

Development of modern chemometrics methods for the analysis of high-dimensional data issued from process monitoring and experimental studies

Michel Thiel

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Sciences

November 2025

Institute of Statistics, Biostatistics and Actuarial Sciences
Louvain Institute of Data Analysis and Modeling
Université catholique de Louvain
Louvain-la-Neuve
Belgium

Thesis committee:

Prof. Bernadette Govaerts, Supervisor (UCLouvain, Belgium)
Prof. Rainer von Sachs, Supervisor (UCLouvain, Belgium)
Prof. Laura Symul, Chairperson (UCLouvain, Belgium)
Dr. Manon Martin, Secretary (UCLouvain, Belgium)
Prof. Helena Geys (UHasselt, Belgium)
Dr. François Stevens (CRA-W, Belgium)

Acknowledgements

Je voudrais d'abord remercier Bernadette pour son soutien, son encadrement et son dynamisme durant toutes ces années. De nos premiers échanges au master de statistique jusqu'à aujourd'hui, tout n'a été que de bons souvenirs : les articles co-écrits, les visites à Beerse, les voyages pour des conférences, et bien d'autres moments. Merci à toi de m'avoir encouragé jusqu'au bout.

Je souhaite également remercier Helena Geys et Janssen Pharmaceutica d'avoir initié ce projet, et de m'avoir donné la liberté de recherche durant les premières années. Je garde un souvenir stimulant de cette période de collaboration et d'échanges avec Tanya, Tor et nombreux collègues sur mes projets en chimiométrie.

Je remercie ensuite les membres de la faculté de statistique : Manon, Rainer, Nancy, Céline, Catherine, l'équipe du SMCS... Même si je ne passais pas souvent, je garde de chaque visite un souvenir chaleureux de nos échanges.

Enfin, je remercie ma famille, qui m'a toujours soutenu et encouragé dans mes avancées. À ma grand-mère, qui me demandait régulièrement « Eh bien, quand vas-tu finir cette thèse ? », je dédie ce travail : c'est aussi grâce à toi qu'il se termine.

Contents

General Introduction	1
1 Foundations and extensions of ASCA methodologies for multivariate designs	11
1.1 Introduction	12
1.2 Methods	15
1.2.1 ANOVA and the General Linear Model GLM	15
1.2.2 ASCA and APCA	21
1.2.3 ASCA+ and APCA+ principles	24
1.3 Case study	29
1.3.1 Balanced Dataset 1 analysis	32
1.3.2 Unbalanced Dataset 2 analysis	36
1.3.3 Balanced Dataset 1 analysis with all factors	38
1.4 Recent developments in ASCA Methodology	42
1.4.1 Rebalancing Strategies for Unbalanced Designs	42
1.4.2 LiMM-PCA: Integrating Dimensionality Reduction with Mixed Models	42
1.4.3 Extending ASCA to Longitudinal Data Analysis	43
1.4.4 Extension to Generalized Linear Models	44
1.5 Conclusion and perspectives	44
2 Practical implementation of ASCA methodologies: the limpca package	47

2.1	Introduction	48
2.2	Limpca package organization and functions	50
2.2.1	Data exploration	52
2.2.2	Modeling and results visualization	53
2.2.3	Application on Urine-Citrate-Hippurate dataset	55
2.2.4	Limpca results	57
2.3	Extension to continuous variables	65
2.3.1	Methodology	65
2.3.2	Application on Sacurine dataset	66
2.3.3	Data visualization	67
2.3.4	ASCA modeling	67
2.3.5	Adaptation in limpca	71
2.4	Measurement of variable importance	73
2.4.1	Motivation	73
2.4.2	Methodology	73
2.4.3	Application on Urine-Citrate-Hippurate dataset	75
2.4.4	Application on Sacurine dataset	77
2.4.5	Methodology in (Nueda et al., 2007)	78
2.4.6	Adaptation in limpca	81
2.5	Confidence ellipsoids	83
2.5.1	Motivation	83
2.5.2	Methodology	83
2.5.3	Application on Urine-Citrate-Hippurate dataset	85
2.5.4	Application on Sacurine dataset	86
2.5.5	Pairwise comparisons of effect level means	87
2.5.6	Adaptation in limpca	89
2.6	Conclusion and perspectives	90
3	Unsupervised chemometric approaches for monitoring chemical reactions: PCA, NMF and MCR	91
3.1	Introduction	92
3.2	PAT data structure	94
3.2.1	Dataset	94
3.3	Methods	95
3.3.1	PCA, NMF and MCR dimension-reduction methods	97
3.3.2	PCA	98
3.3.3	NMF	99
3.3.4	MCR	105

CONTENTS

3.3.5	Methods summary	108
3.3.6	Comparison of dimension reduction results on batch data	109
3.3.7	Multivariate calibration method PLSR	111
3.3.8	Prediction model set-up and comparison	113
3.4	Case study	116
3.4.1	Visualization and interpretation of chemical reactions . .	116
3.4.2	Chemical components prediction models	127
3.5	Conclusion	132
3.6	Appendix	133
4	Supervised chemometric approaches for interpretable and predictive modeling: SNMF	135
4.1	Introduction and objectives	136
4.2	Data set	137
4.2.1	Artificial dataset	138
4.2.2	True quantities	138
4.2.3	True profile	139
4.2.4	Observed spectra	140
4.2.5	Batch generation	141
4.3	Supervised NMF in the literature	143
4.3.1	Supervised NMF for ICU Mortality Classification	143
4.3.2	Semi-Supervised NMF for EEG Classification Tasks	144
4.3.3	Supervised NMF for MALDI Spectrometry Data Classification	144
4.3.4	VC-PCR for Prediction with Variable Clustering	145
4.3.5	Discussion and Comparison	146
4.4	New algorithm for Supervised NMF	147
4.4.1	Objective function	147
4.4.2	Deriving the gradients	147
4.4.3	Gradient descent rules	148
4.4.4	Multiplicative Update rules	149
4.5	Method application on artificial dataset	153
4.5.1	Methodology	153
4.5.2	Application to artificial dataset	155
4.6	Conclusion	170
5	General conclusions	173
5.1	Question 1: How to analyze multivariate data from designed experiments?	174

5.2	Question 2: How to analyze data issued from multivariate process monitoring over time?	175
5.3	Overall contributions and future directions	176
6	General Appendices	179
6.1	Appendix A: Abbreviations	180
6.2	Appendix B: Mathematical notations	182
6.2.1	Notations used in Chapters 1 & 2	182
6.2.2	Notations used in Chapter 3 & 4	182
	Bibliography	184

General Introduction

Context and Motivation

In today's world driven by data, multivariate datasets are increasingly encountered across scientific and industrial domains. Arising from advanced analytical techniques in fields such as biology, chemistry, and manufacturing, these datasets often encompass high-dimensional measurements (e.g., thousands of spectral peaks), with potential noise and correlations. Emerging in the 1970s as a fusion of chemistry, statistics, and computer science, chemometrics plays an essential role in this landscape, enabling the extraction of meaningful patterns and insights (Brereton, 2003).

The core challenge addressed here is to derive meaningful information from complex multivariate datasets generated during processes or experiments, where traditional univariate methods fall short due to inter-variable dependencies and high dimensionality. Motivations stem from real-world needs: in industry, ensuring product quality through real-time monitoring, and in omics (disciplines studying biological molecules), elucidating biological mechanisms from observational data that are not always tied to a predefined design but arise from natural variations. This thesis focuses on developing and extending chemometric methodologies to address challenges in analyzing such data. The applications presented here focus on omics research and pharmaceutical processes but can be extended more broadly to Research and Development (R&D), production, analytical chemistry, and other multivariate data scenarios.

The pharmaceutical sector, in particular, has embraced frameworks like Quality-by-Design (QbD) and Process Analytical Technology (PAT) to ensure product quality through systematic understanding and control of manufacturing processes. Introduced by the International Council for Harmonisation (ICH) guidelines such as Q8(R2), Q9, and Q10, QbD emphasizes building quality into products from the outset by identifying critical quality attributes (CQAs) and critical process parameters (CPPs), rather than relying solely on end-product testing (International Conference on Harmonisation (ICH), 2009; International Conference on Harmonisation (ICH), 2005; International Conference on Harmonisation (ICH), 2008). The Q8 guideline involves defining a design space to build pharmaceutical quality into the product. This design space is a multidimensional region where process parameters can vary without impacting product quality. The Q9 guideline further formalizes the role of quality risk management to guide decisions and ensure patient safety across the product lifecycle. Complementarily, Q10 promotes continual improvement and lifecycle management in the pharmaceutical quality system.

As part of the QbD framework, the PAT initiative promoted by the Food and Drug Administration (FDA) forms the basis of the Good Manufacturing Practice (GMP) rules for the 21st century (De Beer et al., 2008). It has encouraged companies to increase research into and the use of new analytical technologies to perform constant monitoring of the CQAs of raw materials and intermediates, allowing process understanding and control (United States Food and Drug Administration (FDA), 2009; Schaefer et al., 2014).

Matching those PAT requirements, spectroscopic methods such as near-infrared (NIR) and mid-infrared (MIR) have become widely used to facilitate process understanding and control across R&D and production. Some advantages include, for example, high-speed acquisition, minimization of sample preparation, non-destructive nature, and measurements of both physical and chemical properties (Ziémons et al., 2010; Schaefer et al., 2014). In parallel, omics sciences (such as metabolomics, which studies small-molecule metabolites in biological systems) generate vast multivariate datasets from techniques like nuclear magnetic resonance (NMR) or mass spectrometry (MS). While QbD often involves designed experiments, omics data also emerge from observational studies where no explicit experimental manipulation occurs (Madsen, Lundstedt, and Trygg, 2010). Whether used for real-time release testing in industry, or in omics for exploratory biomarker discovery, chemometrics emerges as critical to extract meaningful information from these complex multivariate datasets.

Embedded in this industrial and scientific context, this PhD thesis was initiated as a collaboration between Quantitative Sciences (QS) and Pharmaceutical Development Manufacturing Sciences (PDMS) groups of Janssen Pharmaceutica on the one hand, and Institut de Statistique, Biostatistique et Sciences Actuarielles (ISBA) of Université catholique de Louvain (UCLouvain) on the other hand. This research originated from practical challenges raised by the scientists during pharmaceutical development and omics research, where new statistical tools were needed.

From these challenges, two distinct questions were identified:

1. *How to analyze multivariate data from designed experiments?*
2. *How to analyze data issued from multivariate process monitoring over time?*

For both questions, complex datasets are generated which require the study and development of appropriate chemometrics methods. The structure of this thesis and how those two questions are addressed are detailed in the next sections.

Structure of the thesis

The thesis is organized in two major parts:

Question 1 is addressed in Chapters 1 and 2. They focus on the analysis of high-dimensional data collected through experimental designs and observational studies in omics research. An overview of ASCA+ (for ANOVA-Simultaneous Component Analysis) and APCA+ (for ANOVA-Principal Component Analysis), methodologies developed to tackle this problem, is shown in Figure 1. The idea is to study the impact of experimental factors on a multivariate response matrix by decomposing it into effect matrices estimated with a General Linear Model (GLM). Those effect matrices are then visualized in a lower-dimensional space using Principal Component Analysis (PCA), and the percentage of variance and significance evaluated for each effect.

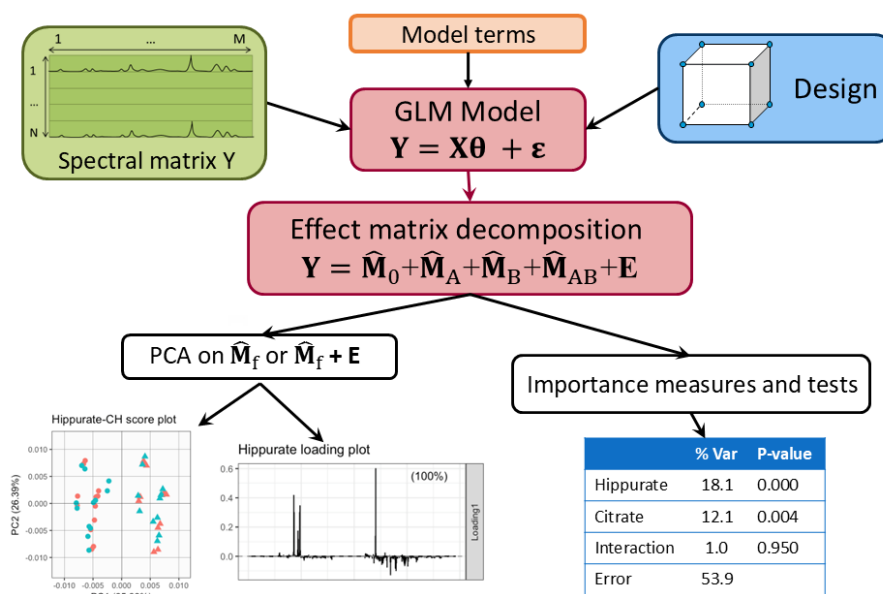


Figure 1: Overview of ASCA+ and APCA+ methodologies.

Question 2 is addressed in Chapters 3 and 4, which treat real-time spectroscopic data acquired during the monitoring of chemical reactions in manufacturing environments. Figure 2 presents how processes can be monitored over time by applying dimension-reduction methods on spectral data (real-time on-line measurements), or calibration models developed on both spectral and off-line data (laboratory analyses of removed samples).

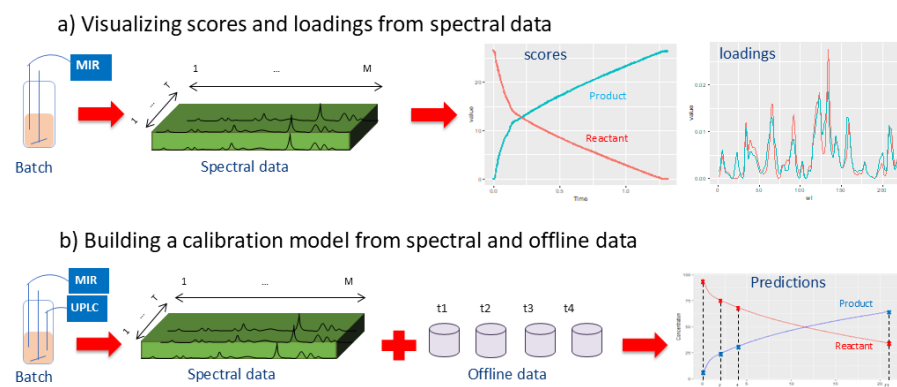


Figure 2: Visualizing reactions from spectral data (a) and predicting components from spectral and off-line data (b).

The following sections will introduce the different chapters of this thesis and how those two questions are addressed.

Chapter 1: Foundations and extensions of ASCA methodologies for multivariate designs

Chapter 1 relies on the article *ASCA+ and APCA+: Extensions of ASCA and APCA in the analysis of unbalanced multifactorial designs* published in *Journal of Chemometrics* in 2017 (Thiel, Féraud, and Govaerts, 2017). It was the first publication for this thesis and addresses the challenge of analyzing multivariate data from designed experiments, a scenario encountered in many scientific fields where high-dimensional and correlated datasets are present. Traditional methods such as ANOVA are limited to univariate responses, and multivariate ANOVA (MANOVA) is not adapted when the number of variables exceeds the number of observations. To address these issues, this chapter digs into the theory of ASCA and APCA, and introduces their enhanced versions, ASCA+ and APCA+, which are developed to handle unbalanced designs as well as more complex designs.

The chapter begins by reviewing the principles of ANOVA within the GLM framework, demonstrating why classical ANOVA estimators, based on simple differences of means, yield biased results in unbalanced designs. ASCA+ methodology overcomes this by using least squares estimators derived from GLM, decomposing a multivariate response matrix into effect matrices for experimental factors and their interactions. PCA is then applied to these effect

matrices to facilitate visualization and interpretation in a reduced dimensional space. The chapter details the use of permutation tests to assess effect significance and Type III sums of squares to quantify the percentage of variance explained, ensuring statistical inference.

A practical example in metabolomics called Urine-Citrate-Hippurate (*UCH*) dataset is used. It illustrates the ability of ASCA+ to disentangle factor effects in a complex design involving ^1H -NMR spectra from rat urine samples spiked with different concentrations of chemicals hippurate and citrate. At the time this work was presented at Janssen Pharmaceutica, several internal datasets were investigated or deliberately generated. However, even though this example originates from the field of omics, it constitutes an ideal case study to prove the advantages of ASCA+ over classical ASCA. A recent review of the literature on the subject is also included.

It can also be stated that the article of Thiel, Féraud, and Govaerts (2017) has had a significant impact on the chemometrics community. As of this writing, it has been cited 64 times in multiple practical implementations and serves as a reference for papers focusing on methodological extensions.

Chapter 2: Practical implementation of ASCA methodologies: the *limpca* package

Chapter 2 is based on the article *limpca: An R package for the linear modeling of high-dimensional designed data based on ASCA/APCA family of methods* (Thiel et al., 2023a), published in *Journal of Chemometrics* in 2023. It translates the theoretical advancements of Chapter 1 into a practical tool by introducing the R package *limpca*. Indeed, after the publication of Thiel, Féraud, and Govaerts (2017) recurring requests came from the scientific community asking whether a user-friendly software was available and it was decided to implement it. This implementation should be seen as a collaborative work between industry and academia. It started with me, with the support of Robin Van Oirbeek, consultant at Janssen Pharmaceutica at that moment, who implemented the formulae of the ASCA+ methodology in a first package version. A key step was to automate the GLM decomposition depending on the model formula. The work was then taken over by Sébastien Franceschini and Nadia Benaïche, master students at ISBA, who progressed in the structure of the different functions available. Finally, a major contribution is due to Manon Martin, who consolidated it as a fully functional R package on Bioconductor

(Thiel et al., 2023b).

The package `limpca` implements ASCA+ and APCA+ methodologies, automates the decomposition of multivariate response matrices, and provides visualizations that enhance interpretability for the users. It also leaves room for several innovative features which are ready to be implemented. Those innovations are the extension to continuous factors, graphical outputs with confidence ellipses rather than data ellipses, and measures of variable importance. These functionalities address practical challenges, such as expanding ASCA+ to new experimental contexts, assessing statistical significance visually, and bridging the gap between univariate and multivariate analysis.

In this chapter, the possibilities offered by the package `limpca` are shown on two different datasets. A different subset of the *UCH* dataset was analyzed than the one in Chapter 1, taking different concentrations of hippurate and citrate. A second dataset called *Sacurine* which investigated the variations in the human urinary metabolome is used. It combines categorical and continuous variables such as the age of subjects, their body mass index, and gender. This serves as a great example extending to continuous factors, but also highlights the package's flexibility to apply ASCA+ and APCA+ to an observational study.

Chapter 3: Unsupervised chemometrics approaches for monitoring chemical reactions: PCA, NMF and MCR

Chapter 3 tackles the second core question of this thesis: how to analyze multivariate spectroscopic data for real-time monitoring of chemical reactions in pharmaceutical manufacturing. This work was published in *Chemometrics and Intelligent Laboratory Systems* in 2021 (Thiel et al., 2021). The study was performed in close collaboration with Janssen Pharmaceutica's colleagues Tastiana Khamiakova, Tor Maes, Nicolas Sauwen, and UCLouvain with Bernadette Govaerts.

The study presents a framework to achieve two objectives using a real case-study dataset of 11 experiments from scientists from PDMS. This dataset, generated in a developmental context, was monitored on-line via mid-infrared (MIR) spectroscopy and off-line via Ultra-Performance Liquid Chromatography (UPLC). The first objective is to visualize and interpret chemical reaction kinetics using dimension-reduction methods on the MIR data: PCA, Non-Negative Matrix Factorization (NMF), and Multivariate Curve Resolution

(MCR). PCA maximizes the variance in the decomposition but lacks chemical interpretability. On the other hand, NMF and MCR with non-negativity constraints generate loadings similar to pure spectra and scores aligning with UPLC profiles. The second objective is to predict chemical component concentrations over time. Partial Least Squares Regression (PLSR) is applied in a one-step approach and compared to two-step approaches combining multivariate regression with PCA, NMF, or MCR scores. Both objectives incorporate model validation, including a leave-one-batch-out approach.

This framework supports real-time process understanding under the Process Analytical Technology initiative and sets the stage for Chapter 4's supervised methods.

Chapter 4: Supervised chemometrics approaches for interpretable and predictive modeling: SNMF

Chapter 4 is based on the master thesis of Julien Van Roy (Roy, 2021) which was supervised by myself, Bernadette Govaerts and Rainer von Sachs. This master continues the ongoing collaboration between Janssen Pharmaceutica and UCLouvain and is in direct continuity of the previous chapter. Chapter 3 demonstrated that PLSR outperforms the two-step approaches combining PCA, NMF, or MCR with multivariate regression in terms of predictions but produced less interpretable loadings than NMF and MCR, motivating the need for a hybrid method.

The primary objective of Chapter 4 is to develop and evaluate a new extension of NMF that integrates a supervised penalty term to balance prediction accuracy and chemical interpretability. Specifically, the supervision term is constructed as $\frac{\lambda}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2$, where $\mathbf{Y}$ denotes the matrix of reference concentrations (off-line data), $\mathbf{H}$ represents the NMF scores (estimated concentrations over time), $\mathbf{B}$ is the matrix of regression coefficients, and λ is a tuning parameter controlling the trade-off with the NMF reconstruction error. This extension is called Supervised Non-negative Matrix Factorization (SNMF), and is compared to PLSR as well as the two-step NMF + regression approach. To ensure confidentiality, an artificial dataset simulating a reaction with three components is employed. The chapter details SNMF's algorithmic implementation, compares its predictive power to that of PLSR and assesses whether chemical interpretability is retained.

This work underscores the potential of supervised chemometrics for real-time release testing within the Process Analytical Technology framework. By combining the predictive strength of PLSR with the interpretability of NMF, Chapter 4 paves the way for future hybrid modeling approaches, offering practical value for pharmaceutical process control.

CHAPTER 1

Foundations and extensions of ASCA methodologies for
multivariate designs

1.1 Introduction

Many modern analytical methods are used to analyze samples resulting from an experimental design, for example in medical, biological, chemical, or agronomic fields. Those methods generate most of the time highly multivariate data like spectra or images, where the number of variables (response descriptors) tends to be much larger than the number of experimental units (Jansen et al., 2005). It is in this context that “omics” technologies like metabolomics, proteomics, transcriptomics or genomics have emerged in the '90s. Due to their high complexity they have boosted new methodological developments in related scientific disciplines such as in chemometrics.

In biology, it is known for years that experimental designs are the most efficient way to extract a maximal amount of information using the fewest number of experiments (Trygg, Holmes, and Lundstedt, 2007; Fisher, 1935).

A common method to analyse the effect of several factors on a single variable is ANOVA (ANalysis Of VAriance) which decomposes its variance into the ones explained by each factor and model effect (Neter et al., 2004). However, in most of the omics purposes, multiple variables are measured to describe phenomena and a more sophisticated approach needs to be implemented. The multivariate extension of ANOVA is called MANOVA (multivariate ANOVA), but this method is unusable when the number of variables exceeds the number of experiments such as often encountered in omics (Stahle and Wold, 1990). Another method widely used to get a simplified view of multivariate datasets is Principal Component Analysis (PCA), but it has the drawback of not taking into account the structure coming from experimental designs (Jansen et al., 2005).

In this context, two methods developed around twenty years ago combine ANOVA and PCA. Those methods are called ANOVA-Simultaneous Component Analysis (ASCA) and ANOVA-PCA (APCA). In a first step, the ANOVA methodology is used to decompose a response matrix into effect matrices, according to the design of experiment, and in a second step, PCA is applied on each effect matrix to represent the information in a space of reduced dimensions. This approach leverages ANOVA to partition the overall response matrix into separate submatrices, each capturing the variation attributable to a specific factor or interaction from the experimental design, plus a residual matrix for unexplained variation. These submatrices can then be individually subjected to dimensionality reduction or other multivariate techniques to facilitate interpretation. ASCA was used for the first time by Smilde et al. (2005) to analyse proton nuclear magnetic resonance (^1H -NMR) spectra from urine samples of guinea pigs developing osteoarthritis. In this study, three groups of 6 guinea pigs were formed, and each group was treated with different

doses of vitamin C. Urine samples of the different guinea pigs were taken on 4 occasions over time (4, 7, 10 and 12 months). Even though not mentioned in these terms, this study was an example of longitudinal design where the groups of pigs were followed repeatedly over a period of time.

APCA was referenced for the first time in the article of Harrington et al. (2005) to analyse mass spectrometry spectra from amniotic fluid. In this study, a premature delivery amniotic fluid was compared to a control sample of pooled amniotic fluids from women with polyhydramnios (excess of amniotic fluid).

Even if ASCA and APCA have been initially implemented on omics data (Harrington et al., 2006; Jansen et al., 2005; Verouden et al., 2009; Cañellas et al., 2013), those two methods have also been applied on spectra from food products (Mazerolles et al., 2011; Sarembaud et al., 2007; Harnly et al., 2014; Climaco-Pinto et al., 2009; Climaco Pinto et al., 2008) and can be used to analyse all types of multivariate data resulting from an experimental design. The idea of partitioning a response matrix into submatrices through ANOVA decomposition has also been combined with other methods than PCA. For instance, Thissen et al. (2009) and El Ghaziri et al. (2015) combined the ANOVA decomposition with Partial Least Squares (ANOVA-PLS) and Boccard and Rudaz (2016) did it with Orthogonal Partial Least Squares (AMOPLS) in order to improve regression models and data interpretation. The combination of ANOVA submatrices with other methods such as Parallel Factor Analysis (PARAFAC) and Common Component and Specific Weight Analysis (CCSWA) can also be mentioned. The first one is named PARAFASCA (Jansen et al., 2008) and the second one ANOVA-ComDim (AComDim) (Jouan-Rimbaud Bouveresse et al., 2011).

In these methods, the matrix decomposition by ANOVA is a key step. However, this ANOVA step such as applied in those articles is limited to balanced designs. Indeed, if the design implies two factors or more and is unbalanced, the estimators of factor effects provided by the classical ANOVA decomposition can be biased. This essential observation motivated the development of this research which introduces two innovative methods, ASCA+ and APCA+ that allow, respectively, to extend the use of ASCA and APCA to unbalanced designs by leveraging least squares estimators from the General Linear Model (GLM), rather than simple differences of means. This ensures unbiased effect estimation, even in complex, real-world scenarios where data imbalances are common due to missing observations or variable sample sizes. Our contributions include: (1) a GLM-based decomposition framework for unbiased variance partitioning; (2) enhanced estimation of factor contributions using Type III sum of squares; and (3) integration of permutation tests for robust significance assessment as explained in our article Thiel, Féraud, and Govaerts (2017). These advancements not only overcome the biases in traditional ASCA/APCA

but also enable seamless combination with methods like PLS, OPLS, PARAFAC, and CCSWA.

Beyond academia, ASCA+ also holds significant industrial impact. In pharmaceutical development or food production, where experimental designs often face imbalances (e.g., due to constraints or equipment failures), ASCA+ can facilitate precise biomarker identification, reducing costs and accelerating product development.

This chapter will first review the ANOVA methods, the General Linear Model (GLM), ASCA and APCA, and demonstrate why effect estimators are biased when the response matrix decomposition is based on classical differences of means. Their extensions ASCA+ and APCA+ are then detailed, highlighting the innovative approach to decompose the response matrix by GLM and an appropriate factor coding. Related problems are also discussed such as the decomposition of the total information in percentage of variation explained by each model effect and the analysis of effects significance through permutation tests. The advantages of using ASCA+ and APCA+ will be shown first on a simple numerical example and then on a ^1H -NMR data set in metabolomics. This chapter and Chapter 2 should be seen as a uniform block. This chapter shows the theory underlying the ASCA+ method and implements it in a first R version. Chapter 2 consolidates the codes into a stable and complete R package. The notations in those two chapters will be the same and detailed in Section 6.2.

1.2 Methods

This section presents first the concepts of ANOVA used in ASCA and APCA and shows precisely the limitations of the classical approach in unbalanced designs. Basics of ASCA and APCA are then reviewed, followed by their adaptation to ASCA+ and APCA+.

1.2.1 ANOVA and the General Linear Model GLM

ANOVA is a statistical method to study the behaviour of a continuous variable depending on one or more independent categorical variables called *factors*, whose possible values are called *levels*. ANOVA stands for ANalysis Of VAriance and is a special case of the General Linear Model (GLM). Multifactorial ANOVA studies the effect of two or more factors, while ANOVA 1 studies the effect of a single factor. ANOVA allows among others to test whether the response variable is significantly different for the levels of each factor. This chapter will discuss it in general terms with illustration to the case of ANOVA 2 for crossed designs. Detailed presentation of ANOVA and GLM can be found in the book *Applied Linear Statistical Models* (Neter et al., 2004).

ANOVA2 decomposition in crossed design

The ANOVA 2 model for a crossed design with a levels for factor A, b levels for factor B and n observations per level is written as:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \varepsilon_{ijk}; \quad (1.1)$$

$$i = 1, \dots, a; j = 1, \dots, b; k = 1, \dots, n$$

where:

y_{ijk} is the observed response for replicate k of level i of factor A, and level j of factor B,

μ is the global response mean,

α_i is the effect of level i of factor A,

β_j is the effect of level j of factor B,

$\alpha\beta_{ij}$ is the interaction term of level i of factor A and level j of factor B and

ε_{ijk} is the error term usually supposed to be $N(0, \sigma^2)$ and independent to each other.

It can be noted that the design is balanced if the number of observations n stays the same for the different factor levels i and j .

This statistical model is over-parametrized and constraints must be set up on its parameters in order to get a unique solution in their estimation: only $(a - 1)$ parameters α_i , $(b - 1)$ parameters β_j and $(a - 1)(b - 1)$ interaction terms $(\alpha\beta)_{ij}$ need indeed to be estimated. Their least squares estimators when the design is balanced and constraints of Equation 1.1 are given in Table 1.1.

Table 1.1: Constraints and estimators of the terms of an ANOVA 2 equation with a balanced design.

Parameters	Constraints	Estimators
μ	–	$\hat{\mu} = \bar{y}_{...}$
α_i	$\sum_{i=1}^a \alpha_i = 0$	$\hat{\alpha}_i = \bar{y}_{i..} - \bar{y}_{...}$
β_j	$\sum_{j=1}^b \beta_j = 0$	$\hat{\beta}_j = \bar{y}_{.j.} - \bar{y}_{...}$
$\alpha\beta_{ij}$	$\sum_{j=1}^b \alpha\beta_{ij} = 0 \forall i$ and $\sum_{i=1}^a \alpha\beta_{ij} = 0 \forall j$	$\hat{\alpha}\beta_{ij} = \bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...}$
ε_{ijk}	–	$\hat{\varepsilon}_{ijk} = y_{ijk} - \bar{y}_{ij.}$

where:

$$\bar{y}_{...} = \frac{1}{abn} \sum_{ijk} y_{ijk}, \bar{y}_{i..} = \frac{1}{bn} \sum_{jk} y_{ijk}, \bar{y}_{.j.} = \frac{1}{an} \sum_{ik} y_{ijk} \text{ and } \bar{y}_{ij.} = \frac{1}{n} \sum_k y_{ijk}$$

This chapter is motivated by the fact that those estimators are only sustainable in balanced cases. In unbalanced cases, they can be biased and it is therefore better to estimate them with an alternative approach like the General Linear Model (GLM) and to use the matrix notation, as shown below. Moreover, formulae of Table 1.1 are specific to ANOVA 2 and may become very complex in multifactorial studies while GLM provides a general framework to express the estimators for any fixed effect ANOVA models.

Fundamental ANOVA decomposition equation provides also an interesting alternative to measure the percentage of the total variance of the response explained by each model term. For example, for the (balanced) ANOVA 2 model, it is written as follows:

$$\begin{aligned}
SST &= SSA + SSB + SSAB + SSE \\
&= \sum_{ijk} (y_{ijk} - \bar{y}_{...})^2 \\
&= \sum_{ijk} (\bar{y}_{i..} - \bar{y}_{...})^2 + \sum_{ijk} (\bar{y}_{.j.} - \bar{y}_{...})^2 + \sum_{ijk} (\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2 \quad (1.2) \\
&+ \sum_{ijk} (y_{ijk} - \bar{y}_{ij.})^2
\end{aligned}$$

The percentage of variation explained by an effect f of the model can then be calculated as follows:

$$\%Var_f = \frac{SSf}{SST} \quad (1.3)$$

The principle behind ASCA, ANOVA-PCA and related methods is then to decompose the $(N \times 1)$ vector $\mathbf{y}$ with all responses (containing all y_{ijk}) in a sum of vectors $\hat{\mathbf{m}}_f$ of size $N \times 1$ corresponding to model effects f . In crossed ANOVA 2, $N = a \times b \times n$ and the decomposition is written as follows:

$$\mathbf{y} = \hat{\mathbf{m}}_0 + \hat{\mathbf{m}}_A + \hat{\mathbf{m}}_B + \hat{\mathbf{m}}_{AB} + \hat{\mathbf{e}} \quad (1.4)$$

Each element m_{fijk} of an effect vector $\hat{\mathbf{m}}_f$ is defined as the estimation of a model parameter corresponding to the observation ijk and effect f . For instance, $\hat{\mathbf{m}}_0$ is a vector of $\bar{y}...$'s, $\hat{\mathbf{m}}_A$ a vector with $\hat{\alpha}_i$'s...

The fundamental equation of ANOVA becomes then in vectorial notations:

$$(\mathbf{y} - \hat{\mathbf{m}}_0)'(\mathbf{y} - \hat{\mathbf{m}}_0) = \hat{\mathbf{m}}_A' \hat{\mathbf{m}}_A + \hat{\mathbf{m}}_B' \hat{\mathbf{m}}_B + \hat{\mathbf{m}}_{AB}' \hat{\mathbf{m}}_{AB} + \hat{\mathbf{e}}' \hat{\mathbf{e}} \quad (1.5)$$

where $\hat{\mathbf{m}}_A'$ is the transposed vector of $\hat{\mathbf{m}}_A$. In most multifactorial experiments this equation is only verified for balanced designs while Equation 1.4 will remain valid in all cases after slight transformations of Table 1.1 formulae.

GLM formulation of ANOVA models

The GLM methodology (Neter et al., 2004) allows to rewrite ANOVA models in a matrix form as a multiple linear regression. This formalization is necessary to generalize the ANOVA model parameters estimation (detailed in Table 1.1 for ANOVA 2) to unbalanced designs. It can be written as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\varepsilon} \quad (1.6)$$

where $\mathbf{y}$ is the response vector of dimension $N \times 1$, $\mathbf{X}$ is the model matrix of dimension $N \times p$, $\boldsymbol{\theta}$ is the parameter vector of dimension $p \times 1$ and $\boldsymbol{\varepsilon}$ is the error vector of dimension $N \times 1$. N is the total number of observations and p is the number of parameters into the model.

Defining the model matrix $\mathbf{X}$ is a key step to estimate the parameters of the vector $\boldsymbol{\theta}$. In every model matrices with categorical factors, a factor with b levels is coded with $b - 1$ columns since the parameter of the last level can be calculated from the parameters of the other levels. The constraints used for parameter estimations have already been explained on Table 1.1. Columns for interaction effects are simply the multiplication of main effect columns. Several

options are proposed in the literature to express the ANOVA model factor effects in the model matrix $\mathbf{X}$ and define the form of the corresponding vector of parameters. They provide different solutions to take into account the constraints among model parameters (see Table 1.1, for ANOVA 2) and guarantee a full rank for the model matrix $\mathbf{X}$. Different coding are possible such as *Helmert-transformation*, *sum coding* or *treatment coding* which are explained in the book *Introduction to General and Generalized Linear Models* (Madsen and Thyregod, 2010). The most suitable approach, in the context of this chapter, is the coding commonly used in industrial experimental designs called *sum coding* or *deviation coding* (Madsen and Thyregod, 2010). The main effect of a factor with a levels is coded in $\mathbf{X}$ by $a - 1$ columns with 1 and 0 for the $(a - 1)$ first levels, and -1 for the last level. An interaction between two factors with a and b levels is coded with $(a - 1) \times (b - 1)$ columns calculated from the combination of all columns of corresponding main effects. The vector of parameters $\boldsymbol{\theta}$ contains then the list of all parameters except those corresponding to the last levels of the factors which can be calculated from the others. This coding has the advantage that each level of a parameter can be compared to other levels, which is not the case of other codings. Indeed, with treatment coding, each level is compared to a reference level, and with Helmert-transformation, each level is compared with the mean of subsequent levels.

For example, for an ANOVA 2 model with $a = 2$, $b = 3$ and $n = 1$, matrices $\mathbf{y}$, $\mathbf{X}$, $\boldsymbol{\theta}$, $\boldsymbol{\varepsilon}$ are given by:

$$\mathbf{y} = \begin{pmatrix} y_{111} \\ y_{121} \\ y_{131} \\ y_{211} \\ y_{221} \\ y_{231} \end{pmatrix}, \mathbf{X} = \begin{pmatrix} \mathbf{X}_0 & \mathbf{X}_A & \mathbf{X}_B & \mathbf{X}_{AB} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 \end{pmatrix}, \boldsymbol{\theta} = \begin{pmatrix} \mu_{..} \\ \alpha_1 \\ \beta_1 \\ \beta_2 \\ \alpha\beta_{11} \\ \alpha\beta_{12} \end{pmatrix},$$

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{111} \\ \varepsilon_{121} \\ \varepsilon_{131} \\ \varepsilon_{211} \\ \varepsilon_{221} \\ \varepsilon_{231} \end{pmatrix} \quad (1.7)$$

Note that, in the $\mathbf{X}$ matrix, each of the four blocks $\mathbf{X}_0$, $\mathbf{X}_A$, $\mathbf{X}_B$ and $\mathbf{X}_{AB}$ is orthogonal with respect to each other. This property is true in balanced designs but not in unbalanced ones. Note also that the use of this coding approach is the key to obtain results in this chapter and can not be replaced by alternative coding solutions usually used in linear models like *treatment coding*.

The regression parameters can be estimated with the ordinary least squares method:

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \quad (1.8)$$

In ANOVA 2, when the design is balanced, this vector $\hat{\boldsymbol{\theta}}$ containing the estimators of unconstrained parameters provides identical values to those given in Table 1.1. Estimators of restricted parameters are obtained from $\hat{\boldsymbol{\theta}}$ as follows:

$$\hat{\alpha}_a = -\sum_{i=1}^{a-1} \hat{\alpha}_i, \hat{\beta}_b = -\sum_{j=1}^{b-1} \hat{\beta}_j, \hat{\alpha}\hat{\beta}_{aj} = -\sum_{i=1}^{a-1} \hat{\alpha}\hat{\beta}_{ij} \text{ and } \hat{\alpha}\hat{\beta}_{ib} = -\sum_{j=1}^{b-1} \hat{\alpha}\hat{\beta}_{ij}$$

Residuals are calculated as vector $\hat{\mathbf{e}}$ of dimension $N \times 1$ where $e_i = y_i - \hat{y}_i$:

$$\hat{\mathbf{e}} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}} \quad (1.9)$$

These least square estimators are unbiased estimators of the unknown ANOVA model parameters, even when the design is unbalanced (unlike the simple differences of means, such as those given in Table 1.1, which provide biased estimators in unbalanced designs). Moreover, GLM approach provides a general and automatic framework to express and estimate models for complicated factorial experiments without having to develop specific formula for each design. Finally, GLM methodology proposes also different approaches to solve the problem of quantifying the amount of information explained by each effect of the model when the fundamental ANOVA equation is not anymore valid in unbalanced designs. For a given effect f of the model, the sum of square SS_f can be re-expressed as a Type III sum of squares, calculated as the difference between the residual sum of squares of the model with and without effect f . This approach is equivalent to classical SS_f when the design is balanced but the last one is more informative when the design is not balanced and allows hypothesis testing by the principle of F tests for nested models.

Advantage of GLM on classical ANOVA: example

This section clearly illustrates the advantage of using GLM decomposition compared to classical ANOVA decomposition in unbalanced designs. Let's take first a two factors balanced design with $a = 2$, $b = 3$ and $n = 2$ and the following known values for the ANOVA 2 model parameters vectors: $\mu = 10$, $\alpha' = (\alpha_1, \alpha_2) = (2, -2)$, $\beta' = (\beta_1, \beta_2, \beta_3) = (1, 0.5, -1.5)$, $\alpha\beta = \mathbf{0}$ and $\sigma^2 = 0$.

From these parameters which suppose that there is no error in the model (i.e. $\epsilon_{ijk} = 0$), the vector of responses calculated with model Equation 1.1 is the following:

$$\begin{aligned} \mathbf{y}' &= (y_{111}, y_{121}, y_{131}, y_{211}, y_{221}, y_{231}, y_{112}, y_{122}, y_{132}, y_{212}, y_{222}, y_{232}) \\ &= (13.0, 12.5, 10.5, 9.0, 8.5, 6.5, 13.0, 12.5, 10.5, 9.0, 8.5, 6.5) \end{aligned} \quad (1.10)$$

This response vector can then be used to re-estimate the model parameters α_i , β_j and $\alpha\beta_{ij}$ by the equations of Table 1.1 or by the least square matrix formula (see Equation 1.8). In the GLM approach, $\mathbf{X}$ is built by stacking twice vertically the model matrix given in Equation 1.7 in order to take into account that, in this example, the number of replicates $n=2$. Vector θ remains the same. The application of these formulae recovers, as expected, the true parameters by the two approaches.

Let's now suppose that trial 11 and 12 of the experiment are missing and that the vector of response becomes:

$$\begin{aligned} \mathbf{y}' &= (y_{111}, y_{121}, y_{131}, y_{211}, y_{221}, y_{231}, y_{112}, y_{122}, y_{132}, y_{212}) \\ &= (13.0, 12.5, 10.5, 9.0, 8.5, 6.5, 13.0, 12.5, 10.5, 9.0) \end{aligned} \quad (1.11)$$

The application of the two estimation methods to this reduced and unbalanced data set provides now different solutions. Through the simple calculation of mean responses differences, the estimators obtained are the following:

$$\begin{aligned} \hat{\mu} &= 10.45, \hat{\alpha}' = (\hat{\alpha}_1, \hat{\alpha}_2) = (1.5, -2.25), \hat{\beta}' = (\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3) = (0.5, 0.667, -1.333), \\ \widehat{\alpha\beta}' &= (\widehat{\alpha\beta}_{11}, \widehat{\alpha\beta}_{12}, \widehat{\alpha\beta}_{13}, \widehat{\alpha\beta}_{21}, \widehat{\alpha\beta}_{22}, \widehat{\alpha\beta}_{23}) \\ &= (0.5, -0.167, -1.67, 0.25, -0.4167, -0.4167) \end{aligned}$$

The estimators of the main effects are biased and an interaction effect appears while it was absent in the true model. The application of the GLM estimators to this unbalanced design provides the original values of the parameters as expected from their unbiased property. Beyond this exemple, simple means in balanced

designs provide unbiased estimates because each subgroup contributes equally, ensuring orthogonal decomposition without confounding. However, in unbalanced designs, simple means become biased as they implicitly weight subgroups based on their actual sample sizes, leading to over- or under-emphasis on certain levels and confounding effects from other factors. In contrast, the GLM approach yields unbiased least squares estimates by adjusting for all factors simultaneously and enforcing equal theoretical weighting across subgroups. Note that it is not straightforward to anticipate intuitively the bias on parameter estimation due to unbalancing. It can even happen that an unbalanced design does not create any bias. The bias depends on the experimental design and can be estimated with matrix calculation but no general formulae are available.

1.2.2 ASCA and APCA

ASCA and APCA decomposition

ASCA and APCA are extensions of ANOVA to multivariate cases. In ANOVA, a vector $\mathbf{y}$ was decomposed into vectors of parameters as shown in Equation 1.1 for ANOVA 2. In ASCA and APCA, a matrix $\mathbf{Y}$ of dimension $(N \times m)$ is decomposed in matrices of effects which have the same dimensions $(N \times m)$. Actually, the ANOVA decomposition is first applied to each column of the matrix $\mathbf{Y}$. The least squares estimators are then collected column by column in matrices of effects given, for the ANOVA 2 model, by the equation:

$$\mathbf{Y} = \widehat{\mathbf{M}}_0 + \widehat{\mathbf{M}}_A + \widehat{\mathbf{M}}_B + \widehat{\mathbf{M}}_{AB} + \widehat{\mathbf{E}} \quad (1.12)$$

Matrix $\widehat{\mathbf{M}}_0$ is the estimated main effect matrix and contains m columns of global means $\bar{y}_{...l}$ for each response variable $l = 1, \dots, m$; matrices $\widehat{\mathbf{M}}_A$ and $\widehat{\mathbf{M}}_B$ are main factor effect matrices and contain arithmetic means $\bar{y}_{i..l}$ and $\bar{y}_{.j.l}$; $\widehat{\mathbf{M}}_{AB}$ is the interaction matrix and contains the $\bar{y}_{ij.l}$'s and $\widehat{\mathbf{E}}$ contains residuals $y_{ijkl} - \bar{y}_{ij.l}$ of the model.

In ASCA, PCA is then applied on the different effect matrices:

$$\mathbf{Y} = \widehat{\mathbf{M}}_0 + \mathbf{T}_A \mathbf{P}'_A + \mathbf{T}_B \mathbf{P}'_B + \mathbf{T}_{AB} \mathbf{P}'_{AB} + \widehat{\mathbf{E}} \quad (1.13)$$

where $\mathbf{T}_f$ and $\mathbf{P}_f$ correspond to score and loading matrices of the effect $f \in \{A, B, AB\}$. Each (multivariate) model effect is then interpreted through score and loading plots.

In APCA, PCA is not directly applied on the effect matrix (e.g. $\widehat{\mathbf{M}}_{\mathbf{A}}$) but on the effect matrix added to the error matrix (e.g. $\widehat{\mathbf{M}}_{\mathbf{A}} + \widehat{\mathbf{E}}$).

Score plots obtained in APCA show then the groups of observations for the different levels of the variables, whereas ASCA shows only means of the levels (Zwanenburg et al., 2011).

A variant of ASCA allows to represent the variability between the observations in the score plot. Loadings $\mathbf{P}_f$ are first calculated on the effect matrix $\widehat{\mathbf{M}}_f$ only, like in ASCA, but score plots are calculated using the residual matrix as follows (Zwanenburg et al., 2011):

$$\mathbf{T}_f^{\mathbf{E}} = (\widehat{\mathbf{M}}_f + \widehat{\mathbf{E}})\mathbf{P}_f = \widehat{\mathbf{M}}_f\mathbf{P}_f + \widehat{\mathbf{E}}\mathbf{P}_f \quad (1.14)$$

where $f \in \{A, B, AB\}$.

The projection $\mathbf{T}_f^{\mathbf{E}}$ describes the variation among factor levels in terms of the loadings $\mathbf{P}_f$ of the PCA model for $\mathbf{M}_f$. Contrary to the variation in APCA, the variation in $\mathbf{T}_f^{\mathbf{E}}$ is not maximized. These adapted ASCA score plots will be called ASCA-E score plots in the next sections.

Both APCA and ASCA-E offer the possibility to visualize the scores taking into account the residual variability of the model. However in practice, the recommendation to the user would be to choose ASCA-E when the residual variability $\mathbf{E}$ is large or might still contain hidden effects. Indeed in ASCA-E, the variation in $\mathbf{T}_f^{\mathbf{E}}$ is not maximized which helps to isolate the effects of true factor in noisy scenarios.

Percentages of variation

ASCA literature proposes also a generalization of the fundamental ANOVA Equation 1.2 to quantify the percentage of variation explained by each effect in the multivariate case. Vis et al. (2007) proposes to quantify the sums of squares of the matrices of effects with the square of the Frobenius norm defined as follows for a matrix $\mathbf{Y}$:

$$\|\mathbf{Y}\|^2 = \sum_{i=1}^N \sum_{j=1}^M |y_{ij}|^2 \quad (1.15)$$

In balanced situation, Equation 1.2 can be generalized as:

$$\|\mathbf{Y}\|^2 = \|\mathbf{M}_0\|^2 + \|\mathbf{M}_A\|^2 + \|\mathbf{M}_B\|^2 + \|\mathbf{M}_{AB}\|^2 + \|\mathbf{E}\|^2 \quad (1.16)$$

The percentage of variation of $\mathbf{Y}$ explained by each model effect is then obtained by:

$$\%Var_f = \frac{\|\mathbf{M}_f\|^2}{\|\mathbf{Y}\|^2 - \|\mathbf{M}_0\|^2} \times 100 \quad (1.17)$$

where $f \in \{A, B, AB, Error\}$.

With unbalanced designs, matrices $\mathbf{M}_A$, $\mathbf{M}_B$ and $\mathbf{M}_{AB}$ are not necessarily orthogonal and

$$\|\mathbf{Y}\|^2 \neq \|\mathbf{M}_0\|^2 + \|\mathbf{M}_A\|^2 + \|\mathbf{M}_B\|^2 + \|\mathbf{M}_{AB}\|^2 + \|\mathbf{E}\|^2 \quad (1.18)$$

and the sum of variation percentages of the different factors is, in most cases, not equal to 100%.

Permutation tests

It is possible to determine which terms of the ANOVA model have a globally significant effect (i.e. on all M variables) by computing p-values using permutation tests. The method explained in this section is based on the article of Zwanenburg et al. (2011).

It consists in generating randomly a number N_u of permutations (e.g. 1000) on the rows of the matrix $\mathbf{Y}$. Then, for the initial and all the permuted data matrices, the statistic $SS_f^u(\mathbf{T}_g)$ is calculated where f is the model effect of interest (e.g. A, B or AB in ANOVA 2), $SS_f^u(\mathbf{T}_g)$ is the square of the Frobenius norm of the scores matrix $\mathbf{T}_g$ for effect f corresponding to the first g principal components. Finally, the calculation of p-values associated to the effect f is calculated as follows:

$$p - value = \frac{\#\{SS_f^u(\mathbf{T}_g) \geq SS_f(\mathbf{T}_g), u = 1 \dots N_u\}}{N_u} \quad (1.19)$$

When the percentage of variance explained by the two first components of an effect matrix is high, taking $g = 2$ is recommended to provide a direct link between the p-values and the corresponding score plot.

This permutation method can be applied to balanced designs as well as unbalanced ones, but the type of decomposition of $\mathbf{Y}$ by ANOVA or GLM and the rows permutation method have to be adapted to the model of interest and to the experimental design (e.g. crossed or nested) (Anderson and Braak, 2003).

It can also be noted that setting up the permutation method for ANOVA models with more than 2 factors is not easy, as explained in Thiel (2015).

1.2.3 ASCA+ and APCA+ principles

ASCA+ and APCA+ principles

In the case of unbalanced designs, effects matrices estimated by classical ASCA and APCA can be biased when they are based on mean responses (see the example in Section 1.2.1). The solution proposed by ASCA+ and APCA+ is then simply to use the GLM decomposition instead of ANOVA decomposition to estimate the effect matrices $\mathbf{M}_0$, $\mathbf{M}_A$, $\mathbf{M}_B$, $\mathbf{M}_{AB}$ and $\mathbf{E}$ in order to get unbiased estimators even with unbalanced designs. In the case where the design is balanced, these estimators are identical to those given in ASCA, but if the design is unbalanced they have the critical advantage of being unbiased. Our novel ASCA+ and APCA+ methods overcome these limitations by providing unbiased estimators through GLM, making them essential for practitioners dealing with real-world unbalanced data such as industrial experiments where replicates vary.

Multivariate GLM and effect matrices estimation

The multivariate GLM model can be written as:

$$\mathbf{Y} = \mathbf{X}\Theta + \mathbf{E} \quad (1.20)$$

where $\mathbf{Y}$ is the response matrix of dimension $N \times m$, $\mathbf{X}$ is the model matrix of dimension $N \times p$, Θ is the parameter matrix of dimension $p \times m$ and $\mathbf{E}$ is the error matrix of dimension $N \times m$.

The model matrix $\mathbf{X}$ does not depend of m and corresponds exactly (according to the design) to the $\mathbf{X}$ matrix presented in the univariate case (see Equation 1.7 for the ANOVA 2 model). The other matrices contain m columns, one corresponding to each variable in the response matrix $\mathbf{Y}$.

Ordinary least square method provides a solution to obtain unbiased estimators of the model parameters:

$$\hat{\Theta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \quad (1.21)$$

The effect matrices for the different terms of the model can then be obtained quite simply by building for each effect f a new model matrix $\mathbf{X}^*$ obtained by keeping in $\mathbf{X}$ only the block $\mathbf{X}_f$ corresponding to the effect of interest, and

replacing all the other columns by zero's. The effect matrix $\mathbf{M}_f$ can then be estimated by multiplying this new $\mathbf{X}^*$ matrix by the matrix of estimated parameters. In other terms, the effect matrix is the product of $\mathbf{X}_f$ and $\hat{\Theta}_f$, the sub-blocks of matrix $\mathbf{X}$ and $\hat{\Theta}$ corresponding to effect f :

$$\hat{\mathbf{M}}_f = \begin{pmatrix} \mathbf{0} & \mathbf{X}_f & \mathbf{0} \end{pmatrix} \hat{\Theta} = \mathbf{X}_f \hat{\Theta}_f \quad (1.22)$$

For the ANOVA 2 model example presented in Equation 1.7, the estimation of effect matrix $\hat{\mathbf{M}}_B$ becomes :

$$\begin{aligned} \hat{\mathbf{M}}_B &= \begin{pmatrix} \mathbf{X}_0 & \mathbf{X}_A & \mathbf{X}_B & \mathbf{X}_{AB} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \end{pmatrix} & \begin{pmatrix} \hat{\mu}_1 & \hat{\mu}_2 & \cdots & \hat{\mu}_m \\ \hat{\alpha}_{11} & \hat{\alpha}_{12} & \cdots & \hat{\alpha}_{1m} \\ \hat{\beta}_{11} & \hat{\beta}_{12} & \cdots & \hat{\beta}_{1m} \\ \hat{\beta}_{21} & \hat{\beta}_{22} & \cdots & \hat{\beta}_{2m} \\ \hat{\alpha}\hat{\beta}_{111} & \hat{\alpha}\hat{\beta}_{112} & \cdots & \hat{\alpha}\hat{\beta}_{11m} \\ \hat{\alpha}\hat{\beta}_{121} & \hat{\alpha}\hat{\beta}_{122} & \cdots & \hat{\alpha}\hat{\beta}_{12m} \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{X}_B \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{pmatrix} \end{pmatrix} \begin{pmatrix} \hat{\beta}_{11} & \hat{\beta}_{12} & \cdots & \hat{\beta}_{1m} \\ \hat{\beta}_{21} & \hat{\beta}_{22} & \cdots & \hat{\beta}_{2m} \end{pmatrix} = \mathbf{X}_B \hat{\Theta}_B \end{aligned}$$

Matrix of errors $\hat{\mathbf{E}}$ is finally simply estimated as the residual matrix of the model:

$$\hat{\mathbf{E}} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}\hat{\Theta} \quad (1.23)$$

This procedure has the big advantage to provide a general solution to build the ASCA effect matrices for any fixed effects ANOVA model. The only need is a routine to build the model matrix $\mathbf{X}$ with the appropriate sum or deviation coding with respect to the model of interest. It offers also a possibility of extension to analysis of covariance models and mixed effect models.

Using other terms, if $\mathbf{L}$ is defined as a $p_f \times p$ contrast matrix used to select the lines of $\hat{\Theta}_f$ linked to effect f , then $\hat{\mathbf{M}}_f = \mathbf{X}\mathbf{L}'\mathbf{L}\hat{\Theta}$. The ASCA⁺ decomposition of the response matrix into effect matrices can also be noted as:

$$\mathbf{Y} = \hat{\mathbf{M}}_0 + \sum_{f=1}^F \hat{\mathbf{M}}_f + \hat{\mathbf{E}} \quad (1.24)$$

To elaborate a bit more on the content of the matrix $\mathbf{L}$, let's use the ANOVA 2 model example presented in Equation 1.7. Assuming a two-level factor B with sum contrasts, where parameters for the main effect of B are in the third and fourth rows of $\hat{\Theta}$, the contrast matrix $\mathbf{L}$ for effect B is:

$$\mathbf{L} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

This $\mathbf{L}$ selects only the rows of $\hat{\Theta}$ corresponding to the parameters of B's main effect. Applying it gives $\hat{\Theta}_B = \mathbf{L}\hat{\Theta}$, and the effect matrix estimation proceeds as shown in the provided example for $\hat{\mathbf{M}}_B = \mathbf{X}_B\hat{\Theta}_B$. In practice, $\mathbf{L}$ is automatically generated ensuring flexibility for any fixed-effects design.

Percentages of variation

In the line of the principle of type III sum of squares presented in Section 1.2.1, the following method is proposed to calculate the percentage of variation explained by an effect f of the ANOVA model:

$$\%Var_f = \frac{\|\hat{\mathbf{E}}_{/f}\|^2 - \|\hat{\mathbf{E}}_{\text{full}}\|^2}{\|\mathbf{Y} - \hat{\mathbf{M}}_0\|^2} \times 100 \quad (1.25)$$

where $\hat{\mathbf{E}}_{/f}$ is the residual matrix of a model estimated without taking into account the effect f in the model matrix and $\hat{\mathbf{E}}_{\text{full}}$ is the error matrix of a complete model. This formula is equivalent to Equation 1.17 for balanced designs.

The percentage of variance explained by the error term will simply be the following:

$$\%Var_e = \frac{\|\hat{\mathbf{E}}_{\text{full}}\|^2}{\|\mathbf{Y} - \hat{\mathbf{M}}_0\|^2} \times 100 \quad (1.26)$$

This approach gives an exact interpretation of the difference of sum of squares but will usually not provide percentages of variances for which the sum equals 100% when the design is not balanced. It can be disturbing for the user and then

to keep this property, an alternative formulation of this percentage of variance is the following:

$$\%Var_{f^*} = \frac{\|\hat{\mathbf{E}}_{/f}\|^2 - \|\hat{\mathbf{E}}_{full}\|^2}{\sum_{i \in F} \|\hat{\mathbf{E}}_{/i}\|^2 - (\#F - 1)\|\hat{\mathbf{E}}_{full}\|^2} \times 100 \quad (1.27)$$

where F is the list of effects of the ANOVA model and $\#F$ is the number of effects in the model.

This approach gives a sum of percentages always equal to 100% but the interpretation is not the exact difference of sum of squares. Note that in the case of balanced designs, the percentages given by both methods are the same. It is up to the user to choose which approach fits best its use.

Effects significance tests

In a multiple effect model, it is interesting to determine which ones have a significant effect on the responses. In classical ANOVA or GLM with a single response, this can be done by F or t tests. In ASCA and APCA, it is common to use permutation tests to generalize these tests to multiple responses (Zwanenburg et al., 2011). However, as discussed in Thiel, 2015, this method is particular to each model and difficult to implement, requiring careful adaptation of the row permutation strategy to preserve the experimental design's structure (e.g., crossed vs. nested factors). For multifactorial designs (> 2 factors), defining appropriate permutations becomes increasingly complex and prone to errors. Additionally, permutation tests can be computationally intensive for high-dimensional datasets, as they necessitate repeated full decompositions over thousands of iterations. A parametric bootstrap test can alternatively be implemented because it has the advantage to be independent from the model structure (compared to permutation tests) and can be adapted to further extensions. To test the significance of an effect f , the bootstrap procedure involves the following steps (Benaiche, 2021):

1. Define the model under H_0 , obtained by excluding the effect f from the model matrix $\mathbf{X}$: $\mathbf{Y} = \mathbf{X}_{/f}\boldsymbol{\Theta}_{/f} + \mathbf{E}_{/f}$.
2. Fit the reduced H_0 and full H_1 models to the response matrix $\mathbf{Y}$.
3. Calculate the test statistic:

$$F_{obs} = \frac{(\|\hat{\mathbf{E}}_{/f}\|^2 - \|\hat{\mathbf{E}}_{full}\|^2)/(p - p_{/f})}{\|\hat{\mathbf{E}}_{full}\|^2/(n - p)}$$

where $p_{/f}$ is the number of parameters of the partial model for one response.

4. Calculate the predicted values $\hat{\mathbf{Y}}_{/f} = \mathbf{X}_{/f} \hat{\boldsymbol{\Theta}}_{/f}$ and the residuals $\hat{\mathbf{E}}_{/f} = \mathbf{Y} - \hat{\mathbf{Y}}_{/f}$ of the model under H_0 to use them in the bootstrap loop.
5. Repeat for a large number B of iterations ($b = 1, \dots, B$):
 - (a) Sample $\hat{\mathbf{E}}_b$ with replacement among the rows of $\hat{\mathbf{E}}_{/f}$
 - (b) Generate a bootstrap response matrix $\mathbf{Y}_b = \hat{\mathbf{Y}}_{/f} + \hat{\mathbf{E}}_b$
 - (c) Estimate the parameters of the full model with this new response matrix $\mathbf{Y}_b$, $\hat{\boldsymbol{\Theta}}_b = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}_b$
 - (d) Calculate the corresponding test statistics F_b by using the optimized formula:

$$F_b = \frac{\text{tr}(\hat{\boldsymbol{\Theta}}_b' \mathbf{L}' (\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{L}')^{-1} \mathbf{L} \hat{\boldsymbol{\Theta}}_b) / (p - p_{/f})}{\|\hat{\mathbf{E}}_{full}\|^2 / (n - p)} \quad (1.28)$$

where $\mathbf{L}$ is, as above, the contrast matrix used to select in $\mathbf{X}$ and $\hat{\boldsymbol{\Theta}}_b$ the columns/lines linked to an effect f . This formula has the advantage of not requiring the estimation of the partial model.

6. Calculate the bootstrap p-value for the effect of interest:

$$p - \text{value}_f^{boot} = \frac{\sum_{b=1}^B I(F_b \geq F_{obs}) + 1}{B + 1}$$

This procedure is implemented in R package `limpca` presented in Chapter 2. In this implementation, the computation of the F_f statistics has been optimized by using the F_b (Equation 1.28) and by using parallel computation in R. In `limpca`, it is also possible to apply such tests on other restricted models than the removal of a single effect (referred below as tests on “combined” effects).

A summary of the different ASCA+ steps is illustrated in Figure 1.1. In a first step, the original response data matrix $\mathbf{Y}$ is decomposed into a series of effect matrices $\hat{\mathbf{M}}$ according to a general linear model (GLM) based on the experimental design. In a second step, the importance of the model effects can be evaluated by estimating the percentage of variance explained by each effect, as well as by testing their significance. In a third step, PCA is applied on the decomposed effect matrices to visualize the model results through informative loading and score plots.

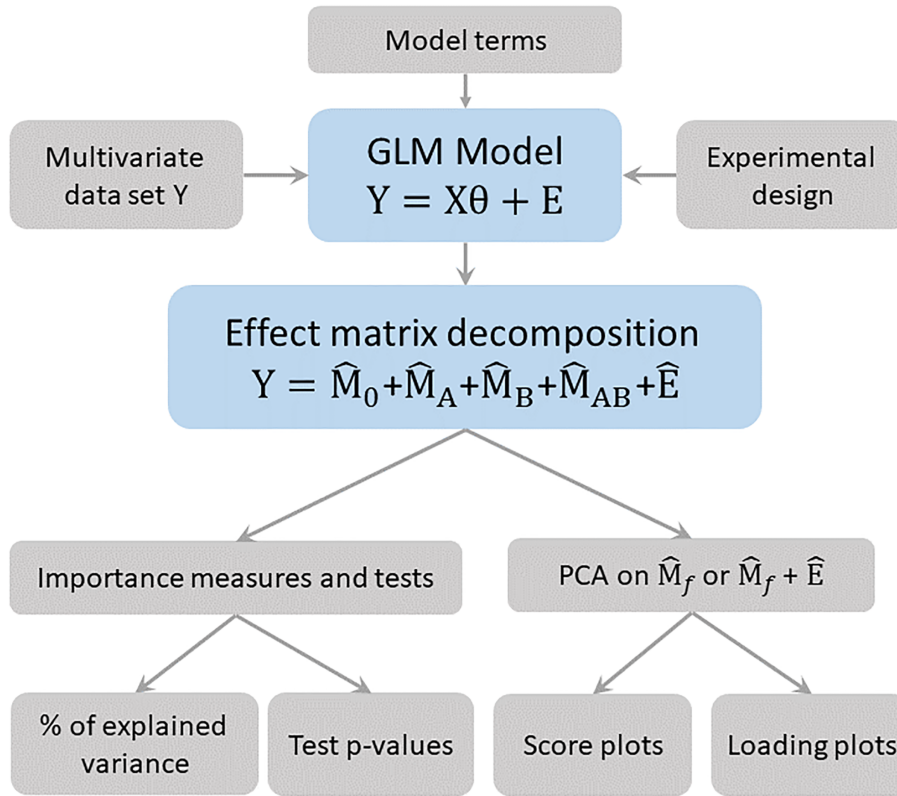


Figure 1.1: Overview of ASCA+ and APCA+ approaches based on (Guisset, Martin, and Govaerts, 2019).

1.3 Case study

In this section, ASCA and APCA methods and their extensions ASCA+ and APCA+ are applied on two datasets. The first dataset (Figure 1.2(a)) consists of 48 spectra derived from a pool of rat urine samples analysed by ^1H -NMR. Four factors of variability were considered in the experimental design:

1. the medium: two media are considered. A medium of a mixture of urine from different rats, and this same medium diluted by 50%.
2. the concentration of citrate in the medium: the two media were spiked with a chemical, citrate, in two concentrations. Concentrations were 0 or 4 mM (Qc).

3. the concentration of hippurate in the medium: the two media were spiked with another chemical, hippurate, also in two concentrations. Concentrations were 0 or 2 mM (Qh).
4. the day: each of the 8 possible mixtures ($2 \text{ media} \times 4 \text{ concentration combinations}$) were analysed twice on 3 different days.
5. the time: the analysis of the samples was done under two times: just after defreezing and one hour later in order to evaluate the stability of samples over time.

There are then $2 \text{ media} \times 4 \text{ concentration combinations} \times 3 \text{ days of measurement} \times 2 \text{ replicates per day} = 48 \text{ samples}$.

In Sections 1.3.1 and 1.3.2, to keep it simple, only the two factors hippurate and citrate are of interest for the matrix decomposition and the other factors are interpreted in the residual variability. The experimental design to consider is then a two factors balanced crossed design with two levels per factor ($a = 2$, $b = 2$). In Section 1.3.3, a complete model taking into account all factors will be used for the analysis of Dataset 1.

The second dataset (Figure 1.2(b)) is simply a subset of Dataset 1 which becomes unbalanced since the number of samples is not the same for the different conditions. Indeed, 6 samples were removed for the combinations in hippurate and citrate "0,0" and "Qh,Qc".

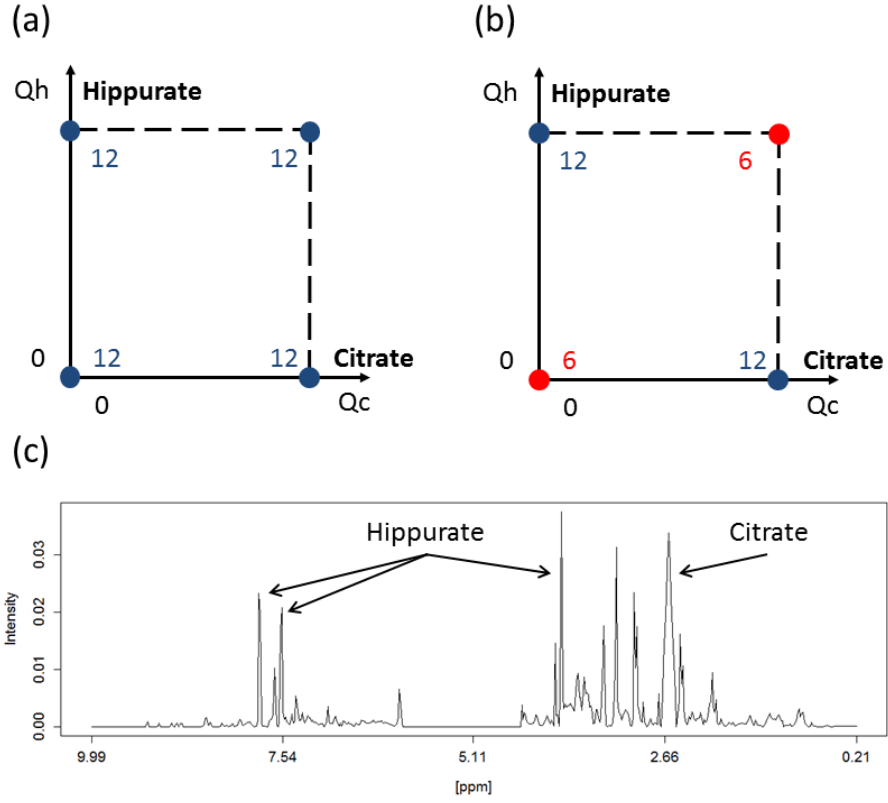


Figure 1.2: Combinations in hippurate and citrate used in the experimental design 1 (a) and 2 (b) and typical spiked urine spectrum (c)

For each dataset, the spectra are gathered in a spectral matrix of dimensions $(N \times M)$. The number of rows N of this matrix corresponds to the number of spectra, whereas the number of columns M of the matrix corresponds to the number of descriptors of each spectrum expressed in ppm (part per million). Each descriptor corresponds to the mean intensity of a spectral region and is considered as a variable in the multivariate dataset. For Dataset 1, the dimension of the matrix is (48×600) since there are 48 spectra with a resolution of 600 ppm. Among all pre-processing steps, it is important to note that the data were normalized to a constant sum M . Moreover, the citrate area, which is unstable, has been aggregated to a unique peak (Rousseau, 2011).

A typical resulting ^1H -NMR spectrum is illustrated in Figure 1.2(c). The descriptors corresponding to citrate and hippurate are highlighted. Hippurate is characterized by multiple descriptors whereas citrate is localized in a single ppm region.

1.3.1 Balanced Dataset 1 analysis

Figure 1.3 presents the score plots of the first two components in PCA, APCA, ASCA and ASCA-E for Dataset 1, while Figure 1.4 presents the related loading plots of the first principal component. With classical PCA, Figure 1.3(a) shows no complete separation of the four design combinations, although partial separation between some classes can be observed. The corresponding loading plot in Figure 1.4(a) shows the loading vectors of hippurate and citrate in the same component and in opposite directions.

Table 1.2 compares the percentages of the total variance explained by the model effects and corresponding p-values (derived from permutation tests), in ASCA and ASCA+, for the balanced and unbalanced design. For the balanced design, results in ASCA and ASCA+ are identical. Factors hippurate and citrate are significant and account respectively for 27.6% and 18.0% of the variability. The interaction effect is not significant and accounts for 0.5% of the variability. It can be noted that the biggest part of the variability in the model is due to the residual error. This large contribution might be due to factors not considered in the model decomposition like Medium and Day, or other sources of variability. For all cases those errors account for around 54% of the variation.

Figures 1.3(b), (c) and (d) show respectively the score plots in APCA for hippurate, citrate and interaction effects. For hippurate and citrate, one can clearly see two groups of points corresponding to the two levels of concentrations of those two chemicals. A clear separation of the groups suggests a strong effect of those factors. For the interaction effect, there is no clear separation between levels of hippurate and citrate, which is expected for this design because there is no significant interaction effect between hippurate and citrate. Indeed, there are four groups of points but the two levels of hippurate and citrate are present in each group. The separation of points in four groups is caused by the two other factors, the Medium with two levels and the factor Day with three levels. Factor Medium creates a first division of points: diluted samples are separated from undiluted ones. For the factor Day, each urine sample was analysed by ^1H -NMR at three different days: day 1, day 3 and day 5. Factor Day creates a second division of points: samples from day 1 are grouped among them and samples from day 3 and 5 are grouped.

Figures 1.3(e), (f) and (g) show respectively the score plots in ASCA for hippurate, citrate and interaction effects. They only show two points corresponding to level means. Indeed in ASCA, PCA is done on factor matrices without adding the model residual matrix. The residual variability is then not observed unlike in APCA. Note also that, for these three graphics, the first component explains 100% of the variability because there are only two levels per experimental factor corresponding to $a - 1$, $b - 1$ and $(a - 1)(b - 1) = 1$ degree of freedom for hippurate, citrate and interaction model effects, respectively.

Figures 1.3(i), (j) and (k) show respectively the ASCA-E score plots for hippurate, citrate and interaction effects. The interpretation of the plots is, as expected, similar to APCA: one can clearly see two groups of points for factors hippurate and citrate for main effect plots and no interaction effect due to hippurate and citrate in the interaction plot.

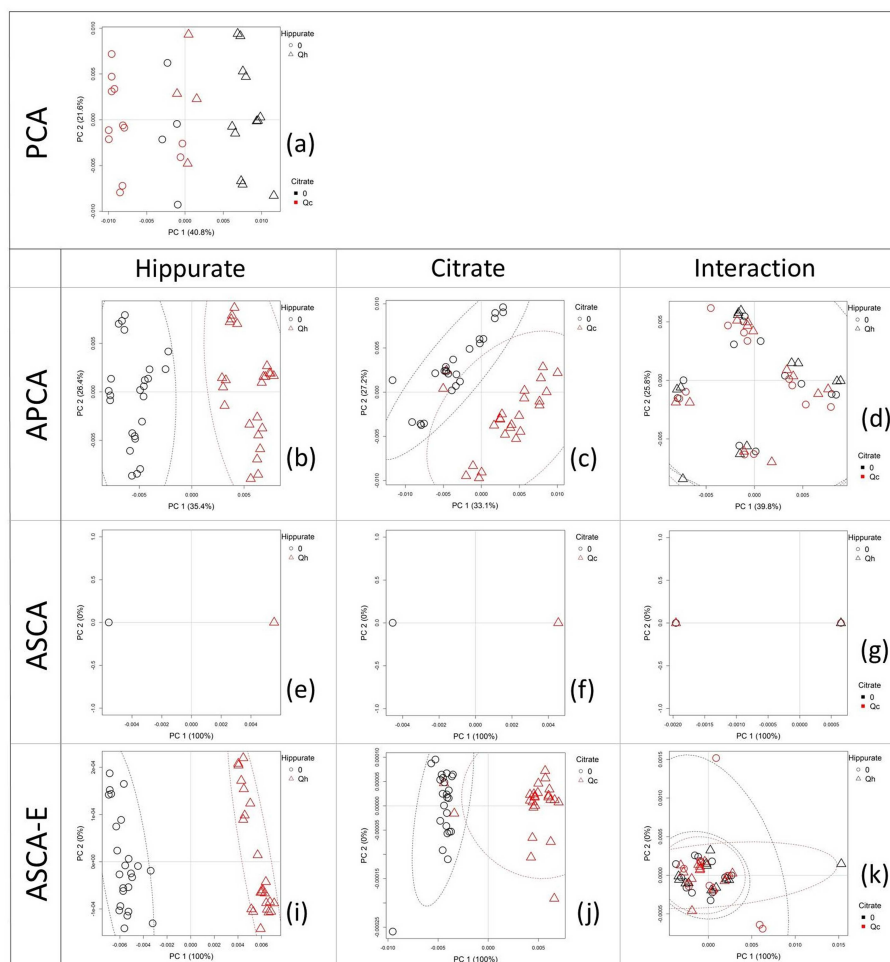


Figure 1.3: Score plots from classical PCA, APCA, ASCA and ASCA-E for hippurate, citrate and interaction effects on the balanced dataset

Figures 1.4(b), (c) and (d) show respectively the loading plots in APCA for hippurate, citrate and interaction effects while Figures 1.4(e), (f) and (g) show the corresponding loading plots in ASCA. Loading plots in ASCA-E are not showed since they are the same as in ASCA. Figures 1.4(b) and (e) correspond, as expected, to the profile of hippurate spectra. In Figures 1.4(c) and (f), the citrate peak appears clearly but is partially hidden by noise in APCA, which is due to the addition of the residuals matrix to the effect matrix in this approach. On Figure 1.4 (c) and (d), one can observe a residual peak around 5.1 ppm corresponding to water. ASCA and APCA methods can then help to discriminate

biomarkers corresponding to a factor of interest which is not the case for PCA where both peaks of hippurate and citrate were superimposed. In Figures 1.4(d) and (g), noise is prominent suggesting a low effect of the interaction model term. This is an expected consequence of the nature of the experimental design and confirmed by Table 1.2 showing the significance of hippurate and citrate and absence of interaction effect.

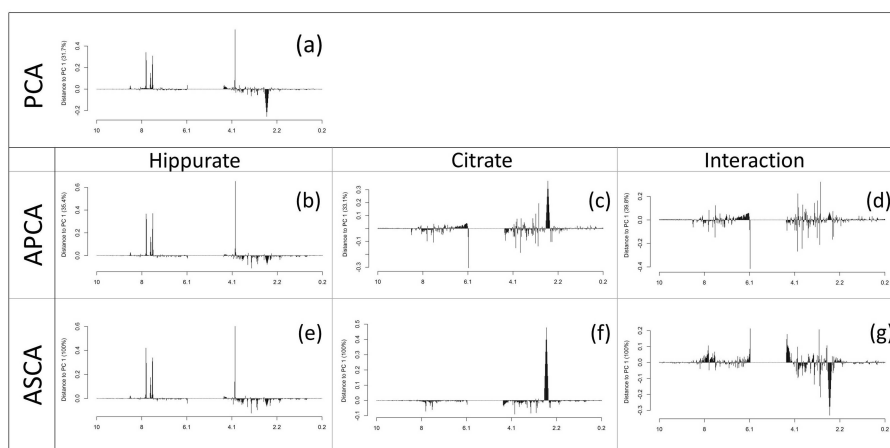


Figure 1.4: First principal component loading plots in PCA, APCA and ASCA for hippurate, citrate and interaction effects on the balanced dataset

Table 1.2: Percentages of variation and p-values calculated in ASCA and ASCA+ for the balanced and unbalanced design.

	Balanced design		Unbalanced design			
	ASCA and ASCA+		ASCA		ASCA+	
Factor	%	p-value	%	p-value	%	p-value
Hippurate	27.6	< 0.001	32.7	< 0.001	18.1	< 0.001
Citrate	18.0	< 0.001	26.8	< 0.001	12.1	0.004
Interaction	0.5	0.984	12.2	0.016	1.0	0.950
Error	53.9	—	54.1	—	53.9	—

1.3.2 Unbalanced Dataset 2 analysis

Table 1.2 shows different results among ASCA and ASCA+ for the unbalanced design. Factors hippurate and citrate are significant for both methods. However the interaction effect is wrongly significant in ASCA and not with ASCA+. In ASCA the interaction effect accounts for 12.2% of the variability, and in ASCA+ it accounts only for 1.0% of the variability. Those results confirm the observations in Figure 1.5 and 1.6. It is interesting to look more in detail at the sums of variation percentages which equal 125.8% in ASCA and 85.1% in ASCA+. This difference is due to the estimation of the model effects since the error estimation stays around 54%. In classical ASCA, effect matrices are estimated using simple differences of group means. In unbalanced designs, unequal subgroup sizes create non-orthogonality and bias which lead to overestimated percentages, especially the interaction effect. In ASCA+, the GLM approach provides unbiased estimates via ordinary least squares. However, in unbalanced designs, the effect matrices $\mathbf{M}_f$ are non-orthogonal and the sum of percentages explained by each effect are underestimated. The sum of the individual effect percentages (based on the squared Frobenius norms of the non-orthogonal effect matrices) plus the residual percentage can be less than or greater than 100%, depending on the data. In practice, when percentages do not sum to 100%, it should alert users to investigate dataset balance and prefer ASCA+ for unbiased interpretation.

Figure 1.5 compares the score plots for hippurate, citrate and interaction effects on the unbalanced Dataset 2 obtained by classical ASCA, ASCA+ and by ASCA and ASCA+ with added errors. For hippurate and citrate effects, ASCA and ASCA+ give similar results with two groups of points corresponding to the factor level means. The two groups are completely separated by the first component explaining 100% of the information. For the interaction effect, results are different. In ASCA, four groups are observed, corresponding to the different combinations of hippurate and citrate. In ASCA+ there are only two groups separated by the first component. This difference is explained because in ASCA+, the estimation is unbiased and the matrix of interaction has only $(a - 1)(b - 1) = 1$ degrees of freedom. In ASCA, the estimation is biased and degrees of freedom are artificially created. When adding errors in the ASCA score plots, the group separation for hippurate and citrate effects is also very close in ASCA and ASCA+. However, in the interaction plot of ASCA, the 4 combinations of hippurate and citrate appear due to the bias and not in ASCA+ where there is, as expected, a random distribution of points since there is no interaction effect in the data.

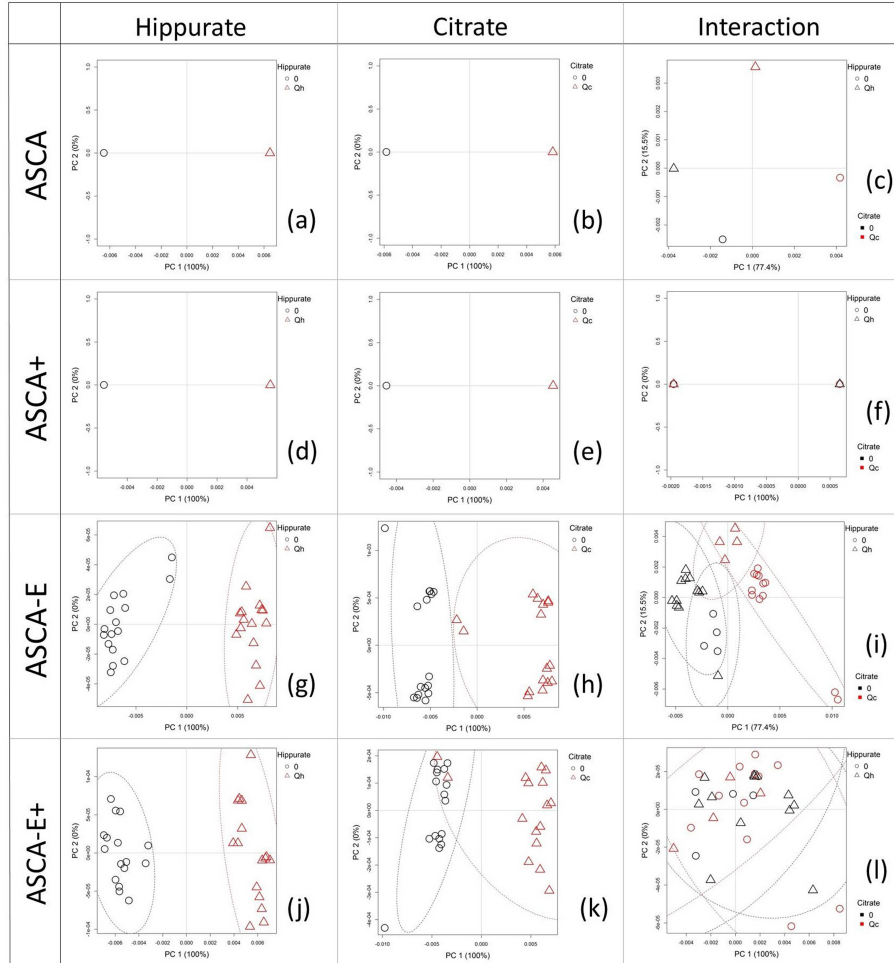


Figure 1.5: Score plots in ASCA and ASCA+ for hippurate, citrate and interaction effects on the unbalanced dataset

Figure 1.6 presents the first principal component vector of loadings provided by ASCA and ASCA+ for hippurate, citrate and interaction effects for the unbalanced Dataset 2. One can see how those two methods highlight biomarkers linked to hippurate, citrate and interaction differently. In ASCA, loadings of the three model effects include opposed peaks of hippurate, citrate which is not expected and does not correspond to what was observed in Figure 1.4 for the balanced design. These poor results are due to the bias in the estimation of the effect matrix as explained in Section 1.2.1. In ASCA+, effect estimations are unbiased and the hippurate and citrate loading plots reflect

perfectly the profile of hippurate and citrate spectra and the interaction loading plot contains only noise since there is no interaction between the two components induced by the simultaneous spiking. ASCA+ allows then to discriminate biomarkers linked to each factor of interest even for unbalanced designs.

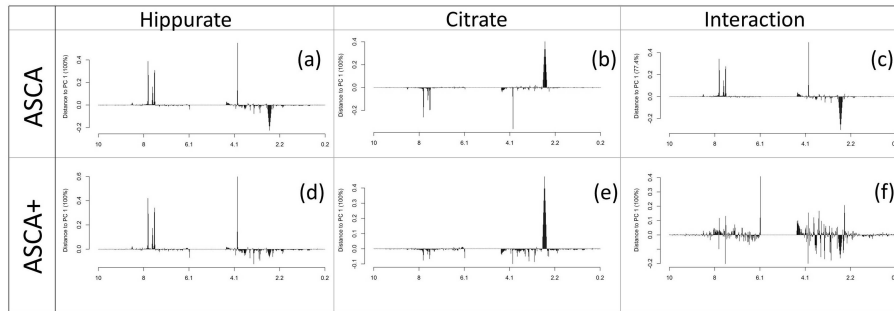


Figure 1.6: Loading plots in ASCA and ASCA+ for hippurate, citrate and interaction effects on the unbalanced dataset

1.3.3 Balanced Dataset 1 analysis with all factors

The second model applied on the balanced Dataset 1 is a full two-way ANOVA containing the main effects of the five factors and all two-way interactions between them. Theoretically, the model could estimate more complex interactions but to simplify interpretation they have been limited to two-ways ones. The R formula can be written as follows:

```
outcomes ~ Hippurate + Citrate + Medium + Time + Day +
  Hippurate:Citrate + Hippurate:Medium + Hippurate:Time +
  Hippurate:Day + Citrate:Medium + Citrate:Time +
  Citrate:Day + Medium:Time + Medium:Day + Time:Day
```

Table 1.3 presents the percentages of variance and the corresponding bootstraps p-values, obtained from 1000 resampled outcomes samples with the package `limpca`. It indicates that all main effects of Hippurate, Citrate, Medium, Time and Day are significant. Considering the interactions, only Medium:Time and Medium:Day are significant.

Those results can be compared to the ones in Table 1.2. The percentages of variation due to Hippurate, Citrate and Hippurate:Citrate are the same since

the design is balanced, but the residual error pass from 53.9 to 11.8%. Indeed, the model is more complete and factors such as Medium, Time and Day are not anymore in the residuals.

Table 1.3: Percentages of explained variance and bootstrap test p-values for each model effect.

	% of variance	Observed p-value
Hippurate	27.6	< 0.001
Citrate	18.0	< 0.001
Medium	16.9	< 0.001
Time	1.7	< 0.001
Day	15.4	< 0.001
Hippurate:Citrate	0.5	0.315
Hippurate:Medium	0.5	0.335
Hippurate:Time	0.2	0.799
Hippurate:Day	1.2	0.192
Citrate:Medium	0.5	0.345
Citrate:Time	0.3	0.63
Citrate:Day	1.0	0.313
Medium:Time	1.4	0.014
Medium:Day	2.2	0.011
Time:Day	0.9	0.355
Residuals	11.8	

Figure 1.7 shows the scores and the loadings for the significant effects in ASCA-E and the residuals.

The score plots of Hippurate, Citrate, and Medium show a clear separation between their levels on the first principal component. For Time, a separation is visible between both groups even though an overlapping appears for some points. For Day effect, some overlapping is present between days 2 and 3, but day 5 appear well apart. For the interaction effects Medium:Time and Medium:Day, a clear separation of the groups is difficult to distinguish. In Chapter 2, tools such as confidence ellipses will show to be more adapted in such cases. For the residuals, most of the observations are centered except a group on the right and an individual observation on the left of the score plot.

For the loadings, the peaks corresponding to Hippurate and Citrate are correctly retrieved as highlighted in green and red respectively. For Medium and Time effects, a peak due to the (removed) water area of the spectra is visible and highlighted in blue. This peak due to water is also present for the

Medium:Time interaction. Interestingly, for the Day effect the most influential peaks correspond to the Hippurate region. Those peaks show a slight shift on the left probably for Day 5. For the Medium:Day interaction none of the previous peaks are detected. Finally, for the residuals the loadings seem rather noisy even though the Hippurate and Citrate regions are still visible.

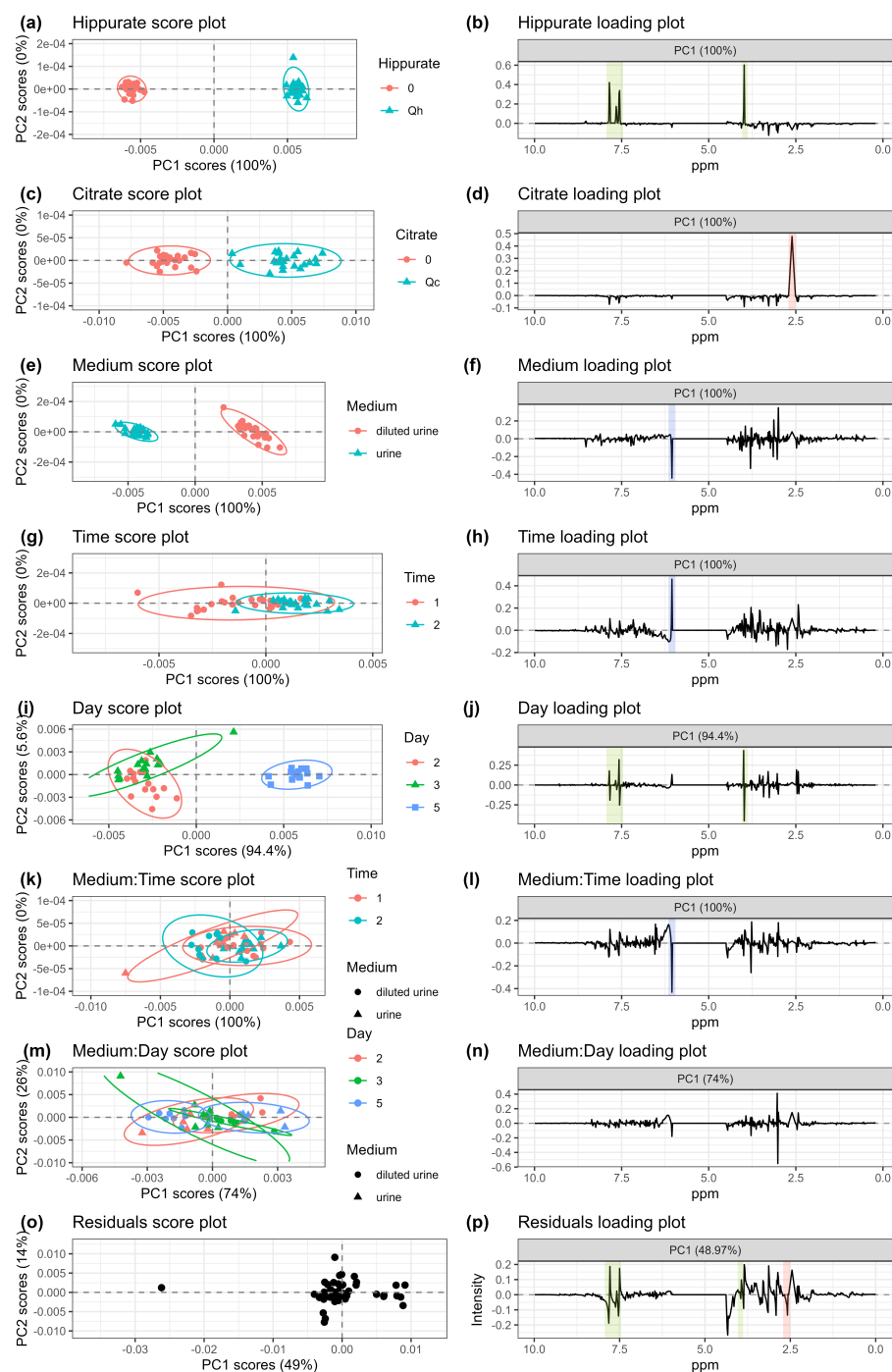


Figure 1.7: Score and loading plots in ASCA-E of the significant effects and the residuals.

1.4 Recent developments in ASCA Methodology

Since the article of Thiel, Féraud, and Govaerts (2017) where the classical ASCA methodology was redefined, the field has continued to evolve to address additional challenges encountered with multivariate designs. Subsequent research has focused for example on refining effect estimation in unbalanced designs, and importantly, extending the methodology to linear mixed models (LMMs) as well as longitudinal data. In this section, we outline these developments.

1.4.1 Rebalancing Strategies for Unbalanced Designs

Although the GLM formulation in ASCA+ yields unbiased estimators, the resulting effect matrices are not guaranteed to be mutually orthogonal. This non-orthogonality can confound the interpretation of the different sources of variation. Indeed, the effect matrices $\mathbf{M}_f = \mathbf{X}_f \hat{\boldsymbol{\Theta}}_f$ exhibit non-zero cross-products ($\mathbf{M}'_f \mathbf{M}_g \neq 0$ for $f \neq g$), introducing overlap in the explained variation. Practically, it means that the sum of the squared Frobenius norms of the effect matrices (plus residuals) does not equal the total variation, as some variation is shared across effects. This can lead to challenges in attributing percentages of variation uniquely to factors (e.g., variances may not sum to 100%), potentially over- or under-estimating individual effects. To mitigate this issue, rebalancing strategies were introduced (e.g., (Figueiredo et al., 2022; Boccad and Rudaz, 2023)). These approaches involve averaging observations within each unique combination of factor levels to create a balanced dataset prior to performing the ASCA decomposition. By replacing individual measurements with the corresponding group averages, the rebalanced data ensure that the subsequent decomposition produces orthogonal effect matrices. Compared to ASCA+, which uses a GLM formulation to obtain unbiased estimators, these rebalancing approaches rely on simple averaging rather than model-based estimation and can not support unified statistical inference on the original observations. Although this strategy enforces orthogonality in a straightforward way, it creates degrees of freedom which increase artificially the testing power. Its role is then more an intermediate refinement rather than a fundamental change in effect estimation.

1.4.2 LiMM-PCA: Integrating Dimensionality Reduction with Mixed Models

A major methodological step was recently proposed by Martin and Govaerts (2020) with the introduction of the LiMM-PCA framework (Linear Mixed

Models – Principal Component Analysis). This approach extends the ASCA+ methodology to handle random effects by combining dimensionality reduction via PCA with linear mixed models (LMMs). The first step involves a PCA transformation of the response matrix to obtain orthogonal components. Then, a separate linear mixed model is fitted for each principal component, allowing for the modeling of fixed effects (e.g., treatment, time and their interactions) and random effects (e.g., subject, batch). This strategy ensures that responses are uncorrelated and simplifies the parallel LMM fitting. Fixed and random effects are reconstructed from the LMM results and projected back into the original variable space, enabling visualizations and variance quantification. Significance testing is performed using log-likelihood ratio (LLR) tests and parametric bootstrap procedures, allowing inference on both fixed and random effects.

The method was demonstrated on two case studies: a sensory dataset and a metabolomics dataset with both fixed and random effects. In the sensory study, it successfully identified significant effects of candies, judges, and their interactions, revealing that some judges were outliers. In the metabolomics study, it quantified biological and analytical variability, confirming the dominance of volunteer and sampling effects. These applications underscore LiMM-PCA's ability to efficiently handle high-dimensional, designed data and linear mixed models.

1.4.3 Extending ASCA to Longitudinal Data Analysis

The increasing prevalence of longitudinal studies, particularly in metabolomics and clinical research, has further stimulated the evolution of ASCA-based methods for longitudinal studies generating repeated measurements over time and within-subject correlations.

Repeated Measures ASCA+ (RM-ASCA+) was introduced by Madssen et al. (2021a) as an adaptation of ASCA+ for longitudinal studies. RM-ASCA+ incorporates LMMs to account for the correlation among repeated observations from the same subjects. In this framework, the multivariate response matrix is decomposed into effect matrices corresponding to the overall mean, time, treatment, and time–treatment interactions. The inclusion of random effects captures inter-individual variability and allows for the application of PCA to these effect matrices, thereby facilitating the visualization and interpretation of temporal effects.

Recognizing that treating time as a categorical factor may limit the ability to capture temporal trends, further improvements have been made. Erdős et al. (2023) extended RM-ASCA+ by incorporating continuous time modeling through polynomials and splines within the LMM framework. This continuous

time extension enables the recovery of smooth, non-linear trajectories, providing a more precise quantification of temporal trends in studies with high-frequency sampling.

It is important to say that LiMM-PCA can also treat longitudinal data but that some adjustments need to be done with the coding. An advancement in this direction was proposed by Tournay (2020), whose master's thesis explored in detail how LiMM-PCA can be adapted to analyze longitudinal high-dimensional data, such as those generated by omics studies. In longitudinal designs, repeated measurements on the same individuals induce intra-subject correlations that violate the independence assumptions of classical multivariate methods. Tournay demonstrated that LiMM-PCA can be successfully adapted to account for this structure by specifying subject-level random effects in the mixed model component. Her thesis formalizes how the PCA-mixed model decomposition can accommodate these repeated measures. She provides a step-by-step implementation, both theoretically and in R, using a case study in metabolomics.

1.4.4 Extension to Generalized Linear Models

Finally, in his master's thesis Cyrille Nguipdop Lowe investigated the use of Generalized Linear Models (GLMs) for the statistical analysis of high-dimensional omics data under designed experiments (Nguipdop Lowe, 2022). His work demonstrated how GLMs can be effectively used to quantify the impact of experimental factors. For transcriptomics (RNA-seq) data, which consists of discrete and overdispersed count values, Cyrille employed negative binomial GLMs to model gene expression differences between conditions, while correctly modeling variance structure.

1.5 Conclusion and perspectives

In summary, this chapter advances ASCA methodologies by introducing ASCA+ and APCA+, which handle unbalanced designs through GLM-based estimation, a critical innovation that addresses biases in classical approaches. By ensuring correct variance partitioning and significance testing, our work empowers precise interpretation in high-dimensional multivariate data such as in omics and chemometrics, where variable counts far exceed samples. Industrially, ASCA+ brings solutions in pharmaceutical development or food production by enabling reliable analysis of imperfect data where imbalances are inevitable. Since our original article (Thiel, Féraud, and Govaerts, 2017), ASCA-based methodologies have also continued to progress. Early efforts focused on correcting bias in

unbalanced designs through rebalancing strategies, which improved the orthogonality of effect matrices. However, the most transformative progress has been achieved through the integration of mixed modeling, as exemplified by LiMM-PCA. This approach has set the stage for advanced longitudinal analyses such as RM-ASCA+ and the master's thesis of Tournay (2020). Chapter 2 will focus on the implementation of ASCA+ into an R package and further extend it to continuous factors. Some research on the detection of influential variables and representation of confidence ellipsoids will also be presented. Finally, for reproducibility purposes, the R codes and datasets used throughout this work are available at <https://github.com/MichelT7/phd-thesis-r-codes>.

CHAPTER 2

Practical implementation of ASCA methodologies: the
limpca package

2.1 Introduction

Chapter 1 introduced the ASCA and APCA methodologies, which combine the principles of ANOVA and PCA to study the effects of experimental factors on multivariate data. Classical ASCA and APCA approaches, initially developed for balanced designs, were discussed along with their limitations with unbalanced experimental designs. To address these limitations, a new methodology named ASCA+ was proposed, relying on the GLM framework to obtain unbiased estimators of model effects, even when the data are unbalanced.

The theoretical foundations of ASCA+ were presented, including the GLM decomposition, the estimation of effect matrices, the calculation of explained variance, and approaches for significance testing (see Figure 1.1). Building on these theoretical developments, this chapter focuses on the practical implementation of ASCA+ and related methodologies through the package R `limpca` (for linear models with principal component effects analysis). Several implementations of ASCA and ASCA-related methods exist in scientific software. Most of these are described in the two recent papers by Bertinetto et al. (Bertinetto, Engel, and Jansen, 2020) and Jarmund et al. (Jarmund, Madssen, and Giskeødegård, 2022). They are either available as R packages or Matlab code and other implemented in dedicated software like `MetaboAnalyst` (Xia et al., 2015). In R, one can cite `lmdme` (Fresno, Balzarini, and Fernandez, 2014), `MetStaT` (Dorscheidt, 2019), `ALASCA` (Jarmund, Madssen, and Giskeødegård, 2022), `gASCA` (Franceschi, 2022) and `multiblock` (Liland, 2023). Two of these are not updated like `lmdme` (since 2014), or have been archived like `MetStaT`. Two others are focused on specific models like `ALASCA` which treats longitudinal and cross-sectional multivariate data with RM-ASCA+ (Madssen et al., 2021b; Bertinetto, Engel, and Jansen, 2020), and `gASCA` devoted to the analysis of count data (Franceschi, 2022). `Multiblock` package, which offers a large collection of methods for multiblock data analysis, includes an implementation of ASCA for a large panel of models but its implementation remains, at this stage, quite basic and documentation limited (Liland, 2023). Compared to these software packages, `limpca` supports all fixed-effects linear models and offers the widest set of modeling outputs, such as general measures of effect importance, bootstrap significance tests, combined effects, and many `ggplot2` graphical outputs. The package is actively maintained on Bioconductor implemented with an aim of generalization to other models such as Linear Mixed Models (LMMs) and Generalized Linear Models (GLMs).

To remind its development, after the publication of Thiel, Féraud, and Govaerts (2017), questions were regularly coming from researchers if codes were

available to test the methodology. From that, I initiated the development of structured codes providing to the user the effect matrices, percentages of variation and significance assessment. An important contributor at this period was Robin Van Oirbeek, consultant at Janssen Pharmaceutica, who helped in automating the GLM decomposition depending on any model formula with fixed categorical factors in a first package version. Then, master students Sebastien Franceschini and Nadia Benaiche took over the development of the package by including additional features and visualizations. Finally, Manon Martin consolidated it as the current version available on Bioconductor. The package `limpca` underwent then all validation steps through Bioconductor's submission review, encompassing checks for code quality, dependency management and documentation completeness. While primarily developed in an academic context, it has a great potential for regulated environments such as the pharmaceutical industry, supported by R validation guidelines.

In this chapter, the ASCA+ methodology is reviewed, the structure, and the main functionalities of the package are presented. Moreover, three extensions proposed for future versions of the package are presented. Those extensions are namely the handling of continuous variables in ASCA+, the measurement of variable importance, and the computing of confidence ellipses. Real datasets are used to illustrate their application and highlight the capabilities of `limpca` for the analysis of multivariate data generated from experimental designs.

2.2 Limpca package organization and functions

This section presents the implementation of the R package `limpca`, standing for *linear models with principal component effects analysis*, which allows the statistical analysis of high-dimensional designed data using the ASCA+ and APCA+ methods (Benaïche, 2021). The package offers a comprehensive workflow first to explore the data set of interest, then to fit a chosen model to the data and finally to analyze the model results through a wide variety of statistics, tables and graphics. The package is now available on Bioconductor with two detailed vignettes and a user's manual (<https://bioconductor.org/packages/release/bioc/html/limpca.html>). `limpca` presents different advantages compared to other software in this field: (1) it is able to treat any balanced or unbalanced experimental design for fixed categorical factors, (2) it offers optimized methods to calculate effect importance and test their significance, (3) it allows the user to represent data and ASCA/APCA results with various and rich `ggplot2`-based graphical outputs that are highly customizable and (4) the package is open to future extensions to more sophisticated statistical models.

The package is organized in two sets of functions. The first set is dedicated to the exploration and visualization of the data set (outcomes and design) before modeling. The second set allows the modeling through the ASCA+ approach, as well as the visualization of the results. All these functions enable to build a comprehensive workflow for ASCA+/APCA+ data analysis. Figure 2.1 presents the set of functions for data exploration and Figure 2.2 presents the set for modeling and results visualization. The respective input and output objects are applied successively. The package can currently be installed from Bioconductor with the following command:

```
if (!require("BiocManager", quietly = TRUE))  
  install.packages("BiocManager")
```

```
BiocManager::install("limpca")
```

Data exploration

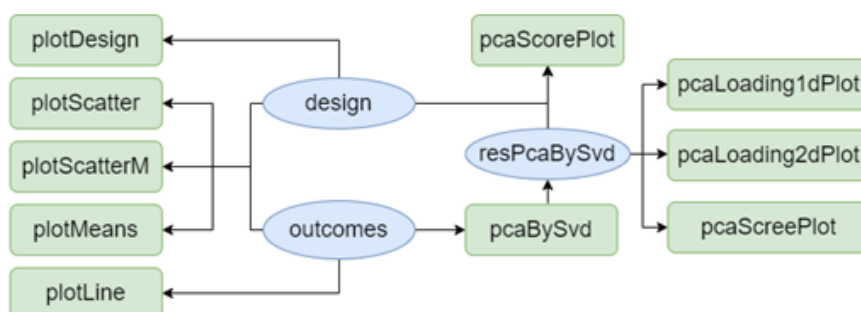


Figure 2.1: Overview of the main `limpca` objects and functions for data exploration.

Modeling and results visualization

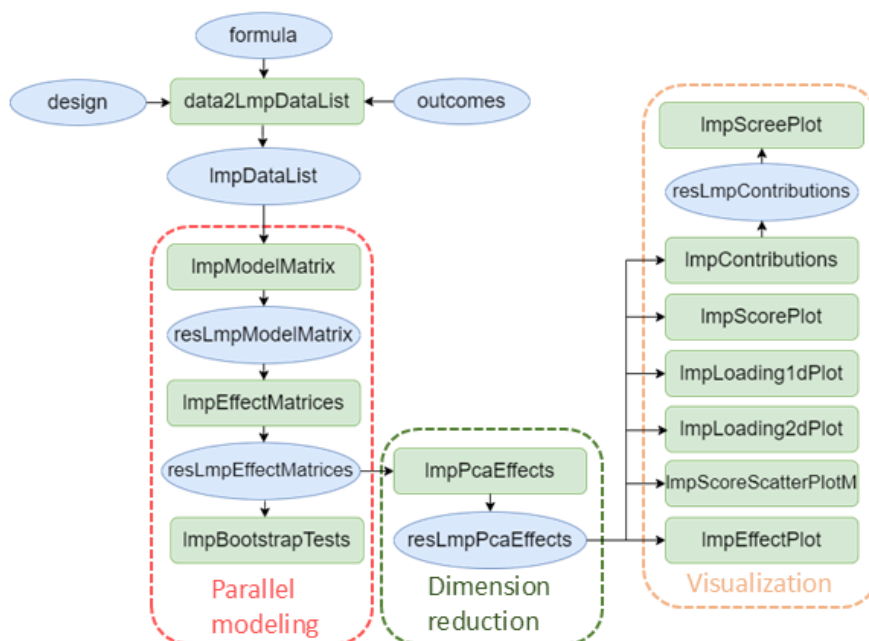


Figure 2.2: Overview of the main `limpca` objects and functions for modeling and results visualization.

2.2.1 Data exploration

This first part of the package aims to explore the data included in the outcomes and design R objects (Figure 2.1). The outcomes object should be a $n \times m$ numerical matrix of n observations and m response variables and the design object must be a $n \times k$ R data frame of n observations and k factors. A first series of graphical functions helps to represent the design and the responses and another set of functions allows to perform a PCA on the response matrix and observe its results according to the levels of the design factors. A practical tip for the users is to start with **plotDesign** to verify balance, then use **pcaScorePlot** and **plotScatterM** to identify outliers or patterns before modeling.

plotDesign

Provides a graphical representation of the experimental design. It allows to visualize factor levels and check the design balancing.

plotScatter

Produces a plot describing the relationship between two columns of the outcomes matrix **Y**. It allows to choose colors and symbols for the levels of the design factors. Ellipses, polygons or segments can be added to group different sets of points on the graph.

plotScatterM

Produces a scatter plot matrix between selected columns of the outcomes matrix **Y** choosing specific colors and symbols for up to 4 factors from the design on the upper and lower diagonals.

plotMeans

Draws, for a given response variable, a plot of the response means by levels of up to three categorical factors from the design. When the design is balanced, it allows to visualize main effects or interactions for the response of interest. For unbalanced designs, this plot must be used with caution.

plotLine

Generates the response profile of one or more observations i.e. plots of one or more rows of the outcomes matrix on the y-axis against the m response variables on the x-axis. Depending on the response type (spectra, gene expression...), point, line or segment plots can be used.

pcaBySvd

Operates a Principal Component Analysis on the $\mathbf{Y}$ outcome/response matrix by a singular value decomposition. Outputs are represented with the functions `pcaScreePlot`, `pcaScorePlot` and `pcaLoadingPlot`.

pcaScreePlot

Returns a bar plot of the percentage of variance explained by each Principal Component (PC) calculated by `pcaBySvd`.

pcaScorePlot

Produces score plots from `pcaBySvd` output with the same graphical options as `plotScatter`. This is a wrapper of `plotScatter`.

pcaLoading1dPlot

Plots loading vectors from `pcaBySvd` output with different available line types. This is a wrapper of `plotLine`.

pcaLoading2dPlot

Produces 2D loading plots from `pcaBySvd` output with the same graphical options as `plotScatter`. This is a wrapper of `plotScatter`.

2.2.2 Modeling and results visualization

The second part of the package contains the functions that apply the ASCA+ modeling step as well as the visualization of the results (Figure 2.2). The functions framed by the red dashed line, perform the modeling and test the significance of the model effects, while those framed by the orange dotted line allow the visualization of the results of PCA dimension reduction of the (augmented) effect matrices.

The analysis starts from the `lmpDataList` object which is a list of three elements produced by the function `data2LmpDataList`. The first two elements are the $n \times m$ outcomes matrix and the $n \times k$ design data frame, both discussed in Section 2.2.1. The third element is the R formula of the model to be estimated. For example, for a two-way crossed ANOVA 2 model with factors A and B, it will take the form: `~A+B+A:B`.

From the `lmpDataList`, a series of functions can be applied consecutively, each one corresponding to a step in the ASCA+ methodology. The GLM is first estimated via `lmpEffectMatrices` and effects importance evaluated. Finally, the result of this modeling, `resLmpPcaEffects`, serves as an input argument to

all visualization and interpretation functions such as `ImpScorePlot` for scores or `ImpLoading1dPlot` for loadings.

data2LmpDataList

Converts data to a `lmpDataList`, the input argument for `lmpModelMatrix`.

lmpModelMatrix

Creates the model matrix $\mathbf{X}$ from the design matrix and the model formula.

lmpEffectMatrices

Estimates the model by OLS based on the outcomes and model matrices provided in the outputs of `lmpModelMatrix` function and calculates the estimated effect matrices $\hat{\mathbf{M}}_0, \hat{\mathbf{M}}_1, \dots, \hat{\mathbf{M}}_f$ and residual matrix $\hat{\mathbf{E}}$. It calculates also the type III percentage of variance explained by each effect.

lmpBootstrapTests

Tests the significance of one or a combination of the model effects using bootstrap. This function is based on the outputs of `lmpEffectMatrices` function.

lmpPcaEffects

Performs a PCA on each of the effect matrices from the outputs of `lmpEffectMatrices`. It has an option to choose the method applied: ASCA, APCA or ASCA-E. Combined effects (i.e. linear combinations of original effect matrices) can also be created and decomposed by PCA.

lmpContributions

Reports the contribution of each effect to the total variance, but also the contribution of each PC to the total variance per effect. Moreover, these contributions are summarized in a barplot.

lmpScreePlot

Provides a barplot of the percentage of variance associated to the PCs of the effect matrices ordered by importance based on the outputs of `lmpContributions`.

lmpScorePlot

Draws the score plots of each single or combined effect matrix provided in `lmpPcaEffects` outputs. As a wrapper of the `plotScatter` function, it allows to visualize the scores of the effect matrices for two components at a time with all

the available options in `plotScatter`.

ImpLoading1dPlot

Plots the loading vectors for each single or combined effect matrix from the `ImpPcaEffects` outputs with line plots. This is a wrapper of `plotLine`.

ImpLoading2dPlot

Draws a 2D loading plot of each effect matrix provided in `ImpPcaEffects` outputs. As a wrapper of the `plotScatter` function, it allows the visualization of effect loading matrices for two components at a time with all options available in `plotScatter`.

ImpScoreScatterPlotM

Plots the scores of all model effects simultaneously in a scatterplot matrix. By default, the first PC only is kept for each model effect and, as a wrapper of `plotScatterM`, the choice of symbols and colors to distinguish factor levels allows an enriched visualization of the factors' effect on the responses.

ImpEffectPlot

Plots the ASCA scores by factor levels for a given model effect and for one PC at a time. This graph is especially appealing to interpret interactions or combined effects. This is a wrapper of `plotMeans`.

2.2.3 Application on Urine-Citrate-Hippurate dataset

The package `limpca` includes two data sets with their dedicated vignettes: UCH and Trout. In this chapter, due to edition limitations, only a basic analysis of the UCH data set is provided. But the reader can find extended results for both datasets in the vignettes on Bioconductor:

<https://bioconductor.org/packages/release/bioc/html/limpca.html>.

The dataset used to illustrate the methods in the next section is a subset of the Urine-Citrate-Hippurate data set described in (Rousseau, 2011), used in Chapter 1 and more extensively by (Guisset, Martin, and Govaerts, 2019) to compare ASCA+/APCA+, PARAFASCA, AComDim and AMOPLS methodologies.

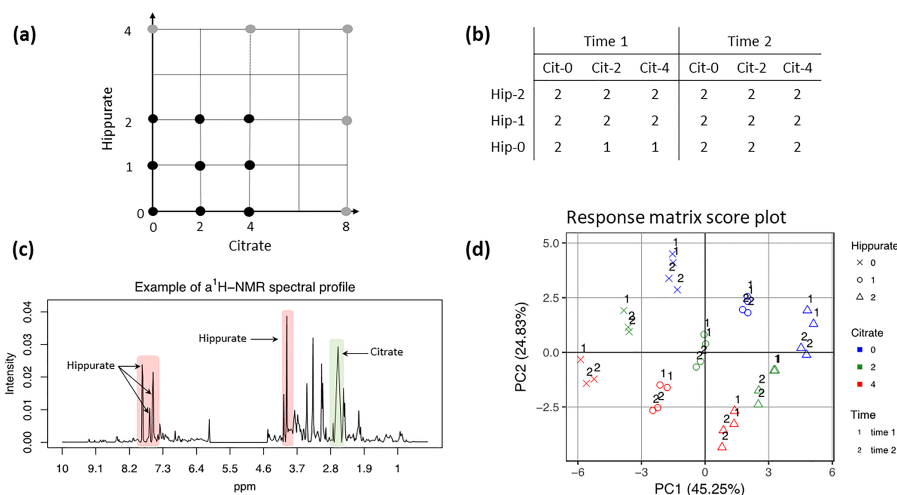


Figure 2.3: Presentation of the UCH data set (a) Experimental design (b) Sample sizes for each of the 9 combinations of Citrate, Hippurate and Time levels (c) Example of a ^1H -NMR spectral profile showing the Hippurate and Citrate peaks (d) Score plot for the first two PCs of the UCH data set response matrix (from (Guisset, Martin, and Govaerts, 2019)).

This dataset contains metabolomic spectral profiles obtained by ^1H -NMR spectroscopy of samples coming from a pool of rat urine samples. These samples were prepared by spiking the pool with known concentrations of chemicals (Hippurate and Citrate) in order to obtain data based on an experimental design and study the quality of resulting spectra. In the subset of this data set used here and in Guisset, Martin, and Govaerts, 2019, and further referred to as the UCH dataset, each of these factors has three levels: no added chemical (a concentration of 0), a medium concentration (2 mM for the Citrate factor ($Q_c/4$) and 1 mM for the Hippurate factor ($Q_h/4$)) and a high concentration (4 mM for the Citrate factor ($Q_c/2$) and 2 mM for the Hippurate factor ($Q_h/2$)). This leads to an experimental design with 9 possible combinations of concentrations as illustrated in Figure 2.3(a). Four samples of each of the nine urine combinations were prepared, then frozen and analyzed (by sets of 18) over two days. Each day, the analysis of the two samples with the same Hippurate and Citrate proportions was done under two times: just after defreezing and one hour later in order to evaluate the stability of samples over time. The UCH data selected for the paper originally contained 36 spectra: 9 combinations of concentrations $\times$ 2 days $\times$ 2 replicates (directly and one hour after defreezing). However, two outliers were removed, resulting in 34 observations and a slightly unbalanced experimental design (see Figure 2.3(b)).

This subset is then different than the ones presented in Section 1.3. Indeed the combinations of Hippurate and Citrate are different, but also the days and only one medium is considered.

The resulting spectral matrix $\mathbf{Y}$ has a dimension of (34×600) since the resolution of the ^1H -NMR spectra equals 600. In these spectra, 4 Hippurate peaks are located around 3.96, 7.54, 7.62 and 7.82 ppm (part per million) and the aggregated Citrate peak is located around 2.6ppm (see Figure 2.3(c)). Figure 2.3(d) presents the score plot of the first two PCs for the response matrix. It shows that the information on the Citrate and Hippurate factor levels from the experimental design can be recovered, but only to some extent and mixed in each principal component. This was already observed on the other subset of this dataset analyzed in Chapter 1.

2.2.4 Limpca results

Data importation and preparation

Before any analysis, the data need to be loaded in R and organized in the form of a list containing a design matrix, an outcomes matrix and a model formula. The UCH object can be loaded with the `data(UCH)` command and contains 3 elements: a matrix of outcomes with 34 observations and 600 response variables representing the spectra coming from the ^1H -NMR spectroscopy, the design matrix with 34 observations and 5 explanatory variables and the R formula for the GLM model used. In the design matrix, Hippurate, Citrate and Time are the only variables used as factors in the model. The two days (Day) are considered as single replicates and included in the residuals. The Dilution is a constant factor. The chosen model becomes then the full three-way ANOVA containing the main effects of the three factors and all two and three way interactions between them. The R formula can be written as follows:

```
outcomes ~ Hippurate + Citrate + Time + Hippurate:Citrate +  
Time:Hippurate + Time:Citrate + Hippurate:Citrate:Time
```

Data visualization

This section shows different functions offered by `limpca` to visualize the data before the modeling step. The design matrix can be visualized with the function `plotDesign` as illustrated in Figure 2.4(a) with the Citrate factor along the x axis, Hippurate along the y axis, Day in rows and Time in columns. The profiles of ^1H -NMR spectra can be visualized with `plotLine` as shown in Figure 2.4(b) for two chosen spectra. Values of two responses (ppm descriptors) for all

samples can be visualized using the function `plotScatter`. Figure 2.4(c) shows the spectral intensities at 2.61 vs 3.98 ppm which are the peaks corresponding to the Citrate and Hippurate respectively. This plot shows 9 groups of points corresponding to the different combinations of Citrate and Hippurate concentrations in the experimental design.

Another way to visualize the values of several variables simultaneously and adding the information from up to 4 factors of the design is to use the function `plotScatterM` working in a similar way as function `ImpScoreScatterPlotM` and demonstrated in Figure 2.7. Finally, the function `plotMeans` allows to visualize, for a given response variable, a plot of the means by factors' level of up to three categorical factors of the design. Figure 2.4(d) shows the mean response at 3.98 ppm, which is one of the peak of Hippurate, as a function of Hippurate, Time and Citrate levels. It can be observed that, as expected, a linear evolution in spectral intensity occurs when the level of Hippurate increases.

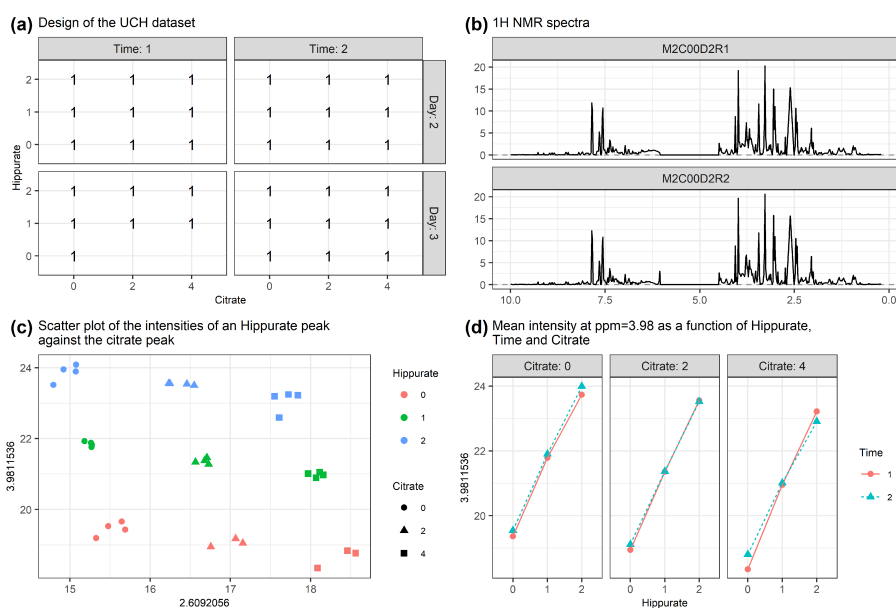


Figure 2.4: (a) Design of the UCH data set, (b) ^1H -NMR spectra, (c) Scatter plot of two responses and (d) typical `plotMeans` output.

Function `pcaBySvd` is useful to compute a PCA decomposition of the outcomes matrix $\mathbf{Y}$. Figure 2.5 (a) shows the percentages of variance explained by the first 6 components obtained with the function `pcaScreePlot`. It can be observed that the first three PCs explain more than 86% of the total variability. Figures 2.5(b) and (c) show the scores and the loadings of the first two PCs plotted with the functions `pcaScorePlot` and `pcaLoading1dPlot`. The score plot allows to see the 9 groups of points associated with the different combinations of Citrate and Hippurate concentrations. The loading plot shows also the variables corresponding to Citrate and Hippurate peaks as explained in Figure 2.3(c). Note that the PCs do not clearly separate the 2 products since the shape of the design is not parallel to the score plot axes and components are then mixtures of the spectral profiles of Citrate and Hippurate.

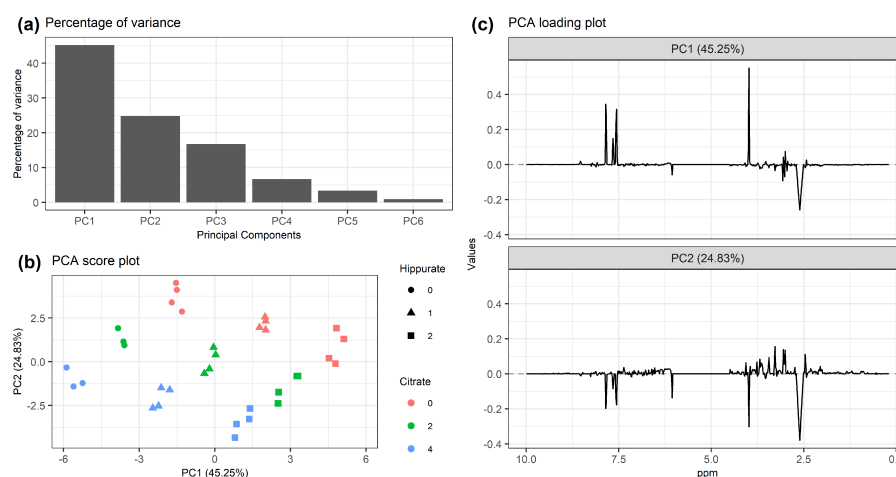


Figure 2.5: PCA percentages of variance, score and loading plots.

Modeling

The modeling step consists of decomposing the outcome matrix $\mathbf{Y}$ into effect matrices and performing PCA on the effect matrices or their combination with the residual matrix. The function `ImpModelMatrix` creates the model matrix $\mathbf{X}$ from the design and the model formula using sum coding for the factor coding. As explained above, the model to be estimated here is a full three way ANOVA model on factors Hippurate, Citrate and Time. Thereafter, the function `ImpEffectMatrices` estimates the GLM and derives the effect matrix for each model effect. This function computes also the percentage of variance linked to

each effect using the type III Sum of Squares.

Bootstrap can then be used to determine which effects are significant using the function `ImpBootstrapTests`. Table 2.1 presents the percentages of variance and the corresponding bootstraps p-values, obtained from 1000 resampled outcomes samples. It indicates that the main effects of Hippurate, Citrate and Time as well as the interaction between Hippurate and Time are significant. These effects explain, respectively, 39.31%, 29.91%, 16.24% and 6.23% (i.e. in total 91.69%) of the total variance of the responses. A barplot of the percentages of variance can be obtained using the function `ImpEffectMatrices` but is not shown here.

Table 2.1: Percentages of explained variance and bootstrap test p-values for each model effect.

	% of variance	p-value
Hippurate	39.31	< 0.001
Citrate	29.91	< 0.001
Time	16.24	< 0.001
Hippurate:Citrate	1.54	0.147
Hippurate:Time	6.23	< 0.001
Citrate:Time	0.54	0.418
Hippurate:Citrate:Time	1.68	0.089
Residuals	4.3	

Modeling visualization

This section presents the different functions available to visualize the results of the modeling step. The `ImpPcaEffects` function has an argument to define which method to use among ASCA, ASCA-E or APCA to decompose by PCA each of the effect matrices $\hat{\mathbf{M}}_f$ obtained in the second step of the analysis.

Through the `ImpContributions` function, the results shown in Table 2.2 are generated. This table indicates, for each effect, the percentage of variance explained by each PC, reported to the effect contribution to the total variance. This gives a view of the global importance of each PC regardless of the effects. For this case study, the analysis will therefore be focused on the first PCs of the three main effects, the interaction effect between Hippurate and Time and the

residuals. The other PCs account for less than 1% of the total variance and are then not of big interest.

Table 2.2: PCs contributions by effect.

	PC1	PC2	PC3	PC4	PC5	Contrib (%)
Hippurate	38.41	0.9	0	0	0	39.31
Citrate	29.37	0.53	0	0	0	29.91
Time	16.24	0	0	0	0	16.24
Hippurate:Citrate	0.68	0.59	0.23	0.04	0	1.54
Hippurate:Time	5.85	0.38	0	0	0	6.23
Citrate:Time	0.49	0.05	0	0	0	0.54
Hippurate:Citrate:Time	0.8	0.46	0.38	0.05	0	1.68
Residuals	2.09	0.73	0.44	0.25	0.19	4.3

Figure 2.6 shows the scores and the loadings for the significant effect matrices obtained with the functions `lmpScorePlot` and `lmpLoading1dPlot` using the ASCA-E method. Residuals of the model are also displayed. For the main effects, all score plots show a clear separation between the different levels of the considered effects on the first PC. The distinct separation of the groups suggests a strong effect of those factors, confirmed by their significance.

For the loadings, note that ASCA and ASCA-E loading plots are the same while the ones for APCA contain the residuals of the model. Figure 2.6 (b) and (d) of Hippurate and Citrate loading plots separate and clearly retrieve the peaks of the Hippurate and Citrate molecules in contrast to the results of the classical PCA where peaks of Hippurate and Citrate were superimposed. Figure 2.6 (f), representing the loadings of the Time effect, shows a high peak on the left of the (removed) water area of the spectra. This is probably due to a shift of most spectral peaks with Time which is visible all over the spectral profile.

In Figure 2.6 (g), the interaction score plot is less straightforward to interpret. This is common in ASCA results and will be rediscussed below. In contrast, the corresponding loading plot in Figure 2.6 (h) is clearly showing a peak shift around 3 ppm. Figure 2.6 (i), representing the Residual score plot, is also interesting as in the chosen linear model, one has made the hypothesis of a negligible and random Day effect. However, the plot shows a quite clear Day effect indicating that this factor could have been included in the model.

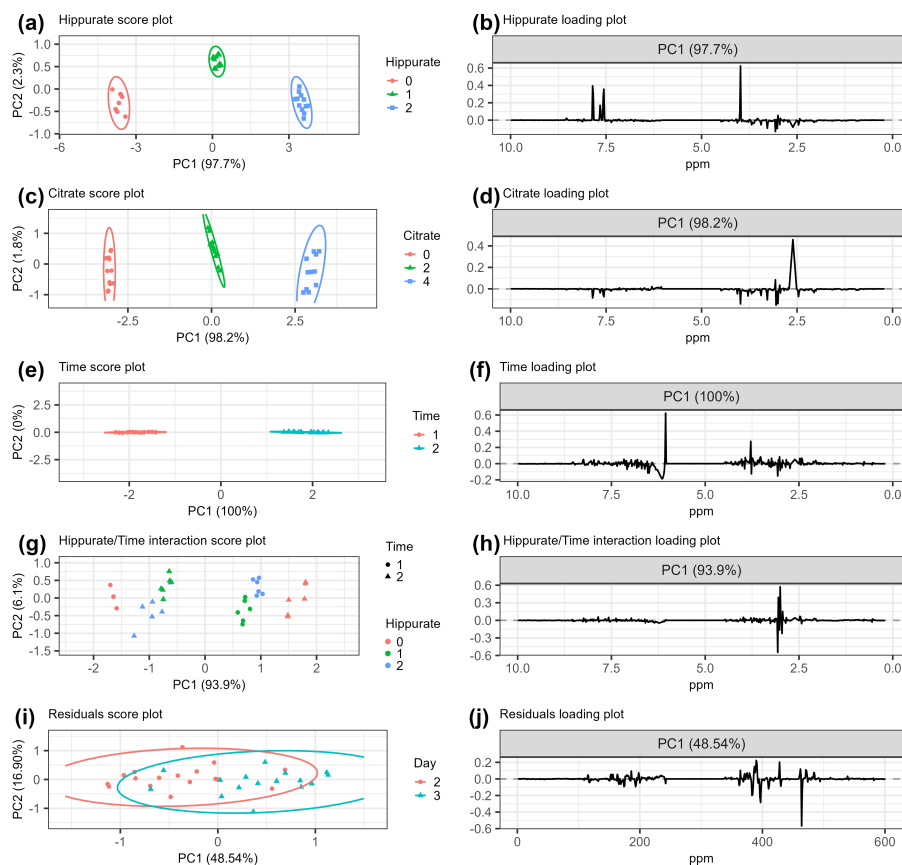


Figure 2.6: Score and loading plots in ASCA-E of the significant effects as well as residuals.

In order to get the full picture, the `ImpScoreScatterPlotM` function can be used to produce a score plot matrix of all the effects simultaneously. Figure 2.7 shows the scores of the first PC for the factors Hippurate, Citrate, Time and the Hippurate:Time interaction obtained with ASCA-E. When using different colors and symbols for factor levels, such graph can be very informative.

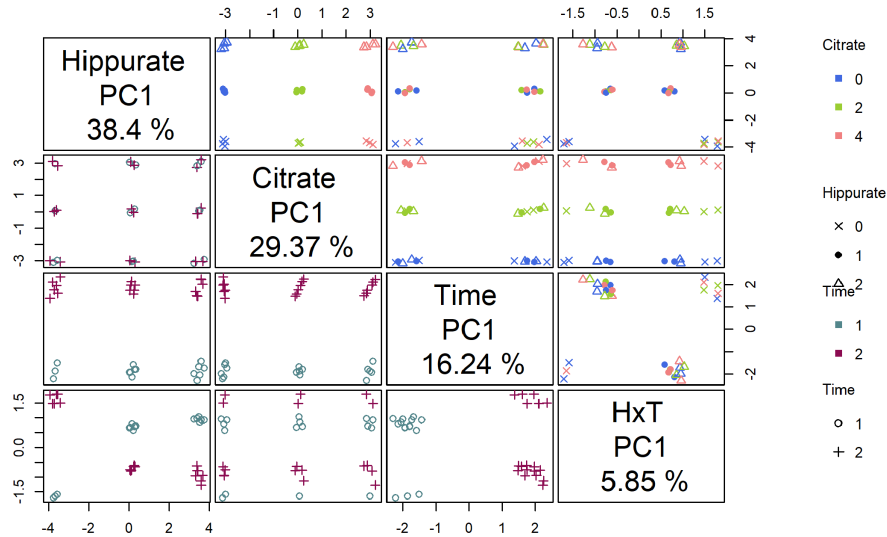


Figure 2.7: Output of the function `lmpScoreScatterPlotM`.

The interpretation of the interaction effects being always quite complicated in ASCA models, `limpca` offers two interesting functionalities. First, effect matrices can be combined before being decomposed by PCA. Second, the function `lmpEffectPlot` allows to represent the ASCA scores of an effect according to the levels of the same effect or another one.

Figure 2.8 illustrates this on the Hippurate:Time interaction. In the first line, the scores of the Hippurate:Time interaction are displayed first for the 2 first PCs. They are then represented in a line plot with the function `lmpEffectPlot` for the two components separately. This allows to better visualize the interaction than with classical score plots. The first line gives the impression of a quite important interaction effect. The second line shows the same graphs but on the basis of the PCA decomposition of the combination (sum) of the Hippurate, Time and Hippurate:Time effect matrices and shows that, even if significant, the interaction is quite small compared to the main Hippurate and Time effects. The Hippurate effect is mainly visible on the first PC and the Time effect on the second one.

In summary, all the graphs given above illustrate how ASCA methodology allows to analyze very deeply the informations contained in multivariate responses

issued from multi-factorial designed experiments and discover information that is difficult to extract from a simple PCA.

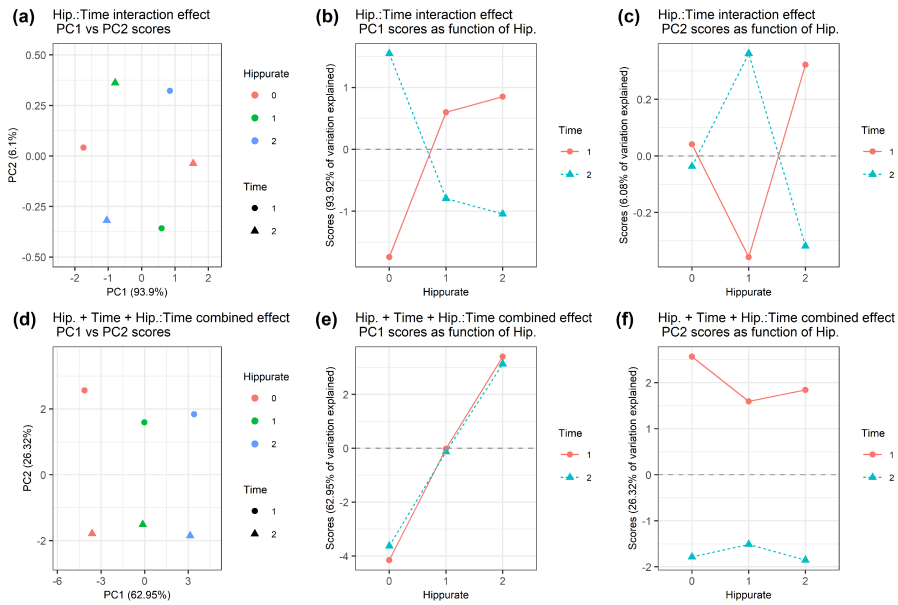


Figure 2.8: Outputs of `1mpEffectPlot` function.

2.3 Extension to continuous variables

In the previous section, the scope was to show the implementation of ASCA+ and APCA+ methodologies in the R package `limpca`. Currently, as ASCA/APCA were originally based on (balanced) ANOVA, `limpca` and ASCA(+) literature are only applied to models and designs including categorical factors even if general linear models can of course include combinations of categorical and continuous variables.

In this section, the extension of the methodology to continuous variables and non-designed studies is discussed, as well as the adjustments needed in `limpca` for this specific case.

2.3.1 Methodology

The general principle proposed here for introducing continuous factors into an ASCA(+) model consists of maintaining the same four-step philosophy as in ASCA+ and APCA+.

1. Estimate the parameters of the general linear model of interest by OLS : $\mathbf{Y} = \mathbf{X}\boldsymbol{\Theta} + \mathbf{E}$ and $\hat{\boldsymbol{\Theta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$
2. Decompose $\mathbf{Y}$ into effect matrices of same dimensions $\mathbf{Y} = \widehat{\mathbf{M}}_0 + \sum_{f=1}^F \widehat{\mathbf{M}}_f + \widehat{\mathbf{E}}$ as in Equation 1.12.
3. Calculate the importance of these effects (or combined effects) in the model with p-values and percentage of explained variance.
4. Vizualize the (significant) effect matrices $\widehat{\mathbf{M}}_f$ trough a PCA decompostion.

The main issue with this implementation is that introducing continuous variables into the model often leads to correlation between the parameters (and non-orthogonality of effect matrices), which clearly hinders the ability to interpret them separately as is done in ASCA. This occurs because, in practice continuous variables frequently exhibit natural correlations with each other or with categorical factors (particularly in observational studies), resulting in linearly dependent columns in the design matrix $\mathbf{X}$. Including polynomial terms such as X and X^2 can also introduce multicollinearity due to their inherent correlation, potentially inflating variance in parameter estimates. This problem is typically less present in designs with categorical factors, when the factors are encoded using sum coding in the model matrix $\mathbf{X}$, and it may be entirely absent in balanced designs. In this later case a complete orthogonal variance decomposition of the outcomes matrix $\mathbf{Y}$ is obtained (i.e. each model matrix $\widehat{\mathbf{M}}_f$ is orthogonal to each other).

To approximate this property when the design matrix includes continuous variables, or in the context of an observational study, continuous variables should then be introduced into the model in a standardized form (centered and scaled). The following transformation is commonly recommended in the experimental design literature and implemented in most design of experiments software.

$$X^s = 2 \frac{(X - \min X)}{(\max X - \min X)} - 1$$

where X is the continuous variable of interest and $\min X$ (resp. $\max X$) is the minimum (resp. maximum) value of X over the dataset. This transformation results in a variable X^s scaled to the range $[-1,1]$ and ensures consistency with the sum-coding of categorical factors. It will be adopted here but could be replaced with mean and standard deviation scaling without big loss of interpretability.

For unbalanced designs or observational studies, this transformation should of course be accompanied by a careful examination of the collinearity between the model effects through the study of the estimated parameters correlation matrix (derived from $(\mathbf{X}'\mathbf{X})^{-1}$) or other indices such as the VIF.

Finally, one last technical point must be addressed in the generalization of ASCA to continuous factors. Most of the results being plots (such as score plots), where the factor levels are represented by colors and symbols, continuous color scales will need to be introduced to allow the result interpretation with respect to both types of factors. Continuous factors can also simply be introduced as categorical in the model. All this will be illustrated below.

2.3.2 Application on Sacurine dataset

This section shows an illustration of the implementation of ASCA+/APCA+ on data combining categorical and continuous variables. It allows also to show that this methodology is not reserved to designed studies but can also be applied to observational studies. The dataset used here comes from the article of Thévenot et al., 2015, which investigated physiological variations in the human urinary metabolome using a cohort of 183 adult volunteers. The urine samples were analyzed by liquid chromatography coupled with high-resolution mass spectrometry (LC-HRMS) to assess the influence of three key factors: age, body mass index (BMI), and gender. The cohort includes 100 males (55%) and 83 females (45%), with an average age of 40.9 ± 10.3 years (mean $\pm$ standard deviation) and a BMI of 23.6 ± 3.1 kg/m². Notably, BMI exhibits a significant positive correlation with age and is higher in males. In this section, 49 metabolites are retained from this dataset resulting in a matrix of outcomes $\mathbf{Y}$ with 183 observations and 49 response variables. In this case study Age has

been considered as a continuous variable, BMI as categorical variable with three groups “underweight”, “normal”, “overweight”, and Gender as a categorical variable too. Note that the outcomes in this dataset were standardized (i.e. mean-centered and variance scaled).

2.3.3 Data visualization

Figure 2.9(a) shows the design matrix of the sacurine dataset. Age is on the x axis, BMI on the y axis and Gender in columns. A metabolite profile for one of the participant can be visualized on Figure 2.9(b). Figure 2.9(c) shows the mean response of testosterone glucuronide over the age of the participants. The mean response is colored by the gender of the participants. It can be observed that this metabolite decreases over the age, with a slope more pronounced for women.

Outcome matrix $\mathbf{Y}$ is as first decomposed by PCA using function `pcaBySvd`. Figure 2.9(d) shows the percentages of variance explained by the first 5 components. The first two PCs explain 38.7% of the total variability and the first five PCs 61.3%. Figure 2.9(e) and (f) show the scores and the loadings of the first two PCs. The scores are shaped by Gender and colored by Age. The color scale has been adapted to display a gradient since Age is continuous. This kind of display is inspired from professional softwares such as JMP to make a distinction with categorical factors. On the loading plot, one can observe that acetaminophen glucuronide is the most influential metabolite on PC1 and p-anisic acid on PC2. Interestingly, on the right side of the score plot (positive PC1 scores) a pack of points is visible and spread linearly over PC2.

2.3.4 ASCA modeling

To keep it simple, the model chosen here to decompose the outcome matrix consists of the main effects of Age and Gender, and the interaction between them. The main effect of BMI was not significant and not retained in the model. The R formula of the “continuous” model can be written as follows:

$$\text{outcomes} \sim \text{Age} + \text{Gender} + \text{Gender:Age}$$

Table 2.3 shows the correlation between the different effects of the “continuous” model. All the correlations are relatively low. The two highest are between the Gender and Gender-Age interaction effects (-0.11), followed by Intercept and Gender (-0.10).

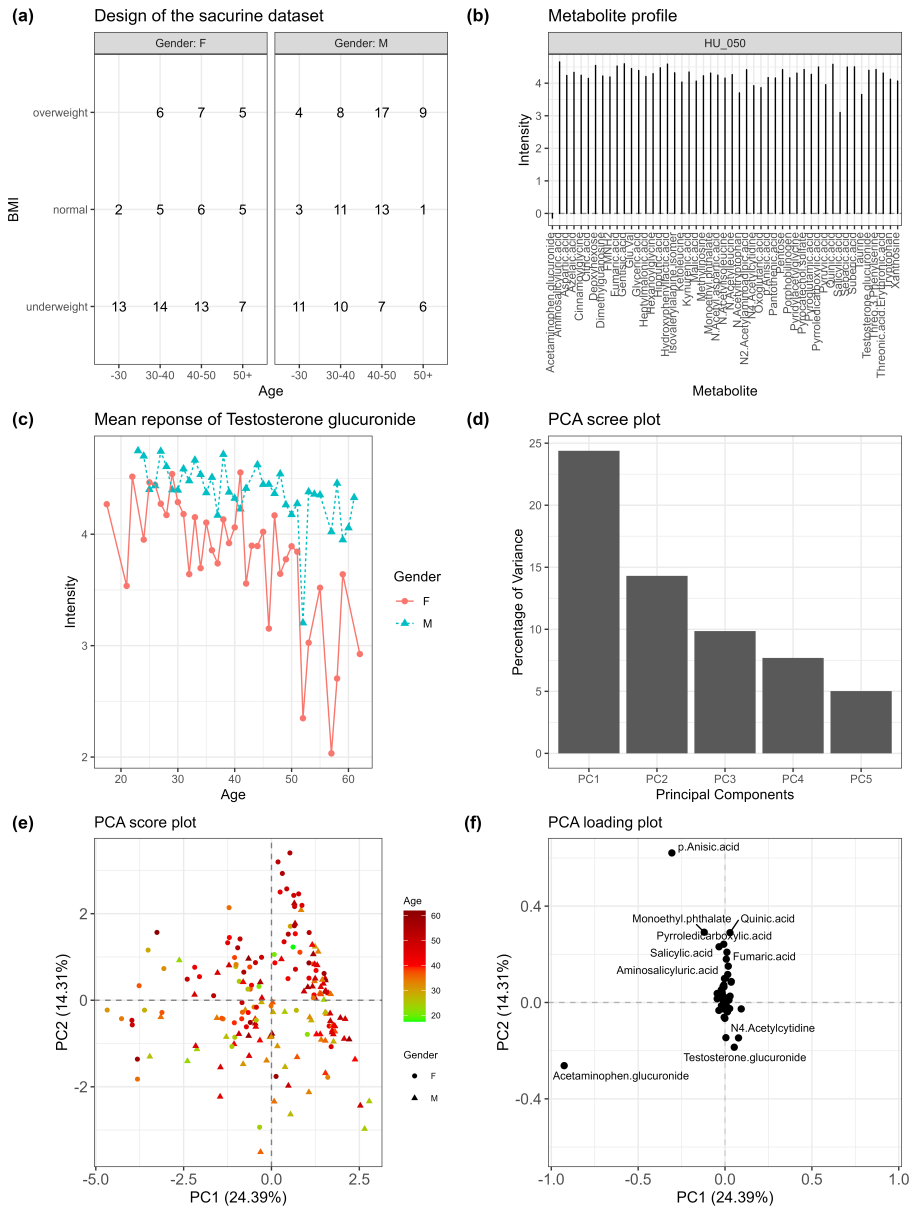


Figure 2.9: (a) Design of the sacurine dataset, (b) metabolite profile, (c) mean response of testosterone glucuronide, (d) PCA percentages of variance, (e) score plot and (f) loading plot.

Table 2.3: Correlation matrix of the parameter estimates of the “continuous” model

	Intercept	Gender	Age	Gender:Age
Intercept	1.00	0.10	-0.11	-0.03
Gender	0.10	1.00	-0.03	-0.11
Age	-0.11	-0.03	1.00	0.04
Gender:Age	-0.03	-0.11	0.04	1.00

Table 2.4 presents the percentages of variance and the corresponding bootstraps p-values, obtained from 1000 resampled outcomes samples. It indicates that the main effects of Age and Gender are significant but not the interaction between them. Similarly, the quadratic effect of Age and its interaction with Gender are not significant neither.

Table 2.4: Percentages of explained variance and bootstrap test p-values for the continuous model.

	% of variance	p-value
Age	3.55	< 0.001
Gender	6.38	< 0.001
Gender:Age	0.72	0.15
Residuals	89.19	

As a point of comparison, the same dataset was used but the variable Age was categorized in 4 groups: -30 , $30-40$, $40-50$ and $50+$. The model chosen to decompose the outcome matrix consist of the main effects of Age and Gender and the interaction between them. The R formula of the “categorical” model can be written as well as:

$$\text{outcomes} \sim \text{Age} + \text{Gender} + \text{Gender:Age}$$

Table 2.5 shows the correlation between the different effects of the “categorical” model. The correlations between Gender and levels of Age are very low (maximum 0.04), reducing the risk of collinearity. Similarly, the correlations between the levels of Age and Gender:Age are small (maximum 0.13). Somehow higher correlations can be observed between Gender and the interaction term Gender:Age (maximum 0.21), which is explained by the

construction of the interaction itself. Finally the correlation between the intercept and the different effects is maximum (0.21).

Table 2.5: Correlation matrix of the parameter estimates of the “categorical” model

	Intercept	Gender	f_Age1	f_Age2	f_Age3	Gender:f_Age1	Gender:f_Age2	Gender:f_Age3
Intercept	1.00	0.06	0.15	-0.14	-0.21	0.03	-0.00	0.04
Gender	0.06	1.00	0.03	-0.00	0.04	0.15	-0.14	-0.21
f_Age1	0.15	0.03	1.00	-0.35	-0.33	0.08	-0.04	-0.06
f_Age2	-0.14	-0.00	-0.35	1.00	-0.19	-0.04	0.07	-0.05
f_Age3	-0.21	0.04	-0.33	-0.19	1.00	-0.06	-0.05	0.13
Gender:f_Age1	0.03	0.15	0.08	-0.04	-0.06	1.00	-0.35	-0.33
Gender:f_Age2	-0.00	-0.14	-0.04	0.07	-0.05	-0.35	1.00	-0.19
Gender:f_Age3	0.04	-0.21	-0.06	-0.05	0.13	-0.33	-0.19	1.00

Table 2.6 presents the percentages of variance and the corresponding bootstraps p-values, obtained from 1000 resampled outcomes samples when all the variables are categorical. It indicates that the main effects of Age and Gender are significant but not the interaction between them.

In both cases, the sum of percentages of variance does not equal to 100%. For the “continuous” model, the sum equals 97.48% and for the “categorical” model the sum equals 99.16%. The percentages of variation and the p-values are coherent between both models.

Table 2.6: Percentages of explained variance and bootstrap test p-values for the categorical model.

	% of variance	p-value
Age	4.52	< 0.001
Gender	5.67	< 0.001
Gender:Age	1.91	0.141
Residuals	87.06	

Figure 2.10 shows the scores and the loadings in ASCA-E for Gender, Age and Age-Gender interaction effect matrices as well as the residuals of the model.

Figure 2.10(a)(c)(e)(g) present the scores for effects Gender, Age, Gender-Age interaction and the residuals. For Gender, the two groups are colored by level and circled by data ellipsoids. The points are well separated along PC1 axis, showing the significance of this factor. For Age, a gradient of color is used and one can observe the age increase along the PC1 axis. For Gender-Age interaction, a gradient of color is used for the Age and shape type

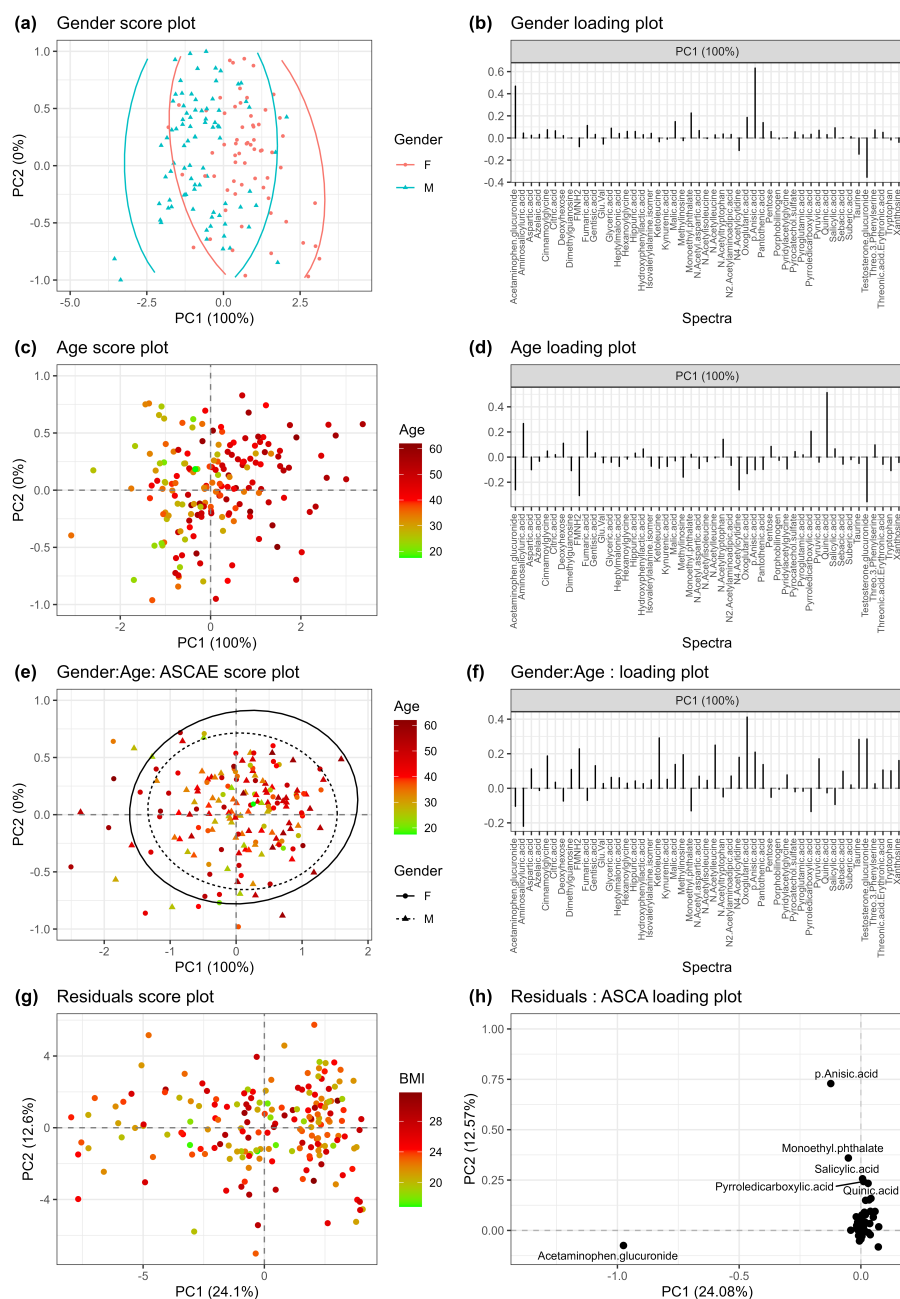
for the Gender. No clear Age gradient or groups of Gender can be observed, confirming the non-significance of this effect. The residuals are colored by BMI values, but do not show particular trend.

Figure 2.10(b)(d)(f)(h) present the loadings for Gender, Age, Gender-Age interaction effects and the residuals. The loadings can be particularly useful to link the metabolites with specific factors. In the case of Gender, acetaminophen glucuronide and p-anisic acid seem to be the most influential. In the case of Age, quinic acid and testosterone glucuronide have the highest loadings. For the Gender-Age interaction oxoglutaric acid is the most visible. Finally, for the residuals it is clearly acetaminophen glucuronide which is the most influential metabolite on PC1 and p-anisic acid on PC2, as seen on Figure 2.9(f).

2.3.5 Adaptation in `limpca`

To implement the extension of ASCA+ to continuous variables in the package `limpca`, some adaptations are necessary to the core modeling and visualization functions. For the modelisation part, the function `limpModelMatrix` needs to be adapted so that the sum coding gets applied on the categorical variables and continuous variables get normalized between $[-1, 1]$. The distinction between categorical and continuous variables is done automatically from the design object and `limpEffectMatrices` is applied directly on the outputs of `limpModelMatrix`. For the visualization part, a direct adaptation concerns functions `plotScatter`, `plotScatterM`, `pcaScorePlot`, `limpScorePlot` and `limpScoreScatterPlotM`. Those function can be enhanced to represent continuous variables via color gradients or point sizes, instead of discrete levels. Similarly, functions `plotMeans` and `limpEffectPlot` can allow continuous effects to be visualized through smooth response curves or gradient color scales.

In summary, the extension of the ASCA methodology to continuous variables offers a useful advancement to the package `limpca` since continuous factors or covariates are encountered quite often in experimental designs. Implementing those developments will enable the use of both categorical and continuous experimental designs in a fully integrated workflow.



2.4 Measurement of variable importance

2.4.1 Motivation

In the statistical analysis of high-dimensional data, a recurring challenge is to identify the variables that are most influenced by the factors of interest in the study. In ASCA/APCA, this is primarily done through the interpretation of the loading plots linked to effect matrices. This approach has two main limitations: (1) these plots are difficult to interpret when the number of variables is very large, and (2) some variables of interest do not always appear in the first PCA dimensions (which are most often examined). Users of the method would also like to have numerical indicators of variable importance, which are not provided in the basic ASCA/APCA methodology nor currently in `limpca`. This section proposes an approach based on two references: the master thesis of Nguipdop Lowe (2022) focusing on the analysis metabolomics and transcriptomics RNA-seq data which is partially inspired by the article of Nueda et al. (2007) focussing on genomics data. The methodology presented in this chapter has been implemented in the `metaboanalyst` platform (Pang et al., 2024) but is finally questionable and discussed here in Section 2.4.

2.4.2 Methodology

The aim of the approach is to compare methodologies to detect influential variables related to model effects combining univariate methods such as ANOVA and multivariate methods such as ASCA+. Indeed, those methods are often used separately while they can work in a complementary way. The methodology proposed in the work of Nguipdop Lowe (2022) is similar to the volcano plots used in -omics data analysis (especially in transcriptomics), which display on an X-Y graph the effect of, in general, a two-level factor on a large number of responses along two axes: fold change on the X-axis, measuring how much a variable's value differs between groups, and p-value on the Y-axis. In the context of ASCA, p-values are obtained from tests assessing the significance of model effects in univariate ANOVA or GLM models, and the fold change replaced by a measure of variable importance derived from the loading plot of the effect matrix of interest. Fold changes can indeed not be applied to this matrix since they may correspond to a factor with more than two levels, may be linked to a continuous variable, or represent an interaction between two or several factors.

First for the p-values computation, as a large number of response variables are involved, it is essential to control the type I error that arises from multiple

testing. Indeed, when multiple tests are performed simultaneously on a dataset, it leads to an increase in the type I error rate (i.e., the probability of rejecting the null hypothesis H_0 when it is actually true). This means that for a test with a fixed significance level $\alpha = 5\%$, applied to m responses in a dataset, the overall probability of making at least one false rejection is no longer 5%, but rather $1 - (1 - 0.05)^m$. As a result, this inflates the number of false positives in the analysis. In the univariate context, two standard correction procedures are commonly employed.

The first approach is the control of the Family-Wise Error Rate (FWER) using the Bonferroni correction Dunn (1961). This method adjusts each p-value as follows:

$$\text{p-val}_{\text{Bonf}} = \min(\text{p-val} \times m, 1)$$

where p-val is the original p-value and m is the total number of tests. A hypothesis is then considered significant if $\text{p-val}_{\text{Bonf}} < \alpha$. Although this method strictly controls the probability of any false positives, it is known to be overly conservative, especially in exploratory settings.

The second and more flexible approach is the control of the False Discovery Rate (FDR), introduced by Benjamini and Hochberg (1995). Its implementation is carried out in four steps:

- Sort the m initial p-values in ascending order;
- Multiply each p-value by $\frac{m}{i}$, where i is the rank of the p-value in the ordered list from the previous step;
- If (after this multiplication) the original order of the p-values is disrupted, then for every pair of p-values whose ranks were modified, the higher p-value is adjusted downward to match the lower one;
- Any adjusted p-value greater than 1 is set to 1.

The adjusted p-values are then computed as:

$$\text{p-val}_{\text{BH}} = \min\left(\frac{\text{p-val} \times m}{i}, 1\right)$$

where i is the rank of the original p-value after sorting in ascending order. FDR control is widely used in metabolomics and transcriptomics, where the primary aim is the identification of potential biomarkers rather than strict confirmation.

Nguipdop Lowe (2022) proposes then to build a bivariate plot on this basis. Each response $\mathbf{Y}_j$ is represented using:

- The vertical axis: $-\log_{10}(\text{adjusted p-value})$ from the univariate F-test;
- The horizontal axis: a multivariate importance score $V_{f,j}$ derived from the loadings of the Principal Component Analysis (PCA) performed on the effect matrix $\widehat{\mathbf{M}}_f$:

$$V_{f,j} = \sqrt{\sum_{t=1}^T (\mathbf{P}_{f(j,t)} \times \lambda_{f,t})^2}$$

where $\mathbf{P}_{f(j,t)}$ is the loading value (in ASCA) of a given variable on the t -th principal component, and $\lambda_{f,t}$ is its corresponding eigenvalue (explained variance). Only components with $\lambda > 1$ are retained.

This visualisation helps to assess whether the variables detected as significant in univariate testing are also influential in multivariate structures. This approach will be showed on Urine-Citrate-Hippurate dataset presented in Section 2.2.3 and Sacurine at Section 2.3.2.

2.4.3 Application on Urine-Citrate-Hippurate dataset

In this example, Urine-Citrate-Hippurate dataset is decomposed by ASCA to take into account only the effects detected previously as significant:

$$\text{outcomes} \sim \text{Hippurate} + \text{Citrate} + \text{Time} + \text{Time:Hippurate}$$

Figure 2.11(a)(b)(c)(d) presents the adjusted p-values vs weighted loadings for Hippurate, Citrate, Time and Hippurate:Time effects. The adjusted p-values are in $-\log_{10}$ scale and the red horizontal line corresponds to the threshold of 0.05. The annotated dots correspond to the most outlying metabolites in terms of adjusted p-values and weighted loadings. Among all the dots, a regression line is displayed in blue with its R^2 . The annotated dots can be put in relationship with loadings plots presented in Figure 2.6 and dataset presentation of Figure 2.3. For Hippurate, the region around 3.96, 7.54, 7.62, 7.82 ppm is observed. For Citrate, the region around 2.6 ppm is clearly detected for all dots which corresponds to a unique peak. For Time, the most outlying point is seen around 6 ppm. And for Hippurate:Time interaction, the shift around 3 is observed.

For this designed dataset, it is interesting to see the link between the variables detected in an univariate and multivariate way.

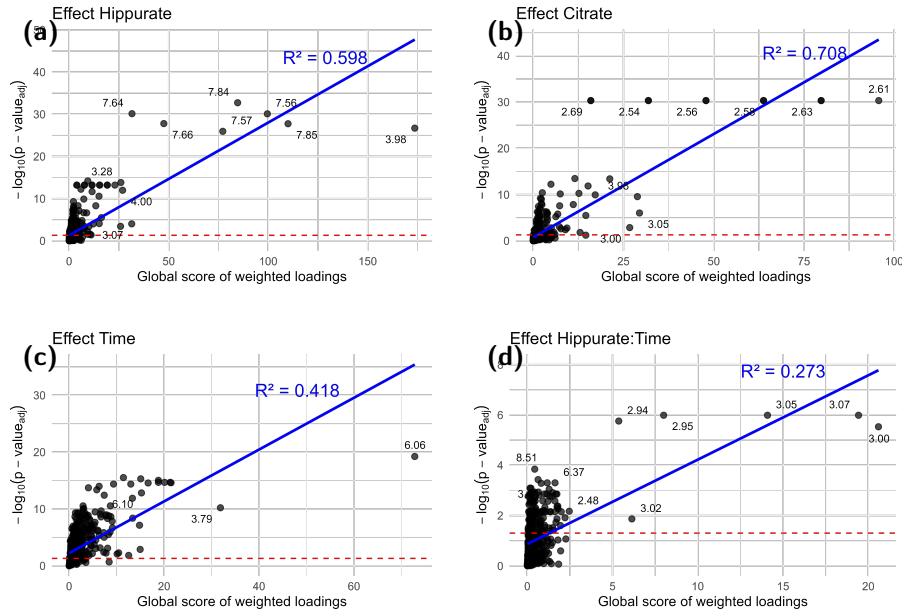


Figure 2.11: Adjusted p-values vs weighted loadings for effects Hippurate, Citrate, Time and Hippurate:Time in the UCH dataset.

Table 2.7 shows for each effect, the 10 most extreme variables (in ppm) ordered by values of weighted loadings. The p-values have been transformed back to visualize more easily their significance.

Table 2.7: Adjusted p-values and weighted loadings by variable for effects Hippurate, Citrate, Time and Hippurate:Time for the UCH dataset.

	Hippurate			Citrate			Time			Hippurate:Time		
	Response	V	p-value	Response	V	p-value	Response	V	p-value	Response	V	p-value
1	3.98	173.66	1.96e-27	2.61	95.71	4.76e-31	6.06	72.80	5.89e-20	3.00	20.60	2.97e-6
2	7.85	109.98	1.75e-28	2.63	79.76	4.76e-31	3.79	31.86	6.22e-11	3.07	19.43	1.03e-6
3	7.56	99.58	7.87e-31	2.59	79.76	4.76e-31	6.15	21.46	2.48e-15	3.05	14.09	1.03e-6
4	7.84	84.66	1.84e-33	2.64	63.81	4.76e-31	6.14	21.27	2.10e-15	2.95	8.00	1.03e-6
5	7.57	77.19	1.10e-26	2.58	63.81	4.76e-31	6.17	20.09	2.10e-15	3.02	6.14	1.36e-2
6	7.66	47.55	1.62e-28	2.56	47.85	4.76e-31	6.19	18.78	9.29e-16	2.94	5.37	1.75e-6
7	7.64	31.55	7.60e-31	2.56	47.85	4.76e-31	6.12	18.78	2.10e-15	2.48	2.46	6.71e-3
8	3.07	31.42	9.51e-5	2.67	31.90	4.76e-31	6.20	16.63	2.69e-15	6.37	1.63	5.00e-4
9	4.00	26.84	1.04e-12	2.54	31.90	4.76e-31	6.10	15.23	1.51e-13	3.15	1.17	5.00e-4
10	3.28	25.96	1.50e-14	2.69	15.95	4.76e-31	6.22	15.07	5.03e-16	8.51	0.44	1.43e-4

2.4.4 Application on Sacrine dataset

Sacrine dataset is decomposed in this example to take into account only the effects of Age and Gender which were detected previously as significant:

$$\text{outcomes} \sim \text{Age} + \text{Gender}$$

Figure 2.12(a)(b) presents the adjusted p-values vs weighted loadings for effects Gender and Age. For Gender the most outlying metabolites are p-anisic and acetaminophen glucuronide, but also testosterone glucuronide. For Age the most outlying metabolites are quinic acid and testosterone glucuronide. Interestingly, acetaminophen glucuronide which was observed previously is detected as extreme and does not have a significant p-value. This figure can be related to the loadings of Figure 2.10.

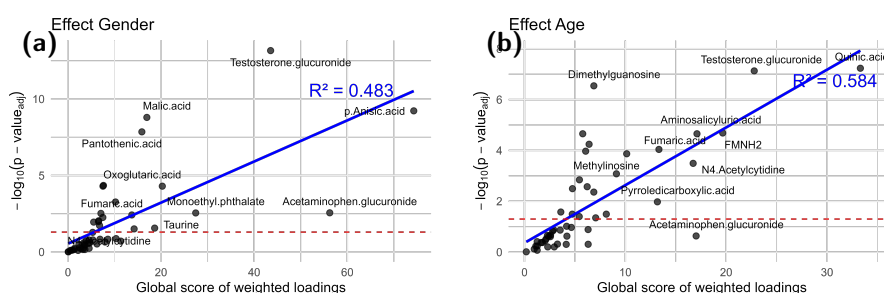


Figure 2.12: Adjusted p-values vs weighted loadings for effects Age and Gender.

Table 2.8 shows for each effect, the 10 most extreme variables (in ppm) ordered by values of weighted loadings. The p-values have been transformed back to visualize more easily their significance.

Table 2.8: Adjusted p-values and loadings for the effects Gender and Age.

Gender				Age			
	Response	V	p-value		Response	V	p-value
1	p.Anisic.acid	74.27	5.97e-10		Quinic.acid	33.31	5.87e-8
2	Acetaminophen.glucuronide	56.23	2.75e-3		Testosterone.glucuronide	22.78	7.49e-8
3	Testosterone.glucuronide	43.53	6.89e-14		FMNH2	19.66	2.07e-5
4	Monoethyl.phthalate	27.44	2.81e-3		Aminosalicyluric.acid	17.10	2.22e-5
5	Oxoglutaric.acid	20.27	5.01e-5		Acetaminophen.glucuronide	17.02	2.32e-1
6	Taurine	18.63	2.75e-2		N4.Acetylytidine	16.73	3.23e-4
7	Malic.acid	16.95	1.61e-9		Fumaric.acid	13.33	9.15e-5
8	Pantothenic.acid	15.88	1.41e-8		Pyrroledicarboxylic.acid	13.18	1.05e-2
9	N4.Acetylytidine	14.22	3.11e-2		Methylinosine	10.14	1.36e-4
10	Fumaric.acid	13.72	3.81e-3		Dimethylguanosine	6.88	2.89e-7

2.4.5 Methodology in (Nueda et al., 2007)

The article of Nueda et al. (2007) proposes another methodology which will be briefly explained here. The GLM decomposition in ASCA produces several matrices $\hat{\mathbf{M}}_f$ as explained in equation:

$$\mathbf{Y} = \hat{\mathbf{M}}_0 + \sum_{f=1}^F \hat{\mathbf{M}}_f + \hat{\mathbf{E}}$$

As seen previously, each effect matrix can be decomposed into scores and loadings by PCA:

$$\hat{\mathbf{M}}_f = \mathbf{T}_f \mathbf{P}_f' = \tilde{\mathbf{T}}_f \tilde{\mathbf{P}}_f' + \mathbf{E}_f$$

where $\tilde{\mathbf{T}}_f$ and $\tilde{\mathbf{P}}_f$ are $N \times R$ and $m \times R$ matrices of $\mathbf{T}_f$ and $\mathbf{P}_f$ if not all the PCs are retained in the decomposition.

The residual matrix $\mathbf{E}_f$ is null if the scores and loadings use all the principal components, otherwise non-null.

The method proposes then to represent the importance of variables on an X-Y plot based on the method proposed by Martens and Næs (1992) in simple PCA. On the x-axis, the leverages are represented and measure the contribution of each response to the multivariate structure captured by the PCA on the submodel. The leverages can be written explicitly for each response Y_j and effect f as:

$$h_{f,j} = \sum_{t=1}^R \mathbf{P}_{f(j,t)}^2$$

where $\mathbf{P}_{f(j,t)}$ is the loading of response j on principal component t .

Each leverage value $h_{f,j}$ represents the non-weighted squared length of the loading vector of variable m across the retained components. Variables with higher leverage values are those that contribute most strongly to the structure identified by the PCA in that submodel.

The method proposes also to represent on the Y axis the Squared Prediction Error (SPE). The SPE measures the part of the variability that remains unexplained by the PCA approximation. For a given response j , the SPE is computed as:

$$\text{SPE}_{f,j} = \sum_{i=1}^N \mathbf{E}_{f(i,j)}^2$$

Each value $\text{SPE}_{f,j}$ corresponds to the sum of squared residuals for response j across its N observations, representing the variance not captured by the PCA model.

The leverages are plotted against SPEs and the article states that:

- Variables with **high leverage and low SPE** are well modeled and influential in defining the multivariate structure;
- Variables with **high SPE** are poorly explained by the PCA model, and may represent outliers or unique profiles;
- Variables with **low leverage and low SPE** are well fitted but not particularly influential;
- Variables with **low leverage and high SPE** are both unimportant and poorly modeled.

A typical plot of leverages against SPEs from metaboanalyst platform is shown in Figure 2.13.

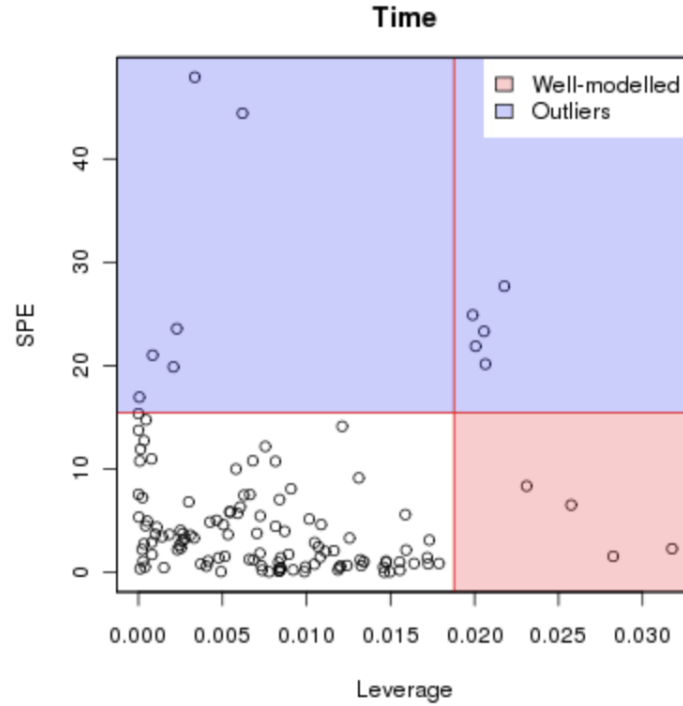


Figure 2.13: Typical leverages vs SPEs plot from metaboanalyst platform.

This methodology is shown in Figure 2.14 on the UCH dataset where some aspects are questionable. For example the methodology does not consider the residuals matrix $\hat{\mathbf{E}}$ of the GLM model, which is the main source of unexplained variability. It uses instead a submodel residual matrix $\mathbf{E}_f$, but does not specify the number of principal components to use. Another point is that the computation of the leverages is non-weighted and does not use the eigenvalues for the weighting as in the case of Nguipdop Lowe (2022). In the following application, all principal components were used. Theoretically, the submodel residual matrix $\mathbf{E}_f$ should be null and the SPEs equal to 0, making the plot uninterpretable. Surprisingly, the computations do not provide $\mathbf{E}_f$ matrices completely equal to 0. As a result, some higher SPEs are visible for the subplots of Hippurate, Citrate, Time and Hippurate:Time. Globally the R^2 does not indicate any link among the SPEs and leverages. However, the metabolites with higher SPEs are the same as detected as influential in Figure 2.11. This

unexpected behavior can most probably be attributed to numerical precision issues in the PCA decomposition, where tiny reconstruction errors are amplified in absolute terms for variables with higher signal magnitudes, such as the influential peaks for Hippurate and Citrate. Consequently, these variables exhibit relatively higher SPE despite the theoretical expectation of zero residuals.

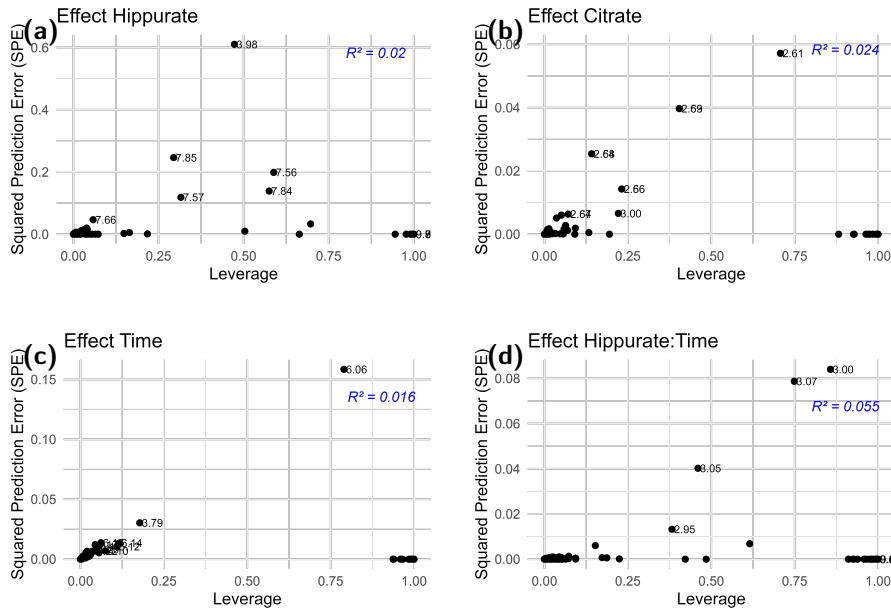


Figure 2.14: SPEs vs leverages for effects Hippurate, Citrate, Time and Hippurate:Time.

2.4.6 Adaptation in `limpca`

The implementation of the methodology for variable importance in the package `limpca` provides a useful tool for identifying significant response variables associated with each model effect. Its integration relies on univariate tests applied to every response j of the outcomes matrix $\mathbf{Y}$. Those univariate tests can be automated in a specific function using the object formula for the GLM decomposition by `limpEffectMatrices()`. The p-values can then be stored as an object after the correction of multiplicity. For each matrix $\hat{\mathbf{M}}_f$, the function `limpPcaEffects()` generates the scores and loading matrices. With those elements, a graphical function displaying p-values versus weighted loadings can be created and used complementarily with `limpLoading1dPlot()` and

`ImpLoading2dPlot()`. Adding this methodology can help to explore with more details which variables drive the multivariate patterns.

2.5 Confidence ellipsoids

2.5.1 Motivation

In ASCA methodology, a primary objective is to perform statistical inference on the parameter vector θ (or its estimate $\hat{\theta}$) to assess the significance of design factors. Traditional data ellipses, centered on sample means and reflecting observed variability in the data space (or score plots of effect matrices) (Friendly, Monette, and Fox, 2013), are not suited for this purpose, as they do not account for the sampling distribution of $\hat{\theta}$ and fail to determine if response means differ significantly between factor levels. In contrast, confidence ellipses enable direct inference on linear combinations of $\hat{\theta}$ in the projected PCA space. Building on Liland et al., 2018, this approach enhances visualization of group separations while providing a rigorous tool for hypothesis testing in high-dimensional datasets. The following subsections summarize the methodology and demonstrate its application in the framework of the package `limpca`.

2.5.2 Methodology

Let's suppose that one is interested by an estimated effect matrix $\hat{\mathbf{M}}_f = \mathbf{X}_f \hat{\boldsymbol{\Theta}}_f$ decomposed by PCA in $\hat{\mathbf{M}}_f = \mathbf{T}_f \mathbf{P}_f'$. $\mathbf{T}_f$ is of dimension $(N \times r_f)$ and $\mathbf{P}_f$ of dimension $(m \times r_f)$, with r_f typically equals to 2 or 3 according to degrees of freedom of an effect f . In the space of principal components of this effect matrix, the goal is then to draw an ellipse for each level of the factor(s) related to $\hat{\mathbf{M}}_f$. Three elements are needed to draw it: the estimated mean, its variance-covariance matrix and an adapted scale factor to define the size of the ellipse.

To derive these, some new statistical hypotheses are taken by Liland et al., 2018 and some linked multivariate statistical results must be reminded. Let's remember the multivariate regression model:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\Theta} + \mathbf{E} \quad (2.1)$$

and note $\mathbf{e}_{i*}$ the i -th line of $\mathbf{E}$. Until now, an independence hypothesis was taken between $\mathbf{e}_{i*}$ vectors (observations or samples) but no specific hypothesis was necessary on the distribution and variance-covariance matrix of each $\mathbf{e}_{i*}$ since simple OLS was used for parameter estimation. In this section, we keep the hypothesis that the error vectors $\mathbf{e}_{i*}$ (i.e., the rows of the error matrix $\mathbf{E}$) are independent and add the fact that they are each identically distributed as a multivariate normal $\mathcal{N}_m(\mathbf{0}, \boldsymbol{\Sigma})$.

From this, it can be shown (see Liland et al. (2018)) that the sampling distribution of a linear combination of OLS estimators is also normally distributed:

$$(\mathbf{l}'\hat{\boldsymbol{\Theta}})' \sim \mathcal{N}_m((\mathbf{l}'\boldsymbol{\Theta})', a\boldsymbol{\Sigma}) \quad (2.2)$$

where $a = \mathbf{l}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{l}$.

Here, $\mathbf{l}$ is a contrast vector of dimension $p \times 1$, which specifies the linear combination of the model parameters $\boldsymbol{\Theta}$ (or their estimates $\hat{\boldsymbol{\Theta}}$) corresponding to a particular effect or factor level of interest. For the UCH dataset in Section 2.5.3, $\mathbf{L}$ is a 3×8 matrix defining the three contrasts vectors for the levels of Hippurate (0, 1, 2) as

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

This $\mathbf{L}$ allows to compute effect-specific mean vectors, centered in the PCA space of $\hat{\mathbf{M}}_f = \mathbf{T}_f\mathbf{P}_f'$.

Moreover, a maximum likelihood estimator of $\boldsymbol{\Sigma}$ is given by $\hat{\boldsymbol{\Sigma}} = \mathbf{S} = \frac{1}{N}\mathbf{E}'\mathbf{E}$ distributed as a Wishart distribution $N\hat{\boldsymbol{\Sigma}} \sim \mathcal{W}_m(\boldsymbol{\Sigma}, N-p)$.

The combination of those results allow to derive the equation of a confidence ellipsoid for $\mathbf{l}'\boldsymbol{\Theta}$ around $\mathbf{l}'\hat{\boldsymbol{\Theta}}$ as:

$$\frac{(N-p)}{aN}(\mathbf{l}'\hat{\boldsymbol{\Theta}} - \mathbf{l}'\boldsymbol{\Theta})\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{l}'\hat{\boldsymbol{\Theta}} - \mathbf{l}'\boldsymbol{\Theta})' = \mathcal{T}_{m, N-p, 1-\alpha}^2 \quad (2.3)$$

where $\mathcal{T}_{m, N-p, 1-\alpha}^2$ is the $1 - \alpha$ percentile of a $\mathcal{T}_{m, N-p}^2$ Hotelling distribution.

Since effect matrices are indeed linear combinations of $\hat{\boldsymbol{\Theta}}$ (see Equation 1.22) and the score plots simple projections in the space of the principal components of the effect matrix, this formula can be used to obtain the ellipsoid of interest in the projected space:

$$\frac{(N-p)}{aN}(\mathbf{l}'\hat{\boldsymbol{\Theta}}\mathbf{P}_f - \mathbf{l}'\boldsymbol{\Theta}\mathbf{P}_f)(\mathbf{P}_f'\hat{\boldsymbol{\Sigma}}\mathbf{P}_f)^{-1}(\mathbf{l}'\hat{\boldsymbol{\Theta}}\mathbf{P}_f - \mathbf{l}'\boldsymbol{\Theta}\mathbf{P}_f)' = \mathcal{T}_{r_f, N-p, 1-\alpha}^2 \quad (2.4)$$

where a value of the contrast vector $\mathbf{l}$ of dimension $(p \times 1)$ must be defined for each factor level of interest (e.g., $(0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$ for the lowest level of Hippurate).

Note finally that Hotelling percentiles are commonly derived from an F table and the relation is given in this case by:

$$\mathcal{T}_{r_f, N-p, 1-\alpha}^2 = \frac{(N-p)r_f}{N-p-r_f+1} \mathcal{F}_{r_f, N-p-r_f+1, 1-\alpha} \quad (2.5)$$

In the UCH Hippurate example ($N = 34$, $p = 8$, $r_f = 2$, $\alpha = 0.05$), this yields $\mathcal{T}^2 \approx 7.04$ (using $\mathcal{F}_{2,25,0.95} \approx 3.40$) and scales the ellipse to reflect sampling uncertainty around projected means from $\mathbf{L}\hat{\Theta}\mathbf{P}_f$. This methodology is applied below on the Urine-Citrate-Hippurate and Sacurine datasets.

2.5.3 Application on Urine-Citrate-Hippurate dataset

In this section, the Urine-Citrate-Hippurate dataset is decomposed by ASCA taking only into account the significant effects detected previously:

$$\text{outcomes} \sim \text{Hippurate} + \text{Citrate} + \text{Time} + \text{Time:Hippurate}$$

Figure 2.15 shows the ASCA-E score plots on the first two PCs for the significant effects of the model: Hippurate, Citrate, Time and Time:Hippurate interaction. The left column displays the data ellipses and the right one the confidence ellipses. The first observation is that confidence ellipses are smaller than the data ones. Indeed they depend on the variability of the estimators of the multivariate model instead of separate groups of points. Their goal is to determine visually if levels of a factor are significantly different. For this objective, the visual inspection of effects Hippurate, Citrate and Time is the same for data and confidence ellipses. For Time:Hippurate interaction one can see a difference and a clear advantage of confidence ellipses. Indeed, the data ellipses are crossing each other, making it difficult to distinguish a separation in the levels. On the other hand, confidence ellipses are well separated, confirming the significant effect of Time:Hippurate interaction presented in Table 2.1. Note that for Time effect, PC1 explains 100% of the variance because Time factor has only two levels and the second dimension is then meaningless. The plot could also be reduced to a single dimension.

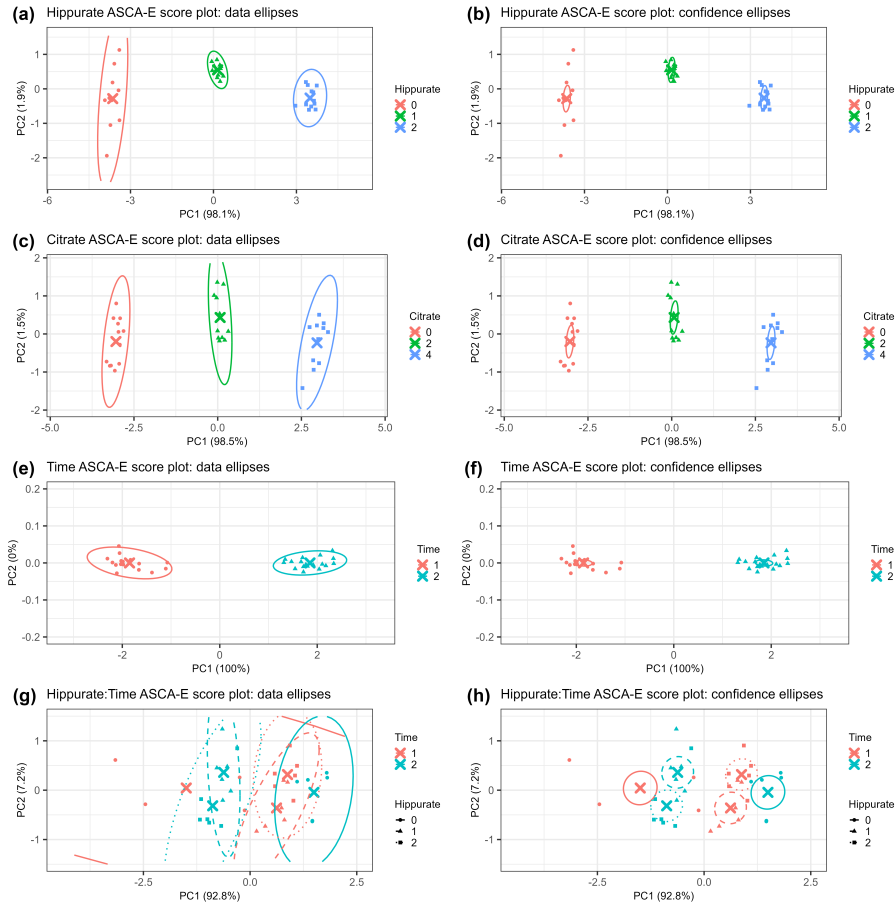


Figure 2.15: Data and confidence ellipses for score plots in ASCA-E for significant effects of Urine-Citrate-Hippurate dataset.

2.5.4 Application on Sacurine dataset

In this example, Sacurine dataset is decomposed by ASCA taking into account only the effects of Age and Gender which were detected previously as significant:

$$\text{outcomes} \sim \text{Age} + \text{Gender}$$

Age factor is categorized in 4 groups as already presented in Table 2.6.

Figure 2.16 shows the score plots on the first two PCs for Age and Gender effects. The left column displays the data ellipses and the right one the

confidence ellipses. The first observation is that for data ellipses cross each other among the group levels. This makes it not straightforward for the user to understand why the factors Age and Gender were detected as significant in Table 2.6. On the opposite, confidence ellipses show clearly the separation of the groups and the significance of Age and Gender factors. This is particularly useful for factor Age where the confidence ellipses of the four levels are clearly separated.

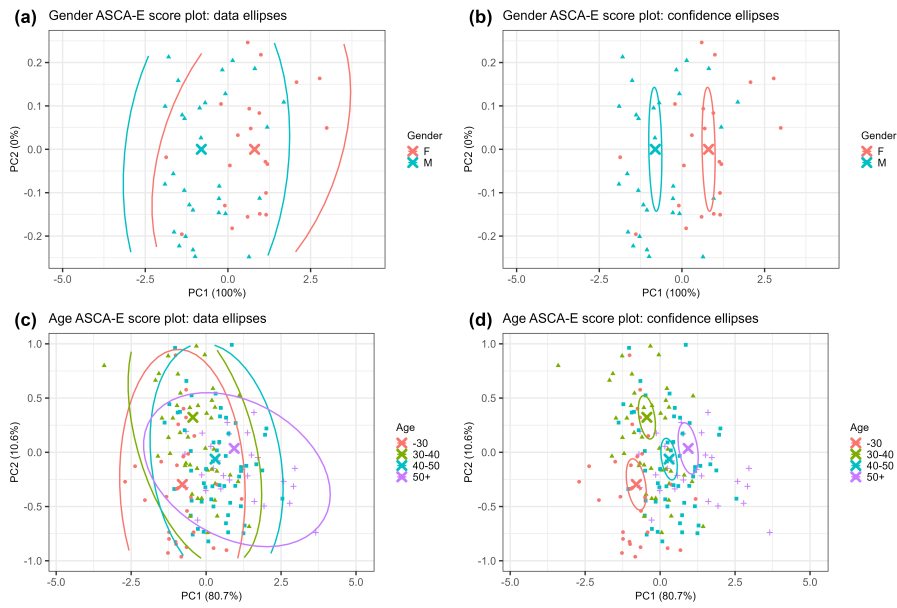


Figure 2.16: Data and confidence ellipses for score plots in ASCA-E for significant effects of categorical Sacurine dataset.

2.5.5 Pairwise comparisons of effect level means

Building on the confidence ellipsoids derived from Equation 2.4, pairwise comparisons of effect level means can be performed to determine which specific levels differ significantly, extending the approach of Liland et al., 2018.

For a pair of levels $k1$ and $k2$ in effect f , define the null hypothesis $H_0 : \mathbf{l}'\boldsymbol{\Theta} = 0$, where $\mathbf{l}$ is a contrast vector such that $\mathbf{l}'\boldsymbol{\Theta} = \theta_{k2} - \theta_{k1}$. The multivariate test statistic in the projected PCA space is then:

$$T^2 = \frac{(N-p)}{aN} \left(\mathbf{l}'\hat{\boldsymbol{\Theta}}\mathbf{P}_f \right) \left(\mathbf{P}_f' \hat{\boldsymbol{\Sigma}} \mathbf{P}_f \right)^{-1} \left(\mathbf{l}'\hat{\boldsymbol{\Theta}}\mathbf{P}_f \right)'.$$

Under H_0 , $T^2 \sim \mathcal{T}_{r_f, N-p}^2$ and the hypothesis can be rejected if $T^2 > \mathcal{T}_{r_f, N-p, 1-\alpha'}^2$, where α' is adjusted for multiple comparisons (e.g., Bonferroni: $\alpha' = \alpha / \frac{k(k-1)}{2}$ for k levels). Equivalently, the statistic can be converted to an F -statistic for p-value computation:

$$F = \frac{(N - p - r_f + 1)}{(N - p)r_f} T^2 \sim F_{r_f, N-p-r_f+1}.$$

This methodology will be applied to the main effects of the datasets in Sections 2.5.3 (Urine-Citrate-Hippurate) and 2.5.4 (Sacurine). As an example, consider the factor Hippurate in the UCH dataset, which has three levels (0, 1, 2). Assuming sum coding, the contrast matrix for the parameters Θ_f (dimension $2 \times m$) is such that the contrasts for pairwise differences are:

- Level 1 vs. 2: $I' = (0 \ 1 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0)$ (row 2 - row 3 of $\mathbf{L}$),
- Level 1 vs. 0: $I' = (0 \ -1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0)$ (row 2 - row 1 of $\mathbf{L}$),
- Level 2 vs. 0: $I' = (0 \ -2 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0)$ (row 3 - row 1 of $\mathbf{L}$).

These contrasts enable testing each pair using the T^2 statistic above.

The pairwise comparisons were applied to the significant main effects of the UCH dataset: Hippurate, Citrate, and Time. Table 2.9 summarizes the results, including the T^2 statistic, F -statistic, raw p-values, Bonferroni-adjusted p-values, and significance decisions at $\alpha = 0.05$. All pairs for each factor show highly significant differences (p-values = 0 after adjustment), consistent with the non-overlapping confidence ellipses observed in Figure 2.15. For Hippurate and Citrate, the largest T^2 values correspond to pairs with the greatest separation in the projected space, reflecting the strong multivariate effects (39.31% and 29.91% of explained variance, respectively). The Time factor, with only two levels, yields a single comparison with a substantial T^2 (761), aligning with its 16.24% explained variance and clear group separation. Note that for Hippurate effect, the largest T^2 is for the pair of levels “2 vs 0” and for Citrate effect the pair of levels “4 vs 0” which is aligned with Figure 2.15.

For the Sacurine dataset, pairwise comparisons were conducted on the significant main effects: Gender and Age (as categorical). Table 2.10 presents the results. The Gender factor, being binary, shows a significant difference between M and F ($T^2 = 104$, adjusted p-value = 0), corroborating the separation in the confidence ellipses. For Age with four levels, all six pairs are significant after Bonferroni adjustment, with T^2 values increasing for pairs involving more distant age groups (e.g., -30 vs 50+ has the highest $T^2 = 84$). This pattern reflects the gradual shifts in metabolic profiles across age

Table 2.9: Pairwise comparison results for main effects in the UCH dataset.

Factor	Pair	T^2	F	Raw p-value	Bonferroni p-value	Significance
Hippurate	1 vs 2	1939	932	0	0	Significant
	1 vs 0	2047	984	0	0	Significant
	2 vs 0	7465	3589	0	0	Significant
Citrate	2 vs 4	1649	793	0	0	Significant
	2 vs 0	1829	880	0	0	Significant
	4 vs 0	7044	3387	0	0	Significant
Time	1 vs 2	761	366	0	0	Significant

categories, as visualized in Figure 2.16, where confidence ellipses are well-separated, emphasizing the methodology's ability to detect nuanced differences in high-dimensional data.

Table 2.10: Pairwise comparison results for main effects in the Sacurine dataset.

Factor	Pair	T^2	F	Raw p-value	Bonferroni p-value	Significance
Gender	M vs F	104	52	0	0	Significant
Age	-30 vs 30-40	31	15	8.03e-07	4.82e-06	Significant
	-30 vs 40-50	45	22	2.02e-09	1.21e-08	Significant
	-30 vs 50+	84	42	1.55e-15	9.33e-15	Significant
	30-40 vs 40-50	21	11	4.65e-05	2.79e-04	Significant
	30-40 vs 50+	45	22	2.26e-09	1.36e-08	Significant
	40-50 vs 50+	14	7	1.20e-03	7.20e-03	Significant

2.5.6 Adaptation in `limpca`

The implementation of confidence ellipses in `limpca` depends on a few elements. The objects to compute the ellipses are provided after running the functions `limpModelMatrix` for the model matrix creation, `limpEffectMatrices` for estimating the model effects and `limpPcaEffects` for performing PCA. The function to compute the ellipses can be automated with the level of confidence as an argument into `limpScorePlot` as is the case for regular data ellipses. Similarly, pairwise comparisons of effect level means can be integrated into `limpca` by extending the ellipse computation to hypothesis tests via the T^2 statistic, with options for multiple comparison adjustments like Bonferroni. This would leverage the same model objects to output tables of p-values and significance decisions, allowing users to identify specific level differences directly within the package workflow. Such functionality would complement the visual ellipsoids, providing a tool for rapid multivariate inference.

2.6 Conclusion and perspectives

This chapter introduced the package `limpca` available on Bioconductor. It showed the ASCA+ methodology and how to implement the different functions on a practical case-study. Currently, the package supports fixed-effects models with categorical factors. However, the extension to continuous factors has been developed (as demonstrated in this chapter) and is ready for implementation, relying on automated contrast coding and generalized model handling; it will be integrated into future versions of `limpca`. The direct challenges are the implementation of graphical outputs and how to display them intuitively for the users. Methodological additions developed in this chapter include also the implementation of measures of variable importance, and the use of confidence ellipsoids. Additionally, pairwise comparisons of factor levels, derived from the confidence ellipsoid framework, offer a statistical tool to pinpoint specific differences in high-dimensional data, further bridging univariate and multivariate analyses. While ASCA and APCA inherently provide a bridge from univariate to multivariate analysis through the integration of PCA on effect matrices, the addition of these functionalities could further enhance users' ability to connect univariate significance tests with multivariate patterns and visually assess the significance of factor levels.

Beyond the current features, `limpca` is designed to support further developments. An important extension would be the integration of random effects, which would enable the analysis of longitudinal, hierarchical, or repeatability studies through mixed-effects models. This aspect was explored in the context of a master's thesis by Visschers (2024), who proposed modifications to the package to handle random effects by integrating the LiMM-PCA methodology (Martin and Govaerts, 2020). Ultimately, `limpca` aims to be a practical toolbox for multivariate analysis in designed experiments and it remains adaptable to emerging needs in omics applications.

CHAPTER 3

Unsupervised chemometric approaches for monitoring
chemical reactions: PCA, NMF and MCR

3.1 Introduction

The quality attributes of active pharmaceutical ingredients (APIs) are critical in the control of pharmaceutical production. Monitoring techniques can be classified as off-line, at-line, and on-line, according to the location of the analytical system in relation to the reactor. Off-line measurements include manual or automatic sampling, transfer to the laboratory, and time-delayed analysis in the laboratory. At-line measurements also involve manual or automatic sampling, but the removed sample is analyzed in close proximity to the reactor. For on-line measurements, the reaction is analyzed in real-time by a sensor directly from the reactor (Lourenço et al., 2012).

Off-line techniques such as Ultra-Performance Liquid Chromatography (UPLC) can achieve accurate and precise quantification but their sampling rate is lower and not efficient to enable real-time knowledge of the process performance (Lourenço et al., 2012).

Effective real-time process understanding can be achieved by process analytical technology (PAT). PAT is a system for designing, analyzing, and controlling manufacturing through timely measurements of critical quality and performance attributes with the goal of ensuring final product quality (International Conference on Harmonisation (ICH), 2009). In this context, PAT methods such as spectroscopy can be very useful in process monitoring and control, and offer a time-saving alternative to off-line techniques/classical methods (Lapointe-Garant, Simard, and Abatzoglou, 2008). Spectroscopic methods generate however large datasets with many correlated variables necessitating the use of dimension-reduction methods to extract the meaningful information and correctly monitor the process (De Beer et al., 2011).

This chapter presents different chemometrics strategies to monitor chemical reactions under a uniform framework. The development of those strategies originates from concrete projects at Janssen Pharmaceutica, where multiple chemical reactions were performed in batches to synthesize an active pharmaceutical ingredient. The chemical reactions were monitored on-line and off-line in a research and development context as an initial step towards industrial production.

This framework aims to answer two objectives:

The first objective is to visualize and interpret chemical reactions in real time or retrospectively using dimension reduction methods. Those methods can be applied individually to each experiment, globally to compare multiple experiments on a same dimension space, or under a leave-one-batch-out approach to visualize a new experiment using a model built on historical batches.

The second objective is to predict chemical concentrations over time using calibration methods. As for dimension reduction methods, calibration methods can be applied to predict concentration individually to each experiment, globally, or under a leave-one-batch-out approach to predict concentrations of a new batch using a calibration set. This batch-based framework can also extend beyond pharmaceutical reactions to other domains with grouped data, such as animal studies or human clinical trials, leveraging the same dimension-reduction and calibration strategies.

The methods developed in this chapter not only serve as an analytical exercise but provide practical tools that enable researchers in pharmaceutical R&D and production to monitor reactions in real-time, facilitating early detection of anomalies and optimizing process efficiency under PAT and QbD frameworks. The methodologies presented in this chapter build on prior work published in Thiel et al. (2021), where NMF and MCR were applied to similar datasets, demonstrating their efficacy in pharmaceutical monitoring.

The following sections will first describe the dataset used in this chapter. Secondly, a methodological section will explain the theory behind dimension-reduction methods PCA, NMF and MCR and how to apply them on on-line spectral data organized in batches. The calibration method PLS will then be shortly reviewed and applied to predict off-line chemical concentrations. An original approach in two steps, combining multivariate regression with PCA, NMF and MCR scores will also be presented. This two-step approach corresponds to Principal Component Regression (PCR) when using PCA scores, and analogous methods for NMF and MCR. For both objectives, performance metrics are presented to evaluate the quality of the fit and predictions. Finally, those different methods will be illustrated on the case study.

This chapter and Chapter 4 should be seen as a uniform block. Indeed, Chapter 4 is based on the research presented here and consequently, similar notations will be used. Those notations can be found in the appendix of this thesis (see Chapter 6). Note also that notations in the next two chapters can be quite different from the ones in Chapter 1 and 2. Chapter 4 will extend this by introducing a supervised NMF variant, demonstrated on an artificial dataset for controlled evaluation of method performance under known conditions, complementing the real-world pharmaceutical data analyzed here.

3.2 PAT data structure

To motivate the goal of the next sections, it is important to explain the type of data addressed by this chapter. The data correspond to multiple experiments performed in batches over several hours and monitored by on-line and off-line methods. On-line methods provide continuous but complex high dimensional signals about the process whereas off-line methods provide accurate measurements of the reagents and products concentrations but their sampling rate is low. The dataset presented in this chapter should be seen as a typical example of such data set and the framework presented here can be applied on other types of data presenting a similar structure.

3.2.1 Dataset

The experiments have been performed at Janssen Pharmaceutica in a research and development context to synthesize an active pharmaceutical ingredient. All experiments aimed to produce an active molecule with the same reaction but the process parameters (e.g. temperature, solvent, catalyzer...) were varied in order to understand their impact on the reaction and to select in future steps the optimal production conditions. Eleven experiments were performed in batches monitored over 21 hours on-line by MIR spectroscopy and off-line by UPLC. Since MIR provides absorbance spectra with positive intensities, a baseline correction method was applied that preserves non-negativity, ensuring no negative values arise from baseline drift or noise subtraction. No other preprocessing steps, such as derivatives (which would introduce negative values), were employed. Indeed, keeping non-negativity is essential for non-negative decomposition methods like NMF and MCR used in this chapter. MIR and UPLC data can be organized in two different matrices:

- A spectral matrix $\mathbf{X}_G$ of dimensions $T \times m$ consisting of individual spectral matrices $\mathbf{X}_i$ stacked vertically with $T = \sum_{i=1}^{11} t_i = 6510$ and $m = 221$. Matrix $\mathbf{X}_G$ corresponds to all the 6510 spectra of the 11 experiments put in rows, while the 221 columns correspond to the spectral variables. The spectral range was between 802 and 1622 nanometers. Table 3.1 shows for each experiment the time of monitoring and the number of spectra recorded.

Table 3.1: Summary of spectral measurements per experiment.

Experiment	Time of monitoring (in hours)	Number of spectra
1	21	420
2	21	420
3	21	630
4	21	630
5	21	630
6	21	630
7	21	630
8	21	630
9	21	630
10	21	630
11	21	630

- An UPLC matrix $\mathbf{Y}_G^*$ of dimensions $S \times n$ consisting of individual matrices $\mathbf{Y}_i^*$ stacked vertically where $S = \sum_{i=1}^{11} s_i = 44$, $s_i = 4$ and $n = 2$.

Note then that, for each of the eleven experiments, samples have been taken at only four time points (0, 2, 4 and 21 hours), and, for each sample, the concentrations of three compounds corresponding to the starting material, the final product and a side product were quantified. The side product was present in very low concentration and will therefore not be used in data analysis of this chapter. The number of compounds n will then be considered equal to 2.

Two other matrices $\mathbf{X}_G^*$ of dimensions $S \times m$ and $\mathbf{X}_i^*$ of dimensions $s_i \times m$ can also be respectively defined as the subsets of $\mathbf{X}_G$ and $\mathbf{X}_i$ with the spectra corresponding to off-line sampling times. Offline samples performed by UPLC rely indeed on extraction and chromatographic separation which can take several minutes per sample. On the other hand, MIR spectroscopy offers rapid analysis due to the direct measurement of molecular vibrations without the need for separation processes.

The structure of the matrices $\mathbf{X}_G$ and $\mathbf{Y}_G^*$ can be visualized in Figure 3.1.

3.3 Methods

This section first describes the theory behind dimension-reduction methods PCA, NMF and MCR. It then explains how these methods can be applied on PAT data organized in batches, using three modeling strategies: (1) individual

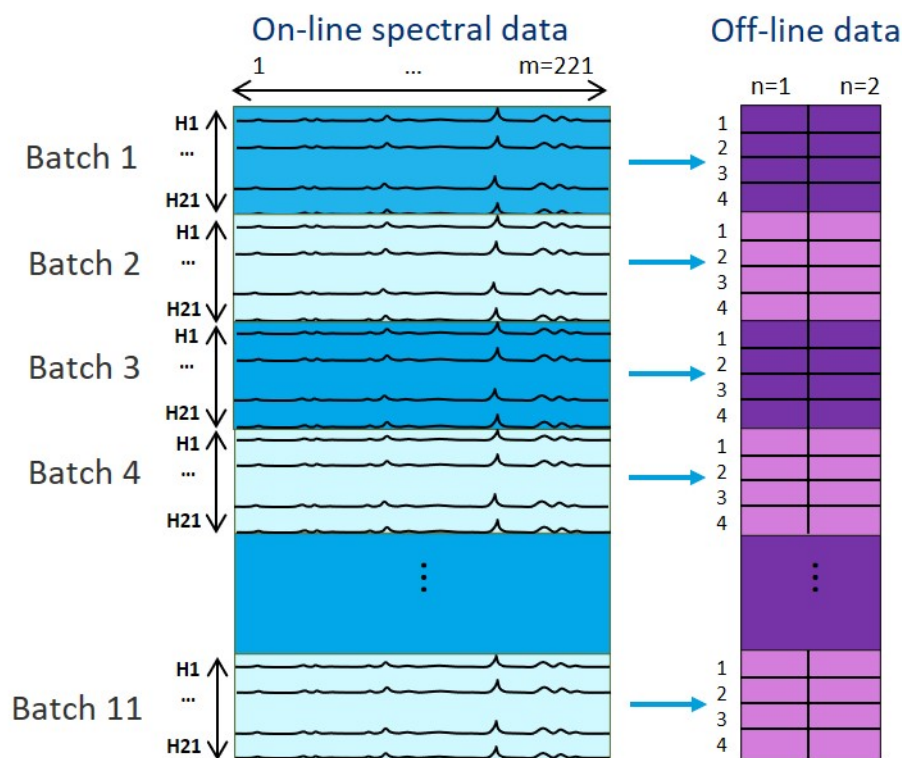


Figure 3.1: Representation of on-line and off-line data.

models fitted separately to each batch for isolated analysis; (2) global models that combine all batches into a single dataset for unified decomposition and comparison; and (3) an original leave-one-batch-out approach introduced in this chapter, which trains a model on historical batches (as a calibration set) and projects a new batch onto this model for real-time monitoring or prediction. For each strategy, metrics are proposed to assess the model's quality of fit, such as explained variance or reconstruction error.

Next, the theory of PLS regression is briefly reviewed, along with its implementation in a one-step approach to directly predict off-line chemical concentrations from spectral data. While PLS can act as a benchmark in comparative studies due to its predictive power, its primary role is as a calibration method, building a direct relationship between high-dimensional spectral data (predictors) and quantitative reference values (responses from UPLC). Additionally, a two-step approach is presented, which first extracts

scores from PCA, NMF, or MCR decompositions and then applies multivariate regression on these scores to predict concentrations. For both the one-step and two-step approaches, performance metrics such as Root Mean Square Error of Cross-Validation (RMSECV) are introduced to evaluate and compare prediction accuracy.

3.3.1 PCA, NMF and MCR dimension-reduction methods

This section summarizes under a common framework the 3 dimension reduction techniques PCA, NMF and MCR with the perspective of applying them to PAT data.

Let's assume a generic data matrix named $\mathbf{X}$ with dimensions $T \times m$ where T denotes the number of rows and m represents the number of columns. Put differently, $\mathbf{X}$ contains T observations with each observation being described by m variables. This matrix is named with the generic notation $\mathbf{X}$ but could correspond to matrices $\mathbf{X}_G$ or $\mathbf{X}_i$ described in the previous section.

The aim of the different dimension-reduction methods is to decompose the matrix $\mathbf{X}$ into the product of two factor matrices $\mathbf{H}$ of dimensions $T \times R$ and $\mathbf{W}$ of dimensions $m \times R$, providing a low-dimension (number of components R) approximation such that $R \leq \min(T, m)$:

$$\mathbf{X} \approx \mathbf{H}\mathbf{W}'$$

The matrix $\mathbf{X}$ can then be expressed as the sum of $\mathbf{H}\mathbf{W}' = \mathbf{X}_{app}$ and the error of the model $\mathbf{E}$ of dimensions $T \times m$:

$$\mathbf{X} = \mathbf{H}\mathbf{W}' + \mathbf{E} = \mathbf{X}_{app} + \mathbf{E}$$

Scores $\mathbf{H}$ and loadings $\mathbf{W}$ matrices represent rows (observations) and columns (variables) into the space of reduced dimensions R . In this context, scores will help to follow reaction components over time while loadings will help to detect influential spectroscopic variables. The components (columns of $\mathbf{H}$) are new variables constructed as linear combinations of the initial variables, with the loadings $\mathbf{W}$ defining the coefficients of these combinations.

The following notations are used throughout all the chapter to simplify method comparison:

$\mathbf{X}$	$T \times m$	data matrix
$\mathbf{h}_r$	$T \times 1$	$\mathbf{X}$ scores vector r ($r : 1, \dots, R$)
$\mathbf{H}$	$T \times R$	matrix of $\mathbf{X}$ scores vectors
$\mathbf{w}_r$	$m \times 1$	$\mathbf{X}$ loadings vector r ($r : 1, \dots, R$)
$\mathbf{W}$	$m \times R$	matrix of $\mathbf{X}$ loading vectors
$\mathbf{E}$	$T \times m$	Decomposition residual matrix
R	scalar	number of used components

3.3.2 PCA

PCA, for principal component analysis, is one of the most widely known dimension reduction techniques for quantitative data matrices. PCA defines a new coordinate system formed by orthogonal axes called the principal components (PCs) that aim to resume the original variables. Those PCs are new variables constructed as linear combinations of the initial variables. In PCA, these combinations are chosen such that the principal components are orthogonal and maximize the explained variance (Varmuza and Filzmoser, 2016).

As seen above, PCA decomposition is defined as:

$$\mathbf{X} = \mathbf{H}\mathbf{W}' + \mathbf{E} = \mathbf{X}_{app} + \mathbf{E}$$

and the scores matrix is calculated as:

$$\mathbf{H} = \mathbf{X}\mathbf{W}$$

Loading vectors are orthogonal to each others and normalized such that $\mathbf{W}'\mathbf{W} = \mathbf{I}_R$ and scores vectors are orthogonal to each others such that $\mathbf{H}'\mathbf{H}$ is diagonal.

It is also important to mention that, for interpretability and prior to PCA decomposition, the matrix $\mathbf{X}$ is generally column-centered by subtracting the matrix $\mathbf{M}$, which is a $T \times m$ matrix consisting of T identical rows, each a replicate of the vector of column means from $\mathbf{X}$, such that $\mathbf{X}^c = \mathbf{X} - \mathbf{M}$. PCA is then applied on $\mathbf{X}^c$: $\mathbf{X}^c = \mathbf{H}^c\mathbf{W}^{c'} + \mathbf{E}^c$. $\mathbf{X}$ is also sometimes standardized when variables have different units. The choice of the number of components to use is done afterwards based on the eigenvalues (the scalar values representing the amount of variance explained by each principal component, with larger eigenvalues corresponding to more informative components), in order to consider enough explained variance. It can be noted that the low dimension results of a PCA can be used as input for other

methods such as regression or clustering.

Several equivalent algorithms are available to perform PCA decomposition such as the covariance method, the singular value decomposition (SVD) or the Non-linear iterative partial least-squares (NIPALS) algorithm (Varmuza and Filzmoser, 2016).

- Singular Value Decomposition (SVD)

The Singular Value Decomposition is the preferred method for numerical accuracy and can be efficiently computed. The SVD for any matrix $\mathbf{X}$ of size $T \times m$ is:

$$\mathbf{X} = \mathbf{H}_0 \mathbf{S} \mathbf{W}'_F$$

where $\mathbf{H}_0$ ($T \times T$) contains the left singular vectors (orthogonal unit vectors providing a basis for the row space of $\mathbf{X}$), $\mathbf{W}_F$ ($m \times m$) contains the right singular vectors (orthogonal unit vectors providing a basis for the column space of $\mathbf{X}$), and $\mathbf{S}$ ($T \times m$) is a diagonal matrix containing the singular values (scalars quantifying the amount of variance captured by each corresponding pair of singular vectors in the decomposition). Regarding the interpretation, $\mathbf{H}_0$ contains the orthogonal and normalized scores such that $\mathbf{H}'_0 \mathbf{H}_0 = \mathbf{I}_p$. Thereafter, the orthogonal and normalized loadings are in the matrix $\mathbf{W}_F$ such that $\mathbf{W}'_F \mathbf{W}_F = \mathbf{I}_p$. The unnormalized scores are obtained as: $\mathbf{H}_F = \mathbf{H}_0 \mathbf{S}$. When $R \leq \min(T, m)$ components are retained in the decomposition, the first R columns of $\mathbf{H}_F$ and $\mathbf{W}_F$ are retained.

- Non-linear iterative partial least-squares algorithm (NIPALS)

Let us now turn to the NIPALS algorithm which offers advantages such as computational efficiency for large or sparse datasets or robustness to missing data. This algorithm is very similar to others presented afterwards. In this scheme, the $\mathbf{X}^c$ matrix is iteratively deflated/peeled and each new PC is computed on the residual matrix. In the Partial Least Squares section, the same concept will be applied. The NIPALS is shown below (Govaerts and Martin, 2018).

3.3.3 NMF

Non-negative matrix factorization (NMF) was first introduced by Paatero and Tapper (1994). It became popular after a publication by Lee and Seung (1999), in which they proposed a simple multiplicative update algorithm for finding a meaningful parts-based representation of facial images (consisting of pixels

(0)	$\mathbf{X}^c$	Start with the initial mean-centered data matrix $\mathbf{X}^c$
(1)	$\mathbf{h} = \mathbf{x}_j^c$	Initialize the score vector using any column of $\mathbf{X}^c$
(2a)	$\mathbf{w} = \mathbf{X}^{c'} \mathbf{h}$	Compute the approximate loading vector
(2b)	$\mathbf{w} = \mathbf{w} / \ \mathbf{w}\ $	Normalize the approximate loading vector
(2c)	$\mathbf{h}^* = \mathbf{X}^c \mathbf{w}$	Calculate an improved score vector
(2d)	$\Delta \mathbf{h} = \mathbf{h}^* - \mathbf{h}$	check the convergence of $\mathbf{h}$
(2e)	$\mathbf{h} = \mathbf{h}^*$	If $\Delta \mathbf{h} > threshold$, go to step (2a) and repeat
(3a)	$\mathbf{h}_r = \mathbf{h}^*$	Define the r^{th} score vector
(3b)	$\mathbf{w}_r = \mathbf{w}$	Define the r^{th} loading vector
(4a)	$\mathbf{X}_{res}^c = \mathbf{X}^c - \mathbf{h} \mathbf{w}'$	Deflate the $\mathbf{X}^c$ matrix
(4b)	$\mathbf{X}^c = \mathbf{X}_{res}^c$	Replace $\mathbf{X}^c$ by $\mathbf{X}_{res}^c$ and go to step (1) to derive the next component

which have positive intensities).

NMF is applicable to a non-negative input matrix $\mathbf{X}$, and will produce non-negative factor matrices $\mathbf{H}$ and $\mathbf{W}$. The NMF decomposition is then defined as:

$$\mathbf{X} = \mathbf{H} \mathbf{W}' + \mathbf{E} = \mathbf{X}_{app} + \mathbf{E}$$

with $\mathbf{X} \in \mathbb{R}_+^{T \times m}$, $\mathbf{H} \in \mathbb{R}_+^{T \times R}$, $\mathbf{W} \in \mathbb{R}_+^{m \times R}$, $\mathbf{E} \in \mathbb{R}_+^{T \times m}$ and $R \leq \min(T, m)$ (Lin, 2007).

The NMF decomposition algorithms require the number of components R as an input. To choose it, it is first of all suggested to rely on existing information about the data or to draw conclusions from the results obtained from other decomposition methods such as PCA (Boutsidis and Gallopoulos, 2008). As such, the NMF model provides physically interpretable factors due to its non-negativity constraints, which is encountered for a wide range of natural signals (Schmidt, Winther, and Hansen, 2009) such as:

- Spectroscopic data

In the context of chemical reactions monitoring, NMF provides an effective way of decomposing spectral data. In this chapter it is shown that the non-negativity constraint enhances the interpretability of the results compared to another classical decomposition: PCA. Indeed, the spectral data typically consists of intensities measured at a set of wavelengths, the intensities being non-negative. Then, the scores and loadings produced will also be non-negative. The loadings will reflect the R basis elements found in the spectra dataset. If the R is well chosen, the basis elements can be interpreted as the chemical components in the reaction. The loadings are then expected to be close to the true spectral

profiles of the components while the scores are expected to give an idea of the proportion of each basis elements in the reaction. This reflects the standard chemometric approach to modeling spectra over time as an additive mixture problem, where known pure spectra can be provided as priors or initialization (e.g., initializing $\mathbf{W}_0$ with pure profiles and optimizing under non-negativity) to resolve evolving concentrations in a reaction, as demonstrated in the case study initialization.

- Image processing

One of the most popular example of the NMF in practice is the image decomposition. For the data input, a set of images is given. Each image is nothing but a vector with pixel intensities, where every pixel intensity should be non-negative. The NMF will extract a number of features from the data input images (for instance: having glasses, a beard, big eyes, etc) which will be stored in the loadings $\mathbf{W}$ and for each input image, a list of scores (the rows of $\mathbf{H}$) will determine the importance of each feature for the particular input image. The approximation of the input image is the linear combination of the features in the loadings (also known as the basis images) and the scores of the image, as shown in Figure 3.2. The square of 7×7 pixels represents the scores for the particular observation i with $\mathbf{H} \in \mathbb{R}_+^{2429 \times 49}$. The square representing the features is actually another view of the loadings $\mathbf{W}' \in \mathbb{R}_+^{49 \times 19^2}$ where each small square representing a specific feature is a row from $\mathbf{W}'$.

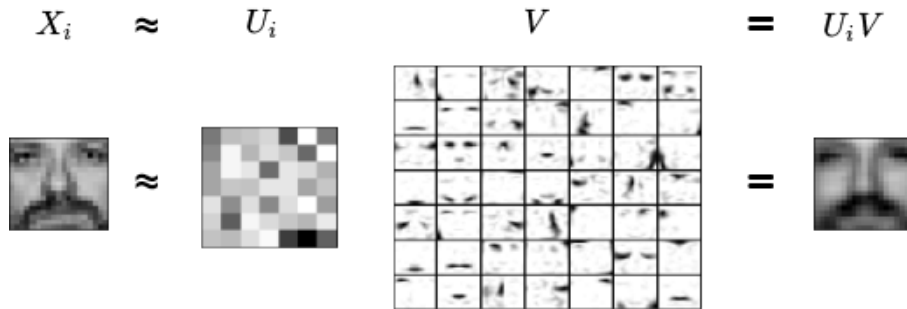


Figure 3.2: Decomposition of faces, MIT Center For Biological and Computation Learning (2429 gray-level 19-by-19 pixels images) with $R = 49$ (Lee and Seung, 1999)

According to Gillis (2014), NMF seems promising in the context of face recognition. For instance, it has been shown more robust than PCA with occluded faces (e.g. wearing sunglasses).

- Text Mining

A typical task in text mining can be the identification of the topic of documents. One can use the term-frequency (TF) matrix, the matrix where each row is a document, each column is a word, and each entry is the number of occurrences of a word in a document. Of course, using the raw counts of words is often not optimal and other weighting techniques exist (for instance, term frequency-inverse document frequency (TF-IDF) gives more weight to the words occurring less frequently).

The NMF on the TF matrix will discover the different R topics among the documents. A document $\mathbf{X}_i$ will be approximated by its R scores and the R basis topics. Each basis topic is of size p , the number of words in the TF matrix. Thus, $\mathbf{X}_i \approx \mathbf{H}_i \mathbf{W}'$ and the R rows of $\mathbf{W}'$ represents the R topics identified. Each basis topic can be interpreted as a “base document”. For instance when approximating a dataset of books, one base topic could represent a crime, a science-fiction or a romance novel by having a large value for the words “guns”, “wizard” or “love”, respectively. By combining the different base topics with a positive score, one can reconstruct a document. Looking at those scores for a particular document allows to classify it among the different topics.

As a conclusion, Gillis (2014) states that “given a set of documents, NMF identifies topics and simultaneously classifies the documents among these different topics”.

Similarity measures

Regarding the formulation of the NMF problem, it is important to be able to measure the difference between the reconstruction $\mathbf{H}\mathbf{W}'$ and the original matrix $\mathbf{X}$ using a similarity measure. Such similarity measure is also needed as a cost function (or objective function) for the optimization problem. The most widely chosen are the Frobenius norm (Squared Euclidean Distance or SED)(Equation 3.1) and the Generalized Kullback–Leibler divergence (GKLD) (Equation 3.2) according to Wang and Zhang (2013). Gillis (2014) however states that the Frobenius norm is the most widely chosen in the literature.

Squared Euclidean Distance:

$$D_{SED}(\mathbf{X}||\mathbf{H}\mathbf{W}') = \frac{1}{2}\|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2 = \frac{1}{2} \sum_{ij} (X_{ij} - [HW']_{ij})^2 \quad (3.1)$$

Generalized Kulback-Leibler Divergence:

$$D_{GKLD}(\mathbf{X}||\mathbf{H}\mathbf{W}') = \sum_{ij} \left(X_{ij} \ln \frac{X_{ij}}{[HW']_{ij}} - X_{ij} + [HW']_{ij} \right) \quad (3.2)$$

For both similarity measures, the lower bound is 0 and the similarity vanishes if and only if $\mathbf{X} = \mathbf{H}\mathbf{W}'$. On the other hand, the GKLD is not a distance but rather a divergence since it is not symmetric: $D_{GKLD}(\mathbf{X}||\mathbf{H}\mathbf{W}') \neq D_{GKLD}(\mathbf{H}\mathbf{W}'||\mathbf{X})$. The GKLD also exhibits some drawbacks. In particular, the gradients used to obtain the scores and the loadings depend on the scales of the matrices. Hence, the convergence speed will be very low as many iterations are required (Wang and Zhang, 2013). Scaling the data is thus needed.

However, both measures can be used as objective function in the algorithm used to derive $\mathbf{H}$ and $\mathbf{W}$: minimize $\|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2$ or $D_{GKLD}(\mathbf{X}||\mathbf{H}\mathbf{W}')$ with respect to $\mathbf{H}$ and $\mathbf{W}$, with constraints $\mathbf{H}, \mathbf{W} \geq 0$.

Algorithms

Although many NMF algorithms exist in the literature, they can be divided into three major categories: gradient descent algorithms, multiplicative update algorithms, and alternating least squares algorithms Berry et al., 2007. It can be mentioned beforehand that none of these algorithms is expected to find a global minimum for the cost function since the cost function (SED or GKLD) is only convex in either $\mathbf{H}$ or $\mathbf{W}$, but not in the two of them together (Lee and Seung, 2001). In the literature, it is seen that the Frobenius norm is the most used.

More details on those 3 categories can be found in the literature and are well summarized in the master thesis of (Roy, 2021). But basically their principle is to fix alternatively one of the factor matrices while optimizing the other one in such a way that the objective function $\|\mathbf{X} - \mathbf{H}_k \mathbf{W}'_k\|_F^2$ is reduced at each iteration k (Gillis, 2014; Lin, 2007).

An example of gradient descent algorithm is as follows:

Gradient Descent Algorithm

1. Fix the number of components R .
2. Initialize matrices $\mathbf{H}_0$ and $\mathbf{W}_0$ to non-negative matrices as random or with prior information.
3. Optimize $\mathbf{H}_k$ and $\mathbf{W}_k$ alternatively for $k = 0, 1, 2, \dots$:
 - (a) Keep $\mathbf{W}_k$ fixed and find $\mathbf{H}_{k+1}$ such that:

$$\|\mathbf{X} - \mathbf{H}_{k+1} \mathbf{W}'_k\|_F^2 < \|\mathbf{X} - \mathbf{H}_k \mathbf{W}'_k\|_F^2$$

where $\mathbf{H}_{k+1} = \text{element max}\{0, \mathbf{H}_k - \alpha_k \nabla_{\mathbf{H}} \|\mathbf{X} - \mathbf{H}_k \mathbf{W}'_k\|_F^2\}$.

- (b) Keep $\mathbf{H}_{k+1}$ fixed and find $\mathbf{W}_{k+1}$ such that:

$$\|\mathbf{X} - \mathbf{H}_{k+1} \mathbf{W}'_{k+1}\|_F^2 < \|\mathbf{X} - \mathbf{H}_{k+1} \mathbf{W}'_k\|_F^2$$

where $\mathbf{W}_{k+1} = \text{element max}\{0, \mathbf{W}_k - \alpha_k \nabla_{\mathbf{W}} \|\mathbf{X} - \mathbf{H}_{k+1} \mathbf{W}'_k\|_F^2\}$.

4. Stop iteration when the convergence criterion is met.
-

Table 3.2: Gradient Descent Algorithm

For the Frobenius norm similarity measure (note the $\frac{1}{2}$ to cancel out the factor 2 in the gradient), the gradients of $\frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2$ are:

$$\begin{aligned} \nabla_{\mathbf{H}} \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2 &= -\mathbf{X}\mathbf{W} + \mathbf{H}\mathbf{W}\mathbf{W}' \\ \nabla_{\mathbf{W}} \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2 &= -\mathbf{X}'\mathbf{H} + \mathbf{W}\mathbf{H}'\mathbf{H} \end{aligned} \quad (3.3)$$

More details about the derivation of those gradients are also given in Section 4.4.2.

The first step of all algorithms is the choice of the number of components R . In the context of this chapter, the number of components R can be chosen to correspond to the number of chemical compounds reacting in the mixture. The idea is that the score matrix $\mathbf{H}$ is proportional to the concentration profiles and that the loading matrix $\mathbf{W}$ corresponds to the spectral profiles.

All NMF algorithms are iterative and are sensitive to the initialization of $\mathbf{H}$ and $\mathbf{W}$. When no prior information is available about them, matrices $\mathbf{H}$ and $\mathbf{W}$ are initialized as dense matrices of random numbers between 0 and 1 (Langville et al., 2014). Those random initializations are repeated a certain number of time and the solution with the lowest cost function is selected. In our context, prior information such as the profiles of the pure component

spectra can be provided as well.

The parameter α_k is the step-size and influences strongly the convergence of gradient descent method. A wrong choice has consequences ranging from a very huge number of iterations to divergence. The gradient of the objective function $\|\mathbf{X} - \mathbf{H}_k \mathbf{W}'_k\|_F^2$ with respect to $\mathbf{H}$ and $\mathbf{W}$ is noted $\nabla_{\mathbf{H}}$ and $\nabla_{\mathbf{W}}$. As the subtraction operations used in each iteration can lead to a violation of the non-negative property of $\mathbf{H}$ and $\mathbf{W}$, negative elements of those matrices are replaced by 0 at each iteration.

There are also several approaches for the stopping criterion of NMF algorithms. Common approaches consist to fix a maximum number of iterations or a time limit to ensure termination of the algorithm (Gillis, 2014). But other options are available Erichson et al., 2016; Gillis, 2014; Cichocki et al., 2009: terminate the algorithm if the value of the cost function is smaller than some stopping condition ε :

$$\|\mathbf{X} - \mathbf{H}_k \mathbf{W}'_k\|_F^2 < \varepsilon$$

or if there is little or no improvement between successive iterations in the minimization of the cost function, for example:

$$\|\mathbf{H}_k \mathbf{W}'_k - \mathbf{H}_{k+1} \mathbf{W}'_{k+1}\|_F^2 < \varepsilon$$

3.3.4 MCR

MCR, for multivariate curve resolution, is the generic name of a family of methods meant to solve chemical mixture problems such as spectroscopic measurements and were first developed in the field of chemometrics (Ruckebusch, 2016).

The approach can be non-iterative and include methods such as Orthogonal Projection Resolution (OPR) and Window Factor Analysis (WFA), or the approach can be iterative and include methods like Multivariate Curve Resolution - Alternating Least Squares (MCR-ALS). In practice, iterative methods are used more frequently and in this chapter only MCR-ALS, which is one of the oldest, most well-known and widely used algorithm, will be analyzed in detail.

The MCR decomposition is defined as:

$$\mathbf{X} = \mathbf{H}\mathbf{W}' + \mathbf{E} = \mathbf{X}_{app} + \mathbf{E}$$

with $\mathbf{X} \in \mathbb{R}_+^{T \times m}$, $\mathbf{H} \in \mathbb{R}_+^{T \times R}$, $\mathbf{W} \in \mathbb{R}_+^{m \times R}$, $\mathbf{E} \in \mathbb{R}_+^{T \times m}$ and $R \leq \min(T, m)$

MCR shares then the same objective as NMF, decomposing a nonnegative matrix $\mathbf{X}$ into non-negative factor matrices $\mathbf{H}$ and $\mathbf{W}$ and update them under additional constraints so that the objective function $\|\mathbf{X} - \mathbf{H}_k \mathbf{W}_k'\|_F^2$ is reduced at each iteration k . The MCR-ALS algorithm can be written as:

MCR-ALS Algorithm

1. Fix the number of components R .
2. Initialize matrices $\mathbf{H}_0$ and $\mathbf{W}_0$ as non-negative matrices, either randomly or with prior information.
3. Optimize $\mathbf{H}_k$ and $\mathbf{W}_k$ alternately for $k = 0, 1, 2, \dots$:

(a) Fix $\mathbf{W}_k$ and find $\mathbf{H}_{k+1}$ such that:

$$\|\mathbf{X} - \mathbf{H}_{k+1} \mathbf{W}_k'\|_F^2 < \|\mathbf{X} - \mathbf{H}_k \mathbf{W}_k'\|_F^2$$

where $\mathbf{H}_{k+1} = \text{element max} \{0, \mathbf{X} \mathbf{W}_k (\mathbf{W}_k' \mathbf{W}_k)^{-1}\}$.

(b) Fix $\mathbf{H}_{k+1}$ and find $\mathbf{W}_{k+1}$ such that:

$$\|\mathbf{X} - \mathbf{H}_{k+1} \mathbf{W}_{k+1}'\|_F^2 < \|\mathbf{X} - \mathbf{H}_{k+1} \mathbf{W}_k'\|_F^2$$

where $\mathbf{W}_{k+1} = \text{element max} \{0, \mathbf{X}' \mathbf{H}_{k+1} (\mathbf{H}_{k+1}' \mathbf{H}_{k+1})^{-1}\}$.

4. Stop iterations when the convergence criterion is met.
-

Table 3.3: Alternating Optimization Algorithm

At each iteration and to ensure non-negativity constraint, the algorithm uses least squares to alternatively optimize $\mathbf{H}$ and $\mathbf{W}$, and negative values are replaced by 0. In practice, $\mathbf{W}_{k+1}$ is obtained by solving the non-negative least squares problem $\min_{\mathbf{W} \geq 0} \|\mathbf{X} - \mathbf{H}_{k+1} \mathbf{W}'\|_F^2$, which can be efficiently computed using dedicated non-negative least squares (NNLS) algorithms rather than randomly. The same principle applies symmetrically when fixing $\mathbf{W}$ to update $\mathbf{H}$. The optimization is usually stopped when the difference of model fit between consecutive iterations does not improve significantly: e.g., a difference of less than 0.1% among the lack of fit (% LOF) criteria between consecutive iterations, or the percentage of variance explained (% var. expl.).

The lack of fit criteria can be defined as: $\%LOF = 100 \frac{\|\mathbf{E}\|_F}{\|\mathbf{X}\|_F}$ and the percentage of variance explained as: $\%var.expl. = 100(1 - \frac{\|\mathbf{E}\|_F^2}{\|\mathbf{X}\|_F^2})$

Other possibilities are using a maximum number of iterations or criteria based on the comparison of shapes of the optimized profiles among iterations (Ruckebusch, 2016). Since MCR was developed in the field of chemometrics, this method proposes additional constraints than component non-negativity to adapt to physical constraints encountered to chemical mixtures. Those constraints are for example unimodality, which allows the presence of a single maximum per concentration profile, or closure, which adapts the total sum of the concentration of compounds to a constant value during all the reaction. Those constraints also help to reduce or suppress the ambiguity of solutions encountered with MCR (and NMF). The ambiguity phenomenon can be described such that different combinations of concentration profiles and spectra can describe equally well, in terms of model fit, the original data set (Tauler, Smilde, and Kowalski, 1995; Tauler and Maeder, 2009).

3.3.5 Methods summary

Table 3.4 briefly summarizes the principal characteristics of methods PCA, NMF and MCR.

Table 3.4: Summary of methods PCA, NMF and MCR.

	PCA	NMF	MCR
Expression	$\mathbf{X}^c = \mathbf{H}^c \mathbf{W}^{c'} + \mathbf{E}^c$	$\mathbf{X} = \mathbf{H} \mathbf{W}' + \mathbf{E}$	$\mathbf{X} = \mathbf{H} \mathbf{W}' + \mathbf{E}$
Objective	Finding new axes (principal components) maximizing the variance of the data	Matrix decomposition into two non-negative matrices	Finding the components of a mixture
Constraints	Orthogonality	Non-negativity of all elements in $\mathbf{H}$ and $\mathbf{W}$	User-defined (e.g. non-negativity, unimodality, closure etc.)
Examples of algorithms	SVD, NIPALS	Multiplicative update algorithms, gradient descent, ALS	MCR-ALS
Prior information	No	Yes	Yes
Main limitations	Interpretation	Ambiguity of the solutions, results influenced by parameter and initialization	Ambiguity of the solutions, results influenced by parameter and initialization
Main advantages	Unique solution, computationally efficient, captures maximum variance	Physically interpretable due to non-negativity (scores resemble concentration profiles, loadings approximate pure spectra), suitable for additive mixtures in chemical reactions	Incorporates chemical knowledge via flexible constraints, resolves mixture components accurately in spectroscopic data for enhanced reaction monitoring

To further emphasize their properties in the context of PAT data: PCA excels in capturing maximum variance through orthogonal components, providing a unique solution ideal for initial exploratory analysis and detecting non-chemical fluctuations, though it may lack direct physical interpretability. NMF and MCR, by enforcing non-negativity, yield factors that are inherently interpretable making them particularly suited for additive chemical mixtures, though both are sensitive to initialization and may converge to local minima.

3.3.6 Comparison of dimension reduction results on batch data

This section explains how NMF, MCR and PCA results quality can be evaluated on spectral data organized in multiple batches. When the data structure consists of batches, it is possible to train models individually on each batch, globally on all batches, or on part of the batches and apply the trained model on the batches left out. Considering an industrial context, the third approach, called in this article the "leave-one-batch-out model", would help for instance to visualize the evolution of a new batch based on a model from historical batches. This section presents these 3 approaches and in Section 3.4.1, these concepts will be applied on pharmaceutical data and compared in terms of interpretability as well as model fit.

Individual model

The first approach consists in applying the decomposition to each matrix $\mathbf{X}_i$ corresponding to the spectra of a single batch or experiment i of size $t_i \times m$. Each matrix can then be decomposed into $\mathbf{X}_i = \mathbf{H}_i \mathbf{W}'_i + \mathbf{E}_i$ and the error of the model is calculated as $\mathbf{E}_i = \mathbf{X}_i - \mathbf{H}_i \mathbf{W}'_i$.

The quality of fit can then be quantified for each batch by the percentage of error as $\%error_i = \frac{\|\mathbf{E}_i\|_F^2}{\|\mathbf{X}_i\|_F^2} \times 100$ and globally by the mean percentage of error by $\frac{1}{I} \sum_{i=1}^I \%error_i$

Global model

The second approach consists in applying the decomposition on the global matrix $\mathbf{X}_G$ of size $T \times m$ where $T = \sum_{i=1}^I t_i$. This matrix corresponds to the spectra of all experimental matrices $\mathbf{X}_i$, with $i = 1 \rightarrow I$, stacked vertically.

The global matrix can then be decomposed into $\mathbf{X}_G = \mathbf{H}_G \mathbf{W}'_G + \mathbf{E}_G$ and the error of the model for each submatrix i calculated as $\mathbf{E}_{G_i} = \mathbf{X}_i - \mathbf{H}_{G_i} \mathbf{W}'_G$.

The percentage of error for one batch can then be calculated as $\%error_i = \frac{\|\mathbf{E}_{G_i}\|_F^2}{\|\mathbf{X}_i\|_F^2} \times 100$ and globally as $\%error = \frac{\|\mathbf{E}_G\|_F^2}{\|\mathbf{X}\|_F^2} \times 100$.

Leave-one-batch-out model

The third approach consists in applying the decomposition to the matrices $\mathbf{X}_{G/i}$ of size $(T - t_i) \times m$ corresponding to the spectra of all batches $\mathbf{X}_G$ without one experiment $\mathbf{X}_i$. Each matrix can be decomposed into $\mathbf{X}_{G/i} = \mathbf{H}_{G/i} \times \mathbf{W}'_{G/i} + \mathbf{E}_{G/i}$.

Using obtained loadings $\mathbf{W}_{G/i}$ and spectra $\mathbf{X}_i$ of the batch left out, scores $\hat{\mathbf{H}}_i$ are calculated by projecting spectra $\mathbf{X}_i$ on the space of $\mathbf{X}_{G/i}$ components. The calculation is done using least squares formula: $\hat{\mathbf{H}}_i = (\mathbf{X}_i \mathbf{W}_{G/i}) \times (\mathbf{W}'_{G/i} \mathbf{W}_{G/i})^{-1}$.

The percentage of error for each submatrix $\mathbf{X}_i$ is then calculated as $\%error_i = \frac{\|\mathbf{X}_i - \hat{\mathbf{H}}_i \mathbf{W}'_{G/i}\|_F^2}{\|\mathbf{X}_i\|_F^2} \times 100$.

The special case of PCA

In PCA, above formulae need to be a little detailed since a centering (and possibly a scaling) is, in most cases, applied before dimension reduction. It is indeed important to emphasize which data should be used for centering in the three approaches.

With the individual model, each $\mathbf{X}_i$ will be column-centered by the matrix $\mathbf{M}_i$ containing the means of each column of $\mathbf{X}_i$ such that $\mathbf{X}_i^c = \mathbf{X}_i - \mathbf{M}_i$. PCA decomposition becomes then $\mathbf{X}_i^c = \mathbf{H}_i^c \mathbf{W}_i^{c'} + \mathbf{E}_i^c$. The percentage of error for each experiment is then defined as $\%error_i = \frac{\|\mathbf{E}_i^c\|_F^2}{\|\mathbf{X}_i^c\|_F^2} \times 100$.

With the global model, $\mathbf{X}_G$ is centered with $\mathbf{M}_G$ containing the means of each column of $\mathbf{X}_G$: $\mathbf{X}_G^c = \mathbf{X}_G - \mathbf{M}_G$. PCA decomposition becomes $\mathbf{X}_G^c = \mathbf{H}_G^c \mathbf{W}_G^{c'} + \mathbf{E}_G^c$ and the percentage of error for each submatrix is $\%error_i = \frac{\|\mathbf{E}_i^c\|_F^2}{\|\mathbf{X}_i^c\|_F^2} \times 100$.

With the leave-one-batch-out model, the training set $\mathbf{X}_{G/i}$ is first centered by $\mathbf{M}_{G/i}$: $\mathbf{X}_{G/i}^c = \mathbf{X}_{G/i} - \mathbf{M}_{G/i}$. The prediction set $\mathbf{X}_i$ is then column-centered by the matrix $\hat{\mathbf{M}}_{G/i}$ of size $(t_i \times m)$ containing the means of the training set $\mathbf{M}_{G/i}$ such that $\mathbf{X}_i^c = \mathbf{X}_i - \hat{\mathbf{M}}_{G/i}$. Scores $\hat{\mathbf{H}}_i^c$ are then calculated simply as $\hat{\mathbf{H}}_i^c = \mathbf{X}_i^c \times \mathbf{W}_{G/i}^c$ and the percentage of error $\%error_i = \frac{\|\mathbf{X}_i^c - \hat{\mathbf{H}}_i^c \mathbf{W}_{G/i}^{c'}\|_F^2}{\|\mathbf{X}_i^c\|_F^2} \times 100$. Centering a new batch using the means from the training set ensures consistent projection into the same reduced-dimensional space as historical batches, facilitating real-time monitoring and comparison without recalibrating the model. Normalization (scaling) is not applied to new batches, as spectral variables share uniform units (absorbance intensities) and

scaling could introduce negative values, compromising non-negativity constraints in NMF/MCR and invalidating cross-method evaluations.

Note that for each of those three cases, even though the centering is advantageous for PCA, the error percentages $\%error_i$ are compared against $\mathbf{X}_i$ not centered in order to allow comparison with NMF and MCR results, which inherently operate on the original non-negative data without centering. Using the centered $\mathbf{X}_i^c$ in the denominator for PCA would scale the error relative to variance alone, potentially underestimating the reconstruction error compared to the raw data and invalidating cross-method evaluations, as NMF and MCR aim to approximate the absolute values of $\mathbf{X}_i$ directly. Using the centered $\mathbf{X}_i^c$ in the denominator for PCA would scale the error relative to variance alone, potentially underestimating the reconstruction error compared to the raw data and invalidating cross-method evaluations, as NMF and MCR aim to approximate the absolute values of $\mathbf{X}_i$ directly.

3.3.7 Multivariate calibration method PLSR

Partial Least Squares Regression (PLSR) is a multiple linear regression technique aimed at predicting/explaining a vector $\mathbf{y}$ of dimensions $T \times 1$ (PLS1) or a matrix $\mathbf{Y}$ of dimensions $T \times n$ (PLS2) from the columns $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ of a matrix $\mathbf{X}$. Generic notations $\mathbf{Y}$ and $\mathbf{X}$ will be used but those matrices could correspond to global matrices $\mathbf{Y}_G^*$ and $\mathbf{X}_G$, as well as individual matrices $\mathbf{Y}_i^*$ and $\mathbf{X}_i$ described in the Section 3.2.1. Contrarily to classical ordinary least squares (OLS) regression, PLS can handle data matrices $\mathbf{X}$ with a number of columns m (variables) higher than the number of rows T (observations/samples). This subsection is devoted to PLS2 and uses the notations of Table 3.5.

Table 3.5: Notations used in PLSR.

$\mathbf{X}$	$T \times m$	data matrix of explanatory variables (column centered and possibly scaled)
$\mathbf{Y}$	$T \times n$	response matrix to be explained by $\mathbf{X}$. $\mathbf{Y}$ is column centered and possibly scaled
$\mathbf{h}_r$	$T \times 1$	$\mathbf{X}$ scores vector r ($r : 1, \dots, R$) with $R \leq \min(T, m)$
$\mathbf{H}$	$T \times R$	matrix of $\mathbf{X}$ scores vectors
$\mathbf{w}_r$	$m \times 1$	$\mathbf{X}$ loadings vector r ($r : 1, \dots, R$)
$\mathbf{W}$	$m \times R$	matrix of $\mathbf{X}$ loading vectors
$\mathbf{u}_r$	$T \times 1$	$\mathbf{Y}$ scores vector r ($r : 1, \dots, R$)
$\mathbf{U}$	$T \times R$	matrix of $\mathbf{Y}$ scores vectors
$\mathbf{q}_r$	$n \times 1$	$\mathbf{Y}$ loadings vector r ($r : 1, \dots, R$)
$\mathbf{Q}$	$n \times R$	matrix of $\mathbf{Y}$ loading vectors
$\mathbf{E}_x$	$T \times m$	$\mathbf{X}$ decomposition residual matrix
$\mathbf{E}_y$	$T \times n$	$\mathbf{Y}$ decomposition residual matrix
$\mathbf{E}$	$T \times n$	matrix of prediction model residuals
$\mathbf{B}$	$m \times n$	matrix of model coefficients

The primary goal of PLSR is to find a linear relation of the form $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$ between the $\mathbf{x}$ - and $\mathbf{y}$ - variables using an $m \times n$ coefficient matrix $\mathbf{B}$ and a prediction error matrix $\mathbf{E}$.

This relation is not found directly since matrices $\mathbf{X}$ and $\mathbf{Y}$ are first decomposed into score matrices $\mathbf{H}$ and $\mathbf{U}$, loading matrices $\mathbf{W}$ and $\mathbf{Q}$ and residuals $\mathbf{E}_x$ and $\mathbf{E}_y$:

$$\mathbf{X} = \mathbf{HW}' + \mathbf{E}_x$$

$$\mathbf{Y} = \mathbf{UQ}' + \mathbf{E}_y$$

In PLS, components are usually calculated iteratively one after the other and at each step r , the covariance between the corresponding $\mathbf{X}$ and $\mathbf{Y}$ score vectors is maximized:

$$\max \text{cov}(\mathbf{h}_r, \mathbf{u}_r)^2 = \max \text{corr}(\mathbf{h}_r, \mathbf{u}_r)^2 \text{var}(\mathbf{h}_r) \text{var}(\mathbf{u}_r)$$

Maximizing the covariance is equivalent to maximize the correlation between the $\mathbf{X}$ and $\mathbf{Y}$ scores AND maximize the variance of the $\mathbf{X}$ and $\mathbf{Y}$ scores which ensures simultaneously the quality of the fit and the quality of the dimension reduction in both $\mathbf{X}$ and $\mathbf{Y}$ spaces.

The orthogonal properties of the scores and loadings depend on the PLS algorithm. NIPALS algorithm was the first algorithm proposed for solving the

PLS problem but other very known algorithms are for example SIMPLS and Kernel algorithms (Varmuza and Filzmoser, 2016; Rosipal and Krämer, 2005).

The way to obtain the matrix of model coefficients $\mathbf{B}$, which will be noted $\mathbf{B}^{PLS}$ in the rest of this chapter, depends on the algorithm which is used but the final coefficients are identical. It can also be noted that $\mathbf{X}$ and $\mathbf{Y}$ are in most cases centered and sometimes standardized before the application of PLS.

3.3.8 Prediction model set-up and comparison

This section presents two methodologies to build prediction models of the concentrations of chemical compounds from the spectral and UPLC data. The “one-step leave-one-batch-out PLS” and the “two-step leave-one-batch-out regression”. Goodness of fit measures are then provided to allow their comparison.

Considering an industrial context, such methodologies can be used to build a model from a set of historical batches, and then apply the model on a new batch to get predictions in real-time without sampling the off-line measurements. The one-step methodology uses PLS to link directly the UPLC data to spectral data. With the two-step methodology, a dimension reduction is first applied on the spectral matrix, e.g. by PCA, NMF or MCR, and then a multivariate regression model is built to predict UPLC data from the obtained scores and UPLC data. The PCA, NMF and MCR analyses described previously are used here as dimension-reduction steps within the two-step prediction framework, where their extracted scores serve as inputs to a multivariate regression model predicting UPLC values. This approach thus connects these unsupervised methods to the supervised prediction context, and is aimed to enhance the interpretability and robustness of the models.

One-step PLS approach

Let's consider the off-line matrix $\mathbf{Y}_G^*$ of dimensions $S \times n$ where $S = \sum_{i=1}^I s_i$, s_i being the number of rows of each subset matrix $\mathbf{Y}_i^*$ corresponding to experiment i . Let's also consider the matrix $\mathbf{X}_G^*$ defined in Section 3.2.1 which is a subset of $\mathbf{X}_G$ with the spectra corresponding to off-line sampling times. $\mathbf{X}_G^*$ is of dimensions $S \times m$, i.e. with the same number of rows as $\mathbf{Y}_G^*$.

The idea of the one-step leave-one-batch-out PLS is to build a PLS model based on $\mathbf{X}_{G/i}^*$ $((S - s_i) \times m)$ and $\mathbf{Y}_{G/i}^*$ $((S - s_i) \times n)$. $\mathbf{X}_{G/i}^*$ and $\mathbf{Y}_{G/i}^*$ being respectively the reduced spectral matrix of all batches, without the i^{th} batch

$\mathbf{X}_i^*$ ($s_i \times m$), and the off-line matrix of all batches without $\mathbf{Y}_i^*$ ($s_i \times n$). The resulting PLS prediction model on calibration data is then written as:

$$\hat{\mathbf{Y}}_{G/i}^* = \mathbf{X}_{G/i}^* \hat{\mathbf{B}}_{G/i}^{PLS}$$

Two-step dimension reduction + regression approach

1. In the first step, a dimension reduction is applied on matrix $\mathbf{X}_{G/i}$ (dimensions $(T - t_i) \times m$) of all spectra without experiment i , to obtain the scores $\mathbf{H}_{G/i}$ and the loadings $\mathbf{W}_{G/i}$ by PCA, NMF or MCR:

$$\mathbf{X}_{G/i} = \mathbf{H}_{G/i} \times \mathbf{W}_{G/i}' + \mathbf{E}_{G/i}$$

$\mathbf{H}_{G/i}^*$ is then defined as the submatrix of $\mathbf{H}_{G/i}$ with the scores corresponding to the off-line data.

2. In the second step, a multivariate regression model is defined to link the spectral data scores $\mathbf{H}_{G/i}^*$ to UPLC data $\mathbf{Y}_{G/i}^*$:

$$\mathbf{Y}_{G/i}^* = \mathbf{H}_{G/i}^* \mathbf{C}_{G/i} + \mathbf{E}$$

Coefficients $\mathbf{C}_{G/i}$ and scores $\mathbf{H}_{G/i}^*$ are estimated by least squares such as $\hat{\mathbf{C}}_{G/i} = (\mathbf{H}_{G/i}^{*'} \mathbf{H}_{G/i}^*)^{-1} \mathbf{H}_{G/i}^{*'} \mathbf{Y}_{G/i}^*$. Rather than directly subsetting the rows of $\mathbf{H}_{G/i}$ (the full scores from the training decomposition) at the sampled timepoints, $\mathbf{H}_{G/i}^*$ are computed using the same projection formula applied to new batches, $\hat{\mathbf{H}}_{G/i}^* = \mathbf{X}_{G/i}^* \mathbf{W}_{G/i} \times (\mathbf{W}_{G/i}' \mathbf{W}_{G/i})^{-1}$ (see Section 3.3.6). This projection mimics how scores would be obtained for a completely new batch in real-time application, ensuring methodological consistency between training and prediction. For PCA, both approaches are equivalent. The resulting model equation can also be written as:

$$\mathbf{Y}_{G/i}^* = \mathbf{H}_{G/i}^* \mathbf{C}_{G/i} + \mathbf{E} = \mathbf{X}_{G/i}^* \mathbf{W}_{G/i} \times (\mathbf{W}_{G/i}' \mathbf{W}_{G/i})^{-1} \mathbf{C}_{G/i} + \mathbf{E}$$

A two-step regression coefficient matrix $\hat{\mathbf{B}}_{G/i}^{2S}$ can then be defined as $\hat{\mathbf{B}}_{G/i}^{2S} = \mathbf{W}_{G/i} \times (\mathbf{W}_{G/i}' \mathbf{W}_{G/i})^{-1} \hat{\mathbf{C}}_{G/i}$ and the resulting two-step prediction model summarized as:

$$\hat{\mathbf{Y}}_{G/i}^* = \mathbf{X}_{G/i}^* \hat{\mathbf{B}}_{G/i}^{2S}$$

Using the model

The model estimated on the training set can then be used to predict the full profiles of the chemical concentrations over time and visualize off-line values directly from the spectra.

For the training set, all intermediate off-line values, which were not sampled, can be predicted and represented graphically from the full spectral matrix $\mathbf{X}_{G/i}$ as $\hat{\mathbf{Y}}_{G/i} = \mathbf{X}_{G/i} \hat{\mathbf{B}}_{G/i}$, with $\hat{\mathbf{B}}_{G/i}$ being $\hat{\mathbf{B}}_{G/i}^{PLS}$ or $\hat{\mathbf{B}}_{G/i}^{2S}$.

Then, off-line values for the batch left out $\mathbf{X}_i$ (or any new batch) can be predicted as $\hat{\mathbf{Y}}_i = \mathbf{X}_i \hat{\mathbf{B}}_{G/i}$.

Finally, predictions of the measured off-line values can be extracted from $\hat{\mathbf{Y}}_i$ as $\hat{\mathbf{Y}}_i^*$ in order to evaluate the quality of the model. A cross validation root mean square error for experiment i can be defined as:

$$RMSECV_{ij} = \sqrt{\frac{\sum_{k=1}^{s_i} (\mathbf{Y}_{ijk}^* - \hat{\mathbf{Y}}_{ijk}^*)^2}{s_i}}$$

with $j : 1, \dots, n$ corresponding to the response columns of $\mathbf{Y}_G^*$, and $k : 1, \dots, s_i$ to the number of rows of the subset matrix $\mathbf{Y}_i^*$ for batch i .

A global RMSECV can be calculated for response j as:

$$RMSECV_j = \sum_{i=1}^I \frac{1}{I} \sqrt{\frac{\sum_{k=1}^{s_i} (\mathbf{Y}_{ijk}^* - \hat{\mathbf{Y}}_{ijk}^*)^2}{s_i}}$$

with i the index of the RMSECV and I the total number of different batches/experiments.

It must also be noted that the reasoning of Section 3.3.6 about centering is also of application for the one-step PLS and two-step data reduction + regression approach.

3.4 Case study

This section uses the dataset described in Section 3.2.1 to illustrate how the techniques presented in Section 3.3 can be used for 2 purposes. Firstly to visualize and interpret chemical reactions, and secondly to predict the concentrations of chemical components. As previously described, PCA, NMF, and MCR are used in this case study within a two-step modeling framework: their scores serve as inputs to multivariate regression models for predicting UPLC concentrations. In contrast, PLS is applied directly in a one-step calibration model linking spectral data to UPLC reference values. Hence, PLS is not merely a benchmark but represents a direct calibration model, while PCA, MCR and NMF support both interpretation and prediction when combined with regression.

3.4.1 Visualization and interpretation of chemical reactions

R packages implementation

R software was used for the implementation of the data reduction techniques PCA, NMF and MCR. This section provides details on this implementation.

For PCA, the function `SVDforPCA` from the R package `MBXUCL` (<https://github.com/ManonMartin/MBXUCL>) (Lapointe-Garant, Simard, and Abatzoglou, 2008) was used. Centering was applied before singular value decomposition but no standardization. `SVDforPCA` is now also available in `limpca` package as function `pcaBySvd`.

For NMF, the function `spectralNMF` from the R package `spectralAnalysis` (version 3.12.0, (Van Oirbeek et al., 2018)) was used. Random initialization as well as initialization using prior loadings have been tested. For random initialization, $\mathbf{H}_0$ and $\mathbf{W}_0$ were set to pseudo-random non-negative values between 0 and 1. The NMF algorithm PGNMF (projected gradient descent algorithm) was used (Lin, 2007). The stopping criterion was set such that if $(\|\mathbf{E}_{k+1}\|_F - \|\mathbf{E}_k\|_F) \leq 0.00001$ the algorithm stopped.

For MCR, the function `als` from the R package `ALS` (version 0.0.6, (Mullen, 2015)) was used. Random initialization as well as initialization using prior loadings were also tested with the same approach as for NMF. The algorithm used for the `als` function is alternating least squares. The stopping criterion was set such that if $\frac{\|\mathbf{E}_{k+1}\|_F - \|\mathbf{E}_k\|_F}{\|\mathbf{E}_k\|_F} \leq 0.001$ the algorithm stopped.

The stopping criteria differ as they follow the conventions and defaults of the respective R packages (absolute change in reconstruction error for spectralNMF to provide precise control over small increments, and relative change for ALS to scale with residual magnitude), ensuring algorithm-specific convergence while yielding comparable model qualities in practice, as evidenced by the similar error percentages in Table 3.6.

Results

This section shows the results of the application of dimension reduction methods PCA, MCR and NMF to the matrix $\mathbf{X}_G$ and submatrices $\mathbf{X}_i$ following the individual, global or leave-one-batch-out methodologies. Error percentages for each batch and methodology were computed in order to compare the fit of the different models, and the scores and loadings were analyzed to compare the interpretability of each method.

Figure 3.3 shows the evolution of the mean error percentage when the number of components in the model increases. Error percentages are presented for PCA, MCR and NMF following the individual, global or leave-one-batch-out methodologies up to 6 components, as this range sufficiently captures the primary chemical signals (e.g., the two main compounds) plus potential minor variations or noise, with diminishing returns in error reduction beyond 6. For NMF and MCR the convergence of the algorithm started only at 2 components. It can be observed that the mean error percentage decreases when the number of components increases and that errors tend to be closer for PCA, NMF and MCR with more components. For the global and leave-one-batch-out methodologies, the slopes of the decrease start to be similar at 3 components, whereas for the individual methodology 4 components seem to be necessary. For the individual model, a shift to the left of PCA compared to NMF and MCR can be observed for the first few components. This suggests that PCA requires one component less to reach the same error percentage as NFM and MCR. One reason could be that PCA needs only 1 component to explain the reactant and product correlated negatively, while NMF and MCR requires 2 components. It can also be observed that the mean error decreases differently for the individual methodology, whereas it is similar for the global and leave-one-batch-out methodologies. The main goal of this figure is to guide the decision for the number of components to use in the model. In this case, the choice was difficult between 2 and 3 components, and there was a trade-off between the fit and interpretability. In concertation with chemists, and to improve interpretability, two components were finally chosen for the calculation of model errors as well as to display scores and loadings. Extra components

were indeed not correctly modeled by NMF and MCR and reduced the interpretability of the first two components modeling the reactant and the product. Results with 3 components are nevertheless given in Appendix 3.6.

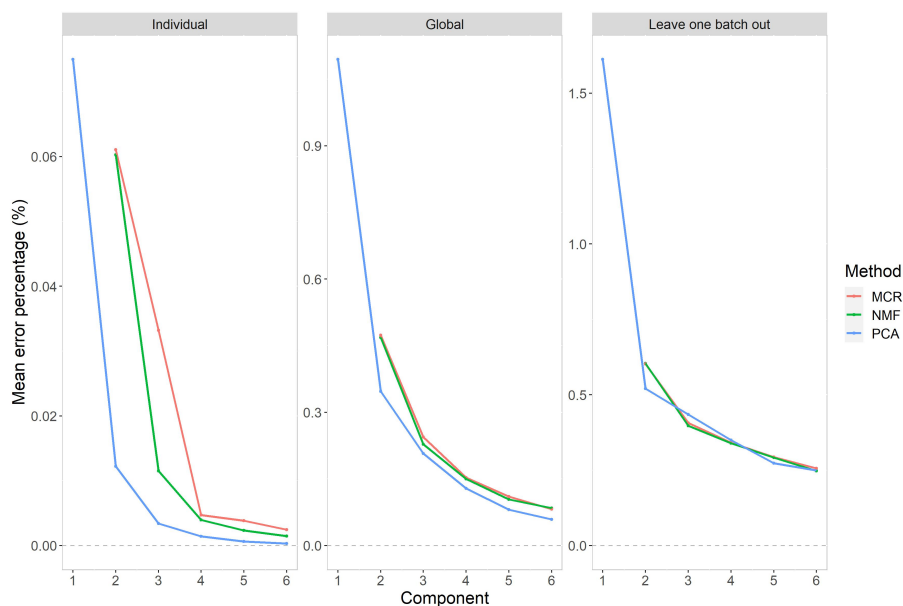


Figure 3.3: Mean error percentages by number of components.

Error percentages for each batch are given by the methodology in Table 3.6 for this choice of two components. Moreover, percentages for NMF/MCR with random priors are also given.

Looking to the mean percentages of error per column, it can first be observed that, for every method (PCA, NMF or MCR), lowest errors are obtained using the individual methodology, followed by the global and leave-one-batch-out approaches. This can be explained since the individual methodology finds loadings and scores for each batch separately, leading to results fitted for each batch. The global methodology finds loadings common to all batches, and next, the scores are extracted per experiment. Those loadings have the advantage to be common for all experiments but lead to less fitted results than the individual methodology. And finally, the leave-one-batch-out methodology finds common loadings to all batches, except the batch left out. The scores are then estimated for the batch left out based on the loadings found for other experiments, leading to higher error percentages than individual and global methodologies. It can then be seen that model errors of NMF and MCR, with and without

priors, are similar for each methodology (individual, global and leave-one-batch-out). Those results are quite expected since even if the algorithms used for MCR and NMF are different, the optimized criterion is the same and results are logically similar. With random priors, iterations were also giving similar results both for NMF and MCR.

Looking at the errors for individual models per batch for PCA, experiment 5 has the highest error percentage, followed by experiments 4, 8, 2 and 11. For NMF and MCR, the highest error percentages are obtained for experiment 8, 11, 5 and 2.

Looking at errors for global models, it can be seen that for PCA, it is experiment 2 which has the highest error, which is also the case for methods NMF and MCR.

Table 3.6: Model error percentages per batch and methodology.

Batch number	PCA centered			MCR			MCR random priors			NMF			NMF random priors		
	Individual	Global	LoBo	Individual	Global	LoBo	Individual	Global	LoBo	Individual	Global	LoBo	Individual	Global	LoBo
1	0.006	0.292	0.348	0.013	0.379	0.436	0.013	0.379	0.435	0.013	0.375	0.436	0.013	0.373	0.433
2	0.016	1.182	1.821	0.101	1.942	2.413	0.101	1.941	2.413	0.098	1.950	2.422	0.098	1.937	2.417
3	0.007	0.228	0.384	0.016	0.192	0.202	0.016	0.193	0.202	0.016	0.176	0.205	0.016	0.173	0.201
4	0.019	0.424	0.501	0.062	0.364	0.416	0.062	0.364	0.416	0.063	0.363	0.417	0.062	0.361	0.414
5	0.048	0.284	0.709	0.125	0.411	0.881	0.125	0.411	0.880	0.124	0.414	0.888	0.124	0.411	0.882
6	0.004	0.095	0.121	0.022	0.197	0.223	0.022	0.198	0.223	0.021	0.188	0.222	0.021	0.188	0.222
7	0.001	0.143	0.163	0.010	0.165	0.181	0.010	0.165	0.181	0.010	0.162	0.183	0.010	0.160	0.181
8	0.018	0.355	0.534	0.144	0.552	0.707	0.144	0.551	0.707	0.141	0.544	0.701	0.141	0.550	0.710
9	0.003	0.119	0.356	0.023	0.182	0.246	0.023	0.183	0.246	0.023	0.162	0.246	0.023	0.162	0.245
10	0.003	0.060	0.068	0.012	0.073	0.078	0.012	0.073	0.078	0.012	0.072	0.081	0.012	0.069	0.078
11	0.008	0.636	0.718	0.143	0.749	0.844	0.143	0.749	0.844	0.142	0.744	0.846	0.142	0.742	0.843
Mean	0.012	0.347	0.520	0.061	0.473	0.602	0.061	0.473	0.602	0.060	0.468	0.604	0.060	0.466	0.603

Individual Models

Figure 3.4 presents the loadings obtained by PCA, NMF and MCR from the individual models applied on experiment 1. Sub-figure A shows the loadings of PCA for the first two principal components, sub-figure B shows the loadings obtained by NMF and MCR corresponding to the reactant and product, and sub-figure C shows the pure spectra of the reactant and product obtained by MIR spectroscopy. The pure spectra were used to initialize NMF and MCR algorithms which allows to make the correspondence among components and chemical compounds. Note that random initialization was also used and provided similar results. It can first be observed that PCA loadings have positive and negative values which makes the correspondence with spectral profiles not straightforward. Indeed, components calculated by PCA are less designed to capture the chemical compounds profiles. In the case of NMF and MCR, the loadings have only positive values, due to the non-negativity constraint, and a correspondence among components and chemical compounds is expected. One can see in Figure 3.4 that the loadings are not exactly the same as the spectral profiles but are easier to interpret for the chemists than in the case of PCA. It can finally be noted that the loadings found by NMF and MCR are extremely close, meaning that those two methods did converge to similar solutions.

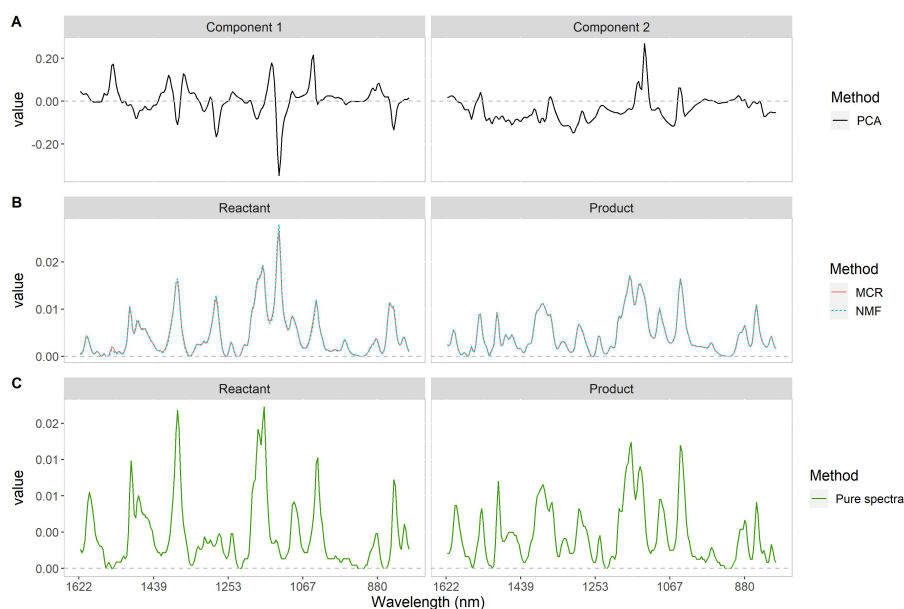


Figure 3.4: Loadings for experiment 1 based on individual models.

Figure 3.5 presents the scores obtained by PCA, NMF and MCR from the individual model applied on experiment 1. Sub-figure A shows the scores of PCA for the first two principal components, sub-figure B shows the scores obtained by NMF and MCR corresponding to the reactant and product, and sub-figure C shows the profiles of the reactant and product concentration of the reaction obtained by UPLC. In sub-figure C, the dots correspond to measured values obtained by UPLC. Since UPLC values were measured only at 0, 2, 4 and 21 hours, a straight dashed line is used to connect them. Those UPLC profiles can be used as reference. It can first be observed that PCA scores have positive and negative values which makes the interpretation not straightforward. The first component might correspond to both the evolution of the product and the reactant since they are negatively correlated, but the second one does not have a clear pattern. In the case of NMF and MCR, the scores have only positive values and correspond to the evolution of the quantities of the chemical compounds of the reaction, which is easier for interpretation. The quantity of the reactant decreases over time and the product increases such as seen with UPLC values. The scale of NMF and MCR scores differ of course from UPLC values but, for monitoring purpose, only the observation of the kinetics trends is important. Finally, it can be noted that scores given by NMF and MCR are also quite close but not identical.

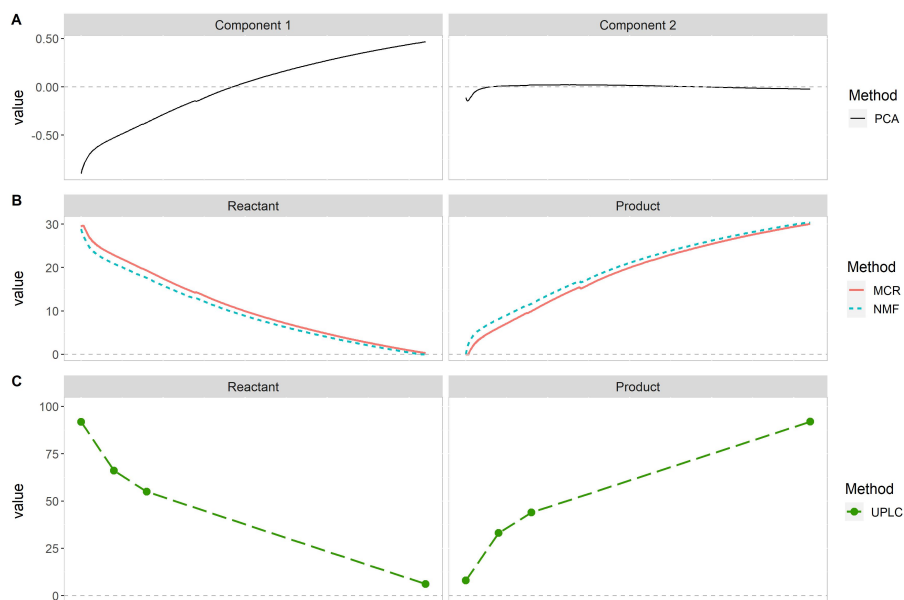


Figure 3.5: Scores for experiment 1 based on individual models.

Figure 3.6 presents the scores obtained by PCA for individual models. Each plot shows the scores of the first against the second principal component for each experiment. Moreover, the line connecting the values has been colored from green to red, to indicate at which time of the 21 hours of reaction the scores correspond. Percentages of variation are also indicated for the first two components. For the first principal component, the percentages range between 84.5 and 99.2%. For the second principal component, the percentages range between 0.5 and 9.5%. It can be seen that experiment 4 presents many fluctuations. Those fluctuations were also detected by NMF on the score plots of the starting material and product (Figure 3.7). Experiment 4 is however not the one with the highest error percentage (0.019%), but experiment 5 as observed previously in Table 3.6 (0.048%). It can be seen that experiments 2, 5, 8 and 11 have all an irregular pattern detected on the second component at the end of the reaction. Those 4 experiments have also higher error percentages than others, with respectively 0.016, 0.048, 0.018 and 0.008%. Interestingly, the fluctuations seen on experiments 2, 5, 8 and 11 were not observed on the score plots of NMF (Figure 3.7). This might suggest that those fluctuations would not be related to chemical components but rather to another factor captured by the second principal component of PCA.

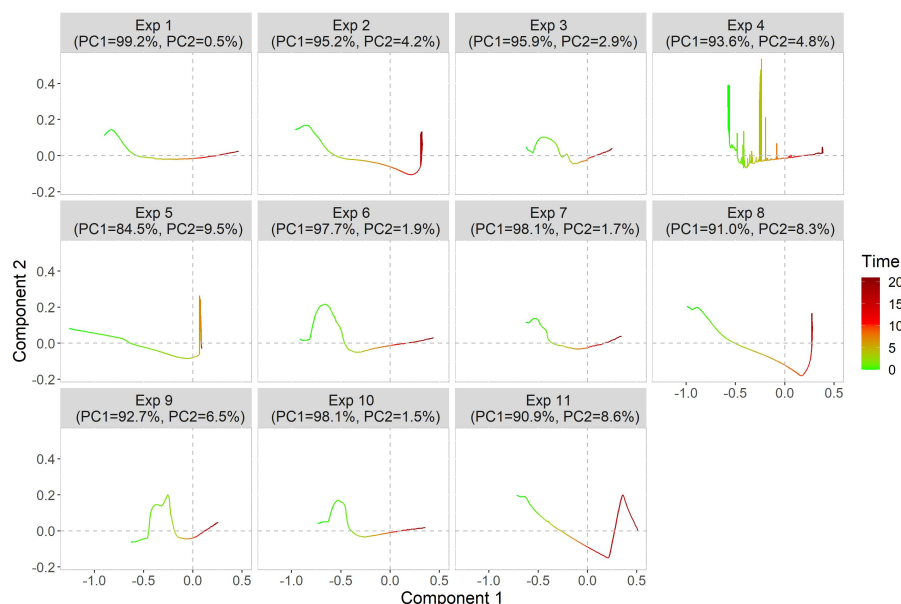


Figure 3.6: Scores 1 vs scores 2 values for each experiment based on individual models.

Global Models

Figure 3.7 presents the scores obtained by NMF for the reactant and product based on the global model (sub-figure A) as well as the UPLC values (sub-figure B). The global model allows to visualize all the reactions on a same space and to compare their kinetics. It was chosen to display only NMF scores since MCR provides similar results. Similarly to the individual models, the pure spectra of the reactant and product were used to initialize the NMF algorithm. It can be observed that all scores have positive values, making the interpretation easier. The reactant scores decrease and the product scores increase for each experiment as expected. Important differences can be observed in the conversion speed among the reactions. Indeed, those reactions come from a research and development context and were performed on extreme conditions to understand the impact of different process parameters. Experiment 5, followed by 2 and 8 present the fastest kinetics. The product increases really fast during the first hour, is then produced linearly, until reaching a plateau. The plateau is reached after around 5 hours for experiment 5, and around 10 hours for experiments 2 and 8. It is also interesting to see an irregular pattern on the second principal component of those experiments on the PCA score plots (see Figure 3.6). On the opposite, experiments 3 and 9 were among the slowest, and did not reach a plateau even after 21 hours. Finally, experiment 4 presents a few fluctuations as seen before with PCA. The advantage of NMF compared to UPLC is clearly seen on this figure. Indeed, since spectral measurements are taken continuously, NMF allows to visualize the complete kinetics trend, and to detect fluctuations or plateaus in the reaction. For UPLC, since only four samples were taken during the reaction, some information can be missed as for example the position of the plateau for experiment 2 to 8.

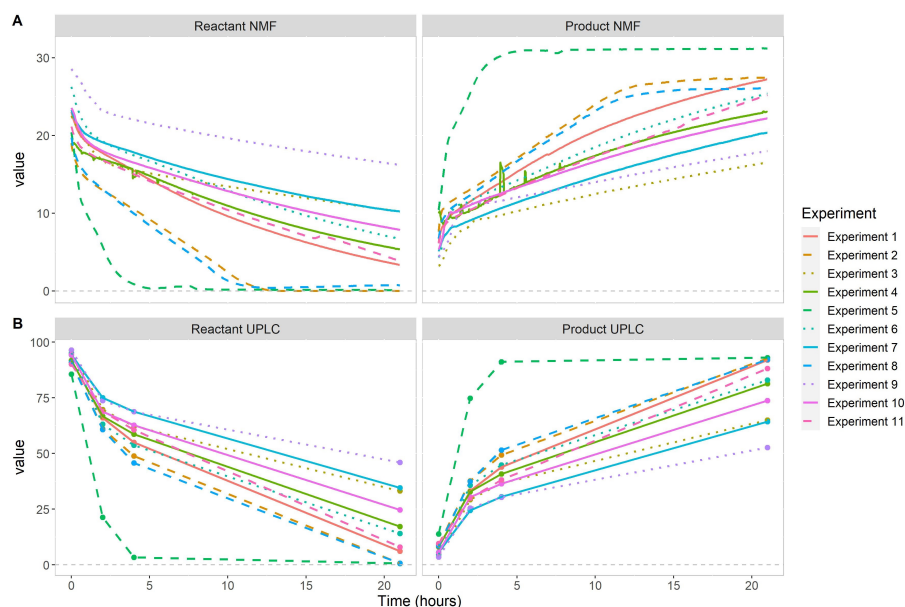


Figure 3.7: UPLC values and NMF scores over time based on the global model.

Leave-One-Batch-Out Models

LoBo models extend the global approach by training on historical batches and projecting new ones, enabling real-time monitoring while assessing generalizability. Although specific figures for LoBo scores and loadings are not presented here, the error percentages in Table 3.6 indicate slightly higher errors for LoBo compared to global models, reflecting the challenge of out-of-sample projection.

A few practical points and results can be emphasized from this case study:

- Chemical reactions can be visualized with individual and global models. Since the dimension-reduction of individual models is done on different spaces, those models are more appropriate to visualize experiments separately than to compare the trends together. They can be used for example in a first step as a benchmark when there are only a few experiments or when the chemical reactions are different. In the case of global models, the dimension-reduction is done on a same space for all experiments, which allows to visualize and compare them together. Global models present moreover the advantage of being more robust, since several experiments are used to build the model, which can prevent over-fitting. The space of a global model can also be used to project in real time the spectra of new batches and visualize directly how they

perform compared to the experiments used to build the model. This can be for example useful to control if a new experiment stays in a certain range or differs from it.

- NMF and MCR provide an interpretable visualization of chemical reactions. Indeed, those methods allow to choose the number of components upfront to correspond to the number of chemical compounds. Moreover, the loadings are closer to the pure spectra and the scores closer to UPLC profiles due to the non-negativity constraint. Even though PCA is less interpretable, applying it remains a good tool in this context since it can help to detect particular behaviours in reactions (e.g. experiment 4) and take the decision to stop them if necessary. Other examples are experiments 2, 5 and 8 which presented an irregular pattern on the second PCA component, at the end of the reaction, and those experiments were seen as having the fastest kinetics in NMF.
- NMF and MCR provide equivalent results even though their algorithms are different. The only constraint applied was non-negativity since the absorbance of the spectra and the concentration of the chemical compounds are supposed to be non-negative. However, it is important to keep in mind that other constraints can be applied depending of the chosen package. This is well known for MCR, which was developed to solve chemical mixture problems, and for which the algorithms propose constraints such as closure and unimodality.

For industry practitioners implementing these methods in pharmaceutical R&D or production, it is recommended to start with individual models for exploratory analysis of single batches, then transition to global or leave-one-batch-out models for robust comparison and real-time monitoring, while validating with error metrics like reconstruction error. Additionally, set the number of components R to match known chemical compounds, such as 2 for reactant and product, and employ R or packages like `nmf` for NMF or `ALS` for MCR, incorporating non-negative preprocessing to preserve data integrity. Finally, integrate these methods with PAT tools for on-line spectroscopy to facilitate early anomaly detection and process optimization.

While this case study focuses on a simple reaction with effectively one reactant and one product ($n = 2$, ignoring the low-concentration side product), the methods extend to more complex multi-reactant systems by increasing R to match the number of components and applying additional constraints (e.g., unimodality or closure in MCR) to resolve overlapping spectra and reduce ambiguity (Tauler, Smilde, and Kowalski, 1995; Ruckebusch, 2016). While the case study focuses on a simple reaction with one primary reactant and product,

NMF and MCR are particularly well-suited for more complex data involving multiple reactants, products, or intermediates. These methods allow the number of components R to be specified upfront to match the expected chemical species, with non-negativity constraints ensuring physically meaningful decompositions; MCR further enhances adaptability through additional constraints like closure or unimodality to resolve overlapping signals in multi-component mixtures.

3.4.2 Chemical components prediction models

Section 3.4.1 was focusing on how to visualize and interpret the kinetics of chemical reactions applying PCA, NMF and MCR on spectral data. Those methods can be useful to follow the trend of a chemical reaction over time but do not allow to quantify the concentration of the chemical components.

This section illustrates the use of PLS and multivariate regression to predict the concentrations of chemical components using the one- and two-step leave-one-batch-out methodologies presented in Section 3.3.8.

Results

Figure 3.8 shows the evolution of the mean RMSECV of the product for one-step PLS, and two-step multivariate regression based on the PCA, NMF and MCR scores, when the number of components in the model increases. For PLS, the mean RMSECV decreases from 1 to 2 components and then remains stable. For PCA, NMF and MCR, the mean RMSECV's globally decrease from 2 to 5 components to get closer to the value of PLS. As for Figure 3.3, the idea of this figure is to guide the decision for the number of components to use in the model. The number of components was not selected independently for each method to optimize RMSECV (e.g., allowing PLS or PCA to use more for potentially lower errors). Instead, we fixed 2 components across all to align with the chemically meaningful factors (reactant and product) best captured by NMF/MCR, prioritizing interpretability over minimal RMSE gains. A dip is observed at 5 components in PCA/NMF/MCR, but we avoided higher numbers as gains beyond 2-3 were marginal (e.g., $< 1\%$ RMSECV improvement for PCA and NMF) and risked reducing physical relevance. In this case, the choice was again to use 2 components since for NMF and MCR only the first two components modelize correctly the chemical compounds and there was not a big gain to use 3 components.

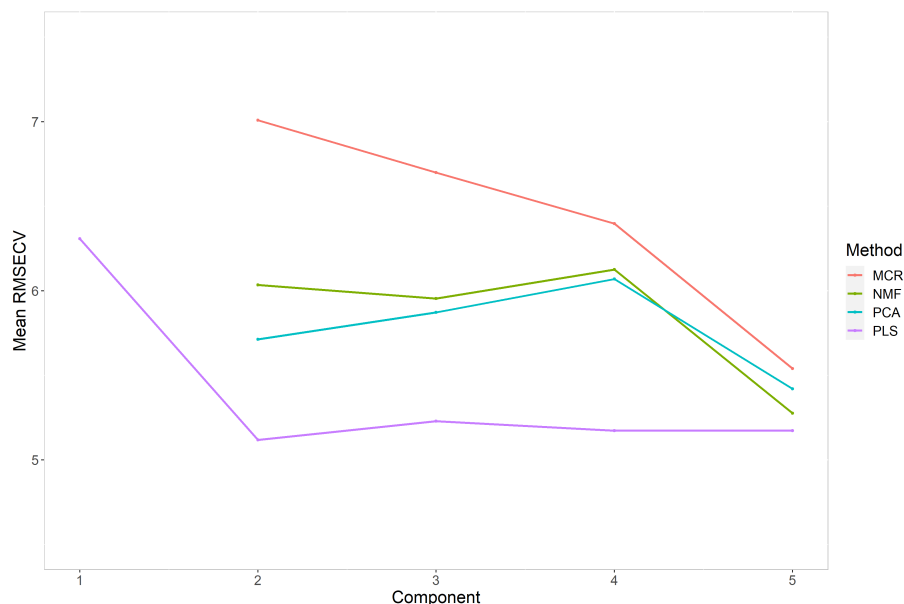


Figure 3.8: Mean RMSECV of the product by number of components.

Table 3.7 presents the RMSECV of the product for each batch obtained by the one-step PLS with 2 components, and two-step multivariate regression based on the scores of PCA, NMF and MCR also with 2 components. For the two-step multivariate regression, only two components were used since only those are meaningful in NMF and MCR. For comparability, the results with the first two components for PCA and PLS were also used.

It can be observed that in average, by methodology, the lowest RMSECV is obtained for the one-step PLS (5.12), two-step regression + PCA (5.71), two-step regression + NMF (6.03) and finally the two-step regression + MCR (7.06). Those results show that the one-step PLS provides slightly lower RMSECV's than the two-step methodology. However, the scores obtained by PCA, MCR and NMF can also be used for prediction, with the further advantage for MCR and NMF to be interpretable. Those RMSECV values should be put into perspective. Indeed, the data used in this article were not designed specifically to build a calibration model, so it would be possible to reach lower RMSECV's with a better design. As a point of comparison it can also be mentioned that UPLC values range from 0 to 100 for each batch, which shows that the differences between the one- and two-step models are relatively small. It can also be pointed out that the one-step PLS is not always the best. Indeed, for some batches, the two-step approach provides lower RMSECV's.

Table 3.7: RMSECV of the product from one-step PLS and two-step multivariate regression with 2 components each.

Batch	One-step PLS	Two-step PCA	Two-step MCR	Two-step NMF
1	1.95	2.84	4.62	3.34
2	4.64	5.12	4.62	5.66
3	7.02	7.26	9.45	9.03
4	10.10	10.39	8.44	10.07
5	12.17	12.36	13.15	12.47
6	5.53	6.53	10.05	6.77
7	0.81	1.46	3.36	1.76
8	4.65	4.52	2.79	4.47
9	4.44	6.39	14.46	6.94
10	1.48	2.15	3.53	2.24
11	3.48	3.83	3.22	3.62
Mean	5.12	5.71	7.06	6.03

This is for example the case for batch 2 (MCR), batch 4 (MCR, NMF), batch 8 (PCA, MCR, NMF), and batch 11 (MCR).

Figure 3.9 presents the predictions for experiment 1 based on the one-step leave-one-batch-out PLS with 2 components, and the two-step leave-one-batch-out regression, based on PCA, MCR and NMF with also two components. This means that the models were built on all experiments except experiment 1, and then applied on the spectra and scores of this experiment. The lines represent the predicted values whereas the points represent the observed UPLC values. One can see that the best fit is obtained for the one-step PLS, whereas the least good is for the two-step regression based on MCR scores as seen on Table 3.7. It is also important to note that the predictions are obtained at all intermediate points not sampled by UPLC, which allows to visualize continuously the reaction. When looking at the kinetics, it can be observed that the reaction is the fastest during the first hour, then continues constantly up to 21 hours, but does not reach a plateau. The gap in UPLC measurements between 4 and 21 hours reflects the limited sampling rate of off-line techniques, which involve time-consuming extraction and chromatographic separation, which allowed here only four samples during the 21-hour experiment (at 0, 2, 4, and 21 hours).

From a practical point of view, this section shows that spectra as well as scores can be used to build a calibration model when reference measurements are available. Three applications are possible. The first one is to build a PLS model on a single experiment to visualize it afterwards. As said before, it allows to interpolate chemical concentrations at all time points which were not sampled off-line. This can help to visualize the complete kinetics or particular

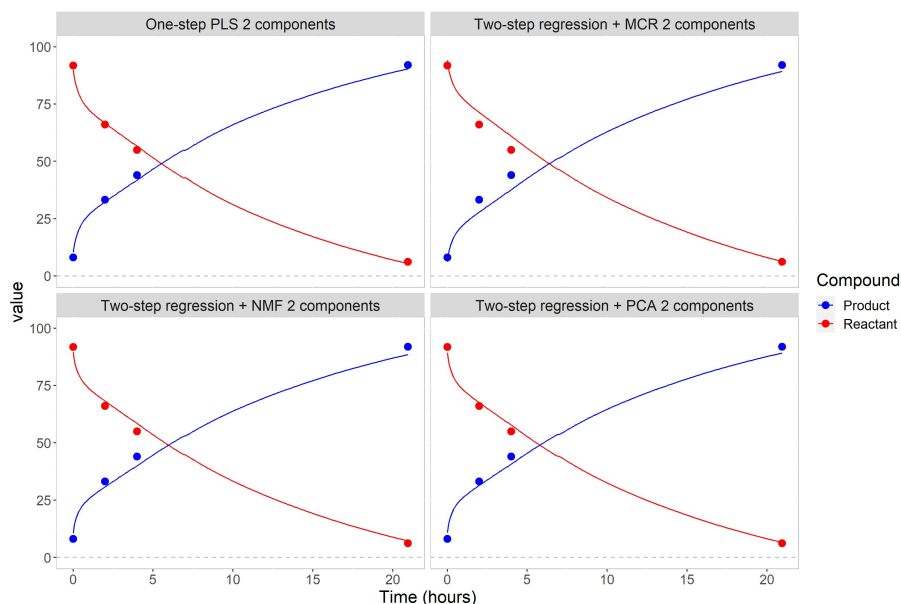


Figure 3.9: Leave one batch out PLS prediction for batch 1.

patterns. The second application is to build a PLS model on multiple experiments. The visualization is also done afterwards but this kind of model has the advantage of being more robust than for one experiment and allows the comparison of multiple experiments together. And finally, as shown with Figure 3.9, the third application is to use a global model to predict the chemical concentrations over all time points of a new reaction instead of taking off-line measurements. Those results show that with 10 experiments and 4 sampling times per experiment, predicting a reaction of 21 hours is possible. The kinetics of this reaction was of course simple, and only two compounds needed to be visualized, but those are coming from a realistic industrial context, and the principle remains the same for more complex reactions.

Two recommendations can be given to build this kind of model in a R&D context. When the goal is to understand the impact of different factors on the reaction, the first recommendation is to generate data from a DoE. Indeed, DoE's can help to capture the effect of the different factors with an optimal number of experiments. Since those reactions occur over time, the second recommendation is to sample off-line measurements at enough time points in order to correctly capture the kinetics of the reaction.

Despite these advantages, practical challenges include handling noisy spectral data (e.g., baseline drift requiring non-negative preprocessing), selecting the

optimal number of components R to avoid over- or under-fitting, and computational demands for large datasets in real-time applications. Validation against reference methods like UPLC is essential, and for complex reactions with more compounds, additional constraints (e.g., closure in MCR) may be needed as well.

3.5 Conclusion

Following the PAT initiative promoted by the Food and Drug Administration, spectroscopic methods have become popular in the pharmaceutical industry to continuously monitor critical quality attributes during chemical processes. Spectroscopic methods generate however multivariate data requiring the use of chemometrics approaches to be interpretable.

This chapter used a set of chemical experiments performed in an industrial R&D context to study the synthesis of an active molecule. Those experiments were monitored on-line by mid-infrared spectroscopy and off-line by UPLC.

A practical framework was presented to answer two objectives. The first objective was to compare chemometrics approaches PCA, NMF and MCR to visualize, monitor and interpret chemical reactions. The second objective was to propose different approaches to build real-time prediction models to predict chemical concentrations using calibration methods PLS and multivariate regression.

For the first objective, it has been seen that methods NMF and MCR allowed a better interpretation of spectral data than PCA thanks to the non-negativity constraint and the prior choice of the number of components. Results also showed that even though NMF and MCR come from different fields, they share the same objectives and provide in practice similar results.

For the second objective, it has been seen that spectra or scores obtained by PCA, NMF and MCR could be used to build a calibration model with UPLC measurements. A one-step approach using the spectra and a two-step approach using the scores were compared and provided close results. In practice, those calibration models allow to predict in real time, and continuously, the chemical concentrations of a new reaction without having to sample off-line measurements. This work brings concrete benefits to the pharmaceutical industry by enabling faster process optimization, reducing production downtime through early anomaly detection, and lowering costs associated with off-line testing. In the context of Janssen Pharmaceutica's R&D, it supports scaling from lab to industrial production by providing robust, interpretable tools for batch monitoring. Broader than the statistical comparisons, this chapter intended to present different tools to assist scientists in the development of chemical compounds.

Chapter 4 will extend this work by proposing a supervised version of the NMF, integrating regression penalty terms.

3.6 Appendix

Figure 3.10 presents the scores obtained by NMF for the reactant, product and a side product based on the global model (sub-figure A) as well as the UPLC values (sub-figure B). It can first be observed that the side product is not well modelled and is too high compared to the main product. A second observation is that the main product is not modelled as well as in Figure 3.7. For example, experiments 2 and 5 reach lower values at the end of the reaction than what is observed with UPLC values.

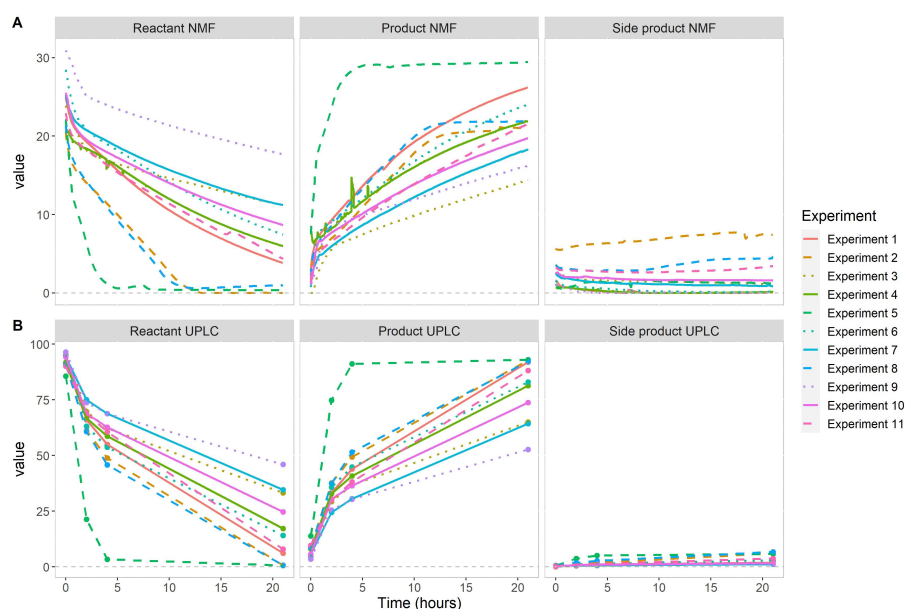


Figure 3.10: UPLC values and NMF scores over time based on the global model.

Table 3.8 shows the error percentages for each batch and methodology with 3 components. Those errors can be compared with table 3.6.

Table 3.8: Model error percentages per batch and methodology.

Batch number	PCA centered			MCR			MCR random priors			NMF			NMF random priors		
	Individual		Global	Individual		Global	Individual		Global	Individual		Global	Individual		Global
	LoBo	Global	LoBo	LoBo	Global	LoBo	LoBo	Global	LoBo	LoBo	Global	LoBo	LoBo	Global	LoBo
1	0.003	0.171	0.234	0.013	0.203	0.247	0.006	0.198	0.251	0.006	0.196	0.250	0.006	0.196	0.250
2	0.008	0.302	1.692	0.101	0.483	1.373	0.016	0.301	1.362	0.017	0.302	1.250	0.015	0.401	1.231
3	0.002	0.123	0.241	0.016	0.129	0.160	0.016	0.154	0.152	0.005	0.136	0.160	0.006	0.132	0.158
4	0.003	0.206	0.331	0.011	0.236	0.290	0.009	0.236	0.286	0.009	0.234	0.285	0.009	0.232	0.290
5	0.009	0.276	0.702	0.050	0.335	0.752	0.048	0.300	0.743	0.048	0.309	0.728	0.048	0.318	0.738
6	0.001	0.074	0.100	0.006	0.081	0.101	0.006	0.093	0.106	0.004	0.087	0.113	0.005	0.084	0.110
7	0.001	0.125	0.150	0.002	0.133	0.160	0.002	0.142	0.160	0.002	0.136	0.159	0.002	0.135	0.158
8	0.007	0.287	0.482	0.017	0.320	0.452	0.016	0.344	0.454	0.018	0.336	0.483	0.019	0.323	0.477
9	0.001	0.053	0.102	0.003	0.107	0.164	0.003	0.118	0.165	0.003	0.109	0.172	0.003	0.110	0.171
10	0.001	0.039	0.045	0.004	0.043	0.051	0.012	0.049	0.051	0.003	0.045	0.051	0.003	0.045	0.050
11	0.003	0.623	0.705	0.143	0.620	0.723	0.143	0.623	0.718	0.010	0.620	0.723	0.008	0.620	0.720
Mean	0.003	0.207	0.435	0.033	0.244	0.407	0.025	0.232	0.404	0.011	0.228	0.398	0.011	0.235	0.396

CHAPTER 4

Supervised chemometric approaches for interpretable and
predictive modeling: SNMF

4.1 Introduction and objectives

This chapter builds upon the methodologies developed in Chapter 3, which explored unsupervised chemometrics strategies for monitoring active pharmaceutical ingredients in chemical reactions. Julien Van Roy's master thesis (Roy, 2021), supervised by myself, Bernadette Govaerts, and Rainer von Sachs, extends this by introducing a supervised variant. While this chapter summarizes Van Roy's work without adding new results, it highlights our contribution in proposing a novel Supervised Non-negative Matrix Factorization (SNMF) objective function with a supervised penalty term, along with gradient descent and multiplicative update rules, to balance interpretability and predictive power in a unified one-step framework. This advances existing two-step approaches from Chapter 3 by potentially improving prediction quality while maintaining chemical interpretability. Such improvements contribute to Quality by Design (QbD) principles in pharmaceutical process monitoring, offering practical tools for researchers in R&D, production, and Process Analytical Technology (PAT).

In Chapter 3, we showed that Partial Least Squares Regression (PLSR) provided superior predictive power for chemical component concentrations compared to the two-step approaches (combining PCA, NMF, or MCR with regression), but at the cost of less interpretable loadings than those from NMF and MCR. Here, we develop and evaluate SNMF as an extension of NMF, incorporating a supervised penalty to optimize both reconstruction and prediction. Specifically, SNMF minimizes the objective function:

$$\min_{\mathbf{H}, \mathbf{W}', \mathbf{B}} \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2 + \frac{\lambda}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2,$$

subject to non-negativity constraints on $\mathbf{H} \geq 0$, $\mathbf{W}' \geq 0$, and $\mathbf{B} \geq 0$, where $\mathbf{X}$ is the spectra matrix (dimensions $T \times m$), $\mathbf{Y}$ the target concentrations (dimensions $T \times n$), $\mathbf{W}$ the loadings (spectral profiles of dimensions $m \times R$), $\mathbf{H}$ the scores (concentration evolutions of dimensions $T \times R$), $\mathbf{B}$ the regression coefficients (of dimensions $R \times n$) and $\lambda > 0$ the supervised penalty weight (see notation table in Section 4.2.1 for dimensions). This improves upon Chapter 3's two-step NMF (first minimizing $\|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2$, then regressing $\mathbf{Y}$ on $\mathbf{H}$) by jointly optimizing decomposition and prediction, potentially enhancing accuracy while preserving interpretability through non-negativity. Using an artificial dataset, we compare SNMF to PLSR and the two-step NMF + regression approach, assessing prediction accuracy via RMSE and interpretability through scores and loadings visualizations. These visualizations qualitatively evaluate chemical meaningfulness, ensuring SNMF does not degrade fit or

interpretability, as further verified by reconstruction errors.

To maintain focus and conciseness, this chapter exclusively considers an artificial dataset, even though two other real cases were tested in (Roy, 2021). Throughout this chapter, we will highlight the most important points of this research, emphasizing the methodological advances and their practical implications in chemometrics modeling. Note that the notations used in this chapter are similar to those in Chapter 3, and both should be seen as a common block. We expected SNMF to enhance prediction accuracy over the two-step NMF + regression from Chapter 3, while preserving non-negativity for interpretable chemical profiles.

4.2 Data set

An artificial dataset will be used in this work to illustrate the methodology. With this example, representative of a typical case in the industry, the aim is to provide an application framework as comprehensive as possible. The following section details it. The notations used are similar to the ones in Chapter 3 (see Table 4.1 for the notations):

Table 4.1: Notations used in Chapter 4

$\mathbf{X}_G$	Matrix of spectra ($T \times m$)
$\mathbf{Y}_G$	Matrix of target concentrations ($T \times n$)
$\mathbf{X}_i$	Matrix of spectra for batch/fold i ($t_i \times m$)
$\mathbf{Y}_i$	Matrix of target concentrations for batch/fold i ($t_i \times n$)
$\mathbf{X}_{G/i}$	Matrix of spectra for all but batch/fold i ($(T - t_i) \times m$)
$\mathbf{Y}_{G/i}$	Matrix of target concentrations for all but batch/fold i ($(T - t_i) \times n$)
Special case:	the target values y are not available for every spectra x
$\mathbf{Y}_G^*$	Submatrix of target concentrations ($S \times m$) with $S < T$
$\mathbf{X}_G^*$	Submatrix of spectra corresponding to the target concentrations ($S \times n$)

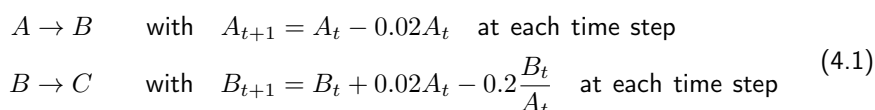
4.2.1 Artificial dataset

The artificial dataset simulates a batch design where reactants are introduced at the start and react over time to produce output chemical components, broadly applicable to pharmaceutical QbD, PAT in real-time reactions, or production.

In the following subsections, the construction of the dataset is thoroughly detailed. The result is a $\mathbf{X}_G$ matrix of dimensions 300×200 with $\mathbf{X}_i$ (50×200) stacked vertically. The target $\mathbf{Y}_G$ is of size 300×3 . Noise is added to the observed spectra but none is added to $\mathbf{Y}_G$.

4.2.2 True quantities

The first step is to define the fictional chemical equations which will generate the observed spectra. In this reaction, 3 components will be present: a reactant **A**, an intermediate reactant **B** and a product **C**.



The reactions will be performed in batches, over a determined number of time steps. Thus, this gives the artificial Equation 4.1. More formally, it yields the iterative update rules for the theoretical true quantities of each component over time, see Equation 4.2.

$$\begin{aligned} \Delta A_t &= 0.02A_t \quad \text{the consumed part of A at each time step} \\ \Delta B_t &= 0.2\frac{B_t}{A_t} \quad \text{the consumed part of B at each time step} \end{aligned} \quad (4.2)$$

$$\text{Hence} \quad \begin{cases} A_{t+1} = A_t - \Delta A_t \\ B_{t+1} = B_t - \Delta B_t + \Delta A_t \\ C_{t+1} = C_t + \Delta B_t \end{cases}$$

These equations generate true quantities as in Figure 4.1, where a speed factor tunes reaction pace across batches, mimicking real variability. At timestep 1, 100 units of component *A* are present, 0 units for components *B* and *C*, and the max time step is set to 300. A speed factor is also added under the form of an additional multiplicative constant such that $\Delta A_t = 0.02A_t \times \frac{300 \times \text{speed_factor}}{\text{max_timestep}}$ and $\Delta B_t = 0.2\frac{B_t}{A_t} \times \frac{300 \times \text{speed_factor}}{\text{max_timestep}}$. Consequently, it allows varying the maximum number of timesteps while keeping the evolution of each component as shown

in Figure 4.1. For instance, in order to ease the computation time, a maximum timestep of 50 might be used instead of 300. In addition, the speed factor will tune the speed of reaction as in a real setup where the components do not react at the same pace for every batch.

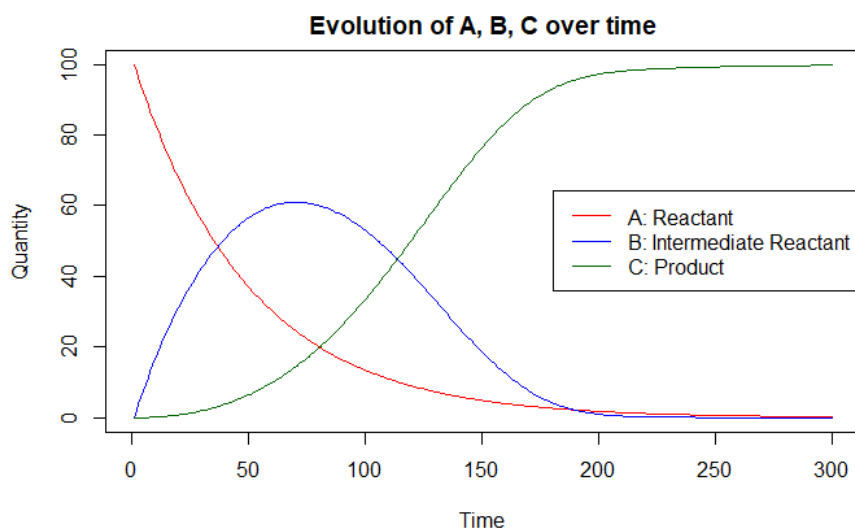


Figure 4.1: Example of evolution of the true quantities over time

4.2.3 True profile

The “true” NIR spectra need then to be calculated and are used as a basis to generate the “observed” spectra (after adding noise and variance). This involves randomly generating three sequences of scalars for each of the three chemical components: (1) the locations of the peaks (μ_i), (2) the variability around those peaks (σ_i), and (3) the intensity factors ($intensity_i$) that scale the amplitude of each peak by multiplying the normalized Gaussian density. A normal density is generated based on the location of each peak and its associated variability. The i -th peak of a component is generated as:

$$peak_i = \frac{intensity_i}{(\sigma_i)\sqrt{2\pi}} \exp\left(-\frac{(x - \mu_i)^2}{2(\sigma_i)^2}\right) \quad (4.3)$$

where $intensity_i$ is the randomly generated scalar determining the peak’s height or amplitude relative to the others.

This gives the true spectra as it can be seen in Figure 4.2. Note that the true profile of each component is normalized such that the area under the curve is 1.

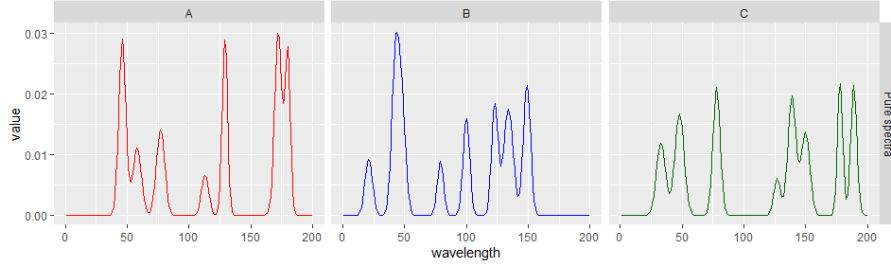


Figure 4.2: True spectra

4.2.4 Observed spectra

Generating observed spectra involves two steps. First, for each component, an approximate profile is created from true parameters (peak location, variability, intensity) with added noise: $e_\mu \sim \mathcal{N}(\text{noise_peak_mean}, \text{noise_peak_sd}^2)$ for peak location, $e_\sigma \sim \max(0, \mathcal{N}(\text{noise_var_mean}, \text{noise_var_sd}^2))$ for variability, and $\text{lognormal noise} \sim \ln \mathcal{N}(\text{noise_lognorm_mean}, \text{noise_lognorm_sd}^2)$ across the spectrum. This reflects natural variability in real spectroscopic data, ensuring distinguishable features.

$$peak_i = \frac{intensity_i}{(\sigma_i + e_\sigma)\sqrt{2\pi}} \exp\left(-\frac{(x - \mu_i + e_\mu)^2}{2(\sigma_i + e_\sigma)^2}\right) \quad (4.4)$$

Second, the spectra for each of the 3 components at timestep t can be combined to give the observed spectra. At each timestep t , the observed spectrum is constructed by combining the contributions of all components. Specifically, for each component, its true spectral profile is scaled by its corresponding true quantity (or concentration) at that timestep. These scaled profiles are then summed to simulate the total spectrum observed at time t . Finally, noise may be added to the resulting spectrum to reflect realistic measurement variability according to Equation 4.4. Then, the spectrum is normalized such that its area under the curve is 1.

The observed spectra for batch number 1 for the case study can be seen in Figure 4.3. Each spectrum is colored following a gradient from blue to yellow,

showing a smooth evolution of the spectra over time (from timestep 1 to 50). The lognormal noise added can also be noticed especially in the leftmost part of the spectral window.

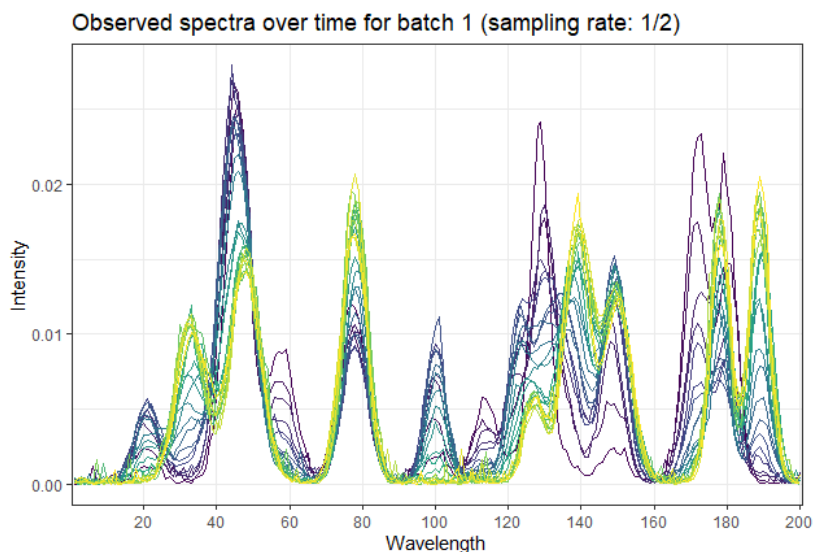


Figure 4.3: Observed spectra for batch 1, where one spectrum is shown at every two timesteps

4.2.5 Batch generation

Having designed functions to automate the creation of the spectra based on the parameters described above, let's now focus on the generation of the batches used in this case study.

The default parameters are as follows: 50 timesteps (yielding 50 spectra per batch), initial concentrations of 100% A and 0% for B and C at timestep 1, 8 peaks per chemical component, a speed factor of 0.9, noise on peak locations $e_\mu \sim \mathcal{N}(0, 0.4^2)$, noise on peak variability $e_\sigma \sim \mathcal{N}(0.3, 0.2^2)$ (with a positive mean to intentionally introduce variability), and background lognormal noise $\sim \ln \mathcal{N}(0, 1)$. These were selected to produce realistic observed spectra.

To simulate industrial variability, six batches are generated with deliberate differences in reaction dynamics and spectral noise, as illustrated in Figure 4.4 corresponding to matrix $\mathbf{Y}_G$ $((6 \times 50) \times 3)$. The first three batches vary primarily in reaction speed: Batch 1 uses a speed factor of 0.9 (faster progression, with A depleting quicker and C accumulating sooner), Batch 2 uses

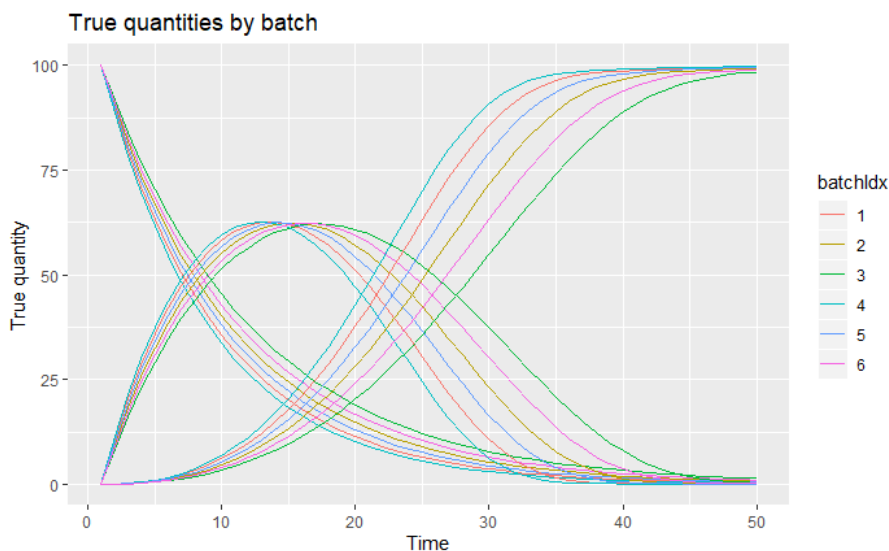


Figure 4.4: True quantities for all batches

0.8 (moderate slowing, leading to delayed B peak and extended intermediate phase), and Batch 3 uses 0.7 (slowest, resulting in higher residual A and lower C at equivalent timesteps). For Batches 4 to 6, the speed factor remains at 0.7, but spectral variability increases via modified noise parameters: $e_{\mu} \sim \mathcal{N}(0, 0.6^2)$ for greater shifts in peak positions (broader spectral features) and $e_{\sigma} \sim \mathcal{N}(0.2, 0.2^2)$ for adjusted peak widths (more subtle shape variations).

This setup creates a gradient of reaction paces and noise levels across batches, mimicking real-world inconsistencies in chemical processes. The first objective is to verify whether the proposed factorization method can recover the underlying true profiles and quantities. The second objective is to evaluate how accurately the supervised NMF method can predict the target response matrix (e.g., compound concentrations) from the spectral data. This predictive performance will later be assessed using cross-validation and compared with alternative modeling approaches (see Sections 4.4 and 4.5).

4.3 Supervised NMF in the literature

As shown in Section 3.3, NMF is a powerful dimensionality reduction technique known for its interpretability. SNMF extends this concept by incorporating supervision through a target matrix, allowing predictions in addition to interpretability. This section summarizes recent literature advancements in the family of SNMF methods.

4.3.1 Supervised NMF for ICU Mortality Classification

The work by Chao et al. (2018) focuses on predicting ICU mortality risk, addressing challenges in modeling complex temporal data. The proposed SNMF integrates a logistic regression term into the objective function:

$$\begin{aligned}
 \min_{\mathbf{H}, \mathbf{W}', b, b_0} \quad & L_1 + L_2 + L_3 \quad \text{s.t.} \quad \mathbf{H} \geq 0, \mathbf{W}' \geq 0 \\
 L_1 = \quad & \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2 \\
 L_2 = \alpha \sum_{i=1}^T \ln \left(1 + \exp(-y_i (\sum_{j=1}^R b_j h_{ij} + b_0)) \right) \\
 L_3 = \quad & \frac{1}{2} \beta \left(\sum_{j=1}^R b_j^2 + b_0^2 \right) + \frac{1}{2} \gamma \|\mathbf{H}\|_F^2
 \end{aligned} \tag{4.5}$$

Here, L_1 corresponds to the classical NMF loss, L_2 is the logistic regression loss for binary classification, and L_3 includes regularization terms to prevent overfitting. The parameters α , β , and γ balance the contributions of these terms and y_i is the univariate binary response variable for the i -th patient.

The authors initialize $\mathbf{H}$ and $\mathbf{W}'$ using Non-negative Double Singular Value Decomposition (NNSVD) (Boutsidis and Gallopoulos, 2008) and derive gradient-based update rules. Positive constraints on $\mathbf{H}$ and $\mathbf{W}'$ are enforced by projecting negative elements to zero.

In Chao et al. (2018) this algorithm was evaluated on simulated and ICU data. Simulated data were generated by adding noise to scores and loadings, and the Area Under the Curve (AUC) was used to measure performance. The proposed method outperformed Fisher Discriminant NMF and standard NMF under varying noise levels. On ICU datasets, the SNMF provided improved AUC scores, although the margin of improvement over classical NMF was modest.

4.3.2 Semi-Supervised NMF for EEG Classification Tasks

Lee, Yoo, and Choi (2010) extended SNMF to account for missing labels, resulting in a Semi-Supervised NMF (SSNMF). The method incorporates a weight matrix $\mathbf{K}$ for the input matrix $\mathbf{X}$ and another weight matrix $\mathbf{L}$ for the target matrix $\mathbf{Y}$. The objective function is:

$$\begin{aligned} \min_{\mathbf{H}, \mathbf{W}', \mathbf{B}} \quad & L_f + L_s \quad \text{s.t.} \quad \mathbf{H} \geq 0, \mathbf{W}' \geq 0 \\ L_f = & \|\mathbf{K} \circ (\mathbf{X} - \mathbf{H}\mathbf{W}')\|_F^2 \\ L_s = & \lambda \|\mathbf{L} \circ (\mathbf{Y} - \mathbf{H}\mathbf{B})\|_F^2 \end{aligned} \quad (4.6)$$

Here, $\mathbf{K}$ and $\mathbf{L}$ are binary matrices, with $K_{ij} = 0$ for missing entries in $\mathbf{X}$ and $L_{i,:} = 0$ for unknown labels in $\mathbf{Y}$. The parameter λ balances reconstruction and prediction, $\mathbf{Y}$ is approximated by $\mathbf{H}\mathbf{B}$ because $\mathbf{H}$ serves as the shared low-dimensional encodings derived from factorizing $\mathbf{X}$, while $\mathbf{B}$ maps these encodings to the labels in $\mathbf{Y}$, allowing partial label information to guide the overall factorization.

The authors derive multiplicative update rules, eliminating the need for learning rates and simplifying parameter selection. They evaluate their method using EEG (electroencephalography) datasets, partitioning data into training and test sets. After applying NMF to the training data, test scores are projected using the learned loadings. Classification is performed using a Support Vector Machine (SVM) trained on the projected scores. The SSNMF demonstrated improved classification accuracy over Fisher Discriminant NMF, highlighting the importance of handling missing labels effectively.

4.3.3 Supervised NMF for MALDI Spectrometry Data Classification

Leuschner et al. (2018) applied SNMF to MALDI imaging data for tumor detection. Their study focused on binary classification and included both unsupervised and supervised extensions to the NMF problem. The general objective function was:

$$\min_{\mathbf{H}, \mathbf{W}'} \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2 + \text{penalty terms} \quad (4.7)$$

Unsupervised Extensions

The unsupervised extensions introduced sparsity (ℓ_1) and orthogonality constraints. For example, the following regularization terms were added:

$$\frac{\lambda}{2} \|\mathbf{W}'\|_1 + \frac{\mu}{2} \|\mathbf{H}\|_F^2 + \frac{\nu}{2} \|\mathbf{W}'\|_F^2$$

Orthogonality was enforced through additional penalties and a new parameter $\mathbf{K}$ to the minimization problem:

$$\frac{\sigma_1}{2} \|\mathbf{I} - \mathbf{W}'\mathbf{K}'\|_F^2 + \frac{\sigma_2}{2} \|\mathbf{K} - \mathbf{W}'\|_F^2$$

These regularizations were designed to extract distinct and interpretable patterns from spectrometry data.

Supervised Extensions

For supervised extensions, the authors incorporated classification penalties, such as linear discriminant analysis or logistic regression, applied to feature data $\mathbf{XW}$ instead of scores $\mathbf{H}$. An example of the classification penalty is:

$$\frac{\gamma}{2} \|\mathbf{Y} - \mathbf{XWB}\|_F^2$$

where $\mathbf{B}$ represents regression coefficients.

The authors tested their approach using 8-fold cross-validation, measuring balanced classification accuracy. Supervised extensions consistently outperformed unsupervised ones. Moreover, the patterns derived from $\mathbf{WB}$ provided valuable insights into spectrometric markers associated with tumors.

4.3.4 VC-PCR for Prediction with Variable Clustering

The work of Marion et al. (2025) introduces Variable Cluster Principal Component Regression (VC-PCR), a two-step prediction method that enhances prediction, variable selection, and clustering in high-dimensional data with cluster structures, such as genomic or spectroscopic datasets. Unlike the classification-focused SNMF methods described above, which integrate supervision directly into the factorization, VC-PCR prioritizes supervised clustering to enhance variable selection in high-dimensional data before applying regression. The methodology begins with supervised variable clustering via Weighted Sparse Orthogonal Semi-Nonnegative Matrix Factorization (SOS-NMF), where a weight matrix $\mathbf{K}$, derived from Ridge or Lasso regression coefficients, encodes the predictive relevance of variables to the response,

optimizing the objective $\min_{\mathbf{H}, \mathbf{W}} \frac{1}{2(n-1)} \|\mathbf{X}\mathbf{X} - \mathbf{H}\mathbf{W}^T\|_F^2 + \lambda \sum_{j=1}^p \|w_j\|_1$ subject to non-negativity on $\mathbf{W}$, orthogonality of $\mathbf{W}$, and unit variance on latent variables $\mathbf{H}$. This supervision improves cluster homogeneity by excluding irrelevant variables, addressing issues like multicollinearity and correlations common in spectral data. The second step computes latent variables from cluster assignments and fits a linear model, similar to PCR, without further sparsity as selection occurs upfront. Optimization alternates updates for $\mathbf{H}$ and $\mathbf{W}$, with experiments showing VC-PCR (especially Ridge-weighted) outperforming competitors in prediction accuracy, selection, and clustering for clustered predictive structures in simulated and real data.

4.3.5 Discussion and Comparison

These studies, including the VC-PCR approach for supervised clustering, highlight different extensions of NMF, focusing on either scores ($\mathbf{H}$) or feature data ($\mathbf{X}\mathbf{W}$) for classification. While Leuschner et al. (2018) demonstrated the utility of using feature data for tumor detection, Chao et al. (2018) and Lee, Yoo, and Choi (2010) used scores for mortality and EEG classification tasks, respectively.

Despite promising results, the studies lack direct comparisons to one-step methods such as PLS-DA. Additionally, the methodology for training and testing SNMF algorithms remains inconsistent, particularly regarding projection steps for new data, because label-based supervision during training cannot apply to label-free new data, resulting in varied methods such as iterative optimization in Chao et al. (2018) and Lee, Yoo, and Choi (2010) versus linear approximation in Leuschner et al. (2018).

The next section explores extensions of the NMF problem for regression, using a sum of squared error penalty. The input will be scores ($\mathbf{H}$), as in Chao et al. (2018) and Lee, Yoo, and Choi (2010), rather than feature data ($\mathbf{X}\mathbf{W}$). This choice leverages the low-dimensional $\mathbf{H}$ matrix derived from the NMF factorization of $\mathbf{X}$, allowing the supervised penalty to focus on essential features and avoiding high-dimensional issues using $\mathbf{X}\mathbf{W}$.

4.4 New algorithm for Supervised NMF

This section proposes a novel SNMF algorithm that jointly optimizes NMF decomposition and supervised prediction. We derive the objective function, gradients, gradient descent (GD) rules, and multiplicative update (MU) rules, including intercept handling. We also discuss non-negativity enforcement and trade-offs for regression coefficients $\mathbf{B}$. Multiplicative update rules are preferred for their convergence guarantees, but hybrids allow flexibility.

4.4.1 Objective function

The usual NMF notation is used in this section. Let us consider a matrix $\mathbf{X}$ containing the T observations and m variables, $\mathbf{X} \in \mathbb{R}_+^{T \times m}$. One obtains the decomposition $\mathbf{X} \approx \mathbf{H}\mathbf{W}'$ with $\mathbf{H} \in \mathbb{R}_+^{T \times R}$ (the scores) and $\mathbf{W} \in \mathbb{R}_+^{m \times R}$ (the loadings) such that $\|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2$ is minimized. At the same time, one wants to minimize the linear regression error between the target variables $\mathbf{Y} \in \mathbb{R}^{T \times n}$, the scores $\mathbf{H}$ and the regression coefficients $\mathbf{B} \in \mathbb{R}^{R \times n}$. Thus, the supervised term $\|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2$ is weighted by λ and added to the objective function. The objective function to minimize is thus:

$$\min_{\mathbf{H}, \mathbf{W}', \mathbf{B}} \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2 + \frac{\lambda}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2 \quad (4.8)$$

For clarity's sake, the following parts are defined: $\Theta = \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2$ and $\Omega = \frac{\lambda}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2$.

4.4.2 Deriving the gradients

The gradients of Θ and Ω are computed to support optimization via gradient-based or multiplicative update rules.

First, the gradient of Θ is derived. Using $\|\mathbf{X}\|_F^2 = \text{Tr}(\mathbf{X}'\mathbf{X})$, Θ can be written as:

$$\begin{aligned} \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2 &= \frac{1}{2} \text{Tr}((\mathbf{X} - \mathbf{H}\mathbf{W}')'(\mathbf{X} - \mathbf{H}\mathbf{W}')) \\ &= \frac{1}{2} \text{Tr}(\mathbf{X}'\mathbf{X} - \mathbf{X}'\mathbf{H}\mathbf{W}' - \mathbf{W}\mathbf{H}'\mathbf{X} + \mathbf{W}\mathbf{H}'\mathbf{H}\mathbf{W}') \\ &= \frac{1}{2} (\text{Tr}(\mathbf{X}'\mathbf{X}) - \text{Tr}(\mathbf{X}'\mathbf{H}\mathbf{W}') - \text{Tr}(\mathbf{W}\mathbf{H}'\mathbf{X}) + \text{Tr}(\mathbf{W}\mathbf{H}'\mathbf{H}\mathbf{W}')) \end{aligned}$$

Using $\nabla_{\mathbf{X}} \text{Tr}(\mathbf{A}\mathbf{X}) = \mathbf{A}'$ and $\nabla_{\mathbf{X}} \text{Tr}(\mathbf{X}'\mathbf{A}) = \mathbf{A}$, the gradients of $\|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2$ with respect to $\mathbf{W}'$ and $\mathbf{H}$ are:

$$\nabla_{\mathbf{W}'} \Theta = \frac{1}{2}(2\mathbf{H}'\mathbf{H}\mathbf{W}' - 2\mathbf{H}'\mathbf{X}) = (\mathbf{H}'\mathbf{H}\mathbf{W}' - \mathbf{H}'\mathbf{X})$$

$$\nabla_{\mathbf{H}} \Theta = \frac{1}{2}(2\mathbf{H}\mathbf{W}'\mathbf{W} - 2\mathbf{X}\mathbf{W}) = (\mathbf{H}\mathbf{W}'\mathbf{W} - \mathbf{X}\mathbf{W})$$

Second, the gradient of Ω is derived. As above, we have,

$$\begin{aligned} \frac{\lambda}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2 &= \text{Tr}\left(\frac{\lambda}{2}(\mathbf{Y} - \mathbf{H}\mathbf{B})'(\mathbf{Y} - \mathbf{H}\mathbf{B})\right) \\ &= \frac{\lambda}{2} \text{Tr}(\mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{H}\mathbf{B} - \mathbf{B}'\mathbf{H}'\mathbf{Y} + \mathbf{B}'\mathbf{H}'\mathbf{H}\mathbf{B}) \end{aligned}$$

The gradients of $\lambda\|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2$ with respect to $\mathbf{B}$ and $\mathbf{H}$ are:

$$\nabla_{\mathbf{B}} \Omega = \frac{\lambda}{2}(2\mathbf{H}'\mathbf{H}\mathbf{B} - 2\mathbf{H}'\mathbf{Y})$$

$$\nabla_{\mathbf{H}} \Omega = \frac{\lambda}{2}(2\mathbf{H}\mathbf{B}\mathbf{B}' - 2\mathbf{Y}\mathbf{B}')$$

These gradients represent the direction to adjust parameters that minimizes the combined reconstruction and prediction errors.

4.4.3 Gradient descent rules

Rules

From the gradients in the above subsection, the (additive) update rules for a Gradient Descent are defined as:

$$\begin{aligned} \mathbf{H} &\leftarrow \mathbf{H} - \gamma_{\mathbf{H}}(\nabla_{\mathbf{H}} \Theta + \nabla_{\mathbf{H}} \Omega) = \mathbf{H} - \gamma_{\mathbf{H}}(\mathbf{H}\mathbf{W}'\mathbf{W} - \mathbf{X}\mathbf{W} + \lambda(\mathbf{H}\mathbf{B}\mathbf{B}' - \mathbf{Y}\mathbf{B}')) \\ \mathbf{W}' &\leftarrow \mathbf{W}' - \gamma_{\mathbf{W}'} \nabla_{\mathbf{W}'}(\Theta) = \mathbf{W}' - \gamma_{\mathbf{W}'}(\mathbf{H}'\mathbf{H}\mathbf{W}' - \mathbf{H}'\mathbf{X}) \\ \mathbf{B} &\leftarrow \mathbf{B} - \gamma_{\mathbf{B}} \nabla_{\mathbf{B}}(\Omega) = \mathbf{B} - \gamma_{\mathbf{B}} \lambda(\mathbf{H}'\mathbf{H}\mathbf{B} - \mathbf{H}'\mathbf{Y}) \end{aligned}$$

Fitting an intercept to $\mathbf{B}$

Like in classical regression, fitting an intercept to the coefficients $\mathbf{B}$ is needed, such that $\tilde{\mathbf{B}} \in \mathbb{R}^{(R+1) \times n}$. The tilde on $\mathbf{B}$ marks a modification from the original matrix by adding an intercept. To do so, a column of 1 is added to the scores $\mathbf{H}$ and a row corresponding to the intercept is added to $\mathbf{B}$, however problems arise with the dimensions as $\tilde{\mathbf{B}} \in \mathbb{R}^{(R+1) \times n}$ and $\tilde{\mathbf{H}} \in \mathbb{R}^{T \times (R+1)}$. The rules change for $\mathbf{H}$ and $\tilde{\mathbf{B}}$ but not for $\mathbf{W}$.

For the update rule of $\mathbf{H}$, the last term is $\tilde{\mathbf{H}}\tilde{\mathbf{B}}\tilde{\mathbf{B}}' - \mathbf{Y}\tilde{\mathbf{B}}'$, where $\mathbf{Y} : T \times n$ and $\tilde{\mathbf{B}} : (R+1) \times n$. The first column of this term is excluded (noted $[-1]$ in the equation). Hence, the scores $\mathbf{H} \in \mathbb{R}^{T \times R}$.

For the update rule of the weights $\tilde{\mathbf{B}}$, a first column full of 1 is added to the scores $\mathbf{H}$ such that $\tilde{\mathbf{H}} \in \mathbb{R}^{T \times (R+1)}$. As a consequence $\mathbf{B} \in \mathbb{R}^{R \times n}$ becomes $\tilde{\mathbf{B}} \in \mathbb{R}^{(R+1) \times n}$.

$$\begin{aligned}\mathbf{H} &\leftarrow \mathbf{H} - \gamma_{\mathbf{H}}(\mathbf{H}\mathbf{W}'\mathbf{W} - \mathbf{X}\mathbf{W} + \lambda(\tilde{\mathbf{H}}\tilde{\mathbf{B}}\tilde{\mathbf{B}}' - \mathbf{Y}\tilde{\mathbf{B}}')_{[-1]}) \\ \mathbf{W}' &\leftarrow \mathbf{W}' - \gamma_{\mathbf{W}}(\mathbf{H}'\mathbf{H}\mathbf{W}' - \mathbf{H}'\mathbf{X}) \\ \tilde{\mathbf{B}} &\leftarrow \tilde{\mathbf{B}} - \gamma_{\mathbf{B}}\lambda(\tilde{\mathbf{H}}'\tilde{\mathbf{H}}\tilde{\mathbf{B}} - \tilde{\mathbf{H}}'\mathbf{Y})\end{aligned}$$

Nonnegativity constraint

A Gradient Descent algorithm will not be able to take the nonnegativity constraints on $\mathbf{H}$ and $\mathbf{W}'$ into account. A projection step will therefore be needed after the update so that every element of $\mathbf{H}$ and $\mathbf{W}'$ remain nonnegative (if an element ≤ 0 it is then replaced by 0.0001). This projection is applied element-wise: $\mathbf{H} \leftarrow \max(\mathbf{H}, 0.0001)$, and similarly for $\mathbf{W}'$, preventing zero-locking where zeros get stuck.

4.4.4 Multiplicative Update rules

Methodology

In a previous section, the Multiplicative Update rules of Lee and Seung (1999) were presented. The idea behind these being that they are equivalent to a rescaled gradient descent. There is another way of understanding the MU which is used to elaborate our own MU rules.

As explained by Lee, Yoo, and Choi (2010), the Multiplicative Update rules can be obtained using another rationale. If the gradient of a function f has a decomposition of the form

$$\nabla_{\alpha} f = [\nabla_{\alpha} f]^+ - [\nabla_{\alpha} f]^-$$

with $[\nabla_{\alpha} f]^+ > 0$ and $[\nabla_{\alpha} f]^- > 0$. Then, MU for parameters α (with $\circ$ being the Hadamard product) has the form

$$\alpha \leftarrow \alpha \circ \left(\frac{[\nabla_{\alpha} f]^-}{[\nabla_{\alpha} f]^+} \right) \quad (4.9)$$

This splitting ensures the update moves in the direction that decreases the objective while preserving non-negativity. Applying Equation 4.9 leads indeed to the MU. This is also put in other words by (Gillis, 2014). He shows (taking for the example the MU for $\mathbf{H}$: $\mathbf{H} \leftarrow \mathbf{H} \circ \frac{[\mathbf{X}\mathbf{W}]}{[\mathbf{H}\mathbf{W}'\mathbf{W}]}$) that

$$\frac{[\mathbf{X}\mathbf{W}]_{ik}}{[\mathbf{H}\mathbf{W}'\mathbf{W}]_{ik}} \geq 1 \quad \Longleftrightarrow \quad (\nabla_{\mathbf{H}} f)_{ik} \leq 0$$

where $\nabla_{\mathbf{H}} f = \mathbf{X}\mathbf{W} + \mathbf{H}\mathbf{W}'\mathbf{W}$. This means that for each element of $\mathbf{H}$, the MU rules either “(i) increase it if its partial derivative is negative, (ii) decrease it if its partial derivative is positive, or (iii) leave it unchanged if its partial derivative is equal to zero” (Gillis, 2014, p. 8). Consequently, the cost function will decrease when using the MU rules.

Rules

From the Section 4.4.2 above, the “parts” of the gradients are:

$$\begin{aligned} \nabla_{\mathbf{W}} \Theta &= [\nabla_{\mathbf{W}} \Theta]^+ - [\nabla_{\mathbf{W}} \Theta]^- \\ &= \mathbf{H}'\mathbf{H}\mathbf{W}' - \mathbf{H}'\mathbf{X} \end{aligned}$$

$$\begin{aligned} \nabla_{\mathbf{H}} \Theta &= [\nabla_{\mathbf{H}} \Theta]^+ - [\nabla_{\mathbf{H}} \Theta]^- \\ &= \mathbf{H}\mathbf{W}'\mathbf{W} - \mathbf{X}\mathbf{W} \end{aligned}$$

$$\begin{aligned} \nabla_{\mathbf{B}} \Omega &= [\nabla_{\mathbf{B}} \Omega]^+ - [\nabla_{\mathbf{B}} \Omega]^- \\ &= \lambda \mathbf{H}'\mathbf{H}\mathbf{B} - \lambda \mathbf{H}'\mathbf{Y} \end{aligned}$$

$$\begin{aligned} \nabla_{\mathbf{H}} \Omega &= [\nabla_{\mathbf{H}} \Omega]^+ - [\nabla_{\mathbf{H}} \Omega]^- \\ &= \lambda \mathbf{H}\mathbf{B}\mathbf{B}' - \lambda \mathbf{Y}\mathbf{B}' \end{aligned}$$

Using these gradients and the methodology in Section 4.4.4, the following MU rules are defined as:

$$\begin{aligned}\mathbf{H} &\leftarrow \mathbf{H} \circ \frac{\mathbf{X}\mathbf{W} + \lambda\mathbf{Y}\mathbf{B}'}{\mathbf{H}\mathbf{W}'\mathbf{W} + \lambda\mathbf{H}\mathbf{B}\mathbf{B}'} \\ \mathbf{W}' &\leftarrow \mathbf{W}' \circ \frac{\mathbf{H}'\mathbf{X}}{\mathbf{H}'\mathbf{H}\mathbf{W}'} \\ \mathbf{B} &\leftarrow \mathbf{B} \circ \frac{\mathbf{H}'\mathbf{Y}}{\mathbf{H}'\mathbf{H}\mathbf{B}}\end{aligned}$$

Fitting an intercept to $\mathbf{B}$

As in the previous section, $\mathbf{B}$ will become $\tilde{\mathbf{B}} \in \mathbb{R}^{(R+1) \times n}$ and $\mathbf{H}$ becomes $\tilde{\mathbf{H}} \in \mathbb{R}^{T \times (R+1)}$.

For the update rule of $\mathbf{H}$, the second term of the numerator is $\lambda\mathbf{Y}\tilde{\mathbf{B}}'$, $\mathbf{Y} : T \times n$ and $\tilde{\mathbf{B}} : (R+1) \times n$, from this term the first column is excluded (noted $[-1]$ in the equation). Same is done for the denominator, after adding a column of 1 to $\mathbf{H}$ for the second term ($\lambda\tilde{\mathbf{H}}\tilde{\mathbf{B}}\tilde{\mathbf{B}}'$, $\tilde{\mathbf{H}} : T \times (R+1)$ and $\tilde{\mathbf{B}} : (R+1) \times k$). The scores $\mathbf{H}$ after the update are thus $\mathbf{H} \in \mathbb{R}^{n \times R}$.

Finally, the weights $\tilde{\mathbf{B}}$ are obtained using $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{B}}$ such that the update rules return $\tilde{\mathbf{B}}$.

$$\begin{aligned}\mathbf{H} &\leftarrow \mathbf{H} \circ \frac{\mathbf{X}\mathbf{W} + (\lambda\mathbf{Y}\tilde{\mathbf{B}})_{[-1]}}{\mathbf{H}\mathbf{W}'\mathbf{W} + (\lambda\tilde{\mathbf{H}}\tilde{\mathbf{B}}\tilde{\mathbf{B}}')_{[-1]}} \\ \mathbf{W}' &\leftarrow \mathbf{W}' \circ \frac{\mathbf{H}'\mathbf{X}}{\mathbf{H}'\mathbf{H}\mathbf{W}'} \\ \tilde{\mathbf{B}} &\leftarrow \tilde{\mathbf{B}} \circ \frac{\tilde{\mathbf{H}}'\mathbf{Y}}{\tilde{\mathbf{H}}'\tilde{\mathbf{H}}\tilde{\mathbf{B}}}\end{aligned}$$

Stopping criterion

Both for the Gradient Descent or the Multiplicative Update rules, the stopping criterion should be defined. A common choice is to measure the objective/loss function at each iteration and stop the algorithm when the difference between consecutive iterations is too low. This is the retained approach for the SNMF algorithm. The parameter measuring this difference is ϵ with the condition: if $|\text{loss}_k - \text{loss}_{k-1}| \leq \epsilon$ then stop the algorithm (k is the index of the iteration number).

Nonnegativity constraint

The Multiplicative Update rules enforce the nonnegativity constraints on $\mathbf{H}$ and $\mathbf{W}$ but they also do so on $\mathbf{B}$. Nevertheless, having only positive coefficients in $\mathbf{B}$ is not necessarily desired since in chemometrics negative coefficients may reflect opposing chemical trends.

Exploring solutions to this problem is necessary. However, using the MU rules for $\mathbf{H}$ and $\mathbf{W}$ and another one relaxing the nonnegativity constraint for $\mathbf{B}$ will lead to the loss of the nonnegativity guarantee on $\mathbf{H}$ and $\mathbf{W}$. Indeed, if elements in $\mathbf{B}$ are negative, negative elements are introduced in the update rule of $\mathbf{H}$, and if $\mathbf{H}$ may contain negative elements, so may $\mathbf{W}$. The projection step will then be necessary if only the MU rules for $\mathbf{H}$ and $\mathbf{W}$ are used. Regarding the update rule of $\mathbf{B}$, a linear regression $\mathbf{Y} \sim \mathbf{HB}$ could be fitted at each iteration (with or without intercept). But then, the update rules are no longer guaranteed to reduce the loss function at each iteration, this is only valid using the MU rules for all parameters. A gradient descent with a carefully chosen learning rate could also be used.

As a conclusion, there is a choice to make between using multiplicative update rules for all parameters, which guarantees a monotonic reduction of the objective function at each iteration but enforces non-negativity on the regression coefficients $\mathbf{B}$, or adopting a hybrid approach (e.g., multiplicative update for $\mathbf{H}$ and $\mathbf{W}'$, and ordinary least squares or gradient descent for $\mathbf{B}$) to allow negative coefficients, at the cost of potentially losing the monotonic convergence guarantee and requiring post-update projections for non-negativity on $\mathbf{H}$ and $\mathbf{W}'$. To guide this choice: opt for full multiplicative update in exploratory analyses where non-negative $\mathbf{B}$ is chemically plausible (e.g., after flipping opposing targets like reactant/product concentrations to align their directions, such as using 100—product concentration) or when convergence speed is critical; use the hybrid for predictive tasks with complex, possibly negative relationships in $\mathbf{B}$.

4.5 Method application on artificial dataset

4.5.1 Methodology

Before analyzing the artificial dataset, a few paragraphs explain first the structure of the algorithms tested and compared, the objective of the application and their implementation in R.

Structure

The organization and objectives of the application are detailed in the next two points. The Figure 4.5 helps to visualize it.

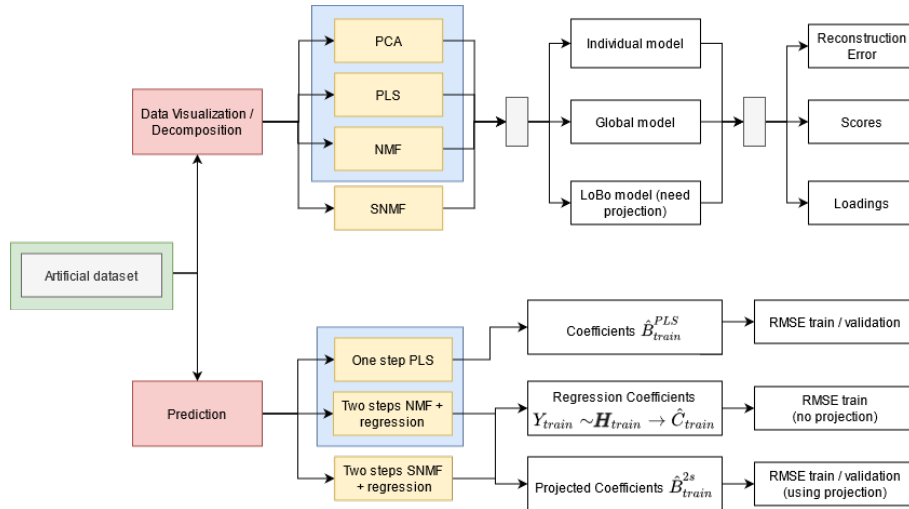


Figure 4.5: Work plan for the case study

1. **Data Visualisation/Decomposition:** the scores and loadings produced from the data matrix are compared for the PCA, NMF, PLS and SNMF. Then, the reconstruction errors are studied. Obviously, the unsupervised methods are expected to perform better at the reconstruction error because it is their only objective. On the other hand, it is interesting to analyse the interpretability of the new algorithm and its reconstruction error compared to a classical NMF and the other methods. This is the reason why the first part of the analysis is not limited to the unsupervised methods only.
2. **Prediction:** the supervised methods are applied and evaluated in cross-validation. The metric used is the RMSE as explained in Section 3.3.8. The

one-step PLS, and the two-steps (S)NMF are compared. Note that PLS produces regression coefficients directly so the RMSE train and validation can easily be obtained.

On the other hand, the first step of the two-steps methods is to produce the OLS regression coefficients $\hat{\mathbf{C}}_{train}$ which result from:

$$\hat{\mathbf{Y}}_{train} = \mathbf{H}_{train} \hat{\mathbf{C}}_{train}$$

We use $\hat{\mathbf{C}}_{train}$ because it provides a direct, interpretable mapping from the extracted scores $\mathbf{H}_{train}$ (factors from the decomposition) to the targets $\mathbf{Y}_{train}$, allowing us to quantify how well the unsupervised features explain the supervised outcomes on the training set itself (each coefficient links a specific chemical factor to a target concentration, aiding for interpretability). While this yields optimal training fit (potentially low RMSE) and potential overfitting to the training data, it serves as a baseline to evaluate decomposition quality before cross-validation with $\hat{\mathbf{B}}^{2s}$. Thus, the training RMSE can be measured on each batch/fold (and not the validation RMSE, since one needs the scores which are only available for the training set).

To obtain regression coefficients applicable to new observations (and not to scores), the extra projection step is required and produces $\hat{\mathbf{B}}_{train}^{2s}$. Then the RMSE train using the projected coefficients can also be calculated from:

$$\hat{\mathbf{Y}}_{train} = \mathbf{X}_{train} \hat{\mathbf{B}}_{train}^{2s}$$

and the RMSE on the validation set (RMSECV) using the projected coefficients results from:

$$\hat{\mathbf{Y}}_{valid} = \mathbf{X}_{valid} \hat{\mathbf{B}}_{train}^{2s}$$

Objective

The objective for the new SNMF algorithm is to have an equivalent reconstruction error as the classical NMF problem (yet, it is expected to be slightly larger since the reconstruction is no longer the only objective of the loss function) and maintain a good interpretability in terms of scores and loadings. This ensures the method remains useful for chemical interpretation in pharmaceutical monitoring.

At the same time, the new algorithm + regression should perform better at predicting $\mathbf{Y}$ compared to a simple NMF + regression model. If it is the case in

cross-validation, it means that the new algorithm is able to capture more relevant patterns to predict the target than the unsupervised NMF.

R implementation

The whole implementation was performed using R (the PCA, NMF, PLS and the proposed Supervised NMF).

- For the PCA, the function `prcomp` was used (from the package `stats` which comes directly with R). The algorithm used by `prcomp` is a SVD.
- For the PLS, the function `pls` from the `pls` package was used. The algorithm chosen was SIMPLS.
- For the NMF, the Brunet algorithm from the `nmf` function from the NMF package is used (recall that it is based on the Kullback-Leibler divergence, not on the Frobenius norm, but it is an enhanced version of the MU algorithm of Lee and Seung (2001)).
- SNMF algorithms were developed in R and are detailed in Julien Van Roy's master thesis (Roy, 2021).

4.5.2 Application to artificial dataset

In this section, the artificial dataset described in Section 4.2.1 is analyzed. Some general parameters for this case study will be fixed as follows:

- The target concentration to predict by the supervised model is the component C (the product).
- For the NMF and the SNMF, 3 components will be used (indeed, it is known by design that there are 3 latent components in the data, matching the reactant, intermediate, and product).
- For the PLS and the PCA, as the data must be first centered, only 2 components will be used.

SNMF parameters

Regarding the SNMF, many parameters are at stake, including the supervised penalty weight λ balancing reconstruction and prediction, initialization methods for $\mathbf{B}$, or use hybrid update rules (e.g., multiplicative for $\mathbf{H}$ and $\mathbf{W}'$, OLS for $\mathbf{B}$). It is chosen to study only 3 different configurations of the SNMF for this case study. Table 4.2 shows the main differences between these models.

SNMF Model Name	init $\mathbf{B}$	update $\mathbf{B}$
SNMF_1: MU with random $\mathbf{B}$ init	random uniform $[0, 1]$	multiplicative
SNMF_2: MU with smart $\mathbf{B}$ init	OLS	multiplicative
SNMF_3: Mixed MU with OLS	OLS	OLS

Table 4.2: 3 different models of SNMF

The regression coefficients $\mathbf{B}$ in the supervised penalty term $\frac{1}{2}\|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2$ can be initialized and updated according to various rules, as explored in these configurations. Having a full Multiplicative Update algorithm for $\mathbf{H}$, $\mathbf{W}$ and $\mathbf{B}$ has the advantage that the loss should decrease at each iteration while at the same time enforcing the non-negativity constraint and having an automatic learning rate. On the other hand, as discussed in the Section 4.4.4, having non-negative $\mathbf{B}$ is not necessarily desirable. Consequently, in addition to the classic MU algorithm '**SNMF1**' with positive $\mathbf{B}$, the second SNMF model '**SNMF2**' enables the coefficients to be negative. Indeed, at initialization, $\mathbf{H}_{init}$ and $\mathbf{W}_{init}$ are the result of an NMF of $\mathbf{X}$, and $\mathbf{B}_{init}$ is set as the OLS regression coefficients of $\mathbf{Y} \sim \mathbf{H}_{init}$, hence possibly negatives. Then, $\mathbf{B}$ is updated by the classic MU rules. Note that there is no longer a guarantee that $\mathbf{H}$ and $\mathbf{W}$ are non-negative since the update rules of $\mathbf{H}$ depends on $\mathbf{B}$, so at each iteration the negative elements of $\mathbf{H}$ are replaced by 0.0001. Then, the '**SNMF3**' model experiments a new way of updating the coefficients $\mathbf{B}$ by setting them to the OLS coefficients from $\mathbf{Y} \sim \mathbf{H}_k$ at each iteration k of the algorithm ($\mathbf{B}_k = (\mathbf{H}_k' \mathbf{H}_k)^{-1} \mathbf{H}_k' \mathbf{Y}$). Again, the negative elements of $\mathbf{H}$ are replaced by 0.0001 at each iteration.

The other SNMF parameters are fixed for all three models. The update of $\mathbf{H}$, $\mathbf{W}$ is done by the MU rules, the maximum number of iterations/epochs is 100, the initialization of $\mathbf{H}$, $\mathbf{W}$ is the result of an NMF on $\mathbf{X}$ with the Brunet algorithm. Regarding the value of λ , the weight for the supervised penalty, a grid search is performed and the mean validation RMSE (the mean RMSECV) are compared in a leave-one-batch-out cross-validation. This tuning ensures a balance between interpretability and predictive performance.

Figure 4.6 shows the evolution of the mean validation RMSE depending on the value of lambda for SNMF1. It can be seen that the lowest mean validation RMSE are obtained with very low values for lambda (thus very low weights on the supervised penalty term). The chosen λ is the one giving the best mean RMSECV. This λ is respectively $1e-5$, $1e-4$, $5e-6$ for the SNMF1, SNMF2 and SNMF3, respectively. The parameter for the stopping criterion ϵ is fixed at

0.001 (when the difference of consecutive objective function is lower than 0.001, the algorithm stops).

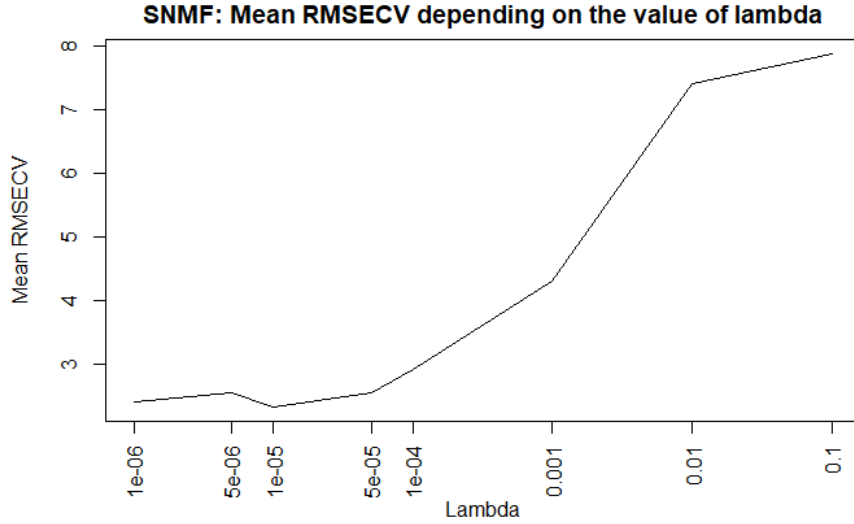


Figure 4.6: Evolution of the mean RMSECV for SNMF1

Analysis of the SNMF algorithms

Before describing the output of the SNMF models, let us focus on the values of the objective function when varying the balancing factor λ . From the objective function of the SNMF (see Equation 4.8), two terms can be distinguished:

- The penalty 1 ('p1') is the NMF loss function: $\frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{W}'\|_F^2$ (measuring spectral reconstruction accuracy)
- The penalty 2 ('p2') is the supervised penalty: $\frac{1}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2$ (measuring prediction error for concentrations)

This analysis highlights the trade-off between these terms across λ values. Figure 4.7 is obtained by fitting global models with different values for λ and measuring the values of 'p1' and 'p2' at the termination of the SNMF1 algorithm. This figure clearly shows the trade-off between the supervised penalty term and the NMF cost function. When λ is small, the NMF loss function 'p1' is small. On the other hand, increasing λ gives more weight to the supervised penalty term ('p2') thus the value of 'p2' is lower but the value of 'p1' increases. Note that the supervised penalty term can take very large values because the coefficients

for the linear regression $\mathbf{Y} = \mathbf{HB} + \mathbf{E}$ may not have converged yet depending on the value of λ . For instance, if λ is very small, the regression coefficients $\mathbf{B}$ at the end of the algorithm may still be close to the random initial ones, therefore making no sense.

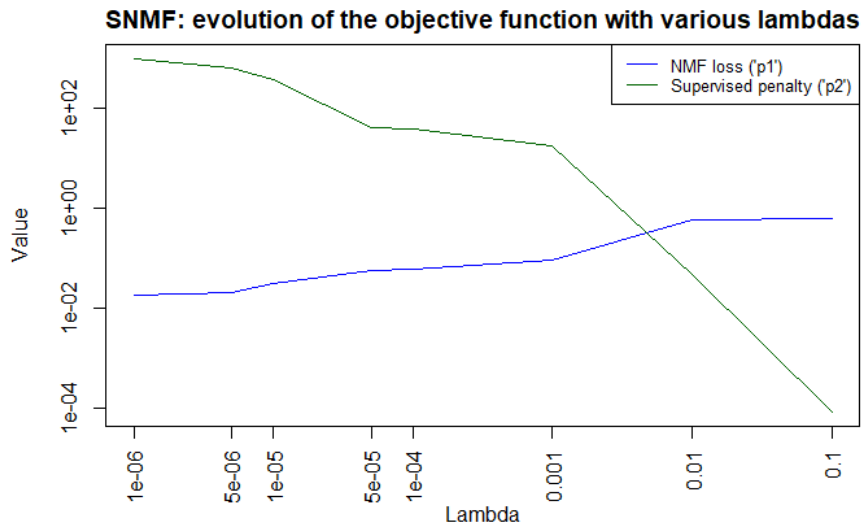


Figure 4.7: Evolution of 'p1' and 'p2' (after the last iteration) for SNMF1 depending on λ (global model)

Let us now focus on the number of iterations and the value of the two terms in the objective function during the algorithm. The analysis is performed for the global model, on all batches. Figure 4.8 shows the evolution of the objective function for the SNMF1 algorithm using the optimal value for λ ($1e-5$).

The NMF loss function starts from a very low value, as it is the result of the Brunet NMF algorithm and then increases, while the supervised penalty and the objective function regularly decrease. The minima for the penalty 1, 2 and the objective function are also shown on the figure through the dots of the same color.

Table 4.3 shows the statistics (value at the first iteration and at the last iteration) for both penalties, the loss function and the number of iterations for the first batch of data. As the SNMF1 starts with a random initialization for $\mathbf{B}$, it is expected that the first value of the penalty 2 is very high. Then, $\mathbf{B}$ is adapted with the MU rules such that the penalty 2 is reduced. Yet, the value of the supervised penalty remains high compared to the SNMF2 and SNMF3. Using

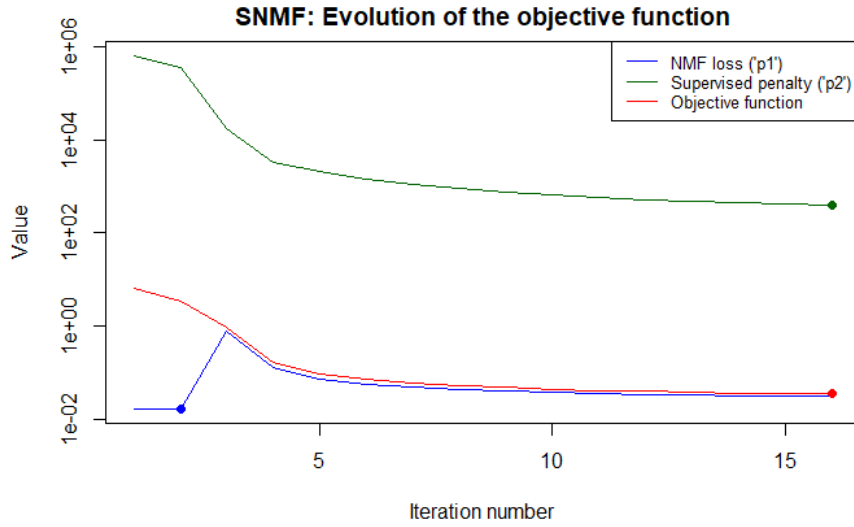


Figure 4.8: Evolution of the objective function for SNMF1 with the optimal λ (global model)

the “clever” initialization for $\mathbf{B}$ for SNMF2 and SNMF3, results in a very low number of iterations. The SNMF2 arrives at a very low value of the supervised penalty using the MU for $\mathbf{B}$, on the other hand SNMF3 reaches a higher value for the same penalty using an OLS update at each iteration.

Overall, the MU rules for $\mathbf{B}$ seem to perform better than using an OLS update at each iteration. In addition, it can be seen that ‘smartly’ initializing the regression coefficients $\mathbf{B}$ reduces the number of iterations needed and can yield lower values for ‘p2’ at the end of the algorithm.

The final value of the loss function is equivalent for SNMF2 and SNMF3 but higher for SNMF1 (both because of a higher NMF cost function and a higher supervised penalty).

	λ	iter	p1.start	p1.end	p2.start	p2.end	loss.start	loss.end
SNMF_1	1e-5	16	0.017	0.031	638538	392.2	6.403	0.035
SNMF_2	1e-4	3	0.017	0.017	164.9	3.467	0.034	0.018
SNMF_3	5e-6	2	0.017	0.017	164.9	29.45	0.018	0.018

Table 4.3: Statistics over the SNMF algorithms for the global model

Recall also that the initialization for $\mathbf{H}$ and $\mathbf{W}$ is the result of a NMF, so a low number of iterations is expected. For the SNMF2 and SNMF3, the results

are thus expected to be very close to the NMF ones. Increasing the number of iterations by using a different λ value had the effect that the RMSECV for the validation sets were higher. It is already a clue that the number of iterations should remain low in order to optimize the RMSECV, in other words, the algorithm struggles to find an optimal decomposition which differs from the original NMF one.

Data visualization and decomposition

This analysis compares decomposition quality across methods, focusing on reconstruction errors, scores, and loadings to assess interpretability and fit (see Figure 4.5).

- Reconstruction error

Table 4.4 shows the reconstruction error (%) for the 3 classical methods: PCA, NMF and PLS. However, the PCA and the NMF are unsupervised methods while the PLS objective also focuses on a target $\mathbf{Y}$, thus the PLS is obviously expected to do worse than the 2 latter methods. The way the reconstruction error is computed for the Individual, Global and Leave-one-Batch-out (LoBo) approaches is explained in Section 3.3.6.

The PCA with 2 components captures 71.7% (first component) and 21.3% (second component) of the variability in the data. Adding a component would only increase this amount by 1.3%. Thus, 2 components seems a reasonable choice for PCA (although, this choice was already made before the analysis, as explained above).

For PCA and PLS, the individual approach has the smallest error, then the global approach, then the leave-one-batch-out (LoBo) has the biggest. It is what is expected since the individual approach fits a model for each batch and then adjust to the specificity of each. The global approach fits only one model for all batches so the specificity of each batch may be lost. The LoBo approach fits a model on all but one batch and then uses the loadings and the data of this batch to obtain projected scores. Thus the reconstruction error is expected to be higher.

However, for NMF, the LoBo error is lower than the individual and the global one. This could be explained by the very high similarity between the batches. Overall, the PCA, NMF and PLS performs roughly equivalently well. The NMF can be highlighted as it exhibits the smallest LoBo error. The PLS also performs almost as well as PCA although its objective also depends on the prediction of the target concentration C .

	PCA			NMF			PLS		
Batch	Individual	Global	LoBo	Individual	Global	LoBo	Individual	Global	LoBo
1	0.748	0.749	0.757	0.766	0.773	0.756	0.748	0.755	0.757
2	0.718	0.714	0.727	0.735	0.743	0.727	0.718	0.725	0.727
3	0.675	0.669	0.689	0.691	0.703	0.688	0.675	0.685	0.688
4	1.396	1.428	1.411	1.428	1.435	1.404	1.402	1.408	1.411
5	1.352	1.37	1.361	1.384	1.389	1.356	1.355	1.36	1.362
6	1.288	1.295	1.298	1.319	1.325	1.294	1.289	1.296	1.298
mean	1.029	1.037	1.041	1.054	1.061	1.037	1.031	1.038	1.041

Table 4.4: Error percentage per batch and methodology (%)

Table 4.5 shows the reconstruction error for the 3 SNMF models. SNMF1 shows the worse reconstruction error for the individual and global approach, but its LoBo error is on the other hand roughly equivalent to the other methods. The SNMF2 and SNMF3 are very close in terms of error and it is also expected since they both have a low number of iterations starting with the results of a NMF and taking into account the additional supervised penalty term. However, the SNMF2 and SNMF3 errors compared to the NMF errors are almost equal, meaning that the quality of the reconstruction is not significantly affected by this new supervised term. In contrast, for SNMF1, which has more iterations, this effect is noticeable but only for the individual and the global approach.

	SNMF1			SNMF2			SNMF3		
Batch	Individual	Global	LoBo	Individual	Global	LoBo	Individual	Global	LoBo
1	1.027	1.55	0.78	0.77	0.778	0.753	0.765	0.773	0.754
2	1.016	1.566	0.75	0.739	0.746	0.723	0.734	0.741	0.724
3	0.994	1.592	0.711	0.693	0.703	0.684	0.689	0.698	0.685
4	1.686	2.206	1.435	1.428	1.439	1.404	1.423	1.432	1.404
5	1.648	2.193	1.387	1.386	1.391	1.355	1.379	1.384	1.355
6	1.603	2.182	1.327	1.316	1.327	1.292	1.313	1.319	1.292
mean	1.329	1.881	1.065	1.056	1.064	1.035	1.05	1.058	1.036

Table 4.5: Error percentage per batch and methodology (%)

Finally, the errors of each batch are in direct relation with the construction of the observed spectra. Recall that the batch 4, 5, 6 have more variability on the position of their peaks. The reconstruction error for each batch is then probably just reflecting the artificially added error which is independent of the pattern in the data.

- Scores and Loadings

Let us now turn to the scores and loadings produced by the methods. Figure 4.9 shows the true quantities of each component (A, B, C) used to generate the data and the scores for the individual model for the first batch, for the PCA and the PLS with 2 components. The similarity between the two models is striking, as the first component is roughly the same and mimics the evolution of the product (C), while the second captures the evolution of the intermediate reactant B and probably the reactant A. The second component for PLS is just the symmetric as the PCA one. In both models, the scores are not smooth and then capture a part of the artificial noise added to the observed data.

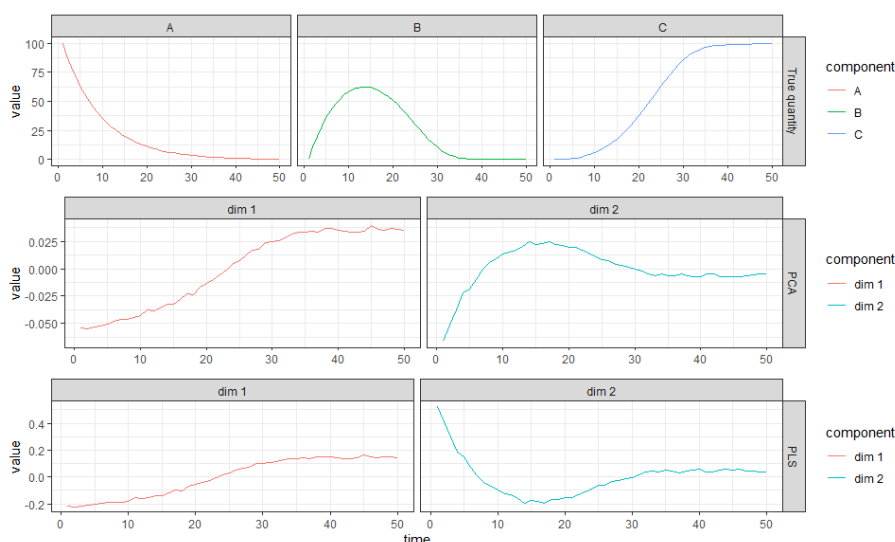


Figure 4.9: PCA and PLS scores for batch 1: individual model

Then, Figure 4.10 shows the true quantities of each component and the scores for the individual model of the batch 1 for the NMF and the SNMF1 (the one which differs the most from the initial NMF) with 3 components. Here, the NMF almost perfectly captures each chemical component through each of its dimensions. The SNMF1 also does, but in addition it also gets closer to the true scale of each chemical component, see the difference between the third component for NMF and SNMF1. As for the previous models, noise is also captured by NMF and SNMF1 in the scores.

Figure 4.11 shows the loadings for the individual model for the batch 1. The similarity between the PCA and the PLS is very noticeable: the first component seems the same and the second for PLS is the symmetric of

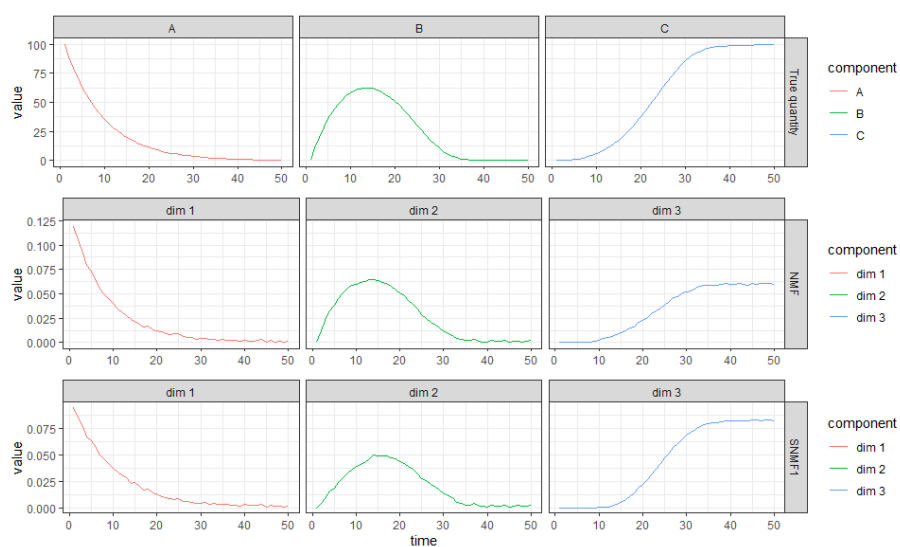


Figure 4.10: NMF and SNMF1 scores for batch 1: individual model

the second for PCA (but the intensities diverge). This overlap suggests shared latent structures despite different objectives.

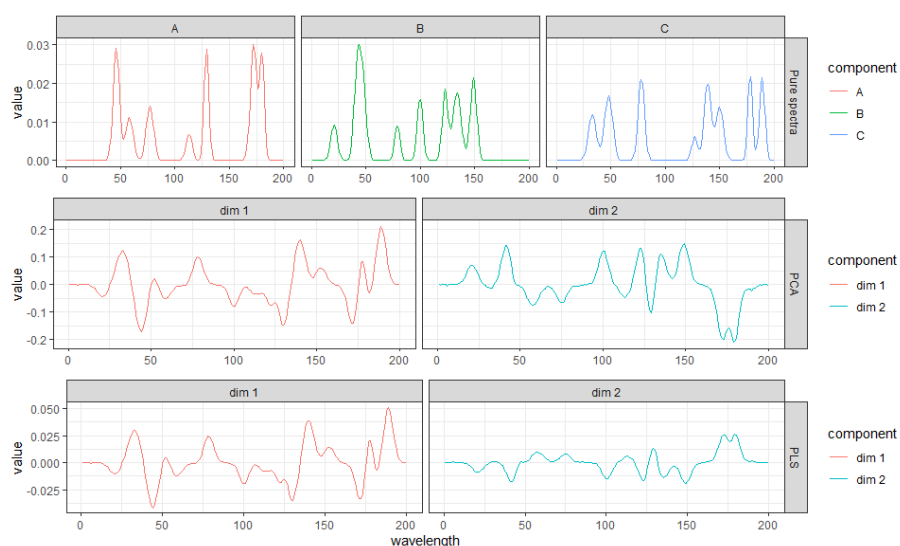


Figure 4.11: PCA and PLS loadings for batch 1: individual model

Figure 4.12 shows the loadings for the individual model for batch 1 for NMF and SNMF1. In both cases, the loadings of each component clearly corresponds to peaks of the true spectra. A bit of noise is captured in the loadings for instance in the dimension 1 and 2 of NMF and SNMF1. The scale between the 2 models' loadings differs: the dimension 1 in SNMF1 is closer to the true spectra regarding the intensity of each peak, the dimension 2 intensity in SNMF1 is in contrast too wide, and the dimension 3 in SNMF1 is very close to the true scale of the true spectra of the component C. Thus, as for the scores, there seems to be an interacting scaling effect for the loadings of the SNMF compared to the NMF. However the peaks are the same for both models.

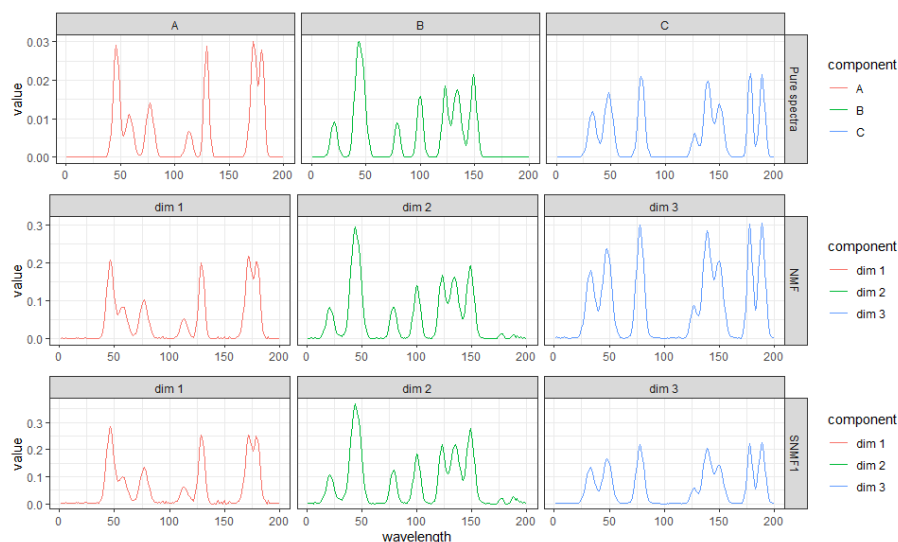


Figure 4.12: NMF and SNMF1 loadings for batch 1: individual model

- NMF VS SNMF

Figure 4.13 shows the difference between the NMF and the SNMF1 results for the individual model for the batch 1. The difference in the loadings is mainly on the intensity of each peak but the peaks identified are the same. The intensity for the dimension 1 and 2 is increased but reduced for the dimension 3 (the dimension representing the response variable). Regarding the scores, an extra black curve represents the evolution of the true quantities (divided by a factor 1200) represented by the matching dimensions. It can be seen that for the dimension 3 (representing the product, the target used for fitting the SNMF algorithm), the SNMF scores

is shifted such that it better matches the true quantity of the product. The scores for the dimension 1 and 2 are also (but more slightly) shifted towards the true quantities. In summary, SNMF maintains NMF's interpretability while slightly enhancing scale alignment to true quantities.

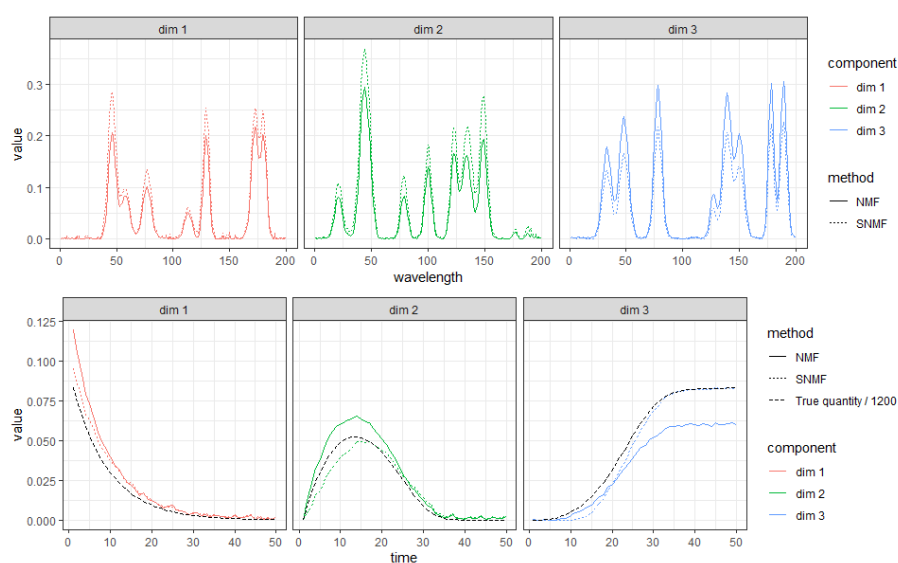


Figure 4.13: SNMF1 versus NMF for batch 1: individual model

Prediction

In this subsection, the prediction accuracy for the product C is investigated. The models used are the one-step PLS, the two-step NMF + regression and the SNMF. The RMSECV are compared as well as the final regression coefficients over the variables (the wavelengths). Details on the RMSECV calculation can be found in Section 3.3.8.

- RMSE

Recall that two different predictions can be produced for the training set for the two-step model: see Figure 4.5 and Section 3.3.8. Hence, two different RMSE train can be obtained for the two-step methods: using $\hat{\mathbf{C}}_{G/i}$ or using $\hat{\mathbf{B}}_{G/i}^{2s}$. Note that, in the tables below, each line corresponds to one fitted model. For instance, to predict batch 1, a model is fitted on batches 2 to 6 and the RMSE validation is the error of prediction on the batch 1 (written in the tables on the line 'Pred. Batch 1'). On the other hand, the RMSE train is the error of this model when predicting the batches 2 to 6, but the result is also written on the same line as the batch 1. So, the line X should be read as "the training and validation RMSE when predicting the batch X".

Table 4.6 shows first the RMSECV for the one-step PLS, the two-step NMF and the SNMFs models. For the PLS with 2 components, the mean validation RMSE is 1.907 and the training RMSE is slightly lower, but not much overfitting is observed. The batches 4 to 6 seem harder to predict: the validation RMSE is higher and the training RMSE is lower, meaning that predicting one of those batches (and hence disregarding it for the training set) yields better training errors. For the NMF + regression, the validation RMSE is higher (2.01) as well as the training RMSE using $\hat{\mathbf{B}}_{G/i}^{2s}$ (1.993), again not much overfitting is observed.

The training RMSE using $\hat{\mathbf{C}}_{G/i}$ is however much lower than the training RMSE using the coefficients of PLS, but this comes at a price: the coefficients $\hat{\mathbf{C}}_{train}$ are optimal (via OLS) for the specific training set only, meaning they may overfit and not generalize well to new data. Despite this limitation, they are easily interpretable, as each coefficient directly corresponds to one latent component (factor) from the decomposition, revealing how each extracted chemical profile contributes to the target. For instance, when predicting the first batch, an NMF is fitted on batches 2 to 6, producing scores and loadings. The coefficients $\hat{\mathbf{C}}_{G/i}$ (with $\tilde{\mathbf{Y}}_{G/i} = \mathbf{H}_{G/i}\hat{\mathbf{C}}_{G/i}$) are 2.448, -5.180, and 1451.676 for the three score

vectors, respectively. Since the target is the product concentration, this indicates that it mainly depends on the third dimension of the scores and is proportional to it, providing chemical insight into the dominant factor.

For the SNMFs models, among the batches, the lower training errors are mostly observed when predicting the batches 4 to 6. Training errors using $\hat{\mathbf{C}}_{G/i}$ widely differ but are always lower than for the two-step NMF (and this is expected since it is in the objective function). Using the coefficients $\hat{\mathbf{B}}_{G/i}^{2s}$, the training and validation errors are a bit higher than for the PLS. However, the two-step SNMF2 and SNMF3 show better training and validation errors than the two-step NMF.

One also notices the difference between the training errors using $\hat{\mathbf{C}}_{G/i}$ for SNMF2 and SNMF3. Indeed, the lowest is achieved for SNMF2, which used MU rules for the coefficients instead of OLS at each iteration like SNMF3. In this sense, the MU rules for the coefficients seem to be more useful for finding optimal training regression coefficients (recall that the MU rule for the scores depends on the regression coefficients, thus the scores might have been adapted in SNMF2 in a different way than in SNMF3 and yield better regression coefficients).

As a conclusion, the two-step NMF is slightly outperformed by the two-step SNMF2 and SNMF3 (with low number of iterations) but not by SNMF1 (full MU with random init for $\mathbf{B}$). None of the models beats the PLS for the training and validation RMSE (using the regression coefficients $\hat{\mathbf{B}}_{G/i}^{PLS}$ versus $\hat{\mathbf{B}}_{G/i}^{2s}$). This underscores SNMF's value in balancing prediction and interpretability.

Table 4.6: Comparison of one-step PLS, two-step NMF + regression, and two-step SNMF + regression: RMSECV

Pred. Batch	PLS			NMF + regression			SNMF1			SNMF2			SNMF3		
	Train	Valid		Train ($C_{G/i}$)	Train ($B_{2s}^{G/i}$)	Valid ($B_{2s}^{G/i}$)	Train ($C_{G/i}$)	Train ($B_{2s}^{G/i}$)	Valid ($B_{2s}^{G/i}$)	Train ($C_{G/i}$)	Train ($B_{2s}^{G/i}$)	Valid ($B_{2s}^{G/i}$)	Train ($C_{G/i}$)	Train ($B_{2s}^{G/i}$)	Valid ($B_{2s}^{G/i}$)
1	1.929	1.715		1.066	2.026	1.835	0.966	2.367	2.233	0.079	2.023	1.813	0.361	2.020	1.829
2	1.932	1.675		1.089	2.035	1.779	0.973	2.329	2.169	0.095	2.032	1.761	0.364	2.030	1.771
3	1.932	1.669		1.071	2.043	1.735	0.969	2.320	2.183	0.096	2.040	1.723	0.363	2.037	1.731
4	1.856	2.073		1.054	1.954	2.188	0.963	2.303	2.383	0.081	1.950	2.190	0.357	1.948	2.176
5	1.851	2.091		1.050	1.951	2.204	0.963	2.288	2.429	0.080	1.948	2.209	0.357	1.946	2.195
6	1.838	2.147		1.055	1.943	2.246	0.949	2.338	2.590	0.070	1.937	2.242	0.348	1.936	2.244
mean	1.890	1.907		1.064	1.993	2.010	0.964	2.324	2.336	0.084	1.989	2.002	0.358	1.987	2.003

- Regression coefficients

Finally, the regression coefficients $\hat{\mathbf{B}}_{G/i}^{2s}$ (from the two-step models: NMF or SNMF followed by regression) or $\hat{\mathbf{B}}_{G/i}^{PLS}$ (from PLS) are compared in Figure 4.14 below. These coefficients are obtained by fitting each model on batches 2 to 6 and using them to predict the held-out batch 1, allowing a direct comparison of how each method weights the spectral variables (wavelengths) for prediction. The same peaks appear across all models, indicating consistent identification of important spectral features. However, the peak intensities differ: the PLS coefficients are always equal to or lower than those from the two-step models. This difference may arise because the two-step models incorporate non-negativity constraints on the scores $\mathbf{H}$ and loadings $\mathbf{W}'$, which can amplify certain weights to satisfy these constraints while preserving interpretability. Among the SNMF variants, SNMF1 shows slight differences from the others, but overall, the coefficients for SNMF2, SNMF3, and NMF are very similar—this is expected, as SNMF2 and SNMF3 use few iterations (starting close to NMF results), limiting divergence from the unsupervised NMF.

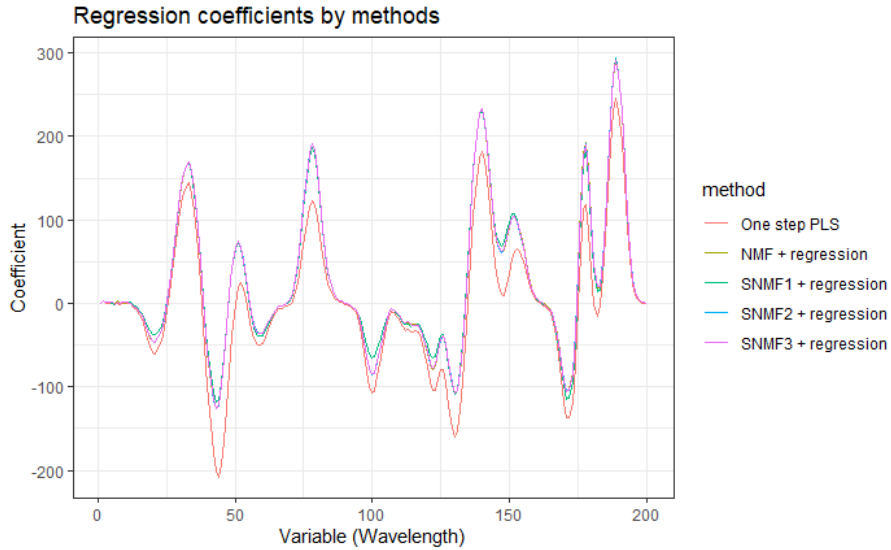


Figure 4.14: Regression coefficients on the variables for the different methods

4.6 Conclusion

In this chapter, the starting point was to monitor chemical reactions over time using on-line methods such as infrared spectra. In this context, the usefulness of NMF was shown for the convenience of the interpretation of the scores and loadings it produced, due to its non-negativity constraint. After a review of the existing SNMF algorithms, a new SNMF objective function with an extra supervised penalty term was proposed as well as gradient descent and multiplicative update rules. This improves existing two-step approaches by unifying steps, enhancing accuracy marginally while preserving interpretability, as motivated in the introduction.

The extra step required to obtain regression coefficients from the scores and loadings of a dimensionality reduction technique such as NMF or SNMF was explained. Hence, it allows to compare multiple algorithms (one-step PLS and two-step (S)NMF) and the accuracy of their prediction for the target variable in a cross-validated scheme.

Then, the artificial dataset was analyzed in the Section 4.5, following a precise methodology. The three SNMF algorithms were fitted and tuned, then the loadings, scores and the reconstruction errors of PCA, (S)NMF and PLS were compared. The prediction accuracy and the regression coefficients of the one-step PLS and the two-step (S)NMF were also shown.

The main conclusions are as follows. First, the reconstruction errors obtained are generally the lowest for PCA or PLS, but those of the NMF and SNMF remain small. Second, the scores and loadings for PLS and PCA show important similarities, and widely differ from the interpretable scores and loadings from (S)NMF. Then, the prediction accuracy in leave-one-batch-out cross-validation was always the best using PLS. On its side, the two-step NMF performs less well but the errors stay reasonable (but recall that the data is centered prior to the PLS). The two-step SNMF is usually able to reduce the training prediction error and translates into a reduction of the validation errors for the artificial data. The comparison of the regression coefficients showed that one-step PLS and the two-step (S)NMF produced similar ones (the same peaks were always identified, only the intensity of those peaks varied). Overall, NMF is shown to work really well for spectral datasets and seems difficult to improve. It is also probably impossible to reach the same regression coefficients as the PLS, as the scores and loadings of (S)NMF are constrained. However, supervising NMF remains valuable for unified modeling, especially when both

prediction and explanation are required.

This chapter summarizes the work from Julien Van Roy's master's thesis (Roy, 2021), without adding new results, but integrates it into the broader PhD context by building directly on the unsupervised NMF approaches developed in Chapter 3. While the artificial dataset focuses on batch processes with a simple reaction (one reactant, intermediate, and product), SNMF has potential in more complex scenarios, such as reactions with multiple reactants or nonlinear kinetics, where the non-negativity constraint could still provide interpretable profiles.

Finally, the trade-off between the accuracy of the prediction and the interpretation of the scores and loadings remains, although the two-step NMF of Chapter 3 can be used to obtain reasonable prediction errors. Further work could include the search for a new two-step method to construct the regression coefficients. A search for new initializations for the scores and loadings and new objective functions could also be undertaken. It may also be useful to analyze the effect of adding noise to the spectra when decomposing them. Nevertheless, the results are expected to be only marginally better, suggesting SNMF's one-step framework is a practical advancement without necessitating further complexity in many cases. Taking a step back, this work provides a tool that advances chemometrics for real-world pharmaceutical monitoring, aligning with QbD by enabling interpretable, predictive models for process optimization.

CHAPTER 5

General conclusions

This thesis aimed to advance two fundamental questions in the field of chemometrics, driven by practical challenges encountered in industrial and scientific environments. Following those challenges, a collaboration was initiated between Janssen Pharmaceutica and ISBA at UCLouvain. The first question focused on disentangling experimental factor effects in designed studies, especially in omics or spectroscopic datasets. The second one concerned the development of interpretable and predictive models for monitoring pharmaceutical chemical reactions. By integrating theoretical extensions, software tools, and novel algorithms, this work not only resolved these issues but also offers practical solutions applicable to broader domains, such as R&D, analytical chemistry, and general process control. In the following lines, we synthesize the key findings, structured by challenges (Question 1: Chapters 1–2 on ANOVA-based multivariate methods; Question 2: Chapters 3–4 on reaction monitoring), evaluate how the initial problems have been addressed, highlight industrial impacts, and propose avenues for future research. Throughout, we emphasize the strengths and weaknesses of the proposed methods to guide their application.

5.1 Question 1: How to analyze multivariate data from designed experiments?

The first question, addressed in Chapters 1 and 2, explored the analysis of multivariate responses from experimental designs. Chapter 1 introduced ASCA+ and APCA+, extensions of ASCA and APCA, developed for unbalanced and complex multifactorial designs. ASCA+ overcame the initial limitations by using least squares estimators within the General Linear Model framework to decompose a multivariate response matrix into effect matrices. This step was followed by PCA for visualization in a reduced-dimensional space. Permutation tests and Type III sums of squares ensured robust statistical inference, quantifying the significance and variance explained by each experimental factor. Application to the *UCH* dataset demonstrated ASCA+ capability to disentangle factor effects in metabolomics, offering substantial improvements over classical ASCA. As seen in the different chapters, this methodology paved the way for advancements in other classes of designs. The strengths of ASCA+ and APCA+ include their ability to handle both categorical and continuous factors, providing clear variance decomposition and statistical inference via permutation and bootstrap tests. ASCA+ excels in isolating pure effects, making it ideal for hypothesis testing in designed experiments, while APCA+ incorporates residuals into PCA decompositions, offering a more informative view. However, weaknesses arise in scenarios with very large residuals, where APCA+ may mask

subtle factor effects, and permutation tests can be computationally intensive for large datasets.

Chapter 2 translated these theoretical advancements into a practical tool by developing the R package `limpca`, a collaborative effort between industry and academia contributors. The package automated the decomposition of multivariate response matrices, implemented ASCA+ and APCA+, and incorporated innovative features such as extensions to continuous factors, confidence ellipses for visual statistical inference, and measures of variable importance. Applications to the *UCH* and *Sacurine* datasets showcased the package's versatility in handling both categorical and continuous variables in experimental and observational studies. Its availability on Bioconductor further enhances its accessibility, meeting the scientific need for user-friendly tools implementing ASCA methodology. `limpca`'s strengths lie in its integration of visualization tools (e.g., score plots with confidence ellipses) and extensions for variable importance, which enhance interpretability and handle multiple testing corrections effectively. Weaknesses include the current lack of random effects modeling, which limits its use to more complex cases like longitudinal data.

The industrial and scientific impact of Question 1 cannot be overstated. In pharmaceutical R&D, where designed experiments often involve spectroscopic data from PAT tools, ASCA+ enables precise process understanding by isolating factor influences (like temperature or concentration) on multivariate responses. For example, in metabolomics applications such as the *Sacurine* dataset, the package's extension to continuous factors (e.g., age as a scaled variable) allowed for nuanced modeling of gradual effects, offering insights into age-gender interactions. While not yet adopted in industry, its open-source nature and integration with Bioconductor position it for uptake.

5.2 Question 2: How to analyze data issued from multivariate process monitoring over time?

The second question was addressed in Chapters 3 and 4, which focused on real-time monitoring of chemical reactions using spectroscopic data, a key aspect of PAT in pharmaceutical manufacturing. Chapter 3 examined unsupervised chemometric approaches (PCA, NMF, and MCR) to visualize and interpret reaction kinetics. Using a real case study of 11 experiments monitored via mid-infrared spectroscopy and UPLC, the chapter showed that NMF and MCR, with their non-negativity constraints, produced loadings similar to pure

spectra and scores aligning with UPLC profiles. In parallel, the chapter evaluated prediction strategies using both one-step PLS and two-step regression based on extracted components. In practice, those calibration models allow to predict in real time, and continuously, the chemical concentrations of a new reaction without having to sample off-line measurements. Strengths of NMF and MCR include their non-negative decompositions, which yield physically meaningful results, outperforming PCA in interpretability for chemical spectra. PCA's strength is its computational efficiency making it suitable for initial exploratory analysis, but its weakness in this context (negative loadings/scores) hampers chemical intuition. Predictive modeling was assessed through one-step PLSR and two-step regressions on decomposed scores, with PLSR excelling in accuracy (lower RMSECV) due to its direct supervision, but lacking the mechanistic insights of NMF/MCR. In alignment with Quality by Design principles, those different methods allow the visualization of reaction progress, anomaly detection, and facilitate process control without prior knowledge.

Chapter 4 extended this work by developing Supervised Non-Negative Matrix Factorization, a novel method that incorporated a supervised penalty term to balance prediction accuracy and chemical interpretability. Tested on an artificial dataset simulating a three-component chemical reaction, SNMF was compared to PLSR and two-step NMF + regression approaches. Although PLSR achieved the lowest prediction errors, SNMF provided a valuable trade-off by retaining interpretable loadings and scores that closely resembled true spectral profiles and component concentrations. The chapter detailed SNMF's algorithmic implementation, including gradient descent and multiplicative update rules. SNMF's strengths include the improvement of predictive power over unsupervised NMF while preserving non-negativity for interpretability. However, weaknesses involve sensitivity to the supervision parameter λ , which could degrade the NMF fit, and higher computational demands than PLSR, which remains more accurate for pure prediction tasks. Overall, these findings laid the groundwork for hybrid modeling approaches that combined the predictive strength of PLSR with the interpretability of NMF, enhancing real-time release testing and process control in pharmaceutical manufacturing.

5.3 Overall contributions and future directions

Overall, this thesis resolves the introduction's goals by providing tools to handle multivariate data from designed experiments and chemical reactions monitored over time. In the first question, ASCA+ and `limpca` empower researchers in

experimental designs and observational studies, such as encountered in omics, while in the second question, the methods advance PAT in pharmaceutical production. Contributions to QbD are evident: ASCA+ aids in identifying critical process parameters, and SNMF supports risk-based control by predicting deviations early. Beyond PAT, this work is applicable to R&D, analytical chemistry, and generally to all multivariate analyses.

Several avenues are opened for future research. The extension of ASCA+ to generalized linear models, noted in Chapter 1, and integrating mixed-effect models with ASCA+, as explored in recent developments, could be integrated in the package `limpca`. For SNMF, future efforts could focus on optimizing initialization strategies, exploring alternative objective functions, or improving robustness to noise in real-world spectroscopic data. In summary, the advancements proposed in this thesis not only resolved specific questions but also establish a foundation for future innovations in the field of chemometrics.

CHAPTER 6

General Appendices

6.1 Appendix A: Abbreviations

ACLS		Alternating Constrained Least Squares
ALS		Alternating Least Squares
ANLS		Alternating Nonnegative Least Squares
ANOVA		Analysis of Variance
ASCA		ANOVA-Simultaneous Component Analysis
APCA		ANOVA-Principal Component Analysis
BMI		Body Mass Index
EEG		Electroencephalogram
GC		Gas Chromatography
GKLD		Generalized Kullback–Leibler divergence
GLM		General Linear Model
HALS		Hierarchical Alternating Least Squares
ICU		Intensive Care Unit
LC-HRMS		Liquid Chromatography High-Resolution Mass Spectrometry
LoBo		Leave-one-Batch-out
LMM		linear mixed model
MALDI		Matrix-Assisted Laser Desorption/Ionization
MANOVA		Multivariate Analysis of Variance
MIR		Mid-Infrared (spectroscopy)
MSI		Mass Spectrometry Imaging
MU		Multiplicative Update (rules)
NIR		Near-Infrared (spectroscopy)
NMF		Non-negative Matrix Factorization
NNLS		Nonnegative Least Squares
NNSVD		Nonnegative Double Singular Value Decomposition

OLS		Ordinary Least Squares
OPLS		Orthogonal Partial Least Squares
PCA		Principal Component Analysis
PLS		Partial Least Squares
RMSECV		Root Mean Squared Error in Cross-Validation
RMSE		Root Mean Squared Error
SNMF		Supervised Non-negative Matrix Factorization
SVD		Singular Value Decomposition
SED		Squared Euclidean Distance
UPLC		Ultra-Performance Liquid Chromatography

6.2 Appendix B: Mathematical notations

6.2.1 Notations used in Chapters 1 & 2

$\mathbf{Y} \in \mathbb{R}^{N \times m}$	Response matrix containing multivariate data
$\mathbf{X} \in \mathbb{R}^{N \times p}$	Design matrix representing experimental factors
$\boldsymbol{\Theta} \in \mathbb{R}^{p \times m}$	Parameter matrix estimated using General Linear Models (GLM)
$\mathbf{E} \in \mathbb{R}^{N \times m}$	Residual matrix
$\mathbf{M}_f \in \mathbb{R}^{N \times m}$	Effect matrix for a specific factor f in the model
$\ \cdot\ _F^2$	Frobenius norm of matrix
SS_f	Sum of squares associated with effect f in ANOVA decomposition
$\%Var_f$	Percentage of variation explained by effect f
$\mathbf{T}_f, \mathbf{P}_f$	Score and loading matrices obtained from PCA applied to effect matrices
f	Index for a model effect (e.g., A, B, or AB)
N	Total number of observations
p	Number of model parameters
m	Number of response variables

6.2.2 Notations used in Chapter 3 & 4

$\mathbf{X} \in \mathbb{R}^{T \times m}$	Input data matrix containing T observations and m variables
$\mathbf{H} \in \mathbb{R}^{T \times R}$	Scores matrix
$\mathbf{W}' \in \mathbb{R}^{R \times m}$	Loadings matrix
R	Number of components (rank of the factorization)
$\mathbf{Y} \in \mathbb{R}^{T \times n}$	Response matrix containing T observations and n nvariables
$\mathbf{B} \in \mathbb{R}^{m \times n}$	Regression coefficient matrix
$(.)'$	Transpose of a matrix, $(A')_{ij} = A_{ji}$

$\ \cdot\ _F$	Frobenius norm, $\ A\ _F = \sqrt{\sum_{i=1}^n \sum_{j=1}^p A_{ij}^2}$
$\circ$	Component-wise multiplication, $(A \circ B)_{ij} = A_{ij}B_{ij}$
$\frac{[\cdot]}{[\cdot]}$	Component-wise division, $\left(\frac{[A]}{[B]}\right)_{ij} = \frac{A_{ij}}{B_{ij}}$
∇f	Gradient of the function f
$\mathbf{1}_{n \times p}$	Vector of ones of dimension $n \times p$
I_n	Identity matrix of dimension n
$\ \cdot\ _1$	ℓ_1 -norm, $\ a\ _1 = \sum_{i=1}^n a_i $

Bibliography

- Anderson, M. and Braak, C. T. (2003). "Permutation tests for multi-factorial analysis of variance". In: *Journal of Statistical Computation and Simulation*. 73.2, pp. 85–113.
- Benaiche, N. (2021). "Stabilisation of the R package LMWiRe - Linear models for wide responses". MA thesis. Louvain-la-Neuve, Belgium: UCLouvain.
- Benjamini, Y. and Hochberg, Y. (1995). "Controlling the false discovery rate: A practical and powerful approach to multiple testing". In: *Journal of the Royal Statistical Society: Series B (Methodological)*. 57.1, pp. 289–300.
- Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., and Plemmons, R. J. (2007). "Algorithms and applications for approximate nonnegative matrix factorization". In: *Computational Statistics & Data Analysis*. 52.1, pp. 155–173.
- Bertinetto, C., Engel, J., and Jansen, J. (2020). "ANOVA simultaneous component analysis: A tutorial review". In: *Analytica Chimica Acta: X*. 6, pp. 100061.
- Boccard, J. and Rudaz, S. (2016). "Exploring omics data from designed experiments using analysis of variance multiblock orthogonal partial least squares". In: *Analytica Chimica Acta*. 920.1, pp. 18–28.
- Boccard, J. and Rudaz, S. (2023). "A unified strategy to rebalance multifactorial designs with unequal group sizes: Application to analysis of variance multiblock orthogonal partial least squares". In: *Analytica Chimica Acta*. 1263.1, pp. 341284.
- Boutsidis, C. and Gallopoulos, E. (2008). "SVD based initialization: A head start for nonnegative matrix factorization". In: *Pattern Recognition*. 41.4, pp. 1350–1362.

- Brereton, R. G. (2003). "The evolution of chemometrics". In: *Analytica Chimica Acta*. 500.1–2, pp. 365–377.
- Cañellas, N., Solà-Alberich, R., Brezmes, J., Mallol, R., Valls, R.-M., Rodríguez, M. A., Vinaixa, M., Anguera, A., and Correig, X. (2013). "Use of multivariate chemometric algorithms on ¹H NMR data to assess a soluble fiber (*Plantago ovata* husk) nutritional intervention". In: *Chemometrics and Intelligent Laboratory Systems*. 121, pp. 1–8.
- Chao, G., Mao, C., Wang, F., Zhao, Y., and Luo, Y. (2018). "Supervised nonnegative matrix factorization to predict ICU mortality risk". *IEEE International Conference on Bioinformatics and Biomedicine*, pp. 1189–1194.
- Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S.-i. (2009). "Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation". John Wiley & Sons.
- Climaco-Pinto, R., Barros, A. S., Locquet, N., Schmidtke, L., and Rutledge, D. N. (2009). "Improving the detection of significant factors using ANOVA-PCA by selective reduction of residual variability". In: *Analytica Chimica Acta*. 653.2, pp. 131–142.
- Climaco Pinto, R., Bosc, V., Noçairi, H., Barros, A. S., and Rutledge, D. N. (2008). "Using ANOVA-PCA for discriminant analysis: Application to the study of mid-infrared spectra of carraghenan gels as a function of concentration and temperature". In: *Analytica Chimica Acta*. 629.1–2, pp. 47–55.
- De Beer, T. R. M., Bodson, C., Dejaegher, B., Walczak, B., Vercruysse, P., Burggraeve, A., Lemos, A., Delattre, L., Vander Heyden, Y., Remon, J. P., Vervaet, C., and Baeyens, W. R. G. (2008). "Raman spectroscopy as a process analytical technology (PAT) tool for the in-line monitoring and understanding of a powder blending process". In: *Journal of Pharmaceutical and Biomedical Analysis*. 48.3, pp. 772–779.
- De Beer, T., Burggraeve, A., Fonteyne, M., Saerens, L., Remon, J. P., and Vervaet, C. (2011). "Near infrared and Raman spectroscopy for the in-process monitoring of pharmaceutical production processes". In: *International Journal of Pharmaceutics*. 417.1–2, pp. 32–47.
- Dorscheidt, T. (2019). *MetStaT: Statistical metabolomics tools*. R package version 1.0.
- Dunn, O. J. (1961). "Multiple comparisons among means". In: *Journal of the American Statistical Association*. 56.293, pp. 52–64.
- El Ghaziri, A., Qannari, E. M., Moyon, T., and Alexandre-Gouabau, M.-C. (2015). "AoV-PLS: A new method for the analysis of multivariate data depending on several factors". In: *Electronic Journal of Applied Statistical Analysis*. 8.2, pp. 214–235.

- Erdős, B., Westerhuis, J. A., Adriaens, M. E., O'Donovan, S. D., Xie, R., Singh-Povel, C. M., Smilde, A. K., and Arts, I. C. W. (2023). "Analysis of high-dimensional metabolomics data with complex temporal dynamics using RM-ASCA+". In: *PLoS Computational Biology*. 19.6, pp. e1011221.
- Erichson, N. B., Voronin, S., Brunton, S. L., and Kutz, J. N. (2016). *Randomized matrix decompositions using R*. arXiv:1608.02148.
- Figueiredo, M. de, Giannoukos, S., Rudaz, S., Zenobi, R., and Boccard, J. (2022). "Efficiently handling high-dimensional data from multifactorial designs with unequal group sizes using rebalanced ASCA (RASCA)". In: *Journal of Chemometrics*. 37.7, pp. e3401.
- Fisher, R. A. (1935). "The design of experiments". Oliver & Boyd Edinburgh.
- Franceschi, P. (2022). *gASCA: glm based ANOVA – Simultaneous component analysis (ASCA)*. R package version 1.0.
- Fresno, C., Balzarini, M., and Fernandez, E. (2014). "lmdme: Linear models on designed multivariate experiments in R". In: *Journal of Statistical Software*. 56.7, pp. 1–16.
- Friendly, M., Monette, G., and Fox, J. (2013). "Elliptical insights: Understanding statistical methods through elliptical geometry". In: *Statistical Science*. 28.1, pp. 1–39.
- Gillis, N. (2014). "The why and how of nonnegative matrix factorization". *Regularization, Optimization, Kernels, and Support Vector Machines*. Chapman & Hall/CRC, pp. 257–291.
- Govaerts, B. and Martin, M. (2018). *Lecture notes - PCA card*.
- Guisset, S., Martin, M., and Govaerts, B. (2019). "Comparison of PARAFASCA, AComDim, and AMOPLS approaches in the multivariate GLM modelling of multi-factorial designs". In: *Chemometrics and Intelligent Laboratory Systems*. 184, pp. 44–63.
- Harnly, J. M., Harrington, P. B., Botros, L. L., Jablonski, J., Chang, C., Bergana, M. M., Wehling, P., Downey, G., Potts, A. R., and Moore, J. C. (2014). "Characterization of near-infrared spectral variance in the authentication of skim and nonfat dry milk powder collection using ANOVA-PCA, pooled-ANOVA, and partial least-squares regression". In: *Journal of Agricultural and Food Chemistry*. 62.32, pp. 8060–8067.
- Harrington, P., Vieira, N. E., Chen, P., Espinoza, J., Nien, J. K., Romero, R., and Yergey, A. L. (2006). "Proteomic analysis of amniotic fluids using analysis of variance-principal component analysis and fuzzy rule-building expert systems applied to matrix-assisted laser desorption/ionization mass spectrometry". In: *Chemometrics and Intelligent Laboratory Systems*. 82.1–2, pp. 283–293.
- Harrington, P., Vieira, N. E., Espinoza, J., Nien, J. K., Romero, R., and Yergey, A. L. (2005). "Analysis of variance–principal component analysis: A soft tool for proteomic discovery". In: *Analytica Chimica Acta*. 544.1–2, pp. 118–127.

- International Conference on Harmonisation (ICH) (2005). *ICH harmonised tripartite guideline Q9 - Quality risk management*.
- International Conference on Harmonisation (ICH) (2008). *ICH harmonised tripartite guideline Q10 - Pharmaceutical quality system*.
- International Conference on Harmonisation (ICH) (2009). *ICH harmonised tripartite guideline Q8(R2) - Pharmaceutical development*.
- Jansen, J. J., Bro, R., Hoefsloot, H. C. J., van den Berg, F. W. J., Westerhuis, J. A., and Smilde, A. K. (2008). "PARAFASCA: ASCA combined with PARAFAC for the analysis of metabolic fingerprinting data". In: *Journal of Chemometrics*. 22.2, pp. 114–121.
- Jansen, J. J., Hoefsloot, H. C. J., van der Greef, J., Timmerman, M. E., Westerhuis, J. A., and Smilde, A. K. (2005). "ASCA: Analysis of multivariate data obtained from an experimental design". In: *Journal of Chemometrics*. 19.9, pp. 469–481.
- Jarmund, A. H., Madssen, T. S., and Giskeødegård, G. F. (2022). "ALASCA: An R package for longitudinal and cross-sectional analysis of multivariate data by ASCA-based methods". In: *Frontiers in Molecular Biosciences*. 9, pp. 962431.
- Jouan-Rimbaud Bouveresse, D., Climaco Pinto, R., Schmidtke, L. M., Locquet, N., and Rutledge, D. N. (2011). "Identification of significant factors by an extension of ANOVA-PCA based on multi-block analysis". In: *Chemometrics and Intelligent Laboratory Systems*. 106.2, pp. 173–182.
- Langville, A. N., Meyer, C. D., Albright, R., Cox, J., and Duling, D. (2014). *Algorithms, initializations, and convergence for the nonnegative matrix factorization*. arXiv:1407.7299.
- Lapointe-Garant, P.-P., Simard, J.-S., and Abatzoglou, N. (2008). "Real-time NIR monitoring of a pharmaceutical blending process through multivariate analysis-derived models". *Proceedings of the WSEAS International Conference on Mathematics and Computers in Science and Engineering*. World Scientific, Engineering Academy, and Society, pp. 34–39.
- Lee, D. and Seung, H. (2001). "Algorithms for non-negative matrix factorization". *Advances in Neural Information Processing Systems*. Vol. 13. MIT Press, pp. 556–562.
- Lee, D. D. and Seung, H. S. (1999). "Learning the parts of objects by non-negative matrix factorization". In: *Nature*. 401.6755, pp. 788–791.
- Lee, H., Yoo, J., and Choi, S. (2010). "Semi-supervised nonnegative matrix factorization". In: *IEEE Signal Processing Letters*. 17.1, pp. 4–7.
- Leuschner, J., Schmidt, M., Fernsel, P., Lachmund, D., Boskamp, T., and Maass, P. (2018). "Supervised non-negative matrix factorization methods for MALDI imaging applications". In: *Bioinformatics*. 35.11, pp. 1940–1947.
- Liland, K. H. (2023). *multiblock: Multiblock data fusion in statistics and machine learning*. R package version 0.2.2.

- Liland, K. H., Smilde, A., Marini, F., and Næs, T. (2018). "Confidence ellipsoids for ASCA models based on multivariate regression theory". In: *Journal of Chemometrics*. 32.5, pp. e2990.
- Lin, C.-J. (2007). "Projected gradient methods for nonnegative matrix factorization". In: *Neural Computation*. 19.10, pp. 2756–2779.
- Lourenço, N. D., Lopes, J. A., Almeida, C. F., Sarraguça, M. C., and Pinheiro, H. M. (2012). "Bioreactor monitoring with spectroscopy and chemometrics: A review". In: *Analytical and Bioanalytical Chemistry*. 404.4, pp. 1211–1237.
- Madsen, H. and Thyregod, P. (2010). "Introduction to General and Generalized Linear Models". Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. ISBN: 9781420091557.
- Madsen, R., Lundstedt, T., and Trygg, J. (2010). "Chemometrics in metabolomics—A review in human disease diagnosis". In: *Analytica Chimica Acta*. 659.1–2, pp. 23–33.
- Madssen, T. S., Giskeødegård, G. F., Smilde, A. K., and Westerhuis, J. A. (2021a). "Repeated measures ASCA+ for analysis of longitudinal intervention studies with multivariate outcome data". In: *PLoS Computational Biology*. 17.11, pp. e1009585.
- Madssen, T. S., Giskeødegård, G. F., Smilde, A. K., and Westerhuis, J. A. (2021b). "Repeated measures ASCA+ for analysis of longitudinal intervention studies with multivariate outcome data". In: *PLoS Computational Biology*. 17.11, pp. e1009585.
- Marion, R., Lederer, J., Goevarts, B., and Sachs, R. von (2025). "VC-PCR: A prediction method based on variable selection and clustering". In: *Statistica Neerlandica*. 79.1, pp. e12358.
- Martens, H. and Næs, T. (1992). "Multivariate calibration". Wiley.
- Martin, M. and Govaerts, B. (2020). "LiMM-PCA: Combining ASCA+ and linear mixed models to analyse high-dimensional designed data". In: *Journal of Chemometrics*. 34.6, pp. e3232.
- Mazerolles, G., Boccard, J., Hanafi, M., and Rudaz, S. (2011). "Analysis of experimental design with multivariate response: A contribution using multiblock techniques". In: *Chemometrics and Intelligent Laboratory Systems*. 106.1, pp. 65–72.
- Mullen, K. M. (2015). *ALS: Multivariate curve resolution alternating least squares (MCR-ALS)*. R package version 0.0.7.
- Neter, J., Kutner, M., Nachtsheim, C., and Li, W. (2004). "Applied linear statistical models". 5th ed. McGraw-Hill/Irwin.
- Nguipdop Lowe, C. (2022). "Combinaison d'analyse univariée et multivariée dans l'analyse de données omiques issues d'un plan d'expérience". Master en statistiques, orientation biostatistiques. Mémoire de master. Louvain-la-Neuve, Belgique: UCLouvain, Faculté des Sciences.

- Nueda, M. J., Conesa, A., Westerhuis, J. A., Hoefsloot, H. C. J., Smilde, A. K., Talón, M., and Ferrer, A. (2007). "Discovering gene expression patterns in time course microarray experiments by ANOVA-SCA". In: *Bioinformatics*. 23.14, pp. 1792–1800.
- Paatero, P. and Tapper, U. (1994). "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values". In: *Environmetrics*. 5.2, pp. 111–126.
- Pang, Z., Lu, Y., Zhou, G., Hui, F., Xu, L., Viau, C., Spigelman, A. F., MacDonald, P. E., Wishart, D. S., Li, S., and Xia, J. (2024). "MetaboAnalyst 6.0: Towards a unified platform for metabolomics data processing, analysis and interpretation". In: *Nucleic Acids Research*. 52.W1, pp. W398–W406.
- Rosipal, R. and Krämer, N. (2005). "Overview and recent advances in partial least squares". *Subspace, Latent Structure and Feature Selection: International Statistical and Optimization Perspectives Workshop*. Springer, pp. 34–51.
- Rousseau, R. (2011). "Statistical contribution to the analysis of metabonomic data in H-NMR spectroscopy". PhD thesis. Université Catholique de Louvain.
- Roy, J. van (2021). "Supervised non-negative matrix factorization for multivariate data: Application to chemical kinetics". Master's thesis. Louvain-la-Neuve, Belgium: UCLouvain.
- Ruckebusch, C. (2016). "Resolving spectral mixtures: With applications from ultrafast time-resolved spectroscopy to super-resolution imaging". Elsevier.
- Sarembaud, J., Pinto, R., Rutledge, D. N., and Feinberg, M. (2007). "Application of the ANOVA-PCA method to stability studies of reference materials". In: *Analytica Chimica Acta*. 603.2, pp. 147–154.
- Schaefer, C., Clicq, D., Lecomte, C., Merschaert, A., Norrant, E., and Fotiadu, F. (2014). "A process analytical technology (PAT) approach to control a new API manufacturing process: Development, validation and implementation". In: *Talanta*. 120, pp. 114–125.
- Schmidt, M. N., Winther, O., and Hansen, L. K. (2009). "Bayesian Non-negative Matrix Factorization". *Independent Component Analysis and Signal Separation*. Springer, Berlin, Heidelberg, pp. 540–547.
- Smilde, A. K., Jansen, J. J., Hoefsloot, H. C. J., Lamers, R.-J. A. N., van der Greef, J., and Timmerman, M. E. (2005). "ANOVA-simultaneous component analysis (ASCA): A new tool for analyzing designed metabolomics data". In: *Bioinformatics*. 21.13, pp. 3043–3048.
- Stahle, L. and Wold, S. (1990). "Multivariate analysis of variance (MANOVA)". In: *Chemometrics and Intelligent Laboratory Systems*. 9.2, pp. 127–141.
- Tauler, R. and Maeder, M. (2009). "2.20 - Two-way data analysis: Multivariate curve resolution – Error in curve resolution". *Comprehensive Chemometrics*.

- Ed. by S. D. Brown, R. Tauler, and B. Walczak. Oxford: Elsevier, pp. 345–363.
- Tauler, R., Smilde, A., and Kowalski, B. (1995). "Selectivity, local rank, three-way data analysis and ambiguity in multivariate curve resolution". In: *Journal of Chemometrics*. 9.1, pp. 31–58.
- Thiel, M. (Sept. 2015). "Utilisation des méthodes multivariées ASCA et APCA dans l'analyse des composantes de variance en métabolomique". Promoteur: Bernadette Govaerts. Lecteurs: Johan Segers, Baptiste Féraud. Mémoire de master en statistiques, orientation biostatistique. Louvain-la-Neuve, Belgique: Université Catholique de Louvain.
- Thiel, M., Benaïche, N., Martin, M., Franceschini, S., Van Oirbeek, R., and Govaerts, B. (2023a). "limpca: An R package for the linear modeling of high-dimensional designed data based on ASCA/APCA family of methods". In: *Journal of Chemometrics*. 37.7, pp. e3482.
- Thiel, M., Benaïche, N., Martin, M., Franceschini, S., Van Oirbeek, R., and Govaerts, B. (2023b). *limpca: Linear Modeling of High-Dimensional Designed Data Based on ASCA/APCA*. R package version 1.4.0.
- Thiel, M., Féraud, B., and Govaerts, B. (2017). "ASCA+ and APCA+: Extensions of ASCA and APCA in the analysis of unbalanced multifactorial designs". In: *Journal of Chemometrics*. 31.6, pp. e2895.
- Thiel, M., Sauwen, N., Khamiakova, T., Maes, T., and Govaerts, B. (2021). "Comparison of chemometrics strategies for the spectroscopic monitoring of active pharmaceutical ingredients in chemical reactions". In: *Chemometrics and Intelligent Laboratory Systems*. 211, pp. 104273.
- Thissen, U., Wopereis, S., van den Berg, S. A. A., Bobeldijk, I., Kleemann, R., Kooistra, T., van Dijk, K. W., van Ommen, B., and Smilde, A. K. (2009). "Improving the analysis of designed studies by combining statistical modelling with study design information". In: *BMC Bioinformatics*. 10.1, pp. 1–12.
- Thévenot, E. A., Roux, A., Xu, Y., Ezan, E., and Junot, C. (2015). "Analysis of the human adult urinary metabolome variations with age, body mass index, and gender by implementing a comprehensive workflow for univariate and OPLS statistical analyses". In: *Journal of Proteome Research*. 14.8, pp. 3322–3335.
- Tournay, M. (2020). "Extension of LiMM-PCA methodology to longitudinal omics data". Master in Statistics, supervised by B. Govaerts. Master thesis. Louvain-la-Neuve, Belgium: Université Catholique de Louvain.
- Trygg, J., Holmes, E., and Lundstedt, T. (2007). "Chemometrics in metabolomics". In: *Journal of Proteome Research*. 6.2, pp. 469–479.
- United States Food and Drug Administration (FDA) (2009). *Guidance for industry PAT - A framework for innovative pharmaceutical development, manufacturing, and quality assurance*.

- Van Oirbeek, R., Blommaert, A., Sauwen, N., Maes, T., Dijkmans, J., Cuypers, J., Khamiakova, T., Thiel, M., and Beleites, C. (2018). *spectralAnalysis: Pre-process, visualize and analyse process analytical data, by spectral data measurements made during a chemical process*. R package version 3.15.
- Varmuza, K. and Filzmoser, P. (2016). "Introduction to multivariate statistical analysis in chemometrics". CRC Press.
- Verouden, M. P. H., Westerhuis, J. A., van der Werf, M. J., and Smilde, A. K. (2009). "Exploring the analysis of structured metabolomics data". In: *Chemometrics and Intelligent Laboratory Systems*. 98.1, pp. 88–96.
- Vis, D. J., Westerhuis, J. A., Smilde, A. K., and van der Greef, J. (2007). "Statistical validation of megavariate effects in ASCA". In: *BMC Bioinformatics*. 8.1, pp. 322–332.
- Visschers, A. (2024). "Extension du package limpca à l'analyse des modèles linéaires mixtes". Supervisor: Bernadette Govaerts. Master's thesis. Louvain-la-Neuve, Belgium: Université Catholique de Louvain.
- Wang, Y. and Zhang, Y. (2013). "Nonnegative matrix factorization: A comprehensive review". In: *IEEE Transactions on Knowledge and Data Engineering*. 25.6, pp. 1336–1353.
- Xia, J., Sinelnikov, I. V., Han, B., and Wishart, D. S. (2015). "MetaboAnalyst 3.0—Making metabolomics more meaningful". In: *Nucleic Acids Research*. 43.W1, pp. W251–W257.
- Ziémons, E., Mantanus, J., Lebrun, P., Rozet, E., Evrard, B., and Hubert, P. (2010). "Acetaminophen determination in low-dose pharmaceutical syrup by NIR spectroscopy". In: *Journal of Pharmaceutical and Biomedical Analysis*. 53.3, pp. 510–516.
- Zwanenburg, G., Hoefsloot, H. C. J., Westerhuis, J. A., Jansen, J. J., and Smilde, A. K. (2011). "ANOVA-principal component analysis and ANOVA-simultaneous component analysis: A comparison". In: *Journal of Chemometrics*. 25.10, pp. 561–567.