance across different classification thresholds. The AUC is the area under the ROC curve and represents the overall performance of the model. A higher AUC value (closer to 1) indicates better discrimination power, i.e., the model is better at distinguishing between positive and negative cases. The AUC is typically calculated by adding successive trapezoid areas below the ROC curve.

$$AUC = \int_0^1 TPR(x)dx, \quad x = 1 - TNR \tag{3.37}$$

- The accuracy (Equation 3.38) quantifies the proportion of correctly predicted instances (both true positives and true negatives) out of the total instances in the dataset. In essence, accuracy indicates how well a model predicts the correct class labels for the given data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.38}$$

  While accuracy is a straightforward and easy-to-understand metric, it might not be suitable for imbalanced datasets where one class significantly outweighs the other, as it can lead to misleadingly high scores. In such cases, complementary metrics like the F1-score or the Matthew's Correlation Coefficient, explained below, are often used to provide a more comprehensive assessment of a model's performance.

- The F1-score (Equation 3.39) is calculated as the harmonic mean of precision and recall and considers both false positives and false negatives, making it particularly useful when dealing with imbalanced datasets.

$$Accuracy = \frac{2TP}{2TP + FP + FN} \tag{3.39}$$

It ranges between 0 and 1, where a higher F1-score indicates a better balance between precision and recall, suggesting a model that provides both accurate positive predictions and captures a substantial portion of actual positives.

- Matthew's Correlation Coefficient (MCC, Equation 3.40) takes into account true positives, true negatives, false positives, and false negatives, providing a balanced evaluation of classification performance, especially for imbalanced datasets, as it doesn't inflate scores due to skewed class distribution.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}} \tag{3.40}$$

  It considers all four outcomes of a binary classification and produces a score between -1 and 1, where 1 indicates perfect predictions, 0 indicates random predictions, and -1 indicates complete disagreement between predictions and actual outcomes.

- Cohen's Kappa (κ, Equation 3.41) is a statistic that measures the agreement between the model's predictions and the actual outcomes while considering the agreement that could be expected by chance. It is particularly useful when dealing with imbalanced datasets. A value of $\kappa = 1$ indicates perfect agreement, 0 indicates agreement by chance, and negative values indicate poor agreement.

$$\kappa = \frac{2(TP \cdot TN - FN \cdot FP)}{(TP + FP)(FP + TN) + (TP + FN)(FN + TN)} \tag{3.41}$$

These metrics are commonly used to assess the performance of classification models and can provide valuable insights into the model's ability to classify positive and negative instances correctly. It is essential to consider these metrics together to get a comprehensive understanding of the model's strengths and weaknesses.

Comparison of models' interpretation

Interpreting machine learning models is crucial for understanding their decision-making processes and gaining insights into the relationships between input features and predictions. Model interpretability ensures that the predictions are not treated as black boxes, allowing users to trust, validate, and improve the model. There are two broad categories of interpretability metrics: model-specific and model-agnostic methods.

Model-specific interpretability metrics aim to elucidate the decision-making process and feature importance within specific machine learning models. These metrics provide insights into the influence of input features on model predictions, and they are tailored to the characteristics of each model type. We categorize these metrics into three distinct groups based on their primary reporting focus:

- Reporting about Magnitude and Direction. This type of metric includes coefficients and weights such as those from linear models (e.g., Linear Regression, Logistic Regression, or the $\boldsymbol{B}$ coefficients from the PLS model, see Section 3.4.1). These coefficients represent the magnitude and direction of the relationship between each feature and the target variable. Positive coefficients indicate a positive correlation, while negative coefficients imply a negative one. The larger the magnitude of a coefficient, the stronger the impact of the corresponding feature on the model's predictions. Another example is the support vectors from Support Vector Machines (SVM), which enclose a subset of data points defining the decision boundary. Identifying the support vectors reveals the most influential data points significantly affecting the model's decision-making process. Analyzing the weights assigned to these support vectors offers insights into the impact of specific data instances on the final classification.
- Reporting about Magnitude. Partially similar to the previous type of metrics, these metrics report the influence of different variables on a certain objective function. For instance, in the CART algorithm, the Variable Importance for the Prediction metrics evaluate input features based on their ability to reduce impurity (e.g., Gini impurity or entropy). By measuring the decrease in impurity caused by each feature, Decision Trees rank features according to their importance. High-ranking features have a greater magnitude and influence in determining the model's predictions.
- Reporting about the Decision-Making Process. This final category refers to schematic visualizations of the overall decision-making process, en-

abling a certain degree of interpretability without directly quantifying the relationship between inputs and outputs. For instance, the inherent structure of Decision Trees provides a transparent representation of the decision-making process. Visualizing the tree reveals the sequence of feature tests performed to arrive at a prediction. Each node in the tree corresponds to a feature test, and each branch represents the outcome of the test, culminating in a final prediction at the leaf nodes.

Model-agnostic interpretability metrics provide insights into machine learning models' behaviour independent of their specific architecture. These metrics apply to many models, including complex black-box models. Their philosophy is to perform a Sensitivity Analysis, generating hypothetical scenarios to answer "what-if" questions posed to the model. For instance, a strategy is to calculate the permutation-based feature's importance. This method evaluates feature importance by randomly permuting the values of a single feature across the dataset and measuring the subsequent drop in model performance. Features that lead to a significant decrease in performance when permuted are considered important to the model's predictions.

Another example is pseudosamples, which involve creating hypothetical samples with different feature values while fixing other features. By observing how the model's predictions change in response to these perturbations, the features' importance and the model's sensitivity to variations in input features can be assessed. Using this concept, the variations of the model can be reported in different ways:

- Global feature importance can be indicated by consistent trends across the entire dataset, visualized via Partial Dependency Plots (PDP) [69]. This tool explores the impact of a single feature on the model's predictions by plotting the feature's values against the corresponding model prediction. By doing so, PDPs reveal descriptively the direction and magnitude of the relationship between the feature and the target variable.
- Local Interpretable Model-agnostic Explanations (LIME) approximate the behaviour of any black-box model near a specific prediction using a simpler, interpretable model [70], [71]. By understanding the interpretable model's predictions for individual instances, LIME provides local explanations, allowing users to comprehend the factors contributing to specific model outputs. For example, Individual Conditional Expectation (ICE) plots extend the idea of PDPs to offer a local perspective. Instead of showing the average impact of a feature, ICE plots illustrate

the model's predictions for each instance separately. By examining ICE plots for different instances, users can understand how the model treats individual data points based on their feature values.

In this thesis, univariate pseudosamples appear in Chapters 9 and 11.

3.5 Missing Data

Missing Data (MD) refers to open entries in a data set. A data set is a matrix where rows are observations and columns represent variables measured for each observation. Missing data is ubiquitous, especially in multivariate data sets, where several variables might be missing for some observations. Most tools for data analysis assume that they will work with a complete and clean database. Thus, a proper imputation of missing data is mandatory in most cases. In this section, the missing data problem will be described, and imputation methods will be addressed.

3.5.1 The missing data problem

The first step when dealing with missing data is to identify the missing data pattern and mechanism. On the one hand, the missing data pattern refers to the structure of missing entries within a matrix. There are four types of missing data patterns:

- Unstructured missing data. Missing data are randomly distributed among the matrix.
- Univariate missing data. Only one variable presents missing data. For instance, if some measurements of a certain experiment could not be obtained.
- Block-wise missing data. This pattern is the multivariate version of the previously mentioned missing data patterns: when missing values are concentrated among various variables, which might be more difficult to measure than others.
- File matching missing data. When certain variables do not share any observed values, since there are no cases with joint information between these features, the imputation of missing entries cannot rely on parameters that relate to these variables.

Once missing data are present in a matrix, the only way to solve the missing data problem is to stop having missing values. This means either working only with observed data or imputing missing values. However, working only with complete observations (Complete Case analysis, CC) results in a considerable loss of information, and using all observed values of each variable (Available Case analysis, AC), changes the sample used to compute estimates depending on the variables. These two drawbacks of CC and AC analyses are worsened in multivariate datasets where a whole observation might be deleted by just having a single missing value and where high correlations worsen the instability caused by changing sample size.

Therefore, neither CC nor AC is optimal for multivariate datasets, making imputation approaches attractive. Still, the applicability of imputation techniques relies on the missing data mechanism defining the relationship between values and their missingness. Little [72] used the following nomenclature for missingness in terms of probability models for $\boldsymbol{M}$, the missingness indicator matrix of the same dimensions as $\boldsymbol{X}$, with 1s in missing entries and 0 otherwise:

- Missing completely at random (MCAR). In this scenario, there is not any relationship between the missingness of a cell and its value. That is: $P\left(\boldsymbol{M}|\boldsymbol{X},\boldsymbol{\phi}\right) = P\left(\boldsymbol{M}|\boldsymbol{\phi}\right), \forall \boldsymbol{X}, \boldsymbol{\phi}$, where $\boldsymbol{\phi}$ are the parameters describing the missing pattern.
- Missing at random (MAR). In this case, missingness depends on the observed part of data, that is $P\left(\boldsymbol{M}|\boldsymbol{X},\boldsymbol{\phi}\right) = P\left(\boldsymbol{M}|\boldsymbol{X}^*,\boldsymbol{\phi}\right), \forall \boldsymbol{X}^*, \boldsymbol{\phi}$.
- Not missing at random (NMAR). The probability of an entry being missing depends on its unobserved value. This can happen when measurements are below or above the detection threshold of the recording device or in surveys when the answer is missing because of the implications of that given response.

If data presents NMAR missingness, the missing data generation mechanism is Nonignorable (NI) and has to be modelled alongside the imputation model. Since a numerical analysis of missing values is not feasible, a proper understanding of the data, its collection, and some expertise on similar cases can help study this issue.

On the contrary, if the reason why data is missing is ignorable (MCAR or MAR scenarios), there is no need to model the missing data mechanism, and the imputation can rely only on the observed relationship between variables. In such a case, there are two different scenarios. If an already-existent model

can be used to impute incomplete cases, the problem can be undertaken by a Model Exploitation (ME) framework. However, finding missing values in all rows is widespread when a model is unavailable. Such a case, when the imputation model has to be estimated despite missing values, is known as the Model Building (MB) framework.

The following section will explain some missing data imputation methods, usable when ignorable missingness is present. Afterwards, multivariate missing data imputation methods based on Principal Component Analysis models will be described, mentioning both their MB and ME implementations and focusing particularly on the results that justify the use of the Trimmed Scores Regression (TSR) algorithm in Chapter 6.

3.5.2 Missing data imputation methods

Missing data imputation aims to provide a plausible value for missing entries. This estimation is done by a prediction based on the observed data. There are many strategies for missing data imputation. Still, they are divided into two categories: single imputation methods providing one estimate per missing value and multiple imputation methods providing several estimates per missing value.

Single imputation

These algorithms predict one value for each missing entry, returning a complete data matrix. Nonetheless, this does not come free of risks. This subsection describes four different strategies sorted by increasing complexity.

The simplest, most risky and least advised option is to perform Unconditional Mean Imputation (UMI), which involves imputing missing values of a variable using the average value of observed values for that variable. This method does not consider the correlation between variables, and the estimated covariance matrix from the imputed data set can be seriously distorted.

Conditional Mean Imputation (CMI) considers the correlation between variables by predicting missing values as a regression based on the observed variables. However, this imputes the mean of the regression, underestimating the variability naturally present in the data. Methods like stochastic regression tried to overcome this by adding some variability to the estimations by drawing samples from an assumed distribution for the residuals. Nonetheless, all these methods would require adjustments if the covariance matrix of the ob-

served values is ill-conditioned since it must be inverted to obtain the missing estimates.

Finally, iterative algorithms are based on the Expectation Maximization (EM) approach. Under the assumption of multivariate normality, let $\boldsymbol{X}$ be a matrix whose distribution is characterized by parameters $\boldsymbol{\Theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. Dempster, Laird and Rubin introduced the term EM [73] to describe an iterative procedure where at each iteration s:

1. The M step performs maximum likelihood estimation (MLE) of $\boldsymbol{\Theta}^{(s)}$ as if there were no missing data.
2. The E step estimates missing data from the observed data, yielding an imputed dataset $\boldsymbol{X}^{(s)}$, which will be used for the M step of the $s+1$ iteration.

Under general conditions, each iteration increases the likelihood $L\left(\boldsymbol{\Theta}|\boldsymbol{X}^{*}\right)$ yielding reliable convergence after some iterations to a local or global maximum.

When $\boldsymbol{X}$ contains MAR or MCAR missing values, likelihood-based inferences about $\boldsymbol{\Theta}$ will be unaffected by the missing pattern $\boldsymbol{M}$ [74]. This allows us to safely ignore the missing data mechanism to estimate $\boldsymbol{\Theta}$. Still, as a drawback, EM algorithms slow their convergence speed as the percentage of missing values increases. Besides, the expectation step can be vulnerable to ill-conditioning or singularity problems. However, this second issue can be overcome by choosing a prediction method based on biased regression, such as the ones explained in Section 3.5.3.

Multiple imputation

One drawback shared by most single imputation methods is the underestimation of the variability, which a single imputation of missing values cannot reflect. Multiple Imputation tackles this problem by creating several imputations for each missing value. This is done by sampling M different sets of observations used to fit the imputation model. Two sources to estimate variability are provided with MI, the variability within estimates from each data set and the variability across the M complete data sets, which can be combined to obtain confidence intervals for the estimates [75].

The most known MI method is the Data Augmentation algorithm [75]. It relies on a two-step procedure consisting of an Imputation step, which estimates $\boldsymbol{X}^{\#(s)}$ from the conditional distribution given $\boldsymbol{X}^*$ and $\boldsymbol{\Theta}^{(s-1)}$, and a Posterior step, which estimates $\boldsymbol{\Theta}^{(s)}$ from $\boldsymbol{X}^{\#(s)}$ and $\boldsymbol{X}^*$. The initial $\boldsymbol{\Theta}^{(0)}$ estimates are usually obtained by EM. The algorithm is run until the convergence of the estimate's distribution. In Model Building contexts, when the distributions must be estimated from matrices with missing values, the DA algorithm is not so used since it still implies the inversion of the covariance submatrix corresponding to the known variables given an observation, which might not be feasible as data sets with high numbers of variables can yield singular covariance submatrices.

The Multiple Imputation with Chained Equations (MICE) algorithm is also well known [76]. This method performs imputation variable-wise, which means that an equation is used to predict each variable according to its relationship with the rest of the variables within the data set. This framework presents some advantages, such as the possibility of dealing with different prediction methods according to the nature of each column of the matrix, i.e., using a classical or logistic regression for continuous and binary variables, respectively.

3.5.3 PCA models for missing data imputation

When dealing with multivariate datasets, PCA has been extensively and successfully used to impute missing values. This section will summarize the available work on this approach until reaching the proposal of the Trimmed Scores Algorithm for PCA-MB. In all the following expressions, the data $\boldsymbol{X}$ is assumed to be already centred and scaled, and each one of its observations can be seen as a vector $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}^* & \boldsymbol{x}^\#\end{bmatrix}$, formed by the concatenation of its $K - R$ observed values and its R missing ones.

Model Exploitation

The first attempts to use PCA for missing data imputation started on the PCA-ME case, based on a known PCA model. Arteaga and Ferrer compared several approaches for missing data imputation with PCA-ME [77]:

- The Trimmed Scores (TRI) method estimates $\boldsymbol{x} = \boldsymbol{P}\boldsymbol{\tau} + \boldsymbol{e}$, where $\boldsymbol{\tau} = \boldsymbol{P}^\top \boldsymbol{x}$, and $\boldsymbol{x}^\#$ has missing entries imputed as the unconditional mean. This method is efficient and simple but can yield large errors if variables with important loadings are missing.

- The Single Component Projection (SCP) method proposed by Nelson, Taylor and MacGregor [78] is based on the NIPALS algorithm, which calculates the principal components sequentially. In each iteration to obtain a new component a, the measured and unexplained part of observation is modelled as $\boldsymbol{x}^*_{a-1} = \tau \boldsymbol{p}^*_a + \boldsymbol{e}^*_a$. The minimization of the error part yields $\hat{\tau}_a = \boldsymbol{p}^{*\top}_a \boldsymbol{x}^*_{a-1} / \left(\boldsymbol{p}^{*\top}_a \boldsymbol{p}_a\right)$ as the least squares estimator of the scores based on the measured values. The part of the observation explained by the new component is then subtracted, repeating the process. This method can yield relevant estimation errors that will be passed over the iterations if they start being large for the computation of the first scores.

- The Projection to the Model Plane method (PMP) is a method to obtain all the scores at one proposed by Wold [79] and Martens and Naes [80]. Its estimator is obtained as: $\hat{\boldsymbol{\tau}} = \left(\boldsymbol{P}^{*\top}\boldsymbol{P}^*\right)^{-1} \boldsymbol{P}^{*\top}\boldsymbol{x}^*$. Therefore, it relies exclusively on the observed part of each observation without attributing any value, as in the TRI method. Its one-shot approach to computing the PCs prevents error propagation from SCP. However, the inversion of $\boldsymbol{P}^{*\top}\boldsymbol{P}^*$ can be threatened if loading vectors are nearly collinear, which can happen as well in TRI and SCP, given the distortion on the orthogonality introduced by missing values. Arteaga and Ferrer [77] proved that PMP was equivalent to two other algorithms:

  - The Iterative Algorithm from Walczak and Massart [81], which initially assumes the unconditionally mean imputation from TRI, but re-iterates the estimation of the missing part until reaching its convergence.

  - The Minimization of the Squared Prediction Error (SPE) method aims to minimize the SPE of observations as a function of their missing part.

- The regression-based methods proposed by Arteaga and Ferrer [82] suggest fitting the regression model $\boldsymbol{X}^{\#} = (\boldsymbol{X}^*\boldsymbol{L})\,\boldsymbol{B} + \boldsymbol{U}$ defining the imputation as a function of a key matrix $\boldsymbol{L}$. They compared two alternatives for this regression:

  - The Known Data Regression (KDR) method uses $\boldsymbol{L}_{KDR} = \boldsymbol{I}_{K-R}$. Therefore:

$$\boldsymbol{B}_{KDR} = \left(\boldsymbol{X}^{*\top}\boldsymbol{X}^*\right)^{-1} \boldsymbol{X}^{*\top}\boldsymbol{X}^{\#} = \left(\boldsymbol{S}^{**}\right)^{-1} \boldsymbol{S}^{*\#} \tag{3.42}$$

And the final estimation of the missing values for an observation is given by regression on its available values:

$$\boldsymbol{x}^{\#} = \boldsymbol{S}^{\#*} \left(\boldsymbol{S}^{**}\right)^{-1} \boldsymbol{x}^{*} \tag{3.43}$$

– The Trimmed Scores Regression (TSR) performs the regression on the observed scores, as $\boldsymbol{X}^{\#} = \boldsymbol{T}^{*}\boldsymbol{B} + \boldsymbol{U} = \left(\boldsymbol{X}^{*}\boldsymbol{L}\right)\boldsymbol{B} + \boldsymbol{U}$, using $\boldsymbol{L}_{TSR} = \boldsymbol{P}^{*}$. Therefore:

$$\begin{aligned} \boldsymbol{B}_{TSR} = &\left[\left(\boldsymbol{X}^{*}\boldsymbol{P}^{*}\right)^{\top}\left(\boldsymbol{X}^{*}\boldsymbol{P}^{*}\right)\right]^{-1}\left(\boldsymbol{X}^{*}\boldsymbol{P}^{*}\right)^{\top}\boldsymbol{X}^{\#} = \\ &\left(\boldsymbol{P}^{*\top}\boldsymbol{X}^{*\top}\boldsymbol{X}^{*}\boldsymbol{P}^{*}\right)^{-1}\boldsymbol{P}^{*\top}\boldsymbol{X}^{*\top}\boldsymbol{X}^{\#} = \\ &\left(\boldsymbol{P}^{*\top}\boldsymbol{S}^{**}\boldsymbol{P}^{*}\right)^{-1}\boldsymbol{P}^{*\top}\boldsymbol{S}^{*\#} \end{aligned} \tag{3.44}$$

And the final estimation is given by regression on the scores of the available values of the observations:

$$\begin{aligned} \boldsymbol{x}^{\#} = &\boldsymbol{S}^{\#*}\boldsymbol{P}^{*}\left(\boldsymbol{P}^{*\top}\boldsymbol{S}^{**}\boldsymbol{P}^{*}\right)^{-1}\boldsymbol{P}^{*\top}\boldsymbol{x}^{*} = \\ &\boldsymbol{S}^{\#*}\boldsymbol{P}^{*}\left(\boldsymbol{P}^{*\top}\boldsymbol{S}^{**}\boldsymbol{P}^{*}\right)^{-1}\boldsymbol{\tau}^{*} \end{aligned} \tag{3.45}$$

The results of their comparison showed that regression-based methods were statistically more efficient than the other methods studied. However, TSR required the inversion of a matrix of sizes $A \times A$, being much more efficient than the inversion of the covariance matrix in KDR of dimensions $(K-R)\times(K-R)$. This KDR issue can be solved using other biased regression methods, such as Principal Component Regression (PCR) or PLS.

Model Building

In this scenario, the PCA model must be estimated simultaneously as missing values are imputed. Regarding PCA-MB, two methods are frequently used by practitioners. The first consists of adapting the nonlinear iterative partial least squares algorithm (NIPALS) to ignore the missing data along its iterative regressions [83]. The second one is the adaptation by Walczak and Massart of the IA – mentioned above – for the MB case by filling in the missing data with the predictions obtained from previous PCA models, iterating until convergence [84].

Folch-Fortuny, Arteaga and Ferrer [85] proposed to adapt the IA method in PCA-MB by replacing the prediction of missing values and applying a regression method as in PCA-ME. Among the adapted regression methods, TSR was the one outperforming the rest. The TSR method can be summarised as follows:

1. Fit the elements of the regression models from Equations 3.42 and 3.44.
2. Estimate the missing part using equations Equations 3.43 and 3.45.

This process was repeated until reaching the convergence of the imputed values, which differed less than a defined tolerance after a certain number of iterations.

Results showed that the TSR method performed remarkably well, representing the best compromise solution among prediction quality and computation time across all data structures. From the rest of the methods analysed, DA and KDR performed well with thin data sets ($N > K$), but they were more time-consuming, and their performance worsened with fat data sets ($N < K$), where DA was unfeasible, and KDR yielded the worst performance. The KDR methods with PCR and PLS overcame the later fat data sets issue but were still more time-consuming. The other methods (NIPALS, PMP and NLP) showed convergence problems for high percentages of missing data.

As a result, the IA from Walczak and Massart was the only method, jointly with TSR, that could be applied to all data sets regardless of their dimensions and the percentage of simulated missing values. However, its performance level in all four data sets was statistically worse than TSR's.

3.6 Outliers

Once a model is obtained, it can raise the question: Is an observation normal according to that model, or is it an outlier? This section uses the framework provided by Latent Variable-based models, such as PCA or PLS, to inform about anomalous values and outliers.

3.6.1 Properties of outliers

An observation can be considered an outlier in terms of a PCA model, according to its values for the Squared Prediction Error (SPE) and the Hotelling's T^2 (T^2, or more specifically, T^2_A for a PCA model with A components). These statistics, obtained from the residuals and the scores respectively, offer complementary information about the distance of an observation to the PCA model and the majority of data. Ferrer provides a comprehensive explanation about the mathematical aspects of SPE and the T^2_A and their use to detect outlying observations [86].

The SPE is the squared Euclidean (perpendicular) distance from the observation $\boldsymbol{x}$ to the $A-$dimensional subspace of the model, that is $SPE = \boldsymbol{e}^\top \boldsymbol{e}$, where $\boldsymbol{e}$ is the error vector of the observation $\boldsymbol{x}$. From the previous expression, the SPE can be rewritten as $SPE = \boldsymbol{x}^\top (\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top)^\top (\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top)\boldsymbol{x}$. Since $(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top)$ is symmetric and idempotent matrix:

$$SPE = \boldsymbol{x}^\top \left(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top\right) \boldsymbol{x} \tag{3.46}$$

Assuming that residuals follow a multivariate normal distribution, Box, Jackson and Eriksson derived approximate distributions for such quadratic forms [87]–[89].

On the other hand, the Hotelling-T^2_A statistic for an observation is defined as

$$T^2_A = \boldsymbol{\tau}^\top \boldsymbol{\Theta}^{-1} \boldsymbol{\tau} = \sum_{a=1}^{A} \left(\tau_a^2 / \lambda_a\right) \tag{3.47}$$

where $\boldsymbol{\Theta}(A \times A)$ is the covariance matrix of $\boldsymbol{T}$ (diagonal matrix of the highest A eigenvalues $\{\lambda_1, \ldots, \lambda_A\}$). It represents the estimated squared Mahalanobis distance from the center of the latent subspace to the projection of an observation onto this subspace.

When diagnosing which variables yield the obtained values for the SPE and the T^2 it can be useful to check the contributions of each variable to each statistic [86].

From these two statistics (the SPE and the T^2), two complementary control metrics are obtained. Firstly, with an appropriate reference set of data, the in-control PCA model is built. The control limits are defined as well using the reference distributions for each statistic.

Regarding the Upper Control Limit (UCL) for the SPE, several procedures can be used. Jackson and Mudholkar showed that an approximate SPE critical value at significance level α is given by

$$UCL(SPE)_\alpha = \theta_1 \left[z_\alpha \sqrt{2\theta_2 h_0^2}/\theta_1 + 1 + \theta_2 h_0 (h_0 - 1)/\theta_1^2 \right]^{1/h_0} \tag{3.48}$$

where $\theta_k = \sum_{j=A+1}^{rank(\boldsymbol{X})} (\lambda_j)^k$, $h_0 = 1 - 2\theta_1\theta_3/3\theta_2^2$, λ_j are the eigenvalues of the PCA residual covariance matrix $\boldsymbol{E}^\top \boldsymbol{E}/(N-1)$, and z_α is the $100(1-\alpha)\%$ percentile of a standard normal variable [88].

Alternatively, one can use an approximation based on the weighted chi-squared distribution ($g\chi_h^2$) proposed by Box [87]. Nomikos and MacGregor suggested a simple and fast way to estimate parameters g and h which is based on matching moments between a $g\chi_h^2$ distribution and the sample distribution of SPE [90]. The mean ($\mu = gh$) and variance ($\sigma^2 = 2g^2h$) of the $g\chi_h^2$ distribution are equated with the sample mean (b) and variance (v) of the SPE sample. Hence, the Upper SPE Control Limit at significance level α is given by

$$UCL(SPE)_\alpha = v\chi^2_{(2b^2/v),\alpha}/(2b) \tag{3.49}$$

where $\chi^2_{(2b^2/v),\alpha}$ is the 100(1-α)% percentile of the corresponding chi-squared distribution with $2b^2/v$ degrees of freedom.

Upper Control Limits (UCL) for the T_A^2 at a significance level (type I) risk α can be obtained assuming that the statistic follows an F distribution

$$T_A^2 \sim A\left(N^2-1\right) F_{A,(N-A)}/(N\left(N-A\right)) \tag{3.50}$$

Thus, the corresponding UCL from Equation 3.50 is given by

$$UCL(T_A^2)_\alpha = A\left(N^2-1\right) F_{(A,(N-A)),\alpha}/(N\left(N-A\right)) \tag{3.51}$$

According to the aforementioned conceptual meaning of these multivariate statistics (SPE and T_A^2), observations above their associated UCL will be representing different types of outliers.

3.6.2 Contamination models

Most approaches used to define and simulate outliers assume the paradigm of rowwise outliers. This paradigm defines an outlier as a whole observation or $K-$dimensional row $\boldsymbol{x}^\top$ in a matrix $\boldsymbol{X}$ of $N \times K$ dimensions. Probably, the most famous model to define this situation is the classical Tukey-Huber Contamination Model (THCM) [49]:

$$X = (1 - B)\, Y + BZ \tag{3.52}$$

In these scenarios, the observed data $\boldsymbol{X}$ is thus a mix of unobserved distributions defining two different submatrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$, representing data from two diverse populations. The term B follows a binomial distribution $B \sim Bin(1, \epsilon)$ where ϵ is a random contamination indicator.

This "rowwise" contamination assumed by the THCM is also known as the Fully Dependent Contamination Model (FDCM), since for a given row, the probability of a cell being contaminated depends on the rest of the cells within the row.

The FDCM framework assumes, first, that the majority of the cases are free of contamination. Secondly, it also implies that contaminated observations should be discarded as a whole, as they come from a completely different population. This conceptual frame influenced the design of robust methods (see Section 3.3.2), relying on identifying a minority of contaminated cases, always assumed to be lower than the 50% of rows, which is the maximal breakdown point of robust covariance estimators [91].

However, these assumptions become limitations when dealing with more recent data sets, as they present more heterogeneous contamination. For instance, in high dimensional datasets, only a fraction of variables may be contaminated, and if so, ignoring such observations completely would be inconvenient, especially if $N < K$.

These limitations of fully-dependent contamination motivated the update of the contamination models, emphasizing the relevance of the so-called cellwise outliers. Cellwise outliers are entries with suspicious values caused by random events such as measurement errors [59], [92], [93]. This contamination model is called the Fully Independent Contamination Model (FICM).

Even a small proportion of these cellwise outliers can affect more than 50% of the observations, which is the maximum contamination that rowwise robust

methods can deal with, and this is critical considering that the cellwise contamination effect is even more pronounced in high-dimensional situations, where they are frequently found. As Alqallaf et al. show, data following FICM can severely upset standard robust procedures, even if the fraction of contaminated cells in the data is quite low [92].

Finally, it is also important to mention that the most likely scenario to be found in real datasets is the coexistence of both types of contamination, also known as the Partially Clean Independent Contamination model (PCICM). This model assumes that a case $\boldsymbol{x}^\top$ is free of contamination with a certain probability $1 - \alpha$ (as in the FDCM), but otherwise, its different K cells are independently contaminated with chance β.

Among the methods in the literature, the MacroPCA algorithm [48] is the only one dealing with a PCICM (cellwise and rowwise contamination) and missing data. Nonetheless, Chapter 6 describes the proposal of a new algorithm dealing with PCICM and missing data: the Robust Adaptation of Trimmed Squares Regression for Anomalous Rows and cells (RadarTSR).

Chapter 4

Material

4.1 Hardware

The computations and results of this work were performed in two units:

- MacBook Pro (Retina, 13-inch, Early 2015), CPU 2,7 GHz Dual-Core Intel Core i5 and 8 GB of RAM.
- LAPTOP-IV8D2E99, CPU 2,3 GHz Intel Core i7-105110U and 16 GB of RAM.

Each part of the thesis indicates the computer being used for its results.

4.2 Software

The OS used were:

- Mac OS versions from Catalina to Big Sur.
- Windows 10

Most calculations were obtained by using self-developed scripts and programs in the following environments, sorted by decreasing usage:

- Matlab from versions 2018b to 2020b.
- RStudio with R (versions from 2020 - 2022). For computations with *cellWise*, results from Sections 6, , 8 and [refSCOUTer].
- Python within Anaconda environment (versions from 2019).

Each part of the thesis indicates the environment and language being used for its execution.

Software packages developed in this thesis are:

- SCOUTer: Simulation of Controller OUTliers. It is available for R as a package submitted and admitted in the Comprehensive R Archive Network (CRAN), with unstable R and Matlab versions available in the GitHub repositories `https://github.com/albagc/SCOUTer.git` (Chapter 5).

- RadarTSR: Robust Adaptation for Datasets with Anomalous Rows and cells of Trimmed Scores Regression. It implemented in Matlab and it is available in the GitHub repository https://github.com/albagc/RadarTSR-matlab-master.git (Chapter 6).
- PLATERO: Plate Reader Operator. It is implemented in Matlab and it is available in the GitHub repository `https://github.com/sb2cl/PLATERO.git` (Chapter 10).

Other code developed without being part of a deployed software toolbox, is mentioned and linked along the thesis.

Other software packages playing a main role along the thesis are:

- ProSensus Multivariate (ProSensus, Inc.)
- CellWise package
- MDI Toolbox
- Minitab 2017
- SIMCA

More transversely used packages, such as ggplot2, miceAdds or ROxygen were part of several parts of the thesis, and they are properly cited in each of the corresponding Sections and contributions along this document.

4.3 Datasets

Different datasets were used to test the performance of the approaches being developed and compared in many parts of this work. Data sets are thoroughly described in each section, although a initial division can be done at this point:

- Simulated datasets: missing data with outliers (Chapter 6).
- Experimental datasets: Fluorescein measurements datasets (Chapter 10).
- Clinical datasets: COVID-19 national database of the Spanish Society of Hospital Pharmacy (Chapter 9), UFPE database (Chapter 7) and CFS (Chapter 8).

Part II

New methodological proposals

Chapter 5

SCOUTer: a standard framework to generate controlled outliers

Part of the content of this chapter has been included in:

> [94]González-Cebrián, A., Arteaga, F., Folch-Fortuny, A. & Ferrer, A. How to Simulate Outliers with the Desired Properties. *Chemometrics And Intelligent Laboratory Systems.* **212** (2021), https://doi.org/10.1016/j.chemolab.2021.104301.

5.1 Introduction

Principal component analysis (PCA) models (explained in depth in Section 3.3.1) are instrumental in the context of highly correlated data sets, given their dimensionality and noise reduction power. Its compression and interpretability make it widely used for Exploratory Data Analysis (EDA). When PCA is used in an EDA framework, a model is built, known as the PCA Model Building (PCA-MB) stage. In its basic definition, PCA uses least squares parameters, which outliers' influence can distort. However, specifically in the first stages of data analysis, such as EDA, it is common to have outliers within the data [48], [77], [95].

Several approaches that avoid this adverse effect have been proposed in the literature to deal with this issue, assembled in what is known as robust PCA methods. There are plenty of strategies to conduct PCA robustly. However, beyond the particularities of each proposal, what defines these algorithms is their ability to neglect the influence of potential outliers during the PCA-MB stage.

To develop methodological work on detecting and treating outliers, it is often helpful to simulate this type of anomalous data. Examining the literature, one can notice that the task of simulating the data sets and outliers in the framework of PCA-MB has been addressed differently [48], [93], [96], [97]. In general terms, all proposals are linked by their definition of outliers by setting the population parameters to which they belong. Thus, observations are classified as outliers because they are drawn from a distribution different from the one that describes the clean data.

However, it is not straightforward to establish the relationship between the chosen parameters for the outliers' distribution and the simulated observations' resulting properties. As a result, simulating observations with the desired distance from the reference data set by setting different parameters of the data distribution becomes practically unfeasible. Moreover, working with this simulation paradigm means making assumptions about the distributions describing the reference and outlying data set. Usually, a multivariate normal distribution is assumed, and the mean vector or the covariance matrix is altered to generate outlying observations. Yet, assuming a particular probability distribution might not be that simple in case one wants to simulate outliers for a real reference data set.

For these reasons, though the traditional paradigm is technically correct, we believe that one could further exploit the information offered by a PCA model

to generate outliers with more control of their properties based on two statistics: the Squared Prediction Error (SPE) and the Hotelling T^2 (also referred from now on simply as T^2). Following this conceptual framework, previously introduced in Section 3.6.1, this chapter proposes a standard framework for outliers definition and simulation based on its characterization in terms of the SPE and the T^2 statistics.

Firstly, the methodology to generate moderate and severe perturbations, based on shift directions of the SPE and the T^2, is explained in Section 5.2. Later on, in Section 5.3, the proposed variants of the algorithm to simulate outliers are introduced, and some examples of how to simulate controlled outliers are shown. Moreover, some practical applications will be provided in Section 5.3.2 to illustrate the potential of the proposed method as a standard framework to simulate outliers. In these examples, our procedure to simulate controlled outliers will be configured to emulate other strategies of outliers generation from the literature on PCA models. Additionally, the consistency of the outlying properties will be assessed by projecting our simulated outliers onto a robust PCA model. Finally, a summary of the main conclusions, including the proposed method's limitations, is provided in Section 5.4.

The Matlab code and documentation for outliers generation are available in the GitHub repository `https://github.com/albagc/SCOUTer.git`. Detailed code lines to reproduce the results from Section 5.3.1 are available in the *howto.pdf* document on the repository. Further details about references for the outliers simulation are also provided.

5.2 Algorithm to generate outliers with the desired properties

The proposed method to generate outliers is based on transforming an observation $\boldsymbol{x}$, with given SPE and T^2 values. This chapter will use a specific notation to refer to the SPE and T^2 values of different observations, where each observation will be indicated as subscripts of the statistics. Therefore, $SPE_{\boldsymbol{x}}$ and $T^2_{\boldsymbol{x}}$, refer to the SPE and T^2 of the observation $\boldsymbol{x}^\top$, respectively. The proposed method is based on the transformation of $\boldsymbol{x}^\top$, into a new observation $\boldsymbol{y}^\top$ with an SPE and/or T^2 values specified by the user ($SPE_{\boldsymbol{y}}$ and $T^2_{\boldsymbol{y}}$, respectively). The transformation will consist of a shift of the observation following a certain direction in the space of the original variables.

Moving the observation $\boldsymbol{x}$ in the direction $\boldsymbol{v}$ to obtain a new observation $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{v}$, we can calculate the new value of the SPE and the T^2 statistics, based on the original values:

$$\begin{aligned} SPE_{\boldsymbol{x}+\boldsymbol{v}} = (\boldsymbol{x}+\boldsymbol{v})^\top \left(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top\right)(\boldsymbol{x}+\boldsymbol{v}) = \\ SPE_{\boldsymbol{x}} + \boldsymbol{v}^\top \left(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top\right)(2\boldsymbol{x}+\boldsymbol{v}) \end{aligned} \tag{5.1}$$

$$T^2_{\boldsymbol{x}+\boldsymbol{v}} = (\boldsymbol{x}+\boldsymbol{v})^\top \boldsymbol{P}\boldsymbol{\Theta}^{-1}\boldsymbol{P}^\top(\boldsymbol{x}+\boldsymbol{v}) = T^2_{\boldsymbol{x}} + \boldsymbol{v}^\top \boldsymbol{P}\boldsymbol{\Theta}^{-1}\boldsymbol{P}^\top(2\boldsymbol{x}+\boldsymbol{v}) \tag{5.2}$$

The next issue is choosing the direction $\boldsymbol{v}$. An obvious choice is to shift the observation in the direction that joins it with the origin of coordinates in the original data space, taking $\boldsymbol{v} = c\boldsymbol{x}$. In this case, it is easy to calculate the change in both statistics:

$$SPE_{\boldsymbol{x}+c\boldsymbol{x}} = (1+c)^2 SPE_{\boldsymbol{x}} \tag{5.3}$$

$$T^2_{\boldsymbol{x}+c\boldsymbol{x}} = (1+c)^2 T^2_{\boldsymbol{x}} \tag{5.4}$$

However, directions of interest are those for which we can control the change that occurs in each statistic. For example, specific directions allow the change in one of both statistics without affecting the other.

In particular, we can move the observation in the direction of its residual vector in the PCA model so that a change in the SPE will occur without modifying the T^2:

$$\boldsymbol{v}_{SPE} = \boldsymbol{e} = \left(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top\right)\boldsymbol{x} \tag{5.5}$$

Similarly, we can move it in the direction that joins the projection of the observation on the model with the origin (i.e., the direction of the predicted observation $\hat{\boldsymbol{x}}$) so that there will be a change in T^2, without modifying the SPE:

$$\boldsymbol{v}_{T^2} = \boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x} \tag{5.6}$$

As both directions are orthogonal, we can compose both displacements in one operator, with control over the amount by which each of them increases. This will be illustrated in the following sections.

Shift of the SPE statistic

If we move the observation $\boldsymbol{x}$ in the direction from Equation 5.5 given by its residual vector (according to the PCA model), multiplied by a scalar a, we get, from Equations 5.1 and 5.2:

$$SPE_{\boldsymbol{x}+a(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^\top)\boldsymbol{x}} = SPE_{\boldsymbol{x}} + a\boldsymbol{x}^\top \left(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^\top\right)\left(2\boldsymbol{x} + a\left(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^\top\right)\boldsymbol{x}\right) = (1+a)^2 SPE_{\boldsymbol{x}} \tag{5.7}$$

$$T^2_{\boldsymbol{x}+a(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^\top)\boldsymbol{x}} = T^2_{\boldsymbol{x}} + a\boldsymbol{x}^\top \left(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^\top\right)\boldsymbol{P}\boldsymbol{\Theta}^{-1}\boldsymbol{P}^\top\left(2\boldsymbol{x} + a\left(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^\top\right)\boldsymbol{x}\right) = T^2_{\boldsymbol{x}} \tag{5.8}$$

We can choose the value a to achieve a target value for the SPE statistic, say $SPE_{\boldsymbol{y}}$:

$$(1+a)^2 SPE_{\boldsymbol{x}} = SPE_{\boldsymbol{y}} \rightarrow a = \sqrt{SPE_{\boldsymbol{y}}/SPE_{\boldsymbol{x}}} - 1 \tag{5.9}$$

Note that the selected direction is the one that maximizes the change in the SPE because the gradient of this statistic is: $\nabla(SPE)(\boldsymbol{x}) = 2\left(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^\top\right)\boldsymbol{x}$.

Shift of the T^2 statistic

If we move the observation $\boldsymbol{x}$ in the direction from Equation 5.6, multiplied by a scalar b, we get, from Equations 5.1 and 5.2:

$$SPE_{\boldsymbol{x}+b\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x}} = SPE_{\boldsymbol{x}} + b\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{P}^\top\left(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^\top\right)\left(2\boldsymbol{x} + b\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x}\right) = SPE_{\boldsymbol{x}} \tag{5.10}$$

$$T^2_{\boldsymbol{x}+b\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x}} = T^2_{\boldsymbol{x}} + b\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{\Theta}^{-1}\boldsymbol{P}^\top\left(2\boldsymbol{x} + b\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x}\right) = (1+b)^2 T^2_{\boldsymbol{x}} \tag{5.11}$$

We can choose the value b to achieve a target value for the T^2 statistic, say $T^2_{\boldsymbol{y}}$:

$$(1+b)^2 T^2_{\boldsymbol{x}} = T^2_{\boldsymbol{y}} \rightarrow b = \sqrt{T^2_{\boldsymbol{y}}/T^2_{\boldsymbol{x}}} - 1 \tag{5.12}$$

We can also select the direction that maximizes the change in the T^2 statistic without changing the SPE statistic, choosing the gradient of the T^2 statistic: $\nabla(T^2) = 2\boldsymbol{P}\boldsymbol{\Theta}^{-1}\boldsymbol{P}^\top\boldsymbol{x}$. We do not use this direction because it is difficult to parametrize the amount of change in the T^2 statistic.

Shift both statistics simultaneously

If we have an observation $\boldsymbol{x}$ with statistics $SPE_{\boldsymbol{x}}$ and $T^2_{\boldsymbol{x}}$, we can transform it into a new observation with statistics $SPE_{\boldsymbol{y}}$ and $T^2_{\boldsymbol{y}}$ combining the aforementioned transformations:

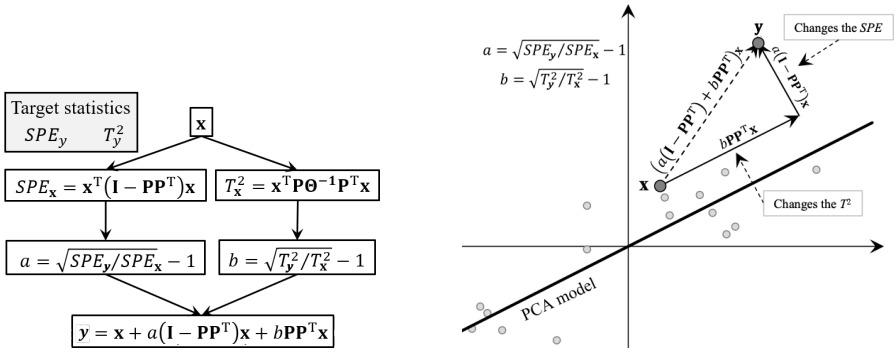
$$\boldsymbol{y} = \boldsymbol{x} + a\left(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top\right)\boldsymbol{x} + b\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x} \tag{5.13}$$

With $a = \sqrt{SPE_{\boldsymbol{y}}/SPE_{\boldsymbol{x}}} - 1$ and $b = \sqrt{T^2_{\boldsymbol{y}}/T^2_{\boldsymbol{x}}} - 1$, as seen in Equation 5.9 and Equation 5.12. The procedure to build a new observation with desired SPE and T^2 statistics, based on an arbitrary prior observation $\boldsymbol{x}$, is illustrated in Figure 5.1a. The visual representation of the algorithm with a model of only one PC for an original space with only two variables is represented in Figure 5.1b.

Furthermore, another aspect can be used to control the outlying behaviour of the new observations. Given the reference and target values of a statistic, one can generate a series of $M-1$ intermediate observations between the reference and the target one: $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{M-1}\}$. Mathematically, the expected value of a statistic H_m as a result of a transition from the reference H_0 to the target value H_M:

$$H_m = H_0 + (m/M)^\gamma (H_M - H_0) \qquad m = 1, 2, ..., M-1 \tag{5.14}$$

Thus, SPE_m and T^2_m will gradually change according to the number of steps and the spacing between them. This spacing is regulated in Equation 5.14 by the γ parameter. As it can be appreciated in Figure 5.2, when this parameter



(a) Scheme to generate a new observation with target statistics.

(b) Simple representation of the transformation from Equation 5.13.

Figure 5.1: Graphical representations of the transformation from the original observation $\boldsymbol{x}^\top$ to the observation $\boldsymbol{y}^\top$.

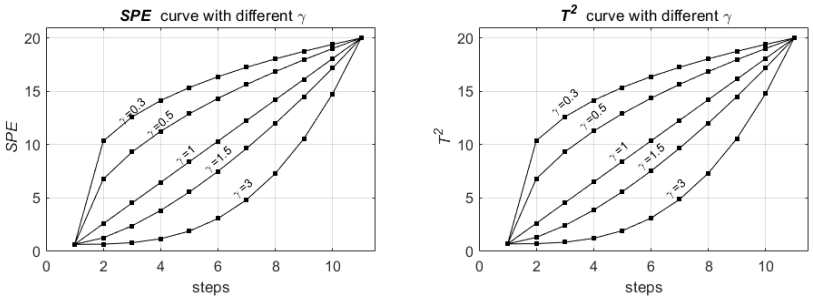


Figure 5.2: Curves for the SPE (left) and T^2 (right) statistics along the shift in 20 steps for different values of their spacing parameters γ.

is set to 1, the spacing between steps is linear, shifting towards a non-linear dynamic as it drifts from 1.

Given that both parameters (γ_{SPE} and γ_{T^2}) can be shifted simultaneously, this gives the user the flexibility to simulate a wider variety of trajectories for each possible combination of values along the spacing of the two parameters. Performing simultaneous shifts with some values for the parameters results in the curves of Figure 5.3.

This framework, including the possibility of controlling the distance between intermediate observations in a series of outliers, can help study and compare the sensitivity of different robust PCA approaches or methods for outlying

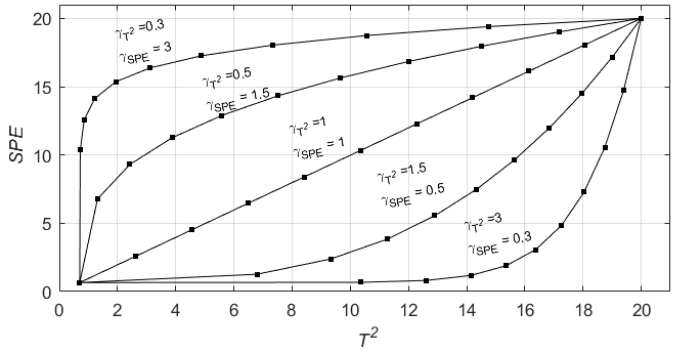


Figure 5.3: Curves for the SPE and T^2 statistics along the shift in 20 steps for different combinations of their γ parameters.

detection. Thus, one could know for what type of outliers and at which step one method performs differently. Finally, considering all these parameters, one has the complete flux diagram of the procedure in Figure 5.4.

If a given observation $\boldsymbol{x}$ is moved in different directions, it will be appreciated in the SPE and T^2 statistics and the scores. Figure 5.5 illustrates various shifts on a five-dimensional observation $\boldsymbol{x}$ according to a reference PCA model.

In Figure 5.5a, red dashed lines represent the UCL for the T^2 and SPE statistics. The ellipse defined in the score plot from Figure 5.5b is the contour curve of the confidence ellipsoid for the T^2 statistic, calculated for a confidence level of $(1-\alpha)\times 100\%$. From Equation 3.51, it is obtained an ellipsoid delimited in each dimension (i.e., PC) of the latent subspace. The contour of that ellipsoid represents a region of the space that holds $T^2 = T^2_{100(1-\alpha)\%CL}$ for each observation lying on that contour. Since the score plot is bi-dimensional, the bi-dimensional representation of the confidence ellipsoid turns into a confidence ellipse. Therefore, observations outside the ellipse will surpass the UCL for the T^2 statistic.

The first directions correspond to the five variables (x1, ..., x5). The trivial direction ($\boldsymbol{v} = \boldsymbol{x}$) is also considered. The direction corresponding to the residual vector ($\boldsymbol{v} = (\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top)\boldsymbol{x}$) is easy to recognize since it causes an increase in the SPE without affecting the T^2 statistic. In the distance plot (Figure 5.5a), it is represented as a vertical arrow, whereas it does not appear in the score plot (Figure 5.5b), given that the projection of $\boldsymbol{x} + a(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x})$ in the model space is the same as that of $\boldsymbol{x}$, for all a values.

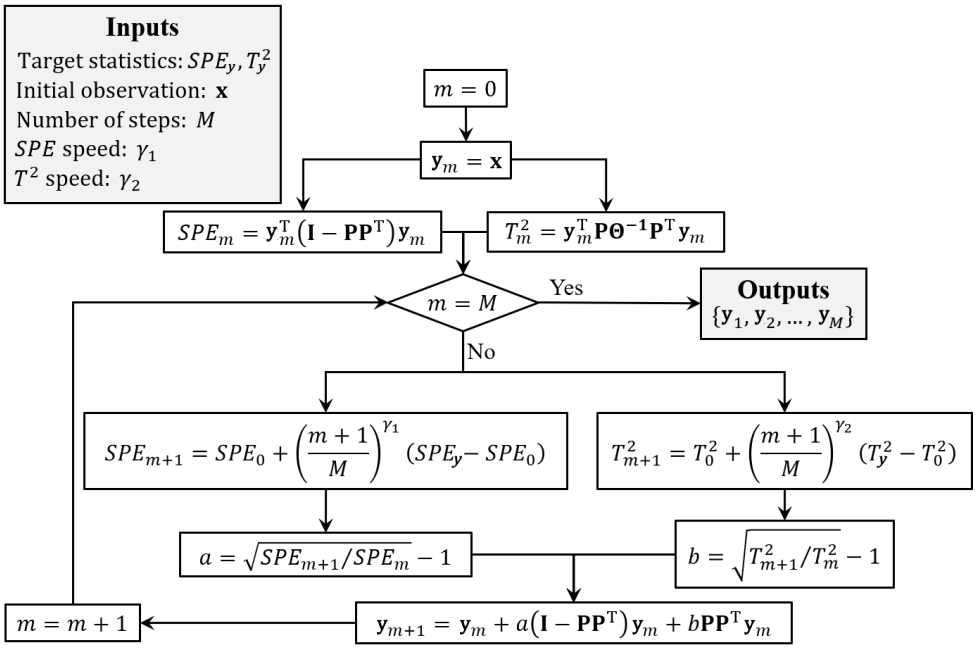
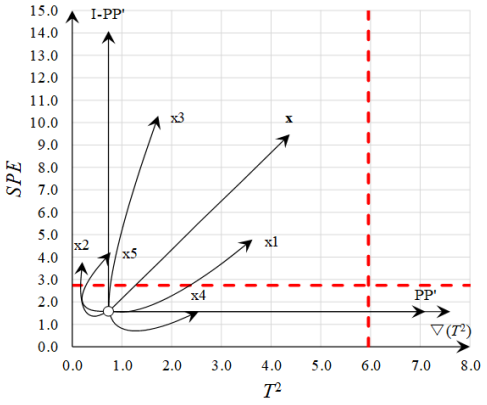
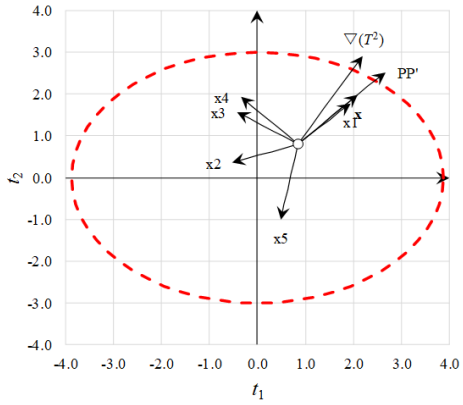


Figure 5.4: Flux diagram of simulation algorithm including all the parameters.



(a) Distance plot with five different shift directions for an observation.



(b) Score plot with five different shift directions for an observation.

Figure 5.5: Five different shift directions for an observation.

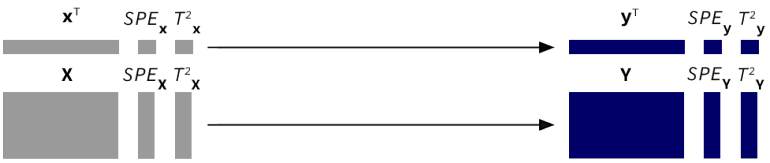
The last two directions are $\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x}$ and $\boldsymbol{P}\boldsymbol{\Theta}^{-1}\boldsymbol{P}^\top\boldsymbol{x}$ (labelled as $\nabla(T^2)$ in Figure 5.5). These two directions are in the model plane, meaning that the SPE will not be affected, which can be appreciated by the horizontal arrows in Figure 5.5a. The magnitude of the shift in the T^2 value is bigger for the $\boldsymbol{P}\boldsymbol{\Theta}^{-1}\boldsymbol{P}^\top\boldsymbol{x}$ direction since it corresponds to the gradient of the T^2 statistic. The trajectory described by the scores when the direction $\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{x}$ is chosen is an extension of the segment that joins the origin (0,0) with the scores of $\boldsymbol{x}$ (i.e., the direction of the predicted observation $\hat{\boldsymbol{x}}$). The trajectory followed when the shift is performed in the direction $\boldsymbol{P}\boldsymbol{\Theta}^{-1}\boldsymbol{P}^\top\boldsymbol{x}$ ($\nabla(T^2)$) is perpendicular to the $(1-\alpha)\times 100$ confidence level Hotelling's T^2 ellipse, which is defined as the level curve for the T^2 statistic.

5.3 Comparative study

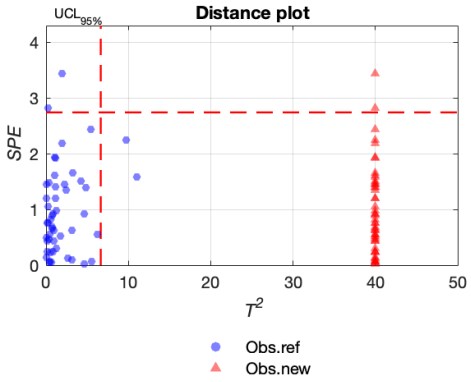
This section shows examples of how to simulate outliers with the desired properties. This section is divided into two main parts. The first part will present results for three different scenarios of outliers simulation. Afterwards, four examples of outliers simulation extracted from literature are emulated using the framework proposed in this work. This exercise aims to show how the technique described in this work can comprise other particular simulation settings. Finally, an assessment of the properties of the simulated outliers in terms of a robust PCA model is also provided.

5.3.1 Cases of use of the proposed method.

These results illustrate three generic simulation scenarios: generating outliers in one step, generating a sequence of outliers, and generating a grid of outliers. For this purpose, a reference matrix $\boldsymbol{X}$ of $n = 50$ observations and $k = 5$ normally distributed variables is simulated. The PCA model based on $\boldsymbol{X}$ is built with two PCs, assuming a type I risk α of 5% and performing a mean centring. All functions, documentation, and scripts to reproduce the following scenarios can be downloaded from the GitHub repository `https://github.com/albagc/SCOUTer.git`. A detailed explanation about the obtention of the following results can be found in the *howto.pdf* file.



(a) Illustration of a one-step simulation of controlled outliers.



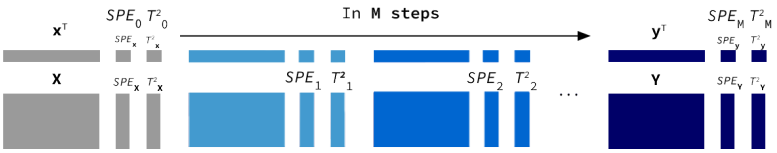
(b) Distance plot with the reference (blue circles) and the shifted (red triangles) data sets, performing a single step keeping the initial $SPE_{\boldsymbol{X}}$ value, but setting a target value $T^2_{\boldsymbol{Y}} = 40$ for all the observations.

Figure 5.6: Concept and result of a one-step simulation with a group of observations.

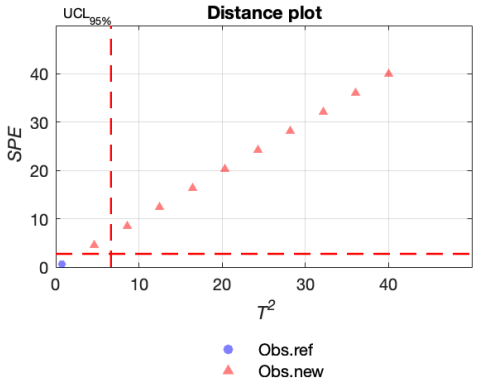
Case I: One-step simulation of outliers.

This is the simplest case, in which from an initial observation $\boldsymbol{x}$ with reference values $SPE_{\boldsymbol{x}}$ and $T^2_{\boldsymbol{x}}$, a new observation $\boldsymbol{y}$ is obtained, with the desired $SPE_{\boldsymbol{y}}$ and $T^2_{\boldsymbol{y}}$ values (Figure 5.6a). The scheme mentioned above can be easily generalized for a set of observations. The original $\boldsymbol{X}$ matrix will be drifted from its initial coordinates in the following example. In this scenario, a set of one-step outliers is generated by increasing only the T^2 value (i.e., extreme outliers). The SPE remains at its reference value.

As it can be seen in Figure 5.6b, all observations have been shifted in their distance to the centre on the model plane, drawing a contour on the score plot for the value $T^2_A = 40$, whereas they have kept their values on the SPE statistic. In other words, this is an example of how to simulate extreme observations.



(a) Illustration of a M-step simulation of controlled outliers.



(b) Distance plot after performing a 10-step shift both in the SPE_x and the T^2 values from one initial observation $\boldsymbol{x}$ (blue circle).

Figure 5.7: Concept and result of a step-wise simulation with an observation.

Case II: Step-wise simulation of outliers

In this scenario, the transition between the reference and the target values for the statistics is performed with a spacing of n steps between them. From a reference observation $\boldsymbol{x}$ (or set of observations $\boldsymbol{X}$) with reference values $SPE_{\boldsymbol{x}}$ and $T^2_{\boldsymbol{x}}$ (or $SPE_{\boldsymbol{X}}$ and $T^2_{\boldsymbol{X}}$), a series of $M-1$ new sets of observations up to $\boldsymbol{y}$ (or $\boldsymbol{Y}$) with the desired $SPE_{\boldsymbol{y}}$ and $T^2_{\boldsymbol{y}}$ (or $SPE_{\boldsymbol{Y}}$ and $T^2_{\boldsymbol{Y}}$) values is generated (5.7a).

The above example (Figure 5.7b) shows a linear spacing between steps for the SPE and the T^2. However, the spacing between steps can be tuned, as seen in Figure 5.2 and Figure 5.3 from Section 5.2.

Case III: Grid-wise simulation of outliers

The step-wise approach performs the same number of steps for both statistics. Finally, the grid-wise case enables different steps for each statistic. Starting from an initial data set $\boldsymbol{x}$ (or $\boldsymbol{X}$) with reference values $SPE_{\boldsymbol{x}}$ and $T^2_{\boldsymbol{x}}$ (or $SPE_{\boldsymbol{X}}$ and $T^2_{\boldsymbol{X}}$), a grid of new observations combining each step of the statistics is obtained (Figure 5.8a). As a result, many data sets are simulated as combinations between the steps of the statistics.

In this last case, a grid with two steps for the SPE and three steps for the T^2 has been produced, setting different spacing parameters for each parameter as well (Figure 5.8b).

Special case: Limitations

This section addresses in further detail the results obtained with the method to simulate outliers with desired properties when used on a matrix with non-linearities or binary data.

The reference matrix $\boldsymbol{X}_0$ is simulated using the functions from Arteaga and Ferrer [98]. The following code lines are the ones used to generate the reference matrix:

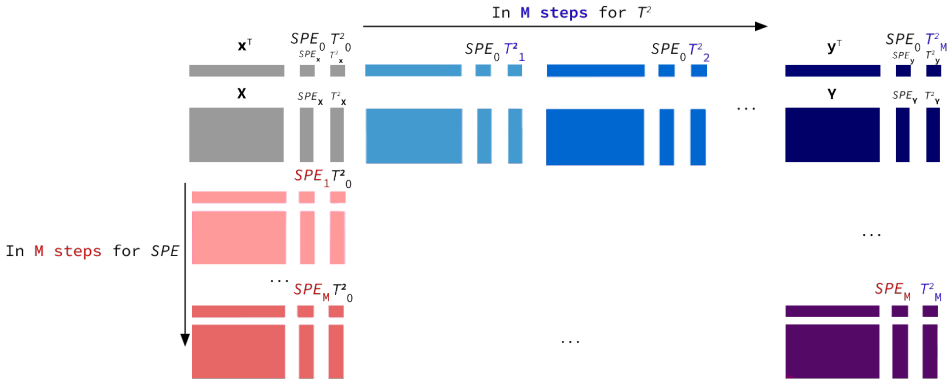
```
[X,S,srnd] = simdataset(100,10,[6,3],ones(1,10));
[X_0,srndn]=randnm(S,100,srnd);
```

The resulting matrix has 100 observations, 10 normally distributed variables, and two principal components, which explain more than 80%

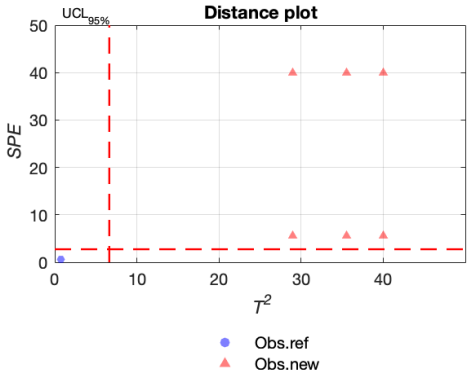
Non linearities. In this case, the matrix will present relations between variables that the classical PCA model cannot capture. To study to what extent this limitation of the PCA model would affect the simulations, we generated outliers with a reference matrix that contained non-linearities and increased only the T^2 of the outliers. This means that the generated observations should not break the correlation pattern described by variables.

The new matrix $\boldsymbol{Y}$ is the result of concatenating the original matrix $\boldsymbol{X}_0$, and a set of non-linear variables generated from the original ones in $\boldsymbol{X}_0$. The non-linear relations included in each variable are:

```
rng(1101)
varind = randperm(10,8);
```



(a) Illustration of a grid-case simulation with M-step shifts for the SPE and the T^2.



(b) Distance plot after performing two steps for the SPE with $\gamma_{SPE} = 3$ and three steps for the T^2 with $\gamma_{T^2} = 0.3$ from one reference observation $\boldsymbol{x}$ (blue circle).

Figure 5.8: Concept and result of a grid-wise simulation with an observation.

```
Y_11 = X_0(:,varind(1)).^2;
Y_12 = X_0(:,varind(2)).^3;
Y_13= exp(X_0(:,varind(3)));
Y_15 = rand(1,1) + X_0(:,varind(5)) + X_0(:,varind(5)).^2;
Y_16 = X_0(:,varind(2)).*X_0(:,varind(4));
Y_17 = X_0(:,varind(6)).*X_0(:,varind(7)).^2;
Y_18 =  exp(X_0(:,varind(3))).^(X_0(:,varind(7)) + X_0(:,varind(8)));
Y_19 = X_0(:,varind(3))*2;

Y = [X_0,Y_11,Y_12,Y_13,Y_14,Y_15,Y_16,Y_17,Y_18,Y_19];
```

As one can notice, the selection of the variables that were non-linearly combined was performed randomly. Also, a linearly generated variable ($\boldsymbol{y}_{19}$) was included in the set to compare if the outliers on this variable still followed their analytic relation with the column used to generate them.

As mentioned, some outliers on the T^2 were generated to keep the original correlation structure between variables. The PCA reference model based on $\boldsymbol{Y}$ had to be calculated to do so. By setting "0" as the second input argument in the PCA-MB function, it returns a suggestion about the number of PCs to consider:

```
pcamodel_ref = pcamb_classic(Y, 0, 0.05, 'cent');

Suggested number of PCs:
- Singular values of covariance matrix > 1 = 6
- Minimum PCs to reach cumulative variance > 80 \% = 3
Select the number of PCs: 3
```

A number of three PCs were selected. Then, outliers on the T^2 were generated, setting the same target value for all of them in the *scout.m* function:

```
T2target = 60*ones(size(Y, 1), 1);
Yextreme = scout(Y, pcamodel_ref, 'simple', 't2y', T2target);
Yall = [Y; Yextreme.X];
```

The resulting outliers are represented in Figure, 5.9 where it can be seen that the new observations accomplish the specified target values for the T^2.

However, the relations between the non-linear variables and the original columns generated have been distorted. Figure 5.10 shows a clear difference between blue and red observations.

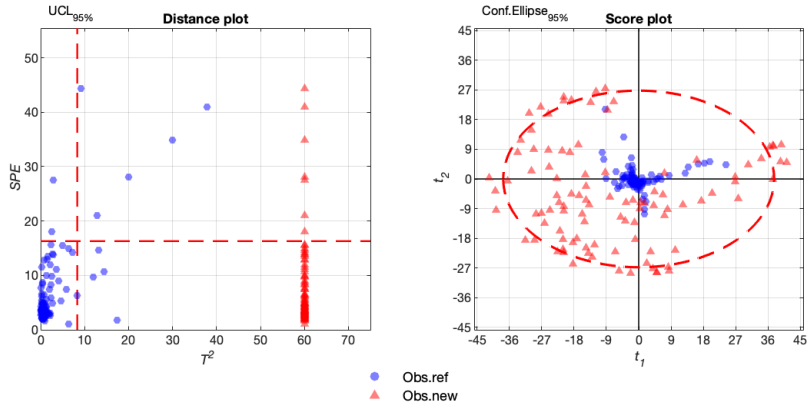


Figure 5.9: Distance (left) and score (right) plot for the reference (blue circles) and the outliers (red triangles) generated.

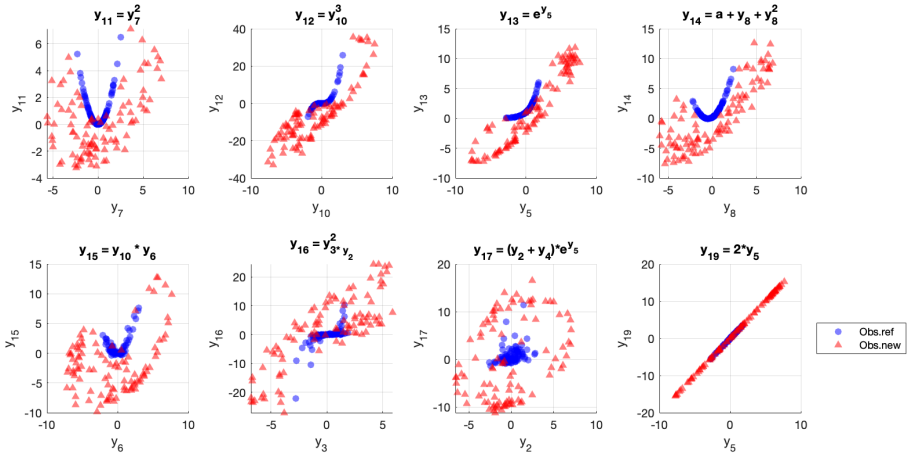


Figure 5.10: Scatter plots with the reference (blue circles) and new (red triangles) observations for all the new variables in $\boldsymbol{Y}$ generated as combinations of the variables in $\boldsymbol{X}_0$.

Whereas the blue circles perfectly describe the analytical relation used to generate them, that is not the case for red triangles since they break the relative pattern between variables. This is not the case for the last variable ($\boldsymbol{x}_{19}$), which was generated as a linear combination. This result reinforces the limitation produced when the method has to consider non-linear relations between the variables.

Binary variables. This second example shows the changes produced on categorical variables when the algorithm is used on a mixed matrix with continuous and categorical data.

In this case, four binary variables with different percentages of 0s and 1s are simulated. The resulting matrix $\boldsymbol{Y}$ has the original variables from $\boldsymbol{X}_0$ and the four additional binary columns.

```
rng(1101)
Y = [X_0,zeros(size(X_0,1),4)];
Y(randperm(size(X_0,1),round(0.2*size(X_0,1))),11) = 1;
Y(randperm(size(X_0,1),round(0.4*size(X_0,1))),12) = 1;
Y(randperm(size(X_0,1),round(0.6*size(X_0,1))),13) = 1;
Y(randperm(size(X_0,1),round(0.8*size(X_0,1))),14) = 1;
```

Similarly, as in 5.3.1, a PCA model is fitted with $\boldsymbol{Y}$, but two PCs were selected in this case.

```
pcamodel_ref = pcamb_classic(Y, 0, 0.05, 'cent');
Suggested number of PCs:
— Singular values of covariance matrix > 1 = 2
— Minimum PCs to reach cumulative variance > 80 \% = 2
```

In this case, we generated outliers increasing the SPE and the T^2, imposing a target value 50 for both of them and all the data points. As shown in Figure 5.11, the set of new observations has the specified values for both statistics.

```
T2target = 50*ones(size(Ybin, 1), 1);
SPEtarget = 50*ones(size(Ybin, 1), 1);
Yout = scout(Ybin, pcamodel_ref, 'simple', 't2y', T2target,'spey',SPEtarget);
Yall = [Ybin; Yout.X];
```

Nonetheless, it is easy to see in Figure 5.12 that new observations are outside the range of accepted values for binary variables. This artefact is produced

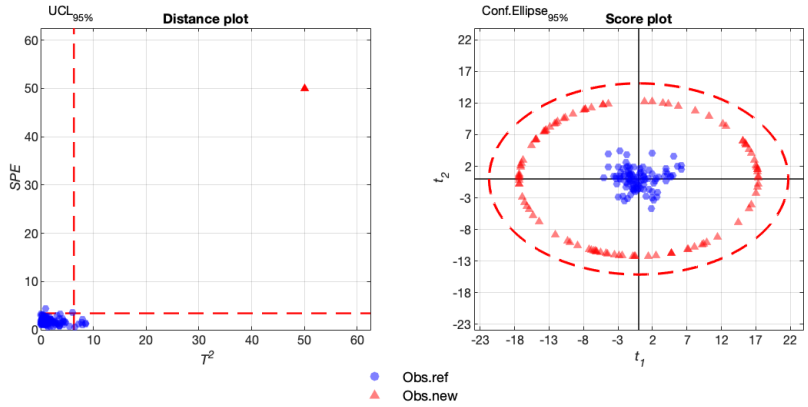


Figure 5.11: Distance (left) and score (right) plot for the reference (blue circles) and the outliers (red triangles) generated.

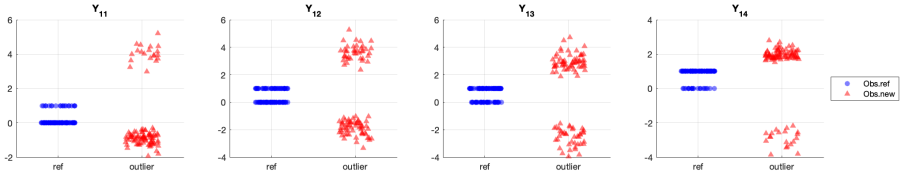


Figure 5.12: Distance (left) and score (right) plot for the reference (blue circles) and the outliers (red triangles) generated.

because the simulation algorithm assumes to work with continuous variables. Consequently, it does not include any constraint in data generation to respect variables' binary or qualitative nature.

5.3.2 Comparison to other simulation methods and PCA frameworks

This section aims to address two critical questions about the simulation method proposed in this work: i) Does the proposed simulation framework encompass other existing simulation strategies, and ii) Will the properties of the simulated outliers be maintained when they are projected onto PCA models fitted with other algorithms rather than the classical least squares version.

Table 5.1: Strategies followed by different authors to simulate the reference data sets and the outlying observations.

Reference.	Simulation of clean data set	Simulation of outliers
[96]	$\boldsymbol{X}_0 \sim N_n(\boldsymbol{0}_n, \boldsymbol{I}_n) \rightarrow \boldsymbol{X}_0 = \boldsymbol{T}_A \boldsymbol{P}_A^\top + \boldsymbol{E}_0$ $\boldsymbol{E}_1 \sim N(\boldsymbol{0}, \boldsymbol{1}) \cdot 0.1$ $\boldsymbol{X}_1 = \boldsymbol{T}_{1,A} \boldsymbol{P}_A^\top + \boldsymbol{E}_1$ $n = 98; k = 20; A = 4$	$\boldsymbol{X}_2 = \boldsymbol{T}_{2,A} \boldsymbol{P}_A^\top + \boldsymbol{E}_2$ $\boldsymbol{E}_2 \sim N(\boldsymbol{10}, \boldsymbol{1})$
[97]	$\boldsymbol{T}_A \sim N_A\ (\boldsymbol{0}_A, \boldsymbol{I}_A)$ $\boldsymbol{P}_A :\perp k \times A$ uniformly distributed pseudorandom numbers $\boldsymbol{E}_k \sim N_k\ (\boldsymbol{0}_k, \boldsymbol{1}_k)/100$ $\boldsymbol{X}_1 = \boldsymbol{T}_A\ \boldsymbol{P}_A^\top + \boldsymbol{E}$	$\boldsymbol{X}_2 \sim N_A\ (\boldsymbol{15}_A, 8 * \boldsymbol{I}_A)$
[85]	$\boldsymbol{X}_1$: Reconstruction of fourth benchmark problem's metabolic network	$\boldsymbol{X}_2$: outliers $x_{ij,2} = -x_{ij,1}$ if $\lvert x_{ij,1} \rvert \leq m_j + 1.5 s_j$
[48]	$\boldsymbol{X}_1 \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_{A09})$ $A = 6PCs, N = 100, K = 200$ $\boldsymbol{\Sigma}_{A09} = \boldsymbol{V}_{A09} \boldsymbol{D}_{A09} \boldsymbol{V}_{A09}^\top$ $\boldsymbol{D}_{A09} = diag(30, 25, \ldots, 5, 0.098, 0.0975, \ldots, 0.005)$	$\boldsymbol{X}_2 \sim N(m\boldsymbol{\nu}_{A+1}, \boldsymbol{\Sigma}_{A09})$ $m \in 1, \ldots, 50$ $\boldsymbol{\nu}_{A+1} = \boldsymbol{V}_{A09}\ [:, A+1]$

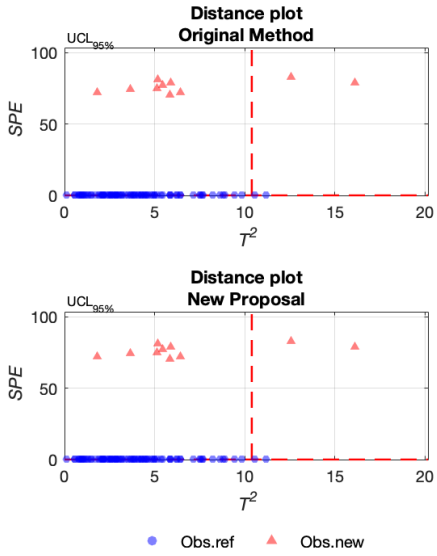
Simulation of other outlier generation strategies

To assess if the proposed method can be seen as a general simulation framework, four strategies to simulate outliers extracted from literature [48], [85], [96], [97] will be redefined in terms of the proposed simulation framework.

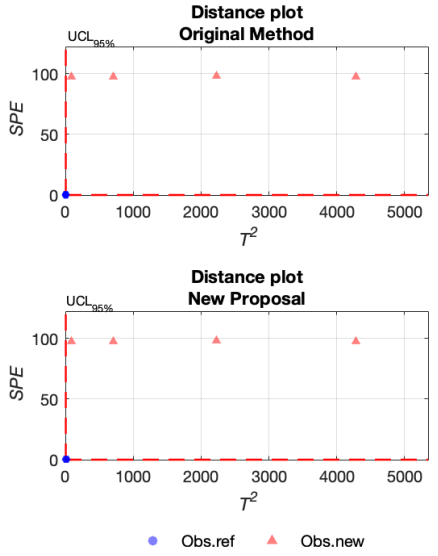
Table 5.1 provides information about the method used in each referred work to simulate the reference data set and the outlying observations. Some notation was adapted from the original works to avoid potential confusion with other terms used in this chapter. The result of executing each simulation procedure is illustrated in Figure 5.13, which provides a graphical comparison between the simulated outliers following the original strategy from the previously mentioned works and using the algorithm proposed in this chapter.

At first glance, one can notice in Figure 5.13 that despite sharing the purpose of simulating outliers, each strategy leads to very different outliers in qualitative and quantitative terms. In Figures 5.13a and 5.13d, outliers are far regarding their orthogonal distance, but their projection onto the model plane seems still under control limits. These plots differ from the ones reported in Figures 5.13b and 5.13c, where outliers are distant regarding the T^2 and the SPE.

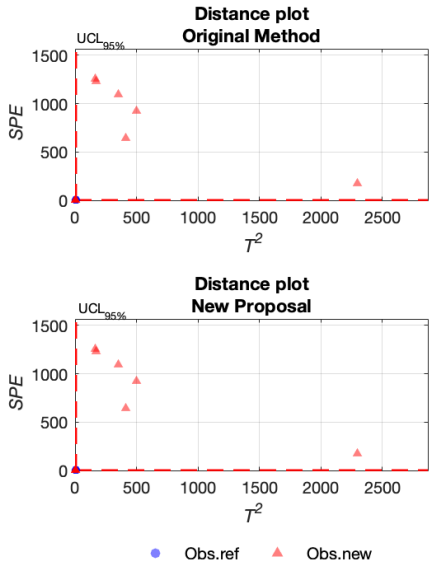
Furthermore, the simulation procedure from Figure 5.13c differs strategically from the others since the same set of observations is shifted 50 steps from their



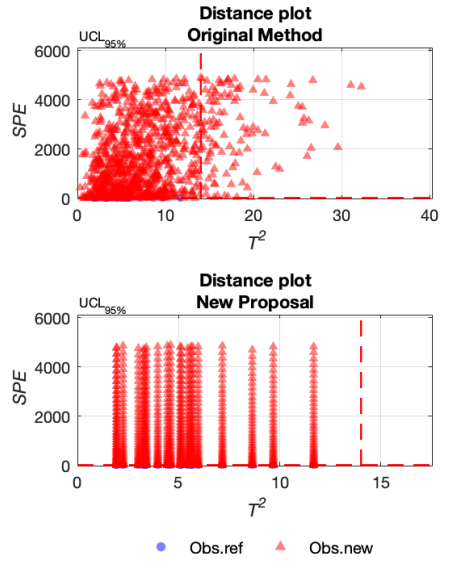
(a) Simulations from [96]



(b) Simulations from [97]



(c) Simulations from [85]



(d) Simulations from [48]

Figure 5.13: Distance plots of the observations simulated using the approach from the original work and the proposed algorithm controlling the outlier properties.

reference values. In Figure 5.13d, the gradual shift of the same set of observations increasing their SPE and randomly shifting the T^2 can be appreciated. This can be seen as well in Figure 5.13c. It also stands out the difference between the upper and lower distance plots in Figure 5.13d. This is because we considered that variations of the T^2 in their simulated outliers were not a strategic feature of the simulation.

Comparing the original methods to simulate outliers (upper row of plots in Figure 5.13), it can be seen that all of them increase the SPE of the outliers, since in the end, despite following different strategies, all procedures to simulate outliers rely on breaking the correlation structure described by the reference data set. This is done differently by each author.

Stanimirova, Daszykowski and Walczak [96] use a simulation strategy relying on adding noise to the outlying observations, whereas Hubert, Rousseeuw, and Vaden Branden [48], the noise is introduced as the new mean vector of the outlying distribution. This results in outliers with an increased SPE but a moderate T^2, as seen in Figures 5.13a and 5.13d. In Serneels and Verdonck [97], outliers are generated by altering the variance of variables, which leads to an increase in the T^2 (Figure 5.13b). The mean vector of the outliers distribution is also changed so that the correlation pattern is not respected anymore, which leads to the increase of the SPE. Finally, Folch-Fortuny at al. [85] shift the sign of randomly selected cells. Consequently, they are breaking the correlation structure, which can also increase the T^2 of the outlying observations (Figure 5.13c).

The comparison between plots from the upper and lower row in Figure 5.13 shows that results obtained by the proposed algorithm to simulate outliers with the desired properties are fairly similar to the ones obtained by other simulation settings. Furthermore, some limitations of the traditional paradigm to simulate outliers can also be seen. This traditional framework relies on changing the distribution parameters that describe the outlying population. Still, there is no direct and transparent relationship between the new parameters of the outlying distribution and their effect on the SPE or the T^2.

Consequently, controlling how this new distribution will affect the outliers' outlying properties is difficult when projected onto the reference PCA model. This can be appreciated because most simulation strategies easily increase the SPE of their observations without controlling its value and having the same control over the T^2 of the outliers. The T^2 seems to be a more uncontrolled parameter, and none of the proposals includes specific outliers for the T^2. This is probably because it is not trivial to find a new mean vector for the

distribution of the outliers that still respects the correlation structure of the reference data set.

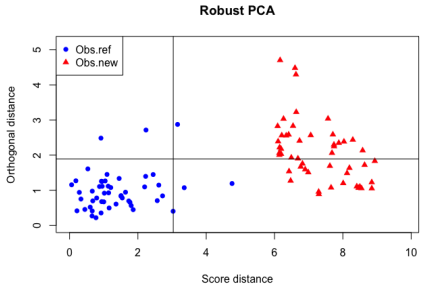
The change from the traditional simulation paradigm to the new one proposed in this work simplifies the relationship between the simulation setup and the properties of the resulting outliers. The algorithm proposed in this work does not rely on the distribution of the reference and the outlying observations, and it has independent control over the SPE and the T^2. This results in a new simulation approach that is versatile enough to encompass other particular simulation strategies (Figure 5.13). Besides, differences between simulation settings can be directly measured regarding the outliers' target SPE and T^2.

Properties of the simulated outliers in a robust PCA model

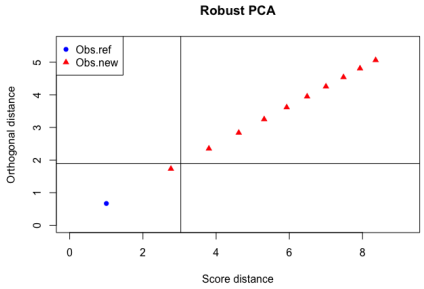
The second aspect of assessing this comparison is to what extent (just quantitative or also qualitative) outliers simulated by the proposed algorithm behave as outliers in terms of other detection techniques. In this sense, it is also interesting to assess if the properties of simulated outliers change when they are expressed in terms of different distance metrics. For instance, some robust PCA techniques differ in the core algorithm to calculate the principal components and the statistics that measure the distance of observation to the model. Hence, the fundamental basis used by our proposed framework to define the outliers differs in these cases. This may affect the properties of simulated observations when they are defined in these new terms.

For this purpose, simulation scenarios from Section 5.3.1 were projected onto a robust PCA model calculated with MacroPCA [48]. This technique can be considered as an ensemble of several outlier detection methods. It includes the *Detect Deviating Cells* (DDC) [59] algorithm as the first step to detect outlying cells, which itself can be regarded as an outlier detection technique. Later on, the MacroPCA algorithm fits a robust PCA model using a version of the ROBPCA algorithm [58], explained in Section 3.3.2 from Chapter 3.

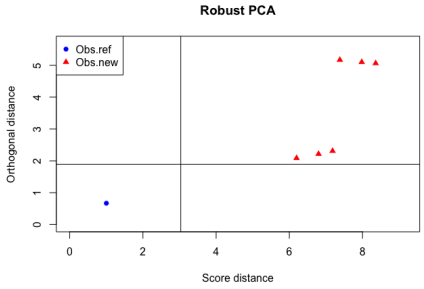
It is worth highlighting that although the distance metrics used in MacroPCA [48] do not coincide with the SPE and T^2, their conceptual meaning is equivalent since they represent the orthogonal distance and the Mahalanobis distance on the model plane, respectively. Thus, we considered MacroPCA clearly representative as a state-of-the-art outlier detection method and a robust PCA model-building algorithm. Moreover, its good performance in outliers detection and the similarity of its distance metrics (orthogonal and score distances)



(a) Data from Figure 5.6



(b) Data from Figure 5.7.



(c) Data from Figure 5.8.

Figure 5.14: Distance plots of the observations simulated in Figure 5.6b when projected onto the PCA model fitted using MacroPCA with the reference data set. Blue circles represent reference observations, whereas red triangles represent the simulated outliers. Black lines represent the Upper Control Limits for the Orthogonal Distance (ordinate) and the Score Distance (abscissa).

to ours (the SPE and the T^2) were considered interesting factors for the comparison.

Results shown in Figure 5.14 were obtained using the *cellWise* package in R (available in `https://CRAN.R-project.org/package=cellWise`).

As shown in Figure 5.14, qualitative properties of the simulated outliers are still met in terms of alternative PCA models and distance metrics. However, there are some differences in the distance values and their Upper Control Limits, which is reasonable given that the Orthogonal Distance and Score Distance are not precisely the SPE nor the T^2. Results in Figure 5.14a also show an increase of the simulated outliers in terms of the orthogonal distance. Given the robust estimation of the covariance determinant in the detMCD step of

MacroPCA, extreme observations in the T^2 were detected as outliers and excluded for the computation of the final PCA model parameters. As a result, since these observations were excluded from the PCA model building at some point, we find it reasonable that they also increased their distance to the model. Nonetheless, in all cases, simulated outliers keep the outlying character that they were asked to represent in the first instance. This can be appreciated by their position above the cut-off values for the distances in all distance plots, indicating the persistence of their outlying properties.

5.4 Conclusions

In this work, a new framework to simulate outliers directly controlling their outlying properties has been proposed. This approach is based on the use of a well-known pair of statistics, the SPE and the Hotelling-T^2 from a PCA model, which evaluates in a complementary way how far an observation is from the majority of the data set (i.e. the outlyingness degree).

Given an observation with initial values for the statistics, a PCA model and target values for the pair of statistics, our simulation method drifts the previous observation in a direction that shifts the initial SPE and Hotelling-T^2 until reaching their target values. This shift direction combines two orthogonal directions, independently controlling the shift on the SPE and the Hotelling-T^2.

This feature is a key factor since it enables specific control over the two properties that define multivariate outliers in a PCA model. This becomes critical, especially when simulating anomalous data, a general procedure when testing the performance of different statistical methods handling datasets with outlying observations.

However, the outliers generation is usually an *ad hoc* procedure lacking standard protocols. It is based most of the time, even when working with PCA models, on distributions and parameters that do not tune either how or how much observation is outlying. This makes the supposed benefits of the different statistical methods depend on the nature of the simulated outliers. Consequently, comparing the different methods reported in the literature becomes difficult or impossible.

Moreover, most simulation methods require an assumption about the distribution of the reference data set and simulate outliers by changing one of its parameters, such as the mean or the covariance matrix. This simulation paradigm

might not be feasible to implement with real data sets when the distribution is unknown. Furthermore, the relationship between the new distribution parameters and the simulated observations' outlying properties is not simple and direct.

In Section 5.3.2, we showed how the methodology proposed in this article successfully encompasses particular simulation strategies presented in the literature in a common framework. Consequently, comparing approaches can be easily measured regarding target specifications or procedures followed to shift the outliers, i.e., one-step, step-wise or grid-wise.

Besides, we also illustrated the shortage of extreme (good leverage) outliers simulated in the literature given the difficulty of modifying the reference distribution while respecting its covariance structure, which is easily achieved by the simulation framework proposed in this chapter (Figure 5.6b). Moreover, in Section 5.3.2, we also showed how the outlying properties are, at least, qualitatively consistent when the simulated outliers are projected on a robust PCA model.

However, the proposed method has some limitations, further addressed in Section 5.3.1. The simulation procedure does not set any restriction in case that binary or categorical variables are present in the matrix. Naturally, this framework is also restricted by the same limitations as the PCA model, such as the inability to model non-linear relations between variables (see Section 5.3.1).

In summary, the framework proposed in this chapter offers the possibility of generating outlying observations with a wide range of desired properties, given that the user can control the pair of statistics that essentially define the outlyingness degree: the SPE and the Hotelling-T^2. This procedure has been implemented in Matlab, providing a set of functions to perform the PCA Model Building and the simulation of controlled outliers. Further details about the Matlab code can be found in the documentation file available in the GitHub repository.

Appendix: Software implementation

Implementation in Matlab

The results shown in Chapter 5 were obtained by executing the functions from the repository available in `https://github.com/albagc/SCOUTer.git`. This GitHub repository contains all the functions and scripts to run the simulations.

They were programmed in Matlab version R2020a 9.8.0.1323502. Moreover, further information about the functions can be found in the *documentation.pdf* and *howto.pdf* documents on the repository.

Implementation in R package

We also implemented SCOUTer in an open-code and license-free environment to make the code available for non-Matlab users. A set of functions and scripts analogous to the one implemented in Matlab can be found as part of the SCOUTer R package, accepted in the CRAN repository `https://github.com/albagc/SCOUTerRpack.git`. [99]. Moreover, an unstable version is also available in the GitHub repository

Implementation in Shiny App

Following the idea of making SCOUTer as accessible as possible, a GUI implementing all the functions of the SCOUTer package was developed using R Shiny. This environment enables the operation of the set of functions of the R package SCOUTer in a user-friendly way, using an interface where any programming is necessary to build a PCA model and simulate the desired outliers. The app is hosted in https://sdralgonceb.shinyapps.io/SCOUTerShinyApp.

The GUI is divided into three main panels. The first one, displayed in Figure 5.15, controls the data loading and offers a quick view of the number of observations and variables in the selected dataset. A demo option also includes the data used in this chapter and in [94].

The second panel (Figure 5.16) controls the PCA-MB step using the reference data. It has several options to select the range of observations to fit the model, the number of PCs, the α parameter to calculate the UCLs and the type of data preprocessing.

Finally, the third panel (Figure 5.17) controls the simulation of outliers, including all the options and parameters mentioned in the chapter. The distance and score plots are interactive: when a datapoint is clicked, its contributions for the SPE and T^2 are displayed below, along with the information about its simulation. At the bottom of this panel is a "Download" section, where the user can select the downloaded file format containing the reference data, the simulated outliers and the reference PCA model.

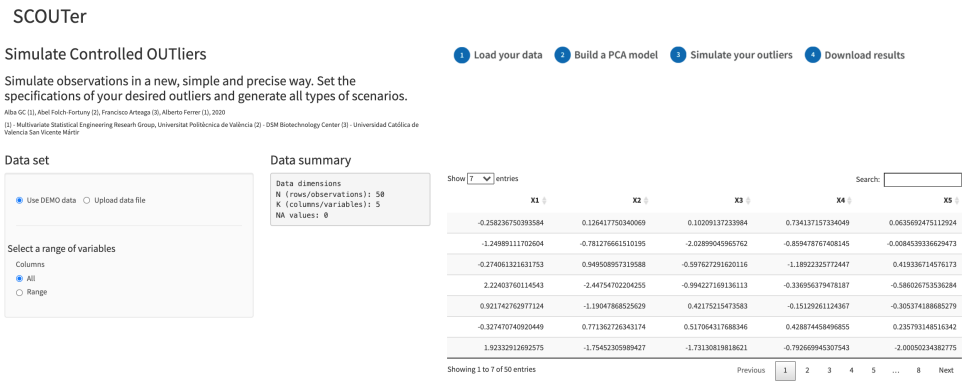


Figure 5.15: Data loading and descriptive summary panel.

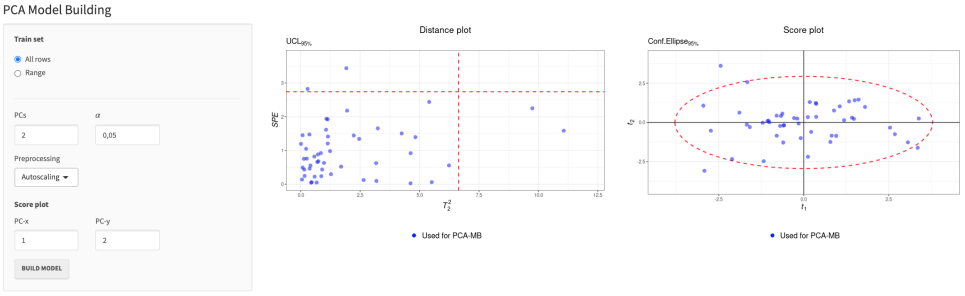


Figure 5.16: PCA Model Building panel.

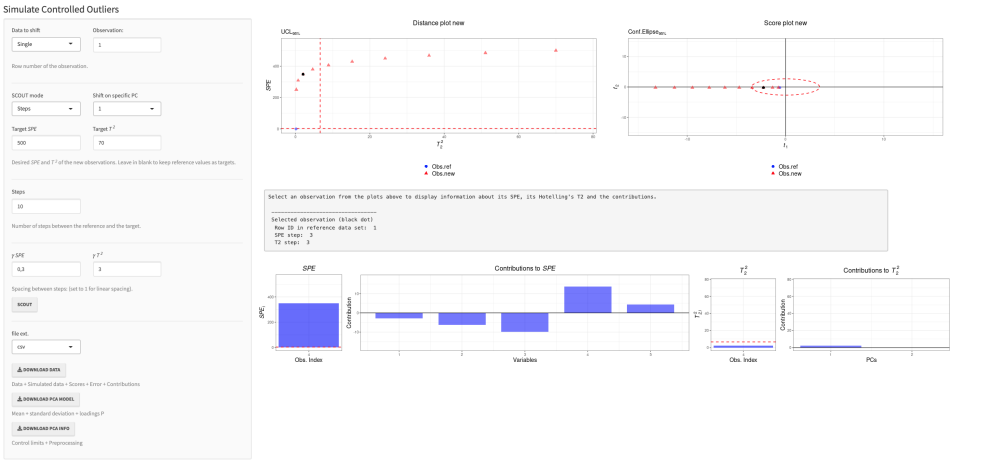


Figure 5.17: SCOUTer and download panel with interactive options to explore the contributions of observations from the distance and score plots.

Chapter 6

RadarTSR: PCA model building with missing data and outliers

Part of the content of this chapter has been included in:

[100]González-Cebrián, Folch-Fortuny, A., Arteaga F. & Ferrer, A. RadarTSR: A New Algorithm for Cellwise and Rowwise Outlier Detection and Missing Data Imputation. *Chemometrics And Intelligent Laboratory Systems.* **247** (2023), https://doi.org/10.1016/j.chemolab.2023.105047

6.1 Introduction

Real datasets often present missing data and/or outliers. Missing data (MD) appears in various scenarios (unanswered survey questions, data acquisition failures, etc.). In massive data collection scenarios, more than 70% entries in a dataset might be missing [101], [102], resulting in many proposals of imputation methods that can be used when the missingness mechanism is ignorable (see [102]). Analogously, there is a whole branch of robust statistics and algorithms to deal with the existence of outliers [103], but most of them assume to work with a full matrix. As a result, the number of approaches simultaneously dealing with missing data and outliers is drastically reduced.

Moreover, most robust approaches assume the existence of rowwise outliers, i.e., observations (rows) whose variables (columns) do not follow the model described by most observations. However, some authors have highlighted the relevance of cellwise outliers. These are entries with suspicious values caused by random events such as measurement errors [59], [92], [93]. Even a low proportion of cellwise outliers can affect more than half the observations, which is the maximum contamination fraction that rowwise robust methods based on affine equivariant estimators can deal with, and this is critical considering that the cellwise contamination effect is even more pronounced in high-dimensional situations, where they are frequently found [91], [92]. Therefore, proposing techniques that not only deal with missing data and/or outliers but also account for the high dimensionality of datasets is a task of great interest.

For this reason, techniques tailored for imputation and outlier correction within Principal Component Analysis (PCA) operate efficiently in scenarios where the missing data generation mechanism is deemed ignorable, typically adhering to Missing Completely At Random (MCAR) or Missing At Random (MAR) assumptions. As far as we are concerned, the MacroPCA algorithm [48] is the only available technique based on a PCA model, which deals with missing values and rowwise and cellwise outliers within a dataset. However, it presents some aspects that are worth to be discussed.

First, the imputation in MacroPCA is done by an iterative estimation yielded by a PCA model [46], [84]. However, as it is proved in [85], when outliers are not present in the dataset, the Trimmed Scores Regression (TSR) algorithm is statistically superior in terms of the Mean Squared Predictive Error (MSPE) in comparison with the Iterative Classic PCA (ICPCA) model. Moreover, it has proved to work well in the context of Model Building (MB, when fitting a model) [85], Model Exploitation (ME, when using a model with new data) [77] and in prediction contexts (when having a matrix of predictors and responses)

[104]. Hence, the choice of ICPCA might be sub-optimal if compared with TSR.

Furthermore, once cellwise contamination is detected, it can be fixed by imputation correcting cellwise outliers and keeping the rows that may contain them. This, added to the (possibly) already present missing data in the same matrix, highlights the importance of using the appropriate missing data imputation technique. Thus, the interest in proposing novel approaches with TSR for missing data imputation and cellwise outliers correction seems clear. However, TSR is a technique based on classical PCA, inheriting its least-squares nature, which is not robust to the effect of outliers. Even a single outlier can have enough leverage to distort the extracted principal components (PCs). This can mislead the interpretations of the PCA model and can also mask outliers, disguising them as non-outlying observations according to the fitted PCs [97].

This poses a direct question: could a robust algorithm using TSR for the cellwise and missing data imputation improve the results yielded by MacroPCA? A straightforward answer to this question could have been to include TSR as the imputation step of the MacroPCA algorithm, but this would inform merely about the optimal imputation strategy in the context of the MacroPCA algorithm. There are many other options for robust PCA algorithms, generally divided into those using robust estimators of the covariance matrix, projection pursuit approaches, or both strategies combined. A good review can be found in [97]. Nonetheless, only a few robust PCA solutions can deal with missing data, and most assume the existence of rowwise outliers, i.e., rows which do not belong to the population defined by the majority of observations.

Therefore, instead of analyzing the optimal combination between an already existing robust PCA algorithm and TSR, we decided to take a different direction and robustify the TSR algorithm for imputation by keeping a balance between adding a low number of robust steps and reaching the necessary robustness of the algorithm in the presence of outliers. By doing so, we might obtain a trade-off solution holding the superiority of least-squares methods in the absence of outliers but with robustness that successfully prevents the outliers' influence on the model estimates.

Finally, a second aspect that deserves further consideration is that rowwise outliers are usually treated as non-imputable cases, i.e., the PCA model fitted with non-outlying observations cannot be used on them. However, we believe that rowwise outliers should be divided into two categories: single and grouped rowwise outliers. Whereas single rowwise outliers are observations that can't match the same pattern as any other observation in the dataset, grouped row-

wise outliers might not fit the reference model but show a consistent agreement with other rowwise outliers. Thus, in the case of having grouped rowwise outliers constituting a cluster, they can (and should) be imputed using their own model. Nonetheless, the furthest point offered by MacroPCA is the detection of such rows as rowwise outliers by setting thresholds to the orthogonal and scores distances (see [48]) of the observations, but without any further attempt to assess the existence of that potential cluster.

In this chapter, we propose the algorithm RadarTSR (Robust Adaptation for datasets with Anomalous Rows of Trimmed Scores Regression) to impute missing data, detect cellwise and rowwise outliers, and impute minority sub-populations, if detected. It uses the TSR algorithm for missing data imputation and cellwise outliers correction, an already validated solution for dealing with missing data [77], [85]. This work upgrades this problem with the potential presence of all the abovementioned outliers, preventing TSR´s breakdown by adding the minimum number of computationally efficient robust steps. Hence, the RadarTSR algorithm takes a heuristic approach due to the interplay between the goals mentioned above, allowing for striking a balance between achieving robustness in the presence of outliers and leveraging the superiority of least-squares methods in the absence of outliers.

Next, the Methodology section briefly introduces the PCA model framework and explains the RadarTSR algorithm in the MB context. The Results section shows a comparison between RadarTSR and other state-of-the-art techniques, including MacroPCA [48], TSR [85], and the Iterative Classic PCA (ICPCA) [84]. Several real and simulated datasets were considered, generating outliers and missing data when data matrices were complete. Finally, the Conclusions section summarises the main remarks and outcomes of the work.

6.2 Methodology

The notation used in this section will use bold uppercase letters to denote matrices, also indicating their dimensions the first time they are mentioned, e.g., $\boldsymbol{X}$ is a matrix of N rows and K columns. Bold lowercase letters denote column vectors of the matrices expressed in uppercase, e.g., $\boldsymbol{x}$ is a column vector of N rows from matrix $\boldsymbol{X}$. Lowercase subindices i and j accompanying vectors or scalars indicate they are a specific row or column from a matrix, respectively, e.g., $\boldsymbol{x}_i$ and x_{ij} are the i-th row and the element located in the i-th row and j-th column of $\boldsymbol{X}$, respectively. The notation used in the chapter can be found in Appendix 6.A.

A central piece of RadarTSR's algorithm is the PCA model. The PCA model [46] performs a low-rank bi-linear decomposition on a matrix $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^\top + \boldsymbol{E}$, where $\boldsymbol{T}$ is the matrix of scores and $\boldsymbol{P}$ is the matrix of loadings. A detailed explanation of the PCA model can be found in Section 3.3.1. Throughout this chapter, two metrics from the PCA model will be key: the Squared Prediction Error ($SPE_i = \boldsymbol{e}_i^\top \boldsymbol{e}_i$) and the Hotelling's T^2 ($T_i^2 = \sum_{j=1}^{a} t_{ij}^2/\lambda_a^2$) of an observation, which measure the orthogonal distance of observation to the model and the Mahalanobis distance in the latent space, respectively. Such metrics, control limits, and schemes can be implemented to detect outliers [86].

In this chapter, several phenomena threatening the fitting of the classical PCA model when present in the matrix $\boldsymbol{X}$ are considered. First, some entries x_{ij} can be missing. The assumed mechanism generating this missingness will be ignorable, i.e., cells will be Missing Completely At Random (MCAR) or Missing At Random (MAR). This assumption is necessary to apply TSR as an imputation technique. For more information on the Missing Data problem, see Section 3.5.2.

Secondly, the data may contain cellwise outliers generated at random. Such randomness assumption enables treating outlying cells as imputable missing entries using TSR. It is worth mentioning that cellwise outliers can only exist in non-outlying rows since outlying values of rowwise outliers are part of their multivariate outlying pattern. Therefore, rows with cellwise outliers are inherently non-outlying rows.

Finally, outlying rows can also be in $\boldsymbol{X}$. Some rows are *single* rowwise outliers, which do not belong to a particular group within $\boldsymbol{X}$. Hence, missing entries within single rowwise outliers will not be imputable. However, there can be *grouped* outlying rows constituting a minority cluster within $\boldsymbol{X}$. In this case, missing entries within grouped rowwise outliers can be imputed with their imputation model.

The following section will explain the steps of the RadarTSR algorithm to detect and correct all the mentioned phenomena.

6.2.1 RadarTSR algorithm for Model Building

This section explains how the RadarTSR algorithm achieves its goals in the context of MB when it seeks to fit a PCA model given a potentially contaminated matrix. Figure 6.1 shows the sequence of steps the RadarTSR algorithm applies to the initial $\boldsymbol{X}$ matrix in an MB scenario.

Given a matrix $\boldsymbol{X}$, the RadarTSR algorithm aims to impute missing cells, to detect and correct cellwise outliers, to detect rowwise outliers, and to impute them if they happen to be grouped rowwise outliers. This multi-goal aspect of RadarTSR leads to several outcomes along the algorithm. The following notation will be used to refer to each one of them:

- the *NA-imputed* matrix, of dimensions $N \times K$ where outlying rows and cellwise outliers are not imputed, but only missing entries of non-outlying rows;
- the *cell-imputed* matrix, of dimensions $N \times K$, with cellwise outliers, imputed for non-outlying rows and missing entries imputed for all rows, no matter the outlyingness; and
- the *cluster-imputed* matrix, of dimensions $N \times K$, with non-outlying rows being cell-imputed, *single* rowwise outliers NA-imputed (i.e., imputing only missing cells) with the reference PCA model, and *grouped* rowwise outliers NA-imputed with their corresponding PCA model.

The vector $\boldsymbol{y}$ indicates the cluster assigned to each N observation and the corresponding PCA model used for its imputation. In a scenario with a single cluster (i.e., without grouped rowwise outliers), $\boldsymbol{y} = \boldsymbol{0}_N$. As new clusters are detected, it generates new corresponding y values, sorting them in descending order according to the number of observations within each cluster. That way, the "zero" cluster tag value always refers to the reference (most numerous) group.

Finally, the reference cluster also has the thresholds for the detection of cellwise (with subindex cw) and rowwise (with subindex rw) outlying events $\left\{c_{cw}, c_{rw}^{p^{cw}}, c_{rw}^{SPE}, c_{rw}^{T^2}\right\}$, the indicator matrices $\mathring{\boldsymbol{M}}$, $\dot{\boldsymbol{M}}$, of dimensions $N \times K$, and the indicator vector $\boldsymbol{m}$ with ones for missing cells, outlying cells and outlying rows, respectively. Further details about the definition of all the elements mentioned earlier can be found in Section 6.2.2 and Appendix 6.A.

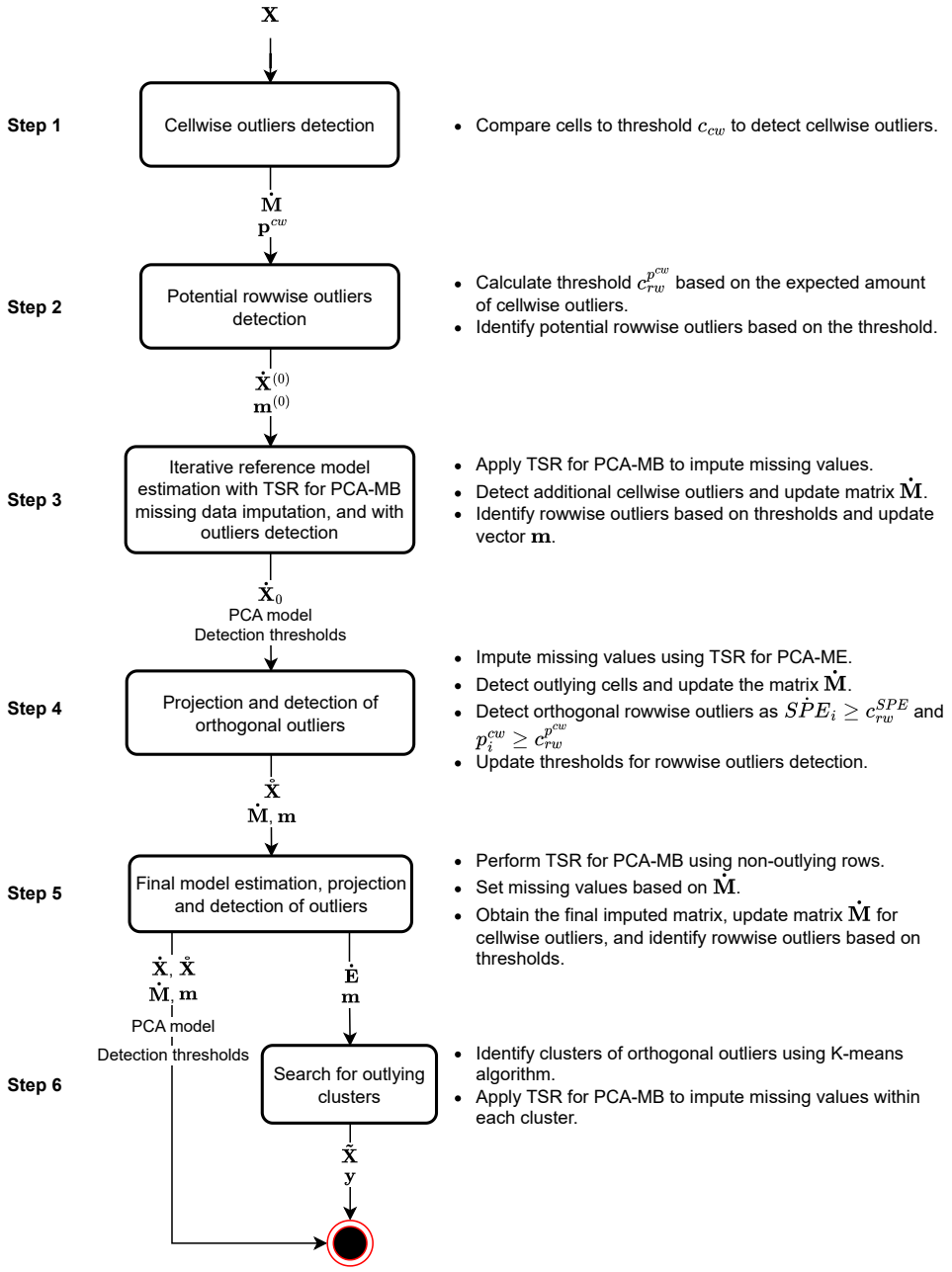


Figure 6.1: Flowchart with the six main steps of the RadarTSR algorithm for PCA-MB

This section explains how the RadarTSR algorithm achieves its goals in the context of MB when it seeks to fit a PCA model given a potentially contaminated matrix. Supplementary Figure 1 shows the sequence of steps the RadarTSR algorithm applies to the initial $\boldsymbol{X}$ matrix in an MB scenario.

Given the input matrix $\boldsymbol{X}$, the RadarTSR algorithm performs six main steps, further explained below. Even after applying the common preprocessing step before fitting a PCA model, the resulting preprocessed matrix will be referred to as $\boldsymbol{X}$ to simplify the notation. To prevent outliers' influence, centring and scaling along the RadarTSR algorithm are performed with 1-step location and scale estimators, respectively. These estimators are the same ones as in [48], referred to as *robloc* and *robscale*, yielding the $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\sigma}}$ parameters, respectively. The autoscaled matrix is the input to the sequence of steps from Supplementary Figure 1, explained in more detail along the following lines:

1. Cellwise outliers detection. The goal is to spot extreme cellwise outliers. This is done by comparing cells from the autoscaled $\boldsymbol{X}$ in absolute value to the threshold $c_{cw} = z_{\alpha/2}$, referring to the $100\,(1-\alpha/2)\%$ percentile of a standardized normal distribution. By default, α is set to 1%. Cells for which $|x_{ij}| > c_{cw}$, are signaled as cellwise outliers in matrix $\dot{\boldsymbol{M}}$, setting the corresponding entries $\dot{m}_{ij} = 1$. This matrix yields the vector of the percentage of outlying cells per row ($\boldsymbol{p}_n^{cw}$) and also indicates which entries x_{ij} must be set as missing so that they can be imputed and corrected in further steps.

2. Potential rowwise outliers detection. This step aims to detect potential rowwise outliers that could be masked after imputing missing cells and cellwise outliers. A threshold on the expected amount of cellwise outliers in a non-outlying row ($c_{rw}^{p^{cw}}$) is calculated as:

$$c_{rw}^{p^{cw}} = robloc\,(\boldsymbol{p}^{cw}) + z_{\alpha}\sqrt{robloc\,(\boldsymbol{p}^{cw})\,(1 - robloc\,(\boldsymbol{p}^{cw}))\,/k} \qquad (6.1)$$

   where $\boldsymbol{p}^{cw}$ is a vector of N elements indicating the proportion of outlying cells detected in each row. The number of outlying cells per row follows a binomial distribution, where the expected rate of cellwise outliers is denoted as p^{cw} and the number of variables is denoted as K. To determine the threshold for identifying rowwise outliers, the normal approximation is used when $Kp^{cw}(1-p^{cw}) \geq 10$; otherwise, the threshold is set based on the binomial distribution. Rows with $p_i^{cw} \geq c_{rw}^{p^{cw}}$ are initially identified as potential rowwise outliers by setting corresponding entries in the vector $\boldsymbol{m}^{(0)}$ to one.

It is worth noting that the proportion-based filter for rowwise outliers is sensitive to the dimensionality of the dataset: low-dimensional rows, even if they exhibit only one outlying cell, could be detected as rowwise outliers. However, rowwise contamination implies that a group of variables deviates from the overall multivariate pattern. Therefore, rows flagged in $\boldsymbol{m}^{(0)}$ are also required to have at least two or more outlying cells ($k_{cw} \geq 2$). On a related note, one might argue that detecting multivariate outlying patterns by searching for univariate outliers, which are assumed to be independent and equally likely, is a naive approach. Nonetheless, the set of flagged rowwise outliers ($\boldsymbol{m}^{(0)}$) is not final and will be updated in subsequent steps ($\boldsymbol{m}^{(s)}$, $s > 0$).

3. Iterative reference model estimation, missing data imputation, and outliers detection. This step incorporates iterative reference model estimation, missing data imputation, and outliers detection using TSR for PCA-MB. The entries of the input matrix $\boldsymbol{X}^{(0)}$ are set as missing according to $\mathring{\boldsymbol{M}}$ and $\dot{\boldsymbol{M}}$ matrices. Outlying observations according to $\boldsymbol{m}^{(0)}$ are also removed from $\boldsymbol{X}^{(0)}$, yielding a matrix of $N' \times K$ dimensions. The term N' will denote the varying number of non-outlying observations along the iterations s. The latent subspace dimension A is determined based on factors such as eigenvalue scree plots or cross-validation methods like the one proposed in [105]. In this case, A is set as the number of PCs reaching an accumulated explained variance above 80%, and a default maximum number of 10 PCs is also set. After the first iteration, half of the observations with an SPE below the upper control limit (i.e., $SPE_i \leq c_{rw,SPE}/2.5$) and with the lowest T^2, are used to fit the PCA model. This is done to avoid the generation of artificial PCs by extreme outliers, which could mask undetected moderate outliers (i.e., SPE outliers). At each iteration s:

   (a) TSR for PCA-MB is applied to non-outlying observations (i.e., for which $m_i^{(s-1)} = 0$) from $\dot{\boldsymbol{X}}^{(s-1)}$, resulting in a PCA model and imputations on $\dot{\boldsymbol{X}}^{(s)}$.

   (b) Projection onto the PCA model yields Squared Prediction Error ($\dot{SPE}_i^{(s)}$) and Hotelling's $\dot{T}^2$ ($\dot{T^2}_i^{(s)}$) for all observations. The autoscaled residual matrix $\dot{\boldsymbol{R}}^{(s)} = \boldsymbol{X}^{(s)} - \hat{\dot{\boldsymbol{X}}}^{(s)}$ is used to flag all cells with $|\dot{r}_{ij}| \geq c_{cw}$ as cellwise outliers, as in Step 1. The *robloc* and *robscale* estimators are applied to autoscale the $\dot{\boldsymbol{R}}$ matrix.

(c) The threshold for the expected percentage of univariate outliers per row, $c_{rw}^{p^{cw}(s)}$, is calculated as in Step 2. Depending on its value, outlying entries in rows with $p^{cw(s)} < c_{rw}^{p^{cw}(s)}$ are set as missing, while rows with $p^{cw(s)} \geq c_{rw}^{p^{cw}(s)}$ retain their outlying entries.

(d) Observations that exceed the thresholds $c_{rw}^{SPE(s)}/2.5$ or $c_{rw}^{T^2(s)}$ are identified as rowwise outliers, marked in the $\boldsymbol{m}^{(s)}$ vector and removed from subsequent iterations. The calculation of the thresholds c_{rw}^{SPE} and $c_{rw}^{T^2}$ is described in Section 6.2.2. The threshold for detecting moderate outliers is directly equal to the Upper Control Limit for the $SPE^{(s)}$ to enhance the sensitivity of RadarTSR in identifying moderate outliers, which is the main goal of this step.

The iterations continue until reaching the maximum number of iterations ($s = 200$ by default) or convergence, defined by a tolerance for the maximal angle between loading vectors from consecutive iterations (by default of 0.005) [48]. The output provided is the cell-imputed original matrix $\dot{\boldsymbol{X}}^{(0)}$, the reference PCA model $\{\hat{\boldsymbol{\mu}}, \boldsymbol{P}, \boldsymbol{\lambda}\}^{(s)}$, and the thresholds $c_{rw}^{p^{cw}(s)}$, $c_{rw}^{SPE(s)}$ and $c_{rw}^{T^2(s)}$ used for rowwise outliers detection. The set of flagged cellwise outliers from Step 1 ($\dot{\boldsymbol{M}}$) is updated with cellwise outliers detected in $\dot{\boldsymbol{R}}^{(s)}$, and all rows with $S\dot{P}E_i > c_{rw}^{SPE(s)}$, are included in $\boldsymbol{m}$.

4. Projection and detection of orthogonal outliers. After the convergence of Step 3, TSR in the Model Exploitation framework is applied to obtain the NA-imputed $\mathring{\boldsymbol{X}}$ matrix. Cellwise outliers are detected from $\mathring{\boldsymbol{R}}$, updating the matrix $\dot{\boldsymbol{M}}$. Orthogonal rowwise outliers are detected as well, updating vector $\boldsymbol{m}$ with ones in rows with an $S\dot{P}E_i \geq c_{rw}^{SPE}$ or a $p_i^{cw} \geq c_{rw}^{p^{cw}}$.

5. Final model estimation and detection of outliers. TSR for PCA-MB is executed without outlying rows according to $\boldsymbol{m}$ and setting all cells flagged in $\dot{\boldsymbol{M}}$ as missing. Then, the resulting model is used to detect cellwise outliers in $\dot{\boldsymbol{R}}$ and update the percentage of cellwise outliers per row. Rows with a $S\dot{P}E_i > c_{rw}^{SPE}$ and with a $\dot{T^2}_i > c_{rw}^{T^2}$, are updated in $\boldsymbol{m}$. The cell-imputed matrix is obtained accordingly, along with the final detection of outliers, yielding $\dot{\boldsymbol{M}}$ and $\boldsymbol{m}$.

6. Search for outlying clusters. This step aims to find groups of moderate outliers that need a different PCA model for a proper missing data imputation. Since outliers were removed in Steps 3 and 4, relevant PCs, presumably due to patterns among previously removed rowwise outliers,

could be extracted from a PCA on the residual matrix $\dot{\boldsymbol{E}} = \dot{\boldsymbol{X}} - \hat{\dot{\boldsymbol{X}}}$. The approach to identify clusters relies on the K-means algorithm further explained in Section 6.2.2. If a cluster is detected, TSR for PCA-MB is applied to its observations. This results in the final cluster-imputed matrix $\tilde{\boldsymbol{X}}$ with missing entries imputed in each row using its cluster's PCA model.

6.2.2 Dealing with rowwise outliers

RadarTSR performs different steps to detect outliers and protect the least-squares core of the TSR algorithm used to build the PCA model and to impute missing data and cellwise outliers. This section gives more technical details first, on rowwise outliers' detection, and second, on the clustering approach.

Rowwise outliers detection

In RadarTSR, rowwise outliers are detected based on the SPE and Hotelling's T^2 statistics. These two statistics define the outlyingness of the observations, and two complementary thresholds, c_{rw}^{SPE} (Equation 6.2) and $c_{rw}^{T^2}$ (Equation 6.3), can be obtained based on percentiles of their assumed distributions [86], [88], [106].

$$c_{rw}^{SPE} = 2.5 \cdot \theta_1 \left[z_\alpha \sqrt{2\theta_2 h_0^2}/\theta_1 + 1 + \theta_2 h_0 \left(h_0 - 1\right)/\theta_1^2 \right] \tag{6.2}$$

$$c_{rw}^{T^2} = 2.5 \cdot A \left(N^2 - 1\right) F_{(A,(N-A)),\alpha} / \left(N \left(N - A\right)\right) \tag{6.3}$$

In Equation 6.2, $\theta_i = \sum_{j=A+1}^{K} (\lambda_j)^i$, and $h_0 = 1 - 2\left(\theta_1 \theta_3\right) / \left(3\theta_2^2\right)$, with λ_j being the eigenvalues of the PCA residual covariance matrix and z_α is the $100\left(1-\alpha\right)\%$ percentile of a standard normal variable. In Equation 6.3, the term $F_{(A,(N-A)),\alpha}$ refers to the $100\left(1-\alpha\right)\%$ percentile of a Fisher distribution with A degrees of freedom in the numerator, $N - A$ in the denominator. The factor of 2.5 is applied to both Equations 6.2 and 6.3 according to a widely used heuristic to determine a threshold value that works reasonably well.

Observations could be moderate outliers if their SPE is above c_{rw}^{SPE}, and/or extreme outliers if their Hotelling's T^2 is above $c_{rw}^{T^2}$. While a high T^2 indicates extreme values that, if the SPE is not high, still respect the general correlation pattern, high SPE values characterize observations far from the latent subspace, i.e., observations not respecting the correlation pattern and not fitting

the PCA model. For this reason, the reference PCA model should not be used to impute missing values in moderate outliers.

Yet, if moderate outliers formed groups of observations distancing with a consistent pattern from the PCA model, we would like to detect such clusters and separately run TSR for PCA-MB on them. The following section describes the strategy based on the $K-$means algorithm for clustering of moderate outliers. However, we emphasize that this strategy is described as a clustering step within RadarTSR but not as a standard and standalone clustering algorithm.

Clustering of moderate rowwise outliers

To determine whether rowwise outliers are single or grouped, we fit a PCA model on the residual matrix $\dot{\boldsymbol{E}}$ of dimensions $N'' \times K$, where N'' represents the number of moderate rowwise outliers identified in Step 5 (Section 6.2.1). The relevant A'' principal components are identified by setting a threshold of 10% on the explained variance. If any PCs surpass this threshold, the clustering process proceeds. It is important to note that a minimum of 5 observations is also required to establish the existence of a cluster. This threshold on the minimum number of observations (c_{n_y}) is a configurable hyperparameter of the RadarTSR algorithm, allowing users to adapt it to their specific dataset.

Next, the scores matrix $\boldsymbol{T}^{(\dot{\boldsymbol{E}})}$ of dimensions $N'' \times A''$ obtained from the residual PCA model is inputted into a K-means algorithm [107]. The K-means algorithm models the data using C cluster means iteratively recalculated after assigning all objects the cluster label of the closest mean ($\widehat{\boldsymbol{\mu}}^{(y)}$ with $y \in 1, \ldots, C$). Therefore it minimizes the within-cluster sum-of-squared errors (Equation 6.4).

$$wcssq = \sum_{y=1}^{C} \sum_{\boldsymbol{t}_i^{(\dot{\boldsymbol{E}})} \in C_y} ||\boldsymbol{t}_i^{(\dot{\boldsymbol{E}})} - \widehat{\boldsymbol{\mu}}^{(y)}||_2^2 \tag{6.4}$$

RadarTSR defaults to using the Euclidean distance to determine the closest mean to each observation, and to determine the number of clusters C, there are two options. Since the appropriate number of clusters can vary depending on the research questions of each user, a manual option is provided, based on visualizing the distribution of the residual scores and the *wcssq* (Equation 6.4) for each number of clusters C, which is similar to a scree plot. Appendix 6.C contains these two outcomes used to determine the number of clusters for real datasets in Section 6.5.2. Although this approach should always be preferred, there is also an automatic option based on the Calinski-Harabasz criterion

(Equation 6.5), which selects C as the value that maximizes the ratio between the overall between-cluster variance and the within-cluster variance (i.e., the $wcssq$)[108].

$$VR = \frac{\sum_{y=1}^{C} ||\widehat{\boldsymbol{\mu}}^{(y)} - \widehat{\boldsymbol{\mu}}||_2^2}{wcssq} \tag{6.5}$$

Finally, it is important to mention that the cluster labels in the returned vector $\boldsymbol{y}$ are sorted in descending order based on their sample sizes. This means that label "0" corresponds to the most populated cluster, followed by "1", and so on.

6.3 Datasets

The comparative study contains six case studies representing different data structures. These cases included two simulated data sets and four real data sets. Section 6.3.1 explains how various artefacts were produced for the simulated datasets. Section 6.3.2 describes the real datasets, outlining their main characteristics, contextualising them, and arguing for their inclusion as part of this chapter.

6.3.1 Simulated datasets

For simulated cases, we generated two clean data structures $\boldsymbol{X}_0$ of $N \times K$ dimensions, referred to in this section as the "wide" ($N < K$) named A09 and the "long" ($N > K$) data set named MDI Sim.

The wide clean data set ($N = 100$, $K = 200$, $A = 6$) was generated following the procedure from [48], using the A09 correlation structure. The A09 correlation structure is given by the expression $\rho_{ij} = (-0.9)^{|i-j|}$, which was applied for each off-diagonal entry (i.e., $\forall i, j \in 1, \ldots, K;\ i \neq j$). Then, the spectral decomposition of the covariance matrix was obtained as:

$$\boldsymbol{\Sigma} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{P}^{\top}$$

Secondly, the diagonal elements of $\boldsymbol{L}$ are replaced by:

$$diag(30,\ 25,\ 20, \ldots, 5,\ 0.098, 0.0975, \ldots, 0.0020, 0.0015)$$

. This way, 6 PCs, whose importance is determined by the eigenvalues of the covariance matrix, will explain 91.5% of the total variance.

The long clean data set ($N = 100$, $K = 10$, $A = 3$) was taken from the simulated example used in [109]. The correlation structure was calculated from the original reference matrix for this data set. Each clean data set included different simulated artefacts, as explained in the following sections.

Simulation of missing at random missingness

To simulate the MAR pattern, missingness must depend on the value of other cells but not on the value of the missing cell. Thus, to generate a percentage p_{MD} of MAR missing cells, the following strategy was applied:

$$\boldsymbol{u}^{(j)} = |\boldsymbol{x}^{(j-1)}| + |\boldsymbol{x}^{(j+1)}| \quad if \quad u_{ij} \geq (100 - p_{MD}) \longrightarrow x_{ij} = NA \tag{6.6}$$

The formula in Equation 6.6 uses the same strategy for MAR missing data simulation as in [48]. For the j-th column $\boldsymbol{x}^{(j)}$ from $\boldsymbol{X}$, a column $\boldsymbol{u}^{(j)}$ is obtained by the addition of adjacent columns $\boldsymbol{x}^{(j-1)}$ and $\boldsymbol{x}^{(j+1)}$ in absolute value. Then, cells x_{ij} among the highest percentile p_{MD} of u_i values are set to missing. The results for the MAR simulations are in Appendix 6.D.

Simulation of cellwise outliers

Once the clean dataset has been simulated, a certain percentage of cells are randomly selected, as for the MCAR data simulation. These cells are replaced by the value $\gamma\sigma_j$, where σ_j^2 is the j-th diagonal element of $\boldsymbol{\Sigma}$ and γ ranges from 0 to 20.

Simulation of single rowwise outliers

To simulate results from Simulations with single rowwise outliers, we used the Simulation of Controlled OUTliers algorithm (SCOUTer [94]), also explained in Chapter 5 of this thesis. This strategy proposes a general framework to simulate outliers having specific properties concerning a given PCA model. Provided a data matrix and a reference PCA model, SCOUTer shifts the specified observations to achieve a target SPE and/or T^2 value. Moreover, the number of intermediate steps and the linearity between the intermediate SPE and/or T^2 values are also tunable options.

To use SCOUTer in these simulations, a PCA model was built with the same reference matrices as in the other simulations (following an A09 or ALYZ

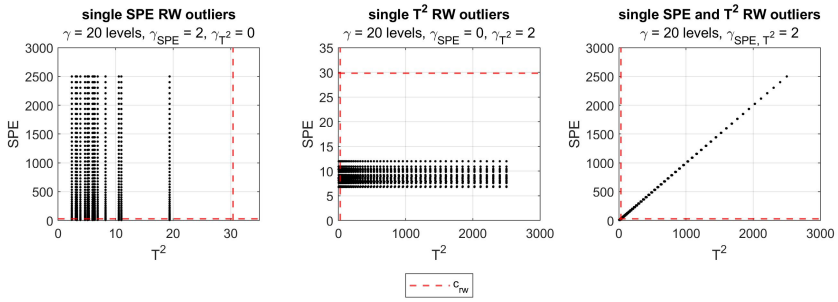


Figure 6.2: Distance plot of the simulated outliers for the case with moderate outliers (first column of plots), with extreme outliers case (second column of plots), and with outliers both for the SPE and the T^2 (third column of plots).

correlation structure). Afterwards, the level of outlyingness achieved by the simulation procedure from [48] was set as a target for the SCOUTer function.

This way, three scenarios could be achieved: simulation of extreme outliers, simulation of moderate outliers, and simulation of severe and extreme outliers. In Figure 6.2, both distance plots for the moderate outliers case and the extreme outliers case are shown. Lines represent the gradual shift of each observation along the 50 steps. As it can be seen, SPE and T^2 values are more spaced towards the final steps of the shift. Thus, there is more concentration of outliers on lower levels of distance, exploring more carefully the point at which outliers become "outlying enough" for each algorithm. The third case is the one in which the SPE and the T^2 vary simultaneously. In this case, models will have to deal with the influence of bad leverage points. These outliers can generate artificial PCs due to their big variance (high T^2). Moreover, these artificial PCs will be unrepresentative of the actual correlation of most of the data set (high SPE).

Simulation of grouped rowwise outliers

Once the structure defining the non-outlying cluster ($\boldsymbol{X}_0$) has been defined, the reference covariance structure is altered for the outlying cluster ($\boldsymbol{X}_1$) by using the first residual eigenvector ($\boldsymbol{\nu}_{A+1}$) as the mean vector of the outlying distribution.

$$\boldsymbol{X}_0 \sim N\left(\mathbf{0}_K, \boldsymbol{\Sigma}_0\right)$$

$$\boldsymbol{X}_1 \sim N\left(\gamma \boldsymbol{\nu}_{A+1}, \boldsymbol{\Sigma}_0\right)$$

This mean vector is multiplied by the factor $\gamma \in 1 \ldots 50$ to tune the magnitude of the outliers.

6.3.2 Real datasets

The first case study, from chemometrics, is the *NIR spectra* dataset. Since this dataset was in the proposal of TSR for PCA-MB with missing data [109], we considered it an interesting case for comparison. It contains K= 401 wavelengths (750 – 1550 in 2 mm increments) of N = 40 diesel fuels obtained at the Southwest Research Institute (SWRI) on a project sponsored by the U.S. Army [110].

The second case study is the *MRI breast* dataset, with perfusion magnetic resonance imaging (MRI) data. This medical image modality expresses each pixel's intensity as the concentration of an injected contrast agent, capturing the diffusion of the contrast in the tissue over a temporal sequence. The analysis of the contrast's washout dynamic can be used to develop effective techniques for cancer diagnosis [111]. The dataset has a $pixels \times frames$ structure, with dimensions of $N = 23193$ pixes (151×432 originally) and $K =$ 6 frames. Clinical experts classified beforehand the pixels as either healthy (i.e., non-outlying class) or corresponding to the Region Of Interest (ROI, i.e., rowwise outliers), which presented a lesion indicating tumour development initiation.

The third case study is the *Glass spectra* dataset. This dataset [112], [113] contains spectra with $K = 750$ wavelengths of $N = 180$ archaeological glass samples from the 16^{th}-17^{th} century, analyzed via electron-probe X-ray micro-analysis (EPXMA). It has been used in several works about robust statistics and clustering, including the work proposing the MacroPCA algorithm [48]. Some spectra were measured with a contamination layer on the detector's surface, decreasing the detector efficiency [114]. This indicates the existence of at least one cluster of grouped rowwise outliers in the dataset.

Finally, the fourth case study is the Digitized Palomar Observatory Sky Survey (DPOSS), a digital version of a three-band photographic survey of the northern sky (POSS-II), which was released to the astronomical community [115] From this database of celestial objects, authors in [48] selected N = 20,000 entries at random, and we also used this same query. Rows represent celestial objects, and $K = 21$ columns represent seven measurements taken with J, F, and N

emulsions, i.e., different colour bands. In this dataset, 50,2% of entries are missing, and 84,6% of rows contain missing entries.

6.4 Comparative study

The case studies were used to compare RadarTSR to ICPCA [84], MacroPCA [48] and TSR [85]. ICPCA and TSR were added to the comparison because they can be seen as the least-squares versions of the missing data imputation algorithms used in MacroPCA and RadarTSR, respectively. The MacroPCA algorithm was included as the best state-of-the-art approach for PCA-MB in the presence of missing data, cellwise and rowwise outliers. Whereas RadarTSR and TSR were executed in Matlab, MacroPCA and ICPCA calculations were run in R, using the *cellWise* package [116].

For both simulated case studies, different scenarios were generated. All the details for the simulation of the artefacts have been described in Section 6.3.1, and this paragraph limits its content to define the values of the parameters used for the simulations. Missing data were generated with seven incremental levels of missing cells, ranging from 5% to 80%, following MCAR and MAR patterns. It was essential to apply the imputation methods since they rely on the assumption of ignorable missingness mechanisms. As per cellwise outliers and rowwise outliers, the parameter γ denoting the distance of the outliers, i.e., the contamination level, was set to range from 0 to 20 for cellwise outliers and from 0 to 50 for rowwise outliers. When missing data appeared in combination with outliers, the percentage of missing cells was always set to 20%. When only one type of outlier (rowwise or cellwise) was generated, its percentage was set to 20% of contaminated rows or cells. If cellwise and rowwise outliers coexisted in the matrix, 10% of each outlier type was generated. Each scenario was repeated 50 times, changing the affected cells and/or rows.

For the *NIR spectra* dataset, MCAR missingness was simulated, ranging from 5% to 80%. The missingness generation was repeated ten times, changing the cells set as missing to estimate the variability expected for the error. For the *MRI breast* dataset, pixels from the ROI (the region with damaged tissue indicating potential tumoural development) were considered rowwise outliers. Since the ROI was constituted only by 84 pixels (0.362 % of rows), a low percentage of MCAR missing data was simulated, setting 10% of the entries as missing to include outliers and missing data. For the *Glass spectra* dataset, since it was originally complete, 40% of MCAR missing data were simulated to include a dataset with all the artefacts (missing data, rowwise outliers, and

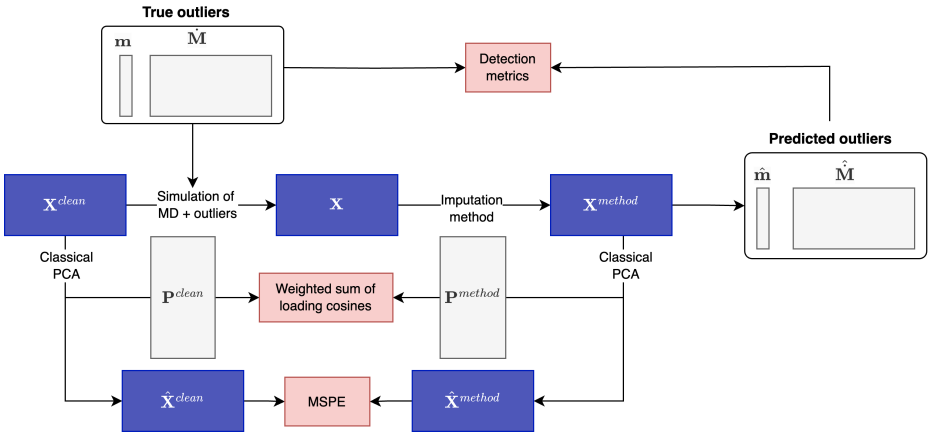


Figure 6.3: Flux diagram illustrating the methodology followed to obtain the MSPE (Equation 6.7), the weighted sum of cosines between loadings (Equation 6.8), and the detection metrics from Table 6.1, used for the comparative study.

cellwise outliers). Finally, for the *DPOSS stars* dataset, it already presented all sorts of events: missing data ($> 50\%$ of entries were missing, and $> 80\%$ of rows had missing values) and outliers, potentially both rowwise and cellwise.

Figure 6.3 illustrates the framework to compute all the performance metrics used for the comparison. For each generated matrix, a classical PCA was applied to the matrix $\boldsymbol{X}^{clean}$ of dimensions $N' \times K$, where N' refers to rows that were not replaced by rowwise outliers (i.e., those for which $m_i = 0$). This yields a reconstructed matrix $\hat{\boldsymbol{X}}^{clean}$, which is compared to the reconstructed matrix $\hat{\boldsymbol{X}}^{method}$ yielded by each one of the methods. Both matrices, $\hat{\boldsymbol{X}}^{clean}$ and $\hat{\boldsymbol{X}}^{method}$, were used to compute the mean squared prediction error (MSPE, Equation 6.7).

$$MSPE = \frac{\sum_{\forall i \in m_i = 0} \sum_{j=1}^{K} \left(\hat{x}_{ij}^{clean} - \hat{x}_{ij}^{method} \right)^2}{N' \cdot K} \tag{6.7}$$

It is important to remark that among the outcomes yielded by MacroPCA, the "cell imputed" option was used in this comparative study. This matrix imputes missing cells for all rows and cellwise outliers (which cannot appear in outlying rows). Additionally, the outcome used to compute the $MSPE$ obtained by RadarTSR is the matrix $\tilde{\boldsymbol{X}}$, which has the missing data imputed

by the model corresponding to the cluster of each observation, and the cell-wise outliers corrected for the reference cluster using the reference model.

Since the $MSPE$ might vary depending on the dataset, it is only a relative value that lets us compare across methods. However, it does not hold an absolute interpretation, i.e., we cannot know if an $MSPE$ value is high or low. To measure the distortion of the PCA model fitted by each method, we also compared loading vectors yielded by each method to the ones obtained from the original clean dataset (Equation 6.8). The dot product between each pair of homologous loading vectors is calculated and then weighted by the percentage of variance explained by that PC according to the clean PCA model.

$$wcos_P = \sum_{j=1}^{A} \left(\boldsymbol{p}_j^{clean}\right)^\top \cdot \boldsymbol{p}_j^{method} \cdot \frac{\lambda_j}{\sum_{j=1}^{A} \lambda_j} \tag{6.8}$$

Finally, several metrics evaluating the detection of cellwise and rowwise outliers were also calculated to compare MacroPCA and RadarTSR (Table 6.1).

Table 6.1: Metrics used to evaluate the results with simulated datasets. Outlying elements (cells and/or rows) are referred to as Positives (P), and then TP stands for True Positives, TN for True Negatives, FP for False Positives, and FN for False Negatives.

Metric	**Evaluation**	**Expression**
Sensitivity	Probability of detecting an outlier	$TP/(TP+FN)$
Precision	Probability for a predicted outlier of truly being so	$TP/(TP+FP)$
Specificity	Probability of detecting a non-outlier	$TN/(TN+FP)$

To assess if the differences between methods were statistically significant, a 3-factor mixed-effect ANOVA model was fitted for each case study, using Method (4 levels) and Artifact (missing data percentage with seven levels, cellwise outlyingness with 20 levels, or rowwise outlyingness with 50 levels) as fixed effects, and Repetition (50 levels for simulated case studies and 10 for real dataset case studies), as a random-effect factor used as a blocking factor.

For the MSPE, a logarithmic transformation was used to expand the differences in scale between lower and higher levels of the artefacts and improve the ANOVA normality assumption. Its corresponding 95% Least Significant Difference (LSD) intervals were plotted (Figures 6.4 to 6.20 and Appendices 6.B and 6.C), with non-overlapping LSD intervals indicating statistically significant differences between the corresponding group means, i.e., showing that some effect or interaction was statistically significant (p value <0.05) in the ANOVA model.

For the *MRI breast*, as the percentage of outlying values was very low (under a 1%), simulating a range of missing values could lead to the deletion of all entries holding the information about the outlying rows. Therefore, only 10% of MCAR missing data was simulated. For the *Glass* and *DPOSS stars* datasets, neither the $MSPE$ nor the detection metrics from Table 6.1 could be calculated because there is not a clean dataset (without any outlier) to use as reference. Therefore, residual maps were used to assess the comparison between methods. Residual maps are an idea from [59], also used in [48] to represent the autoscaled difference between the original matrix $\boldsymbol{X}$ and the reconstruction of the cell-imputed matrix $\hat{\boldsymbol{X}}$ yielded by the PCA model. It is worth mentioning as well that each technique used its own location and scale estimators to normalize the residual matrix, as in [48], i.e., ICPCA and TSR used the least-squares mean and standard deviation, while RadarTSR and MacroPCA used the *robloc* and *robscale* ones.

The resulting matrix is represented with colour-coded cells (see Supplementary Figures 12 and 15). Intense blue is used for extreme negative residuals ($\dot{r}_{ij} \leq -z_{\alpha/2}$), becoming more yellow as residuals tend to zero and turning red as the residuals increase and grow to high positive values ($\dot{r}_{ij} \geq z_{1-\alpha/2}$). Next to the residual map, a grey scale bar indicates the outlyingness of each row. This outlyingness vector $\boldsymbol{d}^{model}$ is calculated as the scaled orthogonal distance of the N cell-imputed observations (i.e., as $\dot{SPE}_i/c_{rw}^{SPE}$ for RadarTSR or as $\dot{OD}_i/c_{OD}$ for MacroPCA), and it attains its darkest colour when the orthogonal distance exceeds the threshold obtained by the fitted PCA model for each observation (i.e. when $d_i^{model} \geq 1$). The more outlying a row is, the higher its distance d_i and the darker it will appear. On the contrary, rows with a low distance to the PCA model will appear white.

6.5 Results

6.5.1 Simulations

The following figures show the average of all performance criteria over the 50 repetitions. The shaded region represents the LSD intervals at a 95% confidence level. As expected, the more missing or outlying values (higher γ values) are considered, the higher the $MSPE$ values. In this section, only the results assuming MCAR missing data are shown. The results with MAR missing data lead to the same conclusions (see Appendix 6.B).

Matrices with missing data

Figures 6.4 and 6.5 show that, for all methods, the $MSPE$ (upper left plots) increases with the percentage of missing cells in the absence of outliers. RadarTSR is overlapped with ICPCA and TSR, which have the best results for most missing data percentages. Yet, MacroPCA usually shows the highest $MSPE$, being closer to the rest for the long dataset (upper left plots in Figures 6.4 and 6.5).

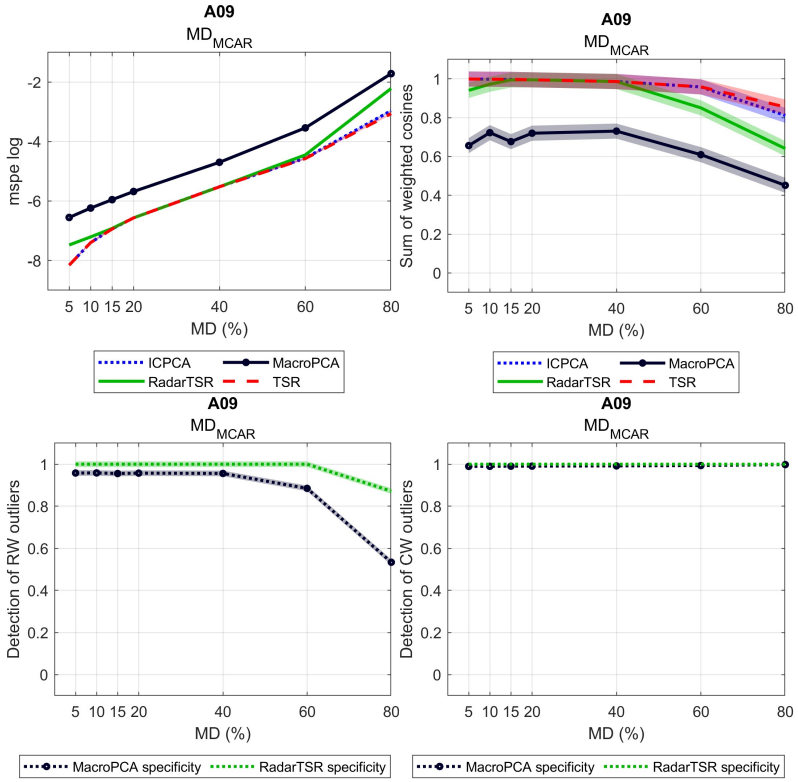


Figure 6.4: Missing data case results for the wide dataset. The upper left plot shows the results for the $MSPE$, the upper right plot shows the weighted sum of cosines between loadings, and the lower left and lower right plots show the detection metrics for rowwise and cellwise outliers, respectively. The x-axis of each plot denotes the MD percentage. The dotted, circles, dashed, and solid lines denote the results of ICPCA, MacroPCA, TSR, and RadarTSR, respectively. The shaded areas represent the 95% LSD confidence intervals of the metrics obtained by each method.

Similar conclusions are obtained from the weighted sum of cosines between loadings (Equation 6.4), proving that RadarTSR succeeds in keeping the least-squares performance in the absence of outliers, at least up to 60% of missing data (upper right plots in Figures 6.4 and 6.5).

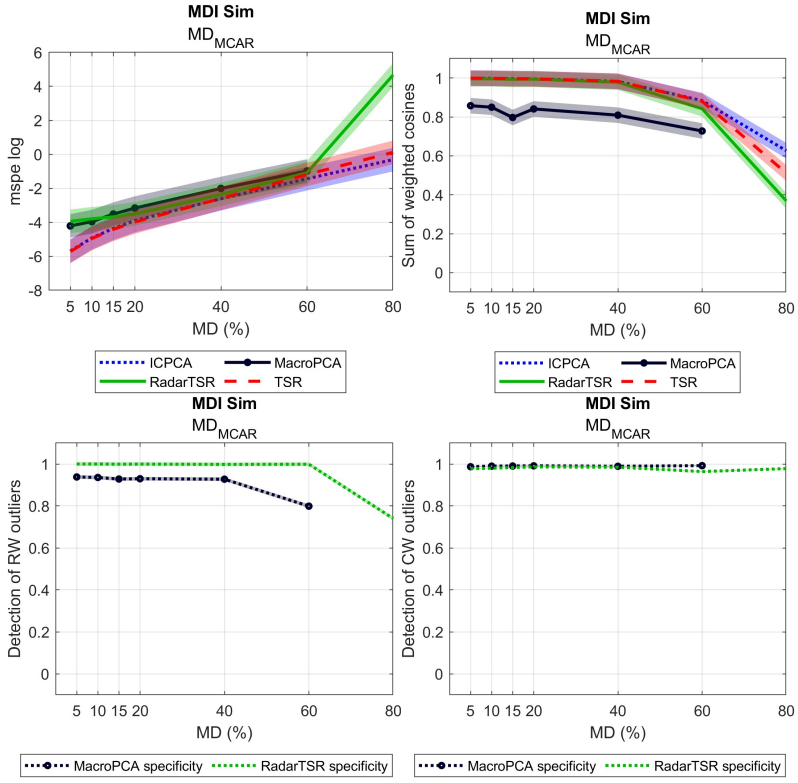


Figure 6.5: Missing data case results for the long dataset. More details are in the caption of Figure 6.4.

Regarding detecting outliers, RadarTSR has higher rowwise specificity and the same high cellwise specificity as MacroPCA. This means that MacroPCA tends to over-detect rowwise outliers. In fact, MacroPCA's rowwise over-detection prevented the obtention of a clean imputed long dataset in a part of the repetitions with 70% of missing data and in all repetitions with 80% of cells missing.

Matrices with missing data and cellwise outliers

Figures 6.6 and 6.7 show in the $MSPE$ (upper left plots) the breakdown of ICPCA and TSR when cellwise outliers are included. In contrast, RadarTSR shows the lowest $MSPE$, the highest similarity to the loadings of the clean PCA model, and both the highest rowwise specificity and cellwise sensitivity.

The increase of the $MSPE$ with the γ (indicating outlier's magnitude) for MacroPCA and RadarTSR is due to their imperfect rowwise specificity and cellwise sensitivity (lower rows in Figures 6.6 and 6.7). It is more noticeable for MacroPCA as its over-detection of rowwise outliers prevents obtaining the cell-imputed version of clean rows, maintaining the effect of cellwise outliers.

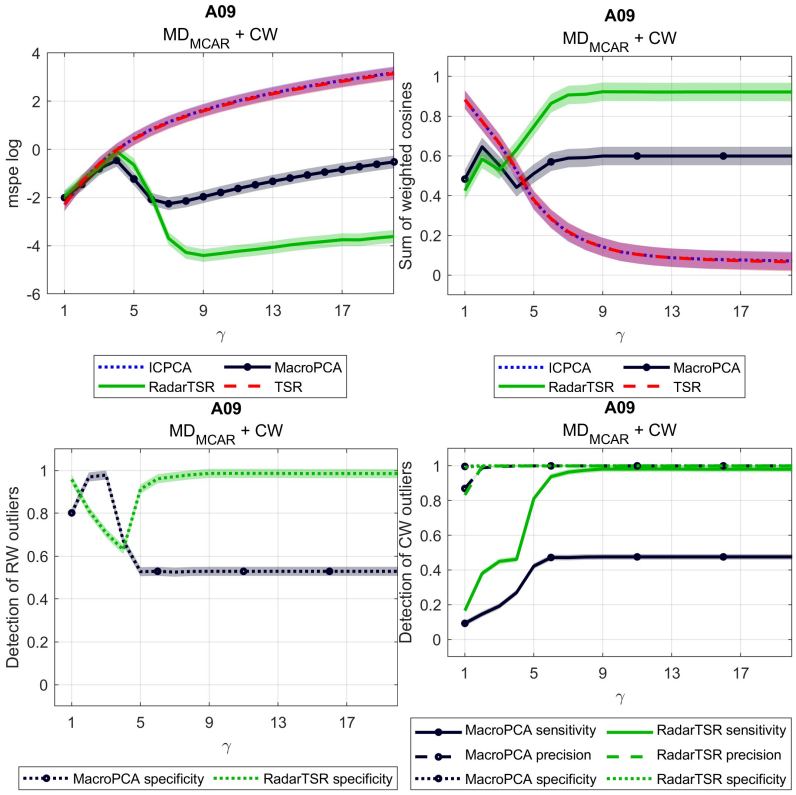


Figure 6.6: Missing data (20%) and cellwise outliers (20%) case results for the wide dataset. The x-axis of each plot denotes the outliers' distance, γ. More details are in the caption of Figure 6.4.

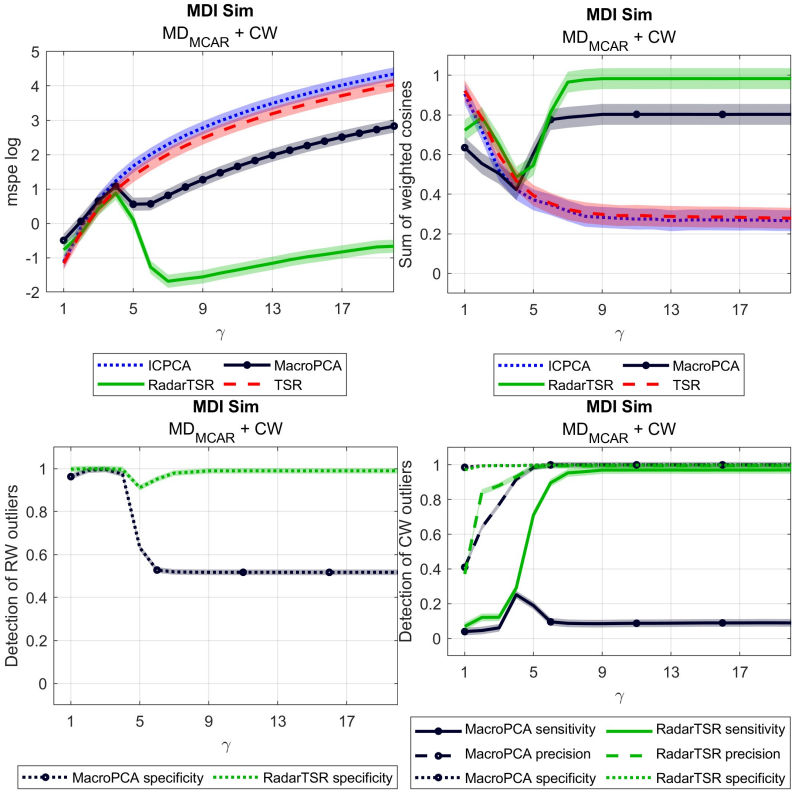


Figure 6.7: Missing data (20%) and cellwise outliers (20%) case results for the long dataset. The x-axis of each plot denotes the outliers' distance, γ. More details are in the caption of Figure 6.4.

Matrices with missing data and single rowwise outliers

Figures 6.8 and 6.9 show the results obtained by adding MCAR missing data and single SPE rowwise outliers. In this case, MacroPCA and RadarTSR obtain a similar MSPE (upper left plots), while both purely least-squares techniques (ICPCA and TSR) show an increase of the $MSPE$ with γ. The weighted sum of cosines between loadings (upper right plots) corroborates $MSPE$ results, with RadarTSR obtaining the loadings closest to the ones from the clean PCA model for both datasets.

Rowwise sensitivity curves (lower left plots) show that RadarTSR detects mild rowwise outliers (at low values of γ) worse than MacroPCA. Besides, the delay

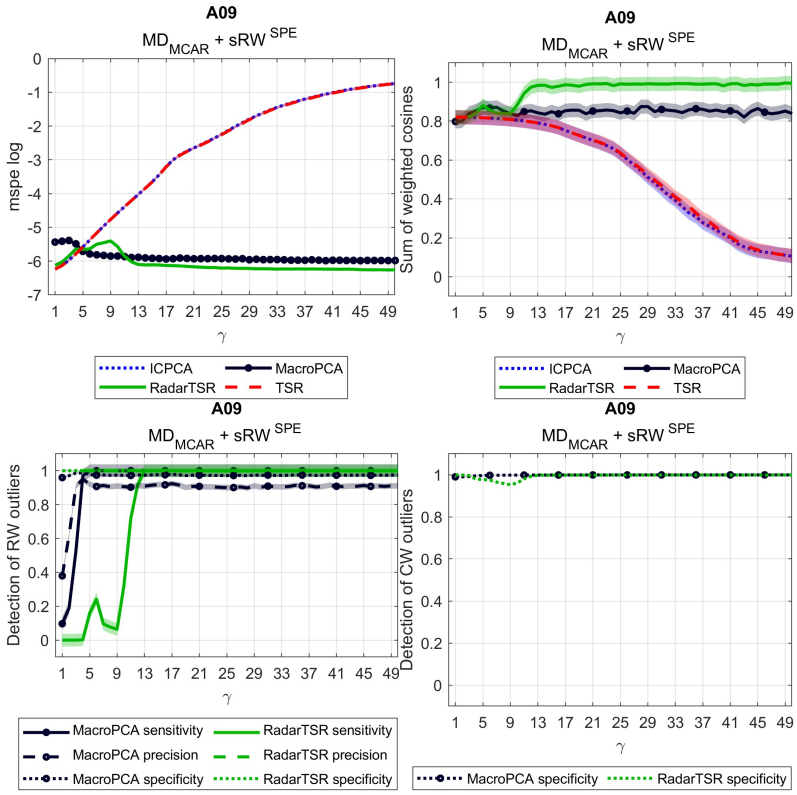


Figure 6.8: Missing data (20%) and single rowwise outliers (20%) case results for the wide dataset. More details are in the caption of Figure 6.4.

of RadarSTR in rowwise outliers' sensitivity is coupled with a decrease in cellwise specificity for the same range of γ values (lower right plots).

This suggests that RadarTSR masks mild rowwise outliers by treating them as rows contaminated with cellwise outliers, correcting their outlyingness. Nevertheless, this masking effect does not come at the cost of distorting the PCA model fitted with RadarTSR more than the one obtained by MacroPCA, as seen in the weighted sum of loading cosines (upper right plots in Figures 6.8 and 6.9). On the contrary, rowwise precision curves (lower left plots) show that MacroPCA tends to over-detect rowwise outliers, as seen previously in Figures 6.6 and 6.7.

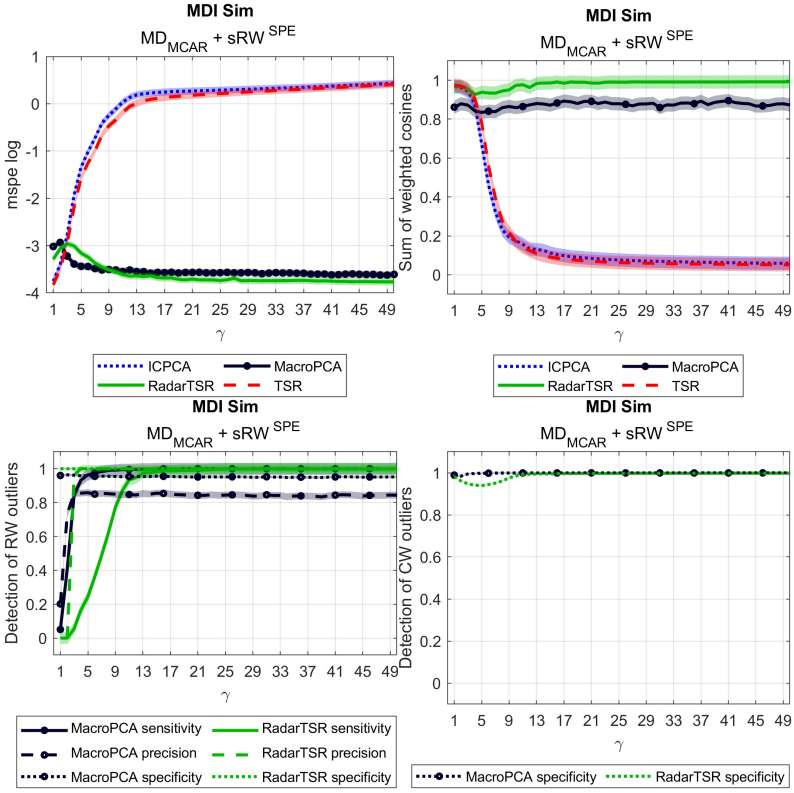


Figure 6.9: Missing data (20%) and single rowwise outliers (20%) case results for the long dataset. More details are in the caption of Figure 6.4.

Figures 6.10 and 6.11 show the results values when single T^2 rowwise outliers and MCAR missing data are present. In this case, ICPCA and TSR obtain a significantly lower MSPE than MacroPCA and RadarTSR. Nonetheless, the order of magnitude of the MSPE $(10^{-5}, 10^{-6})$ makes such differences probably irrelevant in practical terms. This result was expected since purely extreme rowwise outliers still respect the covariance structure of the data.

On a related note, despite being less harmful, the weighted sum of loadings' cosines still displays the effect of extreme rowwise outliers in purely least squares methods (upper right plots in Figures 6.10 and 6.11, respectively). Even if rowwise outliers still respect the covariance structure, classical least squares PCA is sensitive to extreme values because such outliers can inflate

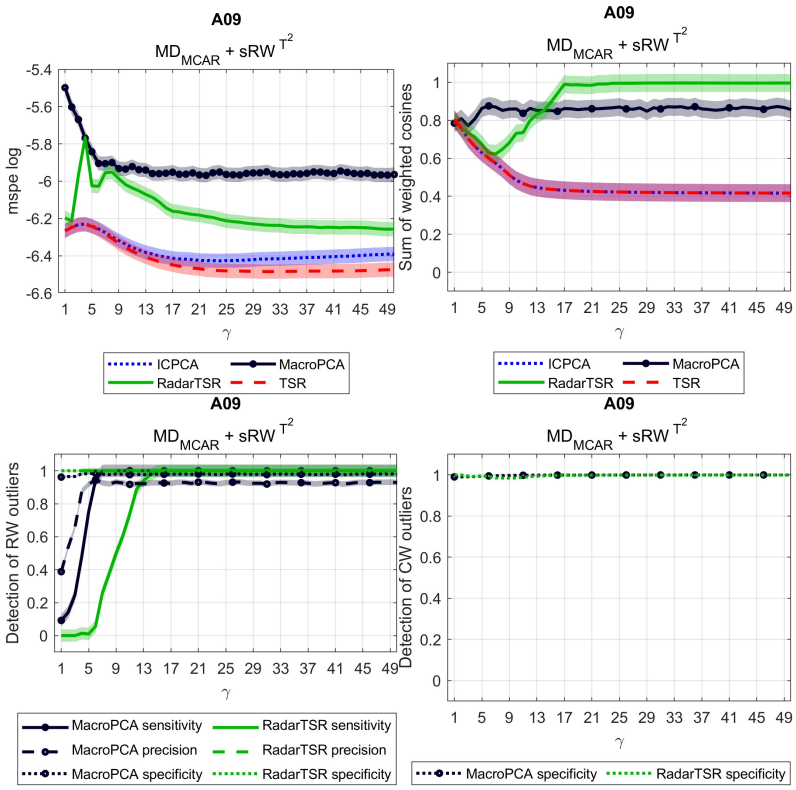


Figure 6.10: Missing data (20%) and single T^2 rowwise outliers (20%) case results for the wide dataset. More details are in the caption of Figure 6.4.

variances or introduce spurious correlations. Besides, single T^2 rowwise outliers seem to have a bigger effect on the wide dataset, where the higher dimensionality already increases the dispersion of the data points, making it easier for inflated variances-covariances to affect the resulting PCA model.

Another noticeable aspect is a more pronounced rowwise outliers' masking effect by RadarTSR for the long dataset (i.e., in low-dimensional cases), as can be seen by comparing Figure 6.11 with Figures 6.9 , 6.10 and 6.11. Although this brings RadarTSR's $MSPE$ and the weighted sum of cosines closer to the values of the rest of the techniques and lowers its rowwise sensitivity and cellwise specificity, it does not prevent RadarTSR from reaching the least affected loadings for the most outlying cases (upper right plot in Figure 6.11).

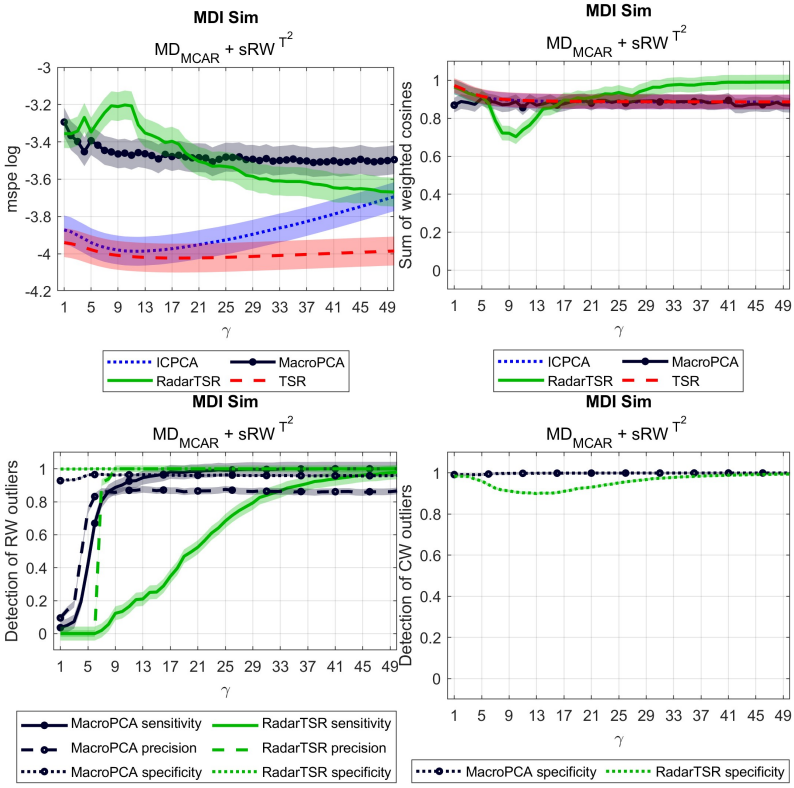


Figure 6.11: Missing data (20%) and single T^2 rowwise outliers (20%) case results for the long dataset. More details are in the caption of Figure 6.4.

Finally, Figures 6.12 and 6.13 show the results when single T^2 and SPE rowwise outliers and MCAR missing data are present.

As it can be seen, the outcomes and conclusions are practically identical to the ones derived from simulations with matrices with missing data and single SPE rowwise outliers (Figures 6.8 and 6.9), being the main difference a better detection of mild rowwise outliers by RadarTSR (lower left plots).

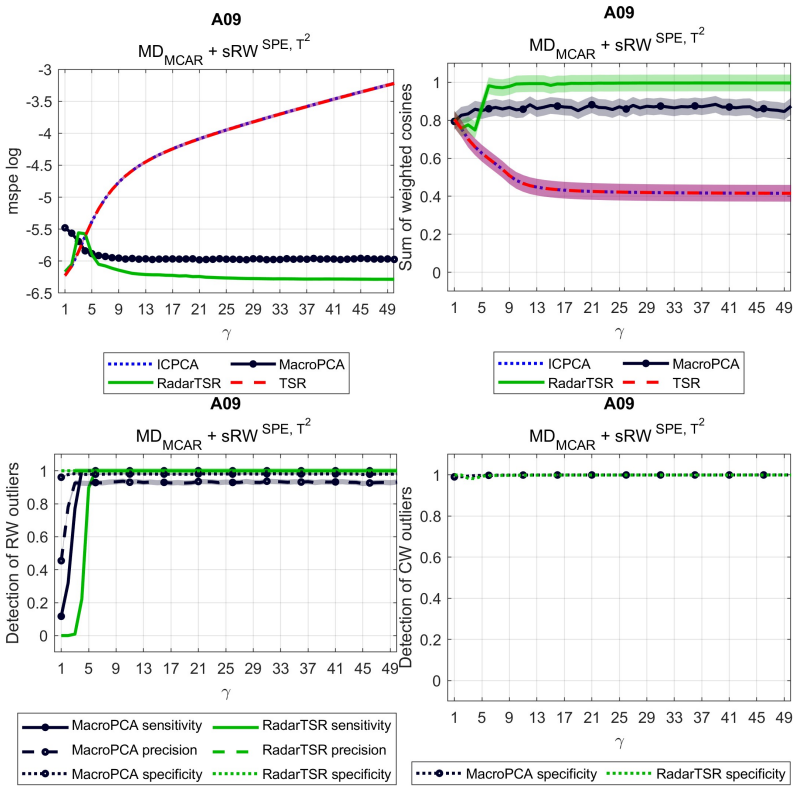


Figure 6.12: Missing data (20%) and single T^2 and SPE rowwise outliers (20%) case results for the wide dataset. More details are in the caption of Figure 6.4.

In the following cases, cellwise outliers were included when the different types of single rowwise outliers were simulated. This is the most complex scenario and also the most likely one to be found in real datasets. As it will be seen, adding cellwise outliers exacerbates, in general, RadarTSR's rowwise masking effect and MacroPCA's rowwise over-detection.

Figures 6.14 and 6.15 show the results when missing data, single SPE rowwise outliers and cellwise outliers are present in a matrix. Since cellwise outliers are present in this scenario, MacroPCA yields higher $MSPE$ values than RadarTSR for both datasets (upper left plots), as the cell-imputed matrix used to calculate the $MSPE$ does not correct cellwise outliers undetected by MacroPCA.

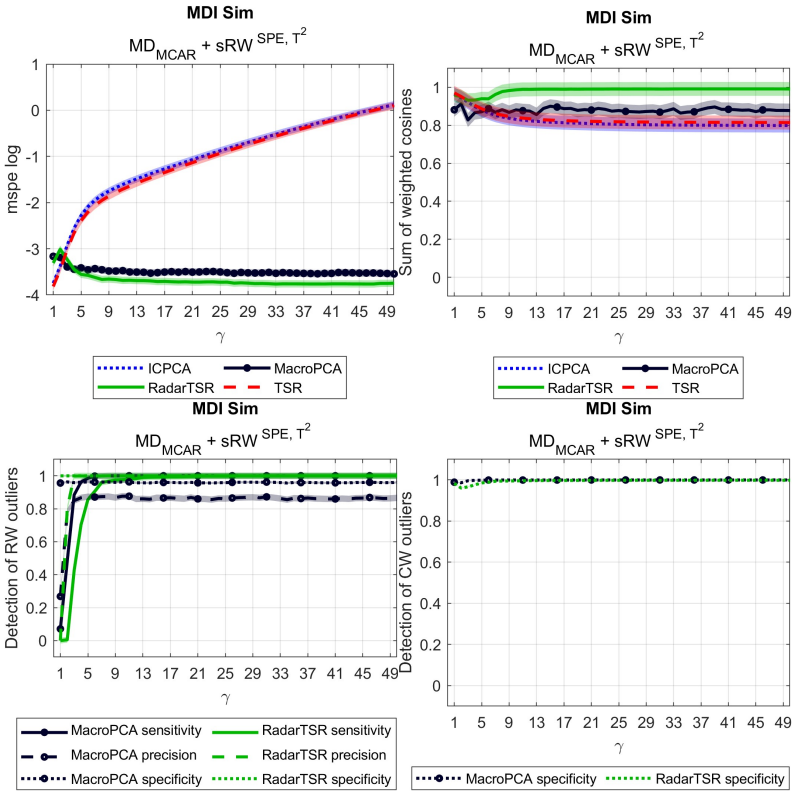


Figure 6.13: Missing data (20%) and single T^2 and SPE rowwise outliers (20%) case results for the long dataset. More details are in the caption of Figure 6.4.

Besides, the masking effect of RadarTSR seems related to the dataset dimension, affecting the long dataset (Figure 6.9) more drastically than the wide dataset (Figure 6.8). Nevertheless, the weighted sum of cosines between loading vectors (upper right plots in Figures 6.8 and 6.9) shows that RadarTSR still has the best performance for the wide dataset, and the long dataset shows, at its worst, an overlap with MacroPCA, which shows the second best performance, followed by ICPCA and TSR.

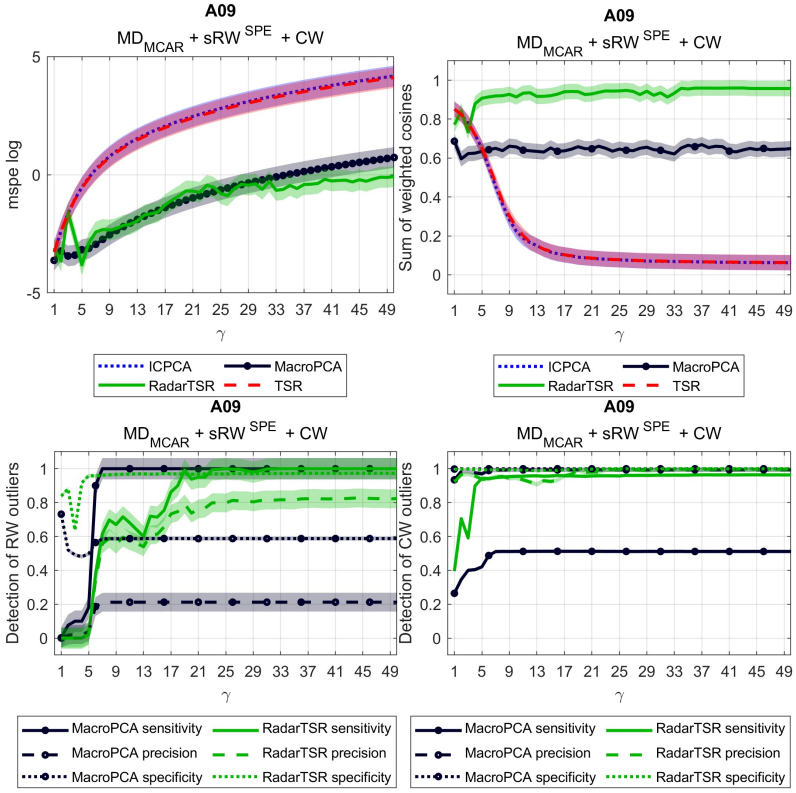


Figure 6.14: Missing data (20%), single rowwise outliers (10%), and cellwise outliers (10%) case for the wide dataset. More details in caption of Figure 6.4.

The following results in Figures 6.16 and 6.17 showcase the effect of adding cellwise outliers to matrices contaminated with single T^2 rowwise outliers. Similarly to Figures 6.10 and 6.11, the performance of purely least squares methods (ICPCA and TSR) is less affected in comparison to cases when single SPE rowwise outliers are present.

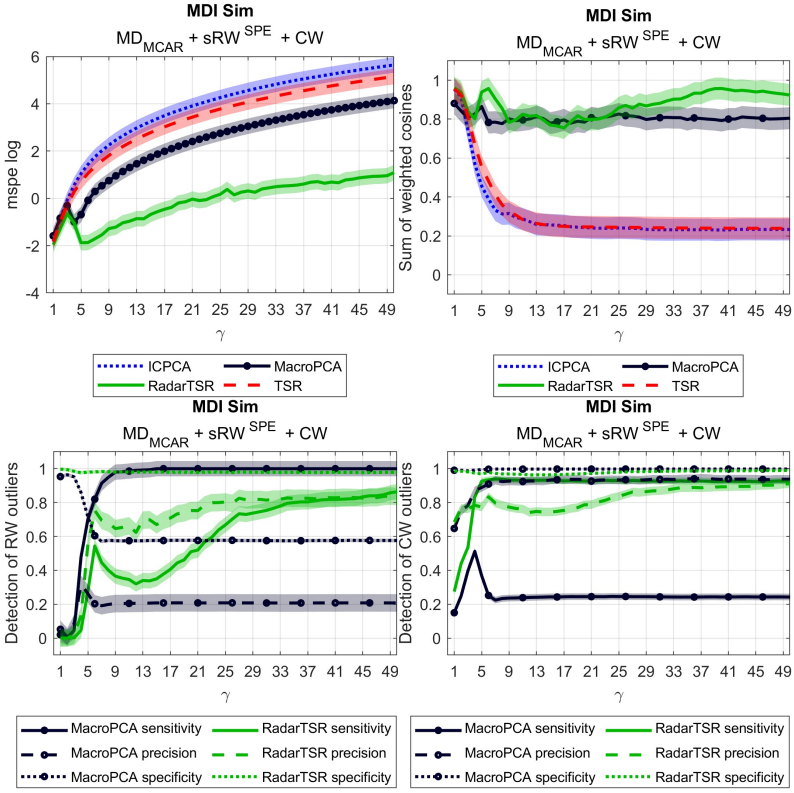


Figure 6.15: Missing data (20%), single rowwise outliers (10%), and cellwise outliers (10%) case for the long dataset. More details in caption of Figure 6.4.

Resembling when only single T^2 rowwise outliers were present (Figure 6.11), the masking effect of RadarTSR is accentuated for the long dataset (i.e., the low-dimensional case, Figure 6.17). This is again appreciated by low rowwise sensitivity values coupled with a decay in cellwise specificity that persists until reaching high values of γ.

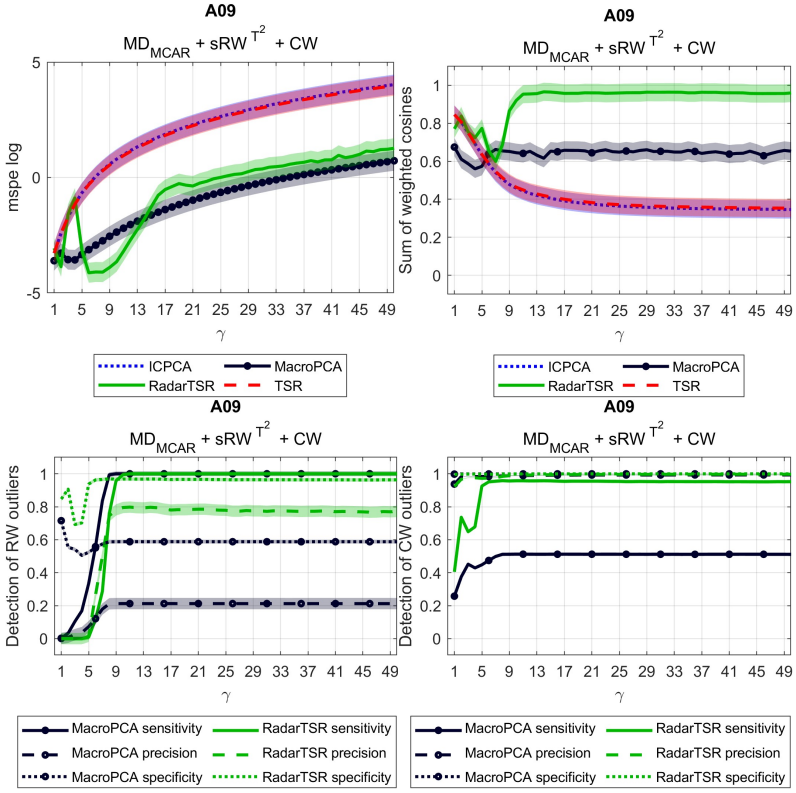


Figure 6.16: Missing data (20%), single T^2 rowwise outliers (10%), and cellwise outliers (10%) case results for the wide dataset. More details are in the caption of Figure 6.4.

Finally, Figures 6.18 and 6.19 show the results when cellwise outliers are added to matrices with single SPE and T^2 rowwise outliers. As can be seen, results resemble, to a great extent, those seen in Figures 6.14 and 6.15, and RadarTSR's masking effect does not persist.

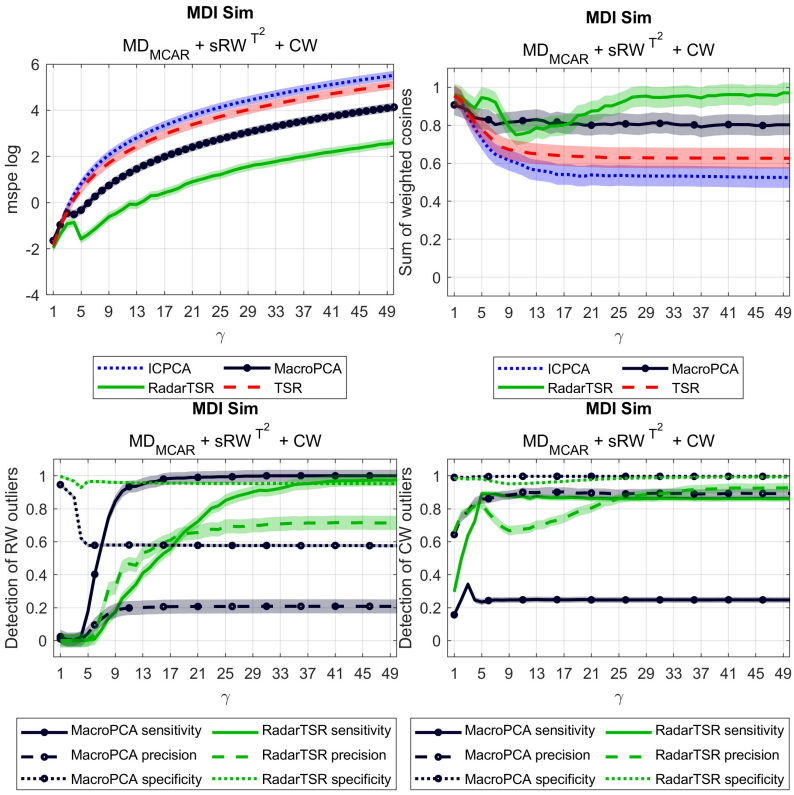


Figure 6.17: Missing data (20%), single T^2 rowwise outliers (10%), and cellwise outliers (10%) case results for the long dataset. More details are in the caption of Figure 6.4.

Matrices with missing data and grouped rowwise outliers

The results of these last simulations include the $MSPE$ for both clean and outlying rows since the latter were generated from the same outlying distribution, and a PCA could be fitted on them.

Figures 6.20 and 6.21 show a very similar picture as the one seen for single rowwise outliers in Figures 6.8 and Figures 6.9, with RadarTSR yielding the lowest $MSPE$ and the least distorted loadings (upper right plots), in general.

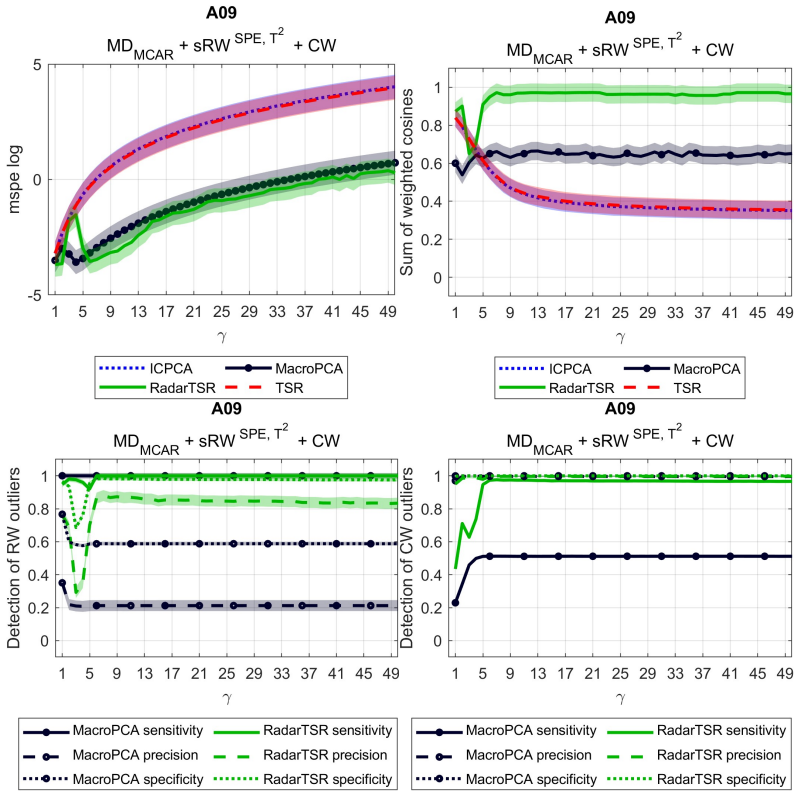


Figure 6.18: Missing data (20%), single T^2 and SPE rowwise outliers (10%), and cellwise outliers (10%) case results for the wide dataset. More details are in the caption of Figure 4 of the main manuscript.

RadarTSR also yields low $MSPE$ values for the outlying cluster of the wide dataset (upper central plot in Figure 6.20). On the contrary, the $MSPE$ of outlying rows from the long dataset increases with the γ parameter (Figure 6.21). Yet, this dataset's rowwise sensitivity curves (lower left plot in Figure 6.21) show that RadarTSR detects these rowwise outliers, even mild ones. This outcome suggests that the clustering step of RadarTSR might be inefficient when dealing with low-dimensional matrices.

Figures 6.22 and 6.23 show similar results when cellwise outliers are also present with missing data and grouped rowwise outliers. RadarTSR is the technique showing the least affected loadings (upper right plots). Still, cellwise outliers

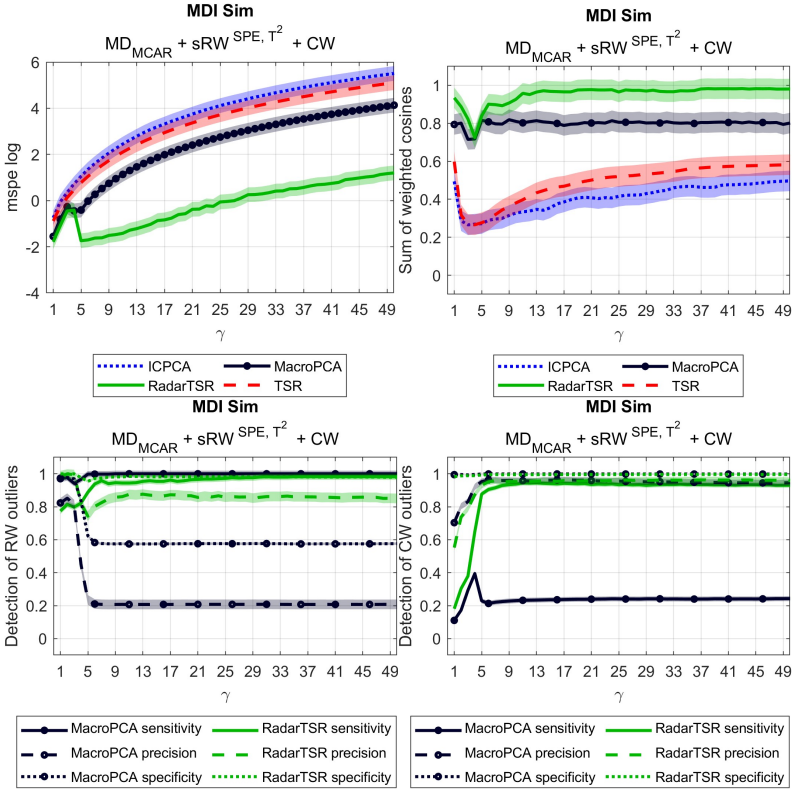


Figure 6.19: Missing data (20%), single T^2 and SPE rowwise outliers (10%), and cellwise outliers (10%) case results for the long dataset. More details are in the caption of Figure 4 of the main manuscript.

and the imperfect precision detecting rowwise outliers increase RadarTSR's $MSPE$ values with γ for both the reference and outlying clusters.

Moreover, RadarTSR also suffers from the mentioned rowwise masking effect with the long dataset, showing a delayed rowwise sensitivity coupled with a delay in cellwise precision. Nonetheless, both ICPCA and TSR yield completely distorted PCA models (upper right plots) and show a monotonic increase of their MSPE, as well as MacroPCA, whose low rowwise precision (lower left plots) prevents the correction of cellwise outliers in clean rows. This results in RadarTSR being the algorithm that yields, in general, the lowest MSPE.

At this point, the shown simulation scenarios have exposed the most important points of the comparative study, compared and summarised in Table 6.2.

Table 6.2: Comparative summary of the metrics (columns) obtained with the simulated scenarios (rows) shown in Section 6.5.1. "$MSPE$" stands for Mean Squared Prediction Error; "wcos$\boldsymbol{P}$" for weighted sum of cosines between loadings;"RW" for rowwise outliers; "CW" for cellwise outliers; "Spec." for specificity; "Sens." for sensitivity; "Prec." for precision; "sRW" for single rowwise outliers; "gRW" for grouped rowwise outliers. Letter "R" means better results of RadarTSR (63.16% of applicable cases); letter "M" means better results of MacroPCA (13.16% of applicable cases); symbol "=" means a tie between RadarTSR and MacroPCA (21.05% of applicable cases), and filled cells are cases in which certain metrics could not be obtained because they were not applicable.

Test case	**MD**		**MD + sRW**		**MD + gRW**	
		CW		**CW**		**CW**
$MSPE$	=	R	=	R	R	R
$wcos\boldsymbol{P}$	R	R	R	R	R	R
RW spec.	R	R	R	R	R	R
RW sens.			M	M	M	R / M
RW prec.			R	R	R	R
CW spec.	R	=	=	=	=	=
CW sens.		R		R		R
CW prec.		=		M		M

First of all, results show that RadarTSR yields $MSPE$ values for non-outlying rows (first row, $MSPE$, of Table 6.2) comparable to the ones obtained by MacroPCA, which is the state-of-the-art method to deal with missing data, cellwise outliers, and rowwise outliers. In cases of the absence of outliers (Figures 6.4 and 6.5), there's a tie in terms of $MSPE$ between RadarTSR and MacroPCA (first row of Table 6.2). Nevertheless, there is a clear superiority in the similarity of loadings yielded by RadarTSR and those fitted with the clean dataset (second row in Table 6.2).

Another substantial difference seen between RadarTSR and MacroPCA is in terms of rowwise sensitivity and precision, where MacroPCA and RadarTSR seem to be antagonistic: whereas MacroPCA shows an over-detection of rowwise outliers (Table 6.2, 4th row), RadarTSR masks them, especially in low-dimensional scenarios, and treats them as cellwise outliers (Table 6.2, 7th column).

In practical terms, the consequences of both phenomena in the $MSPE$ values of the clean rows are almost not appreciable. On the one hand, using

MacroPCA would result in an over-detection of rowwise outliers (relationship between 4th, 5th, and 7th rows in Table 6.2), preventing the correction of cellwise outliers in cellwise contaminated rows. On the other hand, RadarTSR's masking effect (relationship between 4th, 7th, and 8th rows in Table 6.2) could yield a deficient detection of rowwise outliers, as seen especially for the long dataset in Figures 6.14 and 6.22. Yet, as RadarTSR fits a PCA model close to the uncontaminated one, outlying patterns could still be appreciated in the reconstruction error, and its higher rowwise precision (fifth row in Table 6.2) would prevent the loss of clean rows compared to MacroPCA.

An important aspect of commenting on the masking effect of RadarTSR is that it relates to the dataset's dimensionality and accentuates when rowwise and cellwise outliers are present in the matrix. Indeed, it is reasonable that for low numbers of variables (K), it becomes less clear whether a cellwise or rowwise contamination is the one behind outlying values.

To illustrate this, one could consider the minimal multidimensional case with $K = 2$. If a row shows an outlying value, how can it discriminate if such an outlying entry is outlying by itself (cellwise contamination) or represents a different multivariate (rowwise contamination) pattern? This question exceeds the scope of this chapter, but it illustrates an effect that should be considered when applying these techniques and opens an interesting question that could be a matter of further research work. One potential solution to overcome this issue could be to check the matrix of residuals and use specific knowledge about the dataset to determine if rows showing high residuals correspond to a cellwise or rowwise contamination paradigm.

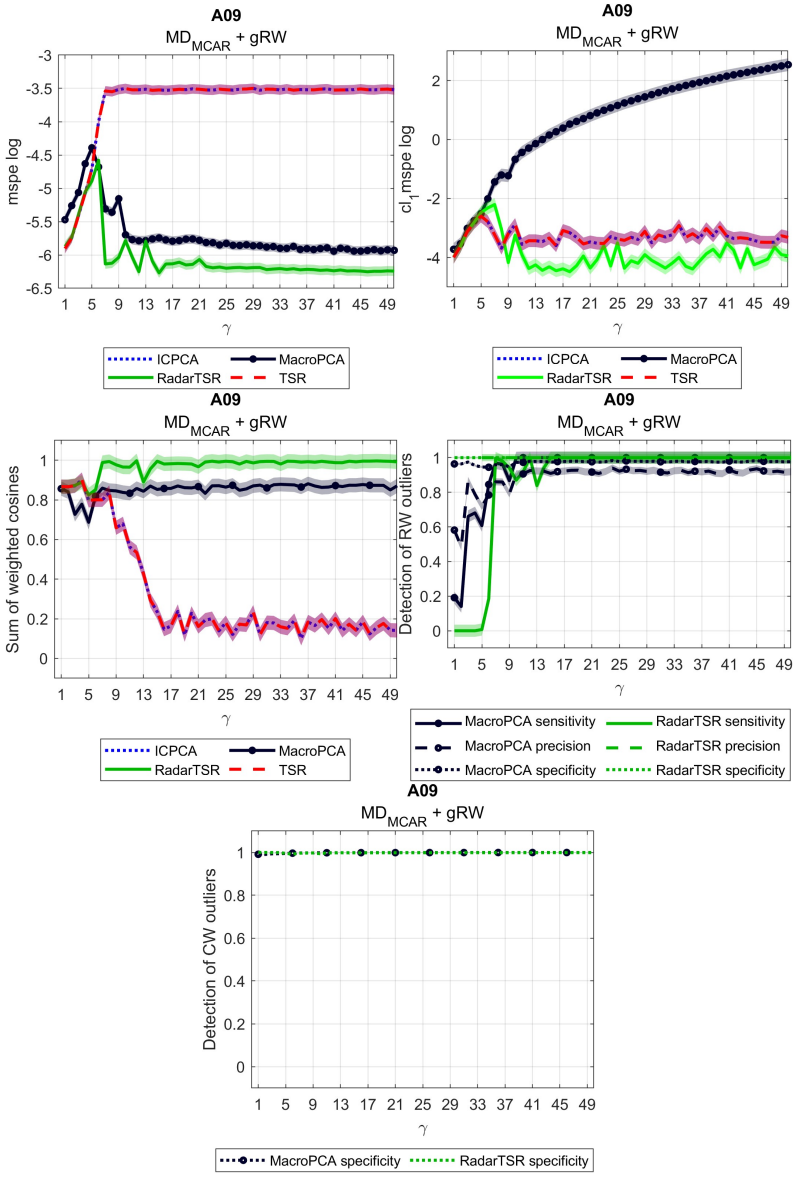


Figure 6.20: Missing data (20%) and grouped rowwise outliers (20%) case for the wide dataset. The upper row of plots shows the MSPE for the clean (left) and outlying (right) rows; the centre-left plot shows the weighted sum of cosines; the centre-right plot shows the rowwise detection metrics; and the bottom plot shows the cellwise detection metrics. More details in caption of Figure 6.4.

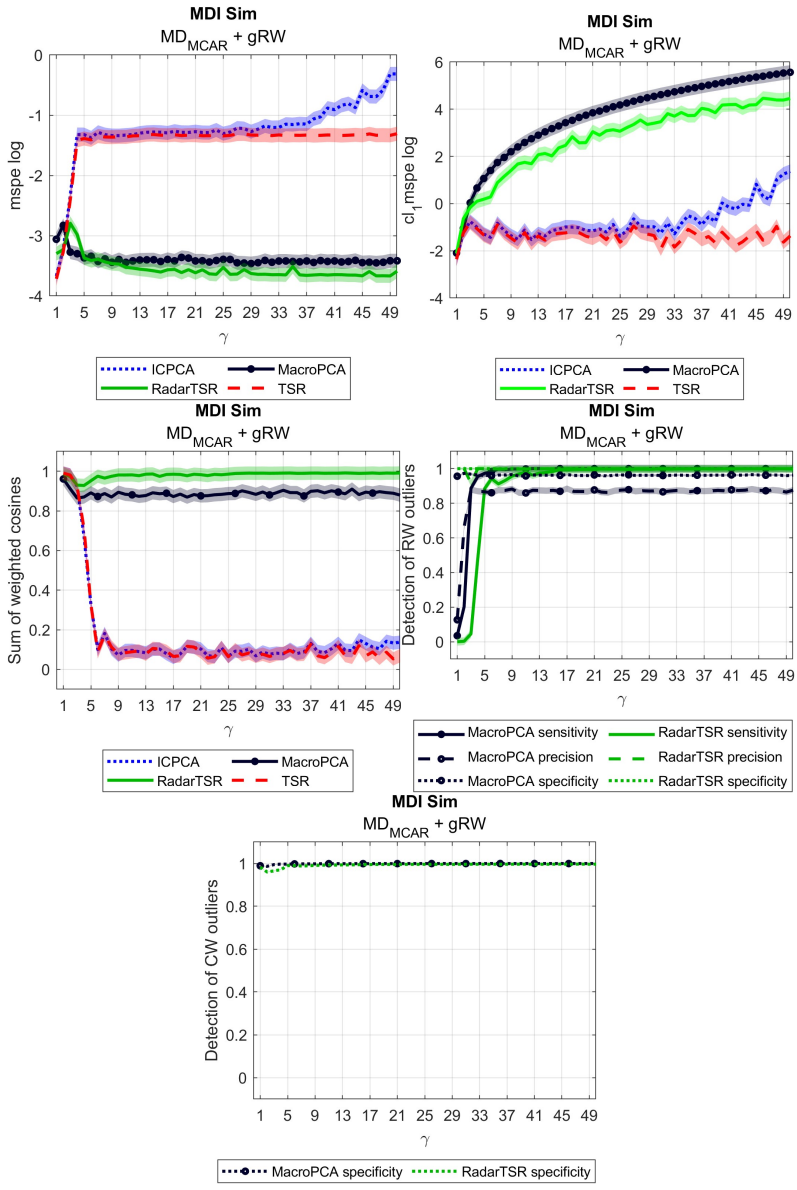


Figure 6.21: Missing data (20%) and grouped rowwise outliers (20%) case for the long dataset. More details in captions of Figures 6.4 and 6.20.

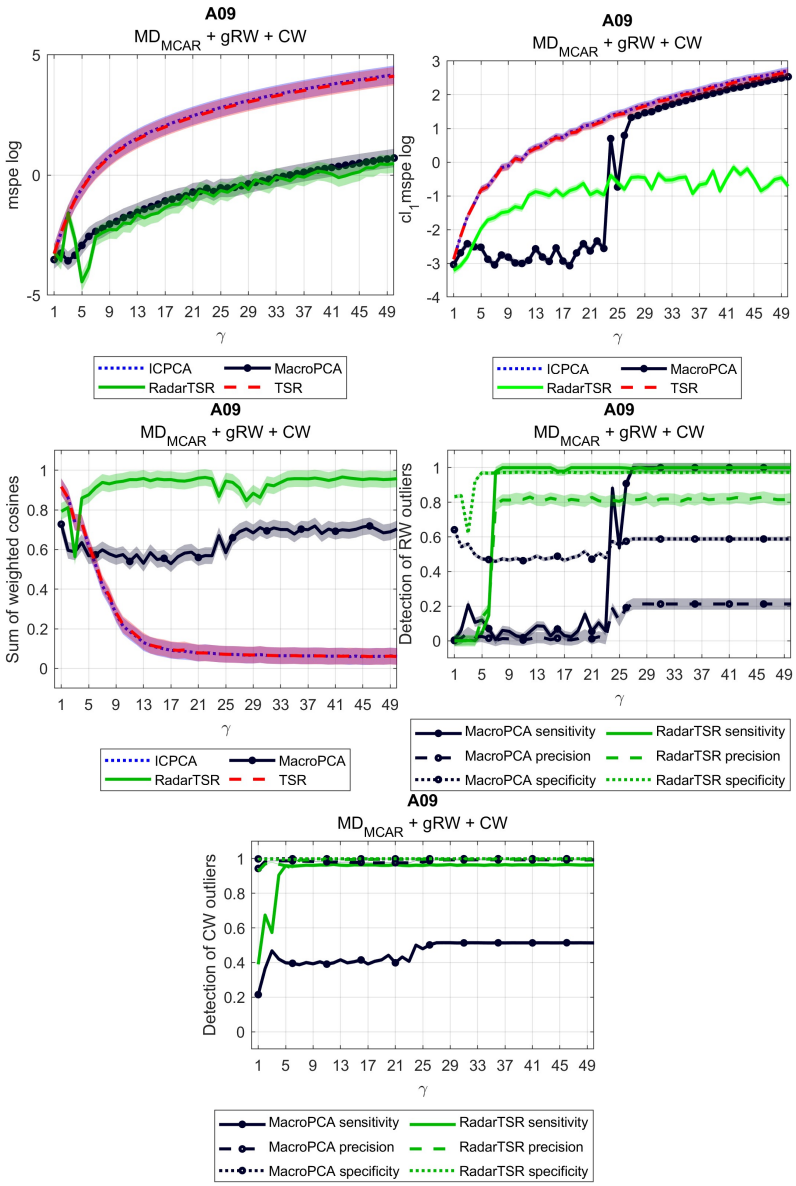


Figure 6.22: Missing data (20%), grouped rowwise outliers (10%) and cellwise outliers (10%) case for the wide dataset. More details in captions of Figures 6.4 and 6.20.

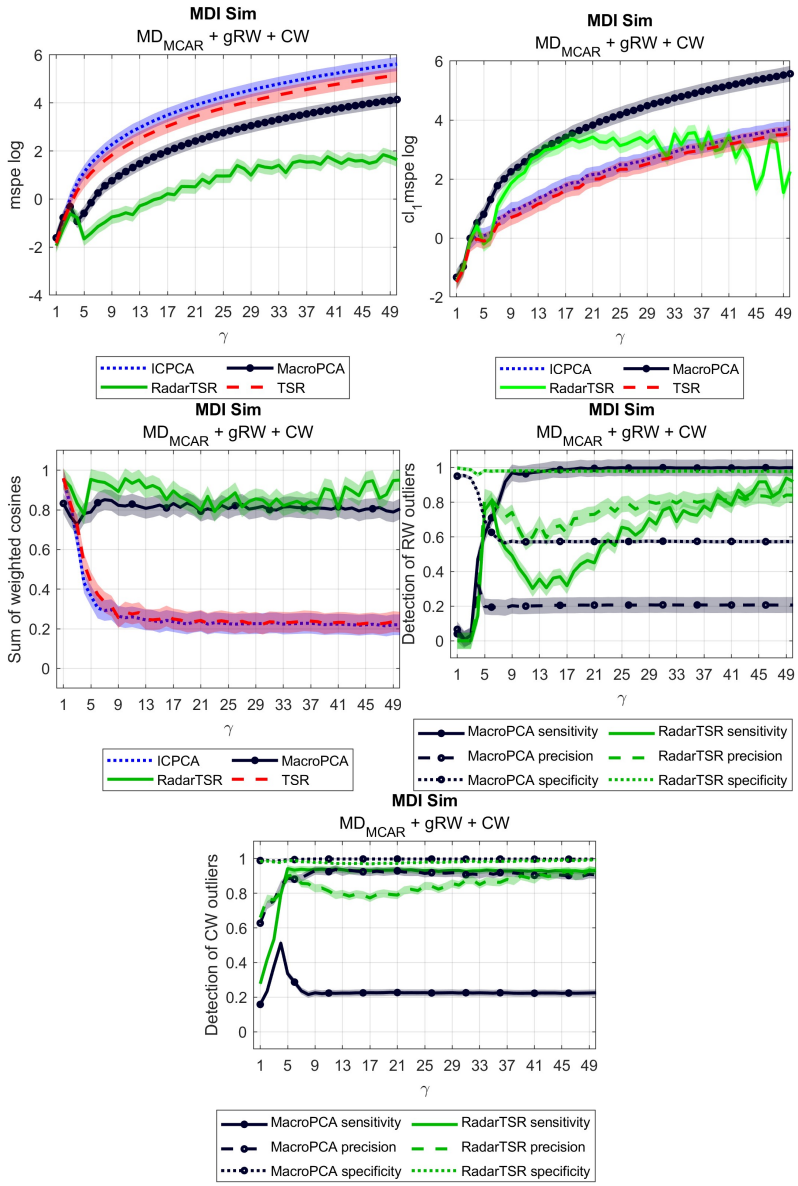
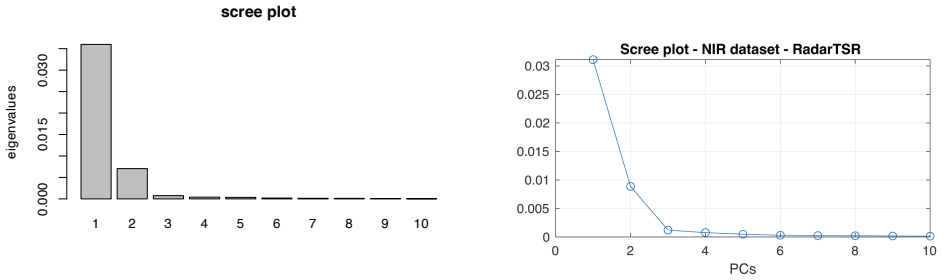


Figure 6.23: Missing data (20%), grouped rowwise outliers (10%) and cellwise outliers (10%) case for the long dataset. More details in captions of Figures 6.4 and 6.20.

6.5.2 Real datasets

NIR spectra data

For this dataset, to select the number of PCs, we did a first inspection of the eigenvalues' scree plot obtained with a matrix with a 5% of missing cells. After examining it, two PCs ($A = 2$) were selected for inclusion in the PCA model. Figure 6.24 shows the scree plots obtained for one iteration of the experiments with MacroPCA (Figure 6.24a) and RadarTSR (Figure 6.24b).



(a) MacroPCA scree plot with the eigenvalues. **(b)** RadarTSR scree plot with the eigenvalues.

Figure 6.24: Scree plots with eigenvalues showing the number of principal components suggested by MacroPCA (left) and by RadarTSR (right) for the NIR dataset for the matrix with a 5% of MCAR missing entries.

Figure 6.25 shows how least-squares techniques (TSR and ICPCA) obtain lower $MSPE$ values than MacroPCA and RadarTSR. The $MSPE$ of RadarTSR overlaps with MacroPCA for most missing data percentages, except for the last one, which yields a significantly lower MSPE. Contrary to Figure 6.4, RadarTSR obtains a significantly higher $MSPE$ than ICPCA and TSR, although the order of magnitude of the $MSPE$ values makes such differences irrelevant in practical terms.

Moreover, the weighted sums of cosines overlap all techniques, and the rowwise specificity shows that rowwise and cellwise specificity are overlapped for MacroPCA and RadarTSR. However, MacroPCA yields more false positives in rowwise outliers detection on average than RadarTSR.

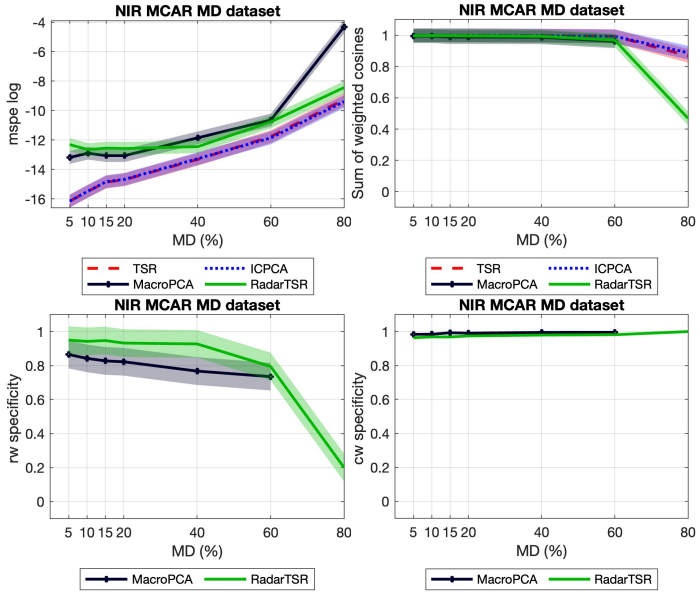


Figure 6.25: NIR spectra dataset. Average and LSD intervals for MSPE, the weighted sum of cosines, rowwise specificity, and cellwise specificity as a function of the percentage of missing cells.

MRI breast data

The original dataset, with the sequence of six frames of perfusion MRI, is shown in Figure 6.26 with the ROI shaded in pink. Since the most interesting outcome of this dataset concerns the detection of outliers, only the results of the outliers detection for MacroPCA and RadarTSR are shown for this case.

The scree plot analysis (Figure 6.27) led to the selection of two PCs ($A = 2$) for the PCA model. Figure 6.28 shows the loadings obtained by MacroPCA and RadarTSR, which are very similar.

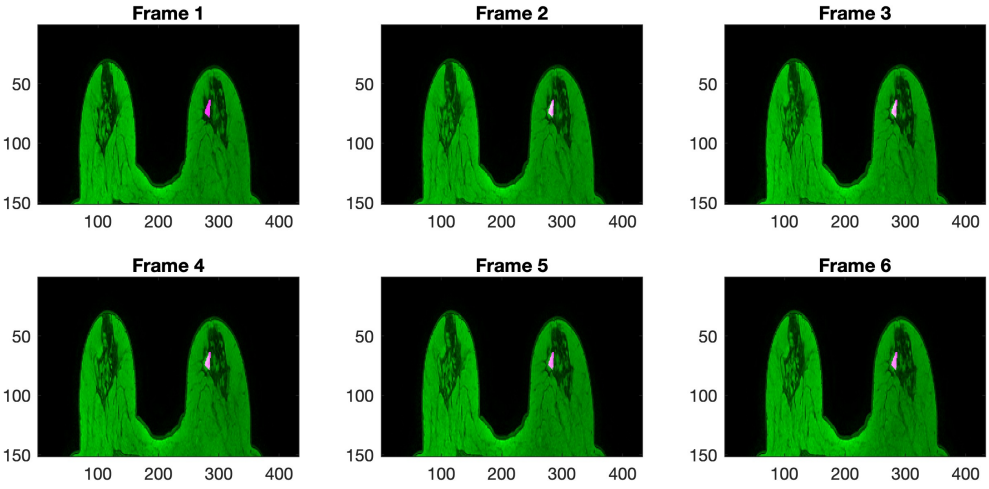
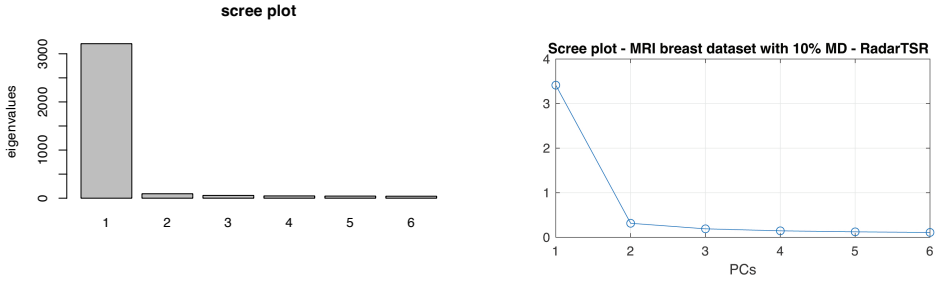


Figure 6.26: MRI breast dataset. The six frames contain the ROI with damaged pixels coloured in pink, healthy pixels coloured in green, and the background colour in black.



(a) MacroPCA scree plot with the eigenvalues. **(b)** RadarTSR scree plot with the eigenvalues.

Figure 6.27: MRI breast dataset. Scree plots with eigenvalues showing the number of principal components suggested by MacroPCA (left) and by RadarTSR (right) for the MRI breast dataset with 10% of MCAR missing entries.

The results in Figures 6.29 and 6.30 show once again that whereas MacroPCA might be more sensitive in detecting rowwise outliers, RadarTSR is more precise and specific. Moreover, the right plot in Figure 6.29 colours rowwise outliers differently depending on the cluster label assigned by RadarTSR, showing the pixels of the ROI belonging to the same cluster (coloured in red), whereas pixels from other regions are labelled to cluster 2 (coloured in blue).

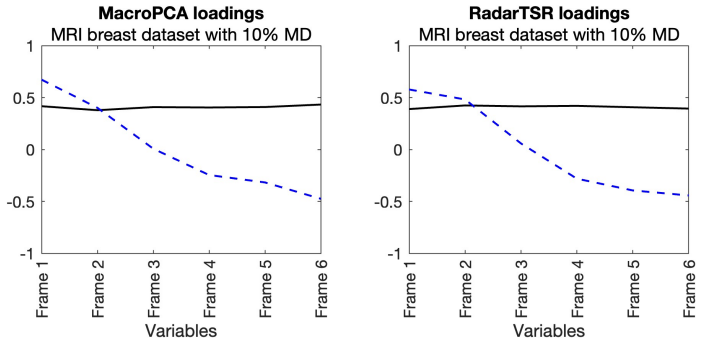


Figure 6.28: MRI breast dataset. Loadings obtained by the MacroPCA (left) and by the RadarTSR (right) algorithm. The loadings represent the first (black full line) and the second (blue dashed line) components.

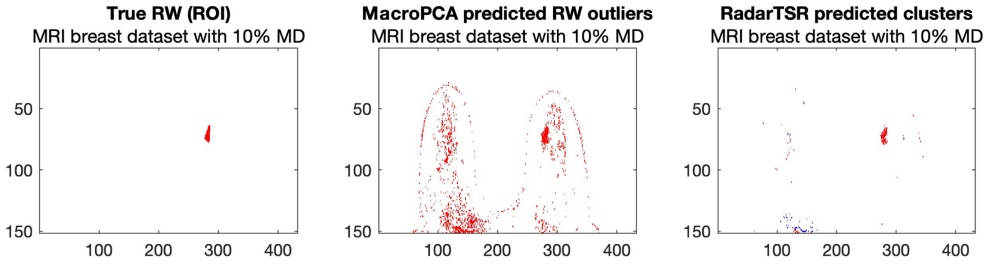


Figure 6.29: MRI breast dataset with 10% of MCAR missing data. Mask with the true ROI marking the outlying pixels (left), mask with the pixels detected as rowwise outliers by MacroPCA (centre), and image of the clusters assigned to the rowwise outliers detected by RadarTSR (right). The pictures correspond to three column vectors of $N = 23,193$ rows which have been reshaped to the original sizes of the images, with 151×432 pixels represented in the vertical and horizontal axes, respectively.

This observation provides additional insight into the handling of rowwise outliers and highlights the ability of RadarTSR to strike a balance between detecting false positives. Unlike MacroPCA, which does not offer this feature, the differences in loadings and outlier distances observed in RadarTSR contribute to a more comprehensive understanding of the presence and impact of rowwise outliers.

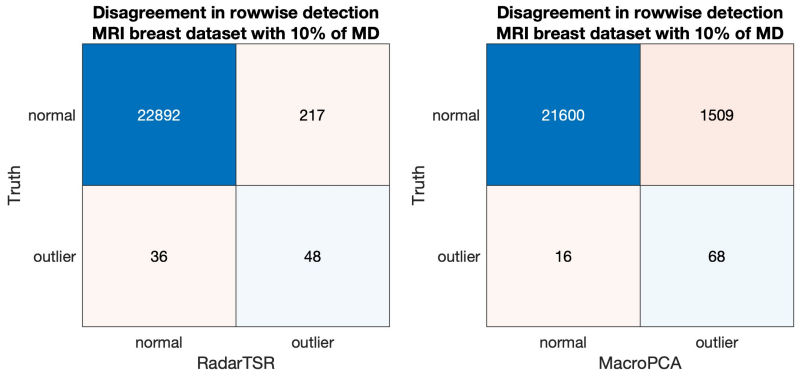


Figure 6.30: MRI breast dataset with 10% of MCAR missing data. Confusion matrix of the detection of rowwise outliers (pixels within the Region Of Interest) by RadarTSR and MacroPCA.

Figure 6.31 shows the distance from the model (left plots) and within the model (right plots) for each pixel, calculated using the metrics implemented by each algorithm. As can be seen, the SPE yielded by RadarTSR (upper left plot) highlights specifically the ROI. In contrast, MacroPCA's Orthogonal Distance (OD) gives a higher intensity to pixels spread over all the healthy tissue.

This is aligned with the higher rate of false positives obtained by MacroPCA (Figure 6.30). Besides, in terms of the distance within the model (right column of plots), the same phenomenon is present. However, both techniques highlight the contours of anatomical structures, indicating that pixels at the borders might emit higher signals but still fit the multivariate pattern of the rest of the healthy pixels.

Finally, Figure 6.32 shows the reconstruction error for each frame. As can be seen, pixels from the ROI are highlighted in most of the frames, with RadarTSR showing a higher precision to highlight pixels almost exclusively within the ROI. This result shows how checking the reconstruction error can complement the rowwise detection of RadarTSR, balancing the potential masking of rowwise outliers.

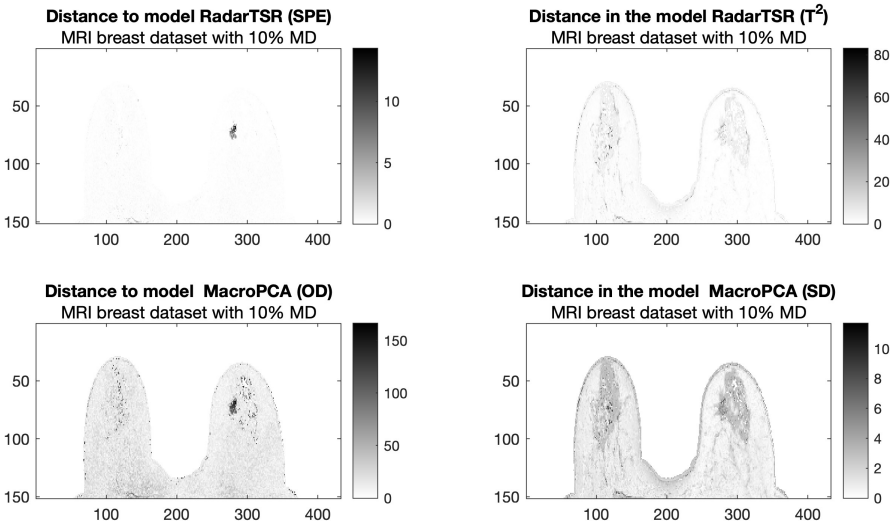
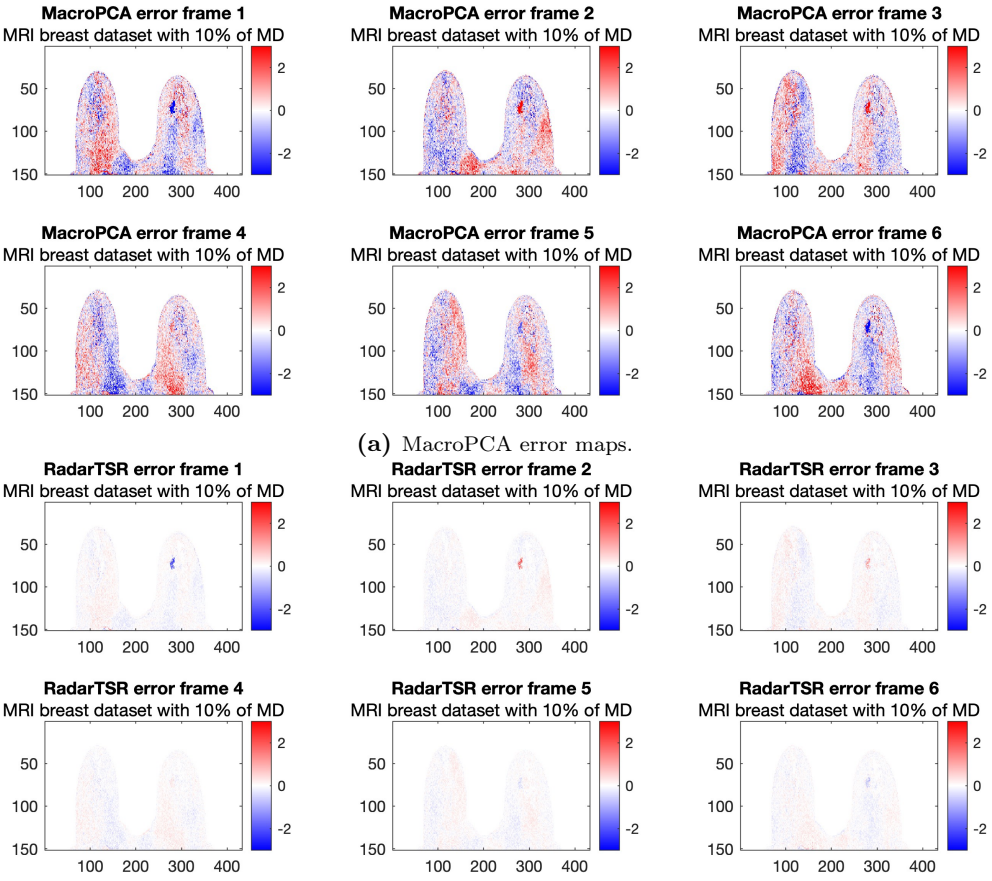


Figure 6.31: MRI breast dataset with 10% of MCAR missing data. Images showing the distances to the PCA model (left) and within the PCA model (right) obtained by RadarTSR (above) and MacroPCA (below) algorithms. For RadarTSR, the SPE and T^2 are used as distances, whereas the orthogonal and score distances (OD and SD, respectively) are used as their homologous metrics obtained by the MacroPCA model (see [48]).

Glass spectra data

Upon reviewing the scree plot, it was determined that a total of four principal components ($A = 4$) should be included in the PCA model (Figure 6.33), which was the same number of PCs selected by the authors for the analysis of the glass dataset without missing data [48].

Figure 6.34 shows the residual maps simulating 40% of MCAR missing data. The colours in these maps represent the average values of 5×5 cell blocks, enhancing visualization. ICPCA and TSR residual maps were not informative as their PCA models failed to account for outlier effects, resulting in outlying values not being evident in the maps. Conversely, MacroPCA and RadarTSR identified several observations (rows 143 to 180, 22 to 30, 53 to 63, and 74 to 76) as having a significant distance from the model (indicated by the black colour in the distance bar). Red cells along these rows indicate positive residuals, suggesting higher observed concentrations for specific compounds than the expected values from the reference PCA model.



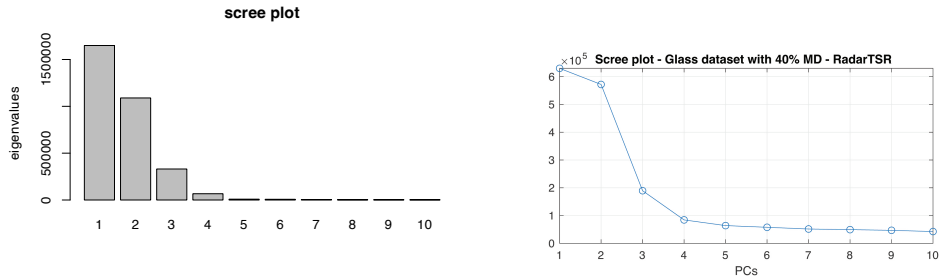
(a) MacroPCA error maps.

(b) RadarTSR error maps.

Figure 6.32: MRI breast dataset with 10% of MCAR missing data. Images with the normalized reconstruction error yielded by MacroPCA (above) and RadarTSR (below).

Figure 6.35a depicts the confusion matrix illustrating the disagreement in rowwise outlier detection between MacroPCA and RadarTSR. Despite some differences, both techniques showed substantial agreement, with MacroPCA identifying only nine additional glass samples as rowwise outliers.

Figure 6.36 showcases the score and distance plots of cell-imputed glass samples obtained through RadarTSR with the cluster tags assigned to detected rowwise outliers. Non-outlying observations are denoted as black circles with a tag "0". Glass samples from group 1 (rows 143 to 180) are represented as red triangles,



(a) MacroPCA scree plot with the eigenvalues. **(b)** RadarTSR scree plot with the eigenvalues.

Figure 6.33: Glass dataset with 40% of MCAR missing data. Scree plots with eigenvalues showing the number of principal components suggested by MacroPCA (left) and by RadarTSR (right) for the MRI glass dataset with 40% of MCAR missing entries.

exhibiting bad leverage data points indicated by a high T^2 and also high SPE. These samples were measured with a less efficient detector, suggesting they may belong to a different population, consistent with the results obtained by RadarTSR and MacroPCA.

Similarly, clusters 2 (rows 53 to 64 and 74 to 76) and 3 (rows 22 to 30) are depicted as blue squares and green diamonds, respectively. The detection of more than one cluster is consistent with previous studies of the dataset [48], [113] demonstrating both the successful detection of outlying sets and the prevention of their influence on the fitted PCA model by RadarTSR.

When the loadings obtained by the different methods are compared (Figure 6.37), there are noticeable differences between the methods. First, whereas MacroPCA, ICPCA, and TSR show the same positive peak for the first loading vector close to the 200th wavelength, RadarTSR does not.

Figure 6.38, which shows the loadings obtained by RadarTSR when the TSR for PCA-MB is applied on clusters "1" and "2", shows the peak in the 1st loading vector of cluster "1" loadings (left plot).

This difference suggests that this cluster could have influenced the PCA models of the rest of the techniques. Besides, the first PC (black-solid line in Figure 6.37) of MacroPCA and RadarTSR captures a negative weight for intermediate wavelengths, which is not captured by ICPCA nor by TSR. Finally, the second PC of RadarTSR and MacroPCA (blue dashed line in Figure 6.37) also captures a correlation pattern not found on ICPCA or TSR loadings.

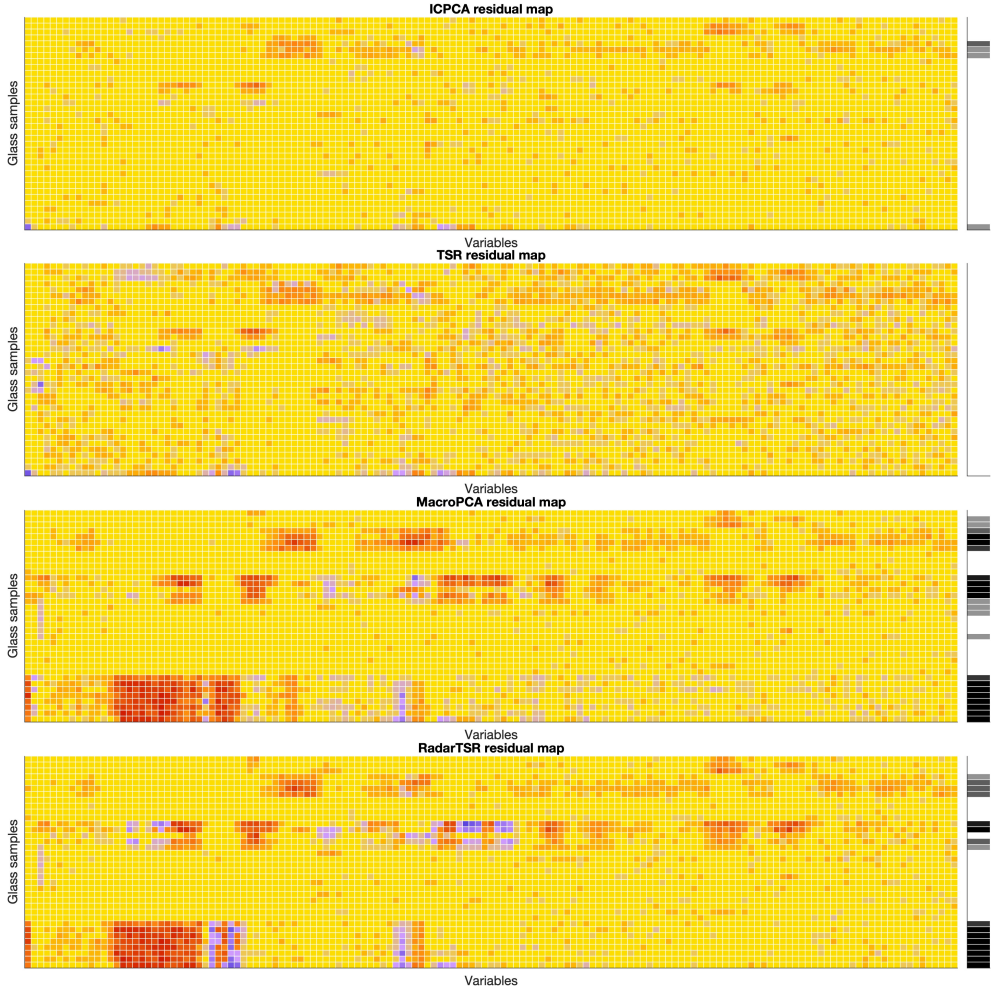
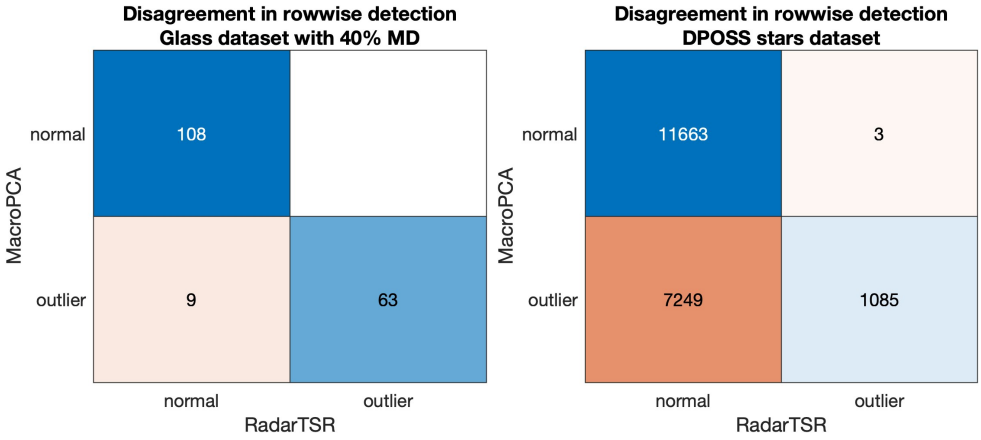


Figure 6.34: Glass dataset with 40% of MCAR missing data. Residual maps were obtained by ICPCA (first), TSR (second), MacroPCA (third), and RadarTSR (fourth).



(a) Confusion matrix for the Glass dataset. **(b)** Confusion matrix for the Glass DPOSS.

Figure 6.35: Confusion matrices of the detection of rowwise outliers by RadarTSR and MacroPCA for the Glass and the DPOSS stars datasets.

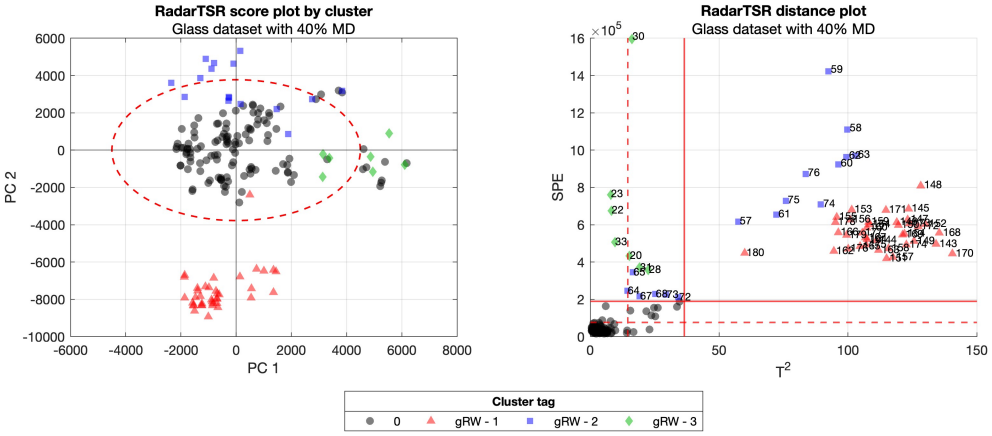


Figure 6.36: Glass dataset with 40% of MCAR missing data. The score plot (left) and distance plot (right) were obtained with RadarTSR. Dashed red lines are the UCLs at a 95% confidence level. Solid red lines represent the thresholds used for outliers detection (c_{rw}^{SPE} and $c_{rw}^{T^2}$, see Section 6.2.2).

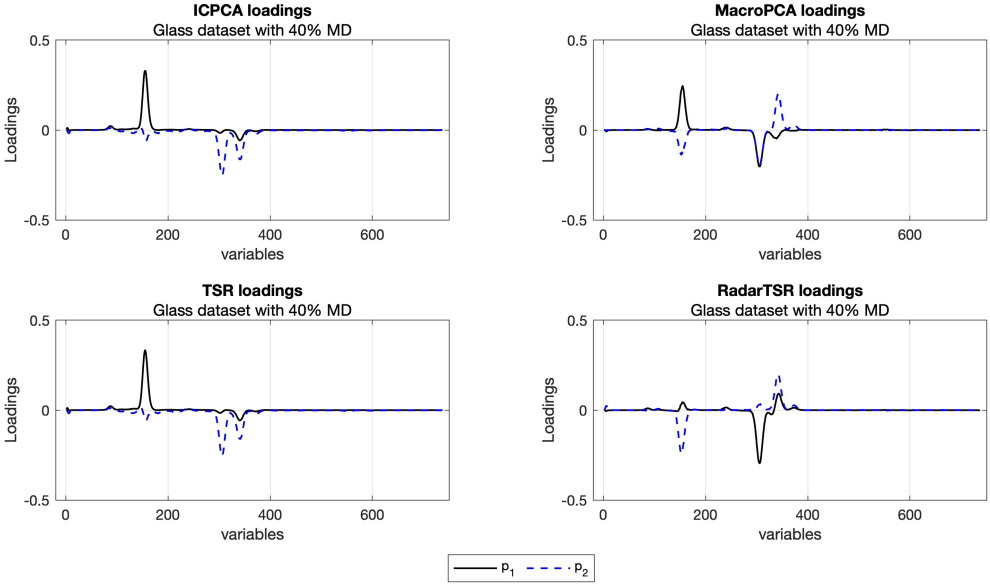


Figure 6.37: Glass dataset with 40% of MCAR missing data. Loading vectors of the first (solid line) and second (dashed line) PCs fitted with each one of the methods.

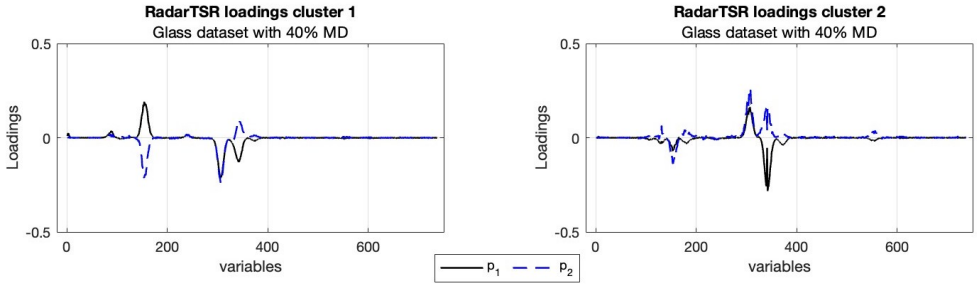


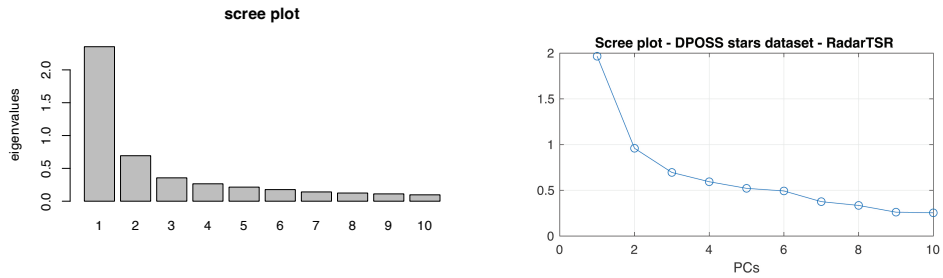
Figure 6.38: Glass dataset with 40% of MCAR missing data. Loading vectors of the first (solid line) and second (dashed line) PCs fitted for the two clusters detected by RadarTSR. The third cluster presented too many missing values to fit the PCA model using the same variables as the original model.

DPOSS stars data

The last real dataset included was the DPOSS stars dataset [48], [115]. Seven variables were recorded at each colour band, resulting in 21 variables denoted

by the variables' names followed by the name of the colour band used for the measurement (J, F, or N). Three variables measure light intensity ("MAper", "MTot", and "MCore"), whereas the rest ("Area", "IR2", "csf", and "Ellip") report the size and shape of the objects.

The scree plot of both methods (Figure 6.39) showed a stabilisation on the eigenvalues decay for the 4th PC, and therefore we set $A = 4$ for both methods. This was also the same number of PCs selected by the authors for the analysis of the DPOSS stars dataset in [48].



(a) MacroPCA scree plot with the eigenvalues. **(b)** RadarTSR scree plot with the eigenvalues.

Figure 6.39: DPOSS stars dataset. Scree plots with eigenvalues showing the number of principal components suggested by MacroPCA (left) and by RadarTSR (right) for the MRI DPOSS stars dataset.

The upper plots depicted in Figure 6.40 illustrate the score plots of Macro-PCA and RadarTSR. In these plots, observations with the lowest and highest outlier distances (OD) based on MacroPCA are represented by black and red dots, respectively. Overall, the loading plots generated by both methods exhibit consistent signs for the first principal component (PC). However, there are notable disparities in the magnitudes of the loadings between the two techniques.

A previous report on this dataset emphasized the significance of variables "MTot", "MCore", "Area", and "IR2" in distinguishing between galaxies and stars [115]. This finding aligns with the loadings captured by the first PC of RadarTSR, indicating that objects with high positive values on this PC are large objects emitting less intense light signals.

Conversely, observations with negative values on the first PC correspond to dense objects with small sizes but emitting more intense light signals. In contrast, the loadings obtained from MacroPCA analysis on the first principal component (PC) indicate that the variable "Ellip" is the most influential one,

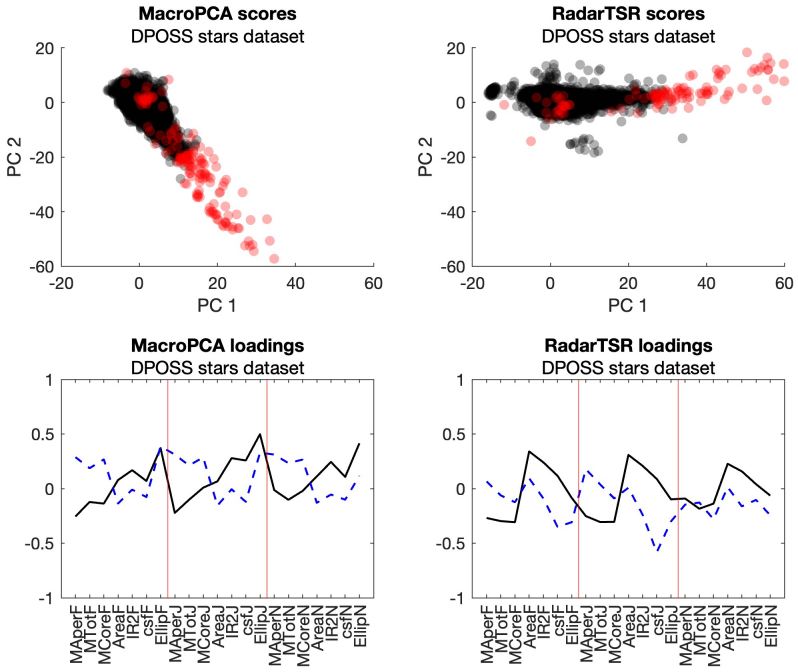


Figure 6.40: DPOSS stars dataset. Scores and loadings were obtained by the MacroPCA (left) and by the RdarTSR (right) algorithm. Black and red dots represent the observations with the lowest and highest OD, respectively, according to MacroPCA. The loadings are shown only of the first (black full line) and the second (blue dashed line) components, with vertical red lines separating the three colour bands.

and it displays a positive correlation with the remaining variables associated with the shape and size of objects. However, when examining the loadings on the second PC, it becomes apparent that "Ellip" is positively correlated with light intensity variables rather than size and shape variables, thereby contradicting the correlation pattern observed in the loadings of the first PC.

Regarding the second PC of RadarTSR, the most relevant variables are "csf" and "Ellip", while the remaining variables' loadings oscillate around zero. Additionally, "csf" presents differences across colour bands, with band J having the most significant loadings, followed by band F, and finally, band N, whose variables are almost irrelevant. This colour-based separation of celestial objects and the hierarchy of variable importance across colour bands are consistent

with previous findings on the DPOSS dataset, which reported a high signal-to-noise ratio for variables from the N colour band [115].

The residual map of RadarTSR is presented in the left plot of Figure 6.41. Similarly to the residual map of MacroPCA (right plot), each row block represents 25 stars, with the top six rows corresponding to the 150 stars with the highest OD according to MacroPCA. In general rowwise outliers typically demonstrate elevated values (indicated by red residuals) of light intensity, size, and shape. This observation implies that these celestial entities might correspond to giant stars, aligning with conclusions drawn from prior investigations conducted by [48].

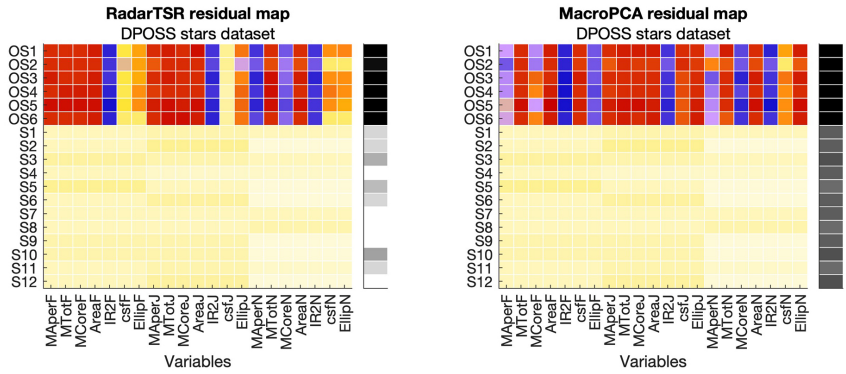


Figure 6.41: DPOSS stars dataset. Residual maps for RadarTSR (left) and MacroPCA show the groups of celestial objects with higher OD at the top of the map and the groups with the lowest OD at the bottom.

The score plot on the left side of Figure 6.42 indicates that RadarTSR suggests one cluster that groups most of the detected rowwise outliers. The loadings presented in Figure 6.42 were obtained when applying TSR for PCA-MB to all rows tagged as "1". While "MAper", "MTot", and "MCore" exhibit negative values for all colour bands, similar to Figure 16, "IR2" and "csf" are less relevant compared to "Area" in these loadings. The variable "Area" represents the total space covered by a celestial object, and "IR2" measures the intensity-weighted second spatial moment of the pixels within an image, indicating the dispersion of objects from their centre. Loadings for the second PC assign positive values to intensity-related variables. Moreover, there is considerable homogeneity in loading values across the three colour bands.

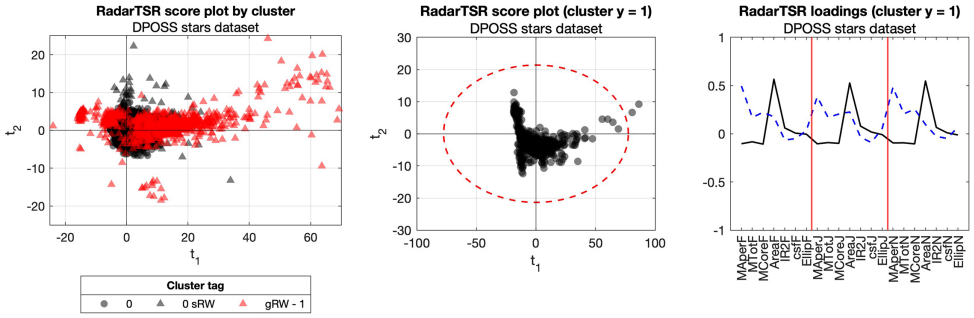


Figure 6.42: DPOSS stars dataset. Score plot (left) showing the clusters of celestial objects suggested by RadarTSR, with non-outlying observations as black circles, single rowwise outliers as black triangles, and grouped rowwise outliers as ref triangles. The centre and right plots show the scores and loadings of the PCA model fitted running TSR for PCA-MB on the observations from cluster "1". The loadings are shown only for the first (black full line) and the second (blue dashed line) components, with vertical red lines separating the three colour bands.

This correlation structure suggests that objects within cluster 1 are homogeneous light sources ("IR2" loadings close to zero) but vary in size and luminosity. Furthermore, observations in the left upper quadrant of the score plot indicate small and very luminous objects. This result suggests that the clustering stage of RadarTSR could potentially improve through the utilization of more robust techniques, enhancing the discrimination of minority clusters within the set of rowwise outliers.

6.6 Conclusions

The RadarTSR algorithm can handle cellwise outliers, rowwise outliers, and missing data, including a hypothetical cluster of rowwise outliers. Simulations showed that its performance was similar in terms of $MSPE$ to MacroPCA's, the state-of-the-art method to deal with cellwise outliers, rowwise outliers, and missing data. RadarTSR always obtained comparable results to MacroPCA in the presence of outliers, even superior in some scenarios regarding PCA model similarity. In the absence of outliers, RadarTSR performed more similarly to least squares methods, making RadarTSR preferable in most scenarios. Moreover, even if the PCA model provided by RadarTSR is not the modelling technique of interest for a given problem, a good imputation of missing data will

positively affect the outcomes obtained by other supervised or non-supervised models fitted on the imputed dataset.

Simulations also showed an antagonistic performance of MacroPCA and RadarTSR: MacroPCA tends to over-detect rowwise outliers, and RadarTSR treats mild rowwise outliers as rows with cellwise outliers, correcting and ultimately masking them. The main consequence of MacroPCA's low precision for rowwise outliers is the loss of observations as part of the reference set. Moreover, this could also be a limitation for high-precision tasks, as with the *MRI breast* dataset, where anomalous pixels would be considered potential tumoral tissue. On the other hand, the main consequence of the masking effect produced by RadarTSR would be the opposite: to include such rowwise outlying observations as part of the reference set. Nevertheless, users could check the RadarTSR residual maps to address this issue, assessing if outlying cells might correspond to a rowwise contamination paradigm. An example of this can also be seen for the *MRI breast* dataset, where the reconstruction errors showed higher error values in magnitude for pixels within the ROI that RadarTSR could have missed. Moreover, the rowwise outliers' masking artefact of RadarTSR is less likely as the number of columns (K) increases, i.e., for high-dimensional datasets, which are the most frequent case in many scientific areas where automated data collection techniques are applied, such as chemometrics.

Besides, RadarTSR was the only technique including a further assessment of the detected rowwise outliers by including a clustering step. This feature was tested with the *MRI breast* dataset, the *Glass spectra* dataset, and the *DPOSS stars* dataset, and the clusters detected by RadarTSR aligned with the subpopulations either known *a priori* or described in previous works [48], [113], [115].

Despite its strengths, the RadarTSR algorithm also has some limitations that should be acknowledged, paving the way for further research and expansions of the method. The first limitation derives from RadarTSR's heuristic nature, which limited the assessment of its general properties to an empirical evaluation comparing its performance to other methods in this study. While this empirical evaluation provides valuable insights, a more formal mathematical analysis of RadarTSR's properties could further enhance its understanding and contribute to its theoretical foundations.

The second limitation concerns the clustering step in RadarTSR, which may benefit from further refinement. In some cases, such as the long dataset with grouped rowwise outliers and the *DPOSS stars* dataset, there is a possibility that clusters could remain undetected or become mixed with larger sets. To

address this limitation, incorporating more robust clustering techniques, such as the K-medoids algorithm, known for its resilience to noise and outliers [117], could enhance the clustering performance of RadarTSR.

The third limitation concerns the applicability of RadarTSR in some scenarios. For instance, RadarTSR assumes ignorable mechanisms producing missing data and the presence of continuous variables. While these limitations are shared by the other methods included in the comparison, they restrict the applicability of RadarTSR to datasets where the missing data mechanism meets the ignorable assumption, and adaptations would be necessary to handle discrete and categorical data. Moreover, the presented work contemplated so far using RadarTSR for PCA-MB. Hence, expanding RadarTSR to the Model Exploitation (ME) scenario is a straightforward future line of research. Nonetheless, this step deserves careful consideration, as it includes nontrivial aspects such as recognizing new clusters of grouped outliers that might not present during the MB stage.

The algorithm was programmed in Matlab. The code and the documentation are available on GitHub. Further improvements to improve its accessibility include programming the algorithm in open code languages like R or Python.

6.A Appendix: Notation

This appendix contains further information about the notation used throughout the chapter.

Table 6.3: Elements of the generic PCA model.

$\boldsymbol{X}$	original matrix
$\boldsymbol{Z}$	standardized matrix
$\hat{\boldsymbol{\mu}}$	location estimator
$\hat{\boldsymbol{\sigma}}$	scale estimator
$\boldsymbol{P}$	loadings of $\boldsymbol{X}$
$\boldsymbol{T}$	scores of $\boldsymbol{X}$ according to $\widehat{\boldsymbol{\mu}}$ and $\boldsymbol{P}$
$\boldsymbol{\Theta}$	covariance matrix of the latent variables
$\boldsymbol{\lambda}$	variances of the scores sorted decreasingly
$\boldsymbol{E}$	residual matrix
$\boldsymbol{R}$	standardized residual matrix
$\hat{\boldsymbol{X}}$	reconstruction of $\boldsymbol{X}$ using a PCA model
SPE	Squared Prediction Error of an observation
T^2	Hotelling's T^2 of an observation
$UCL(SPE)_\alpha$	Upper Control Limit for the SPE assuming a type I risk of α
$UCL(T^2)_\alpha$	Upper Control Limit for the Hotelling's T^2 assuming a type I risk of α

Table 6.4: Notation used for elements of the RadarTSR algorithm.

$\mathring{\boldsymbol{X}}$	NA-imputed matrix with only missing entries of non-outlying rows imputed.
$\dot{\boldsymbol{X}}$	cell-imputed matrix with cellwise outliers imputed for non-outlying rows and missing entries imputed for all rows, no matter the outlyingness.
$\tilde{\boldsymbol{X}}$	cluster-imputed matrix with cellwise outliers imputed for non-outlying rows and missing entries imputed for all rows, using the PCA model corresponding to the cluster tag assigned to each observation.
$\mathring{M}$	logical matrix indicating missing cells
$z_{1-\alpha/2}$	percentile of the normal distribution
c_{cw}	cut-off to detect cellwise outliers
$\dot{\boldsymbol{M}}$	logical matrix indicating outlying cells

$\boldsymbol{p}_n^{cw}$	vector with the proportion of cellwise outliers of each row, obtained from $\dot{\boldsymbol{M}}$
c_{rw}^{pre}	cut-off to detect potential rowwise outliers in Step 2 based on $\dot{\boldsymbol{M}}$
$c_{rw}^{p^{cw}}$	cut-off to detect rowwise outliers based on the proportion of cellwise outliers of each row
c_{rw}^{SPE}	cut-off to detect moderate outliers
$c_{rw}^{T^2}$	cut-off to detect extreme outliers
$\boldsymbol{m}$	logical vector indicating outlying rows, based on $S\dot{P}E$, $\dot{T}^2$, c_{rw}^{SPE} and $c_{rw}^{T^2}$
$\boldsymbol{m}^{(0)}$	logical vector indicating outlying rows at the beginning of the iterative reference model estimation, for the first iteration ($s = 0$), based on c_{rw}^{pre}
$\boldsymbol{m}^{(s)}$	logical vector indicating outlying rows at the end of iteration s of the Step 3, based on $S\dot{P}E^{(s)}$, $\dot{T}^{2(s)}$, $c_{rw}^{SPE(s)}$ and $c_{rw}^{T^2(s)}$
N'	number of observations not considered as rowwise outliers
c_{n_y}	threshold on the minimum number of observations necessary to be considered as a cluster
C_y	number of clusters among rowwise outliers
$\{\hat{\boldsymbol{\mu}}, \boldsymbol{P}, \boldsymbol{\lambda}\}^{(y)}$	elements of the PCA model built using observations from cluster y
$\{\hat{\boldsymbol{\mu}}, \boldsymbol{P}, \boldsymbol{\lambda}\}^{(\dot{\boldsymbol{E}})}$	elements of the PCA model built on the residual matrix $\dot{\boldsymbol{E}}$
N''	number of observations considered as rowwise outliers
A''	number of PCs of the PCA model fitted on the residual matrix of the detected rowwise outliers
$\boldsymbol{T}^{(\dot{\boldsymbol{E}})}$	matrix of scores obtained by projecting the residual matrix $\dot{\boldsymbol{E}}$ on its PCA model
$\boldsymbol{y}$	vector with the cluster tag assigned to each observation
$\hat{\boldsymbol{\mu}}$	vector with the cluster mean
$\boldsymbol{C}$	matrix with all mean vectors $\hat{\boldsymbol{\mu}}_{A''}$ for each cluster

Table 6.5: Notation used for the comparative study.

$\boldsymbol{X}^{clean}$	clean matrix without missing data and outliers
$\boldsymbol{X}^{method}$	imputed matrix yielded by each method
$\boldsymbol{X}^{clean}$	clean matrix without missing data, outliers, and without outlying rows from $\mathring{\boldsymbol{X}}^{method}$, used to build the clean PCA model
$\boldsymbol{X}^{method}$	imputed matrix yielded by each method without outlying rows

$\hat{\boldsymbol{X}}^{clean}$	reconstructed clean matrix by using a PCA model fitted on $\boldsymbol{X}^{clean}$
$\hat{\boldsymbol{X}}^{method}$	reconstructed imputed matrix by using the PCA model fitted by each method
$\boldsymbol{P}^{clean}$	loading matrix obtained after fitting a PCA model on the matrix $\boldsymbol{X}^{clean}$
$\boldsymbol{P}^{method}$	loading matrix yielded by each method
$\boldsymbol{X}^{method}$	imputed matrix yielded by each method
$\dot{\boldsymbol{R}}$	normalized residual matrix obtained as the difference between the original matrix and the imputed matrix yielded by each method
$\boldsymbol{d}^{model}$	distance metric computed as the scaled distance to the model for each one of the techniques, using its distance (SPE for RadarTSR and TSR and OD for MacroPCA and ICPCA) and scaling it by the threshold for such distance computed by each algorithm

6.B Appendix: Results for the time used for the simulated datasets

This appendix contains the results for the time employed by each algorithm to obtain the results of the simulated datasets shown in Section 6.5.

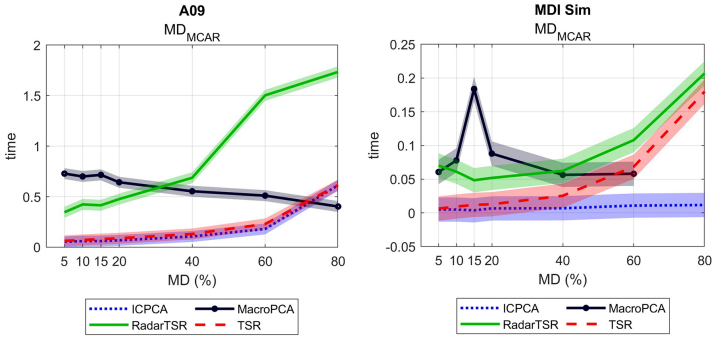


Figure 6.43: Missing data case results. The left plot shows the execution time in seconds for the 50 repetitions of the simulations for the wide dataset and the right plot for the long dataset. The x-axis of each plot denotes the MD percentage. The dotted, circles, dashed, and solid lines denote the results of ICPCA, MacroPCA, TSR, and RadarTSR, respectively. The shaded bars represent the 95% LSD confidence intervals of the metrics obtained by each method.

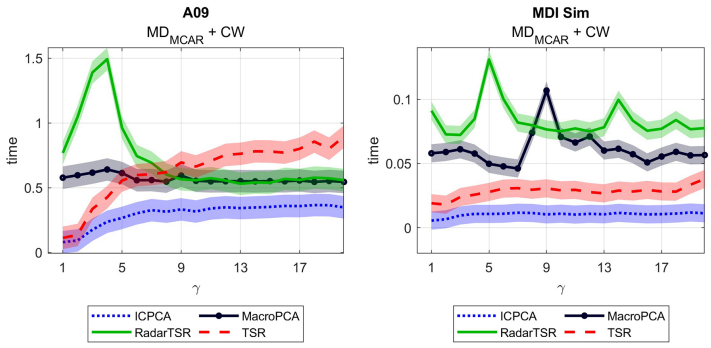


Figure 6.44: Missing data (20%) and cellwise outliers (20%) case results. The x-axis of each plot denotes the outliers' distance, γ. More details are in the caption of Figure 6.43.

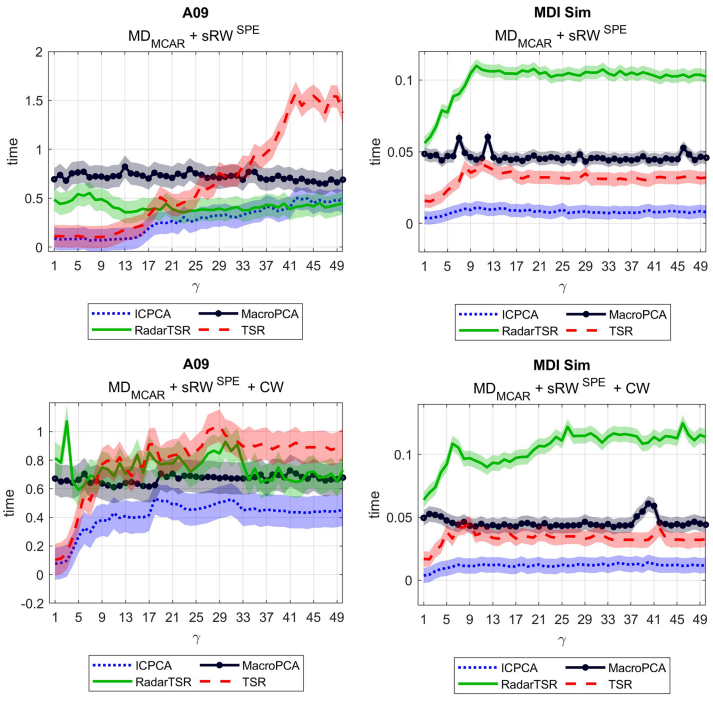


Figure 6.45: Missing data (20%) with single SPE rowwise outliers (20%) in the upper row, and with single SPE rowwise outliers (10%) and cellwise outliers (10%) in the lower row. The x-axis of each plot denotes the outliers' distance, γ. More details are in the caption of Figure 6.43.

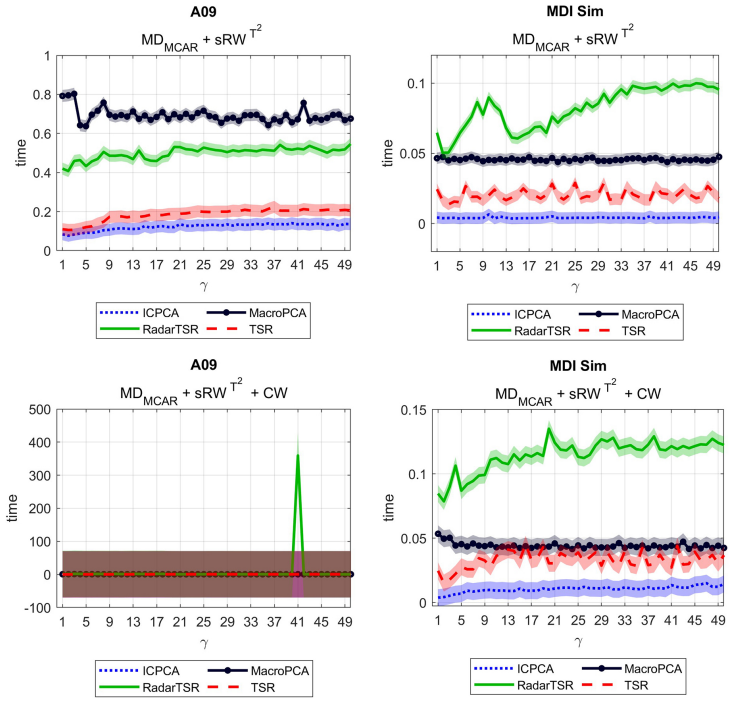


Figure 6.46: Missing data (20%) with single T^2 rowwise outliers (20%) in the upper row, and with single T^2 rowwise outliers (10%) and cellwise outliers (10%) in the lower row. The x-axis of each plot denotes the outliers' distance, γ. More details are in the caption of Figure 6.43.

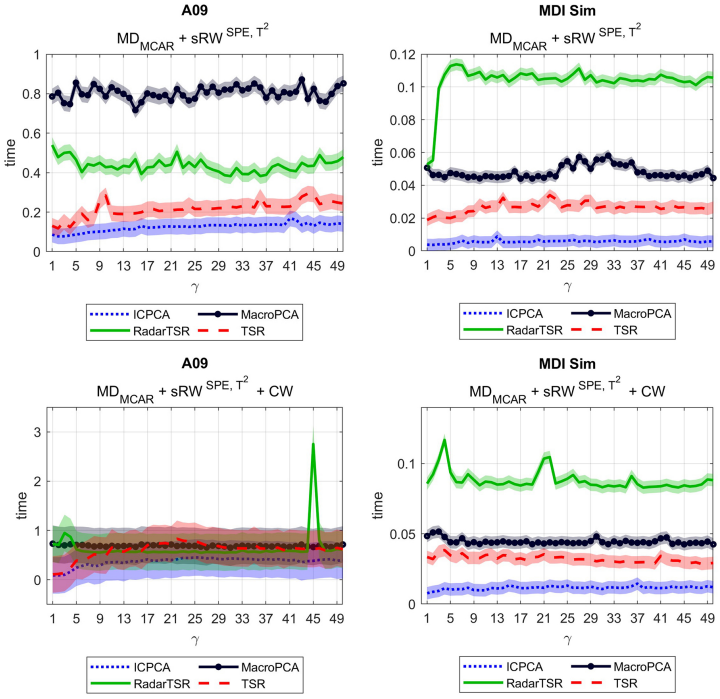


Figure 6.47: Missing data (20%) with single SPE and T^2 rowwise outliers (20%) in the upper row, and with single SPE and T^2 rowwise outliers (10%) and cellwise outliers (10%) in the lower row. The x-axis of each plot denotes the outliers' distance, γ. More details are in the caption of Figure 6.43.

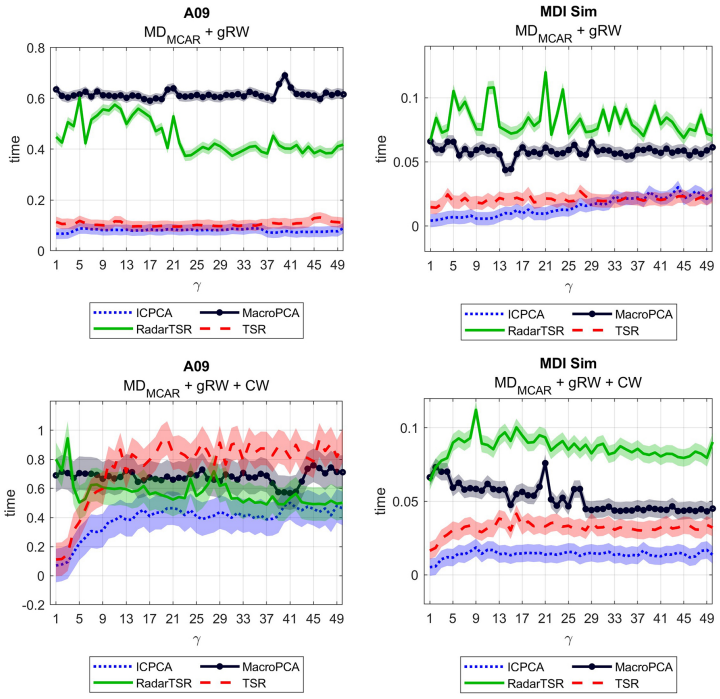


Figure 6.48: Missing data (20%) with grouped rowwise outliers (20%) in the upper row, and with single grouped rowwise outliers (10%) and cellwise outliers (10%) in the lower row. The x-axis of each plot denotes the outliers' distance, γ. More details are in the caption of Figure 6.43.

6.C Appendix: Assessment of the number of clusters for real datasets

This appendix contains the intermediate outputs offered by the RadarTSR algorithm and used to assess the number of clusters after running a PCA on the residual matrix yielded by *MRI breast*, *Glass* and *DPOSS stars* datasets from Section 6.5.2. In this section, the parameter K will refer to the number of clusters (as in, K-means) but not to the dimensionality of the dataset. In the rest of the chapter, the number of clusters is called C.

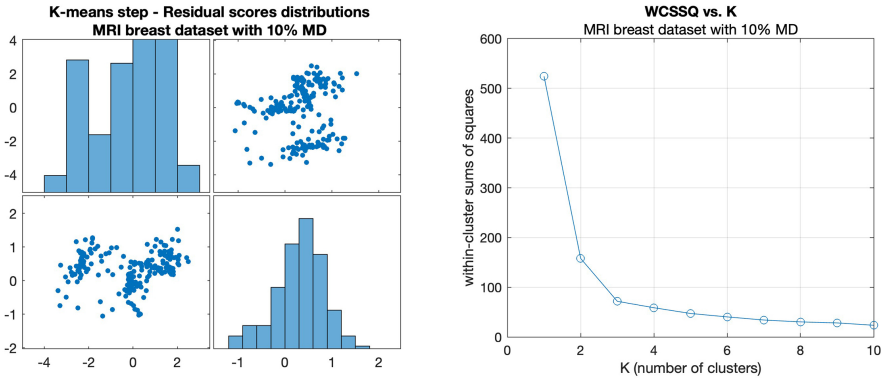


Figure 6.49: Distributions of the scores from the PCA on the residual matrix (left) and within-clusters sums of squares for each number of clusters, K (right) for the MRI breast dataset with 10% of simulated MCAR missing data.

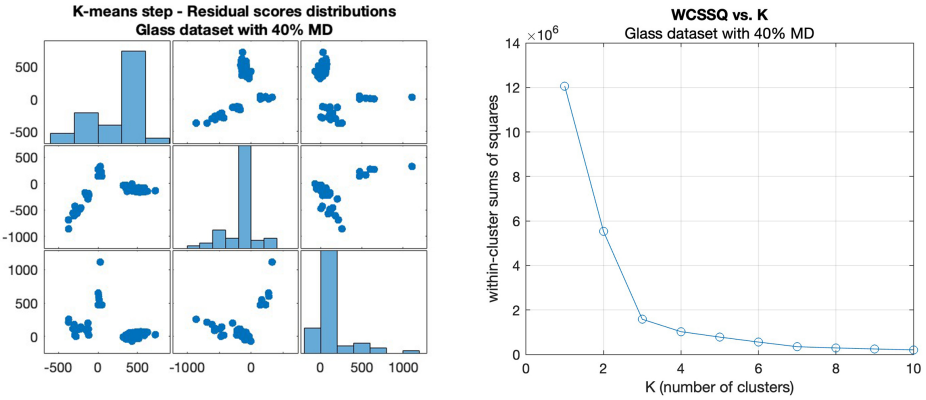


Figure 6.50: Distributions of the scores from the PCA on the residual matrix (left) and within-clusters sums of squares for each number of clusters, K (right) for the Glass dataset with 40% of simulated MCAR missing data.

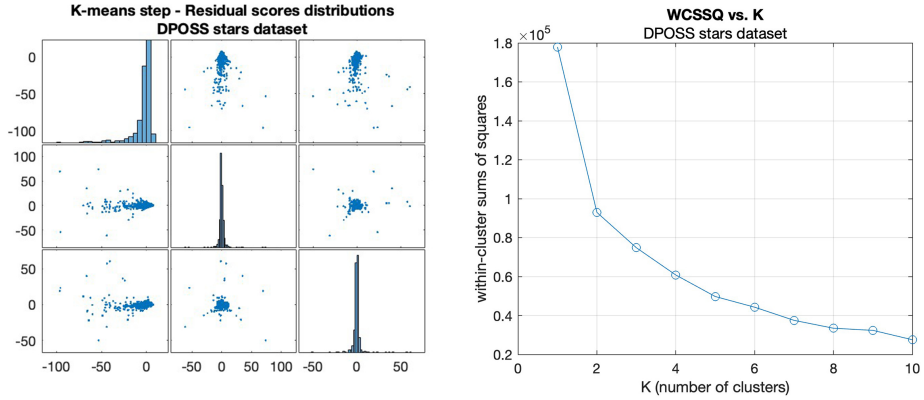


Figure 6.51: Distributions of the scores from the PCA on the residual matrix (left) and within-clusters sums of squares for each number of clusters, K (right) for the DPOSS stars dataset.

6.D Appendix: Results simulating MAR missing data

This appendix includes the results obtained for both simulated data sets when MAR missing data were simulated. The MAR pattern was simulated as explained in Section 6.3.1. In all the following figures, the first row of plots shows the MSPE results; the second shows the weighted sum of cosines between loadings, and the third and fourth rows show the detection metrics for rowwise and

cellwise outliers, respectively. The x-axis of each plot denotes the outliers' distance γ. The first column shows the results for the wide dataset, and the second row for the long dataset. The dotted, circles, dashed, and solid lines denote the results of ICPCA, MacroPCA, TSR, and RadarTSR, respectively. The shaded bars represent the 95% LSD confidence intervals of the metrics obtained by each method.

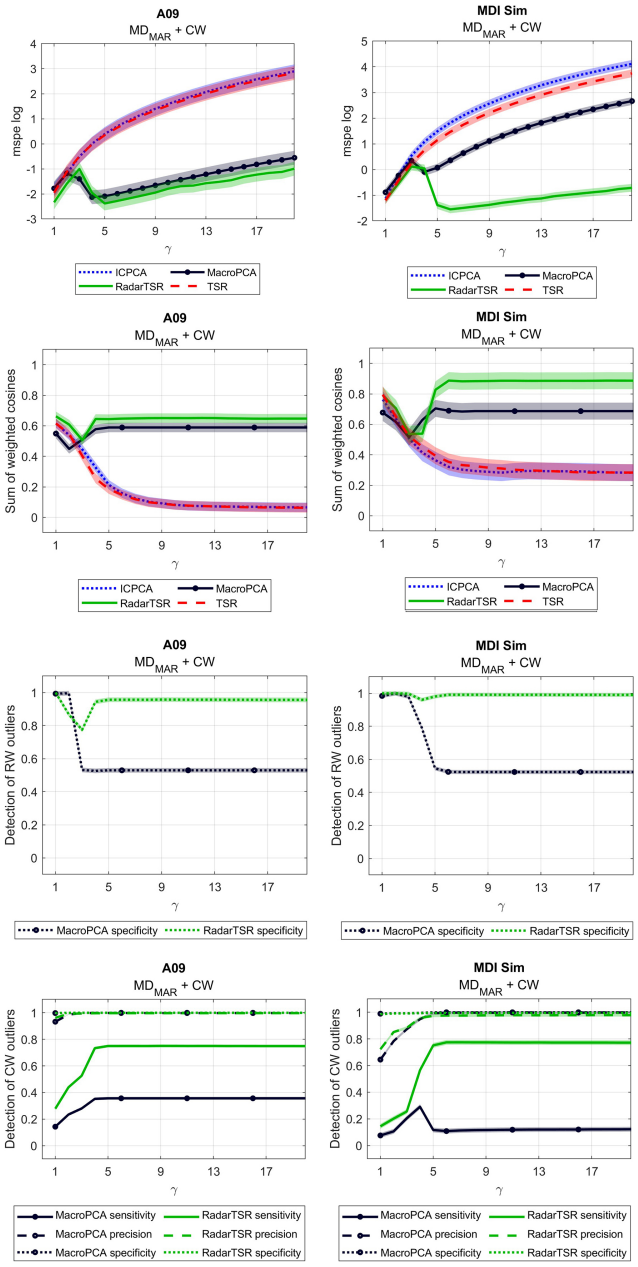


Figure 6.52: Missing data MAR (20%) and cellwise outliers (20%) case. More details are in the introduction of the Appendix.

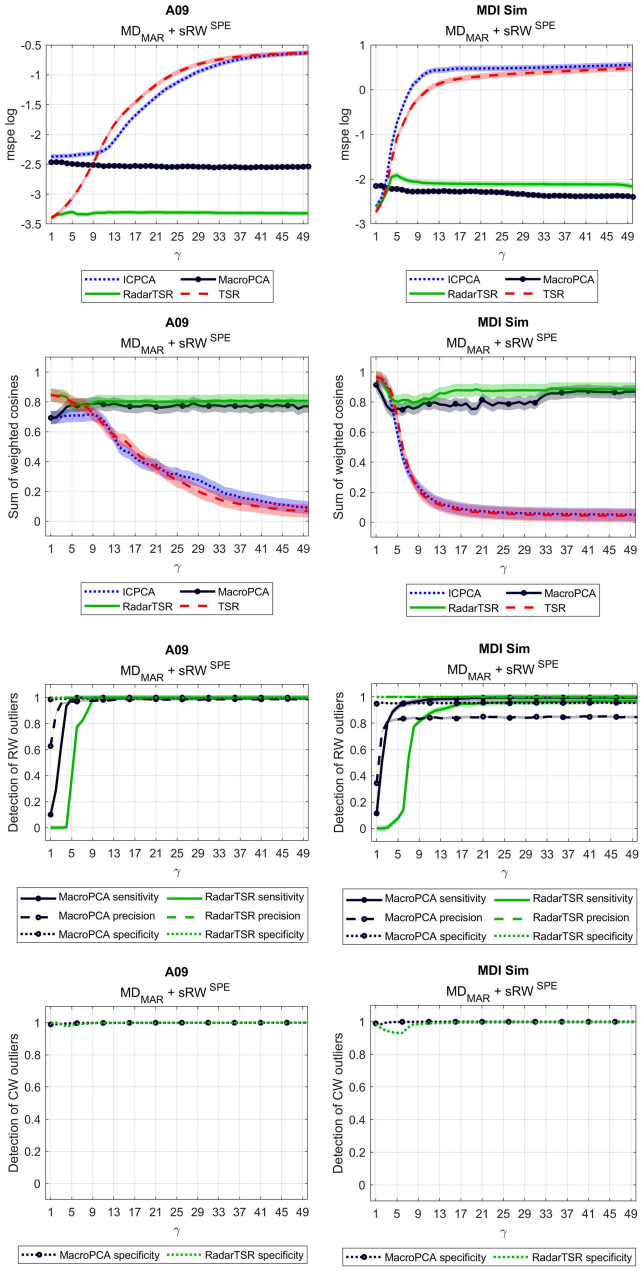


Figure 6.53: Missing data MAR (20%) and single SPE rowwise outliers (20%) case. More details are in the introduction of the Appendix.

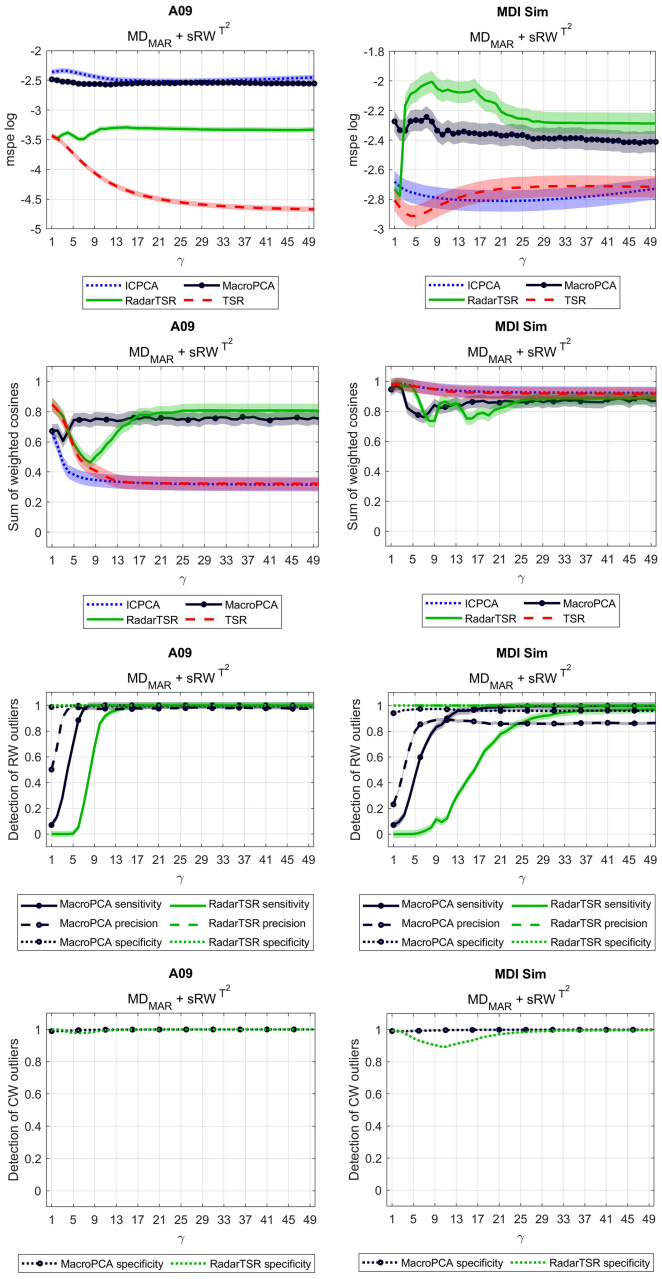


Figure 6.54: Missing data MAR (20%) and single T^2 rowwise outliers (20%) case. More details are in the introduction of the Appendix.

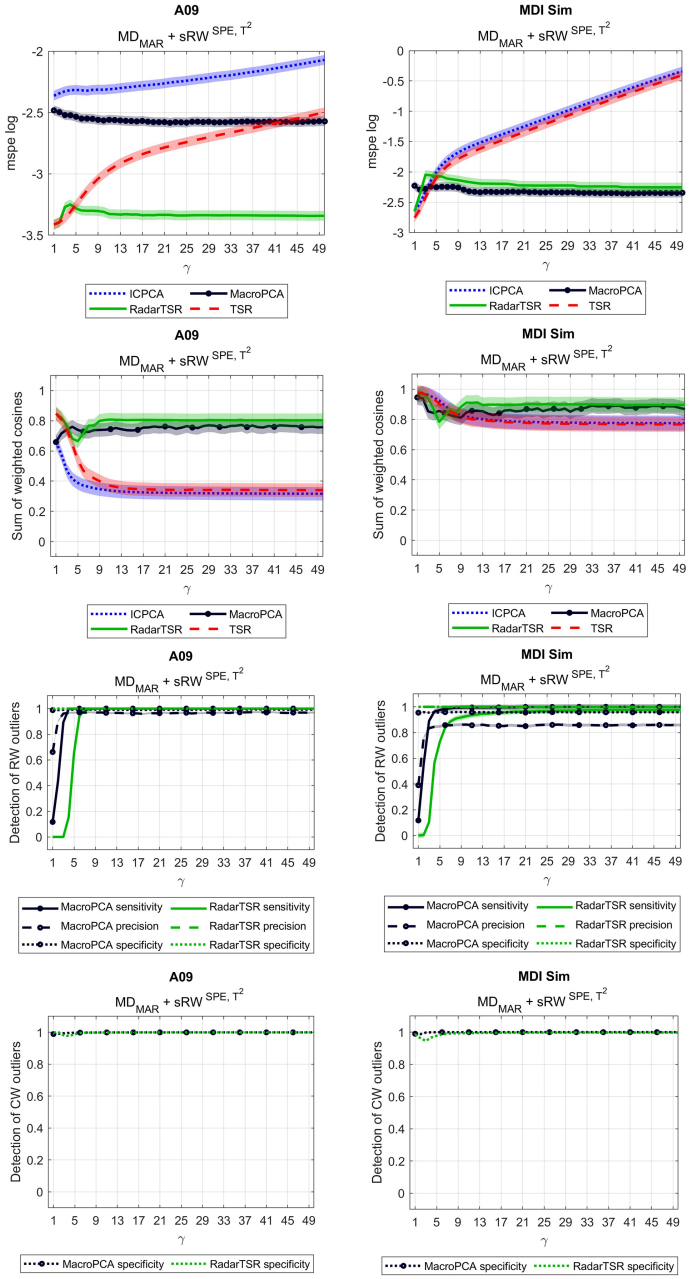


Figure 6.55: Missing data MAR (20%) and single SPE and T^2 rowwise outliers (20%) case. More details are in the introduction of the Appendix.

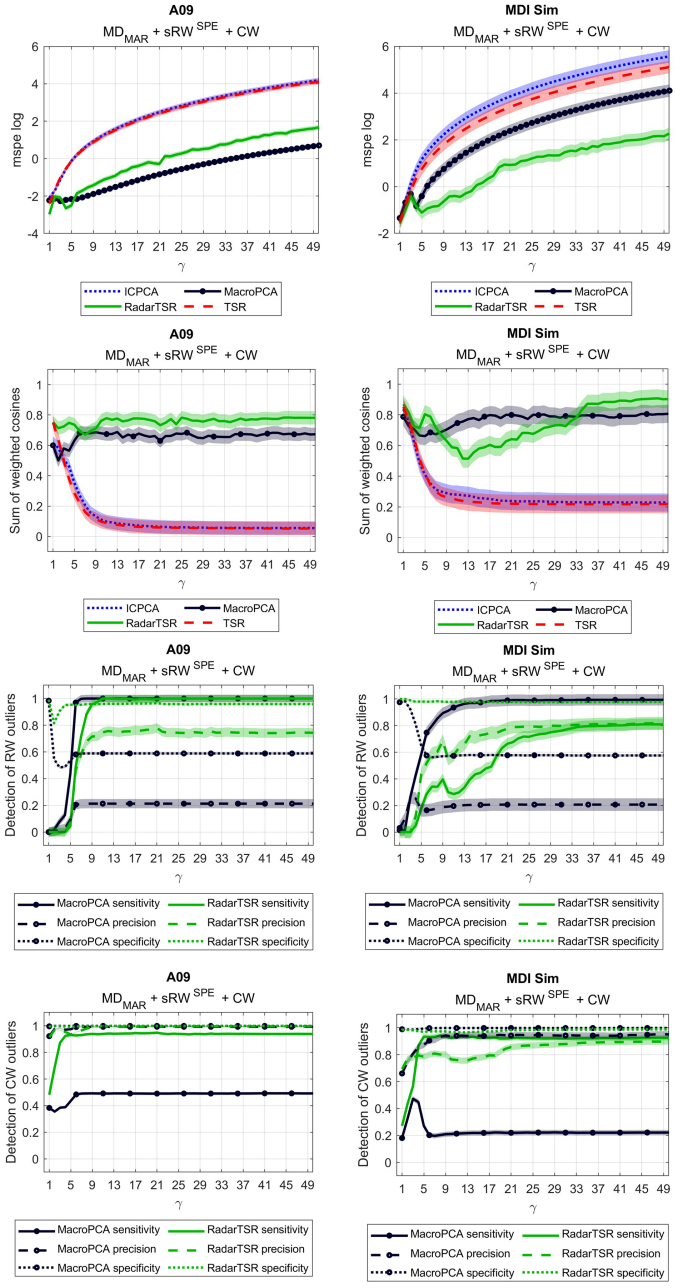


Figure 6.56: Missing data MAR (20%), single *SPE* rowwise outliers (10%) and cellwise outliers (10%) case. More details are in the introduction of the Appendix.

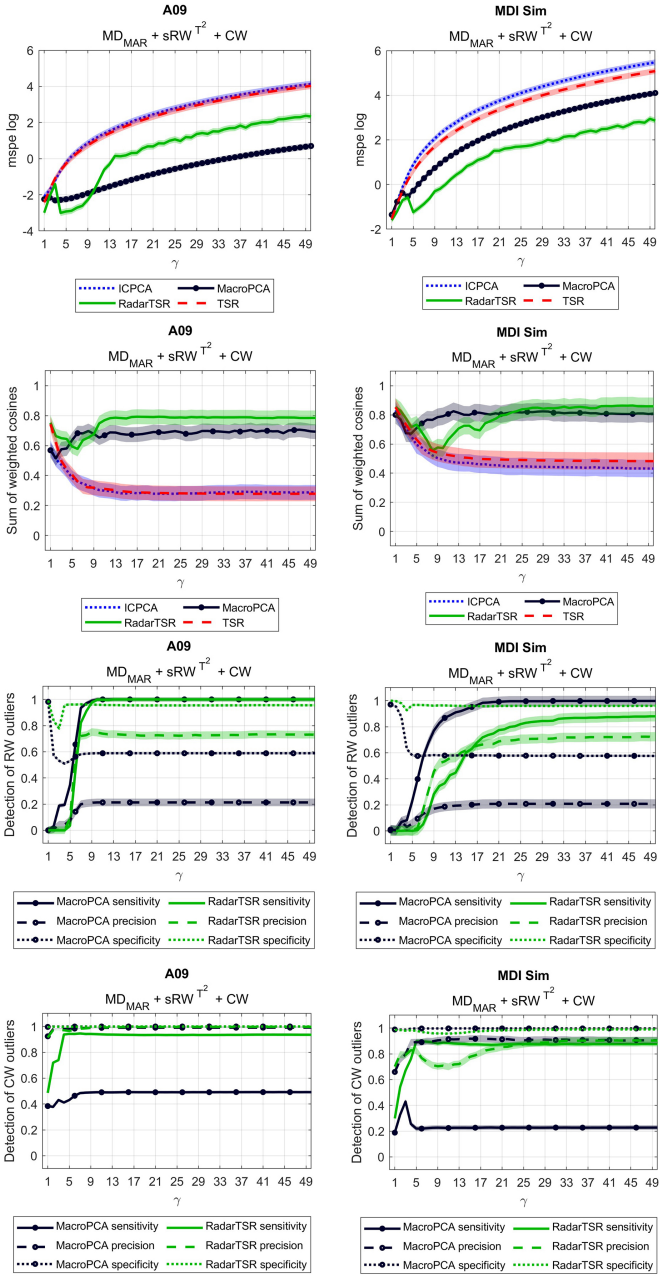


Figure 6.57: Missing data MAR (20%), single T^2 rowwise outliers (10%) and cellwise outliers (10%) case. More details are in the introduction of the Appendix.

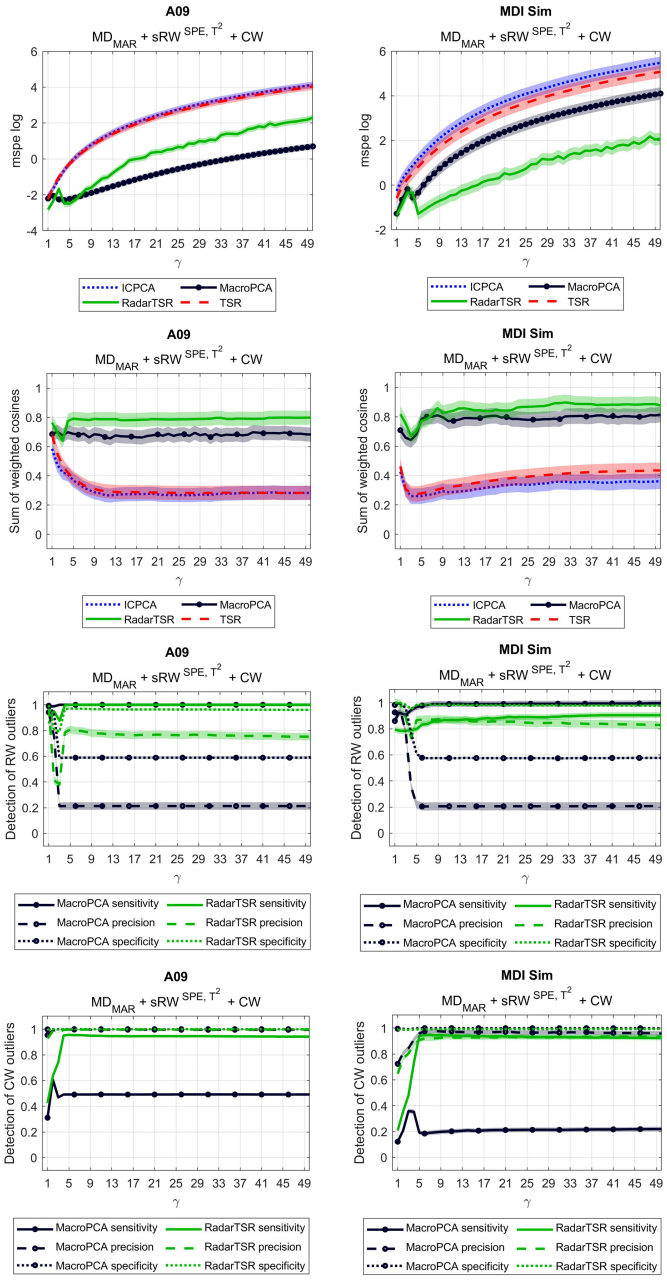


Figure 6.58: Missing data MAR (20%), single SPE and T^2 rowwise outliers (10%) and cellwise outliers (10%) case. More details are in the introduction of the Appendix.

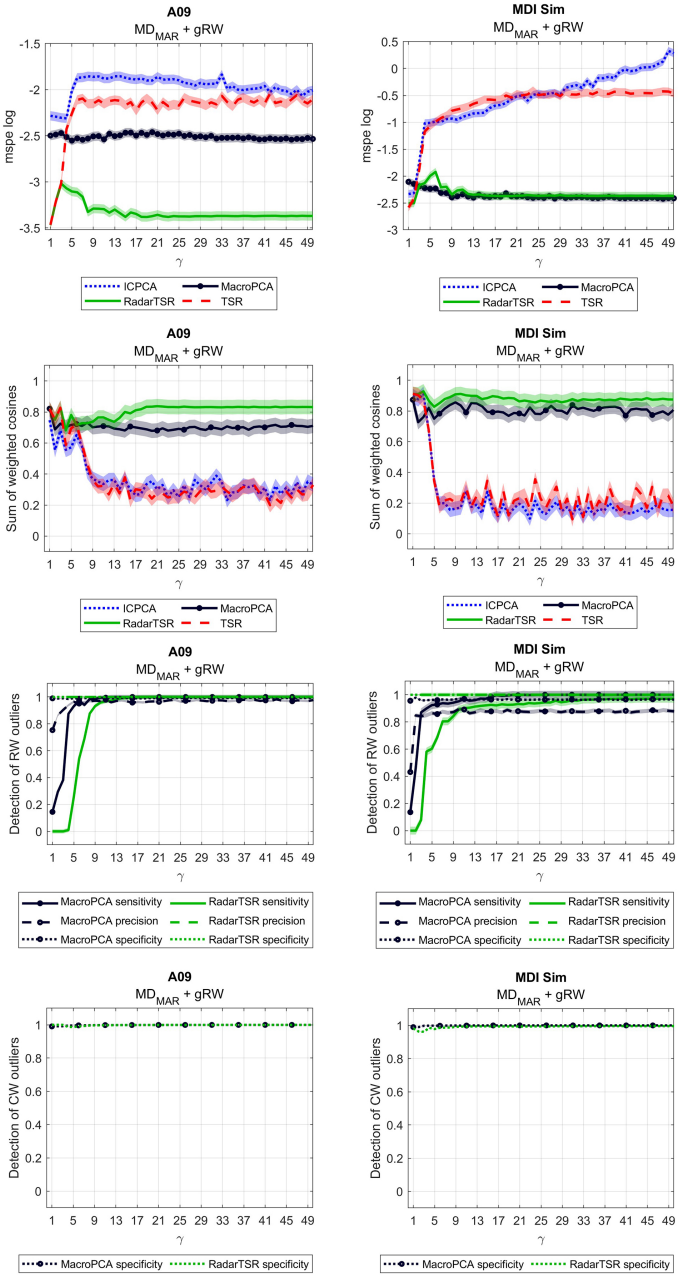


Figure 6.59: Missing data MAR (20%) and grouped rowwise outliers (20%) case. More details are in the introduction of the Appendix.

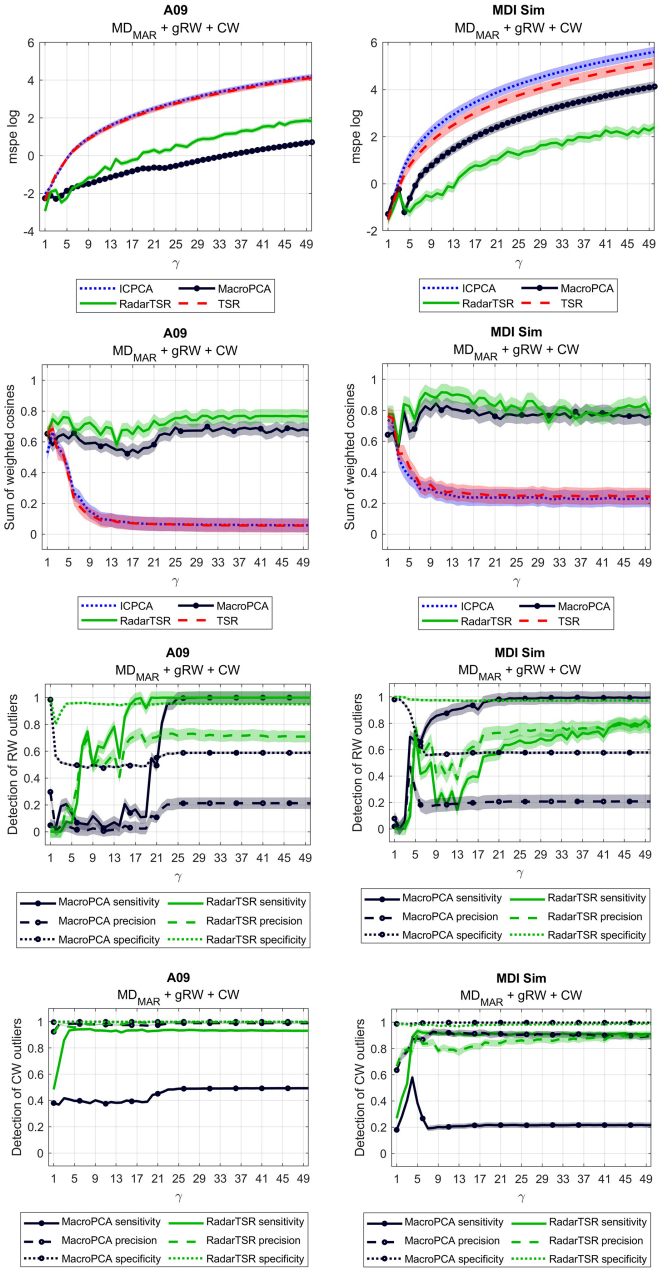


Figure 6.60: Missing data MAR (20%), grouped rowwise outliers (10%) and cellwise outliers (10%) case. More details are in the introduction of the Appendix.

Part III

New applications to real problems in biomedical engineering

Chapter 7

Healthcare process understanding and improvement

Part of the content of this chapter has been included in:

> [118] González-Cebrián, A., Hermenegildo, M., Climente, M., and & Ferrer, A. Multivariate Six Sigma: A case study in an outpatient pharmaceutical care unit. *Quality Engineering*. **34 (2)**, 277–289 (2022), https://doi.org/10.1080/08982112.2022.2042018.

7.1 Introduction

The application and interest of process improvement in hospital environments have grown in the last years [119]–[122]. Strategies such as Lean [123], Six Sigma (6S) [124] or their combination (Lean Six Sigma, L6S) [125], traditionally used in industrial or manufacturing sectors, are being widely used in other contexts, such as finance or healthcare.

There is a bunch of existing work that already shows how 6S and L6S concepts can significantly improve process performance. In terms of hospital service, improving performance can have multiple meanings: reducing prescription errors [122], reducing the waste of time [119], [126], [127], increasing patient satisfaction [128], etc. A hospital is a complex environment with many parallel processes that affect the same issue. For instance, staff rotations, interdependencies between internal services, and specific patient profiles affect and define the optimal workflow that should be applied to each case, and, therefore, data should reflect this reality as accurately as possible.

Thus, to improve the care of these patients using statistical tools like the ones included in the Six Sigma toolkit, it becomes mandatory to deal with increasingly complex datasets. This issue becomes even more critical considering the tendency towards personalized medical care, where the patient becomes the focus of the caring process, which means that forthcoming process improvement should account for information about patients and the hospital processes involving them.

With the new paradigm of Medicine 4.0, it is undeniable that the Six Sigma toolkit needs to be upgraded with machine learning (ML) tools and more sophisticated multivariate statistical techniques, such as latent variable-based models [129]. These tools can be beneficial for discovering patterns, exploring the data, and obtaining accurate predictions. Moreover, when a process's understanding and optimization are pursued, it is also interesting to guarantee causality using data-driven approaches.

This chapter includes a case study in which latent variable-based multivariate statistical techniques, such as PLS, are used as a Six Sigma statistical toolkit for healthcare process improvement. This work was carried out as a 6S project in an Outpatient Pharmaceutical Care Unit in the Department of Pharmacy at Hospital Universitario Doctor Peset in Valencia (Spain). This unit provides prescription drugs and pharmaceutical care services to outpatients. The outcomes of the multivariate Six Sigma approach will be compared

with the conclusions obtained by classical Six Sigma statistical tools, such as the Analysis Of Variance (ANOVA).

7.2 Methods

This section briefly introduces the 6S methodology and then clarifies how the PLS model is integrated into this framework. A description in detail of the PLS model can be found in Section 3.4.1.

- Define: problem selection and benefit analysis.
- Measure: translation of the problem into a measurable form and measurement of the current situation; refined definition of objectives.
- Analyze: identification of influence factors and causes that determine the critical to quality characteristics (CQCs) behaviour.
- Improve: design and implementation of adjustments to the process to improve the performance of the CQCs.
- Control: empirical verification of the project's results and adjustment of the process management and control system so that improvements are sustainable.

To illustrate the inclusion of PLS as a tool for Six Sigma projects, we followed a two-step procedure in this work:

1. To fit a PLS model that will point out interesting(new and suspected) relationships between process inputs and outputs. The general overview of the complex relationships between $\boldsymbol{X}$'s and $\boldsymbol{Y}$'s provided by the PLS weighting plot is a useful tool to drive the following steps in the Analyze phase. This provides a route map of what is worth studying in more depth.
2. To assess these potentially interesting relationships with traditional exploration tools.

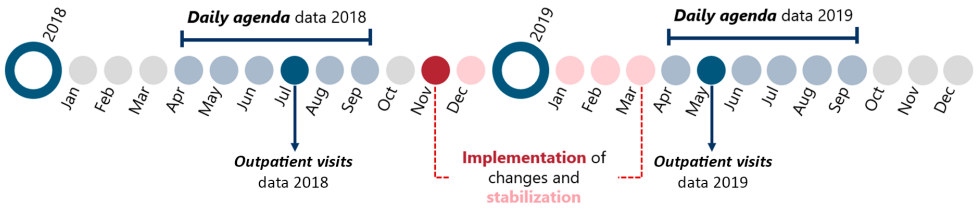


Figure 7.1: Timeline of the Six Sigma project, indicating the data recording periods and implementation of changes.

7.3 Datasets

The following sections describe each one of the datasets acquired for this 6S project. It is important to mention two data collection moments (Figure 7.1). The first took place in 2018, as part of the Measure phase, and the second took place in the same period of 2019, once the changes for the process improvement had been implemented and the process had enough time to stabilize after its new organization.

7.3.1 Daily agenda datasets (2018 and 2019)

This data showed an outlook of the daily activity in the Outpatient Pharmaceutical Care Unit: number of scheduled visits, number of recorded visits at the end of the day, and number of missed visits. Each one of these metrics was shown globally (accounting for all patients) and split by visit type: first, successive, and dispensing visit. The difference between successive and dispensing visits is that the former requires the attention of pharmaceutical staff, given that they may involve changes in medication doses or prescriptions. In contrast, dispensing could be performed both by pharmaceutical and nursery staff.

7.3.2 Outpatient visits datasets (2018 and 2019)

This database was designed on purpose by the Six Sigma technical team. Each day for two weeks, it recorded information about each outpatient visiting the Outpatient Pharmaceutical Care Unit. This required the assistance of additional personnel for the data collection and a strong engagement of all the staff, who responded very well to the demands of the technical team. The confection of a Fishbone diagram was used to determine potential causes affecting

the waiting time. The included variables collected information about several aspects of the visit:

- Information about the visit context: type of visit (type), day of visit (date and weekday), and hour of visit (turn). This last variable was split into three categorical variables: turn 1 (from 8:00 a.m. to 10:30 a.m.), turn 2 (from 10:30 a.m. to 12:30 p.m.), and turn 3 (from 12:30 p.m. to 2:30 p.m.).
- Information about the patient: assigned clinical service (service), the hour of arrival to the desk (arrival), and the hour of start and end of the pharmaceutical care consultation (enter, exit).
- Information about the treatment: if they were stored in the refrigerator (refrigerator), how many units were prescribed (number), and the route of administration (via).
- Information about the pharmacy unit staff attending the patient (professional) and the profile of the attending person (profile).

Two output variables (i.e., critical characteristics, CC) were calculated from all these variables: waiting time and attention time. The waiting time was computed as the minute difference between the entry and the arrival hours. The attention time was calculated as the minute difference between the exit and the entry hours.

7.4 Results

The project timeline went from July 2018 to September 2019 (Figure 7.1). This section will follow the pathway defined by DMAIC steps, illustrating the results and the project's process.

7.4.1 Define Phase

This stage aims to determine the improvement project potentially leading to reduced costs, increased customer satisfaction, etc. This implies a necessity of defining a practical problem to be tackled. By studying the process and its relation to the problem, an assessment of the costs and benefits of addressing the project goal can be evaluated. This provides a first clue about the necessity

of resources, staff involved in the project, and potential constraints. All these initial aspects were portrayed in the Project Charter (Figure 7.2).

This Six Sigma project focused on outpatients' timing (waiting and attention times) during their visit to the hospital's Department of Pharmacy. The reason why this was the main focus was based on data about previous years. The last outpatients' satisfaction questionnaire, performed between November 2016 and February 2017, showed that although the majority were globally very satisfied with the care, half of them evaluated the waiting time as short. This reflected an improvement opportunity based on the voice of the external customer, i.e., the outpatients. Besides, there was also an internal customer: the outpatient pharmaceutical care unit staff (nurses and pharmacists) suffering from daily work overload. Both voices aligned in the same direction: a consistent overflow of patients and long waiting times.

Figure 7.3 confirms the voice of the internal customer. Data about the 2018 year's (January – June) agenda showed an evident overload of patients. This overload was calculated daily as the difference between attended and scheduled outpatients for each day. Figure 7.3a shows this systematic overload, which can be appreciated by the consistent position of data points above the diagonal representing the equivalence between the planned patients on the agenda and the recorded visits for each day. This is also appreciated in Figure 7.3b, where the overload boxplots are above the zero reference (red dashed line) line for every month analyzed.

However, the data recorded until 2018 did not register each of the attended patients, losing the information about the timing per patient, which made it very difficult to formally raise the patient complaints and redesign the unit's workflow. Moreover, a big part of the staff had other activities in the hospital pharmacy department, implying that changes to the organizational scheme would be carried out only under solid evidence supporting the need for improvement since other adjacent unit processes could also be affected.

With the help of the pharmacist staff, a Suppliers, Inputs, Process, Outputs, Customers (SIPOC) diagram was outlined (Figure 7.4).

As a result of the process diagram and considering the Voice of the Customer (VOC), both internal and external, the project team designed and agreed upon a data-collecting scheme. The project was led by a Six Sigma black belt with a high profile in the hospital pharmacy organizational scheme, and the technical team consisted of six Outpatient Pharmaceutical Care Unit staff members and

Project Charter: Definition sheet Date: July 2nd 2018

Title of the project
Reduction of the average waiting time for patients of the hospitalary pharmacy unit of external patients

Project leader BB:	Process owner
Ana Moya	Mónica Climente

Team members

Marta Hermenegildo, Irene Miccichè, Ángel Marcos, Alba González, Tamara Lidia Paredes, Mercedes Riera, Carlos Cortés

Agents involved

Champion: Mónica Climente
Black Belt: Ana Moya
Work team Tamara L.P., Irene M., Mercedes R., Ángel M., Carlos C., Marta H., Alba G.

Problem description

The number of attended patients is growing since 2013, which generates planning difficulties in the agenda of the unit, dealing everyday with a 50% of non-scheduled patients. This generates waiting times of almost one hour. Moreover, the stress generated by this overload seems to be affecting also to the attention time, causing differences on the attention procedure between staff members.

Goals:	Metrics:	Initial value	Target value
Reduction of the average waiting time between 8:30 and 14:30.	Minutes	24	20
Reduction of the overload of visits between 8:30 and 14:30.	Patients/day	28	14

Expected economic results

(Results not measured in ecomomic terms)
Process improvement in terms of eliminating non-quality from the attention service provided to patients.
Process improvement increasing the efficiency of the service and reducing bottlenecks of the process.

Expected benefits for the clients

Increased satisfaction from the external client perspective, thanks to the reduction of the waiting time.
Increased satisfaction of the internal client (hospital pharmacy unit staff), thanks to the reduction of the overload and the pressure in the work environment.

Available resources

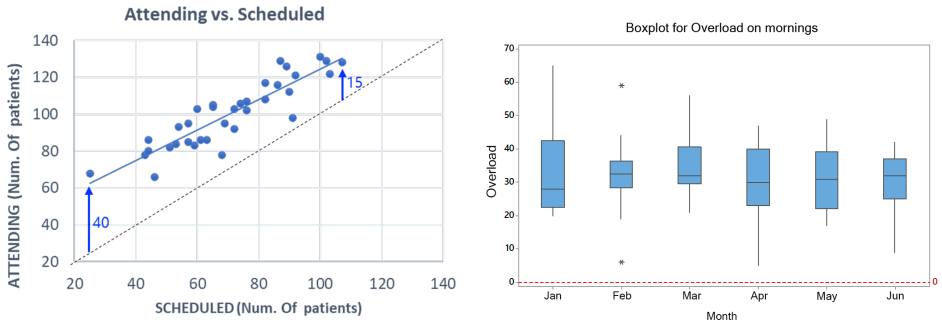
Staff: hospital pharmacy unit personnel (two residents, two pharmaceuticals, one nurse and two auxiliary staff.

Material: laptop and software for the statistical analysis of the data.

Project constraints

Starting date	July 2nd of 2018	Expected ending date:	October 5th of 2018

Figure 7.2: Project Charter of the Project.



(a) Attended vs scheduled patients with the diagonal representing the equivalence between the planned and the recorded visits for each day.

(b) Boxplot of daily overload during the data recorded in 2018 (January to June), where the red dashed line represents the zero reference.

Figure 7.3: Plots illustrating the systematic overload values during 2018 before starting the project, appreciated by the consistent position of data points above the zero-overload diagonal (a) and by the persistent position of the overload boxplots above the zero-overload line (b).

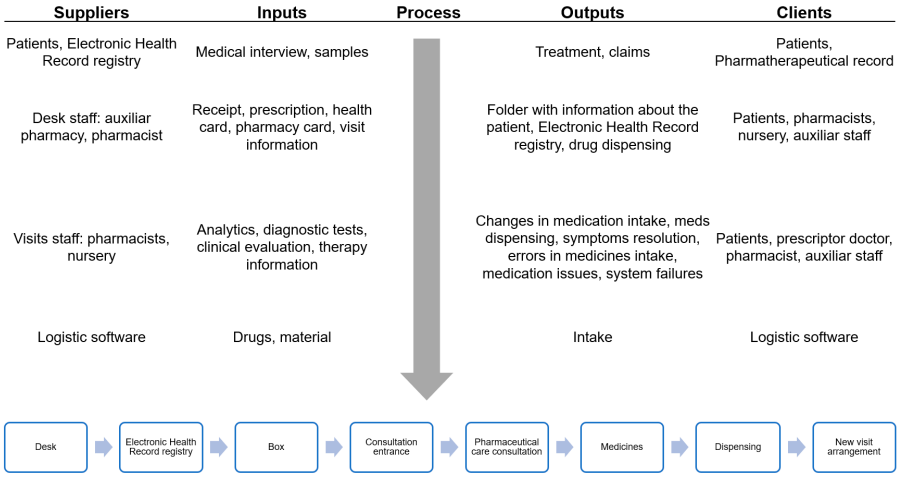


Figure 7.4: SIPOC diagram of the Outpatient Pharmaceutical Care Unit workflow.

two black belts with engineering and statistical backgrounds. The chief of the hospital pharmacy department championed the Six Sigma project.

The initial description of the project was the following: "The number of attended outpatients has been increasing since 2013, stressing out the scheduling of the Outpatient Pharmaceutical Care Unit agenda, and over 50% of the patients who attended daily had not been scheduled for that day. This results in waiting times of nearly an hour. Moreover, the stress of this systematic work overload may be affecting the attention time, generating differences between attending staff and, thus, an undesired variability on the caring process".

7.4.2 Measure Phase

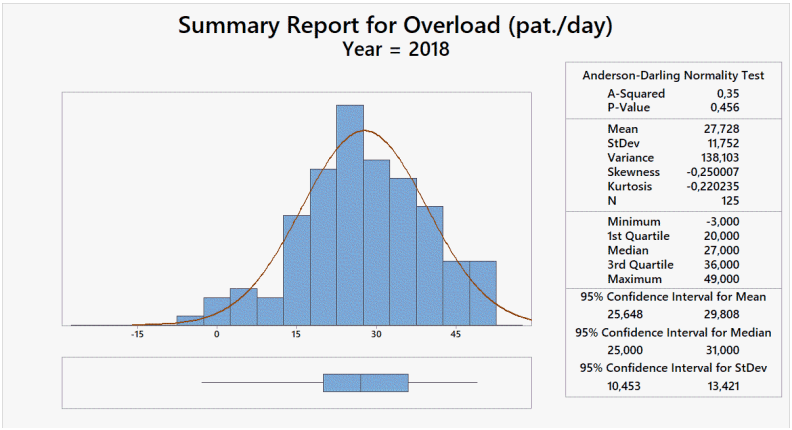
During this phase, the collection of the datasets described in Section 7.3 took place. In particular, the assembly of the Outpatients' visit data set (Section 7.3.2) required the assistance of additional personnel for the data collection and a strong engagement of all the staff, who responded very well to the demands of the technical team. The confection of a Fishbone diagram was used to determine potential causes affecting the waiting time. Two output variables (i.e., critical characteristics, CC) were calculated from all these variables: waiting time and attention time.

The waiting time was computed as the minute difference between the entry and the arrival hours. The attention time was calculated as the minute difference between the exit and the entry hours. The data were validated after checking the existence of transcription errors (such as negative duration). Since the pharmacy staff had notes and records about the visits, some mistakes could be solved. Still, all those entries with misleading information that could not be contrasted were not considered for further analysis.

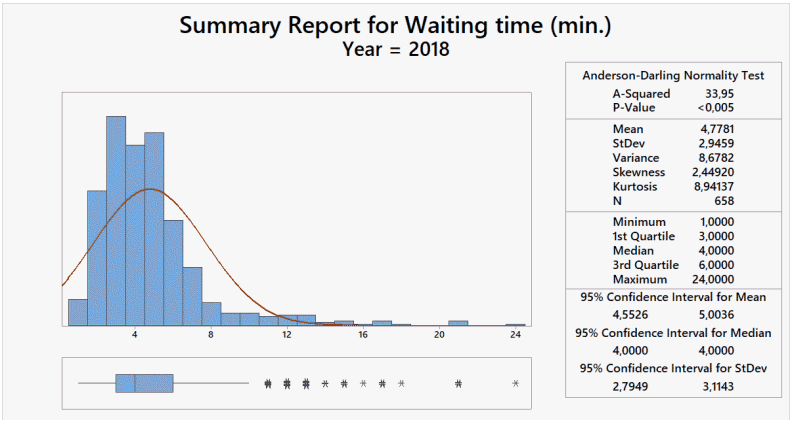
7.4.3 Analyze Phase

In this stage, the goal was establishing factors affecting the CCs: waiting and attention times. The reference values were an average waiting time of 24,17 minutes and an average attention time of 4,78 minutes. These values were obtained from the data recorded in 2018. A descriptive summary of the initial situation from 2018 can be found in Figure 7.5.

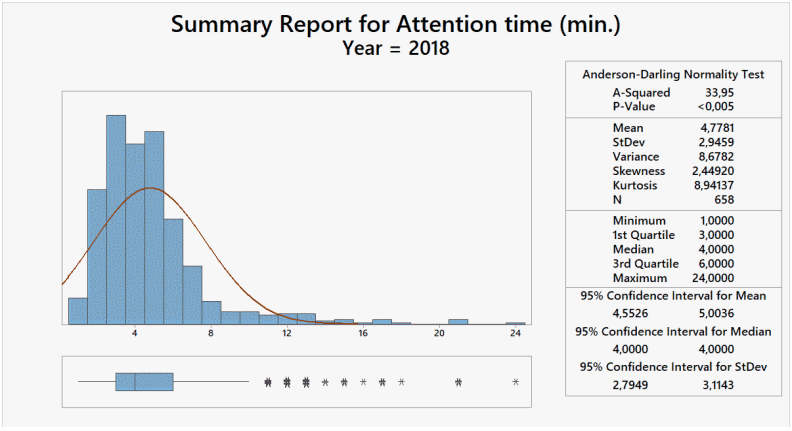
This figure summarizes the CCs (along with the overload of patients) at the beginning of the Six Sigma project. Setting reference values is critical to quantitatively proving the usefulness and success of the improvement actions and



(a) Descriptive summary of the overload in 2018.



(b) Descriptive summary of the waiting time in 2018.



(c) Descriptive summary of the attention time in 2018.

Figure 7.5: Descriptive statistics for CCs according to the data from 2018.

the Six Sigma project overall. Thus, reference values shown in Figure 7.5 will be the base for the later comparison between the pre-Six Sigma project situation of the process (from 2018) and the post-Six Sigma situation (from data recorded in 2019).

Regarding the waiting time, the analysis focused on checking for any pattern of visits related to longer waiting times. Moreover, the study of the visit's data set in 2018 would enable precise quantification of the average waiting time, setting a reference for the Six Sigma project. The attention time presented another casuistic. Given the comments of patients arguing unfair differences in the caring process, the goal was to establish if there was an undesired variability in attention time for the same visit profile. This would reflect a difference in the attention protocol followed by different staff members, which could impact the quality of the caring process.

A PLS model was fitted to get this information, including all predictor variables and CC . This analysis would let us identify the sources of variability of the attention process affecting each CC. Weighting plots were used to interpret the relationships between process variables and CCs found by the model.

The PLS analysis on the outpatient visits data set 2018 ($N = 658$, $K = 13$, $L = 2$) pointed out some interesting facts. In the weighting plot (Figure 7.6), the attention and waiting times (CCs) are represented by red squares along two almost orthogonal directions of variability, showing a lack of relationship between those two CCs. The directions of variability aligned with the CCs (red dashed lines) give information on the degree of correlation between predictors (i.e., process variables) and each of the CCs. The closer to the extreme of a predictor's red dashed line, the more correlated it will be with the CC associated with this direction of variability. This correlation will be positive if the predictor is located on the same side of the CC and negative if it is on the opposite side of the red dashed line.

This plot gives a clear picture of the latent structure of the process in the hospital pharmacy unit, showing that the process affecting the attention time is nearly independent of the process affecting the waiting time. The following analyses will focus on each CC independently to ease the understanding of both processes.

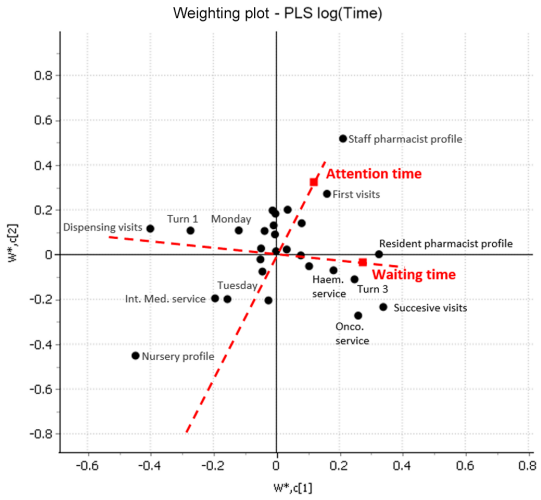


Figure 7.6: Weighting plot highlighting the relationships of process variables to the waiting time and the attention time. The orange-dotted contour circles process variables positively correlated with waiting time, and negatively correlated predictors are contained within the blue-dashed contour.

Waiting time

According to PLS results from Figure 7.6, it would be worth closely checking the relationship between the turn, the type of visit, the oncological and haematological specialities, and the resident profile concerning the waiting time. This can be seen in the weighting plot (Figure 7.7), which shows that successive visits, Oncology or Haematology patient visits, Turn 3 visits, and visits attended by a Resident are related to longer waiting times. On the contrary, dispensing visits and visits occurring in Turn 1 are associated with shorter waiting times.

This information is also displayed in Figure 7.8, where the PLS coefficients indicate the relationship between predictors ($\boldsymbol{X}$) and the response variables ($\boldsymbol{Y}$), in this case, waiting time. This plot shows only statistically significant predictors (whose 95% jackknife confidence intervals do not contain the zero value). The interpretation of this plot is the following: process variables with positive and statistically significant $\boldsymbol{B}_{PLS}$ coefficients are positively correlated to waiting time, while process variables with negative and statistically significant $\boldsymbol{B}_{PLS}$ coefficients are negatively correlated to waiting time.

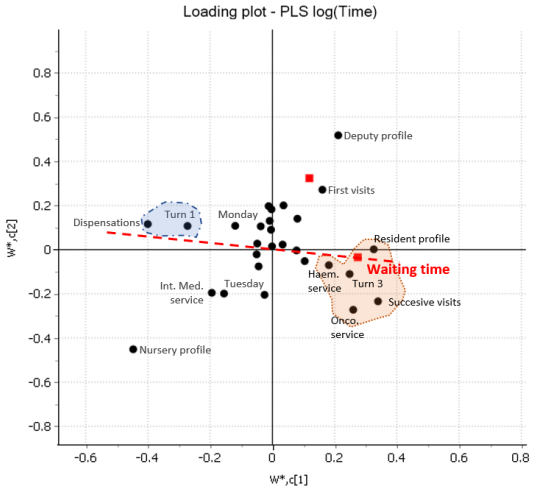


Figure 7.7: Weighting plot highlighting the relationships of process variables to the waiting time and the attention time. The orange-dotted contour circles process variables positively correlated with waiting time, and negatively correlated predictors are contained within the blue-dashed contour.

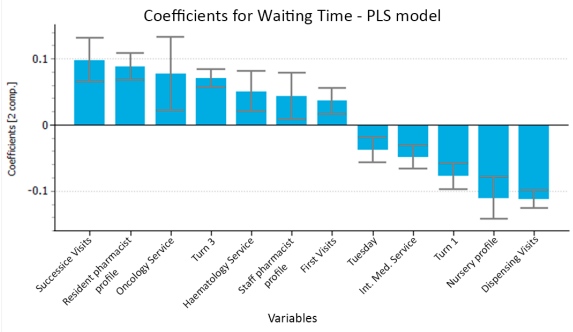
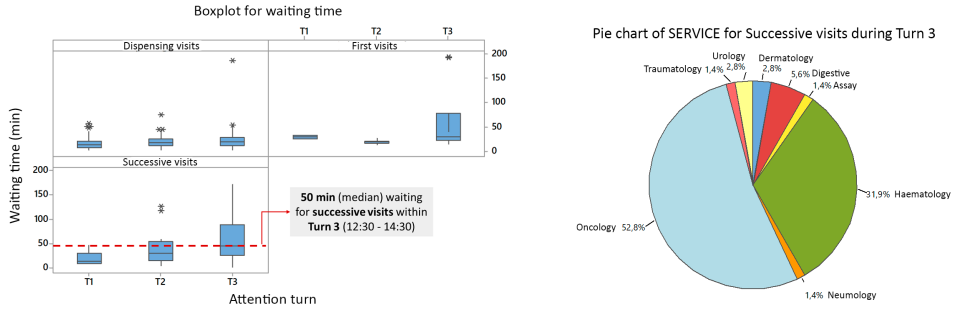


Figure 7.8: $\boldsymbol{B}_{PLS}$ coefficients plotting the relationship between variables in $\boldsymbol{X}$ and the waiting time.



(a) Boxplots of waiting time for each type of visit (dispensing, successive, and first visit) and at each turn (T1, T2, and T3).

(b) Pie chart of the types of assigned hospital service of successive visits during Turn 3.

Figure 7.9: Plots showing the relation between each Turn with the waiting time (a) and with the assigned hospital service (b).

Figure 7.9a shows an increasing trend of the waiting time along with the visit turns. This is particularly evident for successive visits. Besides, more than 80% of successive visits during turn 3 were for patients from the oncology (52,8%) or haematology (31,9%) service, as highlighted in Figure 7.9b.

The association between these process variables relied on oncological and haematological visits attended mainly by a resident and only in Turn 3. Arriving at complex associations like this can be tricky and time-consuming via univariate descriptive charts and analyses. In contrast, this relationship between several factors (onco/haema services, Turn 3, successive visits, and resident profile) and the waiting time stands out from the PLS analysis (Figure 7.9).

As the PLS revealed, All this evidence pointed toward a bottleneck associated with Turn 3 and Oncology/Haematology-associated patients. Moreover, in an eyeshot, the weighting plot from the PLS analysis also showed that visits scheduled at the first hour (Turn 1) or on Mondays seemed to be related to shorter waiting times (Figure 7.9a).

To quantify the statistical significance of these effects through the classical Six Sigma statistical toolkit, we run some ANOVA tests to doublecheck these hypotheses. A univariate ANOVA test finally confirmed the statistically significant effect of assigned hospital service on waiting time (P-value < 0.05).

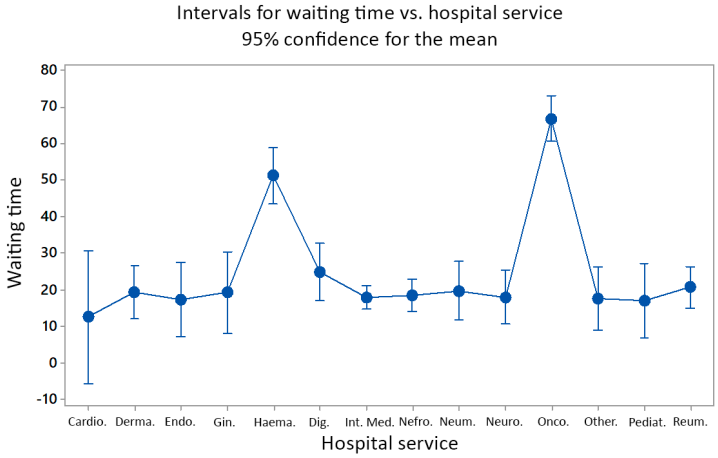


Figure 7.10: 95% confidence intervals for the mean waiting time (minutes) for each assigned hospital service

A Fisher LSD test with a 95% confidence level for multiple comparisons (Table 7.1) shows that Oncological and Haematological visit profiles had statistically significant longer average waiting times than the rest of the hospital services.

Attention time

The PLS weighting plot (Figure 7.11) and the $\boldsymbol{B}_{PLS}$ coefficients plot (Figure 7.12) showed that the nursery staff profile was associated with the shortest attention times, whereas pharmacists and resident profiles, were associated with the longest attention times.

Similar to the waiting time analysis, a univariate doublecheck was carried out. An ANOVA test confirmed the statistically significant relationship between attention time and the professional profile of the attending staff. To make a fair comparison, only those visits that all professional shapes could attend (and hence, were comparable) were included.

As seen, nursery staff showed statistically significant shorter attention times than the other professional profiles staff (Figure 7.13 and Table 7.2).

Table 7.1: Fisher LSD intervals with a confidence level of 95% for the difference between mean waiting time for each hospital service.

Service	N	Average	Group		
Onco.	51	66.88	A		
Haem.	34	51.29		B	
Dig.	32	25.13			C
Reu.	61	20.90			C
Neum.	31	19.87			C
Derma.	37	19.38			C
Gin.	16	19.38			C
Nefro.	98	18.61			C
Neuro.	36	18.17			C
Int.Med.	198	18.11			C
Other	26	17.73			C
Endo.	19	17.58			C
Pediat.	19	17.11			C
Cardio.	6	12.67			C

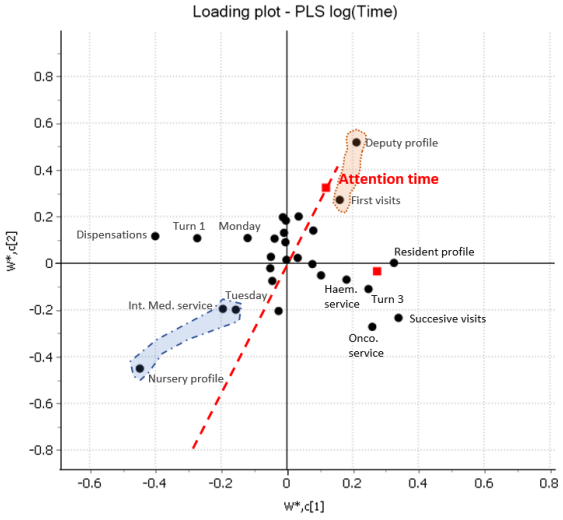


Figure 7.11: PLS weighting plot highlighting the relationship of process variables associated with the attention time. The orange dotted contour circles process variables positively correlated with the attention time, and negatively correlated predictors are contained within the blue dashed contour.

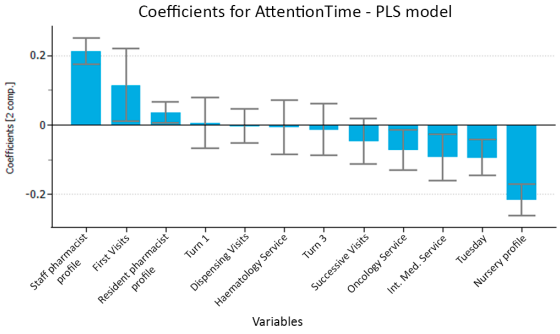


Figure 7.12: PLS coefficients for the relationship with variables in and the attention time.

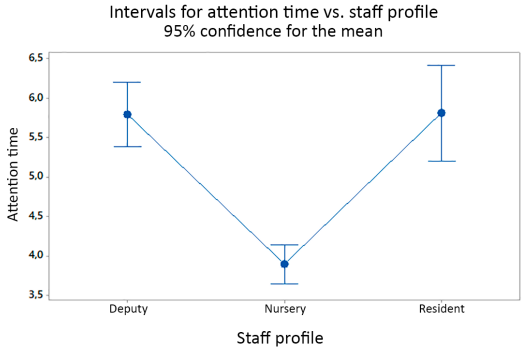


Figure 7.13: 95% confidence intervals for the mean attention time for each professional profile.

Table 7.2: Fisher LSD intervals with a confidence level of 95% for the difference between mean attention time assigned to the different staff profiles.

Staff	N	Average	Group	
Staff pharmacist	128	5.80	A	
Nursery	344	3.90		B
Resident pharmacist	58	5.81	A	

These differences were, on average, two minutes. Considering that the average attention time of these visits was between 4-5 minutes, these differences represented between 40% and 50% of the visit duration. This variability could imply substantial differences in the attention procedure protocol. On the one hand, if shorter times do not mean worse attention, then time is being wasted by more prolonged attention procedures. On the other hand, shorter times could imply less careful attention, which could become critical in health matters such as this one.

7.4.4 Improve Phase

After the analysis performed on waiting and attention times, the technical team of the Six Sigma project had a meeting to discuss the reported results and possible improvement actions.

Regarding the longer waiting times for the Onco-Haema visits, all the team agreed that attending all these visits on a specific Onco-Haema turn had become a bottleneck. This was initially done because these patients may change their medication more frequently, requiring supervision and approval from a pharmacist specializing in oncological and haematological treatments. However, the distribution of the attention time for Onco-Haema visits (Figure 7.14) showed that most of them had a duration below 5 minutes. This meant that most of these visits did not need a comprehensive re-evaluation of the medication and were just for drug dispensing.

To alleviate this bottleneck, it was proposed that those patients whose oncologist had not changed the medication did not need a specific visit. Thus, pharmacists could attend them all morning, not only during turn 3.

Another improvement to reduce the waiting time was to add a specific hour for outpatient scheduling. The usual procedure involved only a day of the schedule. However, Turn 1 in the morning showed little waiting times (see

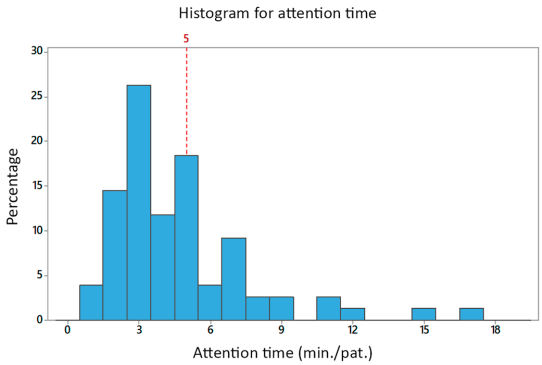


Figure 7.14: Histogram of the attention time for oncological and haematological visits.

Figure 7.9a), which indicated that scheduling more patients at this time would improve the patient flow, preventing the accumulation on Turn 3.

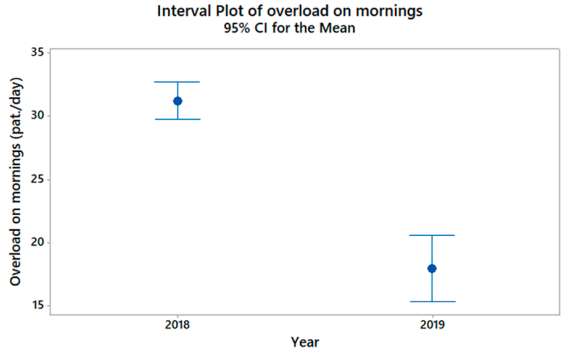
Finally, there were some improvements regarding the variability in the attention time. Attention protocols were designed and implemented to standardize the time and depth of the attention for each visit. All the proposed changes were implemented in November 2018. In May 2019, the Outpatient Pharmaceutical Care Unit had implemented regularly all the proposed changes.

7.4.5 Control Phase

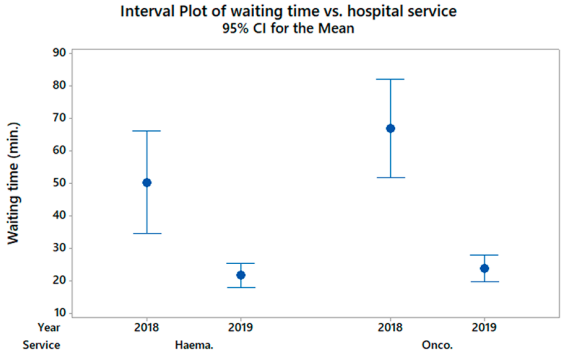
Once the improvements were shown to work, the activity tracking was kept on. An intensive data collection for another two weeks was done to evaluate the effects of the changes in the unit workflow. This data collection yielded two new datasets: the daily agenda data for 2019 ($N = 124$, $K = 5$) and the outpatient visits data for 2019 ($N = 1043$, $K = 13$, $L = 2$). The comparison between the initial and final output values can be seen in Figures 7.15 , 7.16b and 7.17, and Tables 7.3 , 7.4 and 7.5.

As can be seen in Figure 7.15a and in Figure 7.16a 5, the overload of patients changes from its historical values (January – October 2018), gradually decaying over November 2018 to March 2019, and finally stabilizing around April 2019. These differences are stated monthly (Figure 7.16b).

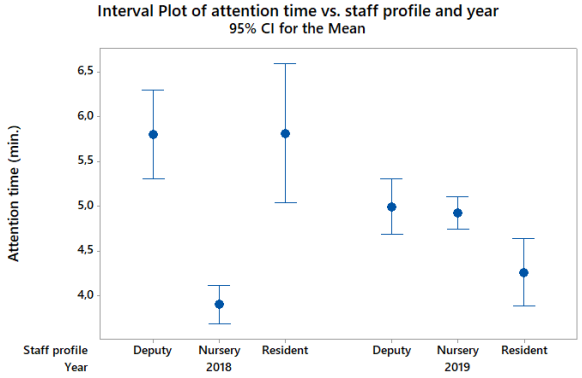
Table 7.3 shows the LSD interval that confirmed these differences to be statistically significant (p-value<0.05): in 2019, there were, on average, nearly 20 patients less daily overload than in 2018.



(a) LSD intervals for the differences in the overload on mornings.

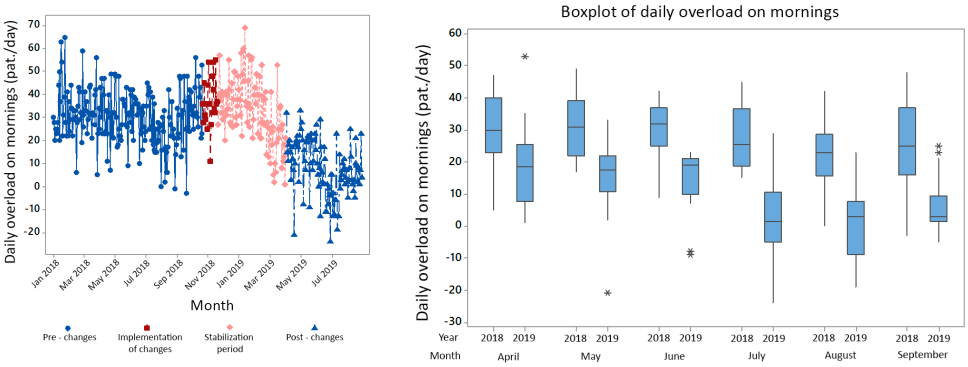


(b) LSD intervals for the differences in the waiting time for oncological and haematological patients.



(c) LSD intervals for the differences in the attention time for the three staff profiles.

Figure 7.15: 95% confidence intervals comparing the situation before (2018) and after (2019) the changes in the Outpatients Pharmaceutical Care Unit.



(a) Time series of the overload from January 2018 to September 2019.

(b) Boxplots of the overload over the comparable months (April to September) of 2018 and 2019.

Figure 7.16: Plots showcasing the temporal evolution of the patients' overload, comparing the situation before the L6S project and after it.

Table 7.3: Fisher LSD interval for the difference between mean outpatients' overload of 2019 and 2018.

Metric	**LSD intervals at 95% for the difference 2019 − 2018 (patients/day)**
Overload	$[-21,38; \quad -15,23]$

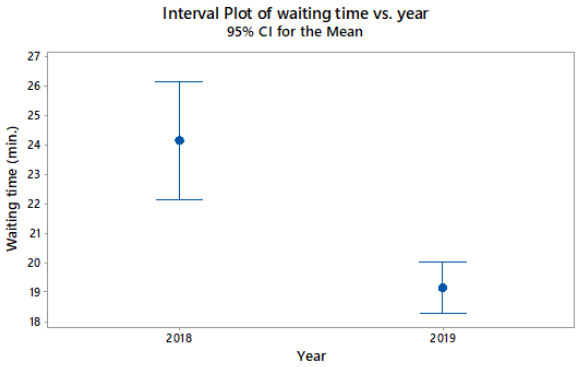


Figure 7.17: Confidence Intervals for the waiting of 2018 and 2019.

Table 7.4: Fisher LSD intervals for the differences between mean waiting times of 2019 and 2018 for oncological and haematological outpatients and all other medical specialities.

Service	**LSD intervals at 95% for the difference 2019 − 2018 (min./patient)**
Onco.	[−56.45; 29.85]
Haema.	[−39.55; 17.78]
Others.	[−1.71; 1.09]

Regarding the waiting time, there was a significant reduction in the overall waiting time, between 3 and 7 minutes per patient. This meant a reduction of the mean waiting time from 24 minutes per patient to 19 minutes (Figure 7.17), achieving the project goal (with one minute more of reduction). This difference was even more noticeable for the waiting time for the Oncological and Haematological patients (Figure 7.15b).

Table 7.4 shows the 95% LSD confidence intervals for the difference between average waiting times for these two services, where a statistically significant reduction can be appreciated. Moreover, there was no statistically significant increase in the waiting time for all other medical specialities (95% LSD interval contains the zero value). This result provided a solid improvement and achievement of the 6S project.

Finally, Figure 7.15 (c) shows that differences between attention times were also reduced. Table 7.5 shows two interesting things. First, the biggest time gap between attention times is now 1.2 minutes/patient, which was a reduction of 52% concerning the previous maximal difference (2.5 min/patient, Table 7.5 3).

Table 7.5: Fisher LSD intervals for the differences between mean attention times of 2019 for different professional profiles.

Profiles	LSD intervals at 95% for differences between staff profiles (min./patient)
Nurs – Staff phar.	[−0.42; 0.27]
Res. Phar. – Staff phar.	[−1.21; −0.26]
Res. Phar. – Nurs.	[−1.09; −0.24]

Table 7.6: Summary of the improvement goals, the implemented changes on the workflow of the hospital pharmacy unit, and the results obtained after the implementation.

Improvement goal	Implemented change	Outcome
To reduce overload	Schedule all patients with day and hour	Reduction of average overload between 15 and 21 patients/-day
To reduce the waiting time of turn 3	Non-specific oncological and hematological visits are moved to turns 1 and 2	Reduction of waiting time between 3 and 7 min./patient
To reduce differences in attention times for Scheduled Dispensing visits	Standard caring protocols	Reduction from 2.5 min. of difference in 2018 to 1.2 min. in 2019.

Secondly, differences are presented for a different professional profile (resident) after the protocol update.

This outcome can be used as evidence for future updates of the attention protocol, focusing now on reducing the variability of the attention time due to the different performances of professional profiles.

Table 7.6 summarizes the goals, the implemented changes, and the outcomes obtained for each one of the improvement goals.

After controlling the improvements, the collection of the agenda database was kept on, registering all the daily information about outpatient schedules. This database provides a continuous flow of data analyzed by the Outpatient Pharmaceutical Care Unit staff and the Department of Pharmacy.

The success of this project also meant the configuration of a solid, continuous improvement technical team in the Outpatient Pharmaceutical Care Unit, which is responsible for future updates and changes in response to the results of this project and to further data.

7.5 Conclusions

In this work, PLS has been incorporated into the Six Sigma toolbox to explore and analyze the data set from a Six Sigma project in a university hospital's Outpatient Pharmaceutical Care Unit in the Department of Pharmacy. In contrast to univariate techniques, PLS shows in a single shot an overall picture of how input and output variables of the caring process are correlated, providing a clear interpretation of the results that becomes crucial for process understanding and implementing actions for improvement. Moreover, PLS findings efficiently guide the confirming process using classical Six Sigma tools such as ANOVA, simplifying the number of statistical tests needed, if needed.

Thus, the classical Six Sigma DMAIC scheme can be upgraded for a more effective and time-saving methodology able to work with increasingly complex databases by including latent variable-based techniques, such as PLS, in the future to the next generation of process improvement methodology in 4.0 environments: the Multivariate Six Sigma.

Chapter 8

Biomarkers extraction for chronic fatigue syndrome

Part of the content of this chapter has been included in:

> [130] González-Cebrián, A. et al., Diagnosis of Myalgic Encephalomyelitis/Chronic Fatigue Syndrome With Partial Least Squares Discriminant Analysis: Relevance of Blood Extracellular Vesicles, *Front. Med.*, **9**, (2022), `https://doi.org/10.3389/FMED.2022.842991`.

8.1 Introduction

Myalgic Encephalomyelitis/Chronic Fatigue Syndrome (ME/CFS) is a highly debilitating disease characterized by unexplained profound fatigue lasting over six months (ICD-10 code R53.82 or G93.3 if post-viral) [131], which is exacerbated by physical, mental or emotional activity, a process known as post-exertional malaise (PEM); lack of restoring sleep, dysautonomia, and frequent additional comorbidities [132].

Despite recent intense biomarker research, ME/CFS diagnosis relies on clinical symptom assessment after excluding potential underlying health problems that could relate to patient symptoms [133]–[135]. Previous studies explored the potential of genome-wide screenings of microRNAs contained in Peripheral Blood Mononuclear Cells (PBMCs) and Extracellular Vesicles (EVs) collected from blood samples [136]–[139]. However, no miRNA was widely validated as a biomarker of ME/CFS, and all identified so far appear to have limited diagnosis value, individually or when combined.

One potential limitation of these studies could be related to the applied statistical methodology. The use of rudimentary statistical methods such as two sample tests (i.e., t-test or Wilcoxon-Mann Whitney test), followed by multiple comparison corrections (i.e., Bonferroni or False Discovery Rate, [140]) for the analysis of $-omic$ data present several drawbacks. These include low statistical power, lack of interpretability of results, and the omission of complex relationships among variables, which could, in principle, be addressed using statistical models such as linear or generalized linear models. However, these methods also suffer from other problems when dealing with $-omic$ data, such as a large number of variables and low sample size, which produces overfitting, and a high correlation among variables, which produces multicollinearity. Those limitations have motivated the development of numerous novel statistical techniques during the last decades [141].

Prediction methods such as Partial Least Squares (PLS) [61] is a technique especially suitable for the analysis of $-omic$ data due to its ability to deal with more variables than observations, and by its good model interpretation capacity [142]. Section 3.4.1 from Chapter 3 provides more information on PLS. PLS was conceived as an alternative to classical regression, modelling the latent space of predictors and responses ($\boldsymbol{X}$ and $\boldsymbol{Y}$ subspaces, respectively) and finding the subspace that maximizes the covariance between both latent subspaces. Barker and Rayens proposed PLS-DA (Discriminant Analysis) as a variant of PLS for binary responses [62].

This chapter used PLS-DA to classify individuals in the healthy control or case group and determine which variables hold the best discriminant power between these two classes of participants. This study was the first to provide a PLS-DA model for accurately diagnosing severe ME/CFS based on a discreet combination of variables. In addition, it was also the first to use Raman fingerprints of EVs to enhance the ability to discriminate severely affected ME/CFS patients from healthy controls.

8.2 Methods

The study included three PLS-DA models:

1. The first one was a multiblock PLS-DA [143] model (Section 8.4.1), applied to over 800 variables obtained from 15 severe ME/CFS female cases and 15 matched healthy controls from the ME/CFS UK monographic biobank. Data included subject phenotyping with validated instruments, complete blood analytics, miRNA profiles from peripheral blood mononuclear cells (PBMCs) and from plasma-isolated extracellular vesicles (EVs), plus EV-associated features, as previously described [137]. The results showed that a combination of 32 variables, including several EV features, best discriminates severe ME/CFS cases from healthy subjects. Raman spectroscopic data further supported the value of EV features for the assessment of ME/CFS.

2. The second PLS-DA model (Section 8.4.2) focused on detecting discriminant regions of the Raman spectra. These results were compared with classification based on Raman spectra using three other binary classification techniques: an adaptation of linear discriminant analysis (LDA) [144] to deal with more variables than observations, random forest (RF) [66] and support vector machines (SVM) [145].

3. Finally, a multiblock approach was used again for the third PLS-DA model (Section 8.4.3), which included the previously mentioned set of 32 variables from the first PLS-DA and the relevant regions of the discriminatory spectra from the second PLS-DA. This approach aimed to determine if the good performance of the first PLS-DA model could be maintained by integrating relevant Raman spectra information and reducing the number of required miRNAs from PBMCs.

These PLS-DA models had two goals: obtaining an accurate classifier usable with new individuals and interpreting the discriminant features. Given the small sample size of the database, we followed a two-step procedure:

1. First, we used all observations (i.e., participants) to fit a PLS-DA model, obtaining a set of statistically significant discriminant predictors. This way, most observations could be used to fit the PLS-DA model, reducing the uncertainty in estimating the model's parameters, which is a critical aspect of the interpretation goal.
2. Secondly, the dataset was split into calibration and validation subsets. The PLS-DA model was fitted using the relevant predictors of observations from the calibration subset, and the model was then used to predict new observations from the validation set. Eight randomly selected individuals were included in the validation subset (four ME/CFS cases and four controls).

It is important to mention that each dataset used different data preprocessing schemes, calibration, and validation schemes.

A multiblock approach with block scaling and variable autoscaling was applied for the first PLS-DA model. Each block contained a different group of variables with similar features. Five blocks were established: (i) Demographic Variables, (ii) Analytic Variables, (iii) PBMCs miRNA expression levels, (iv) EVs miRNA expression levels, and (v) EVs characteristics. The third PLS-DA model (multiblock) included an additional block with relevant Raman profile features.

For the Raman spectra PLS-DA model, the goal was to determine if an accurate diagnostic tool could be developed solely based on Raman spectra differences. It was crucial to compare all classifiers not only in terms of classification performance but also in terms of model stability. For this reason, the chosen setup consisted of a three-fold cross-validation scheme. Each fold contained 1/3 of the data, i.e., each containing ten observations (five of each class). In each round, two folds were used to fit the model, and the other fold was used as an external validation set. This way, all observations were used to fit and validate the model, studying the stability of its performance. In this model, the preprocessing consisted of variable centring.

Once a PLS-DA model is fitted, it is pretty common to follow an iterative depuration procedure variable-wise and observation-wise until a PLS-DA model without outliers and relying only on relevant predictors is obtained. On the

one hand, outlying observations were studied in terms of the Squared Prediction Error (SPE) and Hotelling T^2 metrics (Chapter 3, Section 3.4.1).

On the one hand, it is quite common to find that some predictors are not relevant, and if so, removing irrelevant predictors can reduce the uncertainty of the model's estimates, as the number of parameters to be estimated is decreased. This can be especially helpful for "fat" case studies with $N << K$, as this one. The variable-wise depuration in PLS models is carried out by assessing the b and VIP coefficients. When the confidence interval of a b coefficient contains a zero value or the confidence interval of a VIP coefficient is below one, that predictor might be considered not statistically significant, being removed and refitting the PLS-DA model. For the parameters and outcomes of the PLS-DA model, statistical significance was assessed by jackknife intervals at a 95% confidence level. These intervals are calculated in a cross-validation scheme implemented by the Aspen ProMV© software used to obtain the PLS-DA model.

The performance of the depurated PLS-DA models was evaluated by the R^2 coefficient (goodness of fit) and the Q^2 coefficient (goodness of prediction). Permutation tests were used to assess the statistical significance of the model using the SIMCA© software. The Receiver Operating Characteristic (ROC) curve was also obtained to evaluate the model's classification performance. For each ROC curve, the AUC (Area Under the Curve) was calculated [146].

Finally, to confirm and visualize the discriminant properties of the selected variables (i.e., those showing statistical significance in the PLS-DA), a bivariate two-sample t-test was applied *a posteriori* to each potential biomarker included in the final multiblock PLS-DA model.

8.3 Datasets

Ethical approval of the study was granted by the Public Health Research Ethics Committee DGSP-CSISP, Valencia (Spain), (study number UCV_201701) and by the UCL Biobank Ethical Review Committee-Royal Free London NHS Foundation Trust (B-ERC-RF), (study number EC2017.01) before the UK ME Biobank released the samples.

Patient recruitment and clinical assessment for the monographic UK ME Biobank was mainly performed through the UK National Health Service (NHS) primary and secondary health care services [137]. Compliance with the Canadian Consensus [134], CDC–1994 [133] and Institute of Medicine (IOM, 2015) criteria

were ensured for patient recruitment [147], [148]. Clinical diagnosis was complemented with score differences in the SF-36 questionnaire ([149]) and the GHQ (General Health Questionnaire) ([150]), the last also assessed by a Likert scale [137].

Participants were excluded if: (i) had taken antiviral medication or drugs known to alter immune function in the preceding three months; (ii) had any vaccinations in the preceding three months; (iii) had a history of acute and chronic infectious diseases such as hepatitis B and C, tuberculosis, HIV (but not herpes virus or other retrovirus infection); (iv) had another chronic disease such as cancer, coronary heart disease, or uncontrolled diabetes; (v) had a severe mood disorder; (vi) had been pregnant or breastfeeding in the preceding 12 months; or (vii) were morbidly obese (BMI$\geq$ 40). Relevant guidelines and regulations are performed on all methods. All subjects signed informed consent before samples could be included in the corresponding sample collection.

The final participants were women with an average age of 46.8 (age range 38 - 53) for the disease cohort and 45.2 (age range 18 - 52) years for the healthy-matched control group. Median ages were 48 and 47 for the ME/CFS and healthy control groups, respectively. The average time from disease onset was 17.5 (range 1.5 - 30.9) years, with a median of 18.4 years. Health survey SF-36 and General Health Questionnaire (GHQ) scores, including Likert scale for the GHQ, scores separated ME/CFS and HC groups (p-value$<$0.05) [137].

For all these patients, the miRNAs variables corresponded to Nanostring data generated by Almenar-Pérez et al. [136], available from the NCBI Gene Expression Omnibus (GEO) database (Accession Number GSE141770) and the Supplementary information of the cited article.

The samples for the Raman analysis consisted of EV aliquots from the cited study isolated from 0.5 ml of platelet-poor plasma from 15 severely ill ME/CFS females and 15 age-population-matched healthy females, obtained from dipotassium EDTA blood-collection tubes (Becton Dickinson, Franklin Lakes, NJ, USA) by UK ME Biobank professionals. Tubes were centrifuged at $10,000xg$ for 10 mins, with Total Exosome Isolation Reagent (TEIR) (Invitrogen by Life Technologies, Cat. 4484450), and the isolated EVs were characterized following MISEV (Minimal information for studies of extracellular vesicles) [136], [151].

After diluting the isolated EVs to a concentration of 5$\times$108 EVs/ml in distilled water, 1.5 μL of the suspension was deposited on aluminum Raman slides and exposed to room temperature until the sample was dehydrated. Spectra were acquired using an HR Evolution confocal Raman microscope (Horiba

Jobin-Yvon, UK, Ltd) with a 532 nm laser. Laser power was 4.5 mW, and a filter of 25%. The acquisition time per spectrum was 3 s at a resolution of 4 μm. All spectra were preprocessed by cosmic ray correction, polyline baseline correction, and area normalization using the entire spectral region, using LabSpec 6 (Horiba Scientific, France).

The complete set of data included 34 blood analyte variables, 775 miRNAs expressed above threshold levels (136 in PBMCs and 639 in EVs), EV concentration, size, and z-potential of vesicles prepared with and without proteinase K treatments for a total of six EV-associated measures, together with two demographic variables. It also included fifteen variables obtained from the SF-36 questionnaire [149] and the GHQ questionnaire [150], the last assessed by a Likert scale (Likert, 1932). Table 8.2 contains a legend of each variable name in the upcoming figures and explains its meaning.

Four classification models were trained with a three-fold cross-validation setup to classify a spectrum as either severe ME/CFS or HC using an adaptation of linear discriminant analysis (LDA) [144] to deal with more variables than observations, random forest (RF) [66], a support vector machine (SVM) [145] and PLS-DA [62]. The classifier learning app in MATLAB© was used to optimize model hyperparameters for the LDA, RF, and SVM models. The AUC was calculated for each model, allowing the comparison of their classification performance.

Analysis of predicted and validated miRNA-mRNA interactions was performed with the freely available software MiRTargetLink 2.0© (https://www.ccb.uni-saarland.de/mirtargetlink2) [152]. Gene ontology (GO) enrichment analysis was performed using the miRNA tool incorporated into MiRTargetLink 2.0©, targets were retrieved, sorted by adjusted p-value, and presented in table format. Selected networks of mRNAs targeted by at least two miRNAs were drawn using Adobe Illustrator© software.

8.4 Results

Several ME/CFS PLSA-DA models were obtained and assessed using the data obtained from the 30 participants. However, it is important to mention that variables obtained from questionnaires were excluded in all models reported throughout this section since a diagnostic based solely on objective measurements was pursued.

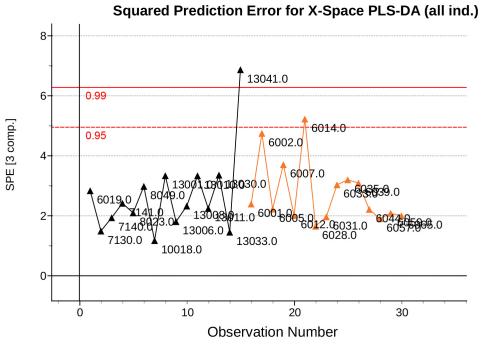


Figure 8.1: Squared Prediction Error (SPE) for the observations (i.e., patients) with the initial PLS (Partial Least Squares)-DA (Differential Analysis) multiblock model. Black triangles are healthy controls, whereas orange triangles are ME/CFS patients. The observation with ID 13041 is an example of an outlier over the SPE control limit (red lines).

8.4.1 PLS-DA model to classify ME/CFS identifies EV features as potential disease biomarkers

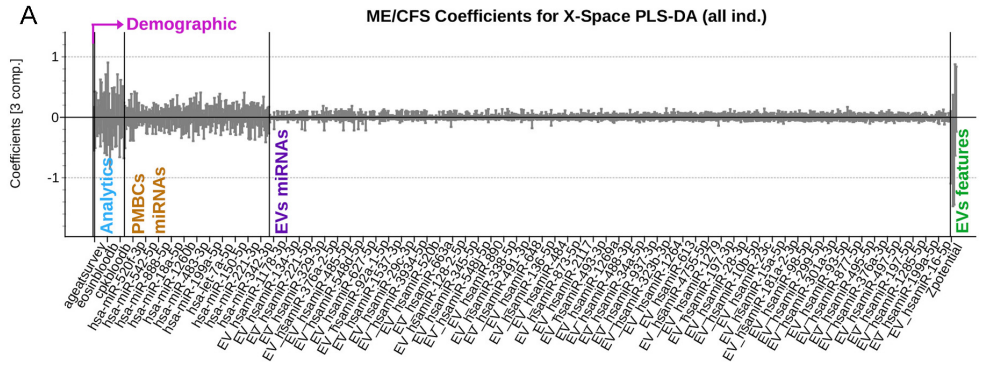
Given the small sample size of the cohort, this first PLS-DA modelling step focused on finding the most statistically significant biomarkers for identifying severe ME/CFS subjects. All observations (i.e., participants) were used to fit the model to reduce the uncertainty in estimating the model parameters as much as possible.

ME/FCS modelling with PLS-DA

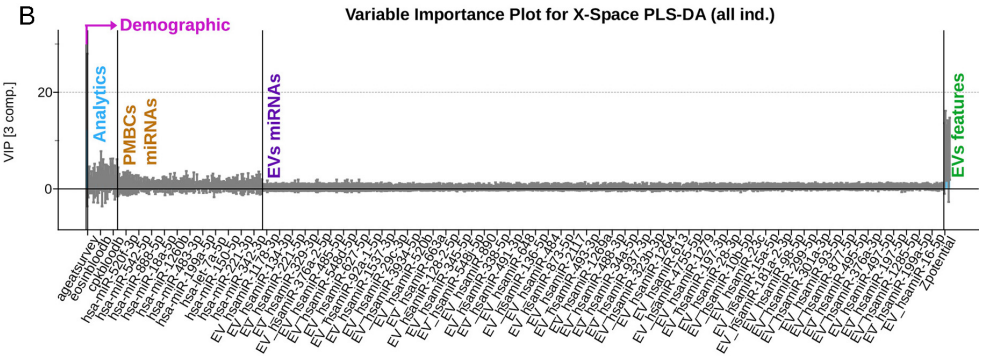
The initial model was fitted with three latent variables (obtained by cross-validation) with a cumulative value of 96% for the R^2 coefficient (goodness of fit) and 68% for the Q^2 coefficient (goodness of prediction). After obtaining the PLS-DA model, we checked for potential outliers, removing subjects with an SPE (i.e., Euclidean distance to the model) surpassing the control limit (an example of an outlier can be seen in Figure 8.1).

The initial PLS-DA model also presented a huge number of predictors having a VIP with a confidence interval clearly below 1 (Figure 8.2a) and non-statistically significant b coefficients (Figure 8.2b). Thus, after performing an iterative variable selection, as described in Methods.

This depurated PLS-DA model with 32 variables (Figures 8.3a and 8.3b) had similar cumulative R^2 and Q^2 values (98.71% and 96.31%, respectively), and



(a) Set of $\boldsymbol{B}_{PLS}$ coefficients.

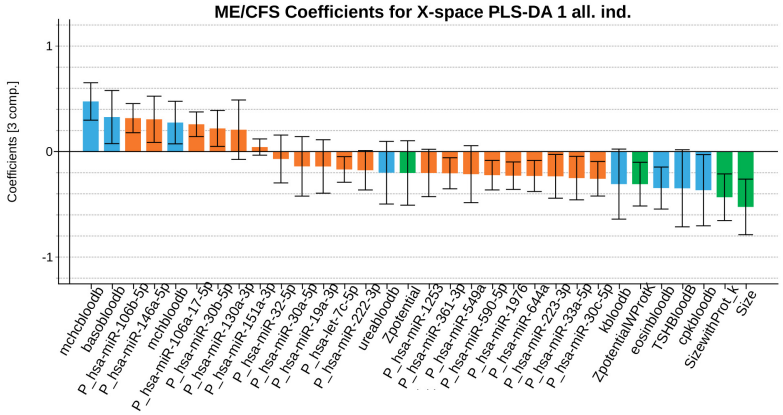


(b) Set of VIP PLS coefficients.

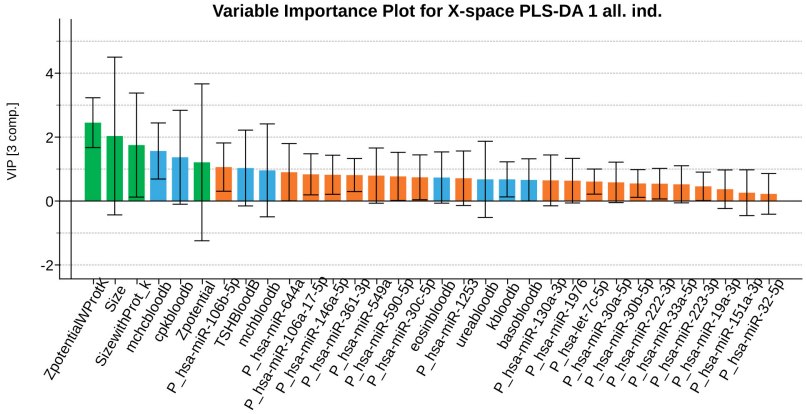
Figure 8.2: PLS-DA multiblock model based on all variables measured from 15 ME/CFS patients and 15 HCs.

the optimal number of components based on cross-validation was three (as the initial model). This was based on the final model with the most discriminant variables obtained a set of $N = 24$ observations, having 12 of each class.

The permutation test (Figure 8.4) showed that the R^2 and Q^2 values of the obtained PLS-DA model (points belonging to the 100% correlation between original y and permuted y) are greater than any of those belonging to the permuted datasets. Thus, the statistical significance of the 98.71% and 96.31% values for the R^2 and Q^2, respectively, is accepted, rejecting the hypothesis of obtaining these values by chance (with p-value < 0.05).



(a) Set of jackknife b coefficients for each variable and the class "Healthy Control" with its 95% jackknife confidence interval, obtained with the full data set. Each variable block is represented by one colour (demographic variables in blue, analytic variables in orange, PBMCs miRNAs variables in green, EVs miRNAs variables in purple, and EVs' characterization in pink).



(b) VIP coefficients of the predictor variables for the PLS-DA model with all the Set of predictors. The colour code is the same as in the b coefficients figure.

Figure 8.3: Partial Least Squares (PLS)-Discriminant Analysis (DA) multiblock model based on 32 variables measured from 12 ME/CFS patients and 12 HCs.

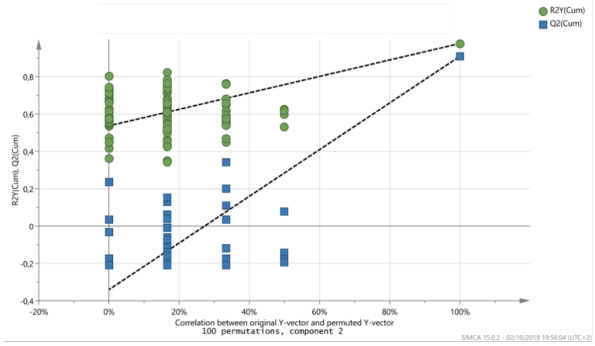


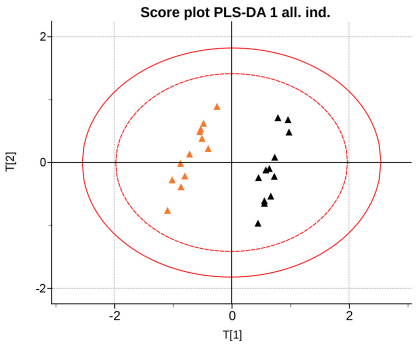
Figure 8.4: Permutation test for the variable filtered PLS-DA model. The values of the model coefficients are expressed in the vertical axis, whereas in abscissa, the correlation between the real response vector and the different permuted versions is expressed.

Additionally, the stability and reliability of the final PLS-DA model in terms of its prediction performance can be visualized both in the scores scatterplot (Figure 8.5a) and in the observed vs prediction plot (Figure 8.6a).

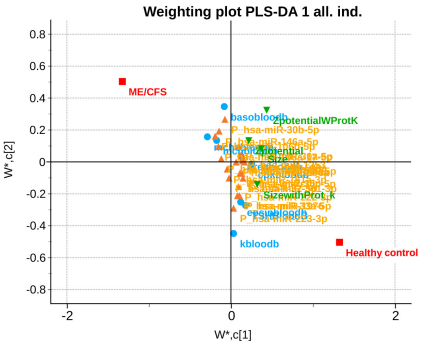
The score scatterplot (Figure 8.5a), showing a clear separation between groups, is directly related to the weighting plot (Figure 8.5b), which shows the correlation structure between the original and the latent variables. Thus, the probability of being a severe ME/CFS individual (orange triangle in the score scatterplot) is positively correlated with the variables at the same side (left) of the weighting plot, which are the same variables with a positive b coefficient for the CFS class. This means those variables tend to have greater CFS values than healthy individuals.

Analogously, the set of variables placed at the opposite semi-plane (right part) of the weighting plot (with negative b coefficients for the ME/CFS class) negatively correlates to the probability of belonging to the CFS class. This means these variables tend to have lower values in ME/CFS than in healthy individuals. Finally, variables near the origin (0,0) point are those with coefficients not statistically different from zero (i.e., no statistical differences in both groups of participants).

Finally, the observed vs. prediction results for the participants showed a class prediction with 95% confidence intervals (magenta lines) using just three components, allowing all 12 patient observations to be correctly classified in the ME/CFS group and all 12 observations from healthy subjects in the HC group (Figure 8.6a). The ROC curve of the model shows a perfect classification of



(a) Score plot. The colour code of observation groups is the same as in Figure 8.1.



(b) Weighting plot. The colour code of variable blocks is the same as in Figure 8.3.

Figure 8.5: Partial Least Squares (PLS)-Discriminant Analysis (DA) multiblock model based on 32 variables measured from 12 ME/CFS patients and 12 HCs. The axis corresponds to the 1st and 2nd components (horizontal and vertical, respectively).

the samples (Figure 8.6b) since the AUC for both classes reaches a value of 1. This means that the model has excellent sensitivity and specificity (equal to 1), i.e., it detects all patients and differentiates all controls as healthy individuals.

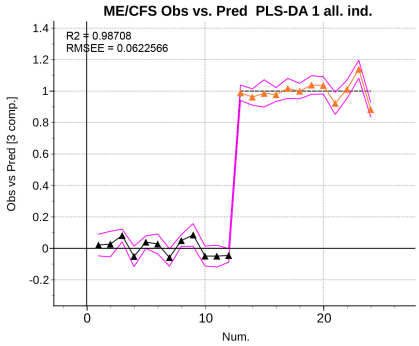
Classification performance of the PLS-DA model with calibration and validation set

The second modelling approach focused on evaluating the potential of our PLS-DA model as a tool to correctly assign new observations to ME/CFS and HC groups. As explained in the Methods section, the database was partitioned into training and validation subsets for this second PLS-DA model.

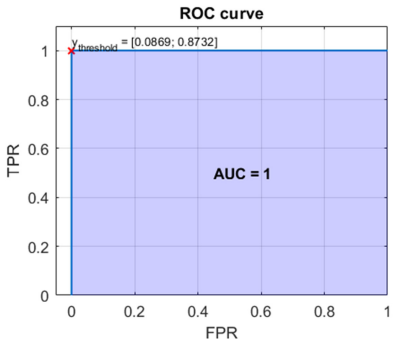
The trained model with three components (the same number as the previous model with all the observations) reaches cumulative values of 99.32% for the goodness of fitting coefficient (R^2) and 88.52% for the goodness of prediction coefficient (Q^2).

The b coefficients obtained are almost of the same order, according to their importance, but with wider confidence intervals (Figure 8.7a and 8.7b). This is caused by removing the validation samples from the training set, decreasing the sample size, and increasing model uncertainty.

Once the model is fitted, the observations of the validation set are projected onto the latent subspace, obtaining their corresponding scores and predictions



(a) Observed vs. Predicted values (with 95% confidence intervals as magenta lines) for participants using three components. Figure 8.1 shows the group colour code, and RMSEE stands for Root Mean Square Error of Estimation.



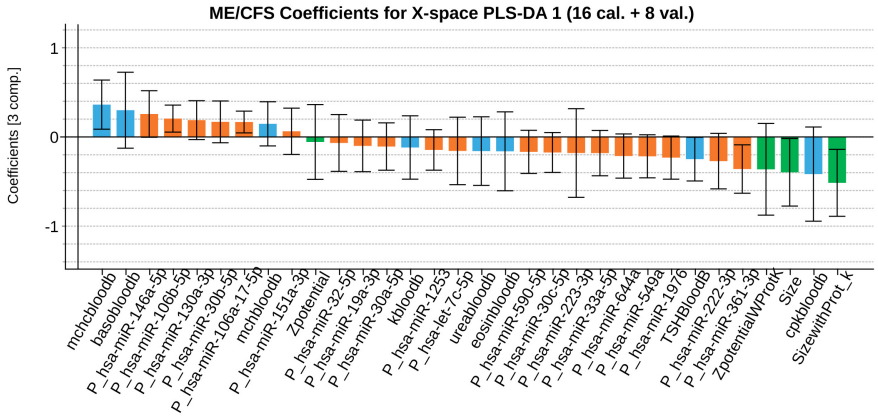
(b) ROC curve for the classification of participants. The red cross locates the optimal performance point (maximum specificity and sensitivity) using the classification threshold between 0.0869 and 0.8732.

Figure 8.6: Partial Least Squares (PLS)-Discriminant Analysis (DA) multiblock model based on 32 variables measured from 12 ME/CFS patients and 12 HCs

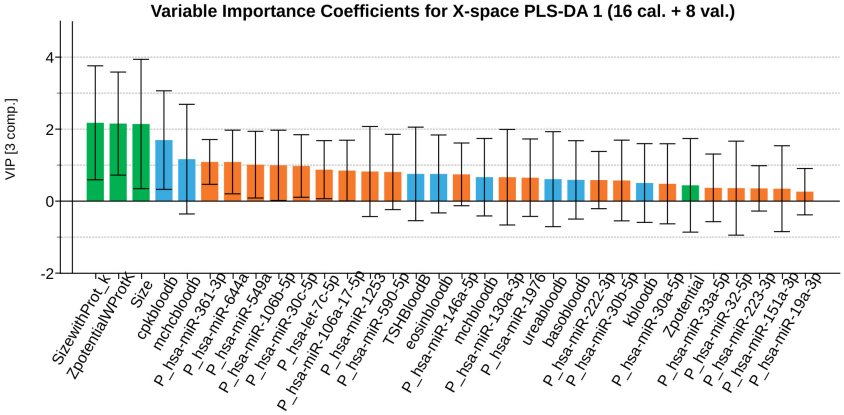
(Figure 8.8a and 8.8b). These results support the validity of the model developed in 3.1.1. for the diagnosis of severe ME/CFS patients.

The ROC curve for the validation samples (Figure 8.8c) shows perfect discrimination (AUC = 1) when the PLS-DA model is used to classify new individuals as healthy or those affected by severe ME/CFS. This means that the model maintains the perfect detection of ME/CFS patients (perfect sensitivity) while keeping the perfect discrimination of healthy controls (specificity = 1).

Intrigued by the fact that four out of the six physical associated parameters of EVs (EV concentration, size and z-potential obtained with or without proteinase K pretreatment), corresponding to the size and zeta potential of vesicles (as described in [137]) were discriminating features selected by our initial PLS-DA model (Figure 8.7), we decided to further explore the differential nature of ME/CFS EVs by Raman spectroscopy analysis. This approach has proven to differentiate EVs from various cell sources [153] and has been successfully used to detect ME/CFS-specific changes in PBMCs [154].

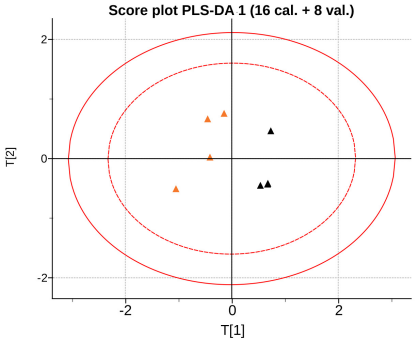


(a) ME/CFS class b jackknife coefficients for the $\boldsymbol{X}$ subspace using the calibration dataset. The colour code corresponds to the block to which each variable belongs, being those analytical variables (blue), PBMCs miRNAs (orange) and EVs characteristics (green). Jackknife confidence intervals were calculated at a 95% confidence level.

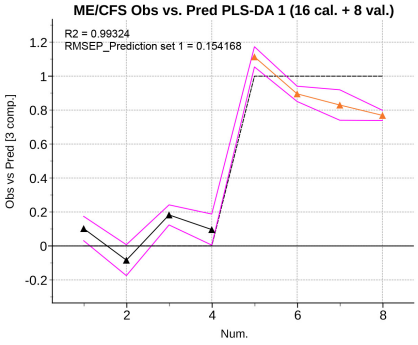


(b) VIP coefficients with jackknife confidence intervals at 95% of confidence for the $\boldsymbol{X}$ subspace using the calibration dataset. Data set legends can be consulted in Table 8.2. The colour code for each variable block is the same as in the rest of the figures.

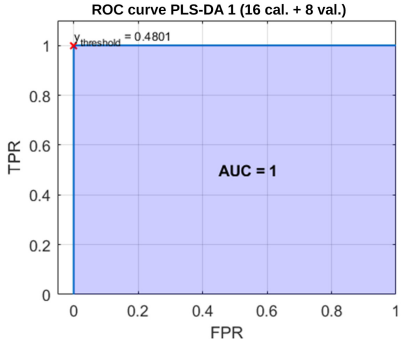
Figure 8.7: Partial Least Squares (PLS)-Discriminant Analysis (DA) multiblock model based on 32 variables fitted using only the patients from the training dataset.



(a) Score plot (1st and 2nd components) for the validation observations. For more information, see Figure 8.5a.



(b) Observed vs Prediction values for the validation observations. For more information, see Figure 8.6a.



(c) ROC curve for classifying the validation observations with the trained dataset. For more information, see Figure 8.6b.

Figure 8.8: Projection of the validation dataset on the Partial Least Squares (PLS)-Discriminant Analysis (DA) multiblock model from Figure 8.7.

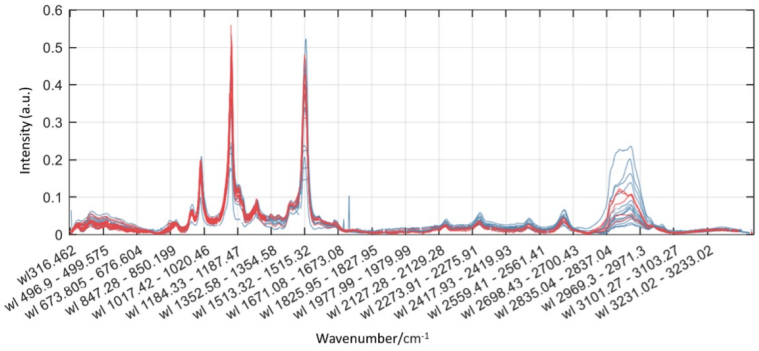


Figure 8.9: Complete Raman spectra of EVs isolated from plasma from 15 ME/CFS patients (red) and 15 matched control individuals (blue).

8.4.2 ME/CFS classification model based on Raman fingerprints

To further investigate the power of Raman spectroscopy to differentiate patients from controls, we again used PLS-DA as a classifier solely based on the whole Raman spectra. The complete spectra of individuals within each group are represented in Figure 8.9 (controls in blue and ME/CFS patients in red).

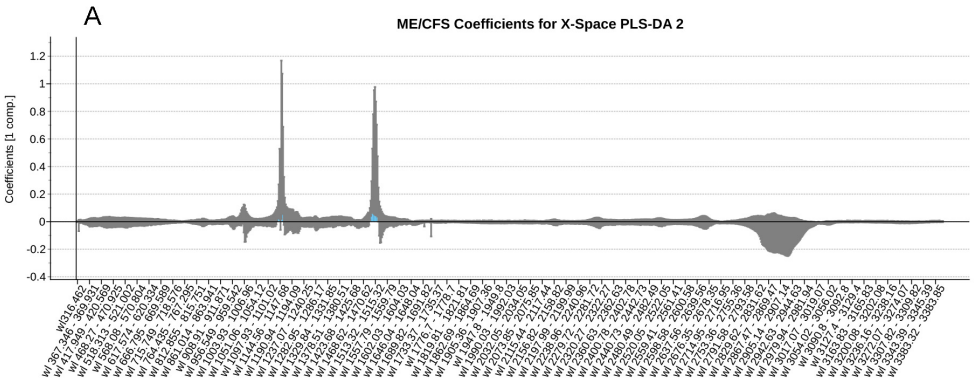
As can be appreciated, signals were already preprocessed and can be directly used for further analysis with multivariate statistics techniques. Due to a slight (though not relevant) mismatch in the wavelengths of different records, abscises axes in Figure 8.9 represent wavelength bins that contain the signal recorded for wavelengths within each interval. We also compared PLS-DA with a modified version of the LDA, RF and SVMs to evaluate if there were more suitable techniques to classify individuals using only the Raman spectra as an input.

PLS-DA model

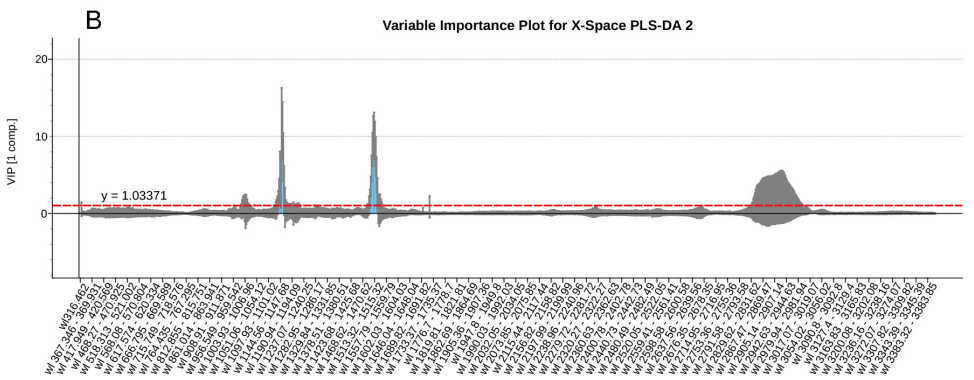
We applied PLS-DA analysis to Raman data to evaluate the biomarker value of the observed differential Raman peaks. The wavelength intervals with discriminant information should appear with significant $\boldsymbol{B}_{PLS}$ or VIP coefficients.

The first PLS-DA model (R^2 of 23.95% and Q^2 of 16.33%) was not able to separate the groups since many variables are non-statistically significant in terms of the $\boldsymbol{B}_{PLS}$ and VIP coefficients. This can be observed from the high number of jackknife confidence intervals for the VIPs below the $VIP = 1$

threshold (see Figure 8.10a) and by the jackknife confidence intervals for the b coefficients that contain a zero value (see Figure 8.10b).



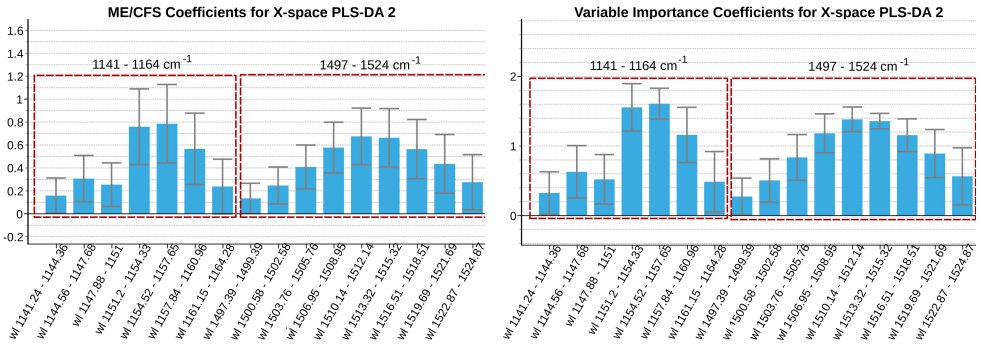
(a) Set of $\boldsymbol{B}_{PLS}$ coefficients.



(b) Set of VIP PLS coefficients.

Figure 8.10: PLS-DA multiblock model based on all variables measured from 15 ME/CFS patients and 15 HCs.

All non-significant variables according to these parameters were deleted, and the model re-estimated. The resulting model selects only one latent variable, slightly increasing its goodness of fit (R^2 of 29.57%) and prediction (Q^2 of 26.36%). Figure 8.11a and 8.11b display the $\boldsymbol{B}_{PLS}$ and VIP coefficients for predicting the ME/CFS class. Variables with positive b coefficients indicate



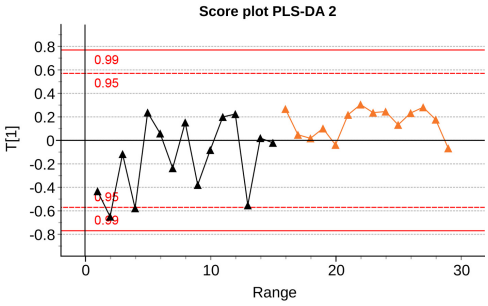
(a) Summary of the b PLS coefficients. **(b)** Summary of the VIP PLS coefficients.

Figure 8.11: Summary of the depurated PLS-DA model with the Raman spectroscopy data. The red cross locates the optimal performance point(maximum specificity and sensitivity) using the classification threshold 0.3935. Data set legends can be consulted in Table 8.2. Black triangles represent healthy controls, whereas orange triangles represent ME/CFS cases.

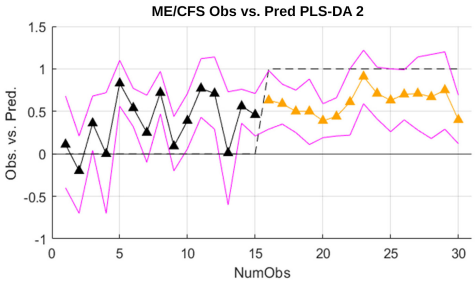
wavelengths of the spectrum for which the ME/CFS patients show a statistically significant higher signal when compared to the movement of the healthy controls.

According to the $\boldsymbol{B}_{PLS}$ coefficients, relevant variables highlight the importance of regions close to the $1158cm^{-1}$ peak. On the other hand, the right window encloses wavelengths relative to the $1521cm^{-1}$ peak. These bands are characteristic of carotenoids with the C–C stretching mode (coupled with C–H in-plane bending) contributing to the $1158cm^{-1}$ band and the $C = C$ stretching mode of the conjugated chain in carotenoids contributing to the $1510cm^{-1}$ band [155]. A later univariate test for these two bands indicated a statistically significantly higher content of carotenoids in ME patients than in healthy controls ($p = 0.003$ and $p = 0.005$).

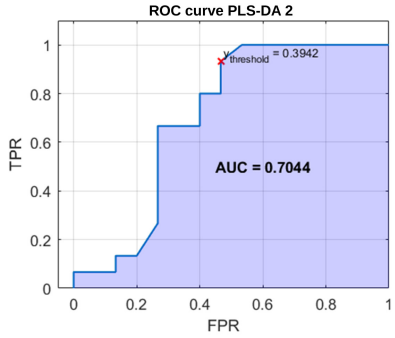
The classification performance of the depurated PLS-DA model (Figures 8.11 and 8.12) is illustrated in the observed vs. predicted values (Figure 8.12b) and in its corresponding ROC curve generated using the 3-fold cross-validation scheme (Figure 8.12c). The model reaches an optimal AUC value of 0.7044, setting a threshold of 0.3942 on the predicted response. Despite the poor performance of the model in terms of classification, there might still be statistically significant information that could be useful in discriminating the two groups.



(a) Summary of the scores. For more information, see Figure 8.5a.



(b) Observed vs predicted values. For more information, see Figure 8.6a.



(c) ROC curve. For more information, see Figure 8.6b.

Figure 8.12: Summary of the depurated PLS-DA model with the Raman spectroscopy data. The red cross locates the optimal performance point(maximum specificity and sensitivity) using the classification threshold 0.3935. Data set legends can be consulted in Table 8.2. Black triangles represent healthy controls, whereas orange triangles represent ME/CFS cases.

Comparison of PLS-DA model to other classification models

To further investigate the value of the Raman spectra in differentiating severe ME/CFS patients from healthy controls, we trained three other binary classification models. We used an adaptation of linear discriminant analysis (LDA) for cases with more variables than observations, a random forest (RF), and a support vector machine (SVM). Some of these techniques (such as RF and SVMs) can model non-linearities, which could improve the outcome yielded by the PLS-DA model.

The classification results were also obtained in MATLAB, training the classifiers with the classification learner app and optimizing model hyperparameters to ensure a fair comparison to the already optimized PLS-DA model. The same 3-fold cross-validation setup for the PLS-DA model was used to compare results. This lets us preserve 2/3 of the data for the training, leaving the other 1/3 of observations for external validation. Moreover, all words had a prediction obtained without using them for the model fitting.

Some classifiers offer the interpretation of the discriminant predictors to a certain extent. However, this was not feasible for all of them, as we address in the following points, providing some details about certain technical aspects considered for each classifier:

- **Linear Discriminant Analysis**. This model was run in Matlab using an algorithm that adapts the classical LDA to deal with more variables than observations. The LDA model was fitted using all the predictors ($K = 1018$ variables). One mathematical aspect of the LDA is that it needs to invert the covariance matrix of the predictors. This step is compromised when the number of variables is higher than the number of observations, as in this case (1018 variables >> 30 observations). Nonetheless, numerical solutions are implemented to enable a solution's obtention. However, this can come at the cost of losing coherence in the model's coefficients. In fact, as for the PLS-DA, a previous depuration step based on the coefficients of the discriminant function was considered. However, as seen in Figure 8.13, there was too much numerical instability to perform this depuration. This can be appreciated by the high number of coefficients containing the zero value, which means that depending on the fitting round, these coefficients could be either positive or negative. The error bars indicate the coefficient's variation range, i.e., its minimum and maximum, along the three cross-validation folds.

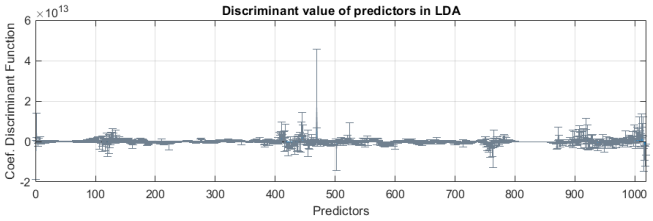


Figure 8.13: Coefficients of the predictors in the discriminant functions fitted for each fold of the cross-validation scheme used for the Raman spectra classifiers.

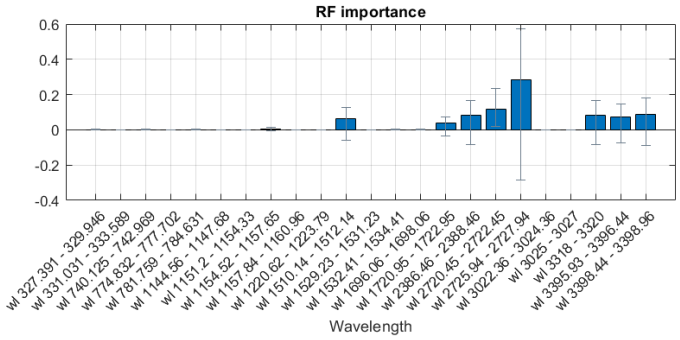


Figure 8.14: Variable importance metrics obtained for the Random Forest model based on the Raman spectra.

- **Random Forest**. The RF model was fitted using all the predictors ($K = 1018$ variables). The MATLAB classification learner app performed the hyperparameters optimization. This optimization involves finding the model configuration that minimizes the cross-validated misclassification rate. In this case, the Variable Importance metrics could also be used to refine the model or to know which variables hold more discriminant power. In this case, variable importance is measured by a permutation test. This test randomly shuffles the values of a given variable and measures the difference in the classification error due to altering that variable. Figure 8.14 illustrates the variable importance for predictors whose importance metrics were above zero over the three cross-validation folds. The fact that they present a positive average importance means that their permutation causes a detriment to the model performance. As it can be seen, some variables selected indicate the importance of wavelengths close to the 1158 cm^{-1} peak and to the 1521 cm^{-1} peak. However, not all the important variables agree with those selected by the PLS-DA

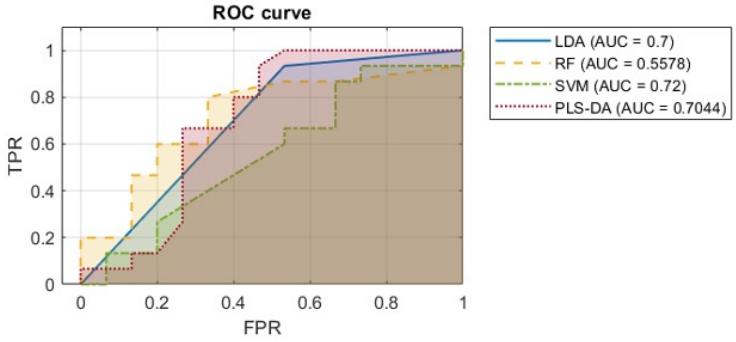


Figure 8.15: ROC curves with their AUCs of the four models classifying ME or HC based on their Raman spectra. The ROC curve is plotted with a true positive rate against a false positive rate.

model, and the meaning of their relevance, according to the RF model, is beyond further study and understanding. However, the fact that their minimum and maximum values (limits of the error bars) have different signs suggests a lack of coherence in the predictors' importance and that their relevance might not be statistically significant.

- **Support Vector Machine**. The SVM model was fitted using all the hyperparameters' optimization of the MATLAB classification learner app. This optimization involves finding the model configuration that minimizes the cross-validated misclassification rate. The optimizer selected a linear kernel function for the first fold, a Gaussian kernel function for the second fold and a polynomial kernel function for the third fold. This incoherence to the optimal SVM suggests a lack of information within the Raman spectra to build a stable and reliable classifier aligned with the conclusions obtained with the rest of the classifiers.

Figure 8.15 shows all ROC curves with their respective AUCs for all four methods. These results suggest that the Raman spectroscopy data does not hold enough information to discriminate between ME/CFS patients and healthy subjects: to achieve a 100% true positive rate, classifiers would produce a high rate of false positives. However, AUC values close to 0.7 (Figure 8.15) suggest that EVs might still represent part of the disease phenotype. For this reason, we proposed the last model, combining our initial biomarkers and EV Raman profiles.

8.4.3 Refinement of the initial PLS-DA model with EV Raman profiles

The results of Raman spectrometry analysis show that further information is required to develop a more comprehensive diagnostic tool. Therefore, we proceeded to reanalyze our first multiblock PLS-DA model (Section 8.4.1) to check if the relevant Raman wavelengths selected by the PLS-DA model on the spectroscopy data (Section 8.4.2) could be helpful to predictors when combined with the previously identified biomarkers.

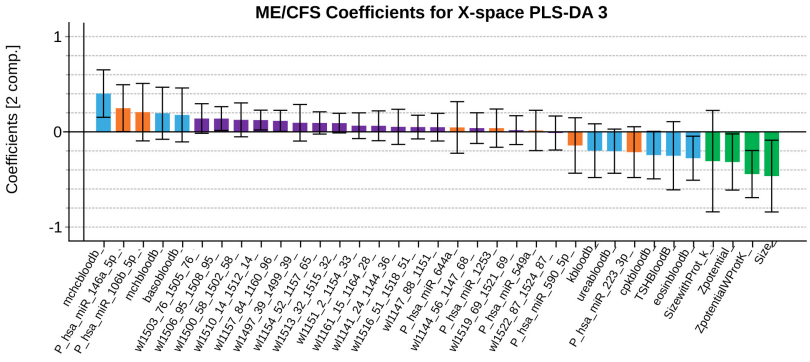
To study this possibility, we fitted a PLS-DA model using the selected variables from the former PLS-DA model, adding the key differential wavelengths from our PLS-DA analysis of Raman spectroscopy data. It is important to highlight that the adequacy of this approach resides in the fact that the samples used to generate the two models came from the same blood samples. The reason for maintaining the use of PLS-DA was that, according to the previous results, it was a technique yielding one of the best classification performances and the only one enabling the interpretation of the discriminant power of the predictors, establishing a set of statistically significant biomarkers.

An initial PLS-DA model was fitted using all observations to allow for selecting key discriminating variables and removing potential outliers. The initial fused model sets an optimal number of nine latent variables (R^2 of 99.37% and Q^2 of 81.15%). This model was depurated observation-wise and variable-wise, as previously described. The b coefficients and VIP coefficients of the final set of selected variables are shown in Figure 8.16a and 8.16b, respectively.

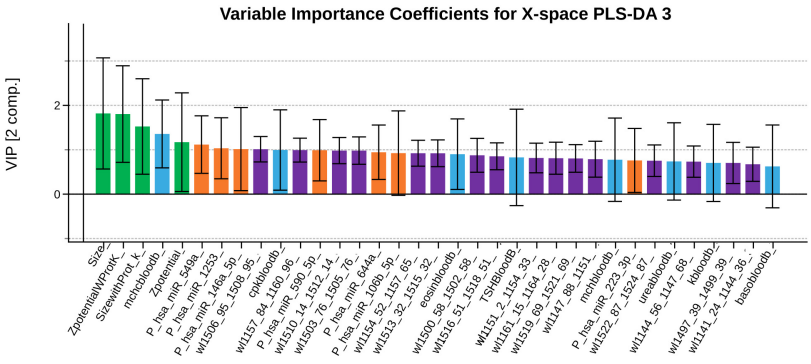
This refined PLS-DA model was fitted based on the final set of selected predictors, excluding the observations used for external validation in the first PLS-DA model. The final model presents a similar performance (R^2 of 93.38% and Q^2 of 77.06%). Figure 8.17 shows the result of the permutation test performed on the PLS-DA model fitted with the calibration set, proving the statistical significance of the yielded coefficients.

The observed vs. predicted values for the observations in the calibration set (Figure 8.18a) and the external validation set show that classes can be perfectly separated (Figure 8.18c). The ROC curves in Figure 8.18b and 8.18d also illustrate this, showing that a threshold on the predicted outcome of 0.481 yields a perfect classification with an AUC of 1.

Inspecting the b PLS and VIP coefficients (Figure 8.16a and 8.16b, respectively), although some of the predictors still appear as statistically non-significant, their jackknife confident intervals are almost under or above zero for the b co-



(a) Summary of the $\boldsymbol{B}_{PLS}$ coefficients.



(b) Summary of the VIP PLS coefficients.

Figure 8.16: Summary of the depurated PLS-DA model with the Raman spectroscopy data. Data set legends can be consulted on Table 8.2. Black triangles represent HCs, whereas orange triangles represent ME/CFS patients. Predictor coefficients in (A, B) are coloured according to their information block (blue for analytical features, orange for PBMCs miRs features, green for EVs features, and purple for Raman spectra features).

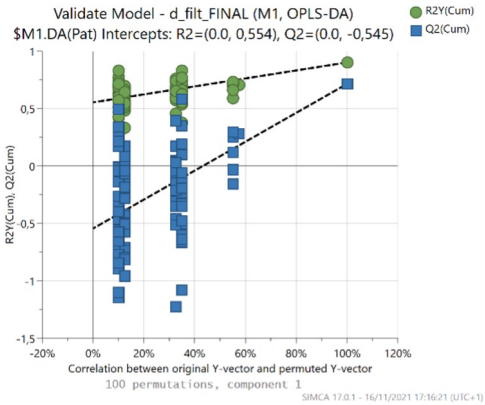
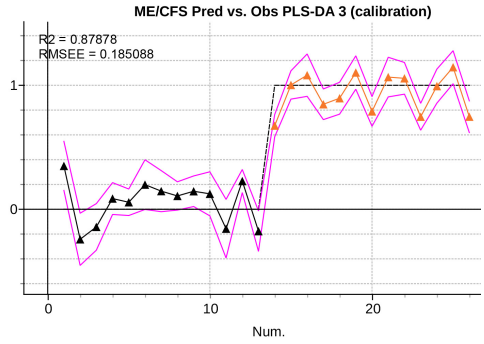


Figure 8.17: Permutation test for the depurated PLS-DA model based on the fused database. The values of the model coefficients are expressed in the vertical axis, whereas in abscissa, the correlation between the real response vector and the different permuted versions is expressed.

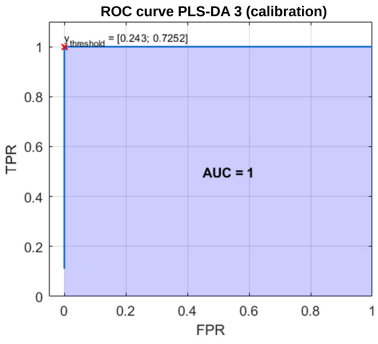
efficients, or almost contain the value $VIP = 1$ for the VIP coefficients. This suggests that the width of the confidence intervals might be influenced by the small sample size, which leads to wide jackknife confidence intervals.

In conclusion, this final model yields a perfect classification (AUC = 1) and has 35 predictors, meaning that some of the most relevant predictors, according to the previous PLS-DA model, have been replaced by wavelength intervals of the Raman spectroscopy analysis. Among these relevant wavelengths, both peaks (around 1158 cm^{-1} and 1521 cm^{-1}) hold important information as potential biomarkers. Most eliminated predictors from the previous PLS-DA model carried information about PBMC miRNAs.

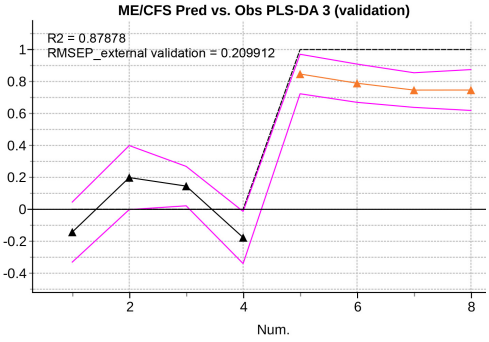
A posterior GO pathway analysis of DE miRNAs from PBMCs selected by our refined PLS-DA model (Figure 8.16) showed that six out of seven common share gene targets with top cellular functions belonging to immunity, neuroinflammation, and metabolism (Table 8.1), all being widely associated with ME/CFS in the literature.



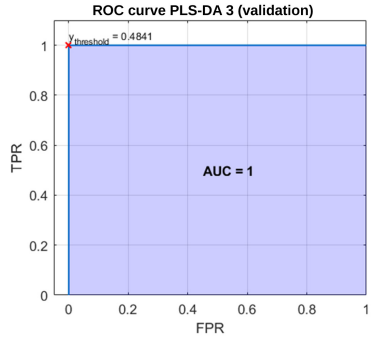
(a) Observed vs. predicted values for the training set. For more information, see Figure 8.6a.



(b) ROC curve for the training set. For more information, see Figure 8.6b.



(c) Observed vs. predicted values for the validation set. For more information, see Figure 8.6a.



(d) ROC curve for the validation set. For more information, see Figure 8.6b.

Figure 8.18: Summary of the depurated PLS-DA model with the Raman spectroscopy data. Data set legends can be consulted on Table 8.2. Black triangles represent HCs, whereas orange triangles represent ME/CFS patients. Predictor coefficients in (A, B) are coloured according to their information block (blue for analytical features, orange for PBMCs miRs features, green for EVs features, and purple for Raman spectra features).

Table 8.1: Top GO categories containing gene targets of at least 2 DE discriminant miRNAs from PMBCs and which were relevant according to the refined PLS-DA model.

GO Subcategory	P-adjusted	Q-adjusted	miRNAs/precursors
positive regulation of T cell-mediated immunity	2,60E-03	2,60E-03	hsa-miR-223-3p, hsa-miR-146a-5p
positive regulation of neuroinflammatory response	2,60E-03	2,60E-03	hsa-miR-223-3p, hsa-miR-146a-5p
positive regulation of type 2 immune response 2,60E-03 2,60E-03	hsa-miR-223-3p, hsa-miR-146a-5p		
adaptive immune response	5,18E-03	5,18E-03	hsa-miR-223-3p, hsa-miR-146a-5p
positive regulation of adaptive immune response	5,18E-03	5,18E-03	hsa-miR-223-3p, hsa-miR-146a-5p
adaptive immune response based on somatic recombination of immune receptors built from immunoglobulin superfamily domains	5,84E-03	5,84E-03	hsa-miR-223-3p, hsa-miR-146a-5p
neuroinflammatory response	5,84E-03	5,84E-03	hsa-miR-223-3p, hsa-miR-146a-5p
regulation of adaptive immune response	5,84E-03	5,84E-03	hsa-miR-223-3p, hsa-miR-146a-5p
regulation of neuroinflammatory response	5,84E-03	5,84E-03	hsa-miR-223-3p, hsa-miR-146a-5p
regulation of reactive oxygen species metabolic process	7,08E-03	7,08E-03	hsa-miR-590-5p, hsa-miR-223-3p, hsa-miR-146a-5p
regulation of carbohydrate metabolic process by regulation of transcription from RNA polymerase II promoter	1,44E-02	1,44E-02	hsa-miR-106b-5p, hsa-miR-223-3p

8.5 Conclusions

In 2015, the Institute of Medicine (IOM) in the US informed that ME/CFS is a medical illness and should not be considered a psychiatric condition [156]. Despite numerous studies supporting the biological basis of ME/CFS by reporting neurologic, [157], immune [158] and metabolic [159] disturbances, ME/CFS biomarker validation remains a significant challenge. Still, the low number of participants and disease heterogeneity hamper the search for biomarkers. Almenar-Pérez et al. [137] attempted to improve patient homogeneity by restricting the inclusion of participants to only severe female cases. This resulted in a large dataset with 34 blood analytic variables, 775 different miRNAs expressed above threshold levels (136 in PBMCs and 639 in EVs), EV concentra-

tion, size, and z-potential. Although limited by the small sample size and the use of univariate statistical tools, results suggested some biological differences with limited diagnostic potential at the individual level. These set the basis for searching for more appropriate tools to analyze this data.

In the current study, we combine these variables and add Raman spectroscopic profiling as a new marker of EV function in the same blood samples. By applying PLS-DA analysis to this large dataset, we identified 32 variables that can effectively differentiate ME/CFS cases from healthy controls (AUC = 1, i.e., sensitivity and specificity = 1, Figure 8.8c).

Moreover, the PLS-DA models also helped to assess the initial hypothesis about the role of EVs. Some EV physical features, including their size and z-potential, were relevant for the effective diagnosis of patients, indicating a potentially important role of EVs in ME/CFS. This brought interesting points for a biomedical discussion.

On the one hand, increased absolute zeta potential values of EVs detected in ME/CFS patients by previous studies [137] suggested differences in the relative abundance of charged groups in their membranes. Modifications of EVs membrane potential have been related to other pathological conditions, including cancer, where the change in EV net charge was attributed to a disbalance in the relative abundance of sialic acid [160]. Interestingly, polysialylation of exosomal membranes has been shown to have a thermo-protecting effect, modulating exosome-plasma membrane interactions and thus their signalling capacity [161]. Further evaluation of these modifications present in ME/CFS, EVs, will be essential to future studies on their functional impacts, as proposed in [162].

On the other hand, results differ from previous works reporting higher counts of EVs in different cohorts of ME/CFS patients [137], [163], [164]. However, the fact that increased EV numbers have been reported for other diseases with an inflammatory component [165], [166] may argue for a restricted disease specificity of this feature.

To unveil potential differences in EVs' composition supporting the reduced diameter zeta potential (increased electronegativity) seen in severe ME/CFS patients [137], EVs from ME/CFS patients were compared to control EVs by Raman micro-spectroscopic analysis. Since Raman spectroscopy has shown its utility in detecting composition differences in patients' EVs [19], [20], it could be developed as a cost-effective diagnostic method with its ability to identify complex patterns in biological materials.

The analysis of EVs Raman spectra showed differences between severe ME/CFS patients and healthy controls related to two carotenoid peaks (Figure 8.11). Zhang et al. recently found a shift of a peak at $1553cm^{-1}$ (tryptophan/amide II) to $1528cm^{-1}$ (carotenoid) in trophoblast-derived EVs during late stages of pregnancy [167], time at which circulating EVs counts increase and inflammatory responses vary [168], [169]. Verma and Wallach [170] already described a relationship between carotenoids and red blood cells (RBC) hemolysis [171] typically linked to disease. Moreover, recent studies have shown reduced RBC deformability in ME/CFS patients [172]. Thus, it is tempting to speculate that the EVs observed by Raman are due to EVs of RBC origin, generated when the RBC is stressed in the patient's circulation.

In support of this hypothesis, it is interesting to observe that increased mean corpuscular haemoglobin (labelled *mch*) and means corpuscular haemoglobin concentration (labelled *mchc*), which have been related as well to decreased deformability of RBCs [173], and were identified by the PLS-DA analysis as discriminant variables (Figure 8.16). Moreover, Fiedor et al. have recently shown that increased beta-carotene concentration in RBC membranes affects cell shape and sensitivity to osmolysis and alters haemoglobin-oxygen affinity with potential physiologic implications [174].

Regardless of EV composition differences described by Raman spectra analysis, the diagnostic value of Raman data seemed limited when compared to the rest of PLS-DA models, including analytic variables, PBMC miRNA profiles, and EV features (Figures 8.6b and 8.18d).

Nonetheless, purifying EVs from plasma may lead to the purification of EV sets that may differ from other procedures. Despite the high purity attributed to EVs prepared by ultracentrifugation, this procedure is laborious and requires a large volume of fluid and expensive equipment. A diagnostic method based on EVs requires a much simpler way, preferably allowing the analysis of small volumes of fluids without compromising performance. Total Exosome Isolation Reagent (TEIR) was selected from the available kits because, according to Helwa et al., it provides higher yields using smaller amounts of plasma when compared to other commercial alternatives or concerning ultracentrifugation, ultrafiltration, or gel chromatography [175]. Moreover, exploratory EV studies using highly purified EV sets (i.e., exosomes) could turn into missing relevant EV subsets, and thus, a less restrictive method was preferred.

Among the blood analytic group of variables, blood creatine phosphokinase (CK, labelled as *cpkbloodb*) level was a feature reported as significant by first and third PLS-DA models (Figures 8.3 and 8.16). This finding is aligned with

CK levels being a clinical feature previously reported as a potential biomarker of ME/CFS, showing significantly reduced levels in an expanded cohort of patients [176]. The CK enzyme is key in ATP homeostasis in these compounds, highly expressed in muscle, heart, and brain. Hence, low levels might reflect energy dysregulation in these tissues and may be linked to the profound fatigue found in ME/CFS patients, with the severe having the lowest CK levels.

Moreover, all miRNAs holding discriminant power came from the PBMCs group but not from the EVs one. This may be associated with the complexity of ME/CFS, requiring features from different compartments for its definition. In support of this argument, a later GO pathway analysis of six out of the seven DE miRNAs from PBMCs selected by the PLS-DA model (Figure 8.16) pointed out that those shared common gene targets with top cellular functions belonging to immunity, neuroinflammation, and metabolism, all being widely associated with ME/CFS in the literature.

In summary, this work describes for the first time an ME/CFS model based on PLS-DA of 32 analytical variables capable of diagnosing the disease with perfect sensitivity and specificity (AUC = 1), further confirming the biologic nature of this disease and highlighting the relevance of patient EV features for their diagnosis. An ME/CFS EV Raman spectroscopic fingerprint is also provided, pioneering the potential use of this method for diagnosing ME/CFS and detecting possible RBC defects in severe ME/CFS.

Finally, we show that although the diagnostic potential of Raman is limited, its simplicity and low sample requirement highlight its potential utility as an early screening tool before more comprehensive testing with miRNAs from PBMCs. Moreover, the inclusion of Raman data for the refinement of our previous model, although incapable of increasing the already perfect separation of cases from controls (AUC=1) (Figures 8.6b and 8.18d), allowed for a significant reduction in the number of PBMC miRNAs from 21 in our initial PLS-DA model (Figures 8.3 and 8.7) to only 7 in the PLS-DA Raman refined model (Figure 8.16).

The findings obtained in this study are expected to pave the way for unravelling the subjacent disease mechanisms in which EVs and PBMC miRNAs participate with clear implications for the future diagnosis and treatment of ME/CFS, perhaps embracing other patient groups suffering from chronic fatigue.

Appendix: Supplementary tables

Table 8.2: Description of variables from Table 8.3

Variable name	Variable description
Z potential w/Prot K	Z-potential (mV) of EVs obtained with Proteinase K pretreatment
Size	EV average size (diameter in nm) of EVs obtained in the absence of Proteinase K pretreatment
Size w/Prot K	EV average size (diameter in nm) of EVs obtained with Proteinase K pretreatment
mchcbloodb	Mean corpuscular haemoglobin concentration (g/L) - baseline
cpkbloodb	Creatine phosphokinase (U/L) - baseline
Z potential	Z-potential of EVs obtained in the absence of Proteinase K pretreatment
P_hsa-miR-106b-5p	PBMC hsa.miR
TSHBloodB	TSH (Thyroid-stimulating hormone) (mU/L) - baseline
mchbloodb	Mean corpuscular haemoglobin (pg) - baseline
P_hsa-miR-644a	PBMC hsa.miR
P_hsa-miR-106a-17-5p	PBMC hsa.miR
P_hsa-miR-146a-5p	PBMC hsa.miR
P_hsa-miR-361-3p	PBMC hsa.miR
P_hsa-miR-549a	PBMC hsa.miR
P_hsa-miR-590-5p	PBMC hsa.miR
P_hsa-miR-30c-5p	PBMC hsa.miR
eosinbloodb	Eosinophils$(10^9/L) - baseline$
P_hsa-miR-1253	PBMC hsa.miR
ureabloodb	Urea(mmol/L) - baseline
kbloodb	Potassium(mmol/L) - baseline
basobloodb	Basophils$(10^9/L) - baseline$
P_hsa-miR-130a-3p	PBMC hsa.miR
P_hsa-miR-1976	PBMC hsa.miR
P_hsa-let-7c-5p	PBMC hsa.miR
P_hsa-miR-30a-5p	PBMC hsa.miR
P_hsa-miR-30b-5p	PBMC hsa.miR
P_hsa-miR-222-3p	PBMC hsa.miR
P_hsa-miR-33a-5p	PBMC hsa.miR
P_hsa-miR-223-3p	PBMC hsa.miR
P_hsa-miR-19a-3p	PBMC hsa.miR
P_hsa-miR-151a-3p	PBMC hsa.miR
P_hsa-miR-32-5p	PBMC hsa.miR
wl 1506.95-1508.95	EV Raman peak wavelentgh range (nm)
wl 1503.76-1505.76	EV Raman peak wavelentgh range (nm)
wl 1510.14-1512.14	EV Raman peak wavelentgh range (nm)
wl 1157.84-1160.96	EV Raman peak wavelentgh range (nm)
wl 1500.58-1502.58	EV Raman peak wavelentgh range (nm)
wl 1154.52-1157.65	EV Raman peak wavelentgh range (nm)
wl 1513.32-1515.32	EV Raman peak wavelentgh range (nm)
wl 1151.2-1154.33	EV Raman peak wavelentgh range (nm)

wl 1497.39-1499.39	EV Raman peak wavelentgh range (nm)
wl 2058.94-2060.9	EV Raman peak wavelentgh range (nm)

Table 8.3: Relation of PLSDA variables sorted by descending relevance.

PLSDA w/out Raman differential peaks	**PLSDA w/ Raman differential peaks**
Z potential w/Prot K	Size
Size	Z potential w/Prot K
Size w/Prot K	Size w/ Prot K
mchcbloodb	mchcbloodb
cpkbloodb	Zpotential
Z potential	P_hsa-miR-549a
P_hsa-miR-106b-5p	P_hsa-miR-1253
TSHBloodB	P_hsa-miR-146a-5p
mchbloodb	wl 1506.95 - 1508.95
P_hsa-miR-644a	cpkbloodb
P_hsa-miR-106a-17-5p	wl 1157.84 - 1160.96
P_hsa-miR-146a-5p	P_hsa-miR-590-5p
P_hsa-miR-361-3p	wl 1510.14 - 1512.14
P_hsa-miR-549a	wl 1503.76 - 1505.76
P_hsa-miR-590-5p	P_hsa-miR-644a
P_hsa-miR-30c-5p	P_hsa-miR-106b-5p
eosinbloodb	wl 1154.52 - 1157.65
P_hsa-miR-1253	wl 1513.32 - 1515.32
ureabloodb	eosinbloodb
kbloodb	wl 1500.58 - 1502.58
basobloodb	wl 1516.51 - 1518.51
P_hsa-miR-130a-3p	TSHBloodB
P_hsa-miR-1976	wl 1151.2 - 1154.33
P_hsa-let-7c-5p	wl 1161.15 - 1164.28
P_hsa-miR-30a-5p	wl 1519.69 - 1521.69
P_hsa-miR-30b-5p	wl 1147.88 - 1151
P_hsa-miR-222-3p	mchbloodb
P_hsa-miR-33a-5p	P_hsa-miR-223-3p
P_hsa-miR-223-3p	wl 1522.87 - 1524.87
P_hsa-miR-19a-3p	ureabloodb
P_hsa-miR-151a-3p	wl 1144.56 - 1147.68
P_hsa-miR-32-5p	kbloodb
	wl 1497.39 - 1499.39
	wl 1141.24 - 1144.36
	basobloodb

Chapter 9

Mortality risk model for covid-19 patients

Part of the content of this chapter has been included in:

> [177] González-Cebrián, A., Borràs-Ferrís, J., Ordovás-Baines, J.P., Hermenegildo-Caudevilla, M., Climente-Marti, M., Tarazona, S., Vitale, R., Palací-López, D., Sierra-Sánchez, J.F., Saez de la Fuente, J. & Ferrer, A. Machine-learning-derived predictive score for early estimation of COVID-19 mortality risk in hospitalized patients. *PLOS ONE*. **17**, (2022), `https://doi.org/10.1371/journal.pone.0274171.https://doi.org/10.1016/j.chemolab.2021.104301`.

9.1 Introduction

The pandemic produced by the SARS-CoV-2 virus in 2020 - 2022 has caused to date (Aug 2022) more than 560 million infections and more than six million deaths worldwide, already ranking in many countries as one of the three main causes of death. In Spain, one of the European countries most affected by this pandemic, there have been over 13 million infections and more than 109,000 deaths [178].

The clinical course of COVID-19 is highly variable, and although most infected patients suffer minor flu symptoms, 10% - 20% of them require hospitalization (mainly due to the development of bilateral pneumonia and hypoxemia), and 10-15% of these develop a severe respiratory illness requiring mechanical ventilation or ICU admission, which increases the risk of death [179]. Progression to severe disease appears to be linked to damages to organs other than the respiratory tract that occur through an organic inflammatory syndrome possibly related to massive cytokines release [180].

In the clinical setting, it is essential to predict the severity level of the disease in a COVID-19 patient admitted to the hospital, both from the individual point of view and what concerns potential health system collapses, whose prevention requires decisions about patient management with appropriate triage criteria. This prediction involves identifying the contributing factors of mortality, which enables the adoption of targeted strategies in high-risk patients [181]. Most therapies (monoclonal antibodies, remdesivir, molnupiravir, specific protease inhibitors, etc.) that could improve the prognosis of this disease are usefully applied early, within the first days after the appearance of symptoms. Therefore, early identification of the risk of death from COVID-19 can be critical.

Several researchers have published observational prognostic studies on COVID-19 patients to identify predictive variables of death or severity of illness. However, later works have highlighted the need for a more precise statistical assessment of these types of studies, ensuring statistical coherence and the prevention of bias in finally proposed models [182], [183].

The objectives of this study are i) to determine key predictors of mortality in adult patients admitted to the hospital with a diagnosis of SARS-CoV-2 infection, ii) to obtain a predictive model of mortality for these patients, and iii) to propose a reliable and easy-to-use mortality risk score that can be calculated readily and straightforwardly at hospital admission.

9.2 Methods

The work aimed to build a model that, provided a set of variables recorded at the hospital admission, could predict the mortality risk of a patient with COVID-19 during admission and until 42 days following hospital discharge. The methodology for model training, evaluation and comparison is illustrated in Figure 9.1.

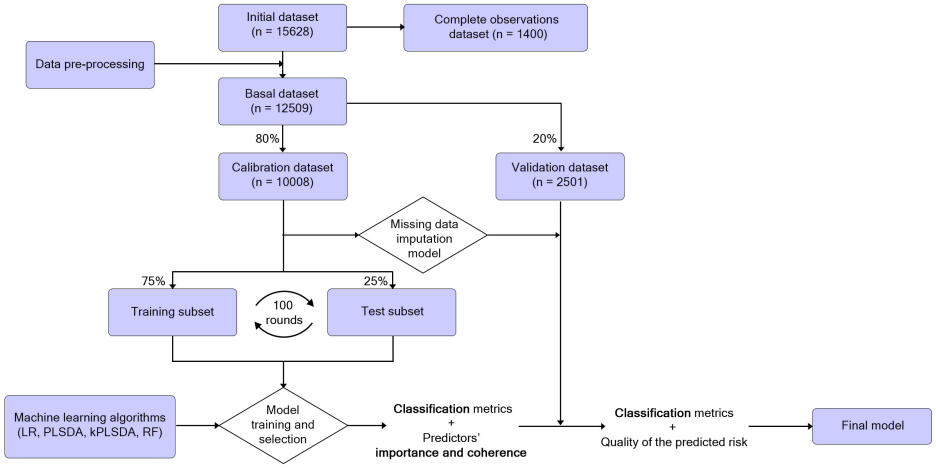


Figure 9.1: Flux diagram of the data used for the mortality prediction model building and validation. Data were stored in the REDCap storage service. The initial database (N = 15,628) was preprocessed and split into calibration (N = 10,008) and validation (N = 2,501) subsets without replacement. The calibration data set was used to set the optimal hyperparameters of the classifiers. The final model was chosen to assess the performance with the validation data set. LR = Logistic Regression. PLSDA = Partial Least Squares–Discriminant Analysis. kPLSDA = kernel PLSDA. RF = Random Forest.

The initial data set (N = 15,628 with 2,846 deceased individuals) was preprocessed to obtain a clean Basal data set (n = 12,509). This depuration process eliminated variables and observations with excessive missing values or errors in the data. A preliminary univariate study (Table 9.3) was conducted to explore potential significant predictors for the mortality outcome. This way, the missing data percentage could be reduced while being cautious of keeping potentially important predictors. Technical details of the data pre-processing step yielding the Basal dataset are described in the following paragraphs.

9.2.1 Missing data cleaning and imputation

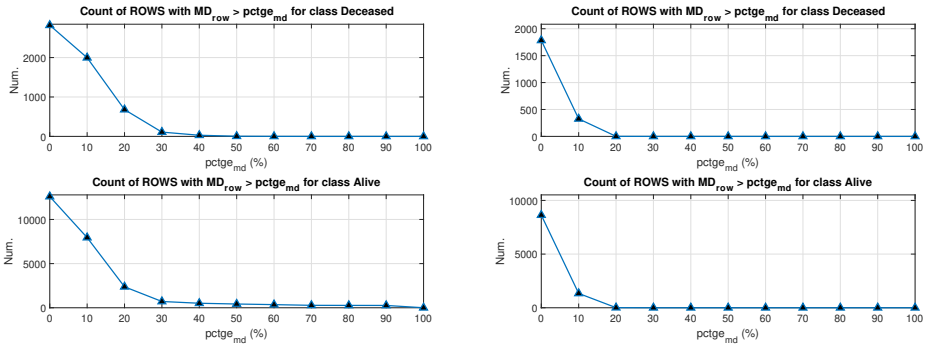
First, the process of "cleaning" and imputing the missing values within the Initial database of the study until reaching a final Basal dataset with complete observations. This goal required finding a balance between the number of variables and observations that would be removed, trying to minimize the loss of information, and, simultaneously, the imputation's impact on interpreting the data modelling results.

Given the unbalancedness between alive and deceased individuals (81,79% vs. 18,21%), the priority was to preserve as many patients of the latter class as possible since they constitute the limiting category. Moreover, ICU-related variables were not considered as not all patients underwent ICU treatment or monitoring, which induced the presence of missing values not directly imputable by standard approaches typically utilized for this purpose. Similarly, nested variables (e.g., the dose of drug A is nested to whether a patient has received the corresponding treatment) were also removed. Other variables, such as the Body Mass Index (BMI), Lymphocytes to C-reactive protein Ratio (LCR), Platelets to Lymphocytes Ratio (PLR), and Neutrophils to Lymphocytes Ratio (NLR), were included since they could be potential biomarkers for mortality.

Investigating the distribution of missing values across patients (i.e., database rows), it was observed that for the group of alive patients, approximately 757 (6%) out of the 12,782 individuals showed more than 30% of missing data records. This percentage increases up to 8.4% (120/2,846) for the deceased class (Figure 9.2a).

Thus, a first cut-off point was established to filter out observations with a percentage of missing entries larger than 30% for the alive patients. After this first step, the residual percentage of missing values was assessed variable-wise (i.e., database columns). As one can easily deduce from Table 9.1, the number of variables with over 50% of missing records was practically the same for both classes of patients. Therefore, those exhibiting more than 50%

This alternating "cleaning" procedure was iterated until variables such as "affected quadrants", "curb65" or "oxygen saturation", appeared as the next ones to be deleted because of their missing data percentage. Given their medical relevance, they must be kept in the final dataset. For this reason, we decided to stop this row/column selection at this point, which yielded a database containing 10,515 alive and 2,085 deceased individuals, each with less than 20% of missing data records (Figure 9.2b).



(a) First iteration of missing data cleaning procedure applied to the Initial dataset.

(b) Second iteration of missing data cleaning procedure applied to the Initial dataset.

Figure 9.2: Number of patients with a percentage of missing values beyond the values expressed along the x-axis for the deceased (up) and the alive group of patients (down), for the first iteration of missing data cleaning (a) and for the second one (b).

This database was then used to obtain one with complete patient observations (i.e., without missing data), consisting of 36 variables, 158 deceased and 1,243 alive individuals. If some of the previously removed variables were found to have complete records for this subset of patients, they were finally re-integrated into the ultimate data structure.

Figure 9.3 shows a bar plot with the percentage of missing data for the remaining removed variables within the patients gathered in the complete database. As can be seen, most of them show more than 25% of missing entries. If complete observations were to be kept, considering these variables would, thus, imply reducing the sample size under study even more. Considering that an already substantial reduction of the number of observations was performed (only data for 9.72% of alive patients and 5.55% of deceased patients were finally analyzed), it was decided not to re-include any of them.

Finally, the missing values on the Calibration dataset were imputed using an extension of Trimmed Scores Regression (TSR) capable of coping with categorical and integer variables. This adaptation was not fully developed but was tested and compared to other approaches, enabling missing data imputation for categorical variables. This drew a direction for a contribution which is still ongoing. More information can be found in the Future Lines section (Chapter 11, Section 11.3.1).

Table 9.1: Percentage of missing values for the variables measured in this study (sorted in descending order). Only those with over 35% of missing records are listed for each category of patients under study.

Deceased patients		**Alive patients**	
Missing Data (%)	**Variable Name**	**Missing Data (%)**	**Variable Name**
65.57%	height	63.17%	lactic acid
62.27%	weight	53.29%	height
52.26%	glasgow	52.89%	glasgow
51.58%	lactic acid	50.48%	ferritin
51.36%	creatine kinase	48.35%	weight
49.28%	ferritin	47.98%	creatine kinase
41.32%	affected quadrants	35.38%	procalcitonin
26.90%	procalcitonin	33,37%	affected quadrants

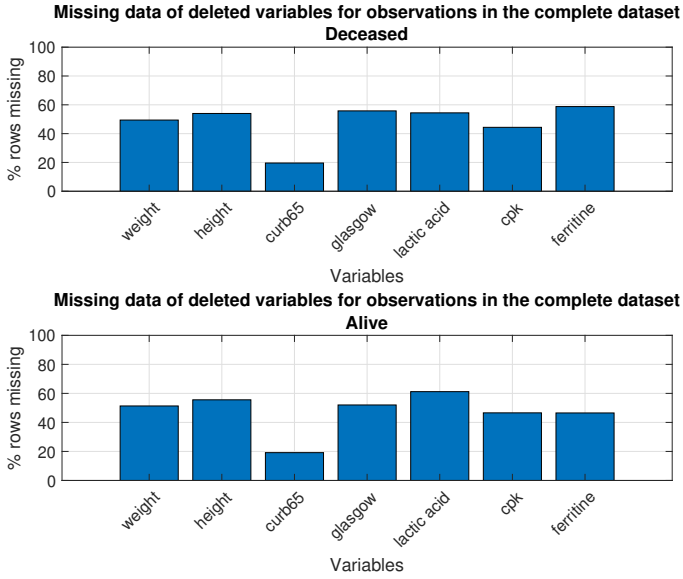


Figure 9.3: Percentage of missing entries within the measured variables excluded by the cleaning procedure for the deceased (above) and the alive (below) patients included in the complete database.

9.2.2 Outlier detection

Mathematically, outlier removal is justified, given the distortions that the corresponding observations can induce in the estimated parameter of the fitted

model, which can bias the study's conclusions. Detecting anomalous observations and studying the reasons behind their anomalous behaviour is key before further using a particular dataset. Thus, in the second step, the complete database was examined by Principal Component Analysis (PCA, Section 3.3.1).

Here, each individual's Distance to the Model (DModX) was assessed to identify the presence of anomalous observations. The DModX statistic assumes abnormally high values when atypical patterns in the correlation structure of the measured variables are observed. Only one patient (highlighted by a red circle) was characterized by a relatively large DModX value. When the contributions of each predictor to the DModX associated with the outlying patient are inspected, the highest one is related to NLR. Plotting the raw values of this variable for all the patients of the complete dataset highlights that the one for the concerned individual is relatively larger compared to those measured for all the other subjects.

9.2.3 Model evaluation

Afterwards, as seen in Figure 9.1, the basal data set was randomly split into the calibration data set ($N = 10{,}008$) and the validation data set ($N = 2{,}501$). The calibration data set was used to fit classifiers and to build a missing data imputation model using an adaptation of the Trimmed Scores Regression method (Section 11.3.1). The imputed Calibration data set was repeatedly (100 repetitions) split into a training subset and a test subset. Four supervised algorithmic techniques were used as classifiers: Logistic Regression (LR) [184], Partial Least Squares Discriminant Analysis (PLSDA) [62], kernel-PLSDA (kPLSDA) [185], and Random Forest (RF) [66].

In each repetition of the calibration, all classifiers were trained and then used to predict the mortality of the test subset. Next, all classifiers were compared in three different types of assessment. The classification performance and the importance and coherence of the measured variables were evaluated as commented on in Section 3.4.4. Besides, a third type of assessment on the quality of risk calibration was also performed. This type of analogy is especially relevant for medical classification models, given the direct implications that an over (or under) estimation of the mortality risk can have in medical decision-making. With this purpose, a calibration curve was fitted using the information about the predicted risk (x-axis) and the observed proportion of deceased patients among those within that group of expected risk (y-axis) [186].

The optimal model was selected considering the model that, with the minimum number of most important predictors, yielded the best classification performance and the best calibration of the predicted risk [186]. Finally, the information about the optimal model was used to configure the mortality score model. This procedure aims to replicate a simplified classifier not based on a complicated and device-based calculus.

All datasets are accessible in ZENODO [187]. The statistical analysis was executed using MATLAB (2020b), R 4.0.2, and Python 3.8.3.

9.3 Datasets

The data used in this study were obtained from the RERFAR-COVID-19-SEFH Registry, a nationwide prospective registry sponsored by the Spanish Society of Hospital Pharmacy (SEFH). It is an extensive repository of anonymized COVID-19 medical records of 15,628 patients admitted to Spanish hospitals from March 20 to July 15, 2020. The Spanish Agency approved the study protocol for Medicines and Medical Devices (AEMPS) and the Institutional Review Boards of the 174 participating hospitals. The protocol is available online at the European Network of Centers for Pharmacoepidemiology and Pharmacovigilance (ENCePP)(R) website [188].

All registered patients were diagnosed with SARS-CoV-2 testing on nasopharyngeal swabs (real-time reverse transcriptase-polymerase chain reaction) at admission. Data were collected and managed using REDCap electronic data capture tools hosted at SEFH [189]. This vast database contained 256 fields for each patient from admission to death or 42 days following hospital discharge. A total number of 1,036 pharmacists from 174 hospitals contributed to the collection of anonymized data from the patient's electronic medical records. A maximum of 200 patients per hospital was recommended to prevent over-representation bias from large hospitals. Patient selection was done by centralized randomizing up to 200 patients in each hospital.

The primary endpoint was all-cause mortality, codified as the binary variable "mortality" with levels "alive" (numerically as zero) or "deceased" (numerically as one). The baseline was the date of hospital admission. The follow-up censoring date was July 15th, 2020; if a patient had not reached the outcome (death) by the time the data were obtained, their outcome was considered null. Clinical routine data from medical records available in the database included demographic variables, clinical conditions at admission, comorbidities (type and number), chronic medication treatments, biochemical and haematological

analytics, and timing of events (from the onset of symptoms to emergency room visit, admission or microbiological diagnosis)–see Table 9.2.

Table 9.2: Blocks of variables included in the data set registered at the admission event of a patient with COVID-19. ACEI - angiotensin-converting enzyme inhibitors; ARB - Angiotensin II receptor blockers; NSAID - Non-steroidal anti-inflammatory drugs.

Block	**Number of variables**	**Variable Names**
Demographic variables	2	Age, Sex.
Clinical variables at admission	6	Fever within previous 24h (Fever 24), Conscience, Respiratory frequency > 24 breaths per minute (Rf 24), Systolic Blood Pressure < 90 mmHg within the previous 24 hours (SBP 90), Affected quadrants, Oxygen saturation.
Comorbidities	11	High Blood Pressure (HBP), Diabetes Mellitus (DM), Chronic Obstructive Pulmonary Disease (COPD), Asthma, Cardiac Failure, Ischemic Heart Disease (IHD), Kidney failure, Cirrhosis, Neurological precedents, Neoplasia, Number of comorbidities
Pharmacological treatments for chronic conditions	4	Previous treatment with ACEI, ARB, Previous treatment with NSAID, Previous treatment with montelukast.
Analytics at admission	12	Creatinine, Lactate dehydrogenase (LDH), Leukocytes, Neutrophils, Lymphocytes, Platelets, C-reactive protein (CRP), Hemoglobin, Procalcitonin (PCT), Neutrophils to Lymphocytes Ratio (NLR), Lymphocytes to CRP Ratio (LCR), Platelets to Lymphocytes Ratio (PLR).
Admission event variables	3	Time between the initial symptoms and the arrival to the emergency room (Time init - urg), Time between the initial symptoms and the hospital admission (Time init - admission), Time between the initial symptoms and the microbiological confirmation (Time init - micro).

9.4 Results

The first part of this section reports a descriptive analysis. Secondly, the results obtained by the four classifiers to predict mortality are presented and compared. Finally, the confection of the mortality score based on the structure of the best model is explained.

An initial univariate analysis was done to identify the variables that could be potentially important in further study steps. Such research was done first with the data set of complete observations. Table 9.3 shows the results only for those predictors found to be the most relevant *a posteriori*, based on the

results obtained by the four classification techniques exploited in this study. Additional univariate analyses were carried out on the imputed Calibration and Validation datasets to check the coherence of the results from Table 9.3.

Table 9.3: Characteristics of patients in the complete data set. Summary of the univariate tests based on the odds-ratio yielded by univariate logistic regression models built between every predictor and the mortality response. p-values in bold are $< 1.10^{-6}$. The mean and standard deviation (in parentheses) values are indicated for each numerical predictor. The number and percentage (in parentheses) of cases are reported for each categorical predictor.

Complete dataset	**Total**	**Alive**	**Deceased**	**Odds ratio**	***p*value**
	$\boldsymbol{N = 1400}$	$\boldsymbol{N = 1243}$	$\boldsymbol{N = 157}$		
Variable	**m(s.d.)/N(%)**	**m(s.d.)/N(%)**	**m(s.d.)/N(%)**	**(95%C.I.)**	
Age, years	63.82 (14.73)	62.44 (14.44)	74.75 (12.27)	1.07 (1.06;1.09)	<0.0001
Oxygen saturation, %	92.9 (5.5)	93.36 (5.1)	89.27 (7.01)	0.91 (0.88;0.93)	<0.0001
Platelets, $10^3/mm^3$	208.63 (87.55)	210.14 (88)	196.66 (83.21)	1 (1;1)	0.07
LDH, U/L	375.17 (193.31)	362.29 (178.78)	477.14 (262.37)	1 (1;1)	<0.0001
Creatinine, mg/dl	1.01 (0.64)	0.97 (0.57)	1.34 (1.01)	1.73 (1.52;1.93)	<0.0001
Lymphocytes, $10^3/mm^3$	1.68 (4.18)	1.73 (4.36)	1.3 (2.27)	0.95 (0.86;1.03)	0.21
Leukocytes, $10^3/mm^3$	7.34 (5.21)	7.17 (5.14)	8.65 (5.59)	1.04 (1.01;1.06)	0.006
Hemoglobin, $10^3/mm^3$	13.89 (1.95)	13.92 (1.94)	13.6 (2)	0.92 (0.83;1)	0.047
D dimer, $10^3/mm^3$	1,233.45 (2,488.68)	1,092.22 (2,017.04)	2,351.62 (4,662.09)	1 (1;1)	<0.0001
Time init - admission, days	7.14 (4.83)	7.27 (4.64)	6.08 (6.07)	0.94 (0.89;0.98)	0.0028
N. of comorbidities	1.22 (1.23)	1.13 (1.2)	1.92 (1.31)	1.57 (1.44;1.69)	<0.0001
Altered conscience				6.09 (5.62;6.55)	<0.0001
No	1,312 (93.71%)	1,189 (95.66%)	123 (78.34%)		
Yes	88 (6.29%)	54 (4.34%)	34 (21.66%)		
Respiratory frequency > 24 bpm				2.58 (2.24;2.92)	<0.0001
No	1,028 (73.43%)	942 (75.78%)	86 (54.78%)		
Yes	372 (26.57%)	301 (24.22%)	71 (45.22%)		
Cardiac failure				2.15 (1.52;2.79)	0.018
No	1,337 (95.5%)	1,193 (95.98%)	144 (91.72%)		
Yes	63 (4.5%)	50 (4.02%)	13 (8.28%)		

Neurological precedents				2.46 (2.05;2.86)	<0.0001
No	1,219 (87.07%)	1,100 (88.5%)	119 (75.8%)		
Yes	181 (12.93%)	143 (11.5%)	38 (24.2%)		
Neoplasia				1.31 (0.7;1.92)	0.38
No	1,307 (93.36%)	1,163 (93.56%)	144 (91.72%)		
Yes	93 (6.64%)	80 (6.44%)	13 (8.28%)		
SBP < 90				4.09 (3.61;4.58)	<0.0001
No	1,313 (93.79%)	1,183 (95.17%)	130 (82.8%)		
Yes	87 (6.21%)	60 (4.83%)	27 (17.2%)		
Kidney failure				2.8 (2.28;3.32)	0.00012
No	1,314 (93.86%)	1,178 (94.77%)	136 (86.62%)		
Yes	86 (6.14%)	65 (5.23%)	21 (13.38%)		

According to the preliminary analysis in Table 9.3, deceased patients presented higher risk factors at hospital admission: age, creatinine levels, SBP under 90 and respiratory frequency above 24 bpm. Deceased patients also exhibited a higher proportion of altered conscience. Besides, the number of comorbidities ($\geq$2) was also significantly higher in deceased individuals, with a higher prevalence of cardiac failure, neurological antecedents, neoplasia, or kidney failure. On the contrary, deceased patients presented significantly lower values for the following protective factors: oxygen saturation, platelets and lymphocytes at hospital admission.

Next, we evaluated the different classification models for predicting COVID-19 outcomes using the framework of repeated training and testing. The Random Forest classifier performed best, with a median AUC of 0.8648. Still, results were very similar for all the methods when including all predictors (Table 9.4). However, as reducing the number of features eases the practical implementation of a classifier, we implemented a forward step-wise approach to select the minimum number of predictors for the final model while maintaining the trade-off between performance and usability.

Using the predictor importance coefficients, all 38 predictors were ranked according to the values of their corresponding coefficients. Figure 9.4 displays the median predictor coefficients over the 100 re-sampling folds. Colour intensity indicates the strength of the relationship between each predictor and the mortality risk. Positive coefficients are represented in red and denote risk factors (positively correlated with the mortality risk), while negative coeffi-

Table 9.4: Evaluation metrics for the calibration data set obtained over the 100 folds of training and testing with the calibration data set. The same values are illustrated in S1 Fig. The classifier LR refers to Logistic Regression, PLSDA to Partial Least Squares—Discriminant Analysis, KPLSDA to Kernel PLS-DA and RF to Random Forest. The parameters correspond to the 2.5% percentile (P2.5), to the 50% percentile (Median) and the 97.5% percentile (P97.5).

Classifier	Value	Sensitivity	Specificity	AUC	Accuracy	F1-score	MCC
LR	P2.5	0.7384	0.7957	0.8611	0.7551	0.5255	0.4430
	Median	0.7468	0.8340	0.8640	0.7607	0.5355	0.4518
	P97.5	0.7817	0.8424	0.8684	0.7842	0.5541	0.4665
PLSDA	P2.5	0.7285	0.7627	0.8428	0.7432	0.5061	0.4187
	Median	0.7612	0.8041	0.8557	0.7670	0.5331	0.4430
	P97.5	0.7788	0.8365	0.8672	0.7820	0.5584	0.4687
KPLSDA	P2.5	0.7530	0.6694	0.8395	0.7486	0.5083	0.4068
	Median	0.7755	0.7729	0.8521	0.7745	0.5396	0.4423
	P97.5	0.8394	0.8297	0.8672	0.7944	0.5784	0.4824
RF	P2.5	0.6861	0.7076	0.8513	0.7158	0.4993	0.4178
	Median	0.7598	0.8129	0.8648	0.7698	0.5397	0.4479
	P97.5	0.8356	0.8711	0.8780	0.8161	0.5738	0.4864

cients are graphed in blue and connote protection factors against mortality by COVID-19.

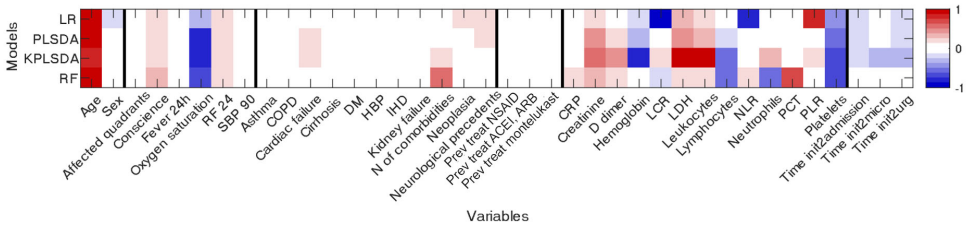


Figure 9.4: Importance metrics for all predictors. Median values (over the 100 re-sampling folds) of the 38 predictor coefficients sorted by type of data blocks (demographic variables, clinical variables at admission, comorbidities, pharmacological treatments for chronic conditions, analytics at admission and information about the admission event).

Instead, the coefficient sign consistency over the 100 folds mentioned above can be inferred from Figure 9.5, where each bar indicates the percentage of times the corresponding coefficient was positive or negative. Low consistency points out unclear relationships with the mortality risk that may arise from the adopted re-sampling scheme and might not be necessarily substantiated by biomedical rationales.

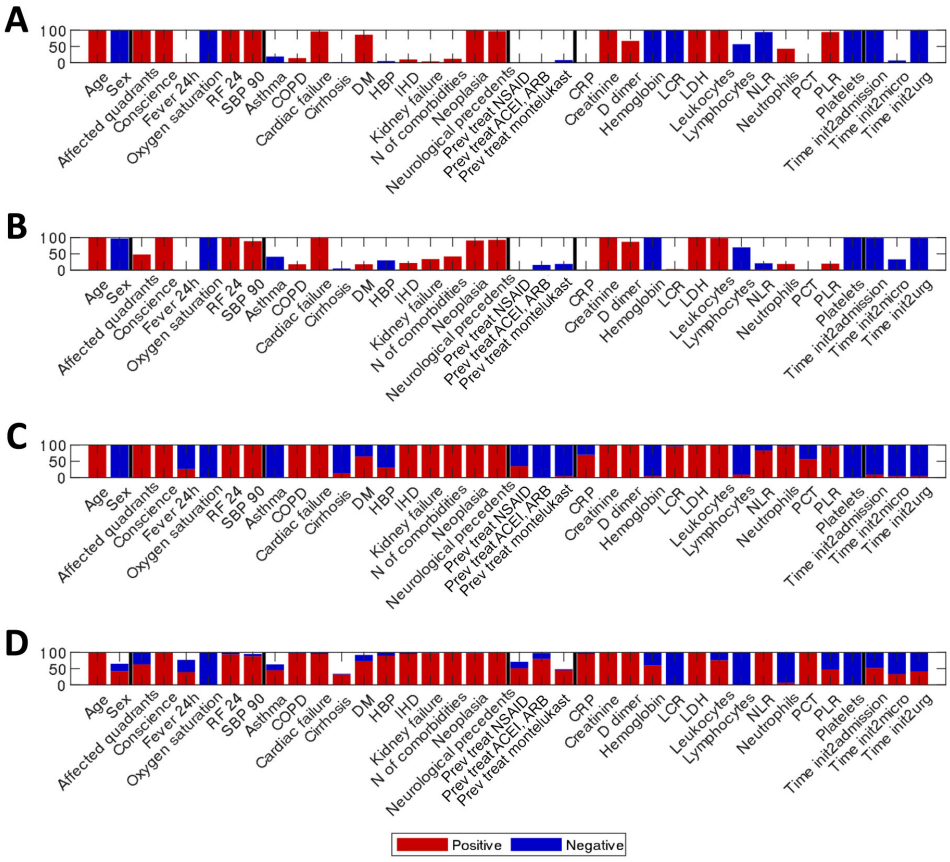


Figure 9.5: Coherence metrics for all predictors and classifiers. Bar charts representing the percentage of folds in which each predictor was found to show a positive (red) or a negative coefficient (blue) for the LR model (A), the PLSDA model (B), the kPLSDA model (C), and the RF model (D). Bars with high colour consistency indicate highly consistent relationships between predictors and mortality.

Based on the coefficients' magnitude and their sign's consistency, a subset of 18 features showing high median coefficient (absolute) values and high sign consistency (above 75%) was selected. Figure 9.6 shows the absolute value of their median coefficients, sorted in descending order.

The ranking of the most important features was finally used for model validation. To this end, an incremental strategy was implemented. The first model

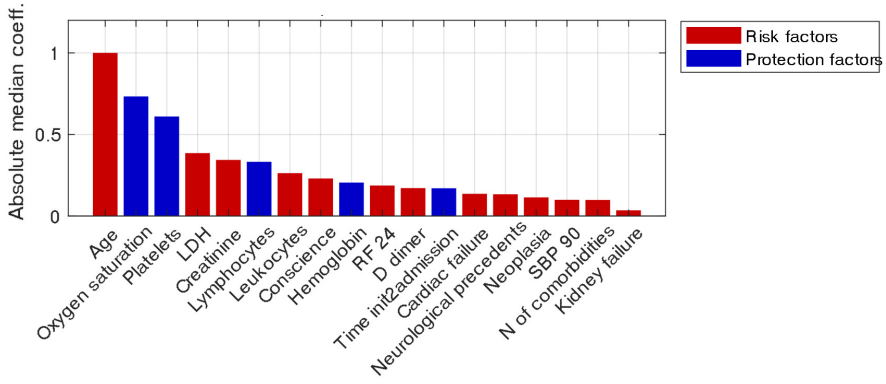


Figure 9.6: Importance of most relevant variables. Ranking (in descending order) of the 18 variables selected according to their Importance and the consistency of their relationship with the mortality risk over the 100 re-sampling iterations.

to predict mortality was fitted using only the most important feature (age), and its corresponding classification metrics were obtained. Afterwards, the second most predictive feature (oxygen saturation) was additionally considered for model calibration and the resulting classification metrics were stored. This was iterated for all 18 features from Figure 9.7.

The results obtained when the trained models were used to classify the validation data set were assessed from two perspectives. In the first place, the sensitivity (Equation 3.34), the specificity (Equation 3.35), the AUC (Equation 3.37), the accuracy (Equation 3.38), the F1-score (Equation 3.39) and the Matthew's Correlation Coefficient (MCC, Equation 3.40), were calculated to report the overall classification performance. The results showed that satisfactory classification metrics were achieved using all the employed classifiers. The LR, RF and kPLSDA classifiers yielded an AUC of around 0.85 with the final validation data set. Besides, their evolution with the number of important variables modelled seemed to agree strongly. This coherence was a good indicator of the overall classification quality. Still, it was important to account for another criterion for determining the best classifier: the quality of the prediction risk.

A second assessment of the quality of the predicted risk was completed in the performance report. Figure 9.7 shows the estimated Intercept and slope of the risk calibration curve fitted for each incremental model and classification

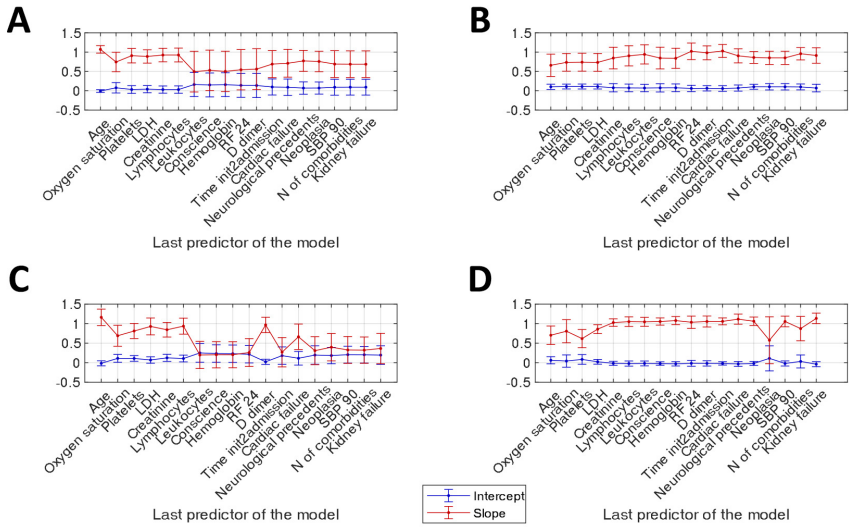


Figure 9.7: Assessment on the quality of the risk calibration. Intercept and slope of the risk calibration curve obtained for each incremental model with LR (A), PLSDA (B), kPLSDA (C) and RF (D).

technique. Confidence intervals were calculated assuming a confidence level of 95% and using the estimated coefficients' standard error.

Figure 9.8 shows each classification technique's calibrated risk prediction curve at its optimal variable number setting. These optimal calibration curves are the closest ones to the dashed diagonal line. Curves in regions aside from the diagonal would indicate an underestimation of the mortality risk (leading to under-treatment) or an overestimation (leading to over-treatment).

In general, all the algorithms had a similar performance, although there were differences in the optimal number of variables. Six variables (age, oxygen saturation, platelets, LDH, creatinine, and lymphocytes) were selected for LR and kernel PLSDA. The PLSDA model reached an optimal performance with ten predictors (from age to rf-24) and RF with five predictors (from generation to creatinine).

Consequently, in light of the results, Random Forest was selected as the best classifier, showing slightly better results and with the minimum number of predictors. Figure 9.9 shows violin plots with the distribution of the five most

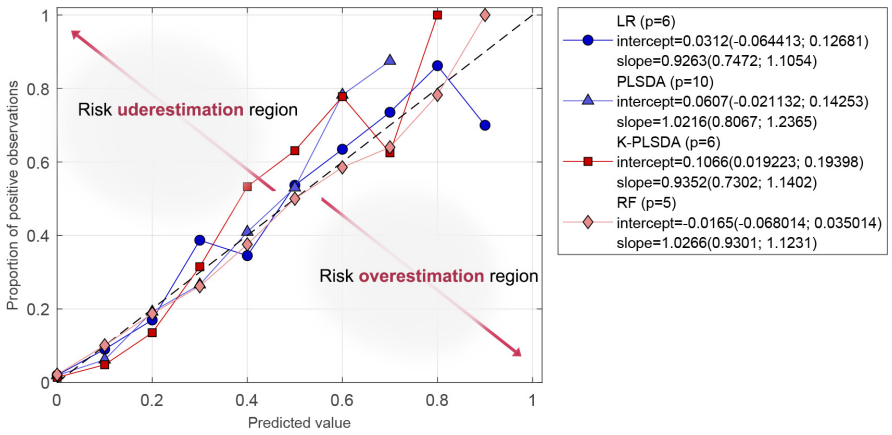


Figure 9.8: Optimal calibration risk prediction curves. Observed mortality (%) vs. the predicted mortality risk for all the classification algorithms under study at their optimal variable number setting. Predicted risk values were rounded to the first decimal digit, i.e., predicted value 0.1 refers to predictions between 0.05 and 0.15.

important variables on this ranking for the deceased and the alive patients from the calibration data set.

The results yielded by the RF classification model suggested that five predictors encoded enough information to predict a given patient's mortality risk accurately. These five predictors were then explored to devise a simplified classifier based on them.

Initially, the marginal distributions of age, oxygen saturation, platelets, LDH and creatinine for each class ("alive" and "deceased") were inspected (Figure 9.10). Values of interest (such as the intersection points between the group distributions of each variable and the percentiles delimiting such distributions) were chosen as thresholds for each predictor.

Next, a regression model was fitted between these five dichotomized variables (as shown in Figure 9.10) and the mortality risk predicted by the RF model. Therefore, the importance of each variable in the mortality score definition can be quantified as its respective regression coefficient. These coefficients were scaled by the minimum one and rounded to the closest integer afterwards, which yielded a scale of variable relative importance for the mortality prediction (Table 9.5).

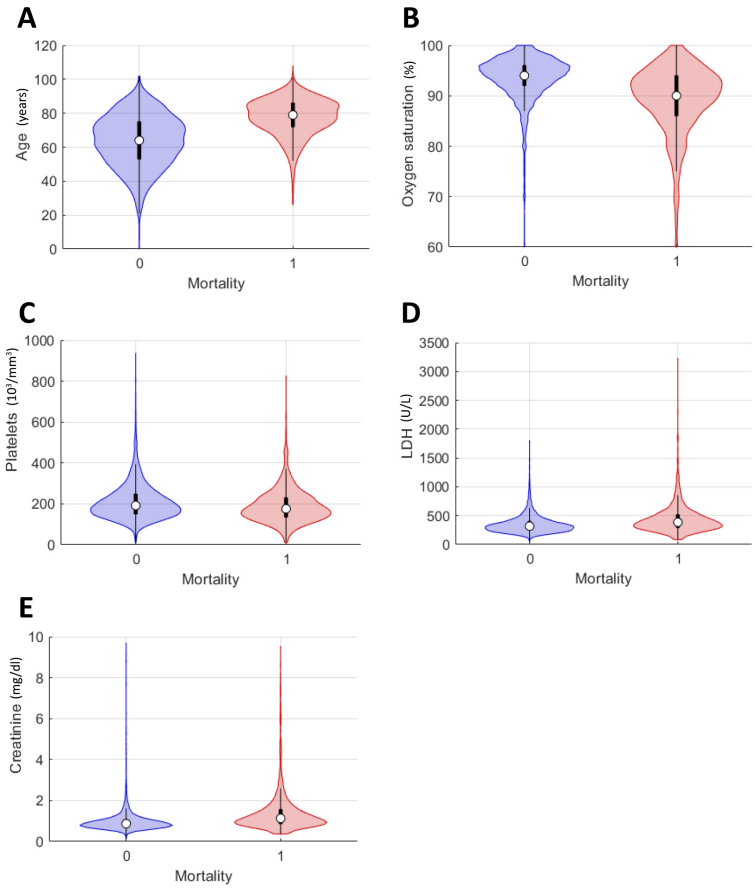


Figure 9.9: Marginal distributions of predictors used by the RF. Violin plots (blue: alive patients; red: deceased patients) for age (A), oxygen saturation (B), platelets (C), LDH (D), and creatinine (E).

The relative importance from Table 9.5 was used to establish more intervals for variables with relative importance above one. These new intervals were based on searching characteristic points of the distributions, such as slope increase or decrease. This way, the importance of each variable according to the model was accounted for to develop a realistic set of scoring rules (Figure 9.11).

It is worth mentioning that, at first, applying these rules leads to a score ranging from zero to 11. However, since the first three levels of the score were grouping very little information about the mortality in the deceased group

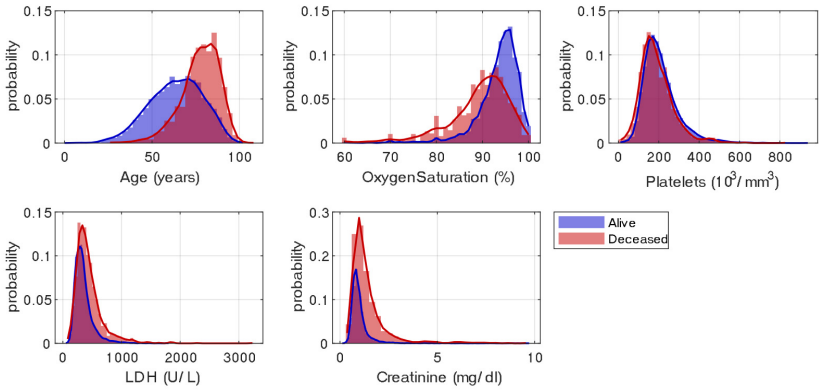


Figure 9.10: Histograms with marginal distributions of the final set of predictors. Age, oxygen saturation, platelets, LDH and creatinine distribution within alive (blue) and deceased (red) patients.

Table 9.5: Relative importance of the five dichotomized variables.

Variable	Relative Importance	Final Relative Importance (rounded)
Age	3.60	4
Oxygen saturation	2.60	3
Platelets	1	1
LDH	1.20	1
Creatinine	1.60	2

(Figure 9.12), they were merged into the "zero" category, resulting in the final score with nine levels, ranging from zero to eight.

The final mortality score ranged from zero to eight, increasing with the mortality risk as shown in Figure 9.13, with the percentage of deceased and alive patients for each level of the mortality score.

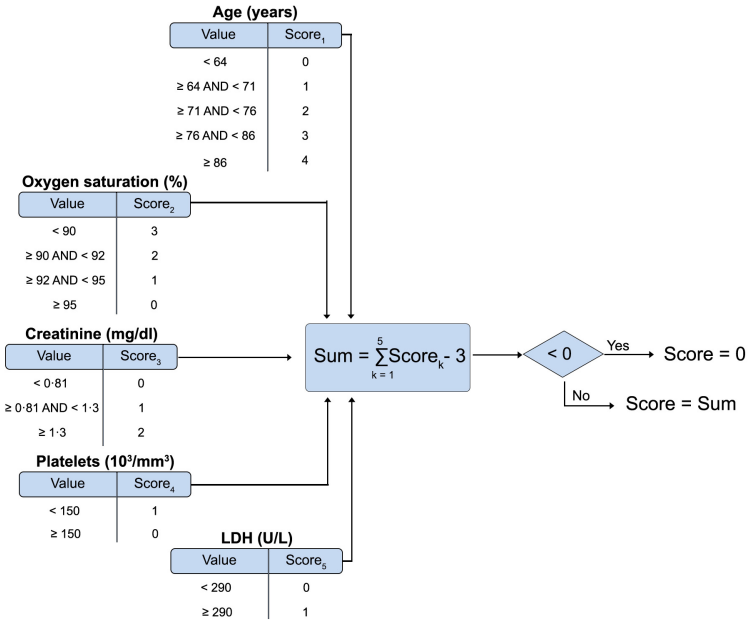


Figure 9.11: Final set of scoring rules. Formulation of the nine-levels mortality score for COVID-19 patients at their hospital admission

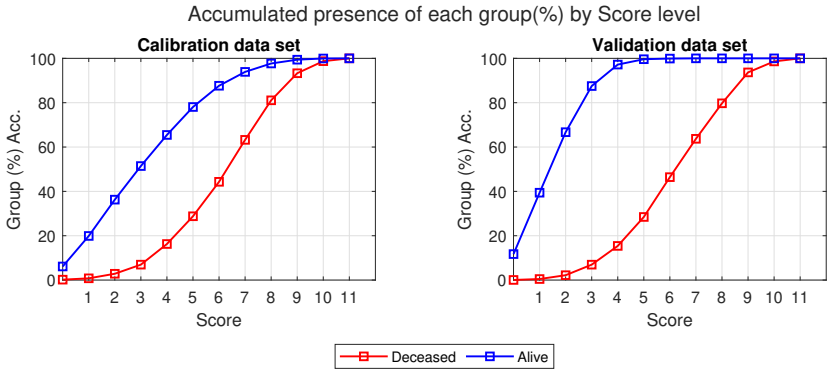


Figure 9.12: Accumulated distributions of deceased (red) and alive (blue) patients along the score values for the calibration dataset (left) and the validation dataset (right).

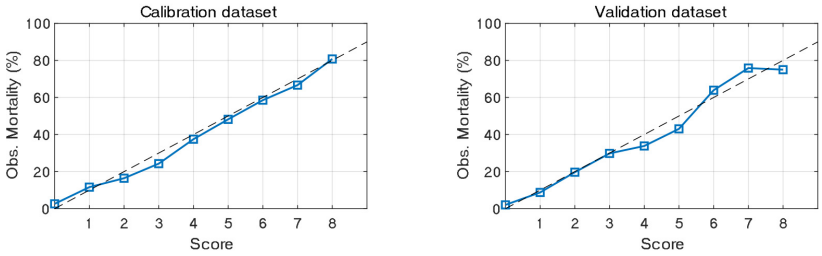


Figure 9.13: Observed mortality vs Score curves. Observed mortality at each level of the score for the Calibration data set and for the Validation data set.

9.5 Conclusions

In this work, we applied machine learning and multivariate statistical classification techniques to data prospectively collected from COVID-19 hospitalized patients in all regions of Spain to build a model for predicting their mortality risk during hospitalization. The final model encompassed five predictors (age, oxygen saturation, creatinine, platelets, and LDH) and was trained based on the Random Forest algorithm. It returned in external validation (when patients not considered for model training and optimization were to be assessed) an AUC of 0.8454.

Virtually all published studies on COVID-19 populations agree that both age and oxygen saturation at hospital admission are closely related to the likelihood of death [181], [190]–[199]. Besides, COVID-19 mortality is strongly linked to a specific inflammation process and a coagulation disorder. Some patients develop a severe inflammatory syndrome, which results in uncontrolled activation of the immune system and a massive release of pro-inflammatory cytokines, which translates into an increase in acute-phase reactants such as C-reactive protein, interleukin-6, ferritin, cell destruction markers such as LDH, and an increase in pro-inflammatory cells such as neutrophils [190], [192], [196], [197], [200], [201].

Another complication that results in high mortality in these patients is coagulation disorders. COVID-19 results in a systemic hypercoagulation state, producing pulmonary thromboembolisms, ischemic strokes, and other conditions, and many patients experience severe complications. This complication can be assessed based on two laboratory parameters: D-Dimer and platelets [191], [193]. Most prognostic studies also identified creatinine or urea as important factors related to mortality risk [181], [190]–[199], [202]. Our data showed

that creatinine is the laboratory parameter that most influences mortality in renal function, indicating whether renal filtering is effective.

Model calibration was carried out, exploiting exclusively information that can be easily recorded at the admission to the hospital of COVID-19 patients. However, even if this information is available at the early stages of their hospitalization, we also tried to reduce as much as possible the number of variables to obtain an accurate mortality risk prediction without compromising its quality and performance.

A clear strength of our work is that the original database contains routinely obtained clinical data readily available at the hospital admission of this kind of patient. In multiple epidemiological studies, age, oxygen saturation, platelets, LDH, and creatinine were previously identified among the core variables related to morality and severe disease development after SARS-CoV-2 infection. These five variables were also the most predictive features in our research, independently of the classification algorithm utilized. In practical terms, this means that predictive models could accurately estimate the mortality risk for a given patient just by recording the values for these five clinical parameters. All these features are coherent with the information already available about this disease [181], [197], [198]. Our results showed that the probability of mortality for COVID-19 depends on variables of different nature and not exclusively on those associated with respiratory functions. In addition to making models more economical, reducing the number of predictors to a minimum set that is easy to measure also enables easy implementation of this predictive strategy for clinical use and further validation with other datasets.

Another strength of this analytical approach lies in the sample size, a prospectively recruited cohort of 12,509 patients, including more than 2,000 deceased individuals. The sample size of previous studies on mortality among COVID-19 inpatients performed at Spanish hospitals ranged between 2,000 and 4,000 individuals [193], [203] with 6.5% and 28.0% of mortality, respectively. Conversely, other studies with a larger sample size could not achieve proper predictive models [204]. Besides, the massive number of articles submitted during the pandemic in 2020 forced editorial offices (even of well-known biomedical journals) to change their policies due to scandals and polemics related to the reliability of the published data [205].

Furthermore, a systematic approach was implemented to compare statistical and machine learning algorithms regarding classification performance and model inference on the predictors. This compelling (but barely used) approach enables a more comprehensive assessment of inferential coherence among differ-

ent methodological strategies, which otherwise would be exploited as black-box techniques. Consequently, we consider that this additional validation increases the reliability of our results.

Our study showed, though, several limitations. This model was fitted in one of the worst moments of the pandemic. The patients included in this study were hospitalized during the first period of the pandemic, so the clinical characteristics of patients in our country today could be different [206]. Moreover, in 2022, patients could have different outcomes given better knowledge of COVID-19 disease and the coverage of the vaccination campaign. However, in many countries - especially less developed ones - the situation may differ greatly from our current scenario, and our findings may still be clinically helpful. In any case, our final prediction model should be tested with an updated and more recent picture of the COVID-19 situation. Although it is not clear if the predictors of mortality have changed, new conditions may result in lower mortality rates for patients with a high-risk profile. Such an assessment constitutes one of the main objectives of our future research work.

In conclusion, we used several statistical and machine learning approaches to obtain a data-driven model based on variables that could be easily acquired at COVID-19 patients' admission to the hospital to determine their mortality risk. This resulted in a final model based on five predictors (age, oxygen saturation, platelets, LDH, and creatinine) that yielded a highly satisfactory classification performance (with an AUC of 0.8454). The interpretation of this model and the investigation of the relationships between these five predictors and the mortality risk contributed to the definition of a mortality score for COVID-19 patients at their admission that can be easily calculated and easily interpreted (it linearly increases along with the mortality risk). Once validated with a prospective cohort representative of the latest COVID-19 management protocols, the mortality prediction model could be used as a powerful tool for the early recognition of the gravity and priority needs of SARS-Cov-2-infected hospitalized patients.

Chapter 10

Fluorescence measurements standardization

Part of the content of this chapter has been included in:

> [207] González-Cebrián, A., Borràs-FerrFerrís, J., Boada, Y., Vignoni, A., Ferrer, A. & Picó, J. PLATERO: A Calibration Protocol for Plate Reader Green Fluorescence Measurements. *Frontiers in Bioengineering and Biotechnology.* **11**, (2023), https://doi.org/10.3389/fbioe.2023.1104445.

10.1 Introduction

Synthetic Biology is a field with an increasing role within the manufacturing industry. However, its settlement from a trial-and-error process to an engineering discipline, embracing more formal methods, requires standards. These facilitate the Design-Build-Test-Learn (DBTL) lifecycle by enabling the integration of inherently different tools and techniques into coherent workflows. The DBTL cycle requires a complete description of the components in a biological system, data to describe the system function and interconnections, and computational models to predict the impact of environmental parameters on the system's behaviour. In this context, data standards expressing genetic constructs and their mathematical models foster information sharing, which is key to overcoming characterization and reproducibility issues across laboratories.

Reproducibility can be ensured by establishing an unbroken chain of calibrations to specified reference standards [208], [209] and quality control of the reference materials used for calibration. Using calibrated absolute standard units and protocols allows measurements and estimations from different sources or measurement device settings to be integrated and compared faithfully into a common domain.

The expression of fluorescent reporters is commonly used for quantifying gene expression levels. Fluorescent dyes are also used for quantifying a wide range of other biological properties. Two main classes of devices are used for measuring fluorescence: flow cytometers and plate readers. A measure of the light emitted by a certain fluorescent molecule, e.g. the Green Fluorescent Protein (GFP), is used to estimate the amount of GFP molecules expressed by the cell. Thus, by linking the expression of a gene of interest to that of GFP, fluorescence measurement can be used to measure the expression level of the first one indirectly.

Two main problems affect the proper characterization of gene expression using fluorescence measurements. On the one hand, the values obtained are affected by the measurement device setup. In particular, the device's gain is set so that measurements do not saturate. Consequently, for a series of related experiments spanning a wide range of fluorescence intensities, it is common for different device gains to be used. This makes comparing results difficult, as the relationship between the actual fluorescence and the gain-affected measurement may be nonlinear.

On the other hand, fluorescence measurements are usually expressed in arbitrary units. Some studies have been trying to normalize fluorescence mea-

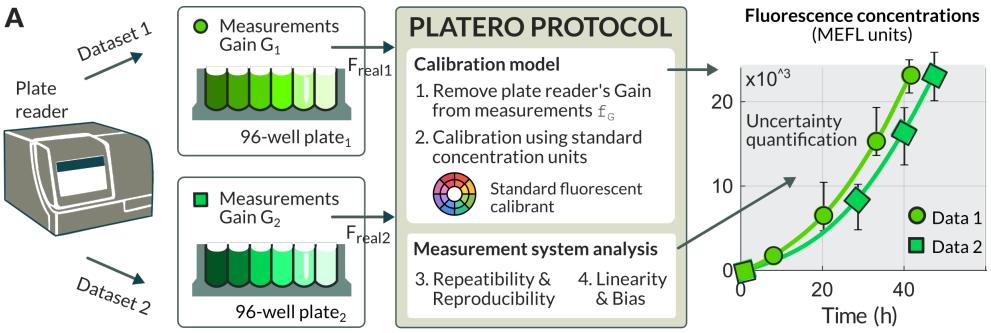
surements with a biological sample cultured in parallel with the experimental models [12]. However, such normalization may produce less precise measurements than normalization using an independent calibrant due to the ill-defined potential variability of the biological samples used for normalization [210].

The model used by PLATERO transformed from fluorescence measurements relative to the plate reader setting and expressed in arbitrary units to units of calibrant concentration, which are absolute, comparable, and independent of the measurement device setup. As for the measurement device setup, we propose a correction of the fluorescence readings by using a gain-effect model. To address the problem of the arbitrariness of units, we use already established protocols [208]–[210] with calibrants that can be used to produce precise estimates of molecules equivalent of fluorescein (MEFL), and fluorescein concentration from fluorescence measurements. The resulting unit calibration model enables users of fluorescence plate readers to bring experimental measurements into a common gain-independent domain. This allows for comparing results obtained from different plate readers possibly located at other laboratories (Figure 10.1a).

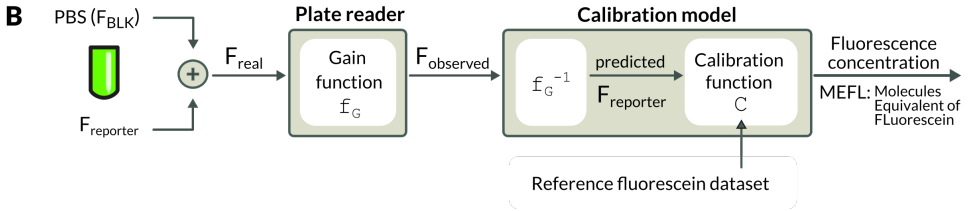
A key aspect of any measurement device and its associated measurement protocol is the analysis of the uncertainty of the calibration and the study of variability and its sources associated with the protocol operations and the measurement device. PLATERO's calibration protocol embeds a Measurement System Analysis (MSA) that provides both an estimation of the uncertainty that we can expect on the predicted concentration value and an assessment of the plate reader being used and the sources of uncertainty.

PLATERO has been implemented as a Matlab toolbox and is freely available at `https://github.com/sb2cl/PLATERO`.

The remaining chapter is organized as follows. Section 10.2 describes the materials used to carry out the protocol and gives a detailed list of instructions. In Section 10.2.2, we explain the methods underlying the calibration model and the associated analysis of uncertainty. In Section 10.4, we describe the results that can be obtained using the protocol and how they can be assessed using the embedded Measurement System Analysis. Finally, a brief discussion is given in Section 10.5.



(a) This flowchart shows how the PLATERO calibration model brings the experimental measurements into a common gain-independent domain using standard concentration MELF units. The calibration protocol embeds a Measurement System Analysis providing an estimation for the uncertainty that can be expected on the predicted concentration value, an assessment of the plate reader being used, and the sources of uncertainty.



(b) This flowchart shows the procedure diagram to retrieve concentration values from observed fluorescence ($F_{observed}$). The $F_{observed}$ values are a function (f_G) of the medium fluorescence (F_{BLK}), the fluorescence of the reporter ($F_{reporter}$), and the gain (G) at which fluorescence values are measured. Once the gain and background effects are removed, the $F_{reporter}$ values are retrieved. The units conversion function (f_{UC}) transforms these corrected fluorescence values into standard concentration units.

Figure 10.1: Workflows illustrating the experimental procedure followed to fit the PLATERO calibration model (a) and then to exploit it with new samples of the fluorescent reporter (b).

10.2 Methods

10.2.1 Experimental procedure

The PLATERO protocol requires preparing serial dilutions of a reference fluorescein sodium salt solution (Sigma-Aldrich #46970) [209] to perform the fluorescence calibration.

This reference solution can be prepared from the fluorescein sodium salt power by weighing and dissolving in a known volume. The concentration of this reference solution can be further confirmed by measuring its absorbance at 492 nm and calculating concentration using an extinction coefficient of 68.029 mM^{-1} cm^{-1}, the appropriate pathlength from your spectrophotometer (normally $\ell = 1\text{cm}$) and the law of Beer-Lambert as follows:

$$C = \frac{Abs_{492}}{\varepsilon \cdot \ell} \tag{10.1}$$

Starting from 1mL of the 10μM reference solution of fluorescein in Phosphate Saline Buffer (PBS), the experimental protocol described in [211], modified from [208], [212]), has to be carried out to get serial dilutions.

In short, following the protocol obtains a serial dilution of fluorescein with five increasing concentrations plus only PBS solution for blanks (F_{BLK}). In our case, we used the concentrations 0.0391, 0.0781, 0.1562, 0.3125, and 0.625 μM. Samples of this serial dilution have to be randomly transferred into a 96-well black/clear flat bottom microplate with 16 replicates for concentration. Therefore, the 96-well plate contains 16 technical replicates per each of the five fluorescein concentrations and 16 technical replicates of the blank F_{BLK}.

Fluorescence measurements of the 96-well plate using a plate reader must be repeated eight times. The plate reader has to be configured to cover a wide range of the spectrum of gains of the plate reader. In our case, we used the Agilent BioTeK Cytation 3 Cell Imaging Multi-Mode Reader to show the capabilities and benefits of using PLATERO. We configured it using an excitation/emission wavelength of 488/530nm. In addition, we arranged it at four different detection gain levels, $G = 50, 60, 70, 80$.

10.2.2 Calibration model

In this section, we first describe the calibration model used by PLATERO, which enables converting from arbitrary fluorescence units to concentration expressed as the equivalent concentration of fluorescein. Next, we show how the uncertainty estimation was included as the final step of the calibration model fitting to validate the conversion model. We show how the Linearity and Bias analysis (L&BA) is applied to obtain the uncertainty of the estimated concentrations the protocol model provides. We used a test based on the confidence interval built around estimating the true concentration of reporters within a well. We will consider the estimation valid if this confidence interval contains the true concentration value. Finally, we describe how to apply the Reproducibility and Repeatability analysis (R&RA) to assess the different sources of the observed variability in the estimations.

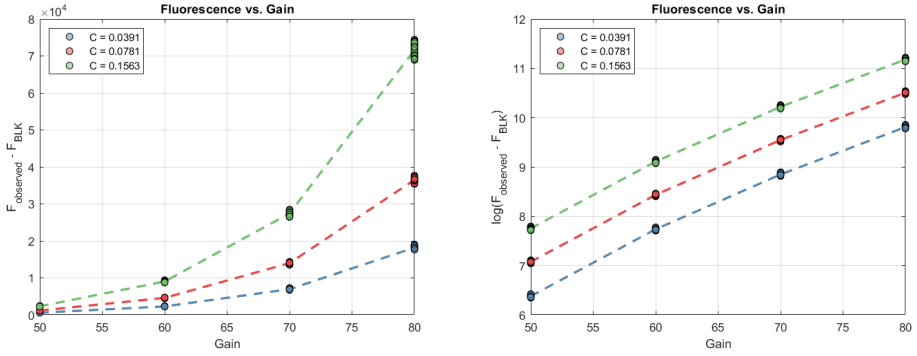
A detailed list of the steps required to apply the full calibration protocol and the Matlab functions performing each step can be found within the PLATERO toolbox available from the GitHub repository `https://github.com/sb2cl/PLATERO`. Throughout the following subsections, some references to the corresponding functions of the toolbox are provided.

Figure 10.1b depicts the different factors involved in the fluorescence measurement provided by a plate reader. To obtain the calibration model used by PLATERO, we have to (1) compensate for the background fluorescence and correct the effect of plate reader gain on the fluorescence observations and (2) convert the arbitrary units of fluorescence of these observations to standard fluorescein concentration units.

Device gain and background fluorescence.

The plate reader gain is one of the key parameters to set up before measuring the fluorescence of a reporter. If the gain is too low, the lower limit of the measured fluorescence range will not be correctly detected by the instrument. Conversely, if the gain is too high, the upper limit of the fluorescence range will saturate, so it cannot be measured. The relationship between the actual fluorescence in a sample (F_{real}) and the fluorescence measured by the plate reader ($F_{observed}$) is a nonlinear function of the gain, as depicted in Figure 10.1.

To obtain the relationship between F_{real} and $F_{observed}$, we carried out an iterative model search looking at the experimental relation between fluorescence F_{real} and the gain G (Figure 10.2). At first glance, visualizing Figure 10.2a,



(a) Fluorescence values (with the medium fluorescence values, F_{BLK} subtracted) of the calibration subset against gain values for each concentration level.

(b) Fluorescence values (with the medium fluorescence values, F_{BLK} subtracted) of the calibration subset in logarithmic scale against gain values for each concentration level.

Figure 10.2: Fluorescence values of the calibration data subset for different gains (a) and considering a log transformation (b). As can be seen, the y-axis represents $F_{observed} - F_{BLK}$, with the additive background noise already removed as suggested later in Equation 10.4, but it is still not F_{real}, as the Gain effect has not been removed yet.

it seems clear that there is not a linear relation between fluorescence values and gain. On the contrary, such a non-linear relation appears to be exponential. For that reason, an exponential effect of the gain is initially proposed in Eq. 10.2.

$$F_{observed} = f_G(F_{real}, G) = F_{real} \cdot e^{b_1 \cdot G} \tag{10.2}$$

Note that taking logarithms in Eq. 10.2 yields the analytical expression of a linear model. Therefore, if it was an adequate approximation, one would expect to see a linear relation when representing the logarithm of the fluorescence as a function of G. However, Figure 10.2b shows a slightly quadratic relation. Hence, it could conceivably be hypothesized to add a quadratic effect to Eq. 10.2 in the exponential term .

From this, we inferred an exponential relationship between F_{real} and $F_{observed}$ with a gain-dependent quadratic term in the exponent:

$$F_{observed} = f_G(F_{real}, G) = F_{real} \cdot e^{b_1 \cdot G + b_2 \cdot G^2} \tag{10.3}$$

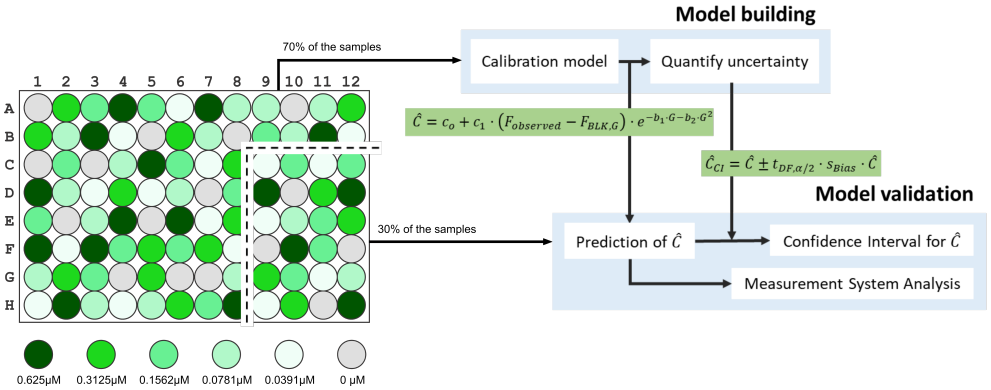


Figure 10.3: Schema representing the assessment on the proposed model done by a model building and a model validation step. Particularly, eleven out of the sixteen wells ($\approx 70\%$) for each concentration level were randomly selected for the Model Building step, and the rest were used for the Model Validation.

Where b_1 y b_2 are the coefficients of the linear and the quadratic terms, respectively, modelling the exponential effect of the gain on the fluorescence. We assumed that the gain correction (Equation 10.3) does not depend on the measured fluorescence values. That is, its structure and the importance of the coefficients depend on the measurement device but not on the range of the fluorescence.

Next, we considered a simple additive relation between the fluorescence signal in a well F_{real}, the actual reporter fluorescence $F_{reporter}$, and the inherent fluorescence background F_{BLK}:

$$F_{real} = F_{reporter} + F_{BLK} \tag{10.4}$$

As depicted in Figure 10.3, F_{real} is the input signal to the plate reader. In practice, separating the $F_{reporter}$ signal from the F_{BLK} noise is not feasible for each well. The background term F_{BLK} is usually estimated by having some wells with culture medium but no fluorescent reporter, and obtaining their averaged measured fluorescence. This common estimate is then used to retrieve the $F_{reporter}$ value for each of the wells in the plate. In our case, F_{BLK} was estimated as the median fluorescence value of wells containing only PBS buffer, acquired at the same gain (see Section 10.2.1). The function *checkblk.m* in the PLATERO toolbox provides fast analysis of the blank wells and prevents including potential outliers, which could distort the estimation of F_{BLK}.

Replacing Equation 10.3 in Equation 10.4, we can obtain the true fluorescence value of the signal of interest $F_{reporter}$:

$$F_{reporter} = (F_{observed} - F_{BLK,G}) \cdot e^{-b_1 \cdot G - b_2 \cdot G^2} \tag{10.5}$$

where $F_{BLK,G}$ refers to the F_{BLK} estimate at a certain gain level G.

The function *gaincfs.m* in the PLATERO toolbox computes the coefficients b_1 and b_2 in Equation 10.5. Further details about using this function can be found in the toolbox documentation.

Conversion of concentration units

To convert the arbitrary units of fluorescence to standard fluorescein concentration units, we assumed a linear model between the reporter fluorescence ($F_{reporter}$) and a concentration, C:

$$C = f_{UC}(F_{reporter}) = c_0 + c_1 \cdot F_{reporter} \tag{10.6}$$

where f_{UC} is the units conversion function, $F_{reporter}$ is obtained from Equation 10.5, c_0 is the intercept term of the linear model and c_1 is the slope of the linear model. Notice one might expect a calibration curve containing the (0,0) point (no fluorescence measured at 0 nM concentration, i.e. $c_0 = 0$). However, the coefficient c_0 is important to capture offset biases introduced by the plate reader.

Estimating coefficients c_0 and c_1 in Equation 10.6 is implemented in the function *cfcoeff.m* in the PLATERO toolbox. This function returns the estimated values and further information about the quality of the fitting.

Finally, the calibration model for the fluorescence concentration can be expressed as in Equation 10.7, where the correction of the gain effect (Equation 10.5) and the conversion of units to equivalent fluorescein concentrations (Equation 10.6) are included.

$$C = c_0 + c_1 \cdot (F_{observed} - F_{BLK,G}) \cdot e^{-b_1 \cdot G - b_2 \cdot G^2} \tag{10.7}$$

10.2.3 Measurement system analysis

Determining the quality of the measurement system is critical to trust the readings of any measurement system. This is done by evaluating the Repeatability and Reproducibility (R&RA) and the Linearity and Bias (L&BA) analyses. On the one hand, the L&BA assesses the variability of the predictions yielded by Equation 10.7 along the range of concentration values, i.e., how much variability should be expected in the predictions.

On the other hand, the R&RA allows us to quantify and decompose the uncertainty as the sum resulting from the different sources of variability, i.e., where is that variability in the predictions coming from? Since the (R&R) analysis will be performed with data already expressed as the predicted concentration (Equation 10.7), assessing the variability of the measurement system will include:

- variability due to lack of repeatability: "Do we get the same predicted concentration value if we measure the same well several times under identical conditions?"
- variability due to lack of reproducibility: "Do we get the same predicted concentration value if we compare values of the same well but measured with different gains?"

Thus, performing a Measurement System Analysis (MSA) that integrates both L&BA and R&RA lets PLATERO not only measure the uncertainty expected from the measurements but also check and validate the calibration model, comparing the reproducibility and repeatability terms.

Linearity and bias analysis

The accuracy of a measurement system (more specifically referred to as *bias*) reflects the difference between the observed measurements and the corresponding *true* values. Besides, the linearity of the measurement system reflects differences in bias over the range of measurements made by the system. We consider a simple model for bias:

$$Bias = \hat{C} - C_T = d_0 + d_1 \cdot C_T \tag{10.8}$$

Where $\hat{C}$ is the predicted concentration value given by Equation 10.7, C_T is the *true* value of the concentration obtained from a master or gold standard, d_0 the intercept, and d_1 the slope of the model.

PLATERO evaluates Equation 10.8 for I wells measured K times each, using J different device gains for each well and L different concentration levels. Therefore, an experiment will have $I \times J \times K \times L$ individual measurements. Using these values, the equation parameters are estimated using the functions provided by PLATERO (see Section 10.4.1). The number of degrees of freedom (DF) of the error after fitting Equation 10.8 is also stored as part of the model estimates because it will be used in Equation (10.12) to obtain the t-Student statistics ($t_{DF,\alpha/2}$)

Once the parameters from Equation 10.8 have been estimated, the linearity and bias contributions, $\%Linearity$ and $\%Bias$ respectively, are calculated to evaluate their relevance as:

$$linearity = Variation \cdot d_1 \tag{10.9}$$

$$\%Linearity = \frac{linearity}{Variation} \cdot 100 = d_1 \cdot 100 \tag{10.10}$$

$$\%Bias = \sum_{l=1}^{L} \frac{Bias_l}{Variation_l} / L \tag{10.11}$$

where $Bias_l$ is the average of Bias for the l-th concentration level, $Variation_l$ is the $6 \cdot \hat{\sigma}_{total}$ for the l-th concentration level (from the R&R analysis in Section 10.2.3), and L is the total number of concentration levels. All terms from Equation 10.8 to 10.11 can be estimated by the function *biasanalysis.m*.

Modelling the bias as in Equation 10.8 is also necessary to consider uncertainty in the predictions. The $(1-\alpha) \cdot 100\%$ confidence interval (CI_C) for a given concentration prediction $\hat{C}$ is calculated by function *cipred.m*, using the expression:

$$CI_C = \hat{C} \pm t_{DF,\alpha/2} \cdot s_{Bias} \tag{10.12}$$

Where $t_{DF,\alpha/2}$ is the $(1-\alpha)$ percentile of a t-Student distribution with DF degrees of freedom (degrees of freedom of the error from the linear model in

Equation 10.8) and s_{Bias} is the estimated standard deviation of $Bias$ (Equation 10.8).

Note that Equation 10.12 assumes that the predictions' uncertainty is the same for all concentrations (i.e., homoscedasticity). However, it is usual to find that the variance for the projections is different across concentrations (i.e., heteroscedasticity). In the case of having a proportional relationship between the error and the magnitude being measured, the heteroscedasticity can be easily neutralized by normalizing the bias values with the observed concentration level in the calibration data set using the following equation:

$$Bias \cdot \frac{1}{C_T} = d_0 \cdot \frac{1}{C_T} + d_1 \tag{10.13}$$

The scaling mentioned above affects the calculation of the confidence intervals. The Equation 10.12 is rewritten as:

$$CI_C = \hat{C} \pm t_{DF,\alpha/2} \cdot s_{Bias} \cdot \hat{C} \tag{10.14}$$

Where s_{Bias} is the estimated standard deviation of the scaled Bias, calculated with Equation 10.8, the last term in Equation 10.14 is the concentration value that undoes the scaling of the Bias and gives to the confidence interval, the amplitude corresponding to a particular concentration level. Ideally, this should be the right concentration level C_T. However, when the true concentration values remain unknown in model exploitation, the predicted concentration ($\hat{C}$) is used.

R & R analysis

Generally, the total observed experimental variability σ_T^2 is the sum of the *part-to-part* variability of the measured magnitude (σ_{P2P}^2), and the inherent variability arising from measurement errors (σ_{MS}^2). In our case, σ_{P2P}^2 arises when different plate wells containing the same concentration yield different fluorescence measurements. This can be explained by the stochastic component of the biochemical reactions within the well and the intrinsic experimental variability introduced during the preparation of the well plate. By contrast, σ_{MS}^2 comes from the measurement device (plate reader) [213], [214]. In turn, the measurement system has two sources of variability: *i)* the variance due to lack of repeatability σ_{Repeat}^2 (observed variability when repeating the same measurement), and *ii)* the variance coming from the lack of reproducibility,

σ^2_{Reprod} (observed variability when the same well is measured under different gains). This can be expressed mathematically as:

$$\sigma^2_T = \sigma^2_{P2P} + \sigma^2_{MS} = \sigma^2_{P2P} + \sigma^2_{Repeat} + \sigma^2_{Reprod} \tag{10.15}$$

The purposes of the R& R analysis (R& RA) are:

1. Determine how much of the total variability is generated by the measurement instrument.
2. Isolate the measurement system components of variability (i.e. σ^2_{Repeat} and σ^2_{Reprod}).
3. Assess whether the measurement instrument is suitable for the intended application.

The R& RA isolates all the components of variability from Equation 10.15 and estimates them individually using Design of Experiments (DOE) and Analysis of Variance (ANOVA).

In our work, the analyzed data come from a DOE with two factors: the well (W) and the device gain (G) at which the fluorescence values were measured. Consider the measurement instrument measures I wells at J gains for K repetitions. The statistical model that describes the sources of variability is represented as follows:

$$y_{ijk} = \mu + W_i + G_j + (WG)_{ij} + \varepsilon_{ijk} \begin{cases} i = 1, 2, \ldots, I \\ j = 1, 2, \ldots, J \\ k = 1, 2, \ldots, K \end{cases} \tag{10.16}$$

where y_{ijk} is an individual measurement of fluorescence, μ denotes the general mean, W_i, G_j and $(WG)_{ij}$ are independent random variables accounting for the effect of the well, the gain, and the interaction between well and gain, respectively, and ε_{ijk} is an independent random variable that represents the random error.

If each variable W_i, G_j, $(WG)_{ij}$ and ε_{ijk} is normally distributed variables with zero mean and variance defined as:

$$var(W_i) = \sigma^2_W \tag{10.17}$$

$$var(G_j) = \sigma_G^2 \tag{10.18}$$

$$var(WG_{ij}) = \sigma_{WG}^2 \tag{10.19}$$

$$var(y_{ijk}) = \sigma_T^2 = \sigma_W^2 + \sigma_G^2 + \sigma_{WG}^2 + \sigma^2 \tag{10.20}$$

Estimating each variance component using ANOVA as shown in [213] is possible. The variability of the wells σ_{P2P}^2 corresponds to σ_W^2, σ_{Repeat}^2 corresponds to the random error σ^2, and σ_{Reprod}^2 corresponds to $\sigma_G^2 + \sigma_{WG}^2$. Thus, the total variability σ_T^2 is estimated as:

$$\hat{\sigma}_T^2 = \hat{\sigma}_{P2P}^2 + \hat{\sigma}_{MS}^2 = \hat{\sigma}_{P2P}^2 + \hat{\sigma}_{Repeat}^2 + \hat{\sigma}_{Reprod}^2 = \hat{\sigma}_W^2 + \hat{\sigma}_G^2 + \hat{\sigma}_{WG}^2 + \hat{\sigma}^2 \tag{10.21}$$

Once we have estimated the variability of each isolated component, it is possible to calculate the respective contribution to the total variability:

$$Cont(\hat{\sigma}_{MS}^2) = \hat{\sigma}_{MS}^2 / \hat{\sigma}_T^2 \tag{10.22}$$

$$Cont(\hat{\sigma}_{Repeat}^2) = \hat{\sigma}_{Repeat}^2 / \hat{\sigma}_T^2 \tag{10.23}$$

$$Cont(\hat{\sigma}_{Reprod}^2) = \hat{\sigma}_{Reprod}^2 / \hat{\sigma}_T^2 \tag{10.24}$$

$$Cont(\hat{\sigma}_{P2P}^2) = \hat{\sigma}_{P2P}^2 / \hat{\sigma}_T^2 \tag{10.25}$$

Note that the R& R analysis will be carried out on the predicted concentration values from Equation 10.7. Hence, when we report the measurement system's performance, we will include the unit conversion operation as part of the measurement system, as illustrated in Figure 10.3. Thus, the results obtained in this analysis will serve as a part of the validation of the unit conversion model proposed in Equation 10.7.

10.3 Datasets

The following section includes a comparison between three different measurement setups:

- Plate reader 1 (*PR 1*): this setup yielded the data already used along the chapter in Sections 10.4.1 , 10.4.1 , 10.4.2 and 10.4.3. In this case, all concentrations, even those outside the calibration range used in Section 10.4.3, were included as part of the validation data set.
- Plate reader 2 experiment 1 (*PR 2 exp. 1*): fluorescence was measured with a different plate reader and modifying a part of the measurement procedure, i.e., the fluorescein dilutions were not stirred between repetitions. The model was fitted with 70% of the measurements and validated with the remaining 30%. In this case, the gains used to calibrate the model were between 60 and 90, and all concentrations were used for training and validation.
- Plate reader 2 experiment 2 (*PR 2 exp. 2*): for the third setup, PLATERO was executed with data from plate reader 2, but following the same measurement procedure as in *PR 1*. The model was fitted with 70% of the measurements and validated with the remaining 30%. In this case, the gains used to calibrate the model were between 60 and 90, and all concentrations were used for training and validation.

The concentrations and repetitions obtained in the experimental protocol above were arranged in a database using the following Design Of Experiments (DOE): for a crossed design with two factors involved (Well and Gain), the measurement instrument measured I wells at J gains for K repetitions or replicas. This way, one set of $I \times J \times K$ measurements was obtained for each one of the L concentration levels. Particularly, our database had 2048 observations. That is, $L = 4$ concentrations (3 fluorescein + 1 empty) $\times 16(I)$ wells $\times 4(J)$ gains $\times 8(R)$ measurement repetitions. Thus, each observation combined a concentration level, a well, the gain used for its acquisition, and the number of replicas. The resulting data and test software are publicly available as a Zenodo repository [215].

10.4 Results

This section goes through the different steps of PLATERO's calibration protocol in a tutorial-like style, showing how to apply it. To this end, we used two other plate readers. Notice the values of parameters in this section, and the results of the evaluation of the variability obtained, are particular to the plate readers we used. The section aims to show how PLATERO is applied and the results and analysis that can be drawn for its application.

The sections are divided into four main parts. Section 10.4.1 describes the results obtained from using PLATERO with a fluorescein calibration dataset, estimating the coefficients in Equation (10.7) and (10.12) to predict the fluorescein concentration of the plate wells. Next, Section 10.4.2 assesses the validity of the gain effect function f_G in Equation 10.3, showing the results obtained from a hypothetical scenario where an incorrect gain effect function f_G is assumed. Finally, Section 10.4.3 describes the results when the expressions fitted in Section 10.4.1 are applied to predict the concentration of samples with fluorescein concentration out of the calibration range. Finally, Section 10.4.4 gives the results obtained for a second plate reader, showing how in this case, PLATERO warned of problems related to the consistency of measurements caused by the device. The datasets we used in all cases and the Matlab scripts running PLATERO on these datasets can be obtained from the Zenodo repository [215].

10.4.1 Model building and validation

This is a two-step procedure, as depicted in Figure 10.3. It is the first task a user of PLATERO must carry out before exploiting the calibration model with fluorescence measurements from cells expressing any fluorescent reporter, in this case, Green Fluorescent Protein (GFP). The experimental calibration protocol followed the steps detailed in Section 10.2.1, yielding a data set with fluorescein measurements.

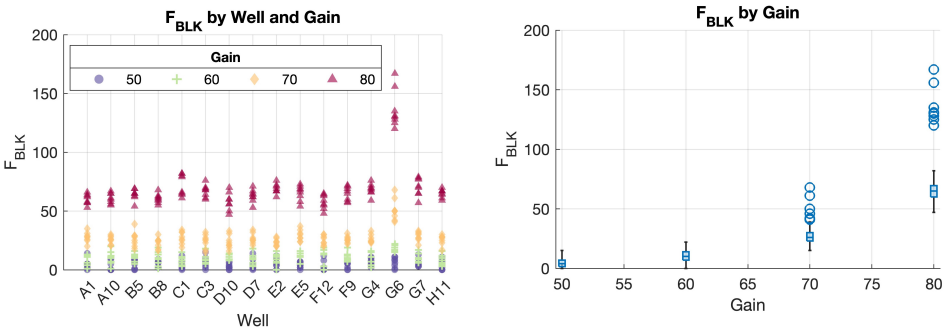
We used the dataset obtained from the first plate reader [215]. This dataset was divided into two subsets: one for the Model Building process and another for the Model Validation. In the following sections, we will show the results obtained with our particular data set in the Model Building (Section 10.4.1) and Model Validation (Section 10.4.1) steps.

Model building

The first step a user of PLATERO must carry out is the Model Building. In our case, the model-building dataset contained approximately 70% of the wells with fluorescein, chosen by random selection. That is, we used 1056 of $F_{observed}$ (3 concentrations×4gains×11wells×8 repetitions). Random selection prevents potential location effects due to the selection of wells in a specific row or column order.

First, the gain effect model (f_G) was fitted for each set of four $F_{observed}$ values acquired from each well at each repetition. Thus, 264 estimates (N=8 repetitions×11 wells×3 concentrations) for the coefficients b_1 and b_2 in Equation 10.5 were obtained. This approach was preferred instead of having one single estimate for each parameter in f_G because we wanted to assess the stability of the $F_{BLK,G}$, b_1 and b_2 parameters from f_G.

The first step for this assessment was to estimate the $F_{BLK,G}$ terms for each gain level. At this point, it is important to acknowledge that working with real data sets may imply the existence of potential outliers due to measurement errors or other issues.



(a) F_{BLK} values by wells and gain.

(b) F_{BLK} values by gain.

Figure 10.4: Fluorescence values for the wells without fluorescein ($F_{BLK,G}$) used to measure the reader bias by wells and gain (a) and just by gain levels (b).

As it can be seen in Figure 10.5a, some observations (well G6) had higher fluorescence measurements than the majority of the values acquired at the same gain. To prevent the influence of these potential outliers in the estimation of $F_{BLK,G}$ in Eq. 10.5, they were calculated as the median fluorescence of the empty wells for each gain level. These were used to correct the additive noise introduced by the medium.

Afterwards, to check if b_1 and b_2 were significantly consistent for all levels of concentration, a nested arrangement employed to estimate components of variance is used, called hierarchical design [216]. Thus, the different wells are hierarchically subsumed under the levels of concentration. The associated ANOVA table is shown below.

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Concentration	0	2	1.1491e-06	0.13	0.874
Well(Concentration)	0.00035	30	1.16589e-05	1.37	0.1049
Error	0.00197	231	8.52553e-06		
Total	0.00232	263			

Constrained (Type III) sums of squares.

(a) ANOVA result for the b_1 coefficient.

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Concentration	2.19985e-10	2	1.09993e-10	0.23	0.795
Well(Concentration)	1.92842e-08	30	6.42808e-10	1.34	0.1191
Error	1.10662e-07	231	4.79055e-10		
Total	1.30166e-07	263			

Constrained (Type III) sums of squares.

(b) ANOVA result for the b_2 coefficient.

Figure 10.5: ANOVA tables assessing the stability of values obtained for the two coefficients in Equation 10.5.

In Figure 10.5, it can be seen that assuming a 5% type I risk α, there were no statistically significant differences between coefficients fitted with data from different concentrations (p-values > 0.05 for the concentration factor). As a result, this validated the third assumption of the same gain effect for all concentration levels. The final b_1 and b_2 values were calculated as the median values of the overall coefficients. The global coefficient values are shown in Table 10.1.

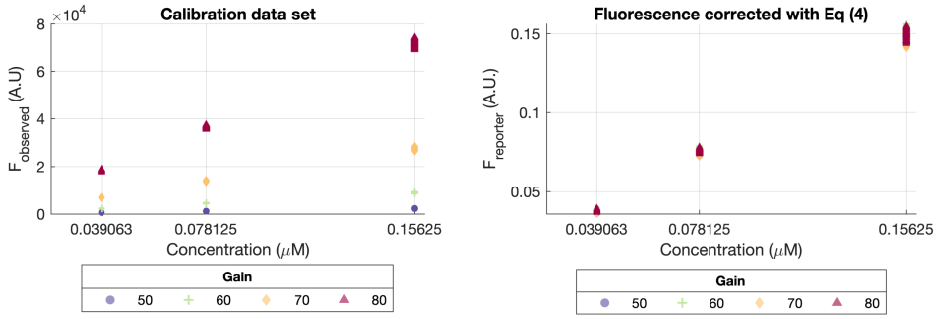
Table 10.1: Estimated coefficients for the gain effect model ($N = 264$)

Coefficients	Median	Interquartile range
b_1	0.24298	0.0035
b_2	$-9.933 \cdot 10^{-4}$	$2.5933 \cdot 10^{-5}$

Median values for b_1 and b_2 (Tables 10.1 and 10.2) were finally considered as the global estimates of the gain effect in the fluorescence measurements $F_{reporter}$ from Equation 10.5. These values are part of the final calibration model (Equation 10.26).

Table 10.2: Coefficients of the units conversion model.

$F_{BLK,G}$				b_1	b_2	c_0	c_1	s_{Bias}	DF
G=50	G=60	G=70	G=80						
4	10	26	65	0.243	$-9.933 \cdot 10^{-4}$	$-1.1185 \cdot 10^{-3}$	1.0576	0.0225	1054



(a) Scatter plot of concentration vs. the raw original values ($F_{observed}$).

(b) Scatter plot of concentration vs. corrected fluorescence values ($F_{reporter}$).

Figure 10.6: Calibration data set before and after correction with Equation (10.5).

Figure 10.6 depicts the difference between the data before and after removing the gain effect f_G (left and right plots, respectively). In Figure 10.6a, the observed fluorescence ($F_{observed}$) at different gain values (ranging from G=50 to G=80) has a clear non-linear effect concerning each fluorescein concentration. On the contrary, this effect disappears in Figure 10.6b, where all data points describe the same linear relationship between the fluorescence $F_{reporter}$ at each concentration value, regardless of the gain.

The second step is to fit the units conversion model, f_{UC}, to obtain fluorescence data expressed in standard fluorescent units. PLATERO automatically fits the units conversion model using the function f_{UC} and the estimated parameters for Equation 10.6. The resulting model for this first plate reader was:

$$\hat{C} = -1.1185 \cdot 10^{-3} + 1.0576 \cdot (F_{observed} - F_{BLK,G}) \cdot e^{-0.24298 \cdot G + 9.933 \cdot 10^{-4} \cdot G^2} \quad (10.26)$$

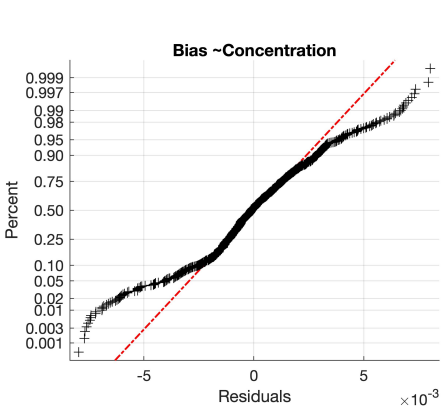
where $F_{BLK,G=50} = 4$, $F_{BLK,G=60} = 10$, $F_{BLK,G=70} = 26$ and $F_{BLK,G=80} = 65$.

Now, for a given value of fluorescence ($F_{observed}$) measured at any gain (G), we can provide a prediction of the fluorescence concentration $\hat{C}$ in standard

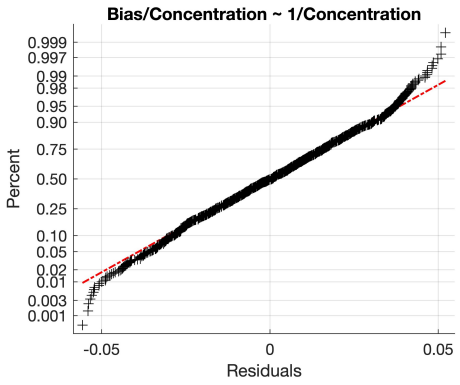
units. However, the fluorescence values in Figure 10.6 showed some variability despite belonging to the same concentration or gain level for either $F_{observed}$ (Figure 10.6a) or $F_{reporter}$ (Figure 10.6b). This resulted from the inherent experimental variability, and we must consider it to provide more confident predictions. To do so, we applied Equation 10.26 to the Model Building dataset and analyzed the error (bias) between the estimated concentration $\hat{C}$ and the true concentration values C_T.

As shown in Figure 10.7a there was a clear heteroscedasticity (i.e., unequal level of dispersion along the range of a variable) in the residuals. This is reflected by bias values being more dispersed for higher concentration values. Indeed, this could affect the quantification of the uncertainty: an increased dispersion of the residuals, along with concentration values, should be reflected accordingly, yielding higher levels of uncertainty for the predictions of higher concentration values. Nonetheless, as the relationship between the error and the magnitude being measured is proportional, heteroscedasticity can be easily neutralized by Equation 10.13 (Figure 10.7b).

Figure 10.7 illustrates the normal probability plots of the residuals after fitting the bias regression model using bias values without (Figure 10.7a) and with (Figure 10.7b) the scaling from Eq. 10.13.



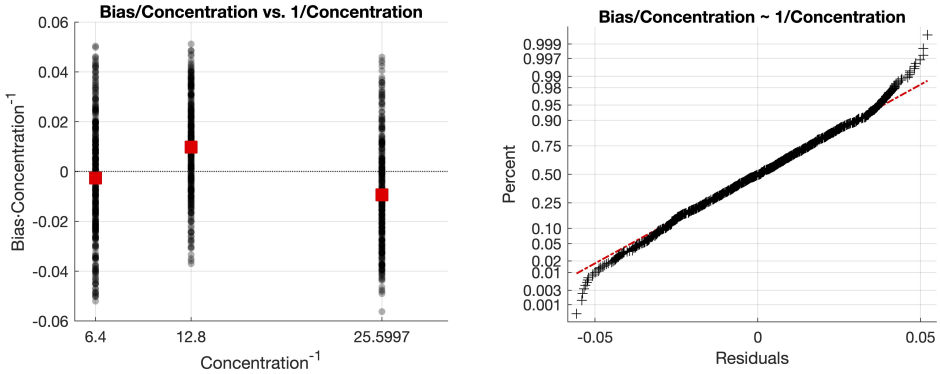
(a) Normal probability plots of the raw residuals.



(b) Normal probability plots of the scaled residuals.

Figure 10.7: Residual analysis with Normal probability plots of the residuals quantifying the uncertainty without including (a) and including (b) the normalization by the concentration values from Eq. 10.13

As it can be appreciated, the right normal probability plot, which included the scaling mentioned above of the residuals by the concentration, fitted the line of



(a) Scatter plot of the scaled bias values, where black dots represent every scaled bias value and red squares are the mean scaled bias value for each concentration level.

(b) Normal probability plot of the scaled residuals. The red dashed line represents the curve described by a perfect normal distribution. Black crosses are the scaled residuals.

Figure 10.8: Results of the Bias and Linearity analysis with the scaled residuals obtained using Equation 10.13 with observations in the Model Building subset.

the normal distribution. Thus, the s_{Bias} term was estimated as the standard deviation of the scaled bias.

Figures 10.8a and 10.8b illustrate the dispersion and the normal probability plot of the scaled residuals, respectively, obtained with Equation 10.13. As it can be seen, the normal probability plot fits the line of the normal distribution, validating the use of the scaled bias model from Equation 10.13.

The last step is to estimate the s_{Bias} term as the standard deviation of the scaled bias. This resulted in the following expression (for the case of our particular calibration dataset) to compute the confidence intervals for the estimated concentration $\hat{C}$, at a $(1-\alpha)\cdot 100\%$ confidence level:

$$CI_C = \hat{C} \pm t_{1054,\alpha/2} \cdot 0.0225 \cdot \hat{C} \tag{10.27}$$

where $\hat{C}$ is the concentration prediction from Equation 10.26, and $t_{1054,\alpha/2}$ is a t-Student statistic automatically calculated by the *cipred.m* function.

To sum up, Table 10.2 contains all the estimates for the parameters obtained in the Model Building step.

Model validation

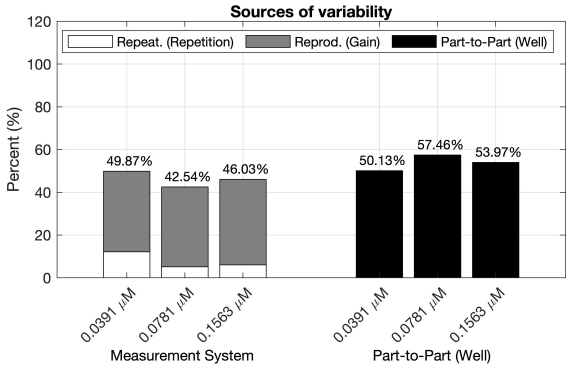
The calibration model fitted in the previous Section 10.4.1 was used to predict the fluorescein concentration levels using the Model Validation data set (see Figure 10.3). Particularly, these predicted concentration values were used:

- to carry out an R&R analysis assessing the sources of variability affecting the predictions for each concentration level;
- to perform a B&L analysis assessing the variability of the scaled residuals across the range of concentration levels; and
- to validate if the confidence intervals of such predictions contained the true fluorescein concentration value.

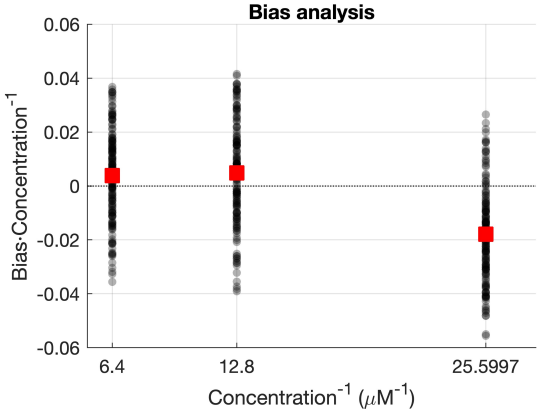
R&R analysis. These analyses (one for each concentration level) seek to evaluate how the experimental conditions (well location, well volume, the gain of the device, or the number of repetitions of a measurement) relate to the differences observed between the values of the predictions of concentration, $\hat{C}$.

The R&R analysis decomposes the total variability seen in $\hat{C}$ into different sources of variability as the result of different experimental factors: (1) the "Part-to-Part" component (σ^2_{P2P}) associated with differences among the wells, (2) the "Reproducibility" component (σ^2_{Reprod}) generated by the differences related to measuring at different gain values, and (3) the "Repeatability" component (σ^2_{Repeat}) that is associated to differences seen between repetitions of the measurements.

Figure 10.9 shows the distributions of these three sources of variability for the concentrations analyzed with the data corresponding to the Plate Reader 1. The part-to-part variability contributed more to the total variability than the reproducibility component for all concentrations. This means that the variability of the predictions of concentration $\hat{C}$ for the same well, even if taken at different gain levels, is lower than that of the predictions obtained for additional wells measured at the same gain. In other words, the gain effect has been successfully modelled and removed. Interestingly, after the unit conversion process, we can still distinguish among the wells with the same fluorescein concentrations.



(a) R&R analysis for the validation dataset. The total variability is decomposed into the Part–to–Part (σ^2_{P2P}) and the Measurement System contribution. In turn, the Measurement System contribution contains both Repeatability plus Reproducibility values. The Part-to-Part variability represents the differences between $\hat{C}$ values for different wells with the same fluorescein concentration. The Reproducibility (Gain) variability (σ^2_{Reprod}) represents the differences between concentration values for the same measurement recorded at different gain levels.



(b) L&B analysis for the validation data set, illustrating the scattering of the scaled bias values (black dots) for each Concentration^{-1} level. The plot also illustrates if there is a tendency between average values of the scaled bias (red squares) and the Concentration^{-1}.

Figure 10.9: Analyses of the validation data set.

Table 10.3: Performance metrics using the calibration and validation sets after using the exponential and linear f_G (Equation 10.5 and 10.28, respectively).

Data set	MSE	MinRE (%)	MaxRE (%)
Calibration	$5.6728 \cdot 10^{-6}$	$0.0040 \cdot 10^{-2}$	5.6197
Validation (exponential f_G)	$3.4285 \cdot 10^{-6}$	$0.0017 \cdot 10^{-5}$	5.5619
Validation (linear f_G)	0.0028	9.12	81.76

L&B analysis. The L&B analysis considers the relationship between the scaled *Bias* (Equation 10.13), and the predicted fluorescence concentration $\hat{C}$. This was necessary to evaluate the model proposed in Equation 10.8. Studying the statistical features of the prediction error across all concentration levels was necessary to consider any corrections to the proposed model that might be required, as shown in the following results. Figure 10.8b plots how, for the Plate Reader 1 we used, the $Bias/Concentration$ values are approximately symmetrically scattered around zero.

As seen from the analysis, there was no linear relationship between the scaled bias and the inverse of the concentration level. This meant the model had properly captured the relationship between the real concentration and the predicted one. Hence, the bias term did not contain any relevant information missed by the model but noise. This was quantitatively represented by the low contribution (in percentage) of both the linearity (1.5218%) and the bias (7.5792%) terms on the total variability of the scaled residuals.

In summary, the residuals' linearity and bias are irrelevant to the total residual variability. Therefore, no important information remained on the prediction errors, i.e., the model used for the prediction was valid. This was also an indicator of the consistency of the functions f_{UC} and f_G for all the concentrations within the experimental range of values.

Confidence intervals. Finally, it is important to assess the prediction error. Table 10.3 lists some common metrics for the prediction error: the means squared error (MSE) and the minimum and maximum relative errors. These metrics can be compared to the results obtained from different datasets or with varying proposals of unit conversion models.

However, the metrics used in Table 10.3 are relative and do not explicitly validate if such error values are small enough to assume that the concentration predictions $\hat{C}$ are close enough to the ones known *a priori* C_T. To this end, PLATERO's last step in the validation process is the analysis of the confidence

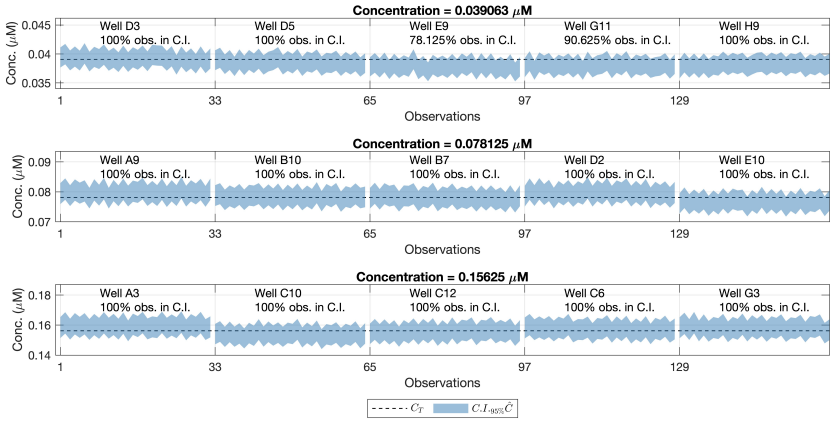


Figure 10.10: Confidence Intervals (95%) for the concentration values (shaded area) and reference concentration values (dashed line). Each row refers to one concentration level, the same as in the Model Building step, measured at the same gains but from different wells. Each column corresponds to a different well and location on the 96-well plate. For each well, we had 32 (8 repetitions $\times$ four gains) predictions of $\hat{C}$.

intervals for the predictions. Specifically, the confidence intervals of every predicted concentration $\hat{C}$ were obtained from Equation 10.27, using the $t_{DF,\alpha/2}$ and s_{Bias} from the Model Building step.

If the confidence interval contains the true concentration value C_T, the prediction will be valid, i.e., the predicted $\hat{C}$ value is close enough to the true concentration. Figure 10.10 depicts the true concentration values for every data as dashed lines with the confidence region (blue shaded area) limited by the extremes of the confidence intervals for the predictions. These values were calculated at a 95% confidence level ($\alpha = 0.05$). Figure 10.10 shows that the confidence intervals for the predictions contain the true concentration value for nearly 95.3125% of the observations in the validation data set.

Notice that wells E9 and G11 (corresponding to $C_T = 0.039063$) had less than 95% of their confidence intervals containing the true concentration value.The key aspect of dealing with this variability is assessing whether the differences between the theoretical concentration value and the obtained one are big enough to consider that a certain well is not a valid replicate for a given concentration level.

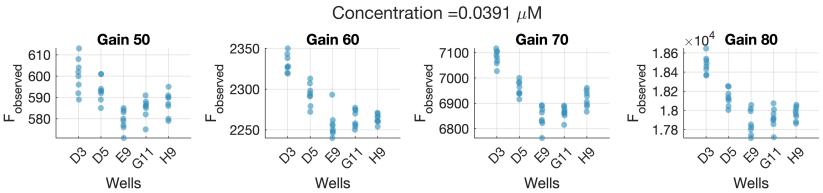


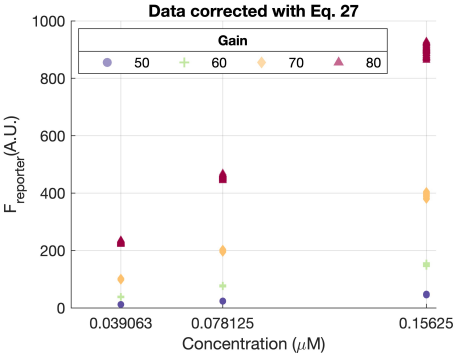
Figure 10.11: Observed fluorescence values ($F_{observed}$) for all wells of the validation data set containing the $C_T = 0.039063\mu M$ concentration level, measured at four different gain levels. The horizontal axes indicate the specific well yielding the $F_{observed}$ measurements.

We did not consider the results obtained in wells E9 and G11 invalidated the calibration model. Instead, we hypothesized that the real concentration on those wells was not exactly the theoretical one (C_T). This was probably the result of the inherent variability of the experimental procedure. The introduction of the human factor leads to small differences between the theoretical and the real fluorescein concentration deposited in the wells.

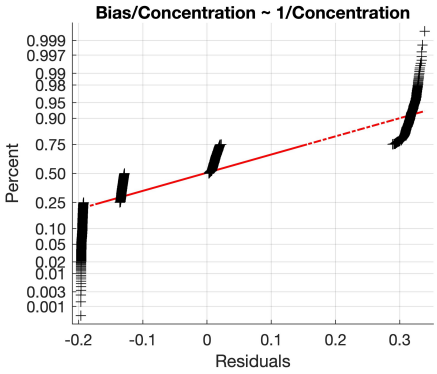
This hypothesis was supported by the fact that the same bias was seen in the confidence intervals from wells E9 and G11 in Figure 10.10. This was also appreciated in the raw fluorescence values $F_{observed}$ obtained for those wells, as seen in Figure 10.11. This means that the difference between the real and the assumed concentration was already present in the sample, and the calibration model did not introduce it.

It is worth mentioning that this variability between wells of the same concentration was already pointed out by the high contribution of the part-to-part (σ^2_{P2P}) variability source in the R&R analysis (Figure 10.9a). Moreover, this also explains the lower average bias for the lowest concentration level, which is the highest $1/Concentration$ level obtained in the L&B analysis (Figure 10.9b).

Finally, the low variability of the reproducibility term ($\sigma^2_{Reprod} < \sigma^2_{P2P}$), together with the soft contributions of the linearity and bias terms, and the high percentage of confidence intervals for the expected concentration containing the true concentration value statistically validate the proposed units conversion model and the assumptions from Section 10.2.2.



(a) $F_{reporter}$ values assuming a linear effect of the gain on the fluorescence measurements. Marker shapes and colours depend on the Gain level G used.



(b) Normal probability plot of the scaled residuals after using the f_G expression from Equation 10.28 to predict the concentration.

Figure 10.12: Results obtained after using Calibration data set before and after correction with Equation (10.28).

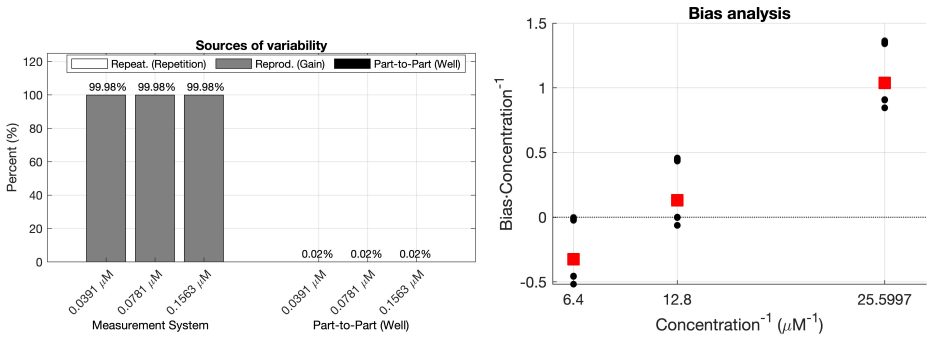
10.4.2 Assessment of the gain effect function

This section will address the question: what if we used an incorrect f_G expression? This is a very legitimate question since the plate reader manufacturer does not formally present the gain effect. In this section, we assumed a widespread correction expression, assuming a linear development of the gain in the fluorescence (Equation 10.28).

$$F_{reporter} = \frac{F_{observed} - F_{BLK}}{G} \tag{10.28}$$

Figure 10.12a shows the result after applying Equation 10.28 to the $F_{reporter}$ values. There are still large differences (up to 3 orders of magnitude) among the values for the same concentration caused by measuring them at different gain levels. The result seen in Figure 10.8b is very different from the one in Figure 10.12b, where values acquired at different gain values are almost indistinguishable after correcting them with the model Equation 10.5. Therefore, the gain effect and its linear form remain on the data after using Equation 10.28.

The same conclusion is attained by inspecting Figure 10.12b, which illustrates the normal probability plot of the residuals after assuming a linear effect of the gain on the measurements. As seen, residuals form small groups, contrary



(a) R&R analysis. **(b)** L&B analysis.

Figure 10.13: Results of the Measurement System Analysis obtained after using Calibration data set before and after correction with Equation (10.28).

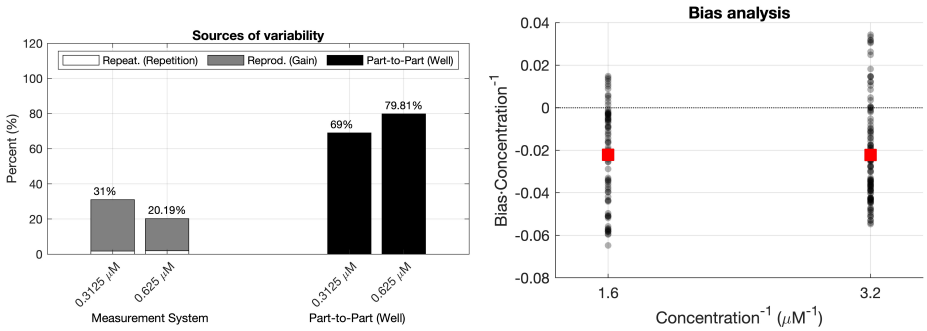
to the distribution seen in Figure 10.12b, indicating that the gain effect has not been completely removed from the observations.

Both results would be enough to validate the calibration function f_G given by Equation 10.3 and 10.5. However, R&R and B&L analysis was carried out for the exponential gain effect.

Figure 10.13a illustrates that the measurement system is the main source of variability among measurements. Specifically, the "Reproducibility (Gain)" term is the one constituting almost 100% of the total variability, indicating that the gain effect correction is not appropriate.

The L&B plot (Figure 10.13b) clearly shows non-null bias values and a linear relationship between the scaled bias and the inverse of the concentration. This outcome differs substantially from the one seen in Figure 10.13, suggesting an incorrect assumption of the gain function f_G.

Finally, the metrics shown in the third row of Table 10.3 also provide a comparative corroboration that concludes the improvement of modelling the gain effect as an exponential relationship among the observed fluorescence ($F_{observed}$), the one emitted by the reporter ($F_{reporter}$), and the gain used for the measurements (G) (Equation 10.5). These results consistently prove that using Equation 10.28 is not properly modelling the gain effect on the measurements.



(a) R&R analysis. **(b)** L&B analysis.

Figure 10.14: Results of the Measurement System Analysis for a dataset with concentration values outside the calibration range used for the model building.

10.4.3 Extrapolation to other concentrations

We also addressed the issue of the goodness of the calibration model outside the concentration range used for the Model Building and Validation steps. Therefore, we used different and higher fluorescein concentrations as a new dataset. The concentrations we used to test the extrapolation ability of the model were 0.3125 μM and 0.625 μM. They were not part of the calibration procedure because the fluorescence measurements saturated at the gains of 70 and 80.

Figure 10.14 shows the R&R and L&B analysis results for these concentrations. As we can see in Figure 10.14a, the reproducibility (Gain) contribution is still lower than the part-to-part (Well) assistance for both concentrations. Thus, the proposed f_G and f_{UC} functions are still valid to convert from arbitrary fluorescence units to standard concentration ones.

However, Figure 10.14b shows a negative bias on average for both concentrations. The same negative bias is also noticed in the confidence intervals for the predicted concentrations of each well (Figure 10.15). This means that the expected concentration is consistently lower than the theoretical concentration.

In Figure 10.15, we can see that some of the wells in the 96-well plate with less than 95% of the confidence intervals contained the reference concentration value from the fluorescein pattern. Nevertheless, as said before, the same bias affecting the predictions can be appreciated in the raw fluorescence values (see Figure 10.16).

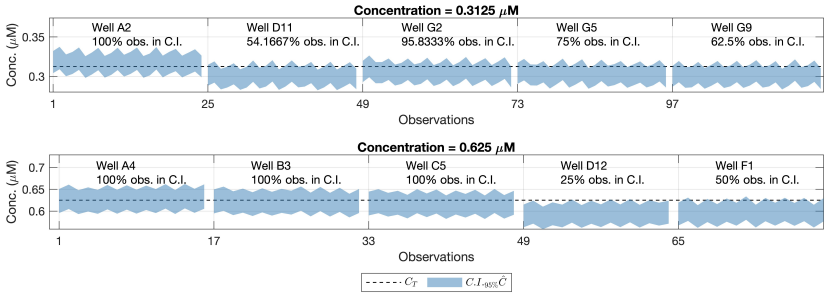


Figure 10.15: 95% Confidence Intervals for the concentration values outside the model buildings' concentration range (shaded area) and reference concentration values (dashed line). Each row refers to one concentration level. Each column corresponds to a well's location on the 96-well plate.

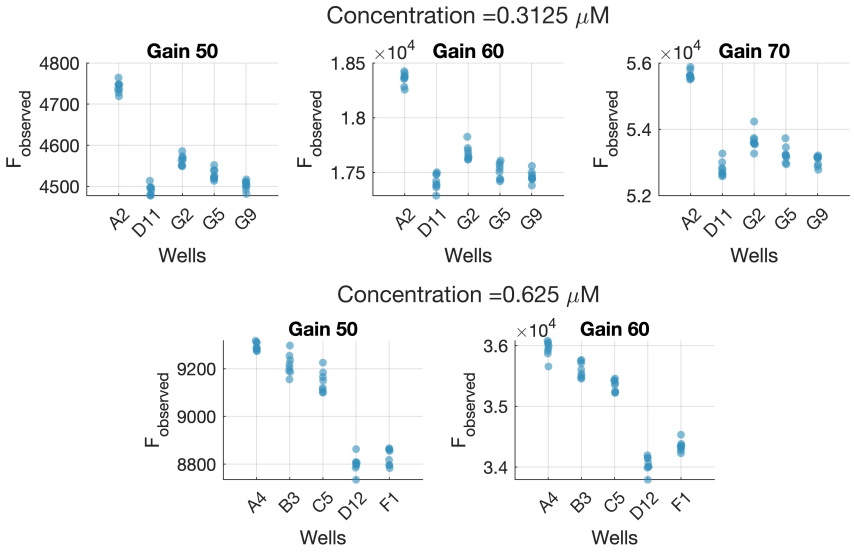


Figure 10.16: Observed fluorescence values ($F_{observed}$) for all wells in the 96-well plate with concentration values outside the calibration range used for the model building. Each row refers to one concentration level. Each plot contains the $F_{observed}$ values recorded at a certain Gain level. The horizontal axes indicate the specific well identity (ID) yielding the $F_{observed}$ measurements.

This result may be explained by the fact that most of the variability is due to the part-to-part term (Figure 10.14), which is close to the 80%, whereas, for the concentrations used for the model calibration, it was below the 60%. This increment in the part-to-part variability could be due to outlying wells whose concentration differs from the rest with the same expected concentration.

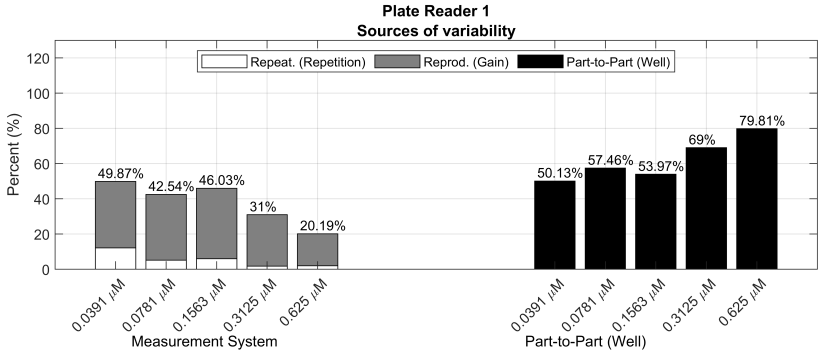
In such a case, one would expect a behaviour similar to one of the wells D12 and F1, with less than the 50% confidence intervals containing the reference concentration value. This consistent bias seen for wells D12 and F1 remains independent of the gain associated with the measurements, which suggests that the error is associated with the real fluorescein concentration deposited in those specific wells. This result also illustrates that confidence intervals for the predicted concentration could be used to detect atypical wells in the plate during the calibration step. This also might help to establish statistically significant differences between the predicted concentration within different wells.

From the results above, we could say that the theoretical expression for the units conversion model could be valid to extrapolate to concentration values outside the calibration range. However, as expected when a model extrapolates, this has some drawbacks and limitations. Consequently, we recommend including these concentrations for the model building and estimating the parameters to estimate the uncertainty better.

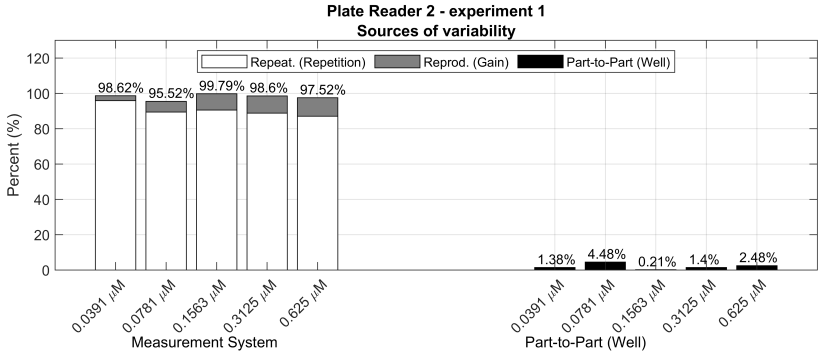
10.4.4 Comparison between plate readers

As Section 10.3 explained, three datasets were obtained under different measurement setups. The appendices contain the results obtained after fitting each one of the models. All the data and software to run the tests and get the results can be obtained from [215]. As it can be seen, an acceptable percentage of confidence intervals contained the theoretical concentration (91.62% for *PR 1*, 97.5% for *PR 2 exp. 1*, and 96.38% for *PR 2 exp. 2*). The contributions of the bias and linearity terms to the residuals model are also acceptable for all of them, at maximum, 10%. However, the R&R results (Figure 10.17 and Table 10.4) point out relevant differences between the measurement systems.

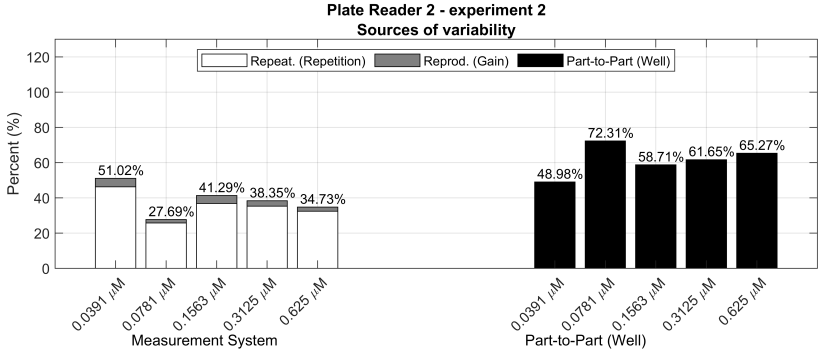
In Figure 10.17, the first contribution plot from *PR 1* illustrates the measurement system's expected and proper performance. This good quality is reflected by both Repetition's contributions being the minimal sources of variability and by Measurement System's contributions being smaller than Part-to-Part's for all concentrations.



(a) Plate Reader 1 Experiment 1.



(b) Plate Reader 2 Experiment 1.



(c) Plate Reader 2 Experiment 2.

Figure 10.17: R&R analyses for the validation datasets measured with different plate readers and experimental procedures.

Table 10.4: Variance in measurements obtained with different plate readers.

Concentration	Source	PR 1	PR 2 exp. 1	PR 2 exp. 2
C = 0.0391	Reprod. (Gain)	$2.14E-07$	$2.96E-07$	$6.89E-07$
	Repeat. (Repetition)	$6.92E-08$	$1.05E-05$	$6,72E-06$
	Part-to-Part. (Well)	$2.85E-07$	$1.50E-07$	$7.12E-06$
C = 0.0781	Reprod. (Gain)	$1.00E-06$	$2.35E-06$	$1.33E-06$
	Repeat. (Repetition)	$1.39E-07$	$3.43E-05$	$1.71E-05$
	Part-to-Part. (Well)	$1.54E-06$	$1.72E-06$	$4.81E-05$
C = 0.1563	Reprod. (Gain)	$3.39E-06$	$1.15E-05$	$5.28E-06$
	Repeat. (Repetition)	$5.11E-07$	$1.13E-04$	$4.31E-05$
	Part-to-Part. (Well)	$4.58E-06$	$2.61E-07$	$6.88E-05$
C = 0.3125	Reprod. (Gain)	$1.89E-05$	$4.85E-05$	$1.65E-05$
	Repeat. (Repetition)	$1.16E-06$	$4.39E-04$	$1.91E-04$
	Part-to-Part. (Well)	$4.47E-05$	$6.93E-06$	$3.34E-04$
C = 0.625	Reprod. (Gain)	$5.03E-05$	$2.11E-04$	$6.69E-05$
	Repeat. (Repetition)	$5.41E-06$	$1.75E-03$	$9.11E-04$
	Part-to-Part. (Well)	$2.20E-04$	$4.99E-05$	$1.84E-03$

This desirable performance seen for *PR 1* measurements is not maintained in the *PR 2* ones. In general, *PR 2* shows a clear increase in the repeatability contribution to the total variability. In the first experiment (*PR 2 exp. 1*), the well plates were not stirred between repetitions, contrary to the indications of PLATERO's experimental standard procedure. This small difference is captured by the R&R analysis, showing an unacceptable contribution of the Measurement System to the total variability generated by a great variability between repetitions.

In the second experiment, *PR2, exp. 2*, the experimental protocol was executed correctly, and the balance between Measurement System's and Part-to-Part's contributions are reestablished, resembling more the values obtained for *PR 1*. However, the Repeatability contribution to the total variability is still higher than for *PR 1*. Table 10.4 shows the absolute values of these contributions. As it can be seen, the absolute values of the Gain's variance term are similar between *PR 1* and *PR 2 exp. 2*, but the absolute value of the variance between repetitions is persistently higher for *PR 2*. This result suggests that the plate reader *PR 2* would require a technical revision to evaluate if low maintenance causes the low repeatability of its measurements. Otherwise, if this is the expected variability for measurements provided by *PR 2*, this outcome would be a quantitative measure of the difference in quality between plate readers.

In summary, the results yielded by the R&R analysis may inform users about the need for maintenance for plate readers and the relative quality of measurements between them. Moreover, as seen for the case of *PR 2 exp. 1*, the

R&R analysis should always be included as part of the calibration model's validation, for they may also detect problems with the experimental protocol.

10.5 Conclusions

In this work, we propose a unit conversion model that enables users of fluorescence plate readers to obtain comparable results. The conversion model is a composition of two functions: the gain effect function (f_G) and the units conversion function (f_{UC}). The uncertainty around the estimates of the model and its parameters may vary depending on the machine being used and the user intervention during the experimental protocol to get empirical data. For this reason, the second pillar of this protocol consists of a Measurement System Analysis (MSA) via an R&R and a B&L analysis.

Three real datasets from two different plate readers were obtained following the proposed procedure and used to assess the performance of the calibration model. The results in Section 10.4.1 showed that over 95% of confidence intervals for the predicted concentration contained the true concentration value. Furthermore, as seen in Figures 10.10 and 10.15, the confidence intervals can be used to detect wells differing from the expected concentration value, assessing the quality of the experimental procedure quantitatively. Additionally, Section 10.4.2 illustrates how the protocol would warn users about the assumption of incorrect gain effect functions. Section 10.4.3 shows how the model performs when it is used with concentration values above the concentration range of values used to fit the calibration model. Although the model seems to extrapolate fairly well, we would encourage users to re-apply the protocol and assess the differences in the model parameters when a different range of concentrations is used. Finally, Section 10.4.4 illustrates the need to include the R&R analysis, informing users about the data quality and potential adjustments required to maintain the quality of plate readers' measurements or fix issues during the experimental protocols used to gather the data.

Beyond the analytical expressions proposed in this chapter for the gain effect (f_G) and for the conversion to concentration units (f_{UC}), the main contribution of this work is the proposal of a whole methodological framework with a solid statistical basis that can and should be applied to test future proposed analytical expressions. Thus, using alternative valid analytical proposals for converting units would not devalue or invalidate the methodology and tools presented in this chapter. Rather than offering a unique and universal expression usable with all plate readers and in all experimental conditions, this work

aims to pave the way towards standardising fluorescence data and expanding the limits of the comparable scenarios in posterior analyses.

In conclusion, proposing a single analytical expression to predict concentration from fluorescence values, estimate the uncertainty expected on such predictions and assess the variability of the plate reader's measurements bring a solid statistical foundation to the presented work. As far as we know, integrating these statistical tools is a differential aspect compared to other proposals to correct the gain effect or convert fluorescence values to concentration values. Moreover, calculating confidence intervals for the predicted concentration constitutes a validation tool for the proposed unit conversion model and its overall assumptions.

Moreover, PLATERO's approach is so general that it can be used to estimate fluorescence concentrations at any wavelength, provided that the appropriate reference fluorescence solution is chosen. The underlying model can be extended not only for fluorescence measurements but also for absorbance or luminescence measurements.

We believe that tools such as the PLATERO (Plate Reader Operator) toolbox will play a key role in Synthetic Biology, enabling the proper comparison of databases coming from different experimental settings, the validation of the quality of the acquired experimental data, and the extension of the measurement system range to a broader one being able to detect more subtle signals but at the same time seeing strong signals.

Part IV

Epilogue

Chapter 11

Conclusions

In this thesis, several problems commonly found in biomedical engineering were addressed using the Statistical Machine Learning philosophy, which focuses not solely on the use of Machine Learning techniques but also on the general knowledge that can be obtained from their resulting models. Additionally, new methodologies were proposed to deal with challenges closely linked to the 4.0 paradigm, where the volume and heterogeneity of data are increased. The context, challenges and relations between the 4.0 paradigms in Medicine, Healthcare and Industry and the philosophy of Statistical Machine Learning were discussed throughout Chapters 2 and 3. The different contributions distinguished two parts after an initial introduction exposing key concepts and implications. Part II: New Methodological Proposals studied the problem of simultaneous outliers detection and missing data imputation. In Part III: Applications in Biomedical Engineering, several issues related to Biomedical Engineering were tackled with a Statistical Machine Learning philosophy, including healthcare process improvement, searching for biomarkers integrating different -omics layers, confectioning a mortality predictive model and proposing a standard framework for fluorescence data normalisation, materialised in a protocol to calibrate conversion models for plate readers models within the context of systems biology. This last part summarises the thesis's conclusions, relevance and future lines.

11.1 Meeting the objectives

This section outlines the main remarks in this document to demonstrate that the objectives have been met.

Objective 1: Propose, implement and deploy new methodologies for statistical machine learning

The most transcendent feature of the Medicine, Healthcare and Industry 4.0 paradigms is probably the reach of data generation to the user level. Data generation is more abundant than ever, and it is expected to expand with a compound annual rate of 36% growth in the volume of healthcare data produced between 2018 and 2025 [22]. This growth comes also entangled with diversifying the sources pouring data, with medical data expected to expand in volume and variety in the upcoming years. Consequently, the expectation is to have tools to deal with a first exploratory analysis of this data as efficiently as possible: automatically squeeze out as much information as possible. From a technical side, such heterogeneous data are often incomplete or unbalanced, which may affect the use of advanced technology for its analysis. Coming across this "dirty" data to obtain a clean and usable data set can need domain knowledge, which can be challenging to integrate quantitatively and qualitatively without tools providing an Exploratory Data Analysis (EDA).

Chapter 5 presented a novel framework to define outliers based on their properties concerning an original PCA model. This framework aims to bring tools to compare objectively and, in a standardised way, to assess the performance of different methods meant to deal with outliers during these first steps typical of the EDA stage when dealing with real datasets. This framework was used in Chapter 6 to simulate outlying observations with specific characteristics.

In Chapter 6, the RadarTSR algorithm was introduced as a versatile method capable of effectively handling cellwise outliers, rowwise outliers, and missing data, even in the presence of hypothetical clusters of outliers. Through simulations, the algorithm's performance was compared to the state-of-the-art method, MacroPCA, revealing similar results in scenarios with outliers and greater similarity to least squares methods in outlier-free situations, making RadarTSR the preferred choice for most scenarios. Additionally, the reasonable imputation of missing data by RadarTSR positively impacted outcomes obtained by other models fitted on the imputed dataset. Yet, RadarTSR's approach to treating mild rowwise outliers as rows with cellwise outliers provided a means of correction and masking. However, careful inspection of residual

maps was recommended to address this effect. The chapter highlighted the potential applications of RadarTSR in various scientific fields utilising high-dimensional datasets. However, the heuristic nature of the algorithm and the clustering step were identified as areas for further improvement, along with adaptations needed for specific scenarios involving discrete and categorical data. Despite these limitations, the chapter concluded that RadarTSR presented a promising approach for handling outliers and missing data, and its implementation in open-source languages could increase its accessibility and practicality.

Hence, the chapters discussed within Part II make a significant stride towards realizing the objective of proposing, implementing, and deploying innovative methodologies for statistical machine learning. The contributions described in the mentioned chapters address challenges presented by the surge in data generation within Medicine, Healthcare, and Industry 4.0 paradigms, providing new tools for advancing statistical machine learning methodologies and offering tangible tools to navigate the complexities of contemporary exploratory data analysis.

Objective 2: Apply existing and novel statistical machine learning techniques to real biomedical engineering problems

From a social perspective, "smart" approaches based on AI and ML models bring value because of their accurate predictions. However, the black-box models at their cores opaque the decision-making process of the resulting algorithms. Such an approach trades generalization for accuracy, which is not necessarily in the best interest of research on understanding biology's or health's nature. This triggers a difficult circular path that can only be broken by the introduction of tools pursuing a good individual outcome – as in good predictive performance –and a collective benefit – derived from a better knowledge and understanding of the tackled casuistic.

Chapter 7 introduces the incorporation of Partial Least Squares (PLS) into the Six Sigma toolbox for analyzing data from a Six Sigma project in a university hospital's Outpatient Pharmaceutical Care Unit. Unlike univariate techniques, PLS provides a comprehensive view of correlations between input and output variables, facilitating process understanding and improvement actions. Integrating PLS with classical Six Sigma tools like ANOVA streamlines the confirmation process, reducing the number of statistical tests required. This enhancement to the traditional Six Sigma DMAIC scheme offers a more efficient and time-saving methodology, making it adaptable to complex databases

and paving the way for the next generation of process improvement, known as Multivariate Six Sigma, in 4.0 environments.

Chapter 8 presents a study that incorporates Partial Least Squares Discriminant Analysis (PLS-DA) to differentiate Myalgic Encephalomyelitis/Chronic Fatigue Syndrome (ME/CFS) cases from healthy controls based on a diverse dataset. The dataset includes blood analytic variables, miRNA profiles, and extracellular vesicle (EV) features, with Raman spectroscopy providing a new marker of EV function. The PLS-DA analysis identified 32 variables that effectively distinguish ME/CFS cases from controls, achieving perfect sensitivity and specificity. The study also explores the role of EVs in ME/CFS, highlighting their potential significance in the diagnosis. Raman spectroscopy revealed differences in carotenoid peaks, suggesting a possible link to stressed red blood cells in the patient's circulation. Despite its limited diagnostic value, Raman data proved valuable for reducing the number of PBMC miRNAs required for diagnosis. The study's findings provide insights into ME/CFS pathogenesis and offer promising implications for future diagnosis and treatment.

Chapter 9 aimed to develop a predictive model for determining mortality risk in hospitalized COVID-19 patients using machine learning and statistical classification techniques. The model was based on data collected from 10 preterm infants in Spain and utilized five predictors: age, oxygen saturation, creatinine, platelets, and LDH. The Random Forest algorithm achieved an AUC of 0.8454 in external validation. These predictors have been consistently linked to COVID-19 mortality in previous studies. The model's strength lies in its simplicity and reliance on readily available clinical data at hospital admission, making it practical for clinical use. The sample size of over 12,000 patients, including over 2,000 deceased individuals, adds robustness to the findings. Additionally, a systematic approach comparing different algorithms enhances the model's reliability. However, the study acknowledges limitations related to the timing of data collection during the pandemic, suggesting the model should be validated with updated data. Overall, the mortality prediction model shows promise as a valuable tool for early recognition and prioritization of severe COVID-19 cases.

Chapter 10 proposes a unit conversion model for fluorescence plate readers to obtain comparable results. The model consists of the gain effect function and the unit conversion function. The uncertainty of the model and its parameters can vary depending on the experimental setup. We conducted a Measurement System Analysis (MSA) via an R&R and a B&L analysis to address this. Real data sets from two plate readers are used to assess the performance of the calibration model, showing that the confidence intervals for predicted

concentrations contain the actual values. The protocol also helps detect wells with unexpected concentration values, assess the quality of experimental procedures, and warn about incorrect gain effect functions. The model seems to extrapolate well outside the concentration range, but users are encouraged to assess differences in model parameters for different concentration ranges. The authors emphasize that the main contribution of this work is the proposal of a methodological framework with a solid statistical basis that can be applied to test future analytical expressions for unit conversion. The PLATERO toolbox presented in this study provides a general approach applicable to various measurements beyond fluorescence. Overall, this methodology will play a crucial role in Synthetic Biology, facilitating the comparison of databases from different experimental settings, validating data quality, and expanding the measurement system range.

In conclusion, the common thread among the mentioned Chapters is the application of existing and novel statistical and machine-learning techniques to address real-world biomedical engineering problems. Each chapter tackles specific challenges in the biomedical field, ranging from process improvement in a hospital pharmacy to predicting mortality risk in COVID-19 patients and identifying apnea episodes in preterm infants. The chapters demonstrate the power of statistical and machine learning methods in handling complex and diverse biomedical data, providing valuable insights and improving decision-making in clinical settings.

Overall, the content of Part III exemplifies the potential of AI and ML models to bring accurate predictions and improve healthcare outcomes. Additionally, there is an emphasis on the importance of a systematic and statistically rigorous approach to validate and interpret the results, ensuring the robustness and reliability of the proposed methodologies. By addressing these challenges and providing practical solutions, the Chapters mentioned above contribute to advancing the field of biomedical engineering and its application in real-world healthcare scenarios.

11.2 Relevance

The relevance of the present PhD thesis is highlighted in the following points:

- This thesis has been developed within the framework of two research projects from the Spanish Ministry of Economy, coordinated among different Spanish sites. An international research stay was carried out (Eind-

hoven). Several healthcare institutions related to the two Chapters have applied the proposed solutions and promoted the diffusion of the results obtained from this work. Therefore, the contributions in this thesis have been disseminated across many levels of institutions linked to different layers of biomedical engineering.

- The SCOUTER algorithm for outliers simulation provides a new methodological framework for standard comparison of methods dealing with outliers, enabling the simulation of virtually all outliers while considering a dataset's natural and specific structure.

- The RadarTSR algorithm has updated the previous TSR algorithm efficiently by including the minimally necessary robust steps. Results have proved that the RadarTSR algorithm yields similar and even better results for the MSPE than MacroPCA, the state-of-the-art method to deal with missing data, cellwise outliers and rowwise outliers. Moreover, the algorithm presented in Chapter 6 also included the distinction between single and grouped rowwise outliers, enabling a further diagnosis of rows classified as rowwise outliers by the algorithm, searching for connections between them.

- The incorporation of Partial Least Squares (PLS) into the Six Sigma toolbox improved the efficiency of a hospital unit by providing a more effective and time-saving methodology for process improvement. This demonstrated the potential for implementing Multivariate Six Sigma in 4.0 environments, paving the way for future process improvement methodologies.

- The use of Partial Least Squares Discriminant Analysis (PLS-DA) in Chapter 8 enabled the differentiation of Myalgic Encephalomyelitis/Chronic Fatigue Syndrome (ME/CFS) cases from healthy controls. Moreover, incorporating Raman spectroscopy as a new marker of EV function expanded the understanding of EVs' role in ME/CFS diagnosis and pathogenesis. The study's findings offered promising implications for future diagnosis and treatment of ME/CFS and highlighted the importance of EVs in the disease.

- The predictive model for determining mortality risk in COVID-19 patients from Chapter 9 offered a valuable tool for early recognition and prioritization of severe cases, enhancing patient management and resource allocation. The model's simplicity, relying on readily available clinical

data at hospital admission, makes it practical for clinical use in real-world settings.

- Chapter 10 proposes a unit conversion model for fluorescence plate readers, providing a standardized approach for obtaining comparable results. The PLATERO toolbox offers a general method applicable to various measurements beyond fluorescence, making it a valuable tool for Synthetic Biology research. The methodology paves the way for expanding the measurement system range and detecting more subtle signals while maintaining data quality and enhancing the comparison and validation of experimental data.

11.3 Future lines

In light of the novel insights garnered from the present thesis, this section delineates prospective avenues for future research to augment existing knowledge and address pertinent areas that could not be explored with depth enough to constitute a chapter of the thesis. The findings of this doctoral study lay the groundwork for potential endeavours, encompassing two main themes: adapting the TSR and RadarTSR to deal with categorical variables and developing multivariate pseudosamples as model-agnostic metrics for ML interpretation. By embarking upon these forthcoming trajectories, the concept of Statistical Machine Learning and the biomedical engineering research community can collectively advance the frontiers of knowledge.

11.3.1 Trimmed Scores Regression with categorical variables

As mentioned in Section 3.5 and Chapter 6, missing data frequently affects datasets containing binary and categorical data referring to the same objects. However, the RadarTSR algorithm proposed in Chapter 6 only contemplates datasets with continuous variables. Hence, a straightforward task includes adapting to datasets, including features of a different nature.

For pragmatic purposes, dealing with the dataset from Chapter 9 meant adapting TSR to work with quantitative and qualitative data. To do so, this strategy focused on the preprocessing of the dataset, performing a block-scaling to ensure that all variables represented a comparable amount of information in terms of variance. Afterwards, hard constraints were imposed to ensure that each block of variables respected its nature (i.e., those binary variables $\in \{0, 1\}$ would always have imputed values in the same domain). These results

were validated by comparing PLS-DA models fitted with the imputed datasets using the adapted TSR (named cat-TSR), an adaptation of the NIPALS algorithm provided by SIMCA © and the MICE algorithm included in the mice R package (`https://cran.r-project.org/web/packages/mice/index.html`).

Nonetheless, even though the results for this particular case were acceptable, dealing with categorical variables in PCA models is a non-trivial task that can be answered from different angles.

On the one hand, from a purely geometrical perspective, i.e., understanding variance as the data scattering along with space, the adaptation of TSR should be regarded as a way of ensuring a fair comparison between the variance of quantitative and qualitative variables. Moreover, some hard constraints to respect the mathematical nature of variables should also be included. The problem of working with binary data could be directly reframed by using Multiple Correspondence Analysis (MCA) [217], [218]. In MCA, as in PCA, a lower-dimensional space represents latent structures correlating the original variables. It is based on an indicator matrix, built with as many rows as individuals and columns as levels of the categorical variables. When a categorical variable has more than two levels, its corresponding dummy variables are obtained and used instead as part of the indicator matrix. When PCA is applied in a matrix with only categorical data, its solution is equivalent to the one obtained by MCA.

On the other hand, from a probabilistic perspective, it's worth noting that while PCA doesn't impose explicit distributional assumptions on the original variables in matrix $\boldsymbol{X}$, it deals with the linear combinations of these variables to generate components (i.e., the scores bmT) whose behaviour is assumed to be approximate normality due to the Central Limit Theorem. However, the presence of qualitative or ordinal data challenges the typical assumptions of normality in these components. An existing solution accounting for this probabilistic scope is the Generalised Simultaneous Component Analysis (GSCA) [219]. This framework splits the original matrix $\boldsymbol{X}$ in a set of qualitative binary data, $\boldsymbol{X}_1$, and a set of quantitative normally distributed measurements $\boldsymbol{X}_2$, both referring to the same observations. In [219], the authors propose a GSCA framework for these coupled datasets. This proposal also includes a penalization for estimating the latent subspace dimension and contemplates the possibility of missing data in the data used for PCA-MB. Nonetheless, the missing data imputation is carried out by a PMP procedure, which was shown to be inferior to TSR in some scenarios [85].

Thus, the challenge of adapting TSR and RadarTSR encompasses several questions:

1. Proving if the adapted TSR still shows superiority in the MSPE over other imputation strategies. To answer this question, either the TSR algorithm could be adapted to deal with categorical variables by adding or modifying the required steps, or the TSR expressions for missing data imputation could be embedded in existing algorithms for PCA-MB with missing data and mixed variables, such as the GSCA algorithm;

2. Successfully adapting TSR also implies providing an imputation that respects the nature of each variable;

3. The adapted TSR algorithm should also represent faithfully and without distortion the amount of explained variance by each variable. This could be assessed by comparing the loading matrices ($\boldsymbol{P}$) with those obtained by other frameworks for PCA-MB with mixed datasets in the presence of missing data.

4. The adaptation of RadarTSR should include all the points mentioned above and the simulations of outliers within discrete variables. This encloses an indirect task, which is to adapt the framework of the SCOUTer simulation algorithm, presented in Chapter 5 and used for simulations in Chapter 6, to work with categorical variables.

So far, as part of this thesis, the first approach mentioned in item 1 (i.e., adapting the TSR algorithm to deal with categorical variables) was implemented as part of a solution needed for Chapter 9. Even though the work is incomplete, the following results illustrate the actual implementation stage. These results were obtained with two goals.

The first goal was to assess the potential distortion introduced using the adapted TSR for missing data imputation. This was done by comparing the terms of a PLS-DA model fitted with the TSR–imputed dataset to the ones obtained when NIPALS and MICE were used for the imputation. Figure 11.7 shows the $\boldsymbol{B}_{PLS}$ coefficients obtained by PLS-DA models fitted with each one of the datasets imputed with the three different techniques. As can be seen, there is a significant agreement in the sign and the magnitude of the variables´ coefficients.

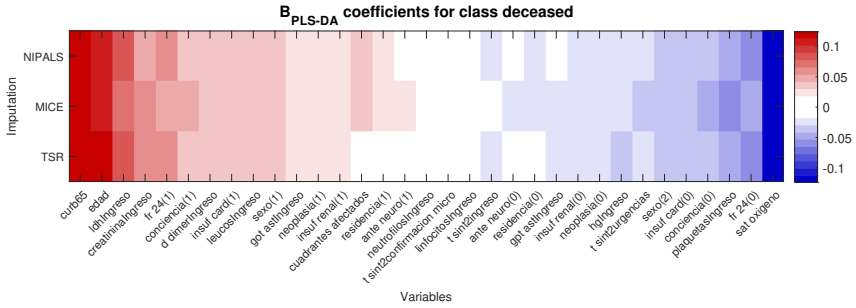


Figure 11.1: Graph displaying the coefficient values $\boldsymbol{B}_{PLS}$ of PLS models obtained using matrices imputed with different techniques. The colours represent positive values (red), values close to zero (white), and negative values (blue), with the intensity of the colour increasing with the magnitude of the coefficient.

Figure 11.2 shows the differences between the $\boldsymbol{B}_{PLS}$ coefficients shown in Figure 11.7 for each variable. As can be seen, the differences are of a low magnitude and very close to zero.

Secondly, once the algorithm had shown acceptable results for its primary use, it was later compared with more mixed datasets (i.e., including continuous and/or categorical variables). These datasets, however, were complete in the first place, enabling the simulation of different percentages of missing data and comparing the imputed values with the real ones. This was impossible with the PROCOVID dataset, which presented missing values from its origin. These results highlighted weaknesses that should be improved, paving the way for future steps in refining the algorithm.

The following Figures show some cases of the results obtained when the adaptation of TSR to deal with categorical variables and the GSCA algorithm were executed for missing data imputation with different datasets. Several percentages of MCAR missing data ranging from 1% to 70% were simulated. Two metrics were used to measure the errors due to the imputation performed with the two different algorithms. On the one hand, the Mean Squared Prediction Error (MSPE) measures the error in predicting real and ordinal variables. On the other hand, the Percentage of Falsely Classified observations (PFC) is used to measure the error caused by the imputation in categorical variables. To compare the results, Least Significant Difference (LSD) intervals were calculated, comparing across methods and missing data percentages.

Figures 11.3 and 11.4 show the results for two datasets containing a mix of continuous and binary variables, with the former containing more continuous

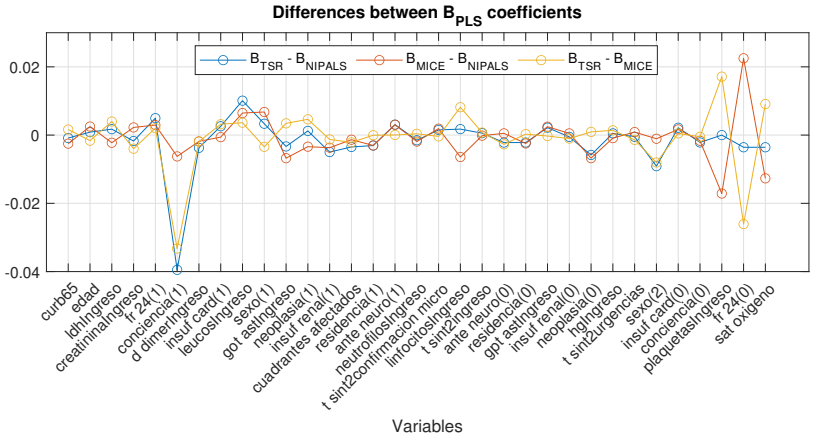


Figure 11.2: Graph illustrating the differences between the coefficient vectors of Partial Least Squares Regression ($\boldsymbol{B}_{PLS}$) according to the employed imputation algorithm.

variables and the latter having more binary ones. Results show that catTSR and GSCA yield similar MSPE and PFC values. Nonetheless, adapting TSR to deal with categorical variables yields statistically significant lower PFC values for the Heart StatLog dataset up to a 50% of missing values (Figure 11.3). On the other hand, GSCA yields lower PFC values from 30% onwards for the Thoracic Surgery dataset.

Figures 11.5 and 11.6 show the results for the Primary Tumour and the SCADI datasets, which contain only binary variables. The first one has a long structure (i.e., $N > K$), whereas the second one has a wide type of structure (i.e., $N < K$). Results show that whereas for the first case, the adaptation of TSR to deal with categorical variables yields statistically significantly higher values for the PFC for more than 30% of missing values (Figure 11.5), it consistently yields a significantly lower PFC for the SCADI dataset (Figures 11.6).

In conclusion, the findings presented in this section strongly suggest that developing a robust and effective adaptation of TSR (Time Series Regression) for handling categorical variables holds practical significance and can potentially become a valuable asset within the data science community. Nonetheless, further algorithm refinement and comprehensive testing across various scenarios and dataset types are necessary to enhance its efficacy. Additionally, exploring the adaptation to address outliers, such as utilizing the RadarTSR algorithm instead of TSR, would be of considerable interest.

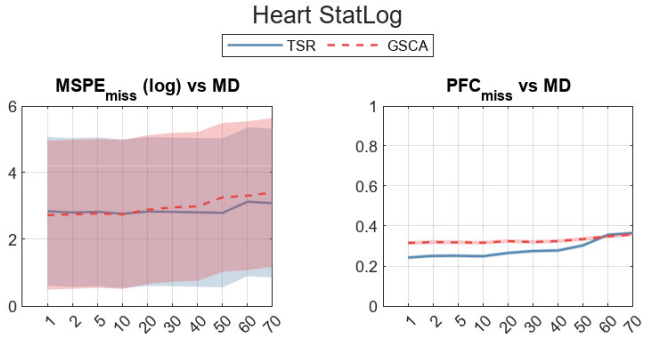


Figure 11.3: Results obtained for the Heart Stat log dataset ($N = 270$, $K = 11$ with 6 real variables and 5 binary variables). The plots show the MSPE (left) and PFC (right) results. Blue solid lines represent the average values for the adapted TSR, and red dashed lines for the GSCA algorithm. The shaded areas of the corresponding colours delimit the LSD intervals.

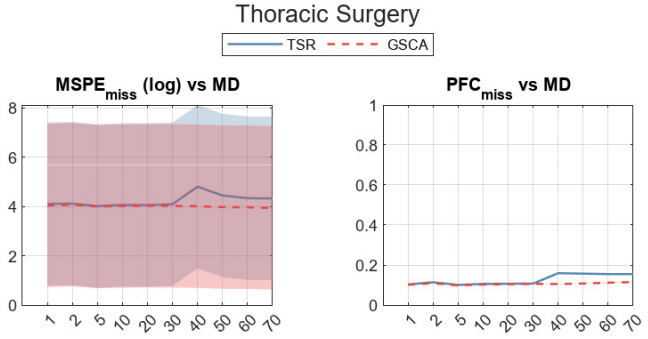


Figure 11.4: Results obtained for the Thoracic surgery dataset ($N = 470$, $K = 14$ with 3 real variables and 10 binary variables). More information is in the caption from Figure 11.3.

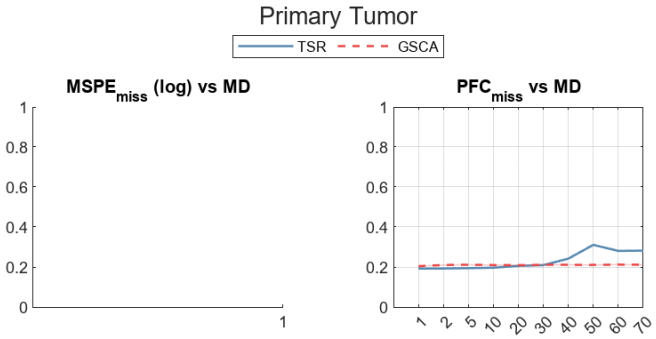


Figure 11.5: Results obtained for the Primary Tumour dataset ($N = 336$, $K = 14$ being all of them binary variables). More information is in the caption from Figure 11.3.

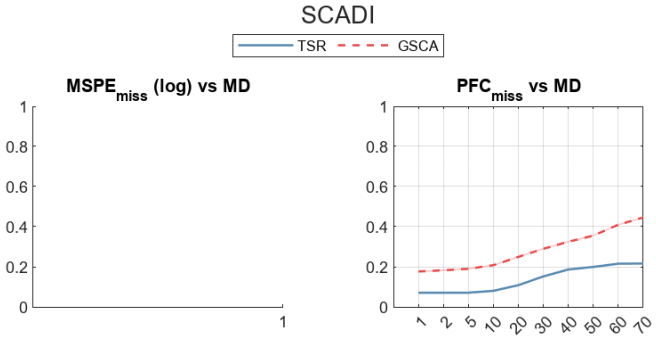


Figure 11.6: Results obtained for the SCADI Tumour dataset ($N = 70$, $K = 205$ being all of them binary variables). More information is in the caption from Figure 11.3.

11.3.2 Multivariate pseudosamples

Despite the potential of pseudosamples, the univariate approach has some weaknesses. In this work, we address two of them: interactions and their outlyingness. When the model's outcome depends on an interaction between predictors, the univariate pseudosampling approach may fail to capture them. Whereas there are proposals to consider the interactions between variables, there is still no assessment regarding the realism of the pseudosamples, which is addressed in the following lines.

To rely on the model predictions, the pseudosamples projected onto it should resemble the predictors' space used to fit the model as much as possible. To put it simply, pseudosamples should not be outliers concerning the model. If pseudosamples are built with variations outside the range of the variables, the pseudo-sample-matrix would not be reliable anymore since it would not represent the $\boldsymbol{X}$ matrix used to train the model. When scaled up to the multivariate perspective, this case, in terms of the univariate range, is what happens with univariate pseudosamples.

Just varying one variable and keeping the rest of the predictors constant assumes orthogonality between all the dataset variables. This, in the real world, is highly unlikely. In the presence of highly correlated predictors, which is a prevalent scenario, samples that do not report this joint variation will be outlying, not representing the reality neither within $\boldsymbol{X}$ nor between $\boldsymbol{X}$ and $\boldsymbol{Y}$. Thus, it seems reasonable that a proper pseudosampling approach designs pseudosamples as realistic as possible, and this requires redefining pseudosamples in terms of a multivariate framework.

Considering the facts described in the previous section, in this chapter, we propose a pseudosampling approach that embraces the multivariate nature of the data while keeping traceability on the independent sources of variation of variables. The key idea is making variations not directly in the real space but in a latent space of reduced dimensionality based on a PCA model built with the training dataset. Afterwards, there are different ways to explore the latent space.

- Linear spacing on the scores. This approach can be regarded as the immediate multivariate version of the approach from [220]. A PCA is built with the observations of each class separately. Later, a linear spacing is carried out on each PC, setting the rest to their low or high level. These combinations between low and high levels are done pairwise between all PCs. The supervised take on the PCA models is done to ensure that the

pseudosamples capture all classes. Otherwise, with unbalanced datasets, there could be an under-representation of minorities. Then, the latent pseudosamples can be back-projected to the original space. *Finally, all these pseudosamples expressed in the original space can be projected onto a global PCA, built with all classes together.*

- Full Design Of Experiments on the scores. In this scheme, a PCA is built as in the previous approach. Later, a full 2^A (being A the latent dimension of the PCA model) DOE is used to simulate the latent pseudosamples. The maximum and minimum values of each $\boldsymbol{t}_a$ vector are used as high and low levels for each PC and class. The next steps follow the previous approach, back-projecting the resulting pseudosamples to the original space.

- Spacing on the Hotelling's T^2. This strategy contemplates pseudosamples as extreme outliers that still follow the correlation pattern of the reference dataset. In other words, pseudosamples can be built by generating a gradient on Hotelling's T^2 by increasing one PC at a time while keeping the Squared Prediction Error (SPE) constant. To do so, the SCOUTer algorithm to simulate controlled outliers presented in Chapter 5 was used.

To illustrate the results yielded by each approach, we compared the inference obtained with toy examples of simulated datasets. In all cases, we know a priori the importance of the predictors and the relationship with the response. We included two non-linear modelling approaches: kernel PLSDA and Random Forest. For the kernel-PLSDA, the kernel with the radial basis function was used in all cases, as it can adapt to different degrees of non-linearity by tuning the hyperparameter σ. Further information about the kernel-PLSDA models can be found in Section 3.4.2.

The first case is the *Triangle dataset* (Figure 11.7) presents a binary classification problem. As it can be seen, this is a linearly separable problem by using two PCs. The loading plot indicates that belonging to class 2 positively correlates with variables 16 to 20 (variables with positive loadings on 1st PC) and negatively correlated with variables from 6 to 15 (variables with negative loadings on 2nd PC).

Figure 11.8 illustrates the three procedures to simulate the multivariate pseudosamples mentioned above. Pseudosamples for each class were simulated using a PCA model fitted with the observations of each class. Afterwards, these pseudosamples were projected onto a PCA model fitted with all observations, and these are the score plots shown in Figure 11.8.

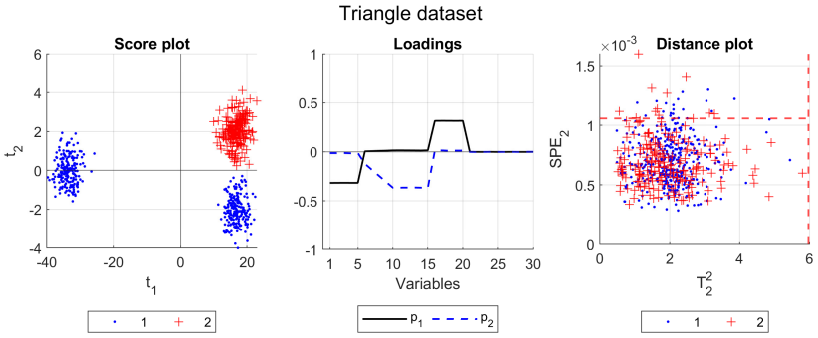


Figure 11.7: Class profiles of the triangle dataset in the score plot (left plot), loading plot (centre) and distance plot (right).

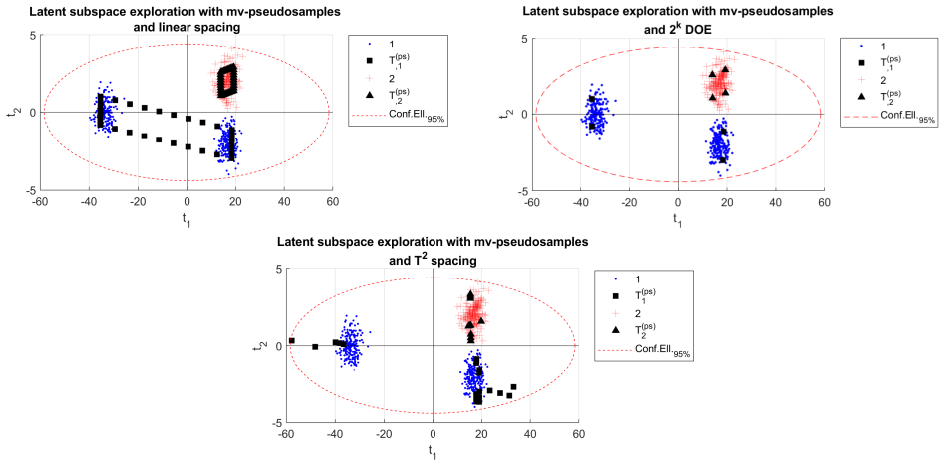


Figure 11.8: Score plots of the triangle dataset with the reference observations and the pseudosamples simulated for each class with the linearly spaced scores (left), following a 2^k latent DOE (centre) and an T^2 spacing (right). Blue dots and red triangles represent the points of classes "1" and "2", respectively. Black squares and black triangles represent the pseudosamples generated by fitting a PCA model on each one of the classes and exploring the generated latent space by each one of the three approaches.

One of the first aspects to check and compare between univariate and multivariate pseudosamples is their faithfulness to the latent structure of the original samples. Figure 11.9 shows the distance plots for the SPE and Hotelling's T^2

of the pseudosamples, using the global PCA model fitted with all the observations from the triangle dataset.

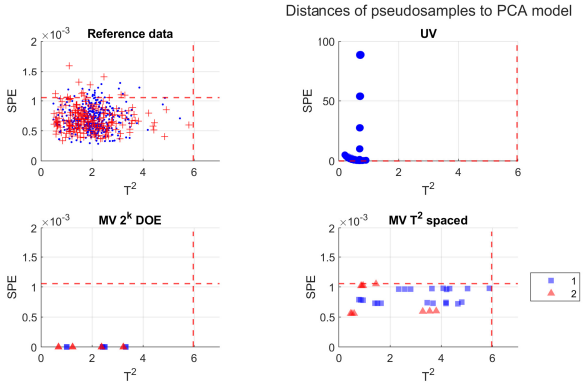


Figure 11.9: Distance plots of the triangle dataset with the pseudosamples simulated via the classical univariate method and the multivariate proposals mentioned in this section. Red dashed lines represent the 95% Upper Control Limits for both the SPE and the T^2, calculated using the PCA model fitted with all the observations of the triangle dataset.

As it can be seen, all multivariate pseudosamples respect the Upper Control Limits for the SPE and the T^2 of the reference PCA model. Yet, this is not the case for univariately generated pseudosamples, which present values within the normal range (distances below the T^2 UCL) but which do not fit the multivariate structure of the PCA model, showing SPE values far above the UCL. This means that these univariate pseudosamples would be outliers concerning the model. Therefore, using pseudosamples with models fitted with the original observations could be conceptually arguable, mainly if predictive models use the captured correlation between variables to calculate the response.

The second scenario, the *Chess dataset* (Figure 11.10), presents a binary classification problem but with a separation of classes that relies on the interaction between groups of variables. As seen in Figure 11.10, observations of class "2" have non-null values for the range of variables between the 6th and the 20th one. On the contrary, class "1" observations are related to having non-null values for the rest of the variables.

Figure 11.11 illustrates the pseudosamples obtained when the three simulation procedures are used, and their distances are shown in Figure 11.12. This second case study shows that only multivariate pseudosamples are within the control limits for the SPE and the T^2.

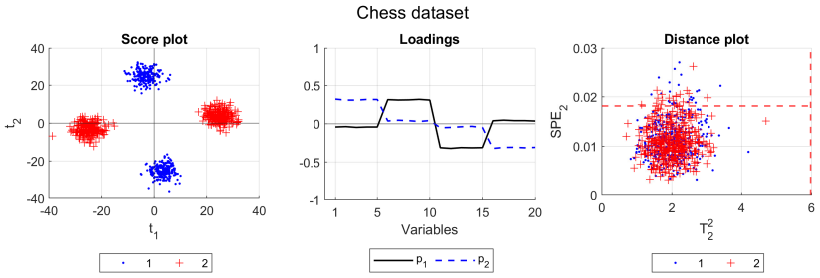


Figure 11.10: Class profiles of the chess dataset in the score plot (left plot), loading plot (centre) and distance plot (right). More information in caption from Figure 11.7.

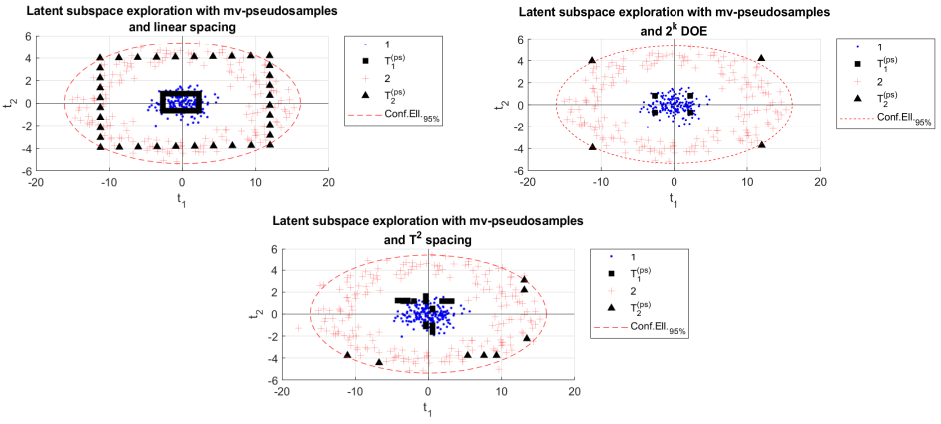


Figure 11.11: Score plots of the chess dataset with the reference observations and the pseudosamples simulated for each class with the linearly spaced scores (left), following a 2^k latent DOE (centre) and an T^2 spacing (right).

Finally, the *Circle dataset* presents the last toy example of a binary classification problem. In this case, the class separation is non-linear, as shown by the radial disposition of observations in the score plot from Figure 11.13. The loading plot shows that belonging to class 2 correlates with big values in magnitude in all variables with non-null loadings either on the 1st or the 2nd PC (i.e., for variables between the 1st and the 20th one).

Figure 11.14 illustrates the pseudosamples obtained when the three simulation procedures are used. Distance plots in Figure 11.15 confirm that only

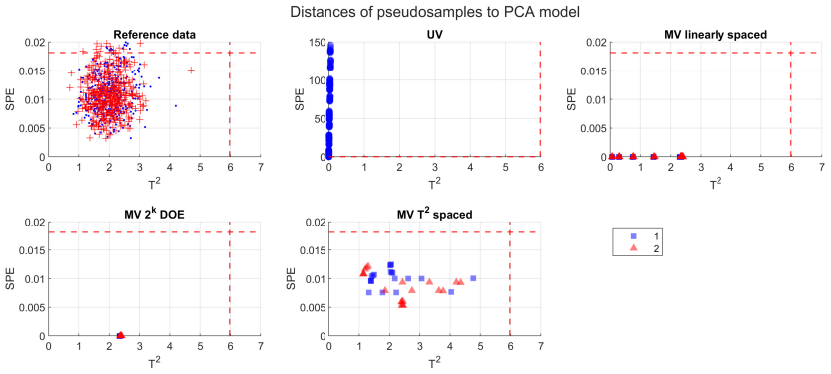


Figure 11.12: Distance plots of the chess dataset with the pseudosamples simulated via the classical univariate method and the multivariate proposals mentioned in this section. More information in caption from Figure 11.9.

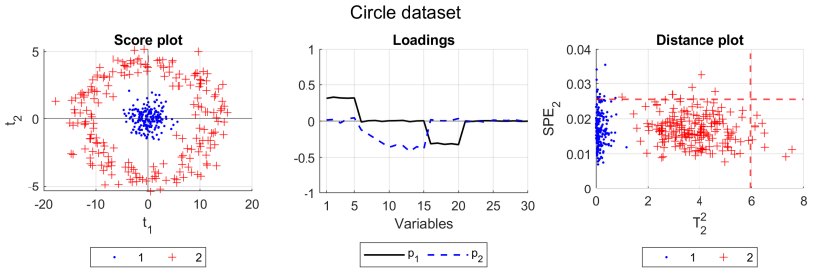


Figure 11.13: Class profiles of the circle dataset in the score plot (left plot), loading plot (centre) and distance plot (right). More information in the caption from Figure 11.7.

multivariate pseudosamples are within the control limits for the SPE and the T^2.

Although Figures 11.9 , 11.12 and 11.15 illustrate the point of pseudosamples' likelihood concerning the original dataset, there is still a question that needs to be addressed. Pseudosamples are used for interpretability purposes, and multivariate pseudosamples should also embrace this aspect. Future steps include proposing parameters or metrics that materialise the relationships seen by the pseudosamples' variations and the variations in the response values. This last aspect also concerns univariate pseudosamples, mainly used only as

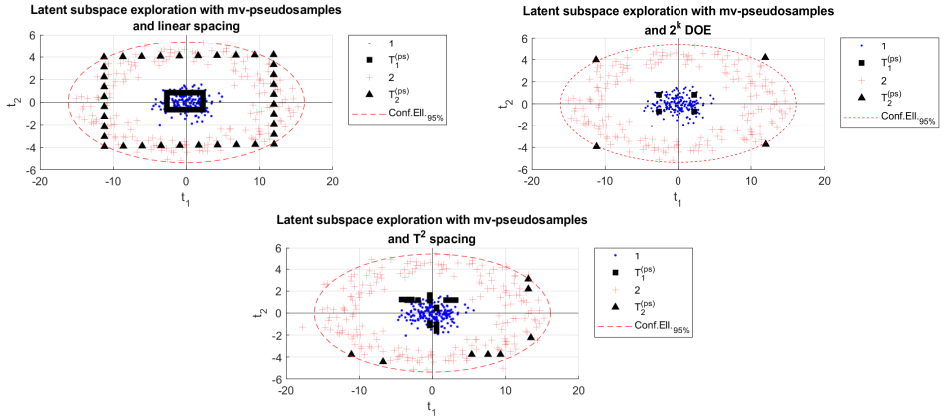


Figure 11.14: Score plots of the circle dataset with the reference observations and the pseudosamples simulated for each class with the linearly spaced scores (left), following a 2^k latent DOE (centre) and an T^2 spacing (right).

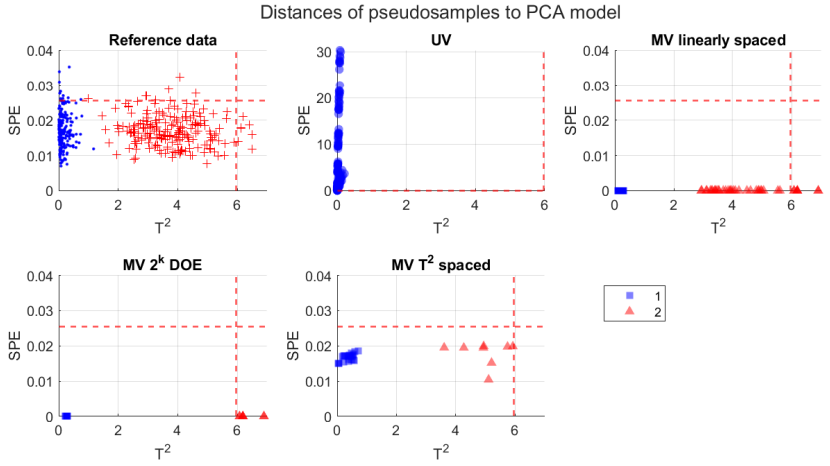


Figure 11.15: Distance plots of the circle dataset with the pseudosamples simulated via the classical univariate method and the multivariate proposals mentioned in this section. More information in caption from Figure 11.15.

an exploratory tool assessing the $\boldsymbol{X} - \boldsymbol{Y}$ mentioned above dependency by visualising the pseudosamples' trajectories.

Bibliography

[1] J. D. Bronzino, *Biomedical Engineering Handbook 2*. Springer Science & Business Media, 2000, vol. 2 (cit. on p. 12).

[2] J. D. Bronzino and D. R. Peterson, *Biomedical engineering fundamentals*. CRC press, 2014 (cit. on p. 12).

[3] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges", in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, pp. 2147–2152. DOI: 10.1109/FSKD.2015.7382284 (cit. on p. 12).

[4] G. Aceto, V. Persico, and A. Pescapé, "Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0", *Journal of Industrial Information Integration*, vol. 18, 2020, ISSN: 2452-414X. DOI: https://doi.org/10.1016/j.jii.2020.100129 (cit. on p. 13).

[5] J. Li and P. Carayon, "Health care 4.0: A vision for smart and connected health care", *IISE Transactions on Healthcare Systems Engineering*, vol. 11, no. 3, pp. 171–180, 2021. DOI: 10.1080/24725579.2021.1884627. eprint: https://doi.org/10.1080/24725579.2021.1884627 (cit. on p. 13).

[6] F. H. Crick, "On protein synthesis", in *Symp Soc Exp Biol*, vol. 12, 1958, p. 8 (cit. on p. 14).

[7] F. Crick, "Central dogma of molecular biology", *Nature*, vol. 227, no. 5258, pp. 561–563, 1970 (cit. on p. 14).

[8] H. Kitano, "Systems biology: A brief overview", *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002. DOI: 10.1126/science.1069492. eprint: https://www.science.org/doi/pdf/10.1126/science.1069492 (cit. on p. 14).

[9] R. Y. Tsien, "The green fluorescent protein", *Annual review of biochemistry*, vol. 67, no. 1, pp. 509–544, 1998 (cit. on p. 14).

[10] S. A. Benner and A. M. Sismour, "Synthetic biology", *Nature reviews genetics*, vol. 6, no. 7, pp. 533–543, 2005 (cit. on p. 15).

[11] National Human Genome Research Institute, *Synthetic biology*, Web Page, 2023 (cit. on p. 15).

[12] J. R. Kelly, A. J. Rubin, J. H. Davis, *et al.*, "Measuring the activity of BioBrick promoters using an in vivo reference standard", *Journal of Biological Engineering*, vol. 3, no. 4, 2009 (cit. on pp. 16, 17, 263).

[13] A. Fedorec, C. Robinson, K. Wen, and C. Barnes, "FlopR: An Open Source Software Package for Calibration and Normalization of Plate Reader and Flow Cytometry Data", *ACS Synthetic Biology*, vol. 9, no. 9, pp. 2258–2266, 2020, PMID: 32854500. DOI: 10.1021/acssynbio.0c00296. eprint: https://doi.org/10.1021/acssynbio.0c00296 (cit. on pp. 16, 17).

[14] S. Castillo-Hair, J. Sexton, B. Landry, E. Olson, O. Igoshin, and J. Tabor, "FlowCal: A User-Friendly, Open Source Software Tool for Automatically Converting Flow Cytometry Data from Arbitrary to Calibrated Units", *ACS Synthetic Biology*, vol. 5, no. 7, pp. 774–780, 2016, PMID: 27110723. DOI: 10.1021/acssynbio.5b00284. eprint: https://doi.org/10.1021/acssynbio.5b00284 (cit. on pp. 16, 17).

[15] R. Spizzo, M. I. Almeida, A. Colombatti, and G. A. Calin, "Long non-coding rnas and cancer: A new frontier of translational research?", *Oncogene*, vol. 31, no. 43, pp. 4577–4587, 2012 (cit. on p. 17).

[16] C. Condrat, D. C. Thompson, M. Barbu, *et al.*, "Mirnas as biomarkers in disease: Latest findings regarding their role in diagnosis and prognosis", *Cells*, vol. 9, no. 2, p. 276, 2020 (cit. on p. 17).

[17] W. Huang, "Micrornas: Biomarkers, diagnostics, and therapeutics", *Bioinformatics in MicroRNA research*, pp. 57–67, 2017 (cit. on p. 17).

[18] A. F. Villaverde, J. Ross, F. Morán, and J. R. Banga, "Mider: Network inference with mutual information distance and entropy reduction", *PLOS ONE*, vol. 9, no. 5, pp. 1–15, May 2014. DOI: 10.1371/journal.pone.0096732 (cit. on p. 18).

[19] C. Krafft, K. Wilhelm, A. Eremin, *et al.*, "A specific spectral signature of serum and plasma-derived extracellular vesicles for cancer screening", *Nanomedicine: Nanotechnology, Biology, and Medicine*, vol. 13, pp. 835–841, 3 Apr. 2017, ISSN: 15499642. DOI: 10.1016/J.NANO.2016.11.016 (cit. on pp. 19, 234).

[20] C. Morasso, D. Sproviero, M. C. Mimmi, *et al.*, "Raman spectroscopy reveals biochemical differences in plasma derived extracellular vesicles from sporadic Amyotrophic Lateral Sclerosis patients", *Nanomedicine: Nanotechnology, Biology, and Medicine*, vol. 29, Oct. 2020, ISSN: 15499642. DOI: 10.1016/J.NANO.2020.102249 (cit. on pp. 19, 234).

[21] F. S. Collins, M. Morgan, and A. Patrinos, "The human genome project: Lessons from large-scale biology", *Science*, vol. 300, no. 5617, pp. 286–290, 2003 (cit. on p. 19).

[22] D. Reinsel, J. Gantz, and J. Rydning, "The digitization of the world from edge to core", *IDC white paper*, no. November, 2018 (cit. on pp. 20, 300).

[23] R. Greenes, *Clinical decision support: the road to broad adoption*. Academic Press, 2014 (cit. on p. 20).

[24] E. S. Berner, *Clinical decision support systems*. Springer, 2007, vol. 233 (cit. on pp. 20, 21).

[25] R. T. Sutton, D. Pincock, D. C. Baumgart, D. C. Sadowski, R. N. Fedorak, and K. I. Kroeker, "An overview of clinical decision support sys-

tems: Benefits, risks, and strategies for success", *NPJ digital medicine*, vol. 3, no. 1, pp. 1–10, 2020 (cit. on p. 21).

[26] M. L. George, *Lean six sigma for service*. McGraw-Hill, 2003 (cit. on p. 22).

[27] E. D. Arnheiter and J. Maleyeff, "The integration of lean management and six sigma", *The TQM magazine*, vol. 17, no. 1, pp. 5–18, 2005 (cit. on p. 22).

[28] R. T. Sutton, D. Pincock, D. C. Baumgart, D. C. Sadowski, R. N. Fedorak, and K. I. Kroeker, "An overview of clinical decision support systems: benefits, risks, and strategies for success", *npj Digital Medicine*, vol. 3, 1 Dec. 2020, ISSN: 23986352. DOI: 10.1038/s41746-020-0221-y (cit. on p. 22).

[29] H. Singh, G. D. Schiff, M. L. Graber, I. Onakpoya, and M. J. Thompson, "The global burden of diagnostic errors in primary care", *BMJ quality & safety*, vol. 26, no. 6, pp. 484–494, 2017 (cit. on p. 22).

[30] H. Singh, A. N. Meyer, and E. J. Thomas, "The frequency of diagnostic errors in outpatient care: Estimations from three large observational studies involving us adult populations", *BMJ quality & safety*, vol. 23, no. 9, pp. 727–731, 2014 (cit. on p. 22).

[31] K. Goddard, A. Roudsari, and J. C. Wyatt, "Automation bias–a hidden issue for clinical decision support system use", *International Perspectives In Health Informatics*, pp. 17–22, 2011 (cit. on p. 22).

[32] J. Ash, D. Sittig, E. Campbell, K. Guappone, and R. H. Dykstra, "Some unintended consequences of clinical decision support systems", in *Amia annual Symposium proceedings*, American Medical Informatics Association, vol. 2007, 2007, p. 26 (cit. on p. 23).

[33] J. Wyatt and D. Spiegelhalter, "Field trials of medical decision-aids: Potential problems and solutions.", in *Proceedings of the annual symposium on computer application in medical care*, American Medical Informatics Association, 1991, p. 3 (cit. on p. 23).

[34] W. Sujansky, "Heterogeneous database integration in biomedicine", *Journal of biomedical informatics*, vol. 34, no. 4, pp. 285–298, 2001 (cit. on p. 23).

[35] M. A. Musen, B. Middleton, and R. Greenes, "Clinical decision-support systems", in *Biomedical informatics*, Springer, 2021, pp. 795–840 (cit. on p. 23).

[36] J. Kabachinski, "A look at clinical decision support systems", *Biomedical Instrumentation & Technology*, vol. 47, no. 5, pp. 432–434, 2013 (cit. on p. 23).

[37] D. O'Reilly, J. Tarride, R. Goeree, C. Lokker, and K. A. McKibbon, "The economics of health information technology in medication management: A systematic review of economic evaluations", *Journal of the American Medical Informatics Association*, vol. 19, no. 3, pp. 423–438, 2012 (cit. on p. 23).

[38] T. Bright, A. Wong, R. Dhurjati, *et al.*, "Effect of clinical decision-support systems: A systematic review", *Annals of internal medicine*, vol. 157, no. 1, pp. 29–43, 2012 (cit. on p. 23).

[39] V. Jacob, A. B. Thota, S. K. Chattopadhyay, *et al.*, "Cost and economic benefit of clinical decision support systems for cardiovascular disease prevention: A community guide systematic review", *Journal of the American Medical Informatics Association*, vol. 24, no. 3, pp. 669–676, 2017 (cit. on p. 23).

[40] C. Main, T. Moxham, J. Wyatt, J. Kay, R. Anderson, and K. Stein, "Computerised decision support systems in order communication for diagnostic, screening or monitoring test ordering: Systematic reviews of the effects and cost-effectiveness of systems", 2010 (cit. on p. 23).

[41] B. Blum, "Clinical information systems – a review", *Western Journal of Medicine*, vol. 145, no. 6, p. 791, 1986 (cit. on p. 24).

[42] E. Bernstam, J. Smith, and T. R. Johnson, "What is biomedical informatics?", *Journal of biomedical informatics*, vol. 43, no. 1, pp. 104–110, 2010 (cit. on p. 24).

[43] A. Samuel, "Some studies in machine learning using the game of checkers", *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959. DOI: 10.1147/rd.33.0210 (cit. on p. 28).

[44] B. Efron, "Prediction, estimation, and attribution", *Journal of the American Statistical Association*, vol. 115, no. 530, pp. 636–655, 2020. DOI: 10.1080/01621459.2020.1762613 (cit. on p. 28).

[45] L. Breiman, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)", *Statistical science*, vol. 16, no. 3, pp. 199–231, 2001 (cit. on p. 29).

[46] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, Nov. 1901, ISSN: 1941-5982. DOI: 10.1080/14786440109462720 (cit. on pp. 33, 102, 105).

[47] F. R. Hampel, "A General Qualitative Definition of Robustness", *The Annals of Mathematical Statistics*, vol. 42, pp. 1887–1896, 6 1971, ISSN: 0003-4851. DOI: 10.1214/aoms/1177693054 (cit. on p. 34).

[48] M. Hubert, P. J. Rousseeuw, and W. Van den Bossche, "MacroPCA: An all-in-one PCA method allowing for missing values as well as cellwise and rowwise outliers", *Technometrics*, vol. 61, no. 4, pp. 1–18, Jan. 2018, ISSN: 0040-1706. DOI: 10.1080/00401706.2018.1562989. eprint: 1806.00954 (cit. on pp. 35, 41, 65, 74, 91–94, 102, 104, 108, 110, 113–117, 120, 148, 150, 153, 154, 156, 158).

[49] P. J. Huber, "Robust Estimation of a Location Parameter", *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964, ISSN: 0003-4851. DOI: 10.1214/aoms/1177703732 (cit. on pp. 35, 64).

[50] R. A. Maronna, "Robust M-Estimators of Multivariate Location and Scatter", *The Annals of Statistics*, vol. 4, pp. 51–67, 1 Jan. 1976. DOI: 10.1214/aos/1176343347 (cit. on p. 36).

[51] N. A. Campbell, "Robust Procedures in Multivariate Analysis I: Robust Covariance Estimation", 1980, pp. 231–237 (cit. on p. 36).

[52] P. J. Rousseeuw, "Least Median of Squares Regression", *Journal of the American Statistical Association*, vol. 79, pp. 871–880, 388 Dec. 1984, ISSN: 0162-1459. DOI: 10.1080/01621459.1984.10477105 (cit. on p. 36).

[53] P. J. Rousseeuw and K. Van Driessen, "A fast algorithm for the minimum covariance determinant estimator", *Technometrics*, vol. 41, pp. 212–223, 3 Aug. 1999, ISSN: 0040-1706. DOI: 10.1080/00401706.1999.10485670 (cit. on pp. 37, 41).

[54] G. Li and Z. Chen, "Projection-Pursuit Approach to Robust Dispersion Matrices and Principal Components: Primary Theory and Monte Carlo", *Journal of the American Statistical Association*, vol. 80, pp. 759–766, 391 1985, ISSN: 01621459. DOI: 10.2307/2288497 (cit. on p. 37).

[55] C. Croux and A. Ruiz-Gazen, "High breakdown estimators for principal components: the projection-pursuit approach revisited", *Journal of Multivariate Analysis*, vol. 95, pp. 206–226, 2005. DOI: 10.1016/j.jmva.2004.08.002 (cit. on p. 37).

[56] C. Croux, P. Filzmoser, and M. R. Oliveira, "Algorithms for projection–pursuit robust principal component analysis", *Chemometrics and Intelligent Laboratory Systems*, vol. 87, no. 2, pp. 218–225, 2007 (cit. on pp. 37, 38).

[57] M. Hubert, P. J. Rousseeuw, and S. Verboven, "A fast method for robust principal components with applications to chemometrics", *Chemometrics and intelligent laboratory systems*, vol. 60, no. 1-2, pp. 101–111, 2002 (cit. on pp. 38, 40, 41).

[58] M. Hubert, P. Rousseeuw, and K. Vanden Branden, "Robpca: A new approach to robust principal component analysis", *Technometrics*, vol. 47, pp. 64–79, 1 Feb. 2005, ISSN: 00401706. DOI: 10.1198/004017004000000563 (cit. on pp. 40, 94).

[59] P. J. Rousseeuw and W. Van Den Bossche, "Detecting Deviating Data Cells", *Technometrics*, vol. 60, no. 2, pp. 135–145, Apr. 2018, ISSN: 15372723. DOI: 10.1080/00401706.2017.1340909 (cit. on pp. 42, 64, 94, 102, 120).

[60] P. Geladi and B. Kowalski, "Partial least-squares regression: A tutorial", *Analytica Chimica Acta Elsevier Science Publishers B.V*, vol. 185, pp. 1–17, 1986. DOI: 10.1016/0003-2670(86)80028-9 (cit. on pp. 42, 44).

[61] A. Höskuldsson, "Pls regression", *Journal of Chemometrics*, vol. 2, pp. 581–591, August 1987 Jun. 1988. DOI: 10.1002/cem.1180020306 (cit. on pp. 42, 208).

[62] Barker, Matthew and Rayens, William, "Partial least squares for discrimination", *Journal of Chemometrics*, vol. 17, pp. 166–173, 3 Mar. 2003, doi: 10.1002/cem.785, ISSN: 0886-9383. DOI: 10.1002/cem.785 (cit. on pp. 44, 45, 208, 213, 245).

[63] M. H. Quenouille, "Problems in plane sampling", *The Annals of Mathematical Statistics*, vol. 20, pp. 355–375, 3 Sep. 1949. DOI: 10.1214/aoms/1177729989 (cit. on p. 44).

[64] J. W. Tukey, "A Problem of Berkson, and Minimum Variance Orderly Estimators", *The Annals of Mathematical Statistics*, vol. 29, pp. 614–623, 2 Jun. 1958. DOI: 10.1214/aoms/1177706647 (cit. on p. 44).

[65] G. Postma, P. Krooshof, and L. Buydens, "Opening the kernel of kernel partial least squares and support vector machines", *Analytica Chimica Acta*, vol. 705, pp. 123–134, 1-2 2011, ISSN: 00032670. DOI: 10.1016/j.aca.2011.04.025 (cit. on p. 46).

[66] L. Breiman, "Random forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324. eprint: /dx.doi.org/10.1023\%2FA\%3A1010933404324 (cit. on pp. 46, 209, 213, 245).

[67] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. CRC Press, 1984 (cit. on p. 46).

[68] B. Efron and R. Tibshirani, "Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy", *Statistical science*, pp. 54–75, 1986 (cit. on p. 47).

[69] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.", *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. DOI: 10.1214/aos/1013203451 (cit. on p. 53).

[70] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144, ISBN: 9781450342322. DOI: 10.1145/2939672.2939778 (cit. on p. 53).

[71] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions", *The Journal of Machine Learning Research*, vol. 11, pp. 1803–1831, 2010 (cit. on p. 53).

[72] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, 2nd. John Wiley & Sons, Inc., Jan. 2014, pp. 1–381, ISBN: 9781119013563. DOI: 10.1002/9781119013563 (cit. on p. 55).

[73] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm", *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977 (cit. on p. 57).

[74] J. L. Schafer, *Analysis of incomplete multivariate data*. CRC press, 1997 (cit. on p. 57).

[75] D. B. Rubin, *Multiple imputation for nonresponse in surveys*. John Wiley & Sons, 2004, vol. 81 (cit. on pp. 57, 58).

[76] I. R. White, P. Royston, and A. M. Wood, "Multiple imputation using chained equations: Issues and guidance for practice", *Statistics in medicine*, vol. 30, no. 4, pp. 377–399, 2011 (cit. on p. 58).

[77] F. Arteaga and A. Ferrer, "Dealing with missing data in MSPC: Several methods, different interpretations, some examples", *Journal of Chemometrics*, vol. 16, no. 8-10, pp. 408–418, 2002, ISSN: 08869383. DOI: 10.1002/cem.750 (cit. on pp. 58, 59, 74, 102, 104).

[78] P. R. Nelson, P. A. Taylor, and J. F. MacGregor, "Missing data methods in pca and pls: Score calculations with incomplete observations", *Chemometrics and Intelligent Laboratory Systems*, vol. 35, no. 1, pp. 45–65, 1996, ISSN: 0169-7439. DOI: https://doi.org/10.1016/S0169-7439(96)00007-X (cit. on p. 59).

[79] S. Wold, C. Albano, W. Dunn, *et al.*, "Food research and data analysis", *London: H. Martens and H. Russwurn Jr*, 1983 (cit. on p. 59).

[80] H. Martens and T. Naes, *Multivariate calibration*. John Wiley & Sons, 1992 (cit. on p. 59).

[81] B. Walczak and D. Massart, "Dealing with missing data: Part i", *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 1, pp. 15–27, 2001, ISSN: 0169-7439. DOI: https://doi.org/10.1016/S0169-7439(01)00131-9 (cit. on p. 59).

[82] F. Arteaga and A. Ferrer, "Framework for regression-based missing data imputation methods in on-line mspc", *Journal of Chemometrics*, vol. 19, no. 8, pp. 439–447, 2005. DOI: https://doi.org/10.1002/cem.946. eprint: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.946 (cit. on p. 59).

[83] P. Nelson, *The treatment of missing measurements in PCA and PLS models*. 2002 (cit. on p. 60).

[84] B. Walczak and D. L. Massart, "Dealing with missing data: Part II", *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 1, pp. 29–42, 2001, ISSN: 01697439. DOI: 10.1016/S0169-7439(01)00132-0 (cit. on pp. 60, 102, 104, 117).

[85] A. Folch-Fortuny, F. Arteaga, and A. Ferrer, "PCA model building with missing data: New proposals and a comparative study", *Chemometrics and Intelligent Laboratory Systems*, vol. 146, pp. 77–88, 2015, ISSN: 18733239. DOI: 10.1016/j.chemolab.2015.05.006 (cit. on pp. 61, 91–93, 102, 104, 117, 306).

[86] A. Ferrer, "Multivariate Statistical Process Control Based on Principal Component Analysis (MSPC-PCA): Some Reflections and a Case Study in an Autobody Assembly Process", *Quality Engineering*, vol. 19, no. 4,

pp. 311–325, Oct. 2007, ISSN: 0898-2112. DOI: 10.1080/08982110701621304 (cit. on pp. 62, 105, 111).

[87] G. E. P. Box, "Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, I. Effect of Inequality of Variance in the One-Way Classification", en, *The Annals of Mathematical Statistics*, vol. 25, no. 2, pp. 290–302, 1954, ISSN: 0003-4851. DOI: 10.1214/aoms/1177728786 (cit. on pp. 62, 63).

[88] J. Jackson and G. S. Mudholkar, "Control Procedures for Residuals Associated with Principal Component Analysis", *Technometrics*, vol. 21, no. 3, pp. 341–349, Feb. 1979, ISSN: 00401706. DOI: 10.2307/1267757 (cit. on pp. 62, 63, 111).

[89] L. Eriksson, T. Byrne, E. Johansson, J. Trygg, and C. Vikström, *Multi- and Megavariate Data Analysis: Principles and Applications.* 3rd. Umetrics Academy, 2001 (cit. on p. 62).

[90] P. Nomikos and J. F. MacGregor, "Multivariate SPC Charts for Monitoring Batch Processes", *Technometrics*, vol. 37, no. 1, pp. 41–59, Feb. 1995, ISSN: 0040-1706. DOI: 10.1080/00401706.1995.10485888 (cit. on p. 63).

[91] H. P. Lopuhaa and P. J. Rousseeuw, "Breakdown Points of Affine Equivariant Estimators of Multivariate Location and Covariance Matrices", *The Annals of Statistics*, vol. 19, no. 1, pp. 229–248, 1991. DOI: 10.1214/aos/1176347978 (cit. on pp. 64, 102).

[92] F. Alqallaf, S. Van Aelst, V. J. Yohai, and R. H. Zamar, "Propagation of outliers in multivariate data", *Annals of Statistics*, vol. 37, no. 1, pp. 311–331, 2009, ISSN: 00905364. DOI: 10.1214/07-AOS588 (cit. on pp. 64, 65, 102).

[93] C. Agostinelli, A. Leung, V. J. Yohai, and R. H. Zamar, "Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination", *Test*, vol. 24, no. 3, pp. 441–461, 2015, ISSN: 11330686. DOI: 10.1007/s11749-015-0450-6. eprint: 1406.6031 (cit. on pp. 64, 74, 102).

[94] A. González-Cebrián, F. Arteaga, A. Folch-Fortuny, and A. Ferrer, "How to Simulate Outliers with the Desired Properties", *Chemometrics and Intelligent Laboratory Systems*, p. 104 301, 2021, ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2021.104301 (cit. on pp. 73, 98, 114).

[95] A. Smoliński, B. Walczak, and J. W. Einax, "Exploratory analysis of data sets with missing elements and outliers", *Chemosphere*, vol. 49, no. 3, pp. 233–245, 2002, ISSN: 00456535. DOI: 10.1016/S0045-6535(02)00326-0 (cit. on p. 74).

[96] I. Stanimirova, M. Daszykowski, and B. Walczak, "Dealing with missing values and outliers in principal component analysis", *Talanta*, vol. 72, no. 1, pp. 172–178, 2007. DOI: 10.1016/j.talanta.2006.10.011 (cit. on pp. 74, 91–93).

[97] S. Serneels and T. Verdonck, "Principal component analysis for data containing outliers and missing elements", *Computational Statistics and Data Analysis*, vol. 52, no. 3, pp. 1712–1727, 2008, ISSN: 01679473. DOI: 10.1016/j.csda.2007.05.024 (cit. on pp. 74, 91–93, 103).

[98] F. Arteaga and A. Ferrer, "How to simulate normal data sets with the desired correlation structure", *Chemometrics and Intelligent Laboratory Systems*, vol. 101, no. 1, pp. 38–42, 2010, ISSN: 01697439. DOI: 10.1016/j.chemolab.2009.12.003 (cit. on p. 85).

[99] A. González-Cebrián, F. Arteaga, A. Folch-Fortuny, and A. Ferrer, *CRAN - Package SCOUTer*, 2020 (cit. on p. 98).

[100] A. González-Cebrián, A. Folch-Fortuny, F. Arteaga, and A. Ferrer, "Radartsr: A new algorithm for cellwise and rowwise outlier detection and missing data imputation", *Chemometrics and Intelligent Laboratory Systems*, vol. 247, p. 105 047, 2024, ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2023.105047 (cit. on p. 101).

[101] B. Grung and R. Manne, "Missing values in principal component analysis", *Chemometrics and Intelligent Laboratory Systems*, vol. 42, no. 1-2, pp. 125–139, 1998, ISSN: 01697439. DOI: 10.1016/S0169-7439(98)00031-8 (cit. on p. 102).

[102] F. Arteaga, A. Folch-Fortuny, and A. Ferrer, "2.29 - missing data", in *Comprehensive Chemometrics (Second Edition)*, S. Brown, R. Tauler, and B. Walczak, Eds., Second Edition, Oxford: Elsevier, 2020, pp. 615–639, ISBN: 978-0-444-64166-3. DOI: https://doi.org/10.1016/B978-0-12-409547-2.14718-3 (cit. on p. 102).

[103] P. J. Rousseeuw and M. Hubert, "Anomaly detection by robust statistics", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 2, pp. 1–14, 2018, ISSN: 19424795. DOI: 10.1002/widm.1236. arXiv: 1707.09752 (cit. on p. 102).

[104] A. Folch-Fortuny, F. Arteaga, and A. Ferrer, "PLS model building with missing data: New algorithms and a comparative study", *Journal of Chemometrics*, vol. 31, no. 7, pp. 1–12, 2017, ISSN: 1099128X. DOI: 10.1002/cem.2897 (cit. on p. 103).

[105] E. Saccenti and J. Camacho, "On the use of the observation-wise k-fold operation in PCA cross-validation", *Journal of Chemometrics*, vol. 29, no. 8, pp. 467–478, 2015, ISSN: 1099128X. DOI: 10.1002/cem.2726 (cit. on p. 109).

[106] A. Ferrer, "Latent structures-based multivariate statistical process control: A paradigm shift", *Quality Engineering*, vol. 26, no. 1, pp. 72–91, 2014, ISSN: 08982112. DOI: 10.1080/08982112.2013.846093 (cit. on p. 111).

[107] S. Lloyd, "Least squares quantization in pcm", *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982 (cit. on p. 112).

[108] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis", *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974 (cit. on p. 113).

[109] A. Folch-Fortuny, F. Arteaga, and A. Ferrer, "Missing Data Imputation Toolbox for MATLAB", *Chemometrics and Intelligent Laboratory Systems*, vol. 154, pp. 93–100, 2016, ISSN: 18733239. DOI: 10.1016/j.chemolab.2016.03.019 (cit. on pp. 114, 116).

[110] S. A. Hutzler and G. B. Bessee, "Remote near-infrared fuel monitoring system", SOUTHWEST RESEARCH INST SAN ANTONIO TX-

TARDEC FUELS and LUBRICANTS RESEARCH FACILITY, Tech. Rep., 1997 (cit. on p. 116).

[111] E. Aguado-Sarrió, J. Prats-Montalbán, R. Sanz-Requena, G. Garcia-Martí, L. Martí-Bonmatí, and A. Ferrer, "Biomarker comparison and selection for prostate cancer detection in dynamic contrast enhanced-magnetic resonance imaging (dce-mri)", *Chemometrics and Intelligent Laboratory Systems*, vol. 165, pp. 38–45, 2017, ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2017.04.003 (cit. on p. 116).

[112] K. H. Janssens, I. Deraedt, O. Schalm, and J. Veeckman, "Composition of 15?17th Century Archaeological Glass Vessels Excavated in Antwerp, Belgium BT - Modern Developments and Applications in Microbeam Analysis", G. Love, W. A. P. Nicholson, and A. Armigliato, Eds., Vienna: Springer Vienna, 1998, pp. 253–267, ISBN: 978-3-7091-7506-4 (cit. on p. 116).

[113] P. Lemberge, I. De Raedt, K. H. Janssens, F. Wei, and P. J. Van Espen, "Quantitative analysis of 16–17th century archaeological glass vessels using PLS regression of EPXMA and μ-XRF data", *Journal of Chemometrics*, vol. 14, no. 5-6, pp. 751–763, Sep. 2000, ISSN: 0886-9383. DOI: https://doi.org/10.1002/1099-128X(200009/12)14:5/6<751::AID-CEM622>3.0.CO;2-D (cit. on pp. 116, 150, 158).

[114] S. Serneels, C. Croux, P. Filzmoser, and P. J. Van Espen, "Partial robust M-regression", *Chemometrics and Intelligent Laboratory Systems*, vol. 79, no. 1-2, pp. 55–64, 2005, ISSN: 01697439. DOI: 10.1016/j.chemolab.2005.04.007 (cit. on p. 116).

[115] S. G. Djorgovski, R. R. Gal, S. C. Odewahn, *et al.*, "The Palomar Digital Sky Survey (DPOSS) 1", Tech. Rep., 1998 (cit. on pp. 116, 153, 154, 156, 158).

[116] J. Raymaekers and P. Rousseeuw, *Cellwise: Analyzing data with cellwise outliers*, R package version 2.5.2, 2023 (cit. on p. 117).

[117] E. Schubert and P. J. Rousseeuw, "Fast and eager k-medoids clustering: O(k) runtime improvement of the pam, clara, and clarans algorithms", *Information Systems*, vol. 101, p. 101 804, Nov. 2021, ISSN: 0306-4379. DOI: 10.1016/J.IS.2021.101804 (cit. on p. 159).

[118] A. González-Cebrián, M. Hermenegildo, M. Climente, and A. Ferrer, "Multivariate Six Sigma: A case study in an outpatient pharmaceutical care unit", *Quality Engineering*, vol. 34, no. 2, pp. 277–289, 2022. DOI: 10.1080/08982112.2022.2042018 (cit. on p. 183).

[119] S. L. Furterer, "Applying Lean Six Sigma methods to reduce length of stay in a hospital's emergency department", *Quality Engineering*, vol. 30, pp. 389–404, 3 Jul. 2018, ISSN: 15324222. DOI: 10.1080/08982112.2018.1464657 (cit. on p. 184).

[120] A. Honda, V. Zanetti-Bernardo, M. Cecílio-Gerolamo, and M. Davis, "How lean six sigma principles improve hospital performance", *Quality Management Journal*, vol. 25, pp. 70–82, 2 2018, ISSN: 10686967. DOI: 10.1080/10686967.2018.1436349 (cit. on p. 184).

[121] Diego Tlapa and Carlos A. Zepeda-Lugo and Guilherme L. Tortorella and Yolanda A. Baez-Lopez and Jorge Limon-Romero and Alejandro Alvarado-Iniesta and Manuel I. Rodriguez-Borbon, "Effects of Lean Healthcare on Patient Flow: A Systematic Review", *Value in Health*, vol. 23, pp. 260–273, 2 2020, ISSN: 15244733. DOI: 10.1016/j.jval.2019.11.002 (cit. on p. 184).

[122] I. Font Noguera and M. J. Fernández Megía and A. J. Ferrer Riquelme and S. Balasch I Parisi and M. D. Edo Solsona and J. L. Poveda Andres, "Mejora del proceso farmacoterapéutico del paciente hospitalizado mediante la metodología Lean Seis Sigma", *Revista de Calidad Asistencial*, vol. 28, pp. 370–380, 6 2013, ISSN: 1134282X. DOI: 10.1016/j.cali.2013.04.003 (cit. on p. 184).

[123] James P Womack and Daniel T Jones, "Lean thinking-banish waste and create wealth in your corporation", *Journal of the Operational Research Society*, vol. 48, p. 1148, 11 1997, ISSN: 0160-5682 (cit. on p. 184).

[124] Kevin Linderman and Roger G Schroeder and Srilata Zaheer and Adrian S Choo, "Six Sigma: a goal-theoretic perspective", *Journal of Operations Management*, vol. 21, pp. 193–203, 2 2003, ISSN: 0272-6963. DOI: https://doi.org/10.1016/S0272-6963(02)00087-6 (cit. on p. 184).

[125] M. Harry, P. Mann, O. De Hodgins, R. Hulbert, and C. Lacke, *Practitioner's guide to statistics and lean six sigma for process improvements*. John Wiley & Sons, 2010, ISBN: 0470114940 (cit. on p. 184).

[126] Giovanni Improta and Maria Romano and Maria Vincenza Di Cicco and Anna Ferraro and Anna Borrelli and Ciro Verdoliva and Maria Triassi and Mario Cesarelli, "Lean thinking to improve emergency department throughput at aorn cardarelli hospital", *BMC Health Services Research*, vol. 18, pp. 1–9, 1 2018, ISSN: 14726963. DOI: 10.1186/s12913-018-3654-0 (cit. on p. 184).

[127] Jehni Robinson and Melody Porter and Yara Montalvo and Carol J. Peden, "Losing the wait: improving patient cycle time in primary care", *BMJ open quality*, vol. 9, 2 2020, ISSN: 23996641. DOI: 10.1136/bmjoq-2019-000910 (cit. on p. 184).

[128] Wei Min Ma and Hui Zhang and Neng Li Wang, "Improving outpatient satisfaction by extending expected waiting time", *BMC Health Services Research*, vol. 19, pp. 1–7, 1 2019, ISSN: 14726963. DOI: 10.1186/s12913-019-4408-3 (cit. on p. 184).

[129] A. Ferrer, "Multivariate six sigma: A key improvement strategy in industry 4.0", *https://doi.org/10.1080/08982112.2021.1957481*, vol. 33, pp. 758–763, 4 2021, ISSN: 15324222. DOI: 10.1080/08982112.2021.1957481 (cit. on p. 184).

[130] A. González-Cebrián, E. Almenar-Pérez, J. Xu, *et al.*, "Diagnosis of myalgic encephalomyelitis/chronic fatigue syndrome with partial least squares discriminant analysis: Relevance of blood extracellular vesicles", *Frontiers in Medicine*, vol. 9, 2022, ISSN: 2296-858X. DOI: 10.3389/fmed.2022.842991 (cit. on p. 207).

[131] T. Boerma, J. Harrison, R. Jakob, C. Mathers, A. Schmider, and S. Weber, "Revising the ICD: explaining the WHO approach", *The Lancet*, vol. 388, pp. 2476–2477, 10059 Nov. 2016, ISSN: 1474547X. DOI: 10.1016/S0140-6736(16)31851-7 (cit. on p. 208).

[132] E. W. Clayton, "Beyond myalgic encephalomyelitis/chronic fatigue syndrome: An IOM report on redefining an illness", *JAMA - Journal of the*

American Medical Association, vol. 313, pp. 1101–1102, 11 Mar. 2015, ISSN: 15383598. DOI: 10.1001/JAMA.2015.1346 (cit. on p. 208).

[133] Keiji Fukuda and Stephen E. Straus and Ian Hickie and Michael C. Sharpe and James G. Dobbins and Anthony Komaroff, "The chronic fatigue syndrome: A comprehensive approach to its definition and study", *Annals of Internal Medicine*, vol. 121, pp. 953–959, 12 Dec. 1994, ISSN: 00034819. DOI: 10.7326/0003-4819-121-12-199412150-00009 (cit. on pp. 208, 211).

[134] B. Carruthers, A. Jain, K. De Meirleir, *et al.*, "Myalgic encephalomyelitis/chronic fatigue syndrome: Clinical working case definition, diagnostic and treatment protocols", *Journal of Chronic Fatigue Syndrome*, vol. 11, pp. 7–115, 1 2003, ISSN: 10573321. DOI: 10.1300/J092V11N01_02 (cit. on pp. 208, 211).

[135] B. Carruthers, M. Van de Sande, K. De Meirleir, *et al.*, "Myalgic encephalomyelitis: International Consensus Criteria", *Journal of Internal Medicine*, vol. 270, pp. 327–338, 4 2011, ISSN: 13652796. DOI: 10.1111/J.1365-2796.2011.02428.X (cit. on p. 208).

[136] E. Almenar-Pérez, T. Sánchez-Fito, T. Ovejero, L. Nathanson, and E. Oltra, "Impact of polypharmacy on candidate biomarker mirnomes for the diagnosis of fibromyalgia and myalgic encephalomyelitis/chronic fatigue syndrome: Striking back on treatments", *Pharmaceutics*, vol. 11, 3 Mar. 2019, ISSN: 19994923. DOI: 10.3390/PHARMACEUTICS11030126 (cit. on pp. 208, 212).

[137] E. Almenar-Pérez, L. Sarria, L. Nathanson, and E. Oltra, "Assessing diagnostic value of microRNAs from peripheral blood mononuclear cells and extracellular vesicles in Myalgic Encephalomyelitis/Chronic Fatigue Syndrome", *Scientific Reports*, vol. 10, 1 Dec. 2020, ISSN: 20452322. DOI: 10.1038/S41598-020-58506-5 (cit. on pp. 208, 209, 211, 212, 219, 233, 234).

[138] A. K. Cheema, L. Sarria, M. Bekheit, *et al.*, "Unravelling myalgic encephalomyelitis/chronic fatigue syndrome (ME/CFS): Gender-specific changes in the microRNA expression profiling in ME/CFS", *Journal of Cellular and Molecular Medicine*, vol. 24, pp. 5865–5877, 10 May 2020, ISSN: 15821838. DOI: 10.1111/JCMM.15260 (cit. on p. 208).

[139] E. Nepotchatykh, W. Elremaly, I. Caraus, *et al.*, "Profile of circulating microRNAs in myalgic encephalomyelitis and their relation to symptom severity, and disease pathophysiology", *Scientific Reports*, vol. 10, 1 Dec. 2020, ISSN: 20452322. DOI: 10.1038/S41598-020-76438-Y (cit. on p. 208).

[140] 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)*, vol. 57, pp. 289–300, 1 Jan. 1995. DOI: 10.1111/J.2517-6161.1995.TB02031.X (cit. on p. 208).

[141] D. Hervás Marín, "Use of multivariate statistical methods for the analysis of metabolomic data", Oct. 2019. DOI: 10.4995/THESIS/10251/130847 (cit. on p. 208).

[142] E. Saccenti, H. C. Hoefsloot, A. K. Smilde, J. A. Westerhuis, and M. M. Hendriks, "Reflections on univariate and multivariate analysis of metabolomics data", *Metabolomics*, vol. 10, pp. 361–374, 3 2014, ISSN: 15733890. DOI: 10.1007/S11306-013-0598-6 (cit. on p. 208).

[143] J. A. Westerhuis, S. P. Gurden, and A. K. Smilde, "Generalized contribution plots in multivariate statistical process monitoring", *Chemometrics and Intelligent Laboratory Systems*, vol. 51, no. 1, pp. 95–114, 2000, ISSN: 0169-7439. DOI: 10.1016/s0169-7439(00)00062-9 (cit. on p. 209).

[144] R. A. FISHER, "The use of multiple measurements in taxonomic problems", *Annals of Eugenics*, vol. 7, pp. 179–188, 2 Sep. 1936. DOI: 10.1111/J.1469-1809.1936.TB02137.X (cit. on pp. 209, 213).

[145] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, vol. 20, pp. 273–297, 3 Sep. 1995, ISSN: 0885-6125. DOI: 10.1007/BF00994018 (cit. on pp. 209, 213).

[146] T. Fawcett, "An introduction to roc analysis", *Pattern Recognition Letters*, vol. 27, pp. 861–874, 8 Jun. 2006, ISSN: 01678655. DOI: 10.1016/J.PATREC.2005.10.010 (cit. on p. 211).

[147] E. M. Lacerda, E. W. Bowman, J. M. Cliff, *et al.*, "The uk me/cfs biobank for biomedical research on myalgic encephalomyelitis/chronic fatigue syndrome (me/cfs) and multiple sclerosis", *Open Journal of Bioresources*, vol. 4, Feb. 2017. DOI: 10.5334/OJB.28 (cit. on p. 212).

[148] E. M. Lacerda, K. Mudie, C. C. Kingdon, J. D. Butterworth, S. O'Boyle, and L. Nacul, "The uk me/cfs biobank: A disease-specific biobank for advancing clinical research into myalgic encephalomyelitis/chronic fatigue syndrome", *Frontiers in Neurology*, vol. 9, Dec. 2018, ISSN: 16642295. DOI: 10.3389/FNEUR.2018.01026 (cit. on p. 212).

[149] C. A. McHorney, J. E. Ware, and A. E. Raczek, "The MOS 36-item short-form health survey (Sf-36): II. Psychometric and clinical tests of validity in measuring physical and mental health constructs", *Medical Care*, vol. 31, pp. 247–263, 3 1993, ISSN: 15371948. DOI: 10.1097/00005650-199303000-00006 (cit. on pp. 212, 213).

[150] C. Jackson, "The general health questionnaire", *Occupational Medicine*, vol. 57, pp. 79–79, 1 Aug. 2006, ISSN: 0962-7480. DOI: 10.1093/OCCMED/KQL169 (cit. on pp. 212, 213).

[151] C. Théry and K. Witwer, "Isev2018 abstract book", *Journal of Extracellular Vesicles*, vol. 7, p. 1 461 450, sup1 Apr. 2018. DOI: 10.1080/20013078.2018.1461450 (cit. on p. 212).

[152] F. Kern, E. Aparicio-Puerta, Y. Li, *et al.*, "Mirtargetlink 2.0 - interactive mirna target gene and target pathway networks", *Nucleic Acids Research*, vol. 49, pp. 409–416, W1 Jul. 2021, ISSN: 13624962. DOI: 10.1093/NAR/GKAB297 (cit. on p. 213).

[153] M. Dash, K. Palaniyandi, S. Ramalingam, S. Sahabudeen, and N. S. Raja, "Exosomes isolated from two different cell lines using three different isolation techniques show variation in physical and molecular characteristics", *Biochimica et Biophysica Acta - Biomembranes*, vol. 1863, 2 Feb. 2021, ISSN: 18792642. DOI: 10.1016/J.BBAMEM.2020.183490 (cit. on p. 219).

[154] J. Xu, M. Potter, C. Tomas, *et al.*, "A new approach to find biomarkers in chronic fatigue syndrome/myalgic encephalomyelitis (CFS/ME) by

single-cell Raman micro-spectroscopy", *Analyst*, vol. 144, pp. 913–920, 3 Feb. 2019, ISSN: 13645528. DOI: 10.1039/C8AN01437J (cit. on p. 219).

[155] H. Horiue, M. Sasaki, Y. Yoshikawa, M. Toyofuku, and S. Shigeto, "Raman spectroscopic signatures of carotenoids and polyenes enable label-free visualization of microbial distributions within pink biofilms", *Scientific Reports*, vol. 10, 1 Dec. 2020, ISSN: 20452322. DOI: 10.1038/S41598-020-64737-3 (cit. on p. 224).

[156] C. on the Diagnostic Criteria for Myalgic Encephalomyelitis/Chronic Fatigue Syndrome, B. on the Health of Select Populations, and I. of Medicine, "Beyond myalgic encephalomyelitis/chronic fatigue syndrome", *Beyond Myalgic Encephalomyelitis/Chronic Fatigue Syndrome: Redefining an Illness*, pp. 1–282, Feb. 2015. DOI: 10.17226/19012 (cit. on p. 233).

[157] I. Murga-Gandasegui, L. Aranburu Laka, P. Ángel Gargiulo, J. Gómez-Esteban, and J. Lafuente Sánchez, "Myalgic encephalomyelitis/chronic fatigue syndrome: A neurological entity?", *Medicina (Lithuania)*, vol. 57, 10 Oct. 2021, ISSN: 16489144. DOI: 10.3390/MEDICINA57101030 (cit. on p. 233).

[158] G. Morris and M. Maes, "A neuro-immune model of Myalgic Encephalomyelitis/Chronic fatigue syndrome", *Metabolic Brain Disease*, vol. 28, pp. 523–540, 4 Dec. 2013, ISSN: 08857490. DOI: 10.1007/S11011-012-9324-8 (cit. on p. 233).

[159] A. Brown, D. Jones, M. Walker, and J. Newton, "Abnormalities of AMPK activation and glucose uptake in cultured skeletal muscle cells from individuals with chronic fatigue syndrome", *PLoS ONE*, vol. 10, 4 Apr. 2015, ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0122982 (cit. on p. 233).

[160] Y. Guo, J. Tao, Y. Li, *et al.*, "Quantitative localized analysis reveals distinct exosomal protein-specific glycosignatures: Implications in cancer cell subtyping, exosome biogenesis, and function", *Journal of the American Chemical Society*, vol. 142, pp. 7404–7412, 16 Apr. 2020, ISSN: 15205126. DOI: 10.1021/JACS.9B12182 (cit. on p. 234).

[161] K. Sapoń, I. Gawrońska, T. Janas, A. F. Sikorski, and T. Janas, "Exosome-associated polysialic acid modulates membrane potentials, membrane thermotropic properties, and raft-dependent interactions between vesicles", *FEBS Letters*, vol. 594, pp. 1685–1697, 11 Jun. 2020, ISSN: 18733468. DOI: 10.1002/1873-3468.13785 (cit. on p. 234).

[162] M. B. Monzón-Nomdedeu, K. J. Morten, and E. Oltra, "Induced pluripotent stem cells as suitable sensors for fibromyalgia and myalgic encephalomyelitis/chronic fatigue syndrome", *World Journal of Stem Cells*, vol. 13, pp. 1134–1150, 8 2021, ISSN: 19480210. DOI: 10.4252/WJSC.V13.I8.1134 (cit. on p. 234).

[163] J. Castro-Marrero, E. Serrano-Pertierra, M. Oliveira-Rodríguez, *et al.*, "Circulating extracellular vesicles as potential biomarkers in chronic fatigue syndrome/myalgic encephalomyelitis: An exploratory pilot study", *Journal of Extracellular Vesicles*, vol. 7, 1 Jan. 2018, ISSN: 20013078. DOI: 10.1080/20013078.2018.1453730 (cit. on p. 234).

[164] L. Giloteaux, A. O'Neal, J. Castro-Marrero, S. M. Levine, and M. R. Hanson, "Cytokine profiling of extracellular vesicles isolated from plasma in myalgic encephalomyelitis/chronic fatigue syndrome: A pilot study", *Journal of Translational Medicine*, vol. 18, 1 Oct. 2020, ISSN: 14795876. DOI: 10.1186/S12967-020-02560-0 (cit. on p. 234).

[165] L. Rajendran, M. Honsho, T. R. Zahn, *et al.*, "Alzheimer's disease beta-amyloid peptides are released in association with exosomes", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, pp. 11 172–11 177, 30 Jul. 2006, ISSN: 00278424. DOI: 10.1073/PNAS.0603838103 (cit. on p. 234).

[166] M. Logozzi, A. De Milito, L. Lugini, *et al.*, "High levels of exosomes expressing cd63 and caveolin-1 in plasma of melanoma patients", *PLoS ONE*, vol. 4, 4 Apr. 2009, ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0005219 (cit. on p. 234).

[167] H. Zhang, A. C. Silva, W. Zhang, H. Rutigliano, and A. Zhou, "Raman spectroscopy characterization extracellular vesicles from bovine placenta and peripheral blood mononuclear cells", *PLoS ONE*, vol. 15, 7 Jul. 2020, ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0235214 (cit. on p. 235).

[168] B. S. Holder, C. L. Tower, C. J. Jones, J. D. Aplin, and V. M. Abrahams, "Heightened pro-inflammatory effect of preeclamptic placental microvesicles on peripheral blood immune cells in humans1", *Biology of Reproduction*, vol. 86, 4 Apr. 2012, ISSN: 0006-3363. DOI: 10.1095/BIOLREPROD.111.097014 (cit. on p. 235).

[169] A. Sabapatha, C. Gercel-taylor, and D. D. Taylor, "Specific isolation of placenta-derived exosomes from the circulation of pregnant women and their immunoregulatory consequences", *American Journal of Reproductive Immunology*, vol. 56, pp. 345–355, 5-6 Nov. 2006, ISSN: 10467408. DOI: 10.1111/J.1600-0897.2006.00435.X (cit. on p. 235).

[170] S. P. Verma and D. F. H. Wallach, "Carotenoids as raman-active probes of erythrocyte membrane structure", *BBA - Biomembranes*, vol. 401, pp. 168–176, 2 Aug. 1975, ISSN: 00052736. DOI: 10.1016/0005-2736(75)90301-6 (cit. on p. 235).

[171] B. J. Bulkin, "Raman spectroscopic study of human erythrocyte membranes", *BBA - Biomembranes*, vol. 274, pp. 649–651, 2 Aug. 1972, ISSN: 00052736. DOI: 10.1016/0005-2736(72)90214-3 (cit. on p. 235).

[172] A. K. Saha, B. R. Schmidt, J. Wilhelmy, *et al.*, "Red blood cell deformability is diminished in patients with Chronic Fatigue Syndrome", *Clinical Hemorheology and Microcirculation*, vol. 71, pp. 113–116, 1 2019, ISSN: 18758622. DOI: 10.3233/CH-180469 (cit. on p. 235).

[173] O. Linderkamp, P. Y. Wu, and H. J. Meiselman, "Deformability of density separated red blood cells in normal newborn infants and adults", *Pediatric Research*, vol. 16, pp. 964–968, 11 1982, ISSN: 15300447. DOI: 10.1203/00006450-198211000-00013 (cit. on p. 235).

[174] J. Fiedor, M. Przetocki, A. Siniarski, *et al.*, "beta–Carotene–Induced Alterations in Haemoglobin Affinity to O2", *Antioxidants*, vol. 10, pp. 1–2, 3 Mar. 2021, ISSN: 20763921. DOI: 10.3390/ANTIOX10030451 (cit. on p. 235).

[175] I. Helwa, J. Cai, M. D. Drewry, *et al.*, "A comparative study of serum exosome isolation using differential ultracentrifugation and three commercial reagents", *PLoS ONE*, vol. 12, 1 Jan. 2017, ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0170628 (cit. on p. 235).

[176] L. Nacul, B. De Barros, C. Kingdon, *et al.*, "Evidence of clinical pathology abnormalities in people with myalgic encephalomyelitis/chronic fatigue syndrome (ME/CFS) from an analytic cross-sectional study", *Diagnostics*, vol. 9, 2 2019, ISSN: 20754418. DOI: 10.3390/DIAGNOSTICS9020041 (cit. on p. 236).

[177] A. González-Cebrián, J. Borràs-Ferrís, J. P. Ordovás-Baines, *et al.*, "Machine-learning-derived predictive score for early estimation of COVID-19 mortality risk in hospitalized patients", *PLOS ONE*, vol. 17, no. 9, pp. 1–17, Sep. 2022. DOI: 10.1371/journal.pone.0274171 (cit. on p. 239).

[178] *WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data* (cit. on p. 240).

[179] E. Burn, C. Tebé, S. Fernandez-Bertolin, *et al.*, "The natural history of symptomatic COVID-19 during the first wave in Catalonia.", eng, *Nature communications*, vol. 12, no. 1, p. 777, Feb. 2021, ISSN: 2041-1723 (Electronic). DOI: 10.1038/s41467-021-21100-y (cit. on p. 240).

[180] J. N. Gustine and D. Jones, "Immunopathology of Hyperinflammation in COVID-19.", eng, *The American journal of pathology*, vol. 191, no. 1, pp. 4–17, Jan. 2021, ISSN: 1525-2191 (Electronic). DOI: 10.1016/j.ajpath.2020.08.009 (cit. on p. 240).

[181] S. R. Knight, A. Ho, R. Pius, *et al.*, "Risk stratification of patients admitted to hospital with covid-19 using the ISARIC WHO Clinical Characterisation Protocol: development and validation of the 4C Mortality Score", *BMJ*, vol. 370, p. 22, 2020. DOI: 10.1136/bmj.m3339 (cit. on pp. 240, 258, 259).

[182] L. Wynants, B. Van Calster, G. S. Collins, *et al.*, "Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal", *BMJ*, vol. 369, p. m1328, Apr. 2020. DOI: 10.1136/bmj.m1328 (cit. on p. 240).

[183] P. M. E. L. van Dam, N. Zelis, S. M. J. van Kuijk, *et al.*, "Performance of prediction models for short-term outcome in COVID-19 patients in the emergency department: a retrospective study", *Annals of Medicine*, vol. 53, no. 1, pp. 402–409, 2021, PMID: 33629918. DOI:

10.1080/07853890.2021.1891453. eprint: https://doi.org/10.1080/07853890.2021.1891453 (cit. on p. 240).

[184] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, ser. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc., Mar. 1992, ISBN: 9780471725299. DOI: 10.1002/0471725293 (cit. on p. 245).

[185] B. Schölkopf, A. J. Smola, and F. Bach, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001, p. 664, ISBN: 0262194759 (cit. on p. 245).

[186] B. Van Calster, D. J. McLernon, M. Van Smeden, *et al.*, "Calibration: The Achilles heel of predictive analytics", *BMC Medicine*, vol. 17, no. 1, pp. 1–7, 2019, ISSN: 17417015. DOI: 10.1186/s12916-019-1466-7 (cit. on pp. 245, 246).

[187] A. González-Cebrián, J. Borràs-Ferrís, J. P. Ordovás-Baines, *et al.*, *Procovid dataset*, Aug. 2022. DOI: 10.5281/ZENODO.6948496 (cit. on p. 246).

[188] Spanish Society of Hospital Pharmacy, "Spanish Registry of treatment efficacy against SARS-CoV-2 COVID-19", European Network of Centres for Pharmacoepidemiology and Pharmacovigilance, Jerez de la Frontera, Tech. Rep., 2020 (cit. on p. 246).

[189] P. A. Harris, R. Taylor, B. L. Minor, *et al.*, "The REDCap consortium: Building an international community of software platform partners.", eng, *Journal of biomedical informatics*, vol. 95, p. 103 208, Jul. 2019, ISSN: 1532–0480 (Electronic). DOI: 10.1016/j.jbi.2019.103208 (cit. on p. 246).

[190] A. Vaid, S. Somani, A. J. Russak, *et al.*, "Machine Learning to Predict Mortality and Critical Events in a Cohort of Patients With COVID-19 in New York City: Model Development and Validation.", eng, *Journal of medical Internet research*, vol. 22, no. 11, pp. 1438–8871, Nov. 2020. DOI: 10.2196/24018 (cit. on p. 258).

[191] R. Murri, J. Lenkowicz, C. Masciocchi, *et al.*, "A machine-learning parsimonious multivariable predictive model of mortality risk in patients

with Covid-19", *Scientific Reports*, vol. 11, no. 1, p. 21 136, 2021, ISSN: 2045–2322. DOI: 10.1038/s41598-021-99905-6 (cit. on p. 258).

[192] J. L. Domínguez-Olmedo, Á. Gragera-Martínez, J. Mata, and V. Pachón Álvarez, "Machine Learning Applied to Clinical Laboratory Data in Spain for COVID-19 Outcome Prediction: Model Development and Validation", *Journal of Medical Internet Research*, vol. 23, no. 4, e26211, Apr. 2021, ISSN: 1438–8871. DOI: 10.2196/26211 (cit. on p. 258).

[193] J. Torres-Macho, P. Ryan, J. Valencia, *et al.*, *The PANDEMYC Score. An Easily Applicable and Interpretable Model for Predicting Mortality Associated With COVID-19*, 2020. DOI: 10.3390/jcm9103066 (cit. on pp. 258, 259).

[194] D. A. Berry, A. Ip, B. E. Lewis, *et al.*, "Development and validation of a prognostic 40-day mortality risk model among hospitalized patients with COVID-19", *PLOS ONE*, vol. 16, no. 7, P. Abete, Ed., e0255228, Jul. 2021, ISSN: 1932–6203. DOI: 10.1371/journal.pone.0255228 (cit. on p. 258).

[195] K. Hajifathalian, R. Z. Sharaiha, S. Kumar, *et al.*, "Development and external validation of a prediction risk model for short-term mortality among hospitalized U.S. COVID-19 patients: A proposal for the COVID-AID risk tool", *PLOS ONE*, vol. 15, no. 9, Y. R. Kou, Ed., e0239536, Sep. 2020, ISSN: 1932–6203. DOI: 10.1371/journal.pone.0239536 (cit. on p. 258).

[196] X. Guan, B. Zhang, M. Fu, *et al.*, "Clinical and inflammatory features based machine learning model for fatal risk prediction of hospitalized COVID-19 patients: results from a retrospective cohort study", *Annals of Medicine*, vol. 53, no. 1, pp. 257–266, Jan. 2021, ISSN: 0785–3890. DOI: 10.1080/07853890.2020.1868564 (cit. on p. 258).

[197] W. Liang, H. Liang, L. Ou, *et al.*, "Development and Validation of a Clinical Risk Score to Predict the Occurrence of Critical Illness in Hospitalized Patients With COVID-19", *JAMA Internal Medicine*, vol. 180, no. 8, pp. 1081–1089, Aug. 2020, ISSN: 2168-6106. DOI: 10.1001/jamainternmed.2020.2033 (cit. on pp. 258, 259).

[198] A. S. Yadaw, Y. Li, S. Bose, R. Iyengar, S. Bunyavanich, and G. Pandey, "Clinical features of COVID-19 mortality: development and validation of a clinical prediction model", *The Lancet Digital Health*, vol. 2, no. 10, e516–e525, Oct. 2020, ISSN: 2589-7500. DOI: 10.1016/S2589-7500(20)30217-X (cit. on pp. 258, 259).

[199] G. Halasz, M. Sperti, M. Villani, *et al.*, "A Machine Learning Approach for Mortality Prediction in COVID-19 Pneumonia: Development and Evaluation of the Piacenza Score", *Journal of Medical Internet Research*, vol. 23, no. 5, e29058, May 2021, ISSN: 1438-8871. DOI: 10.2196/29058 (cit. on p. 258).

[200] H. Chubb, S. E. Williams, J. Whitaker, J. L. Harrison, R. Razavi, and M. O. Neill, "Diagnostic Electrophysiology & Ablation Cardiac Electrophysiology Under MRI Guidance : an Emerging Technology Diagnostic Electrophysiology & Ablation", *Arrhythmia & Electrophysiology Review*, vol. 6, no. Ivc, pp. 85–93, 2017, ISSN: 2050-3369. DOI: 10.15420/aer.2017 (cit. on p. 258).

[201] B. M. Henry, M. H. S. de Oliveira, S. Benoit, M. Plebani, and G. Lippi, "Hematologic, biochemical and immune biomarker abnormalities associated with severe illness and mortality in coronavirus disease 2019 (COVID-19): a meta-analysis", *Clinical Chemistry and Laboratory Medicine (CCLM)*, vol. 58, no. 7, pp. 1021–1028, 2020. DOI: doi:10.1515/cclm-2020-0369 (cit. on p. 258).

[202] R. Chen, W. Liang, M. Jiang, *et al.*, "Risk Factors of Fatal Outcome in Hospitalized Subjects With Coronavirus Disease 2019 From a Nationwide Analysis in China.", eng, *Chest*, vol. 158, no. 1, pp. 97–105, Jul. 2020, ISSN: 1931–3543 (Electronic). DOI: 10.1016/j.chest.2020.04.010 (cit. on p. 258).

[203] J. Berenguer, P. Ryan, J. Rodríguez-Baño, *et al.*, "Characteristics and predictors of death among 4035 consecutively hospitalized patients with COVID-19 in Spain", *Clinical Microbiology and Infection*, vol. 26, pp. 1525–1536, 11 Nov. 2020, ISSN: 1198-743X. DOI: 10.1016/j.cmi.2020.07.024 (cit. on p. 259).

[204] J. M. Casas-Rojo, J. M. Antón-Santos, J. Millán-Núñez-Cortés, *et al.*, "Clinical characteristics of patients hospitalized with COVID-19 in Spain:

Results from the SEMI-COVID-19 Registry.", eng, *Revista clinica espanola*, vol. 220, no. 8, pp. 480–494, Nov. 2020, ISSN: 2254-8874 (Electronic). DOI: 10.1016/j.rce.2020.07.003 (cit. on p. 259).

[205] T. E. O. T. L. Group, "Learning from a retraction", *The Lancet*, vol. 396, no. 10257, p. 1056, 2020, ISSN: 0140–6736. DOI: 10.1016/s0140-6736(20)31958-9 (cit. on p. 259).

[206] A. Castro-Balado, I. Varela-Rey, E. J. Bandín-Vilar, *et al.*, "Clinical research in hospital pharmacy during the fight against COVID-19.", *Farmacia hospitalaria: organo oficial de expresion cientifica de la Sociedad Espanola de Farmacia Hospitalaria*, vol. 44, no. 7, pp. 66–70, 2020, ISSN: 1130-6343 (cit. on p. 260).

[207] A. González-Cebrián, J. Borràs-Ferrís, Y. Boada, A. Vignoni, A. Ferrer, and J. Picó, "Platero: A calibration protocol for plate reader green fluorescence measurements", *Frontiers in Bioengineering and Biotechnology*, 2023. DOI: 10.1371/journal.pone.0274171 (cit. on p. 261).

[208] Y. Boada, A. Vignoni, I. Alarcon-Ruiz, *et al.*, "Characterization of Gene Circuit Parts Based on Multiobjective Optimization by Using Standard Calibrated Measurements", *ChemBioChem*, vol. 20, no. 20, 2019, ISSN: 14397633. DOI: 10.1002/cbic.201900272 (cit. on pp. 262, 263, 265).

[209] J. Beal, G. S. Baldwin, N. G. Farny, *et al.*, "Comparative analysis of three studies measuring fluorescence from engineered bacterial genetic constructs", *PloS one*, vol. 16, no. 6, e0252263, 2021. DOI: 10.1371/journal.pone.0252263 (cit. on pp. 262, 263, 265).

[210] J. Beal, T. Haddock-Angelli, G. Baldwin, *et al.*, "Quantification of Bacterial Fluorescence using Independent Calibrants", *PLoS ONE*, vol. 13, e0199432, 6 Jun. 2018 (cit. on p. 263).

[211] A. Vignoni and Y. Boada, *Platero – green fluorescence calibration in plate readers. protocols.io*, 2022. DOI: 10.17504/protocols.io.5qpvobdwdl4o/v1 (cit. on p. 265).

[212] J. Beal, C. Telmer, A. Vignoni, *et al.*, "Multicolor Plate Reader Fluorescence Calibration", *Synthetic Biology*, Jul. 2022, ysac010, ISSN: 2397–7000. DOI: 10.1093/synbio/ysac010. eprint: https://academic.oup.

com/synbio/advance-article-pdf/doi/10.1093/synbio/ysac010/45084330/ysac010.pdf (cit. on p. 265).

[213] A. Zanobini, B. Sereni, M. Catelani, and L. Ciani, "Repeatability and Reproducibility techniques for the analysis of measurement systems", *Measurement: Journal of the International Measurement Confederation*, vol. 86, pp. 125–132, 2016, ISSN: 02632241. DOI: 10.1016/j.measurement.2016.02.041 (cit. on pp. 272, 274).

[214] D. C. Montgomery, *Introduction to statistical quality control*. John Wiley & Sons, 2020, ISBN: 1119723094 (cit. on p. 272).

[215] A. González-Cebrián, J. Borràs-Ferrís, Y. Boada, A. Vignoni, A. Ferrer, and J. Picó, *PLATERO: calibration and analysis of GFP plate readers. Experimental data sets and test software*, version v1.0, Oct. 2022. DOI: 10.5281/zenodo.7224497 (cit. on pp. 275, 276, 291).

[216] E. George, W. G. Hunter, and J. S. Hunter, *Statistics for experimenters: Design, innovation, and discovery*. Wiley, 2005 (cit. on p. 278).

[217] B. Le Roux and H. Rouanet, *Multiple correspondence analysis*. Sage, 2010, vol. 163 (cit. on p. 306).

[218] M. Greenacre and J. Blasius, *Multiple correspondence analysis and related methods*. CRC press, 2006 (cit. on p. 306).

[219] Y. Song, J. A. Westerhuis, N. Aben, L. F. Wessels, P. J. Groenen, and A. K. Smilde, "Generalized simultaneous component analysis of binary and quantitative data", *Journal of Chemometrics*, vol. 35, no. 3, e3312, 2021 (cit. on p. 306).

[220] J. Engel, G. Postma, I. van Peufflik, L. Blanchet, and L. Buydens, "Pseudo-sample trajectories for variable interaction detection in dissimilarity partial least squares", *Chemometrics and Intelligent Laboratory Systems*, vol. 146, pp. 89–101, Aug. 2015, ISSN: 01697439. DOI: 10.1016/j.chemolab.2015.05.010 (cit. on p. 312).