UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Ph.D. Dissertation

Chemometric Approaches for Systems Biology

Author

Abel Folch Fortuny

Ph.D. supervisors

Alberto J. Ferrer Riquelme
Francisco J. Arteaga Moreno

A doctoral thesis submitted to

Department of Applied Statistics, Operations Research and Quality

Valencia, November 2016

Cover credits:

Design:	Abel Folch Fortuny
Pictures:	Carmelo Folch Santajustina
Reviews:	Alberto Ferrer
	José María González-Martínez
	Elena M. Tur

ISBN 978-90-65624-00-0

Published by:

Delft Academic Press / VSSD uitgeverij
Leeghwaterstraat 42
2628 CA Delft
+31 (0) 15 2782124
dap@vssd.nl
www.delftacademicpress.nl

als meus

Abstract

The present Ph.D. thesis is devoted to study, develop and apply approaches commonly used in chemometrics to the emerging field of systems biology. Existing procedures and new methods are applied to solve research and industrial questions in different multidisciplinary teams. The methodologies developed in this document will enrich the plethora of procedures employed within omic sciences to understand biological organisms and will improve processes in biotechnological industries integrating biological knowledge at different levels and exploiting the software packages derived from the thesis.

This dissertation is structured in four parts. The first block describes the framework in which the contributions presented here are based. The objectives of the two research projects related to this thesis are highlighted and the specific topics addressed in this document via conference presentations and research articles are introduced. A comprehensive description of omic sciences and their relationships within the systems biology paradigm is given in this part, jointly with a review of the most applied multivariate methods in chemometrics, on which the novel approaches proposed here are founded.

The second part addresses many problems of data understanding within metabolomics, fluxomics, proteomics and genomics. Different alternatives are proposed in this block to understand flux data in steady state conditions. Some are based on applications of multivariate methods previously applied in other chemometrics areas. Others are novel approaches based on a bilinear decomposition using elemental metabolic pathways, from which a GNU licensed toolbox is made freely available for the scientific community. As well, a framework for metabolic data understanding is proposed for non-steady state data, using the same bilinear decomposition proposed for steady state data, but modelling the dynamics of the experiments using novel two and three-way data analysis procedures. Also, the relationships between different omic levels are assessed in this part integrating different sources of information of plant viruses in data fusion models. Finally, an example of interaction between organisms, oranges and fungi, is studied via multivariate image analysis techniques, with future application in food industries.

The third block of this thesis is a thoroughly study of different missing data problems related to chemometrics, systems biology and industrial bioprocesses. In the theoretical chapters of this part, new algorithms to obtain multivariate exploratory and regression models in the presence of missing data are proposed, which serve also as preprocessing steps of any other methodology used by practitioners. Regarding applications, this block explores the reconstruction of networks in omic sciences when missing and faulty measurements appear in databases, and how calibration models between near infrared instruments can be transferred, avoiding costs and time-consuming full recalibrations in bioindustries and research laboratories. Finally, another software package, including a graphical user interface, is made freely available for missing data imputation purposes.

The last part discusses the relevance of this dissertation for research and biotechnology, including proposals deserving future research.

Resumen

Esta tesis doctoral se centra en el estudio, desarrollo y aplicación de técnicas quimiométricas en el emergente campo de la biología de sistemas. Procedimientos comúnmente utilizados y métodos nuevos se aplican para resolver preguntas de investigación en distintos equipos multidisciplinares, tanto del ámbito académico como del industrial. Las metodologías desarrolladas en este documento enriquecen la plétora de técnicas utilizadas en las ciencias ómicas para entender el funcionamiento de organismos biológicos y mejoran los procesos en la industria biotecnológica, integrando conocimiento biológico a diferentes niveles y explotando los paquetes de software derivados de esta tesis.

Esta disertación se estructura en cuatro partes. El primer bloque describe el marco en el cual se articulan las contribuciones aquí presentadas. En él se esbozan los objetivos de los dos proyectos de investigación relacionados con esta tesis. Asimismo, se introducen los temas específicos desarollados en este documento mediante presentaciones en conferencias y artículos de investigación. En esta parte figura una descripción exhaustiva de las ciencias ómicas y sus interrelaciones en el paradigma de la biología de sistemas, junto con una revisión de los métodos multivariantes más aplicados en quimiometría, que suponen las pilares sobre los que se asientan los nuevos procedimientos aquí propuestos.

La segunda parte se centra en resolver problemas dentro de metabolómica, fluxómica, proteómica y genómica a partir del análisis de datos. Para ello se proponen varias alternativas para comprender a grandes rasgos los datos de flujos metabólicos en estado estacionario. Algunas de ellas están basadas en la aplicación de métodos multivariantes propuestos con anterioridad, mientras que otras son técnicas nuevas basadas en descomposiciones bilineares utilizando rutas metabólicas elementales. A partir de éstas se ha desarrollado software de libre acceso para la comunidad científica. A su vez, en esta tesis se propone un marco para analizar datos metabólicos en estado no estacionario. Para ello se adapta el enfoque tradicional para sistemas en estado estacionario, modelando las dinámicas de los experimentos empleando análisis de datos de dos y tres vías. En esta parte de la tesis también se establecen relaciones entre los distintos niveles ómicos, integrando diferentes fuentes de información en modelos de fusión de datos. Finalmente, se estudia la interacción entre organismos, como naranjas y hongos, mediante el análisis multivariante de imágenes, con futuras aplicaciones a la industria alimentaria.

El tercer bloque de esta tesis representa un estudio a fondo de diferentes problemas relacionados con datos faltantes en quimiometría, biología de sistemas y en la industria de bioprocesos. En los capítulos más teóricos de esta parte, se proponen nuevos algoritmos para ajustar modelos multivariantes, tanto exploratorios como de regresión, en presencia de datos faltantes. Estos algoritmos sirven además como estrategias de preprocesado de los datos antes del uso de cualquier otro método. Respecto a las aplicaciones, en este bloque se explora la reconstrucción de redes en ciencias ómicas cuando aparecen valores faltantes o atípicos en las bases de datos. Una segunda aplicación de esta parte es la transferencia de modelos de calibración entre instrumentos de infrarrojo cercano, evitando así costosas re-calibraciones en bioindustrias y laboratorios de investigación. Finalmente, se propone un paquete software que incluye una interfaz amigable, disponible de forma gratuita para imputación de datos faltantes.

En la última parte, se discuten los aspectos más relevantes de esta tesis para la investigación y la biotecnología, incluyendo líneas futuras de trabajo.

Resum

Aquesta tesi doctoral es centra en l'estudi, desenvolupament, i aplicació de tècniques quimiomètriques en l'emergent camp de la biologia de sistemes. Procediments comúnment utilizats i mètodes nous s'apliquen per a resoldre preguntes d'investigació en diferents equips multidisciplinars, tant en l'àmbit acadèmic com en l'industrial. Les metodologies desenvolupades en aquest document enriquixen la plétora de tècniques utilitzades en les ciències òmiques per a entendre el funcionament d'organismes biològics i milloren els processos en la indústria biotecnològica, integrant coneixement biològic a distints nivells i explotant els paquets de software derivats d'aquesta tesi.

Aquesta dissertació s'estructura en quatre parts. El primer bloc descriu el marc en el qual s'articulen les contribucions ací presentades. En ell s'esbossen els objectius dels dos projectes d'investigació relacionats amb aquesta tesi. Així mateix, s'introduixen els temes específics desenvolupats en aquest document mitjançant presentacions en conferències i articles d'investigació. En aquesta part figura una descripació exhaustiva de les ciències òmiques i les seues interrelacions en el paradigma de la biologia de sistemes, junt amb una revisió dels mètodes multivariants més aplicats en quimiometria, que suposen els pilars sobre els quals s'assenten els nous procediments ací proposats.

La segona part es centra en resoldre problemes dins de la metabolòmica, fluxòmica, proteòmica i genòmica a partir de l'anàlisi de dades. Per a això es proposen diverses alternatives per a compendre a grans trets les dades de fluxos metabòlics en estat estacionari. Algunes d'elles estàn basades en l'aplicació de mètodes multivariants propostos amb anterioritat, mentre que altres són tècniques noves basades en descomposicions bilineals utilizant rutes metabòliques elementals. A partir d'aquestes s'ha desenvolupat software de lliure accés per a la comunitat científica. Al seu torn, en aquesta tesi es proposa un marc per a analitzar dades metabòliques en estat no estacionari. Per a això s'adapta l'enfocament tradicional per a sistemes en estat estacionari, modelant les dinàmiques dels experiments utilizant anàlisi de dades de dues i tres vies. En aquesta part de la tesi també s'establixen relacions entre els distints nivells òmics, integrant diferents fonts d'informació en models de

fusió de dades. Finalment, s'estudia la interacció entre organismes, com taronges i fongs, mitjançant l'anàlisi multivariant d'imatges, amb futures aplicacions a la indústria alimentària.

El tercer bloc d'aquesta tesi representa un estudi a fons de diferents problemes relacionats amb dades faltants en quimiometria, biologia de sistemes i en la indústria de bioprocessos. En els capítols més teòrics d'aquesta part, es proposen nous algoritmes per a ajustar models multivariants, tant exploratoris com de regressió, en presencia de dades faltants. Aquests algoritmes servixen ademés com a estratègies de preprocessat de dades abans de l'ús de qualsevol altre mètode. Respecte a les aplicacions, en aquest bloc s'explora la reconstrucció de xarxes en ciències òmiques quan apareixen valors faltants o atípics en les bases de dades. Una segona aplicació d'aquesta part es la transferència de models de calibració entre instruments d'infrarroig proper, evitant així costoses re-calibracions en bioindústries i laboratoris d'investigació. Finalment, es proposa un paquet software que inclou una interfície amigable, disponible de forma gratuïta per a imputació de dades faltants.

En l'última part, es discutixen els aspectes més rellevants d'aquesta tesi per a la investigació i la biotecnologia, incloent línies futures de treball.

Acknowledgements
Agradecimientos
Agraïments

I have a great deal of people to acknowledge here. Alberto, has sido más que un supervisor para mí. Con tus enseñanzas, consejos, rigor, experiencia y críticas, me has guiado desde el Trabajo de Fin de Máster a la Tesis de Doctorado. Ahora, como científico, te considero mi padre académico. Francisco, he disfrutado de cada reunión que hemos tenido discutiendo sobre algoritmos, estimadores y código MATLAB. Me has enseñado mucho sobre datos faltantes. De hecho, todas las ideas que se nos han quedado en el tintero darían para otra tesis. Me gustaría también dar gracias al resto de miembros del grupo. José Manuel, m'has recomfortat en més d'un moment de baixó durant la tesi. José María, gracias por ayudarme en mis comienzos. Y Daniel y Eric, os deseo el mayor de los éxitos en la investigación. Especial mención se merece Raffaele, mi padrino, con quien atesoro montones de experiencias durante estos cuatro años. Espero que continúen durante mucho tiempo.

Julio, Rui, Age and Huub, thanks for accepting me as a visiting researcher in your groups. Most of the work compiled here would have been impossible without your collaboration. Also, thanks to my coworkers and colleagues during these days in Vigo, Lisbon and Amsterdam. Muchas gracias también al resto de coautores de los trabajos en los que se basa este libro, en especial a Jesús y a Gabi.

Moltes gràcies també als Dàtils: Héctor, Sayd, Rober, Joan, Fonsi, Diego i Pedro. Heu sigut l'element de desconexió que m'ha servit per a traure endavant aquest treball: Falles, Miravetes, cap d'anys i celebracions varies que recordaré sempre. Many thanks also to my predoc friends in Valencia, y a las largas tardes en el Carmen contándonos nuestras penas y celebrando nuestros éxitos. En especial, Quique, el meu padrí. La nostra amistat començà entre paelles i mantes al coll, però es feu gegant durant el doctorat. Mai farem l'última muixeranga.

Moltíssimes gràcies papà i mamà, la vostra llavor ha sigut incommensurable. L'educació i els valors que m'heu donat des de xicotet han sigut el ciments sobre els que he construit no només aquest document, sinó la meua vida mateixa. Paco i Ana, la vostra experiència i ajuda han sigut de gran importància per a mi durant aquests anys. Norat i Alicia, gràcies per acompanyar-me en aquest trajecte, sempre un pas davant de mi. I finalment, Elena, el teu treball és possiblement el més sutil i a la vegada el més important en aquesta tesi. M'has donat l'estabilitat, la concentració i la força necessàries per a conseguir aquest grand finale. La teua passió ha sigut font d'inspiració durant aquests (més de) quatre anys. Aquest llibre és prova de que res és impossible al teu costat. Te uic!

Contents

Part I

Prologue

Chapter 1

Justification, Objectives and Contributions

1.1 MultiScaleS and SynBioFactory projects

The present thesis has been developed with funding from a research personnel formation (FPI)) grant from the Spanish Ministry of Economy from 2012 to 2016. This grant was related to a project entitled *Multi-scale inference, monitoring, optimization and control: from engineered cells to bioreactors (MultiScaleS)* (reference DPI2011-28112-C04-02), which was carried out between 2012 and 2014. MultiScaleS is now continued through another, still in progress, project called *Synthetic biology for bioproduction enhancement: design, optimization, monitoring and control (SynBioFactory)* (reference DPI2014-55276-C5-1R), which will last until 2017. MultiScaleS and SynBioFactory are projects coordinated among different research groups in different sites across Spain: Multivariate Statistical Engineering Group (GIEM) and Group of Control of Complex Systems (GCSC) from the Technical University of Valencia (UPV), BioProcess Engineering Group (BPEG) from Marine Research Institute - Spanish Research Council (IIM-CSIC), and Group of Statistics and Stochastic Processes (GSSP) from the Technical University of Cartagena (UPCT). Biopolis S.L., a tailor-made biotechnology company, acted as an active partner in both research projects.

White (industrial), green (agriculture) and red (health) biotechnology use enhanced and/or engineered microorganisms as cell factories to produce high-added values specialty metabolites (e.g. amino acids, vitamins, and food additives, biofuels, biofilms and tissues). Biotechnological engineering is of paramount importance for the future of health, chemical, food and other process industries. Yet, the current state-of-the-art, characterized by uncertainty and lack of in-depth real-time knowledge about the process state, forces industry to operate their bioprocesses at too conservative, suboptimal and not intensified regimes, so as to avoid undesirable microorganism physiological states. Such practices cause problems such as poor efficiency, lack of process stability and increased waste of product.

In order to surmount these difficulties, there is a need of identifying desirable metabolic (physiological) states (such as ones of high productivity) and of developing bioreactor optimization, monitoring and control methods so as to lead the system to the desired state in the course of a process, while considering the metabolic state and constraints. This implies considering processes in a wide range of temporal scales (from seconds for metabolic fluxes, minutes for the aggregated population and extracellular metabolites dynamics, to hours for genetic regulation) and spatial ones (from the intracellular dynamics to the microorganisms population inside the bioreactor).

With the new paradigm established by systems biology, it is unsufficient to analyse a single biological level (metabolic) to fully understand the behaviour of living organisms. For this, other biological levels have to be studied (genomic, transcriptomic, proteomic) to find relationships among them and describe systematically and accurately how changes ocurred at a single level are transferred as a cascade to subsequent biological layers.

Within the systems and synthetic bioprocesses context, the aim of MultiScaleS was to provide systematic methods, tools and protocols for inference, real-time monitoring, optimization and feedback control of biosystems by means of multiscale strategies, spanning from micro (e.g. metabolic, protein and genetic networks) to macro scales (e.g. population macroscopic dynamics as used in the context of bioreactors monitoring and control). MultiScaleS expected results were instrumental to achieve end products inside specifications and optimal productivity while operating at intensified regimes. These results were also expected to be applied within other industrial contexts characterized by muti-scale dynamics and coordination of dynamical agents.

The project focused on:

1. Investigating, improving and exploiting topics concerning multi-scale model building and analysis methods and tools, including systematic model building and experimental design, grey modelling, scaling-up, inference in biological systems, and multicellular coordinated dynamics analysis.
2. Novel multi-scale optimization and control methods, including new metaheuristics for optimization and optimal control in metabolic engineering, optimal integrated design and control (steps towards synthetic biology), model-based software sensors (observers) accounting for multiple scales, bioreactor

control considering the metabolic state and constraints, and control of cell interactions.

3. Application to biotechnological industrial production, with special emphasis on how low level biological metabolic states can be controlled at the bioreactor level.

The continuation of the previous project, SynBioFactory, is an application driven project targetting several challenging methodological problems in the interface between synthetic biology (SynBio), systems engineering, and bioprocess engineering. The project will apply SynBio for bioproduction enhancement, with an emphasis on the role that engineering design methods can play exploiting optimization, monitoring and feedback control. Its ultimate goal is to help SynBio to become an engineering discipline. SynBioFactory, thus, emphasizes engineering principles and methodology in designing, constructing and characterizing biological systems from traditional genetic engineering research. Within this framework, SynBioFactory addresses two practical problems in the bioprocess industry that have the common final objective of understanding and driving the microorganism to the desired state in order to maximise yield and productivity:

1. Develop efficient production systems for protein synthesis and expression, with emphasis on control of protein expression variability and host-circuit interaction.

2. Rational design and optimization of synthetic pathways for the synthesis of commodities, with emphasis on methods and circuits to drive metabolic fluxes so as to maximise yield and productivity, and to manage metabolic burden.

SynBioFactory does not forget neither the relevance of the methodological aspects, nor the current SynBio need of availability of biological parts (biobricks), biological devices, and software tools to decouple design from implementation. Therefore, transversal to the executive goals above, two methodological ones will be considered:

1. Fostering SynBio to become engineering by making the process of designing more systematic (standarized), modular, predictable, robust, scalable, and efficient.

2. Implementation of software methods and biobricks on an open-source, public-access basis.

To achieve these goals, SynBioFactory uses methods from the areas of mathematical optimization, systems engineering and control, and multivariate statistics. These methods are the essential enabling technologies that, along with metabolic engineering and DNA synthesis and assembly, allow to provide proper solutions to the previous challenges.

1.2 Objectives of this thesis

In this thesis, some of the previous project aims are addressed, specifically the goals concerning the model reduction and analysis of biological systems, and the application of these methods to biotechnological production, providing tools for data understanding. The objectives of this thesis are: i) build models integrating information from different biological levels, ii) develop missing data (MD) methods and outlier detection and correction procedures in systems biology and bioprocesses, and iii) address near infrared (NIR) spectroscopy and image analysis problems in bioprocesses.

Objective 1: Build models integrating information from different biological levels

To understand the behaviour of microorganisms, and relate the biological information at different levels, multivariate models are fitted to different kinds of biodata. In particular, this thesis focuses on:

- Developing hybrid-modelling methodologies to combine first principles and data-driven models.
- Fitting latent variable and soft modelling methods to build small metabolic models of steady state flux data sets.
- Studying elemental pathways in metabolic networks and developing new methods for relating their activation patterns with the behaviour of the organism.
- Analysing non-steady state metabolic data and proposing a new framework to establish differences between experimental conditions.
- Fusing biological information at different levels to identify functional modules in networks.

Objective 2: Develop missing data methods and outlier detection and correction procedures in systems biology and bioprocesses

One of the big challenges in data analysis in any research field is how to deal with missing and faulty data. Especially in systems biology this problem is of paramount importance, since experiments at small scale and large scale bioprocesses are controlled using instruments, which may eventually fail during data acquisition. In this area, the following issues are addressed:

- Building exploratory and predictive models using incomplete data sets.
- Inferring biological networks with missing values and outliers.
- Transferring missing data imputation methods to the scientific community via user-friendly software.

Objective 3: Address NIR and image analysis problems in bioprocesses.

The ultimate goal of MultiScaleS and SynBioFactory consist of developing methodologies to analyse data from bioprocesses. For this, some problems commonly faced in (bio)industries at this level are studied in this thesis. The interest is focused on:

- Solving the problem of calibration transfer between near infrared instruments.
- Applying multivariate models to hyperspectral images for discrimination purposes.

Regarding the Objective 1, two contributions are presented in Chapter 5, adressing the first two bullet points. Instead of working only with first principles information or only with experimental measurements, a grey modelling approach is proposed in this work, combining a given metabolic model of *Pichia pastoris* and a set of extracelullar fluxes measured in different cultures. Using this modelling the coherence between both sources of information is assessed and the internal fluxes can be inferred using monte carlo (MC) sampling. Afterwards, different multivariate exploratory approaches are applied. On the one hand, principal component analysis (PCA) enhanced by missing data methods in the context of exploratory data analysis (MEDA) are used in order to find the orthogonal pathways representing the main traces of flux data. On the other hand, a soft modelling method called multivariate curve resolution (MCR) is applied on the same data set, including biological constraints to improve the interpretability of the pathways. Both works are a collaboration with the GCSC and the bio-based company Biopolis.

In Chapters 6 and 7, steady and non-steady state fluxes, respectively, are analysed using a new framework modelling based on elementary flux mode bilinear decomposition. This methodology consists of projecting the flux data in the set of elementary modes (EMs) of the metabolic network, which are the simplest pathways in networks from a stoichiometric point of view. Chapter 6 presents the principal elementary mode analysis (PEMA) method and toolbox, a PCA-like method to find the common set of active EMs across different experiments. In Chapter 7, dynamic elementary mode analysis (dynEMA) and dynamic elementary mode regression discriminant analysis (dynEMR-DA) are described, which are exploratory and regression techniques, respectively, to non-steady state flux analysis integrating time dynamics using N-way modelling. The first work is a collaboration between GIEM group and the Systems Biology Engineering group at the New University of Lisbon (UNL), and it has been developed during a research stay in 2014, under the supervision of Prof. Rui Oliveira. The second contribution is a collaboration among different universities: UPV, University of Amsterdam (UvA), Free University of Amsterdam and University of Groningen. This work has been done during a 4-month research stay at UvA, within the BioData Analysis group, and under the supervision of Dr. Huub C.J. Hoefsloot and Prof. Age K. Smilde.

A data fusion approach is presented in Chapter 8, combining different sources of biological information: genomic, proteomic and phenotypic. This work seeks to model the effect that mutations performed at the ribonucleic acid (RNA) of *Potyviruses* provoke in the protein-protein interaction network (PPIN) of the organism, and how the two biological layers affect the physiological performance of the virus. The two contributions associated to this chapter represent a joint project among GIEM, GCSC and the Evolutionary Systems Virology group at the Institute of Cellular and Molecular Plant Biology - CSIC (IBMCP-CSIC).

Different works are presented to cover the second objective of the present thesis. The first work addressing the missing data problem is described in Chapter 10, where new methods for PCA model building with missing data are presented and compared to other state-of-the-art techniques. The proposed methods are regression-based approaches adapted from the PCA model exploitation context to model building. Afterwards, these novel methods are integrated in a MATLAB graphical user-friendly interface (GUI) called Missing Data Imputation (MDI) Toolbox in Chapter 12. This way, these novel competitive approaches are combined with a graphical interface in MATLAB platform, allowing researchers to impute their missing values in data sets in a straightforward way. Also within exploratory techniques, these methods are adapted to a maximum-likelihood (ML) environment in Chapter 13, where they are integrated in maximum likelihood PCA (MLPCA) algorithm. The accuracy in the reconstruction of metabolic flux data sets with missing measurements is tested, among other data sets, in this chapter. Also, NIR and Fourier transformed infrared (FTIR) microspectroscopic data sets are analysed in Chapters 10 and 13, respectively. Finally, the imputation methods presented in Chapter 10 are adapted to a predictive environment in Chapter 15, when fitting partial least squares (PLS) regression algorithms. Contributions in Chapters 10, 12, 13 and 15 are a collaboration between GIEM and the Catholic University of Valencia (UCV).

The effect of both missing measurements and outliers is addressed in Chapter 11. In this work, a new methodology is presented to preprocess biological data as a prior step of network inference, particularly for metabolic and gene regulatory networks. This way, the effect of the imputation and the outlier correction is assessed jointly with a state of the art network inference method based on mutual information distance and entropy reduction (MIDER). MIDER uses information theory concepts such as entropy associated to a variable or common information between two variables. This contribution was achieved during a 2-week research stay at the IIM-CSIC in Vigo (Spain), and respresents a collaboration between GIEM and BPEG, both groups members of MultiScaleS and SynBioFactory.

The third objective has been attained via two collaborative projects between parties outside MultiScaleS and SynbioFactory projects. The first one, addressing the calibration transfer problem, is a collaboration with Shell Global Solutions B.V. One of the most common problems in experimental sciences, and also (bio)industries, consists of transferring calibration models, developed on one instrument, to another one. To provide an efficient solution to this problem, avoiding time-consuming complete recalibrations, different methods are proposed in Chapter 14. Among these approaches, two are based on the results of Chapters 10 and 13.

The second chapter related to Objective 3 is a collaboration between GIEM and the Valencian Institute for Agricultural Research (IVIA). This food industry project is aimed at developing multivariate discriminant models to detect early stages of rottenness in visible/NIR (VIS/NIR) hyperspectral images taken from oranges. For this, different orange and tangerine varieties are used in an experiment *versus* control study to elucidate which wavelengths of an hyperspectral camera are the most relevant ones for discrimination between groups. The final output of this work consists of developing on-line camera systems in fruit packinghouses to remove the slightly decayed fruit from the chain before storing them. The results of this study are presented in Chapter 9.

Based on the previous objectives, the thesis is structured as follows. Within the present part *Prologue*, the content of the thesis is outlined in Chapter 1, an introduction to chemometrics and systems biology are presented in Chapters 2 and 3, respectively, and some comments on the materials used in the thesis are made in Chapter 4. In *Part II: Modelling biological organisms*, the chapters covering the first objective are presented, including also the multivariate modelling of hyperspectral images in food industry (Chapter 7). In *Part III: Missing data*, the contributions addressing Objective 2 are presented in chapter form, including the NIR calibration transfer problem presented in Chapter 14. Finally, the conclusions, relevance and future work of this thesis are included in the last part *Epilogue*.

1.3 Contributions

1.3.1 Articles in peer-reviewed journals

[1] González-Martínez, J.M., Folch-Fortuny, A., Llaneras, F., Tortajada, M., Picó, J. & Ferrer. A. Metabolic flux understanding of *Pichia pastoris* grown on heterogeneous culture media. *Chemometrics and Intelligent Laboratory Systems* **134**, 89-99 (2014).

[2] Bosque, G., Folch-Fortuny, A., Picó, J., Ferrer, A. & Elena, S.F. Topology analysis and visualization of Potyvirus protein-protein interaction network, *BMC Systems Biology* **8**:129 (2014). Highly accessed article.

[3] Folch-Fortuny, A., Tortajada, M., Prats-Montalbán, J.M., Llaneras, F., Picó, J. & Ferrer, A. MCR-ALS on metabolic networks: Obtaining more meaningful pathways. *Chemometrics and Intelligent Laboratory Systems* **142**, 293-303 (2015).

[4] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. PCA model building with missing data: New proposals and a comparative study. *Chemometrics and Intelligent Laboratory Systems* **146**, 77-88 (2015). #8 in TOP25 from July-September 2015.

[5] Folch-Fortuny, A., Villaverde, A.F., Banga, J.R. & Ferrer, A. Enabling network inference methods to handle missing data and outliers. *BMC Bioinformatics* **16**:283 (2015).

[6] Folch-Fortuny, A., Bosque, G., Picó, J., Ferrer, A. & Elena, S.F. Fusion of genomic, proteomic and phenotypic data: the case of potyviruses, *Molecular BioSystems* **12**, 253-261 (2016).

[7] Folch-Fortuny, A., Marques, R., Isidro, I., Oliveira, R. & Ferrer, A. Principal elementary mode analysis (PEMA). *Molecular BioSystems* **12**, 737-746 (2016). 2016 Hot Article.

[8] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Missing Data Imputation Toolbox for MATLAB, *Chemometrics and Intelligent Laboratory Systems* **154**, 93-100 (2016).

[9] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Assessment of maximum likelihood PCA missing data imputation. *Journal of Chemometrics* **30**, 386-393 (2016).

[10] Folch-Fortuny, A., Prats-Montalbán, J.M., Cubero, S., Blasco, J. & Ferrer, A. VIS/NIR hyperspectral imaging and N-way PLS-DA models for detection of decay lesions in citrus fruits. *Chemometrics and Intelligent Laboratory Systems* **156**, 241-248 (2016).

[11] Folch-Fortuny, A., Vitale, R., de Noord, O.E. & Ferrer, A. Calibration transfer between NIR spectrometers: new proposals and a comparative study. *Journal of Chemometrics*, accepted.

[12] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. PLS model building with missing data: New algorithms and a comparative study. *Journal of Chemometrics*, submitted.

[13] Folch-Fortuny, A., Teusink, B., Kiers, H.A.L., Hoefsloot, H.C.J., Smilde, A.K. & Ferrer, A. Dynamic elementary mode analysis of non-steady state flux data. In preparation.

1.3.2 Conference contributions

Folch-Fortuny, A., Tortajada, M., Prats-Montalbán, J.M., Llaneras, F., Picó,, J. & Ferrer, A. MCR on metabolic networks: obtention of meaningful pathways. *V Chemometrics Workshop for Young Researchers*, Badajoz, Spain, 2013.

Bosque, G., Folch-Fortuny, A., Picó, J., Ferrer, A. & Elena, S.F. Topological analysis and visualization of Potyvirus protein-protein interaction network, *Advanced Lecture Course on Systems Biology*, Innsbruck, Austria, 2014.

Folch-Fortuny, A., Tortajada, M., Prats-Montalbán, J.M., Llaneras, F., Picó, J. & Ferrer, A. MCR-ALS: a useful tool for bioprocess understanding. *European Conference on Process Analytics and Control Technologies (EuroPACT2014)*, Barcelona, Spain, 2014.

Folch-Fortuny, A. & Ferrer, A., Multivariate Statistical Models in Systems Biology. *1st Meeting of PhD Students of the Technical University of Valencia*, Valencia, Spain, 2014.

Folch-Fortuny, A., Arteaga, A. & Ferrer, A. PCA model building with missing data: New approach and a comparative study. *International Chemometrics Research Meeting (ICRM2014)*, Nijmegen, The Netherlands, 2014.

Folch-Fortuny, A., Arteaga, A. & Ferrer, A. On the equivalence between projection to the model plane and maximum likelihood PCA for model building with missing data. *International Chemometrics Research Meeting (ICRM2014)*, Nijmegen, The Netherlands, 2014.

Folch-Fortuny, A., Villaverde, A.F., Banga, J.R. & Ferrer, A. Enabling Network Inference Methods to Handle Missing Data, *Conference on Computational Methods in Systems Biology (CMSB2014)*, Manchester, United Kingdom, 2014.

Folch-Fortuny, A., Bosque, G., Picó, J., Ferrer, A. & Elena, S.F. Latent Structures-Based Modelling of Mutated Protein-Protein Interaction Networks. *Conference on Computational Methods in Systems Biology (CMSB2014)*, Manchester, UK, 2014.

Folch-Fortuny, A., Bosque, G., Picó, J., Ferrer, A. & Elena, S.F. Genomic, proteomic and phenotypic data fusion in potyviruses. *Physical Virology: From Structure to Evolution (BioFiViNet 3)*, Bilbao, Spain, 2015.

Folch-Fortuny, A., Arteaga, A. & Ferrer, A. PCA model building with missing data: New proposals and a comparative study. *Scandinavian Symposium on Chemometrics (SSC14)*, Cagliari, Italy, 2015.

Folch-Fortuny, A., Vitale, R., de Noord, O.E. & Ferrer, A. Fast and efficient calibration transfer between near infrared instruments imputing unmeasured spectra. *Scandinavian Symposium on Chemometrics (SSC14)*, Cagliari, Italy, 2015.

Hervás, D., Folch-Fortuny, A., Lahoz, A., Ferrer, A. & Prats-Montalbán J.M. Variable selection in N-PLS. *Three-way Methods in Chemistry and Psychology (TRICAP2015)*, Pecol - Val di Zoldo, Italy, 2015.

Folch-Fortuny, A. & Ferrer, A. Chemometric approaches for systems biology, *Statistical Methods for Omics Data Integration and Analysis (SMODIA2015)*, Valencia, Spain, 2015.

Folch-Fortuny, A., Kiers, H.A.L., Hoefsloot, H.C.J., Smilde, A.K. & Ferrer, A. Dynamic elementary mode modeling of non-steady state flux data. *Chemometrics Workshop for Young Researchers*, Valencia, Spain, 2015.

Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Trimmed scores regression (TSR). *29th Annual Chemometrics Symposium*, Wageningen, The Netherlands, 2015.

Folch-Fortuny, A., Teusink, B., Kiers, H.A.L., Hoefsloot, H.C.J., Smilde, A.K. & Ferrer, A. Modelling non-steady state metabolic fluxes using dynamic elementary modes. *XIII Symposium on Bioinformatics (JBI2016)*, Valencia, Spain, 2016.

Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Missing Data Imputation (MDI) Toolbox for MATLAB. *Chemometrics in Analytical Chemistry (CAC2016)*, Barcelona, Spain, 2016.

Folch-Fortuny, A., Kiers, H.A.L., Hoefsloot, H.C.J., Smilde, A.K. & Ferrer, A. Dynamic elementary mode modelling of non-steady state flux data. *Chemometrics in Analytical Chemistry (CAC2016)*, Barcelona, Spain, 2016.

1.3.3 Software

PEMA toolbox. Built in MATLAB. In colaboration with UNL. Available in `http://mseg.webs.upv.es`.

MDI toolbox. Built in MATLAB. In colaboration with UCV. Available in `http://mseg.webs.upv.es`.

MD and outlier detection and correction modules in MIDER software for network inference. Built in MATLAB/Octave. In colaboration with IIM-CSIC. Available in `http://gingproc.iim.csic.es/mider.html`.

1.3.4 Awards

Best oral presentation (audience award) at the 1st Meeting of PhD Students of the UPV.

Accessit in 12th University Contest "Arquimedes" for the Introduction to Scientific Research.

Special Prize of CSIC in 12th University Contest "Arquimedes" for the Introduction to Scientific Research.

Chapter 2

On chemometrics

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

[1] González-Martínez, J.M., Folch-Fortuny, A., Llaneras, F., Tortajada, M., Picó, J. & Ferrer. A. Metabolic flux understanding of Pichia pastoris grown on heterogeneous culture media. *Chemometrics and Intelligent Laboratory Systems* **134**, 89-99 (2014).

[3] Folch-Fortuny, A., Tortajada, M., Prats-Montalbán, J.M., Llaneras, F., Picó, J. & Ferrer, A. MCR-ALS on metabolic networks: Obtaining more meaningful pathways. *Chemometrics and Intelligent Laboratory Systems* **142**, 293-303 (2015).

[4] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. PCA model building with missing data: New proposals and a comparative study. *Chemometrics and Intelligent Laboratory Systems* **146**, 77-88 (2015). #8 in TOP25 from July-September 2015.

[6] Folch-Fortuny, A., Bosque, G., Picó, J., Ferrer, A. & Elena, S.F. Fusion of genomic, proteomic and phenotypic data: the case of potyviruses, Molecular Biosystems **12**, 253-261 (2016).

[9] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Assessment of maximum likelihood PCA missing data imputation. *Journal of Chemometrics* **30**, 386-393 (2016).

[10] Folch-Fortuny, A., Prats-Montalbán, J.M., Cubero, S., Blasco, J. & Ferrer, A. VIS/NIR hyperspectral imaging and N-way PLS-DA models for detection of decay lesions in citrus fruits. *Chemometrics and Intelligent Laboratory Systems* **156**, 241-248 (2016).

[11] Folch-Fortuny, A., Vitale, R., de Noord, O.E. & Ferrer, A. Calibration transfer between NIR spectrometers: new proposals and a comparative study. *Journal of Chemometrics*, accepted.

[12] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. PLS model building with missing data: New algorithms and a comparative study. *Journal of Chemometrics*, submitted.

2.1 Introduction

Univariate and bivariate statistics have been applied to solve problems since early 20th century. Student's t tests, bivariate correlations, analysis of variance (ANOVA) and linear regression have been used in many research areas by the first applied statisticians, like Fisher, Tukey, Youden, Gosset or Box [14, 15]. As the technology evolved in many research areas, these techniques became insufficient to exploit the data-rich environments, especially in engineering, chemistry and biology. The relative low cost of measuring devices allows registering a wide range of variables at high sampling rates during experiments and (bio)processes. Thus, in the 70s and 80s new methods were developed to deal with multivariate different-source data sets coming from chemistry and process industry, which are the basis of what it is known today as chemometrics.

Chemometric approaches can be strongly divided in two groups: exploratory and regression models. The first group seeks to understand high dimensional data sets, via compression and selection of the most relevant features in data. The second group aims at relating different sources of information, being the most common situation a set of explanatory or predictor variables and a set of dependent variables or responses. These methods are used mainly for classification among classes, discrimination and prediction.

Several methodologies have been proposed in chemometrics to give support and improve the performance of the aforementioned models. Among the main concerns in chemometrics [16], arise how to: preprocess data, deal with missing data, detect outliers, design experiments, validate models, transfer multivariate models, optimize processes, fit nonlinear data, and deal with N-way data structures.

In this thesis, some of the previous problems are faced within the context of systems biology. In this way, existing methods and new approaches are proposed, in collaboration with multidisciplinary teams, to deal with different problems in omic sciences and bioprocess industries.

2.2 Notation

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

Column vectors are represented as bold lower-case letters (e.g. $\mathbf{v}$) and row vectors as $\mathbf{v}^{\mathrm{T}}$, representing $^{\mathrm{T}}$ the operator transposed. When referring to the elements within a vector, scalars with subindices will be used between brackets (e.g. $\mathbf{v} =$

$[v_1, \ldots, v_K]$). If a vector is built concatenating two vectors, the previous notation will be also used omitting the commas (e.g. $\mathbf{u}^{\mathrm{T}} = [\mathbf{v}^{\mathrm{T}}\ \mathbf{w}^{\mathrm{T}}]$).

Matrices will appear in this work as bold capital letters (e.g. $\mathbf{X}$). Observations or individuals within matrices are usually represented by rows, while variables are represented as columns. When a matrix is built concatenating submatrices, the same notation as in vectors will be used (e.g. $\mathbf{Z} = [\mathbf{X}\ \mathbf{Y}]$). The same notation commented on vectors is applicable to rows and columns of matrices (e.g. the rows of matrix $\mathbf{X}$ will be represented as $\mathbf{x}_n^{\mathrm{T}}$, and columns as $\mathbf{x}_k$). When possible, the same letters will be used for the dimension of a mode and the index of one of its elements (e.g. kth column of a matrix with K variables).

N-dimensional arrays will be denoted as bold capital letters underlined as many times as each additional dimension above two (e.g. $\underline{\mathbf{X}}$ is a three-way data structure).

Either latin and greek characters will be used to represent scalars, vectors and matrices. When a vector or matrix has the same value in all entries, bold numbers will be used (e.g. $\mathbf{1}_N^{\mathrm{T}}$ is a row vector with N ones).

The mathematical operator × is used in this thesis to denote the size of the modes of a matrix (e.g. $\mathbf{Y}$ is a $N \times M$ array). No mathematical operator is used for products between scalars, vectors and matrices. Operator $\circ$ will denote the Hadamard element-wise product between vectors or matrices:

$$\mathbf{v} \circ \mathbf{w} = [v_1, v_2, v_3] \circ [w_1, w_2, w_3] = [v_1 w_1, v_2 w_2, v_3 w_3] \tag{2.1}$$

and finally, $\otimes$ will denote the Kronecker tensor product between vectors or matrices, that is:

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \otimes \mathbf{Y} = \begin{bmatrix} x_{11}\mathbf{Y} & x_{12}\mathbf{Y} \\ x_{21}\mathbf{Y} & x_{22}\mathbf{Y} \end{bmatrix} \tag{2.2}$$

Squares and rectangles are used throughout this thesis in figure drawings as a representation of matrices. N-way arrays are represented when possible (e.g. three-way arrays).

Finally, after the Bibliography, the index of abbreviations and acronyms is shown, referencing the first appearance in the document. When two pages are included for a single item, the second one references a more specific description of the corresponding term.

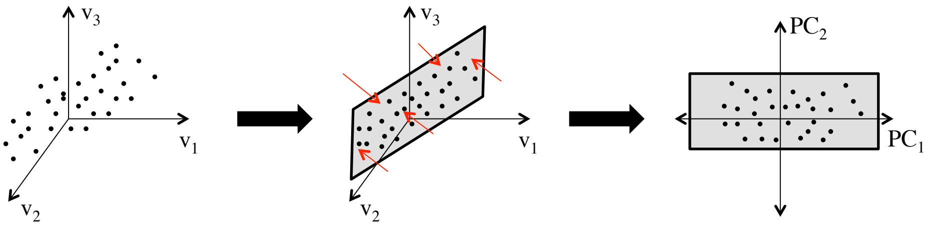


Figure 2.1: Small example of a projection of a bunch of 3-dimensional points to the 2-dimensional loadings plane of PCA, obtaining the score values in the latent space.

2.3 Exploratory data analysis

2.3.1 Principal component analysis (PCA)

The aim of principal component analysis (PCA) [17] is to find the subspace of the variable space where data mostly vary [18]. The original variables, commonly correlated, are linearly transformed into a lower number of uncorrelated variables, the so-called principal components, (PCs). Figure 2.1 shows an example to illustrate this process. The PCA model has the following expression:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \tag{2.3}$$

where $\mathbf{X}$ is a $N \times K$ data matrix, $\mathbf{T}$ is the $N \times A$ score matrix containing the projection of the objects in the A PCs subspace, $\mathbf{P}$ is the $K \times A$ loading matrix containing the linear combination of the variables represented in each of the PCs, and $\mathbf{E}$ is the $N \times K$ residual matrix. The choice of the number of PCs in the model, A, depends on the aim of the study [19].

Outliers in PCA

When a PCA model is fitted, two types of outliers can appear [20]: squared prediction error (SPE) and Hotelling-T^2 outliers. Figure 2.2 illustrates the difference between both types of outliers. SPE measures the squared euclidean (perpendicular) distance from an observation in the K-dimensional original variable space to the A-dimensional latent subspace [21]. It is expressed as:

$$SPE_n = \mathbf{e}_n^{\mathrm{T}}\mathbf{e}_n \tag{2.4}$$

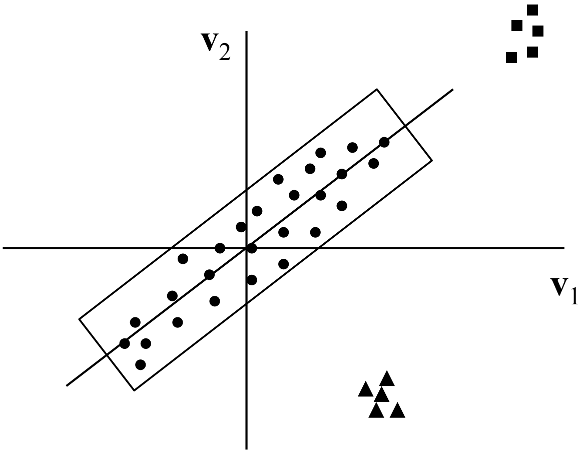


Figure 2.2: Small example of a two-variable data set with a one dimensional latent space (straight black line). The solid squares (triangles) represent Hotelling-T^2 (SPE) outliers.

where $\mathbf{e}_n^{\mathrm{T}}$ is the nth row of the residual matrix $\mathbf{E} = \mathbf{X} - \mathbf{TP}^{\mathrm{T}}$. By taking the eigenvalues of the covariance matrix of the residual matrix $(\lambda_{A+1}, \ldots, \lambda_K)$, the control limit of the SPE [22] is computed as follows:

$$SPE_\alpha = \theta_1 \left[\frac{z_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0} \tag{2.5}$$

where $\theta_k = \sum_{j=A+1}^{K} (\lambda_j)^k$, $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$ and z_α is the $100(1-\alpha)$ percentile of a standard Normal distribution.

The second type of outlier in PCA is detected via the Hotelling-T^2 statistic, which represents the estimated squared Mahalanobis distance [23] from the center of the latent subspace to the projection of an observation onto this subspace [21]. This statistic is used in multivariate monitoring to compute the squared distance between one object and the model's centre according to the covariance structure [24]. When the data is centered (the mean of each column of $\mathbf{X}$ is equal to zero), the distance between an observation $\mathbf{x}_n^{\mathrm{T}}$ and the centre of the original K-dimensional variable space is:

$$\chi_n^2 = \mathbf{x}_n^{\mathrm{T}} \mathbf{\Sigma}^{-1} \mathbf{x}_n \tag{2.6}$$

where $\mathbf{\Sigma}$ is the real covariance matrix of the original K-dimensional variable space, and χ^2_n follows a χ^2 distribution with K degrees of freedom. In practice, the mean and the covariance matrix are estimated by the data matrix $\mathbf{X}$ as $\mathbf{S} = \mathbf{X}^{\mathrm{T}}\mathbf{X}/(N-1)$. So the approximation to the Mahalanobis distance is the Hotelling-T^2:

$$T^2_n = \mathbf{x}_n^{\mathrm{T}}\mathbf{S}^{-1}\mathbf{x}_n \tag{2.7}$$

The control limit for the Hotelling-T^2 [25] is computed as:

$$T^2_\alpha = \frac{(N^2-1)A}{N(N-A)}\mathrm{F}_\alpha(A, N-A) \tag{2.8}$$

where $\mathrm{F}_\alpha(A, N-A)$ is the $100(1-\alpha)$ percentile of a Snedecor's F distribution with $(A, N-A)$ degrees of freedom.

2.3.2 Missing-data methods for exploratory data analysis (MEDA)

MEDA [26] can be seen as a substitute of rotation methods with better properties. First of all, it is more accurate than rotation methods in the detection of relations between pairs of variables. Also, it is robust to the overestimation of the number of PCs and it does not depend on the normalization of the loadings.

Let $\mathbf{X}$ be a $N \times K$ data matrix. Once the PCA has been performed, the MEDA approach consists of the following steps for each variable k:

1. Build matrix $\tilde{\mathbf{X}}_k$, a $N \times K$ matrix full with zeros except in the kth column, where it contains the kth column of $\mathbf{X}$, $\mathbf{x}_k$:

$$\tilde{\mathbf{X}}_k = [\mathbf{0}, \ldots, \mathbf{0}, \mathbf{x}_k, \mathbf{0}, \ldots, \mathbf{0}] \tag{2.9}$$

2. Estimate the scores from $\tilde{\mathbf{X}}_k$ using the missing data method known data regression (KDR), which is statistically superior to other imputation techniques [27]:

$$\hat{\mathbf{T}} = MD(\tilde{\mathbf{X}}_k) \tag{2.10}$$

3. Estimate the reconstruction of the original data with A components and compute the estimation error:

$$\hat{\mathbf{X}} = \hat{\mathbf{T}}\mathbf{P}^{\mathrm{T}} \tag{2.11}$$

$$\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}} \tag{2.12}$$

where $\mathbf{P}$ is the estimated loading matrix from $\mathbf{X}$, $\hat{\mathbf{X}}$ is the estimation matrix and $\mathbf{E}$ the estimation error matrix:

4. Compute an index of goodness of prediction [28] in all columns except the kth one

$$Q^2_{kl} = 1 - \frac{\sum_{n=1}^{N}(E_{nl})^2}{\sum_{n=1}^{N}(X_{nl})^2}, \quad \forall l \neq k \tag{2.13}$$

where X_{nl} is the element located at the nth row and the lth column of $\mathbf{X}$, and E_{nl} is its estimation error. The closer Q^2_{kl} is to 1, the more related are variables k and l.

Once the values of Q^2_{kl} for all possible combinations of k and l are computed, a matrix $\mathbf{Q}^2$ can be constructed so that Q^2_{kl} is located at row k and column l. This matrix is similar in nature to the element-wise squared correlation matrix. Structural relations between variables are detected as high values in $\mathbf{Q}^2$, but the direct/inverse pair-wise relation is not represented on the matrix because of the squared values. To avoid obvious relations, the values of principal diagonal of $\mathbf{Q}^2$ matrices are set to zero. When the number of variables is large, matrix $\mathbf{Q}^2$ can be shown as a grey map to improve interpretability.

2.3.3 Maximum likelihood principal component analysis (MLPCA)

Let $\mathbf{X}$ be an N by K matrix, $\mathbf{x}_n^{\mathrm{T}}$ its nth row and $\mathbf{x}_k$ its kth column. Each row represents a point in the K-dimensional space of the $\mathbf{X}$ observations, and each column a point in the N-dimensional space of the $\mathbf{X}$ variables. Row n can be decomposed in $\mathbf{x}_n^{\mathrm{T}} = \mathbf{x}_n^{0,\mathrm{T}} + \boldsymbol{\varepsilon}_n^{\mathrm{T}}$, where $\mathbf{x}_n^{0,\mathrm{T}}$ are the true values and $\boldsymbol{\varepsilon}^{\mathrm{T}}$ are their measurement errors [29, 30]. As well, column k can be decomposed in its true and error parts: $\mathbf{x}_k = \mathbf{x}_k^0 + \boldsymbol{\eta}$. Both errors are assumed normally distributed in each of the K and N dimensions, respectively.

The maximisation of the likelihood is obtained by minimising the following objective function:

$$S^2 = \sum_{n=1}^{N}(\mathbf{x}_n^{\mathrm{T}} - \hat{\mathbf{x}}_n^{\mathrm{T}})\boldsymbol{\Sigma}_n^{-1}(\mathbf{x}_n - \hat{\mathbf{x}}_n) = \sum_{k=1}^{N}(\mathbf{x}_k^{\mathrm{T}} - \hat{\mathbf{x}}_k^{\mathrm{T}})\boldsymbol{\Psi}_k^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k) \tag{2.14}$$

where $\mathbf{\Sigma}_n$ is the covariance matrix of the errors $\boldsymbol{\varepsilon}_n^{\mathrm{T}}$ of observation $\mathbf{x}_n^{\mathrm{T}}$, and $\boldsymbol{\Psi}_k$ is the covariance matrix of the errors $\boldsymbol{\eta}$ of variable $\mathbf{x}_k$. The estimation of both vectors arise from:

$$\hat{\mathbf{x}}_n = \hat{\mathbf{P}}(\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{\Sigma}_n^{-1}\hat{\mathbf{P}})^{-1}\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{\Sigma}_n^{-1}\mathbf{x}_n \tag{2.15}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{U}}(\hat{\mathbf{U}}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\hat{\mathbf{U}})^{-1}\hat{\mathbf{U}}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{x}_k \tag{2.16}$$

where $\hat{\mathbf{U}}$ $(N \times A)$, $\hat{\mathbf{D}}$ $(A \times A)$ and $\hat{\mathbf{P}}$ $(K \times A)$ represent the singular value decomposition (SVD) of $\hat{\mathbf{X}} = \hat{\mathbf{U}}\hat{\mathbf{D}}\hat{\mathbf{P}}^{\mathrm{T}} = [\hat{\mathbf{x}}_1 \ldots \hat{\mathbf{x}}_K] = [\hat{\mathbf{x}}_1 \ldots \hat{\mathbf{x}}_N]^{\mathrm{T}}$, using A dimensions or components.

MLPCA algorithm is an alternating least squares procedure that starts imputing initial guesses for $\hat{\mathbf{U}}$ and $\hat{\mathbf{P}}$ based on the SVD decomposition of $\mathbf{X}$. At each iteration, the algorithm has two steps. The first one consists of projecting the rows $\mathbf{x}_n^{\mathrm{T}}$ on the columns of $\hat{\mathbf{P}}$, computing the objective function, and recalculating $\hat{\mathbf{U}}$ and $\hat{\mathbf{P}}$ from an SVD using the estimations. The second step consists of projecting the columns $\mathbf{x}_k$ on the columns of $\hat{\mathbf{U}}$, computing also the objective function, and finally recalculating again $\hat{\mathbf{U}}$ and $\hat{\mathbf{P}}$ from an SVD. Convergence is achieved when the difference between the estimations of the observations are below a specified threshold [24, 29].

2.3.4 Multivariate curve resolution (MCR)

MCR [31–33] focuses in performing a bilinear decomposition of a mixture in their pure components:

$$\mathbf{X} = \mathbf{C}\mathbf{S}^{\mathrm{T}} + \mathbf{E} \tag{2.17}$$

where $\mathbf{S}$ is a matrix containing in its columns the spectra of the pure components, $\mathbf{C}$ gathers the concentration profiles of each component, and $\mathbf{E}$ is the residual matrix [34]. One of the most used versions of MCR is its alternating least squares (ALS) implementation, that is MCR-ALS, implemented in the MCR-ALS toolbox for MATLAB [35, 36].

MCR-ALS needs an initial estimate of either $\mathbf{C}$ or $\mathbf{S}$ matrices to trigger the ALS procedure. Despite many initial guesses can be used [34], the most used ones are simple-to-use interactive self-modeling analysis (SIMPLISMA) [37–39] and evolving factor analysis (EFA) [40, 41]. The first one works selecting in a sequential way the variables in the row or in the column direction that have less information in common with the previously selected ones. EFA locates the increase and decay of a component through the variation of rank, so the concentration window of each one can be determined, and then the approximate concentration profiles can be generated [34].

2.4 Regression models

2.4.1 Principal component regression (PCR)

The PCs extracted by a PCA model can be related to a set of dependent variables **Y** arranged by columns. This model is known as principal component regression (PCR). Given a score matrix **T**, fulfilling:

$$\mathbf{T} = \mathbf{X}\mathbf{P} \tag{2.18}$$

where **P** is the loading matrix of a PCA, **Y** can be expressed as:

$$\mathbf{Y} = \mathbf{T}\mathbf{B} + \mathbf{F} \tag{2.19}$$

where **F** is the residual matrix, and **B** is the regression matrix obtained solving 2.19 using a least squares approach:

$$\hat{\mathbf{B}} = (\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{Y} \tag{2.20}$$

As opposed to multivariate linear regression (MLR) the inversion of $\mathbf{T}^{\mathrm{T}}\mathbf{T}$ should give no problem, since scores are orthogonal [42]. Also, the PCs whose corresponding eigenvalues are close to zero can be left out to avoid collinearity problems [43].

2.4.2 Partial least squares regression (PLS)

Partial least squares regression (PLS) is a multivariate projection method commonly applied to model the inner relationships between a set of **X** $(N \times K)$ variables (descriptors, predictors or process variables) and a set of **Y** $(N \times M)$ variables (output, responses or quality variables) reducing significantly the dimensionality of the initial data set [42]. As opposed to PCR, the PLS model finds a set of A latent variables (LVs) that maximise the covariance between **X** and **Y**.

The first step of PLS consists of obtaining the score matrix **T** $(N \times A)$ of **X** as linear combinations of its original variables:

$$\mathbf{T} = \mathbf{X}\mathbf{W}^{*} \tag{2.21}$$

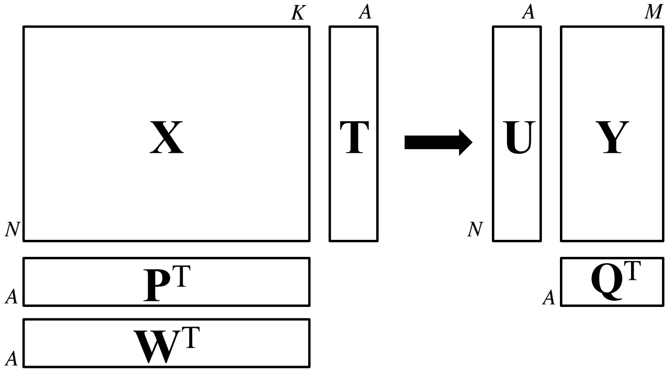


Figure 2.3: Scheme of PLS data matrices.

where $\mathbf{W}^*$ is the normalized weights $K \times A$ matrix.

These new variables are, multiplied by the loading matrix $\mathbf{P}$ $(K \times A)$, good summaries of $\mathbf{X}$, i.e. the residual matrix $\mathbf{E}$ $(N \times K)$, in equation $\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E}$, has entries close to zero. Additionally, the $\mathbf{T}$ variables are built in such a way [44] that they are good predictors of $\mathbf{Y}$.

The $\mathbf{Y}$ variables can be reconstructed as:

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} + \mathbf{G} \tag{2.22}$$

where $\mathbf{U}$ $(N \times A)$ and $\mathbf{Q}$ $(M \times A)$ are the score and loading matrices of $\mathbf{Y}$ in the PLS model, and $\mathbf{G}$ is the $N \times M$ residual matrix. The inner relationship between scores in $\mathbf{X}$ and $\mathbf{Y}$ is $\mathbf{U} = \mathbf{T} + \mathbf{G}$.

Since $\mathbf{T}$ are good predictors of $\mathbf{Y}$, the dependent variables can be expressed as follows:

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^{\mathrm{T}} + \mathbf{F} = \mathbf{X}\mathbf{W}^*\mathbf{Q}^{\mathrm{T}} + \mathbf{F} = \mathbf{X}\mathbf{B}^* + \mathbf{F} \tag{2.23}$$

where $\mathbf{F}$ $(N \times M)$ is the residual matrix and $\mathbf{B}^*$ $(K \times M)$ is the PLS regression coefficient matrix and the normalized weights are obtained as $\mathbf{W}^* = \mathbf{W}(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}$. All matrices can be visualised in Figure 2.3.

Cross-validation and jackknife confidence intervals

Cross-validation (CV) is a resampling technique widely used in statistics and chemometrics [45]. The aim of CV is to assess the number of relevant components to be extracted in the multivariate model. This procedure groups the observations and then fits as many PLS models as groups, leaving each time a single group out. Then, the sum of squares of the differences between the actual **Y** values and the predicted ones is used to estimate the predictive ability of the model [44]. CV is usually performed one component after another, until the predictive power of the model decreases. When one single sample is left out of the model, the leave one out version is being applied. Simultaneously with the CV, the Jackknife confidence intervals (CI) for the PLS regression coefficients can be computed, at a certain confidence level, from all models fitted.

2.4.3 Joint-Y PLS (JYPLS)

JYPLS regression is a non-linear iterative partial least squares (NIPALS) algorithm variant, initially developed for modelling the latent variable structure shared by two or more sets of data (say **X**s) via a PLS-based regression against their corresponding responses (say **Y**s). When only two different couples of data blocks are dealt with, namely $\mathbf{X}_\mathrm{a}$-$\mathbf{Y}_\mathrm{a}$ and $\mathbf{X}_\mathrm{b}$-$\mathbf{Y}_\mathrm{b}$, the mathematical formulation of the JYPLS model is given by:

$$\mathbf{Y} = \begin{bmatrix}\mathbf{Y}_\mathrm{a}\\ \mathbf{Y}_\mathrm{b}\end{bmatrix} = \begin{bmatrix}\mathbf{T}_\mathrm{a}\\ \mathbf{T}_\mathrm{b}\end{bmatrix}\mathbf{Q}^\mathrm{T} + \mathbf{F} \tag{2.24}$$

$$\mathbf{X}_\mathrm{a} = \mathbf{T}_\mathrm{a}\mathbf{P}_\mathrm{a}^\mathrm{T} + \mathbf{E}_\mathrm{a} \tag{2.25}$$

$$\mathbf{X}_\mathrm{b} = \mathbf{T}_\mathrm{b}\mathbf{P}_\mathrm{b}^\mathrm{T} + \mathbf{E}_\mathrm{b} \tag{2.26}$$

$$\mathbf{T}_\mathrm{a} = \mathbf{X}_\mathrm{a}\mathbf{W}_\mathrm{a}^* \tag{2.27}$$

$$\mathbf{T}_\mathrm{b} = \mathbf{X}_\mathrm{b}\mathbf{W}_\mathrm{b}^* \tag{2.28}$$

where $\mathbf{T}_\mathrm{a}/\mathbf{T}_\mathrm{b}$, $\mathbf{P}_\mathrm{a}/\mathbf{P}_\mathrm{b}$ and $\mathbf{W}_\mathrm{a}^*/\mathbf{W}_\mathrm{b}^*$ are the JYPLS scores, loadings and weighting matrices related to $\mathbf{X}_\mathrm{a}/\mathbf{X}_\mathrm{b}$, respectively. The originality of this approach concerns the fact that only one single set of loadings, **Q**, is derived for both $\mathbf{Y}_\mathrm{a}$ and $\mathbf{Y}_\mathrm{b}$, which defines a combined plane mapped by the **Y** joint array (see Equation 2.24).

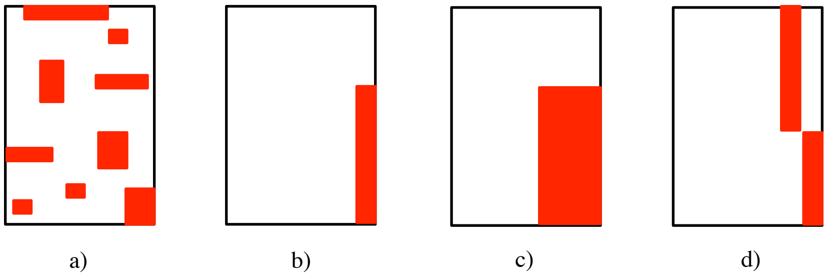


Figure 2.4: Missing data patterns: a) nonstructured, b) univariate, c) block-wise, and d) file matching.

2.5 Missing data

Multivariate data sets are usually arranged in matrices having non-registered cells, i.e. missing values. MD can appear in a wide range of contexts and for a different number of reasons: respondents not answering to some questions in surveys, values outside the instrument range or missing owing to malfunctions of the sensor, failure in the communication between the instrumentation and the digital control system (DCS), sensor with different sampling rates, errors during data acquisition, and so on [46, 47].

There are two critical aspects when dealing with missing values in multivariate data sets: the MD pattern and the MD mechanism [48]. The MD pattern describes the fashion in which the missing values appear in the data matrices. Let $\mathbf{X}$ denote an $N \times K$ data matrix with missing values, being $\mathbf{x}_n^{\mathrm{T}} = [x_{n1}, \ldots, x_{nK}]$ its nth row and $\mathbf{x}_j = [x_{1j}, \ldots, x_{Nj}]^{\mathrm{T}}$ its jth column. The MD indicator matrix $\mathbf{M}$ is a $N \times K$ binary matrix with entries $m_{ij} = 1$ when x_{ij} is missing and $m_{ij} = 0$ otherwise.

The most common MD patterns can be visualised in Figure 2.4. a) denotes nonstructured MD, which is the usual assumption in many MD algorithms. With this pattern, the missing values appear at random in the whole data matrix. In b) the missing values appear just in one variable, which is a common situation in experiments, when not all variables have been measured yet. c) presents the blockwise MD pattern, the extention of b) to multiple variables. This pattern appear in, e.g. chemometric processes and quantitative structure-activity relationship (QSAR) studies when some variables are costly or difficult to measure [47]. Finally, d) denotes the file matching pattern, which usually appear when two or more variables have non-coincident time point measurements. This last case is critical since for some groups of variables there is no common information on the joint distribution of the variables. Consequently, the parameters relating to the association between these variables are not estimable from the data [47, 49].

The second critical aspect is the MD mechanism, which gives information about how the missing values are produced in the data matrix. Let $\mathbf{X}_{\text{obs}}$ and $\mathbf{X}_{\text{mis}}$ denote the observed and missing part of $\mathbf{X}$. Bear in mind that $\mathbf{X}_{\text{obs}}$ and $\mathbf{X}_{\text{mis}}$ are not square matrices but specific entries in the data matrix (0s and 1s, respectively, in the corresponding $\mathbf{M}$ matrix). To state the MD mechanisms it is necessary to find out why values are missing. Different values in the data set may be missing for different reasons, but the important question is whether the variables that are missing are missing because they are related to the underlying values of the variables in the data set.

There are three mechanisms generating MD [48], characterized by the conditional distribution of $\mathbf{M}$ given $\mathbf{X}$, say $\mathrm{P}(\mathbf{M} \mid \mathbf{X}, \boldsymbol{\varphi})$, where $\boldsymbol{\varphi}$ is a vector of unknown parameters:

1. Missing completely at random (MCAR) mechanism arise when there is no relationship between values of the variables (observed and missing) and the probability that they are missing. The missing elements are produced at random in all variables and observations. Therefore, $\mathrm{P}(\mathbf{M} \mid \mathbf{X}, \boldsymbol{\varphi}) = \mathrm{P}(\mathbf{M} \mid \boldsymbol{\varphi})$ for all possible $\mathbf{X}, \boldsymbol{\varphi}$ [47]. In this case the reason why a value is missing does not depend on the true unobserved value. However, this does not imply directly that the MD pattern is unstructured.

2. Missing at random (MAR): in this mechanism, missingness depends only on the observed data $\mathbf{X}_{\text{obs}}$ and not on the values that are missing. $\mathrm{P}(\mathbf{M} \mid \mathbf{X}, \boldsymbol{\varphi}) = \mathrm{P}(\mathbf{M} \mid \mathbf{X}_{\text{obs}}, \boldsymbol{\varphi})$ for all possible $\mathbf{X}_{\text{mis}}, \boldsymbol{\varphi}$ [47]. This hypothesis assumes that in the available measurements there is enough information to estimate the MD. This mechanism can appear, e.g. due to sensor maintenance associated with process operating procedures, or when samples are not measured when certain process measurements are outside the safe limits [47].

3. Not missing at random (NMAR) or nonignorable (NI) mechanism is produced when the probability that an element is missing depends on the unobserved value of the missing elements, so $\mathrm{P}(\mathbf{M} \mid \mathbf{X}, \boldsymbol{\varphi})$ can not be simplified. Classical examples of this mechanism are censored data (values below/above the detection limit) and respondants not answering subtle questions in surveys (e.g. drug abuse in teenagers). NMAR means that we need to model the missing data mechanism to get good estimates of the parameters of interest, and this requires quite specialized methods [47].

Many methods have been proposed in the literature when dealing with missing data in MCAR and MAR mechanisms [47]. These methods can be split in three groups:

1. Single imputation methods: they fulfill MD in a single step, giving a unique estimation for each missing value.
2. Iterative methods: these approaches impute the missing values at different steps within an algorithm, until the values stabilize or some other criterion is achieved (e.g. maximum number of iterations).
3. Multiple imputation methods: they give not just one but several values for each missing value, representing a distribution capable of reflecting the sampling variability [47, 50].

2.5.1 PCA model building and model exploitation with missing data

When fitting PCA, two problems related to missing data appear: (1) exploiting fitted PCA models when some measurements are missing in new observations, i.e. the model exploitation (ME) problem (PCA-ME); and (2) building PCA models from data sets with missing measurements, i.e. the model building (MB) problem (PCA-MB).

In PCA-ME a lot of methods have been reported in the literature. Wise and Ricker [51] present a method that consists of imputing the values that minimise the squared prediction error (SPE) for the new incomplete observation, based on the known PCA model. Nelson et al. [52] study and compare several methods: the single component projection method (SCP), the projection to the model plane method (PMP) and the conditional mean replacement method (CMR). Walczak and Massart [53] study the adaptation of the iterative algorithm (IA) to the prediction of scores for new objects with missing elements. Arteaga and Ferrer [27] also introduce several methods: the trimmed scores method (TRI), the known data regression method (KDR) (which is equivalent to CMR, as proven in [27]) and the trimmed scores regression method (TSR). Additionally, they show that the regression-based methods (KDR and TSR) are statistically more efficient than the other methods studied. Arteaga and Ferrer [54] propose a framework that allows writing the regression-based methods by a unique expression, function of a key matrix.

Regarding PCA-MB there are two methods that are frequently used by the practitioners. The first one consists of adapting the nonlinear iterative partial least squares algorithm (NIPALS) algorithm [55] to deal with incomplete observations by performing the iterative regressions using the present data and ignoring the missing data [30]. The second one is the aforementioned IA [53] that basically consists of filling in the missing data with the predictions obtained from previous PCA models iterated recursively until convergence. Other methods rely on maximum likelihood-based estimations of missing data, like the expectation maximization (E-M) algorithm [56–58]. A more complex method is data augmentation (DA) [56–59]. DA is a multiple imputation method, i.e. for each missing value several values are imputed randomly, and it requires the computation of prior distributions of the parameters. Both E-M and DA are not so widely used as the previous ones (NIPALS or IA). The reason is that the usual chemometrics data sets have strongly correlated variables with a low number of observations. The use of either DA or E-M implies the inversion of the covariance submatrix corresponding to the known variables given an observation, which in these data sets often is not feasible due to submatrices are singular. A recent approach for PCA-MB with MD is the nonlinear programming approach (NLP). In this method the PCA model is obtained solving a nonlinear programming problem, in which the errors between the non-missing values and the model estimations are minimised [60].

Other methods, not so popular, were compared by Liu and Brown in [61]: the algorithm of Krzanowski based on SVD[62], the general iterative principal component imputation (GIP) [63], the multiple imputation by chained equations (MICE) [64], and two regularized versions of the known E-M algorithm: one based on ridge regression (r-EM) [65] and the other one based on a truncated total least squares regression (t-EM) [66].

There are also other imputation methods compared in the literature [67] that are strongly not recommended, like the listwise deletion or complete case analysis (CC) (in which any observation with missing values is removed) and the unconditional mean imputation (MI). The former implies a huge loss of information, leading to loss of precision and bias. The latter distorts the multivariate empirical distribution of the samples, i.e. tends to deform nonlinear quantities (e.g. variances, covariances) [47]. The nearest neighbour method has been also suggested to fulfil the missing values in incomplete datasets [68], however, its applicability with high percentages of missing values is limited.

2.5.2 PLS model building and model exploitation with missing data

When fitting PLS models, as in PCA, the problems with missing data appear both in MB and in ME. In PLS-MB, two methods are the most used among chemometricians: IA [69] and NIPALS [55]. These methods are the extentioned of the aforementioned ones (for PCA-MB) to a PLS enviornment. More details can be found in [47]. In PLS-ME different methods have been proposed in the literature. When no missing data is considered in $\mathbf{Y}$ matrix, the same approaches presented in PCA-ME can be used, that is, the regression-based methods, SCP, PMP, CMR, IA, NIPALS and the minimization of the SPE [27, 51–54]. When considering missing data both in $\mathbf{X}$ and $\mathbf{Y}$, an adaptation of TSR from PCA-ME to PLS-ME has been proposed in [70]. In this algorithm, the weights of the

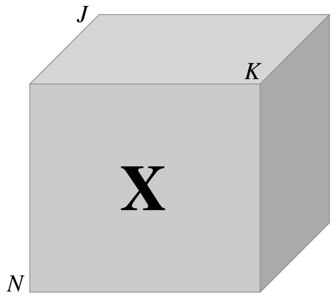


Figure 2.5: Three-way data array.

available data in a PLS model are used to impute the missing values. Also, IA and NIPALS can also impute MD both in **X** and **Y**.

2.6 N-way data

N-way data analysis comprises approaches for analysing N-dimensional arrays, with $N > 2$. The most common situation is having a three-way data structure (see Figure 2.5). In this example, the $\underline{\mathbf{X}}$ data set has K variables, measured on N individuals, along J time points.

Three-way data can be studied decomposing the multidimensional array in two-way data matrices: the $J \times K$ horizontal slice gives the whole data for a particular object, the $N \times J$ vertical slice gives the information for a given variable, and the $N \times K$ frontal slice represent the data at a specific time point [71].

Two-way projection methods, such as PCA and PLS, can be applied directly to these data if the 3-way array is unfolded to build a two-dimensional data matrix. In this way, the data can be unfolded by slices in either of the three possibilities commented in the previous paragraph. However, the most common approaches are i) variable-wise unfolding (VWU), where the third mode is unfolded one slice below another to build a $NJ \times K$ data matrix, and ii) batch-wise unfolding (BWU), where also the third mode is decomposed but one slice after another to build a $N \times KJ$ data matrix.

When fitting N-way projection methods, such as 3-way PCA or PLS, the extention of 2-way to 3-way data analysis implies dealing with three data matrices in the model, that is, the score matrix, retaining information about the observations or individuals, and two loading matrices describing the relationships among variables in the second and third mode.

2.6.1 N-way Partial least squares regression (NPLS)

N-way PLS (NPLS) regression [72] is the natural extension of PLS to N-way structures, which tries to maximize the covariance between the $\underline{\mathbf{X}}$ and $\mathbf{Y}$ data arrays. NPLS discriminant analysis (NPLS-DA) [44, 73] was proposed for studying N-dimensional data structures with discriminant purposes among groups of observations, e.g. experiment versus control studies.

In NPLS-DA, the $\underline{\mathbf{X}}$ $(N \times K \times J)$ data matrix is the datacube represented in Figure 2.5. Considering $\mathbf{X}$ $(N \times JK)$ the BWU version of the datacube $\underline{\mathbf{X}}$, NPLS tries to find latent spaces $\mathbf{W}^J$ and $\mathbf{W}^K$ that maximise the covariance between $\mathbf{X}$ and a dummy vector $\mathbf{y}$, so it can be expressed as:

$$\mathbf{X} = \mathbf{T}(\mathbf{W}^J \otimes \mathbf{W}^K)^{\mathrm{T}} + \mathbf{F} \tag{2.29}$$

afterwards decomposing $\underline{\mathbf{X}}$ from $\mathbf{X}$ using the improved NPLS version expression [74], in order to obtain residuals with better statistical properties:

$$\mathbf{X} = \mathbf{TG}(\mathbf{W}^J \otimes \mathbf{W}^K)^{\mathrm{T}} + \mathbf{F}' \tag{2.30}$$

here, $\otimes$ is the Kronecker product, $\mathbf{W}^J$ and $\mathbf{W}^K$ refer to the weights of the third and second mode, respectively; whereas $\mathbf{T}$ matrix gathers the scores of the observation at each component extracted, and $\mathbf{G}$ is the core array of a Tucker3 decomposition when using $\mathbf{T}$, $\mathbf{W}^J$ and $\mathbf{W}^K$ as loadings, in order to obtain a better (or at least not worse) approximation of the $\underline{\mathbf{X}}$ 3-way array [71, 74]. Finally, $\mathbf{F}$’ incorporates the residuals.

Chapter 3

On systems biology

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

[1] González-Martínez, J.M., Folch-Fortuny, A., Llaneras, F., Tortajada, M., Picó, J. & Ferrer. A. Metabolic flux understanding of Pichia pastoris grown on heterogeneous culture media. *Chemometrics and Intelligent Laboratory Systems* **134**, 89-99 (2014).

[2] Bosque, G., Folch-Fortuny, A., Picó, J., Ferrer, A. & Elena, S.F. Topology analysis and visualization of Potyvirus protein-protein interaction network, *BMC Systems Biology* **8**:129 (2014).

[5] Folch-Fortuny, A., Villaverde, A.F., Banga, J.R. & Ferrer, A. Enabling network inference methods to handle missing data and outliers. *BMC Bioinformatics* **16**:283 (2015).

[6] Folch-Fortuny, A., Bosque, G., Picó, J., Ferrer, A. & Elena, S.F. Fusion of genomic, proteomic and phenotypic data: the case of potyviruses, *Molecular BioSystems* **12**, 253-261 (2016). Highly Accessed Article.

[7] Folch-Fortuny, A., Marques, R., Isidro, I., Oliveira, R. & Ferrer, A. Principal elementary mode analysis (PEMA). *Molecular BioSystems* **12**, 737-746 (2016). 2016 Hot Article.

[13] Folch-Fortuny, A., Teusink, B., Kiers, H.A.L., Hoefsloot, H.C.J., Smilde, A.K. & Ferrer, A. Dynamic elementary mode analysis of non-steady state flux data. In preparation.

3.1 Introduction

3.1.1 Systems biology: a paradigm shift

Systems biology is a multidisciplinary research field that applies established methodologies, and develops new ones, to build models of biological systems integrating information at different levels. This new discipline has become very popular during the last decade, with the explosion of high-throughput technologies. Systems biology allows scientists with different backgrounds (biologists, chemists, computer scientists, mathematicians and statisticians) to work together towards a unified understanding of biological processes [75].

The data integration proposed in systems biology represents a change of paradigm in the study of biological entities (see Figure 3.1). During the 20th century, organisms had been studied using a reductionist approach, that is building theories and developing tools to analyse data within a single biological level. These levels constitute the different omic sciences [76–78]:

- Genomics: aims at identifing the whole genome of an organism, including information about the structure and biological function of genes.
- Transcriptomics: compares gene expression profiles via RNA between biological samples or experimental conditions to identify differences that could help to infer the function of the genes or understand the ongoing biological processes.
- Proteomics: analyses the structure and function of proteins and their associations (i.e. interactions).
- Metabolomics: identifies and quantifies intra and extracellular metabolites, describing the reactions that produce and consume them.
- Fluxomics: deals with the amount of information carried through reactions, the metabolic fluxes, during experiments under certain conditions.

The change of paradigm proposed by systems biology consists not only of studying each omic layer as an island but connecting phenomena across biological levels, thus relating changes in genes, carried to the proteins via transcription and the effect on the metabolic fluxes they enable (see Figure 3.1). A cell, for example, is a combination of tight interconnections among deoxyribonucleic acid (DNA), RNA, proteins and metabolites, and the behaviour of the organism cannot be reduced to the sum of individual pieces [79–82]. The problem of integrating all these information in a single model is that the assumptions made at each level are transferred as a cascade through the subsequent levels, thus the missing, noisy or faulty data may affect the system understanding of the organism [82, 83].

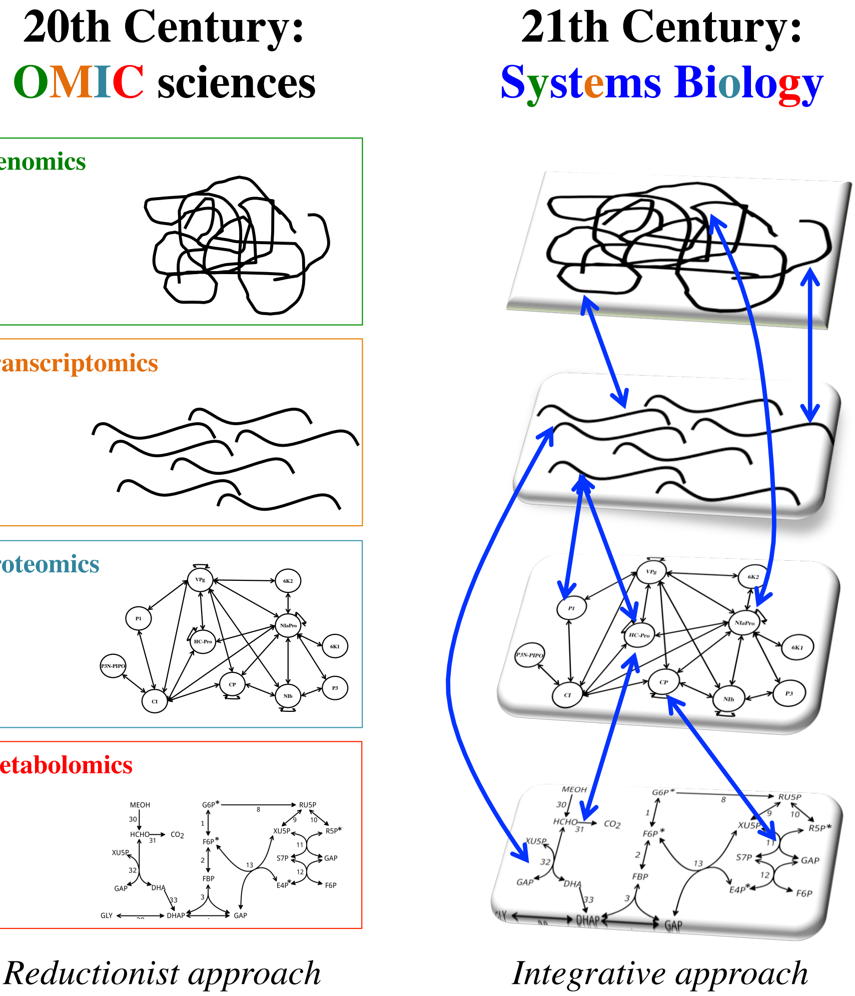


Figure 3.1: Change of paradigm in biology.

3.1.2 Origins

It is considered that systems biology has two historical roots [84], which found a convergence point at the end of the 20th century [85]. The first root is related to the study of genetic material. The DNA coding was the initial breakthrough in this section, followed by the improvement of recombinant technologies during the 60s and 70s. The final discovery, leading to the obtention of large amounts of biological data, was the genome sequencing, which happenned in 1995 for the Haemophilus influenzae [86] (first genome sequenced) and 2001 for the Homo Sapiens [87].

The second root is based on the study of the interactions between multiple molecules. The non-equilibrium thermodynamics theory, set in 1931, was the first step of this branch of knowledge, based on the production of entropy and negative entropy of biochemical processes. Once the concepts of regulation were defined in the 50s, several mathematical models were formulated in order to describe this regulatory circuits, studying the cell as a network. Finally, large genome-scale models, including kinetics, were published in the late 80s and the 90s, treating the genome as a system.

3.1.3 Aim and goals

Research on systems biology is not only driven by disciplines within Biology: metabolomics, proteomics, genomics, but other fields are needed, like mathematical modelling and multivariate statistical analysis. These disciplines provide the essential analytical tools for acquiring, storing, analysing, graphically displaying, integrating, and mathematically modelling biological information [88].

The ultimate goal in systems biology is the understanding of the whole organism by modelling, predicting and controlling the behaviour of all its components [82]. A system-level understanding can be derived from four key properties [80, 89]:

1. System structure identification. In this initial step the knowledge about the object of study needs to be translated into a biological model describing and representing the relations the internal modules of the cell (e.g. genes, metabolites) through metabolic reactions or other biological transformations. At this point, big data problems arise, due to the gigantic size of databases. Thus, it is necessary to develop new computational methods to aid in the data processing and analysis [82].

2. System behaviour analysis. The observed behaviour of the organism caused by changing the environment, by introducing external elements, or by changing parts of the organisms itself (i.e. mutations) [88] is analysed in order to understand its functioning.

3. System control. Once the structure and the behaviour has been understood, systems biology aims at controlling the organism, leading the organism to maintain certain physical properties, or to improve towards a preferrable physiological state.

4. System design. Ultimately, it is desired to develop technologies that lead to design biological systems with the aim of providing cures for diseases [89], or, for example, what amount of which substrate leads a particular cell to produce certain protein of industrial interest.

The impact of systems biology in biotechnological processes is so great that the term "industrial systems biology" is today very common within this kind of industries [90, 91]. Measurement, monitoring, modelling and control (the so-called M3C methodology) are critical for obtaining high value-added biochemicals [92]. The purpose of industrial systems biology is to use the biological knowlege acquired on microorganisms to engineer efficient cell factories with the ultimate goal of converting raw substrates into different products. For example, the baker's yeast *Saccharomyces cerevisiae* can be used for transforming corn and wheat to commercial end-products as antibiotic, enzymes or vitamis [77].

The next sections in this chapter describe each omic science and its relationships with the subsequent biological levels. These fields are described as deep as needed for understanding the problems faced in this thesis. Therefore, some biological layers and methodologies are explained in more detail than others.

3.2 Genomics and transcriptomics

Cells are among the smallest biological entities studied in systems biology in order to understand the behaviour of the organism to which they belong. The genes, which are the hereditary units of biological organisms, are localized along the double-stranded DNA chains [78]. DNA, discovered back in 1953 by Watson and Crick [93], is formed by chains of four types of monomers: adeninde (A), guanine (G) cytosine (C) and thymine (T). The complete DNA sequence, the so-called genome, contains information about the whole set of proteins that the organism can synthesize [94].

The gene expression in cells is a two-step process. First the DNA is transcribed into RNA molecules or transcripts [78], which are single-stranded molecules with the same monomers as DNA but substituting T by uracil (U). During transcription, genes are used as a template to synthesize shorter molecules (RNA polymers). Finally, transcripts are translated into proteins [94], which are the ultimate molecules that control and establish the cellular biochemical status. During translation, the information in the RNA is read by codons (groups of three monomers), and each codon specifies a single amino acid in the resulting protein. There are 20 amino acids in total.

One way of studying the relationships between the genes and transcripts and the behaviour of the cell consists of performing mutations in the RNA and evaluate the changes in the organism [88]. However, only a small part of the whole genome (3-8% in humans) is considered functional, i.e. it encodes proteins or functional RNA [95]. This means that most of the DNA may have other structural or regulatory functions [78].

Finally, with the systems biology paradigm, researchers have focussed not only in identifying, listing and describing genes and transcripts but also in studying the way they interact among themselves. This way, gene regulatory networks (GRN) and transcriptomic networks can be built based on these interactions, being the genes or transcripts the nodes and the interactions the edges [82]. These networks are usually inferred using reverse engineering procedures, based on experimental or simulated measurements [5, 82, 96–98].

3.3 Proteomics

The term proteome or proteomics was coined in 1995 [99]. Proteomics complements gene sequence data with protein knowledge about where, in which quantity and under what conditions proteins are expressed [100]. The research fields of genomics and proteomics are remarkably different, but, as commented on the previous section, they are strongly related. The integration of both omic sources of information, via protein-DNA interactions data and gene expression data [101–104], increases the accuracy of techniques dealing with biological networks [82].

Proteomics is a data-rich environment, since proteins participate somehow in almost all biological processes and they also have diverse properties, which contribute to our systematic understanding of organisms [105]. The original goal of proteomics was the identification of the whole set of proteins expressed by organisms. Once this was (almost) achieved, the current goals of proteomic research became more diverse and directed toward the determination of diverse protein properties in biological systems, including sequence, quantity, state of modification, interaction with other proteins, activity, subcellular distribution and structure [105].

During the last decade there has been an increasing number of studies of protein-protein interactions (PPIs) and the effect that these interactions cause on a wide range of biological processes [106]. PPIs are defined as physical contacts that take place in cells through molecular docking [107]. Proteins work typically linked to other molecules including lipids, nucleic acids or other proteins [108]. Biological activity usually arises from the association of several proteins, which form protein complexes [109]. In viruses, interactions between proteins play vital roles in many processes during infection such as virus trafficking between the nucleus and the cytoplasm, formation of replication complexes, assembly of virions, or virus transmission to other cells. Traditionally, PPIs have been studied using methods such as coimmunoprecipitation or chromatography [110]. However, over the past decade two experimental strategies have been used to detect these interactions: yeast two-hybrid (Y2H) [107, 111, 112] and affinity purification coupled with mass spectrometry (AP-MS) [113]. Additionally, bimolecular fluorescence complementation (BiFC) [114, 115] has grown in popularity during the last few years because it allows PPIs visualization in living cells, which is a key aspect to understand their cellular functions.

PPIs form networks of linked proteins which are called consequently PPINs [107]. PPINs represent a map of physical contacts or functional interactions between proteins [107]. PPINs can be seen as a visual representation of the complete map of interactions that a system (pathway, cell, living organism) establishes in a particular moment and for a certain time window. Detection methods (specially Y2H) opened the possibility to tackle PPIs on a genome wide scale, producing complete PPINs, which have been called interactomes [116–119]. Viral PPINs have also been developed [120, 121], revealing quite useful biological information.

3.4 Metabolomics and fluxomics

The lowest biological level analysed in this chapter is the metabolic layer. As seen in the previous section, genes are associated to proteins, although it is not a one-to-one relationship. Genes and proteins are finally associated to metabolic reactions, forming the gene-protein-reaction (GPR) associates, thus enabling the production and consumption of metabolites [122].

When concatenating reactions, a metabolic network is built, describing how the initial metabolites (substrates) in the network are produced and consumed in a thermodynamically feasible way until reaching the end-metabolites or products. In this way cellular metabolism is represented by all these reactions, involved in the conversion of the carbon source into the building blocks needed for macromolecular biosynthesis [123].

The general aim of modelling metabolic systems consists of studying:

- The steady state behaviour of organisms, that is, the metabolic flux distributions crossing the metabolic network long after stimuli.
- The transient metabolite concentrations, which can be extended to study the fluxes in non-steady state conditions, since the concentrations of the

metabolites are themselves function of the fluxes and the properties of the enzymes that activate them [123].

The first approach employs the stoichiometry and reversibility associated to the reactions in the metabolic network, and establishes the structure onto which the behaviour of the organism is studied [75, 124]. This mathematical framework is explained in detail in the next subsections.

Regarding the second approach, the metabolic structure is the basic requirement to model the intracellular kinetics of the organism [75, 125]. In this case, the state of the network at a particular time point of the biological process is defined by the concentration of each metabolite in the cell, and metabolites may interact via one or more reactions. Each reaction is represented by an ordinary differential equation (ODE) relating the quantity of reactants (or inputs) to the quantity of postreaction (or output) products, according to a reaction rate and other parameters [88]. Since metabolic networks may have hundreds of reactions, the corresponding system of differential equations is very difficult to solve. However, when given an initial conditions of the network, the concentration of the metabolites along time can be simulated to produce a state transition path or trajectory, i.e. the succession of states adopted by the network over time [88]. Other methodologies proposed for studying non-steady state data are kinetic modeling [126], ^{13}C-metabolic flux analysis (MFA) [127], dynamic flux balance anlysis (FBA) [128], and a recently proposed approach combining time-resolved metabolomics and dynamic FBA (MetDFBA) [129].

Different techniques have been proposed in the literature to measure metabolites concentrations in experimental cultures, e.g. ^{13}C MFA, gas chromatography - mass spectrometry (GC-MS) and liquid chromatography - mass spectrometry (LC-MS) [123]. The early stages of the experiment, when the transient concentrations change along time, can be used to obtain the non-steady state metabolic fluxes. When the concentrations become stable, the steady state metabolic flux distribution is reached.

3.4.1 First principle models

First principles-based models of microbial systems can be developed to describe the principles that govern cellular behaviour and achieve a predictive understanding of cellular functions [130–132]. Metabolic networks are modelled assuming that certain constrains operate under steady-state conditions, such as environmental constraints [133], regulatory constraints [134, 135], gene expression data [136], mass balances or reactions irreversibilities [137] (the so-called constraint-based perspective) [122, 138, 139]. The imposed constraints define a solution space that encloses all the possible states of the network (i.e. flux distributions through the reactions).

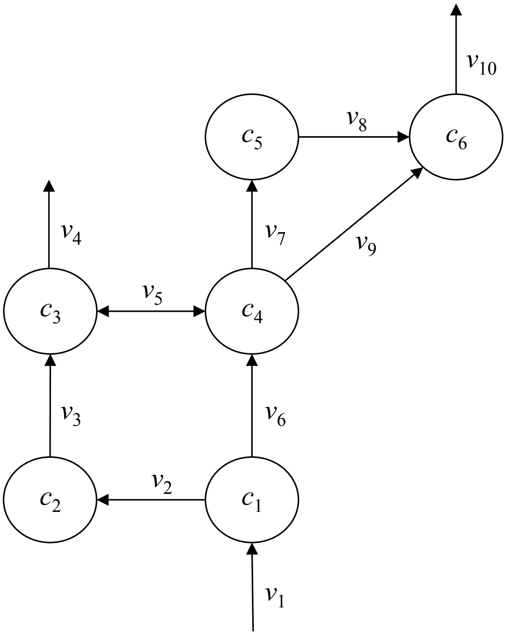


Figure 3.2: Small example of a metabolic network.

3.4.2 Stoichiometric modelling

To build a constraint-based model, the stoichiometric information embedded in the metabolic network (i.e. metabolites or cofactors involved in each reaction) must be arranged into an $N \times K$ matrix $\mathbf{S}$ (the so-called stoichiometric matrix). Rows of this matrix represent the N metabolites, columns the K metabolic reactions and each element (n, k) the stoichiometric coefficient S_{nk} of the nth metabolite in the kth reaction. A value of $S_{nk} = -1$ indicates that the nth metabolite is consumed by the kth reaction. In contrast, a $S_{nk} = 1$ indicates the nth metabolite is produced by the kth reaction. Finally, a value of $S_{nk} = 0$ stands for the nth metabolite is not involved in the kth reaction.

Figure 3.2 shows an example of a small metabolic network with 6 metabolites and 10 reactions, being one of them reversible (v_5), i.e. the reaction can be produced in both directions. The stoichiometric matrix $\mathbf{S}$ of this network has 6 rows and 10 reactions (see Table 3.1).

The stoichiometrix matrix is used, in combination with the flux vector $\mathbf{v} = [v_1, ..., v_K]$, the metabolites concentration $\mathbf{c} = [c_1, ..., c_N]$ and the specific growth rate of the cell μ, to represent the mass balances through the metabolic network. The ODE describing this process is as follows:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
c_1	1	-1	0	0	0	-1	0	0	0	0
c_2	0	1	-1	0	0	0	0	0	0	0
c_3	0	0	1	-1	-1	0	0	0	0	0
c_4	0	0	0	0	1	1	-1	0	-1	0
c_5	0	0	0	0	0	0	1	-1	0	0
c_6	0	0	0	0	0	0	0	1	1	-1

Table 3.1: Stoichiometric matrix for the network detailed in Figure 3.2.

$$\frac{d\mathbf{c}}{dt} = \mathbf{Sv} - \mu\mathbf{c} \tag{3.1}$$

This equation is called the dynamic mass balance equation, and describes the evolution of the concentration of each metabolite over time [139]. In stoichiometric modelling, the dynamic intracellular behaviour is disregarded on the basis assumption of pseudosteady state for the internal metabolites [137]. This assumption is supported by the observation that intracellular dynamics are much faster than extracellular dynamics. Therefore, it is sensible to assume that these compounds reach the steady state instantaneously and, hence, its transient behaviour can be omitted. In addition, the dilution term $\mu\mathbf{c}$ is also discarded because it is generally much smaller than the fluxes affecting the same metabolite. Under these considerations, the general equation can be expressed as:

$$\mathbf{Sv} = \mathbf{0} \tag{3.2}$$

This equation constrains the K-dimensional space of feasible solutions. Also, an extra constraint is added, assuming that some of the fluxes of the metabolic network flow only in one direction:

$$\mathbf{Dv} \geq \mathbf{0} \tag{3.3}$$

where $\mathbf{D}$ is a $K \times K$ diagonal matrix with binary values: 1 for the irreversible fluxes and 0 for the reversible ones. In the example shown before, $\mathbf{D}$ would be a diagonal matrix with ones in the diagonal except in position $D_{5,5} = 0$. The main direction in reaction v_5 is considered when c_3 is consumed to produce c_4 (see Table 3.1).

Finally, a maximum value for each of the K flux values is computed:

$$\mid v_k \mid \leq \mid v_{k,max} \mid \tag{3.4}$$

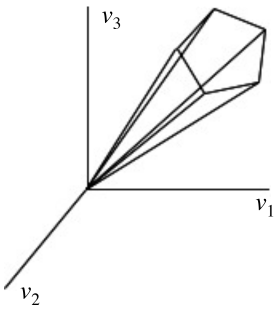


Figure 3.3: Example of a feasible space solution with 3 metabolic fluxes.

The combination of the constraints imposed by Equations 3.2-3.4 define a space (a bounded convex cone) of feasible steady-state flux distributions (see Figure 3.3): only flux vectors that fulfill Equation 3.2-3.4 are considered valid cellular states.

Metabolic networks can be studied bearing in mind that some reactions are reversible and others not. However, reversible fluxes can be split in two different ones in order to assume that all fluxes are irreversible. Following this approach, v_5 has to be split, in Figure 3.2, into two fluxes v_{5_1} and v_{5_2}, the first one consuming c_3 to produce c_4 and the other one consuming c_4 to produce c_3.

3.4.3 Network-based pathway analysis: elementary modes (EMs)

A metabolic network can be seen as a graph, where the metabolites are the nodes and the reactions are the edges (see Figure 3.2). Network-based pathways analysis is founded on that concept. This methodology is not focused on the flux values through the reactions, but on the thermodinamically feasible pathways from the inputs to the outputs [85]. Within this field, the concept of elementary mode (EM) is key. The set of EMs arises from the stoichiometric matrix $\mathbf{S}$, and each EM is defined as a minimal set of cellular reactions able to operate at steady-state, with each reaction weighted by the relative flux they need to carry for the mode to function [140].

The set of EMs is obtained from convex analysis [141] and it is unique for a given metabolic network. The EMs are usually organized in a data matrix, $\mathbf{EM} = [\mathbf{p}_1 \ldots \mathbf{p}_Z]$ $(K \times Z)$, having the Z EMs by columns, the K reactions in the metabolic network by rows, and the relative fluxes in its entries. Since this set represents a convex basis, any particular steady-state flux distribution $\mathbf{v}^{\mathrm{T}} = [v_1 ... v_K]$ can be obtained as a non-negative linear combination of EMs:

$$\mathbf{v} = \sum_{z=1}^{Z} \lambda_z \mathbf{p}_z \tag{3.5}$$

where K matches the number of reactions in the network, $\mathbf{p}_z^{\mathrm{T}} = [p_{z1} \dots p_{zK}]$ is the zth EM and λ_z is the positive weighting factor multiplying it. In the previous equation all EMs are used, but most of them are multiplied by a $\lambda_z = 0$. Therefore, any flux distribution can be represented using a positive linear combination of some, E, EMs.

Current algorithms for the computation of EMs face a common problem when dealing with highly interconnected metabolic networks [142]. In such cases, the combinatorial explosion of the number of EMs renders the analysis of large networks difficult. Very recently, two new methods [143, 144] have been proposed to compute the EMs in large networks in a fast and efficient way.

Some methods have been proposed in the literature to select a set of representative or active EMs. One such attempt is the concept of the α spectrum [145], which involves a linear optimization to determine how the extreme pathways (EPs) (a systemically independent subset of EMs) contribute to a given steady-state flux distribution. This algorithm allows the determination of maximum and minimum possible weightings for each extreme pathway. A different approach involves the quadratic decomposition of a single steady-state flux into a set of EMs [146]. In this algorithm, a particular set of EMs is chosen, based on the minimization of the weighting vector length, i.e. $\boldsymbol{\lambda}^{\mathrm{T}} = [\lambda_1 \dots \lambda_E]$. A reinterpretation of this methodology was also proposed by projecting the flux space into the yield space [147], thus restricting the search for active EMs in a bounded convex space.

3.4.4 Possibilistic consistency analysis

When combining a stoichiometric model of an organism with experimental measurements it is mandatory to check whether the measurements comply with the proposed constraint-based model. The simplest consistency analysis could be performed by checking that the flux states shown by cells fulfill the constraints imposed by the model (see Equations 3.2-3.4) [132]. However, this simple approach would be impractical because measurements are imprecise and do not exactly satisfy the constraints. Such difficulty is overcome by taking into account uncertainty as follows:

$$w = v + e \tag{3.6}$$

where e represents the deviation error between an actual flux v and its measured value w.

The consistency analysis can be also formulated as a possibilistic constraint satisfaction problem [148]. The basic idea is that a flux vector fulfilling Equations 3.2 and 3.3, and compatible with the measurements will be considered as "possible", otherwise as "impossible". This can be refined to cope with measurements errors by introducing the notion of "degree of possibility" [149].

This degree of possibility provides an indication of the consistency between the model and the measurements. A possibility equal to one must be interpreted as complete agreement between the model and the original measurements. Lower values of possibility imply that certain error in the measurements is needed to find a flux vector fulfilling the model constraints.

Possibilistic consistency analysis consists of four steps:

1. **Model and measurements constraints**. Firstly, the constraints conforming the model (Equation 3.2-3.4) are considered. Then measures of (some) extracellular fluxes are incorporated as additional constraints (Equation 3.7):

$$\begin{cases} \mathbf{w} = \mathbf{v} - \boldsymbol{\varepsilon}_1 + \boldsymbol{\mu}_1 - \boldsymbol{\varepsilon}_2 + \boldsymbol{\mu}_2 \\ \boldsymbol{\varepsilon}_1, \boldsymbol{\mu}_1, \boldsymbol{\varepsilon}_2, \boldsymbol{\mu}_2 \geq \mathbf{0} \\ \boldsymbol{\varepsilon}_2 \leq \boldsymbol{\varepsilon}_{2,max} \\ \boldsymbol{\mu}_2 \leq \boldsymbol{\mu}_{2,max} \end{cases} \tag{3.7}$$

   where vectors $\mathbf{v}$ and $\mathbf{w}$ represent the actual and measured fluxes, respectively. Note that both vectors differ because of errors and imprecision (uncertainty). This uncertainty is represented by the vectors of slack variables $\boldsymbol{\varepsilon}$'s and $\boldsymbol{\mu}$'s.

2. **Possibility**. The basic building block of possibility theory is a user-defined possibility distribution $\pi(\delta) : \Delta \to [0, 1]$, where $\delta = \{\mathbf{v}, \mathbf{w}, \boldsymbol{\varepsilon}, \boldsymbol{\mu}\}$ denotes each candidate solution of Equation 3.7 in Δ. This function defines the possibility of each solution δ in Δ, ranging between impossible ($\pi = 0$) and fully possible ($\pi = 1$). Among different possible choices, a simple way to define possibility is using a linear cost index such as:

$$Z(\delta) = \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\varepsilon}_1 + \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\mu}_1 \tag{3.8}$$

   and define the possibility of each solution δ as follows:

$$\pi(\delta) = e^{-Z(\delta)}, \quad \delta \in \Delta \tag{3.9}$$

   where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are row vectors of user-defined, sensor accuracy coefficients.

The interpretation of Equations 3.7-3.9 may be: $\mathbf{v}_m = \mathbf{w}$ is fully possible; the more $\mathbf{v}$ and $\mathbf{w}$ differ, the less possible such situation is.

3. **Representing uncertainty**. Two pairs of vectors of slack variables are chosen to represent the uncertainty of each measurement: $\boldsymbol{\varepsilon}_2$ and $\boldsymbol{\mu}_2$ define an interval of fully possible values, and $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\mu}_1$ penalise values out of it (with weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$). This is achieved choosing two vectors of bounds. Hence, in all computations the uncertainty of each measurement is represented as follows:

   (a) Full possibility ($\pi = 1$) is assigned to values with less than ± 5% of deviation.

   (b) Larger deviations are penalised, so values with a deviation equal to 20% have a possibility of $\pi = 0.1$ (and those with a deviation equal to ± 10% have a $\pi \approx 0.5$).

   (c) Uncertainty is considered as symmetric, and thus $\boldsymbol{\alpha} = \boldsymbol{\beta}$.

   This is achieved choosing bounds $\boldsymbol{\varepsilon}_{2,max}$ and $\boldsymbol{\mu}_{2,max}$ and weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for each measurement: (i) implies that $\boldsymbol{\varepsilon}_{2,max} = \boldsymbol{\mu}_{2,max} = 0.05\mathbf{w}$, and (ii) defines $\boldsymbol{\alpha}$, noticing that $0.2\mathbf{w} = \boldsymbol{\mu}_{1,20\%} + \boldsymbol{\mu}_{2,max}$, then $\boldsymbol{\alpha} = -log(0.1)/(0.2 - 0.05)/\mathbf{w}$.

4. **Possibilisitic consistency evaluation**. The most possible solution of the constraint-satisfaction problem is the maximum possibility (minimum-cost) solution, which can be obtained solving a linear programming problem (LP):

$$Z^{min} = \min_{\boldsymbol{\varepsilon},\boldsymbol{\mu},\mathbf{v}} Z \tag{3.10}$$

   subject to Equations 3.2-3.4 and the experimental measurements. This solution has an associated degree of possibility:

$$\pi^{mp} = e^{-Z^{min}} \tag{3.11}$$

This value, π^{mp} in [0,1], grades the consistency between model and measurements. A possibility equal to one must be interpreted as complete agreement, while lower values imply that there is some error in the measurements, the model or both, which severity depends on how the uncertainty has been defined (see above) [132, 150].

3.5 For every omic science: Network inference

The problem of inferring complex networks is frequently addressed in several research areas within biology such as metabolomics, proteomics and genomics. In the context of cellular networks the individuals (network nodes) are biochemical entities such as metabolites, proteins or genes. Many different approaches have been applied to date in this area, none of which can be singled out as the best one for all problems. Comparisons often find large discrepancies between the predictions of different algorithms, making network inference an open problem in bioinformatics research [151–155].

Network reconstruction usually begins with collecting data from each individual that participates in the network. Then, based on the relationships detected among them, links between the individuals are established. When no prior knowledge is introduced, the reconstruction of complex networks depends solely on the available data. In many cases, the data collection represents a challenge itself when experimental measurements are involved.

Once the data is collected, different approaches can be applied to reverse engineer networks. Among other techniques, e.g. correlation-based procedures, information theory concepts such as entropy and mutual information [156, 157] can be used to establish relationships among entities in a network. Information theory methods have strengths such as good scalability and the ability of detecting nonlinear relationships, and are widely used for the reverse-engineering of biological networks [97]. Examples of this procedures [98] are: Context Likelihood of Relatedness (CLR) [158], used for inferring transcriptional interactions, the Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) [159] and time-delayed ARACNE [160], used for GRN, MRNET [161], a maximum relevance/minimum redundancy (MRMR)-based method used for inferring genetic networks from microarray data, or MI3 [162], applied in transcriptional regulatory networks.

3.5.1 Mutual information distance and entropy reduction (MIDER) method

The theoretical foundation of network inference method MIDER is information theory, which is based on the concept of entropy as defined by Shannon [156]. The entropy of a discrete variable X with alphabet χ and probability mass function $p(x)$ is:

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x) \tag{3.12}$$

where the logarithm to the base 2 is usually chosen. For continuous variables the $\sum$ is replaced by $\int$. It is possible to calculate the joint entropy of a pair of variables (X, Y) as $H(X, Y) = -\sum_x \sum_y p(x, y) \log p(x, y)$. Another important quantity, conditional entropy $H(X|Y)$, can be calculated as the entropy of a random variable Y conditional to the knowledge of a second one, X:

$$\begin{aligned} H(Y|X) &= \sum_x p(x) H(Y|X = x) = -\sum_x p(x) \sum_y p(y|x) \log p(y|x) = \\ &= -\sum_x \sum_y p(x, y) \log p(y|x) \end{aligned} \tag{3.13}$$

The relation between joint and conditional entropy is expressed as $H(X, Y) = H(X) + H(Y|X)$. A related concept is the relative entropy, which measures the distance between two distributions p and q and is defined as:

$$D(p||q) = -\sum_x p(x) log \frac{p(x)}{q(x)} \tag{3.14}$$

It has two important properties: it is always non-negative, and it is zero if, and only if, $p = q$. The relative entropy between the joint distribution $p(x, y)$ and the product distribution of two variables, $p(x)p(y)$, is called mutual information [157], I, that is:

$$\begin{aligned} I(X, Y) &= \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)} = H(X) - H(X|Y) = \\ &= H(X) + H(Y) - H(X, Y) \end{aligned} \tag{3.15}$$

The concept of mutual information can be used to detect relationships between variables of any kind, since it is a measure of the amount of information that one random variable contains about another. Indeed, it has been used as the basis of many inference methods, e.g. the aforementioned ones [98, 158, 160–162], among others. Mutual information is a symmetric measure that does not assume any property of the dependence between variables, such as linearity or continuity. Hence it is more general than linear measures such as the correlation coefficient, and it has been shown that it is capable of inferring more interactions [158]. Intuitively, if two components of a network interact strongly, their mutual information will be large; if they are not related, their mutual information will be (theoretically) zero, or (in practice, when estimated from data) very small.

MIDER [98] uses these information theory concepts to infer relationships between variables, and to discriminate between direct and indirect interactions. Its core feature is entropy reduction, which consists of calculating the reduction of the entropy of a variable Y caused by other variables in the network. Theoretically, if a variable Y is completely independent of a set of variables $\mathbf{X}$, then $H(Y|\mathbf{X}) = H(Y)$; otherwise $H(Y|\mathbf{X}) < H(Y)$. Therefore, if a subset of variables $\mathbf{X}$ reduces the entropy of Y to the minimum (i.e. if adding an additional variable to $\mathbf{X}$ does not produce further reductions in the entropy of Y), we have found the complete set of connections between Y and the remaining variables in the network.

The MIDER methodology uses the aforementioned concepts in the following procedure [98]:

1. Based on the estimation of time-lagged multidimensional entropies and mutual information, MIDER estimates the distance between variables and then projects the distance matrix onto a 2-D space using multidimensional scaling.

2. Then, the links between variables are established based on the first, second and third order conditional entropies.

3. The strength of a link between two variables is calculated from the relative reduction of the entropy of the first variable caused by the addition of the second variable.

4. Finally, directionality is assigned to the inferred links. The direction of a link connecting two variables X and Y is the one that gives the maximum transfer entropy. The transfer entropy from X to Y is calculated as:

$$T_{X\rightarrow Y} = H(Y^t|Y^{t-\tau}) - H(Y^t|Y^{t-\tau}, X^{t-\tau}) \tag{3.16}$$

   where t indicates the lag - obtained in the first step - for the $X - Y$ pair.

In [98], the performance of MIDER was benchmarked against several methods from the literature: CLR, ARACNE, MRNET, MI3, and TD-ARACNE, obtaining competitive results.

Chapter 4

Material

4.1 Hardware

All the computations in this document have been carried out with a notebook Intel Core i7, CPU 2,9 GHz, 8GB of RAM.

4.2 Software

The software packages used here are:

- Mac OS X 10.11.13
- MATLAB 2012b (mainly), 2013a and 2014b (The MathWorks, Inc.).

Most computations of this thesis have been run in MATLAB environment, using own code, including method implementation and computation routines. Additionally, two software packages are derived from this document:

- Missing Data Imputation Toolbox (MDI Toolbox).
- Principal elementary mode analysis Toolbox (PEMA Toolbox).

The first one for imputing missing values in data matrices (see Chapter 10) and the second one to fit PEMA models using metabolic flux data (see Chapter 6).

Many other source code files are derived from contributions in this thesis. The URLs of these freely available files will be given when presented within chapters. When no URL is available, an appendix with the MATLAB code will be included at the end of the corresponding chapter.

Other software packages are used throughout the present work at specific steps within chapters:

- COnstraints Based Reconstruction and Analysis toolbox (COBRA toolbox) [163].
- Multivariate Exploratory Data Analysis (MEDA) Toolbox [164].
- MCR-ALS Toolbox [35, 36].
- EFMTOOL [165].
- COmplex PAthway SImulation (COPASI) software [166].
- N-way toolbox [167].
- ProSensus MultiVariate (ProSensus, Inc.) [168].
- Phi toolbox [60].
- MIDER Toolbox [98].
- PLS Toolbox (Eigenvector Research, Inc.) [169].

COBRA is used to generate metabolic flux solutions in Chapter 5. MEDA and MCR-ALS Toolbox are used in the same chapter to apply MEDA and MCR-ALS methods, respectively. EFMTOOL is used in Chapters 6 and 7 to enumerate the complete set of elementary modes in metabolic networks. COPASI is used for simulating non-steady state metabolic fluxes in Chapters 7. N-way toolbox is used in Chapters 7 and 9 for building NPLS-DA models. ProSensus MultiVariate is used in Chapter 8 for fitting the PLS models. MIDER is used in tandem with missing data imputation and outlier detection and correction own procedures in Chapter 11. Finally, specific methods within Phi and PLS toolbox are used in Chapters 10 and 14, respectively, for comparative studies: the NLP approach from Phi for missing data imputation, and piecewise direct standardisation (PDS) from PLS Toolbox for calibration transfer between NIR instruments.

4.3 Biological organisms

The novel methods and modelling contributions presented in this thesis have been applied to different biological organisms. In chapters 5, 6 and 7 different metabolic models are proposed for:

- *Pichia pastoris*: a yeast widely used for protein production in biotechnological industries.

- *Escherichia coli*: a bacterium commonly present in animal intestines.
- *Saccharomyces cerevisiae*: known as the baker's yeast, used also in fermentation industries.

For *P. pastoris* different metabolic models are used, since the methodologies proposed here have been developed at many sites during the PhD (UPV, UNL, UvA). For *S. cerevisiae* two metabolic models are proposed in Chapter 7 due to data availability reasons, i.e. the simulated model have slightly different metabolites than the actual experimental measurements.

In Chapter 8 the PPIN of *Potyvirus*, a pathogen of many plant species, is studied, using experimental data from *Tobacco etch virus* mutants, grown in *Nicotiana tabacum* plant.

Finally, two food products, biological organisms at macro level, are used here:

- Multivariate image analysis (MIA) models are fitted in Chapter 9 to discriminate between sound oranges and fruits infected by another organism, *Penicilium digitatum* fungus.
- Novel calibration transfer procedures are tested in Chapter 14 using spectral data from corn samples.

4.4 Datasets

Different data sets are used in this document to evaluate the performance of novel methods and to compare them to the established ones. These datasets are very diverse. Some case studies use data derived from biological organisms, e.g. concentrations, metabolic networks, images, and others are benchmarks commonly used in the literature for method comparisons, such as synthetic networks or spectral measurements of organisms or end-products.

Due to the large quantity of datasets used in this thesis, being each one used in one single contribution (with few exceptions), each dataset will be presented at its corresponding chapter.

Part II

Modelling biological organisms

Chapter 5

Metabolic flux understanding

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

[1] González-Martínez, J.M., Folch-Fortuny, A., Llaneras, F., Tortajada, M., Picó, J. & Ferrer. A. Metabolic flux understanding of *Pichia pastoris* grown on heterogeneous culture media. *Chemometrics and Intelligent Laboratory Systems* **134**, 89-99 (2014).

[3] Folch-Fortuny, A., Tortajada, M., Prats-Montalbán, J.M., Llaneras, F., Picó, J. & Ferrer, A. MCR-ALS on metabolic networks: Obtaining more meaningful pathways. *Chemometrics and Intelligent Laboratory Systems* **142**, 293-303 (2015).

5.1 Introduction

Within systems biology, first principles-based models of microbial systems are employed to discern the principles that govern cellular behaviour and achieve a predictive understanding of cellular functions. The development of this type of models based solely on fundamental or knowledge information has the drawback that the unknown part of the process is not represented as well as some of the underlaying assumptions (e.g. specific kinetics of the reaction system, unknown dynamics, values of the model parameters, objective functions) may not be valid for all the metabolic possible states of the network [170, 171]. To address this problem, grey models that combine knowledge-based models, which fit the theoretical behaviour, and empirical models, which fit any remaining systematic variation, can be used [172].

In the context of grey modelling, there are different approaches to descompose the data into the three types of variation (known causes, unknown causes and residuals) [173], which be roughly classified into three categories. The first category are the models based on known constraints. There exist general frameworks that enable to impose very specific constraints on each type of information, e.g. observed experimental information [174] or transformations on the original variables [175]. These methods are based on the projection of a data matrix, followed by multivariate model decomposition. PCA is one of the most applied multivariate statistical projection methods to reveal the internal structure of the cell. This analysis is commonly preceded by a MC sampling in order to produce a data set of possible states or feasible solutions from which the PCA elucidates the meaningful principal components (PCs) [176–178]. PCA has also been compared to other multivariate techniques, such as Multivariate Linear Regression [178] and PARAFAC [179, 180] in the field of systems biology. Partial Least Squares regression (PLS) has been applied directly [181, 182] and combined with hierarchical clustering PLS (HC-PLSR) [183] to deal with situations where the input-output relations (e.g. the effect of the substrates consumption of the cell or the environmental conditions in the production of a particular protein) are highly nonlinear or non-monotone. Recently, grey component analysis (GCA) has been proposed using a cost function to maximise the interpretability of the solutions by forcing the decomposition towards the direction of the prior information - a chemically or biologically meaningful solution - [184].

A second strategy is formed by methods based on introducing *a priori* knowledge by means of mathematical relations that describe the system behaviour or dynamics. The starting point is some specific structure based on first principles mathematical relations, where some functions must be estimated. Different tools can be used to calculate these functions, such as artificial neural networks (ANNs) [185] or kalman filters [186, 187].

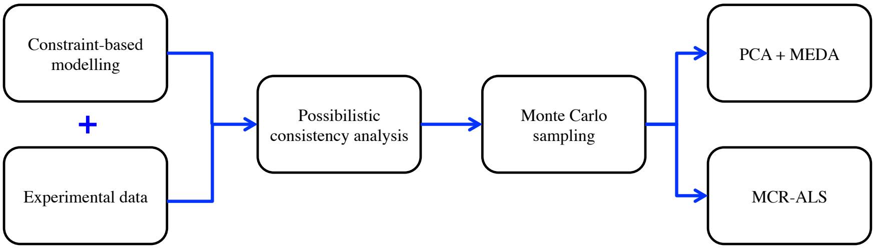

Constraint-based modelling
+
Experimental data
Possibilistic consistency analysis
Monte Carlo sampling
PCA + MEDA
MCR-ALS

Figure 5.1: Flow diagram of the grey modelling.

Finally, a third category are the methods based on incorporating the fundamental knowledge through constraints on the modelling algorithms. For instance, some model parameters can be forced to have values within certain regions in the parameters space [188]. Projection to latent pathways (PLP) [189] has been recently formulated as a modification of the PLS regression algorithm by using the concept of EMs (more details in Section 3.4.3). This method is devoted to obtain a more biologically explanatory set of LVs relating the observed behaviour of the cell and its initial conditions.

The complexity of data available from microbial systems requires the design of sophisticated grey models that combine data-driven and knowledge-based information at different scales for biochemical process understanding. The main goal in this chapter is to use this hybrid framework to analyse the behaviour of the methylotrophic yeast *P. pastoris* [132], as a first step to analysing which conditions and through which reactions the cell achieves an optimal state for our interests. Several scenarios corresponding to different chemostat runs are collected from the literature [190–198] with the aim of starting the analysis with a rich data set of different culture conditions. A recently developed adaptation [132, 148] of the possibilistic theory [149] is applied in order to check the consistency between model and data. For the completion of the unmeasured data, a MC sampling method is applied to produce feasible flux solutions for the microbial system under study. At this point, two methodologies are compared here. First, a hard modelling approach, PCA in combination of MEDA, is applied to obtain a reduced number of orthogonal PCs explaining most of the variance of the collected and sampled data. Second, a soft-modelling method, MCR, is applied on the same data in order to i) include constraints in the model, to make it more biologically meaningful and ii) obtain non-orthogonal components. The whole process can be visualized in Figure 5.1.

This chapter is organized as follows. Section 5.2 presents the metabolic network reconstruction of the yeast *P. pastoris* and the different scenarios used in the study. Section 5.3 describes the grey modelling approach proposed in detail. Section 5.4 shows the results applying hard and soft methodologies. In section 5.5 the results between both approaches are compared. Finally, some conclusions are drawn in Section 5.6.

5.2 *Pichia pastoris* metabolic model

5.2.1 Metabolic network reconstruction

The constraint-based model, whose corresponding metabolic network is shown in Figure 5.2, has been used throughout this work. The model is a simplified representation of the whole metabolism of the yeast *P. pastoris*, meaning that only a reduced number of biochemical reactions has been included (45), from the larger amount available from genomic information (more than 1200). The reactions were selected on the basis of previous models found in literature, as lumped equivalents of more complex pathways. This model has been previously validated in [132, 150] and is the only one used in the referred experiments throughout this work. The model represents the most significant features of *P. pastoris* central carbon metabolism, including the main catabolic pathways of the yeast, such as glycolysis, the citric acid cycle, glycerol and methanol oxidation and fermentative pathways [150]. Anabolism is introduced through the pentose phosphate pathway and a general lumped biomass equation according to which growth is assumed to depend exclusively on key biochemical precursors. Branch-point metabolites, such as NADH, NADPH, AcCoA, oxalacetate and pyruvate, are considered in compartmentalized cytosolic and mitochondrial pools [190].

5.2.2 Experimental data set

In this chapter, experimental data from several fermentation runs with different *P. pastoris* strains have been taken from the available literature, building the different scenarios considered for the subsequent statistical analysis. For the sake of visualization, the 40 scenarios under study have been grouped attending to the experimental substrates (i.e. glucose, glycerol, methanol, and glycerol and methanol mixtures) (see Figure 5.3). Scenario A1 corresponds to a strain expressing the Fab fragment of the human anti-HIV antibody 3H6 [190]. Scenarios from B1 to B7, and C1 and C2, are from a strain expressing a *Rhizopus oryzae* lipase (ROL) [191, 192]. Scenarios from D1 to D10 come from a *P. pastoris* strain expressing and secreting recombinant avidin [193]. Scenario E1 has been obtained from a macrokinetic model for *P. pastoris* expressing recombinant human serum albumin (HSA) [194]. Scenarios from F1 to F7 come from a *P. pastoris* strain genetically modified to produce sea raven antifreeze protein [195]. Scenarios from G1 to G10 are obtained from a *P. pastoris* strain producing recombinant human chymotrypsinogen B [196]. Scenario H1 has been obtained from the continuous fermentation of a *P. pastoris* strain for the extracellular production of a recombi-

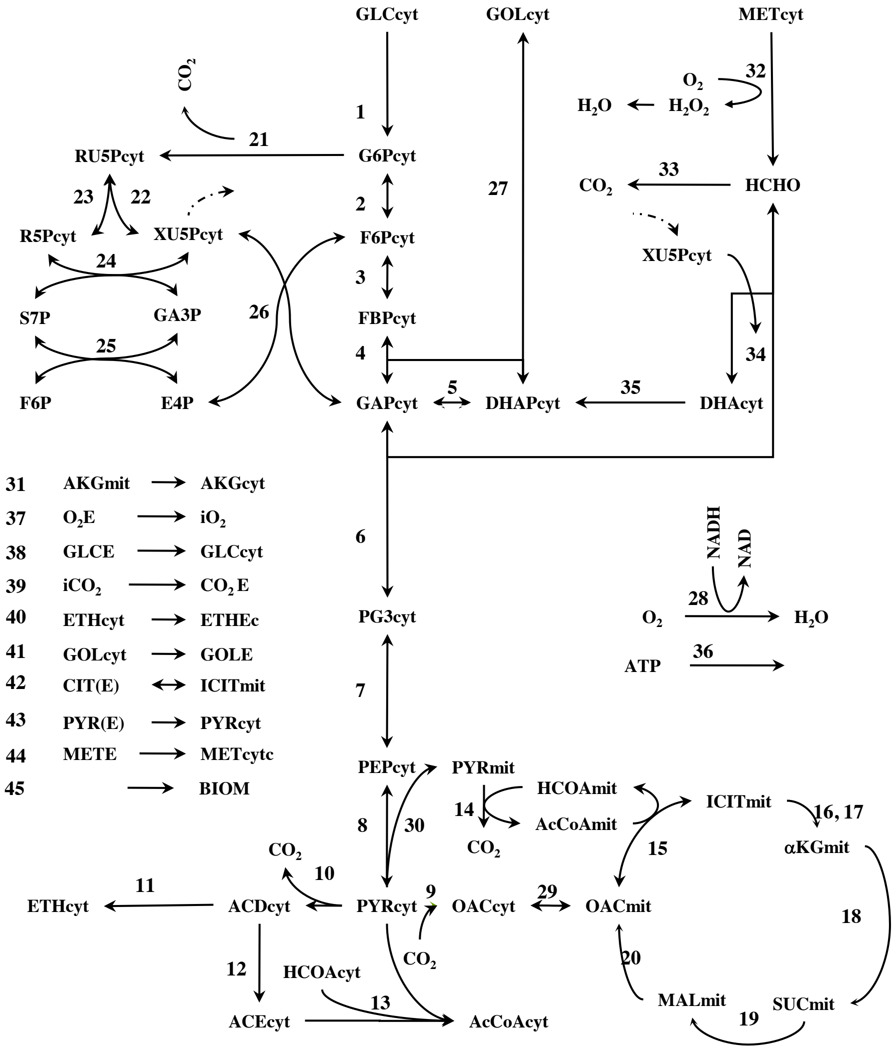


Figure 5.2: Metabolic network of *P. pastoris.*

nant ovine interferon protein [197]. Finally, scenario I1 comes from the expression of recombinant chitinase with a genetically modified *P. pastoris* strain [198]. The data for all these scenarios are detailed in Figure 5.3.

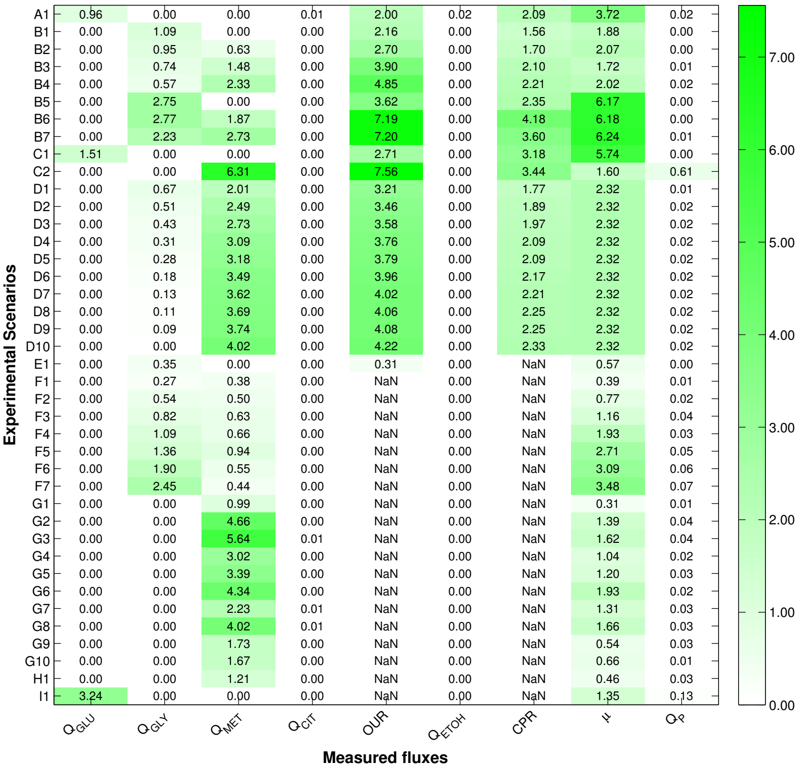
At this point, there is a paramount comment that is in due. Batch effects, which are defined as systematic non-biological variation between groups of samples (or batches) due to experimental artefacts [199–202], can be present in data collected from different cultures. In case that replicates of the same scenario are collected (i.e. same strain and same quantities of initial substrates) and the presence of batch effects is statistically confirmed, this artificial variation must be removed. Otherwise, the bias introduced by the non-biological nature of this kind of effects may confound true biological differences [201], affecting the results of statistical analysis. In this study, the scenarios within a single strain of *P. pastoris* have different initial substrate quantities (see Figure 5.3). Hence, the variation observed across scenarios can be due to these different initial conditions, which were applied with the aim of obtaining different flux values. This fact jointly with the scarcity of information about the experimentation conditions disable the possibility to straightforwardly confirm actual batch effects in data.

5.3 Grey modelling

5.3.1 Possibilistic consistency analysis

The first step of the grey modelling consists of validating whether the experimental scenarios taken from the literature are consistent with the metabolic model presented in the previous section. From each one of the 40 scenarios, the flux values through the external reactions, which are different depending on the initial conditions of each experiment, are validated against the stoichiometric modelling of the *P. pastoris*. The most possible solution for each scenario (i.e. experimental dataset) is computed to perform a possibilistic consistency analysis.

The possibility values (π) for the experimental scenarios are shown in Table 5.1. The majority of datasets are highly consistent with the model (65% are fully possible, and 87% have a possibility higher than 0.5). There are, however, 4 out of 40 datasets with a possibility lower than 0.25 (i.e. a possibility that is equivalent to an error of 14% in one measurement, or to an error of 8% in three measurements). These scenarios (B3, B4, C2, and E1) are not fully consistent with the model. The inconsistency can be due to a) limitations of the model, which may be unable to capture phenomena ocurring in those experiments, b) larger errors than expected in the data measured in those scenarios, or c) the two previous reasons acting simultaneously. For this reason, we decided to remove these scenarios (B3, B4, C2, and E1) so they are not considered in the following analysis.



	Q_{GLU}	Q_{GLY}	Q_{MET}	Q_{CIT}	OUR	Q_{ETOH}	CPR	μ	Q_P
A1	0.96	0.00	0.00	0.01	2.00	0.02	2.09	3.72	0.02
B1	0.00	1.09	0.00	0.00	2.16	0.00	1.56	1.88	0.00
B2	0.00	0.95	0.63	0.00	2.70	0.00	1.70	2.07	0.00
B3	0.00	0.74	1.48	0.00	3.90	0.00	2.10	1.72	0.01
B4	0.00	0.57	2.33	0.00	4.85	0.00	2.21	2.02	0.02
B5	0.00	2.75	0.00	0.00	3.62	0.00	2.35	6.17	0.00
B6	0.00	2.77	1.87	0.00	7.19	0.00	4.18	6.18	0.00
B7	0.00	2.23	2.73	0.00	7.20	0.00	3.60	6.24	0.01
C1	1.51	0.00	0.00	0.00	2.71	0.00	3.18	5.74	0.00
C2	0.00	0.00	6.31	0.00	7.56	0.00	3.44	1.60	0.61
D1	0.00	0.67	2.01	0.00	3.21	0.00	1.77	2.32	0.01
D2	0.00	0.51	2.49	0.00	3.46	0.00	1.89	2.32	0.02
D3	0.00	0.43	2.73	0.00	3.58	0.00	1.97	2.32	0.02
D4	0.00	0.31	3.09	0.00	3.76	0.00	2.09	2.32	0.02
D5	0.00	0.28	3.18	0.00	3.79	0.00	2.09	2.32	0.02
D6	0.00	0.18	3.49	0.00	3.96	0.00	2.17	2.32	0.02
D7	0.00	0.13	3.62	0.00	4.02	0.00	2.21	2.32	0.02
D8	0.00	0.11	3.69	0.00	4.06	0.00	2.25	2.32	0.02
D9	0.00	0.09	3.74	0.00	4.08	0.00	2.25	2.32	0.02
D10	0.00	0.00	4.02	0.00	4.22	0.00	2.33	2.32	0.02
E1	0.00	0.35	0.00	0.00	0.31	0.00	NaN	0.57	0.00
F1	0.00	0.27	0.38	0.00	NaN	0.00	NaN	0.39	0.01
F2	0.00	0.54	0.50	0.00	NaN	0.00	NaN	0.77	0.02
F3	0.00	0.82	0.63	0.00	NaN	0.00	NaN	1.16	0.04
F4	0.00	1.09	0.66	0.00	NaN	0.00	NaN	1.93	0.03
F5	0.00	1.36	0.94	0.00	NaN	0.00	NaN	2.71	0.05
F6	0.00	1.90	0.55	0.00	NaN	0.00	NaN	3.09	0.06
F7	0.00	2.45	0.44	0.00	NaN	0.00	NaN	3.48	0.07
G1	0.00	0.00	0.99	0.00	NaN	0.00	NaN	0.31	0.01
G2	0.00	0.00	4.66	0.00	NaN	0.00	NaN	1.39	0.04
G3	0.00	0.00	5.64	0.01	NaN	0.00	NaN	1.62	0.04
G4	0.00	0.00	3.02	0.00	NaN	0.00	NaN	1.04	0.02
G5	0.00	0.00	3.39	0.00	NaN	0.00	NaN	1.20	0.03
G6	0.00	0.00	4.34	0.00	NaN	0.00	NaN	1.93	0.02
G7	0.00	0.00	2.23	0.01	NaN	0.00	NaN	1.31	0.03
G8	0.00	0.00	4.02	0.01	NaN	0.00	NaN	1.66	0.03
G9	0.00	0.00	1.73	0.00	NaN	0.00	NaN	0.54	0.03
G10	0.00	0.00	1.67	0.00	NaN	0.00	NaN	0.66	0.01
H1	0.00	0.00	1.21	0.00	NaN	0.00	NaN	0.46	0.03
I1	3.24	0.00	0.00	0.00	NaN	0.00	NaN	1.35	0.13

Figure 5.3: Set of 40 experimental scenarios corresponding to *P. pastoris* chemostat cultures grown on glucose, glycerol and methanol mixtures. For each scenario, the values of measured fluxes belonging to substrate and product specific consumption and production are shown. The substrates are glucose (Q_{GLU}, corresponding to reaction 1 in Figure 5.2), glycerol (Q_{GLYC}, reaction 27), methanol (Q_{MET}, reaction 32), citrate (Q_{CIT}, reaction 42) and oxygen (OUR, reaction 28). The products are ethanol (Q_{ETOH}, reaction 40), carbon dioxide (CPR, reaction 39), biomass (μ, reaction 45) and protein (Q_P, reaction 46). Note that NaN values stand for missing measured external fluxes.

Scenario	**Group**	π
A1	glucose	1,000
B1	glycerol	1,000
B2	glycerol + methanol	0,739
B3	glycerol + methanol	0,246(*)
B4	glycerol + methanol	0,082(*)
B5	glycerol	1,000
B6	glycerol + methanol	0,819
B7	glycerol + methanol	0,319
C1	glucose	0,658
C2	methanol	0,052(*)
D1	glycerol + methanol	1,000
D2	glycerol + methanol	1,000
D3	glycerol + methanol	1,000
D4	glycerol + methanol	1,000
D5	glycerol + methanol	1,000
D6	glycerol + methanol	0,908
D7	glycerol + methanol	0,709
D8	glycerol + methanol	0,637
D9	glycerol + methanol	0,614
D10	methanol	0,500
E1	glycerol	0,065(*)
F1	glycerol + methanol	1,000
F2	glycerol + methanol	1,000
F3	glycerol + methanol	1,000
F4	glycerol + methanol	1,000
F5	glycerol + methanol	1,000
F6	glycerol + methanol	1,000
F7	glycerol + methanol	1,000
G1	methanol	1,000
G2	methanol	1,000
G3	methanol	1,000
G4	methanol	1,000
G5	methanol	1,000
G6	methanol	1,000
G7	methanol	1,000
G8	methanol	1,000
G9	methanol	1,000
G10	methanol	1,000
H1	methanol	1,000
I1	glucose	1,000

Table 5.1: Possibility values (π) for each scenario. Those scenarios that are not consistent (i.e. $\pi < 0.25$) with the constrained-based model are signaled with (*).

5.3.2 MC sampling

The experimental data found in the literature represent partial flux solutions, since few fluxes (9) of the metabolic network have been experimentally measured (see Figure 5.3). In this context, MC sampling methods can be used to produce complete feasible flux distributions across the cell, in this case 45 fluxes (see Figure 5.2), without adding any other assumption nor biasing (i.e. keeping the current uncertainty) [176–178, 203, 204]. This way, the available experimental data (measured fluxes) and the first principles knowledge captured by the model (stoichiometry) are coupled together, providing a new richer dataset amenable to further analysis with a multivariate statistical method.

In order to deal with experimental errors, external fluxes are allowed to vary within a defined range of values centered on the original mth measured value. The upper (lower) bound of this range is the sum (subtraction) of the measured value and the maximum value between ρ and a fraction ζ of the measured value:

$$UB_m = v_m + \max(\rho, \zeta v_m) \tag{5.1}$$

$$LB_m = v_m - \max(\rho, \zeta v_m) \tag{5.2}$$

where UB_m, LB_m are the upper and lower bounds for the MC sampling, v_m is a measured flux, and ρ and ζ are heuristic values recommendable to be set to $\rho = 0.001$ and $\zeta = 0.1$, respectively.

At this point, it is worth commenting that the feasible solutions for each scenario are obtained by sampling within the *slice* of the cone defined by Equations 3.2-3.4 and the experimental data, i.e. the measured fluxes reduce the feasible solution space from the initial cone, which is bounded only by the constraint-based model, to the portion of it fulfilling these specific experimental measurements. The complete procedure can be visualized in Figure 5.4.

Notice that there are scenarios lacking measurements of some external fluxes (see Figure 5.2). In the Monte Carlo sampling, these fluxes are allowed to vary within the whole slice of the cone defined by the measured external ones and the constraint-based modelling.

Once the sampling has been performed, using COBRA Toolbox, a feasible flux solution matrix $\mathbf{X}$ is built. $\mathbf{X}$ has the complete 3600 sampled flux solutions in its rows (36 scenarios × 100 samples) and the corresponding 45 flux values and the protein production for each scenario in its columns.

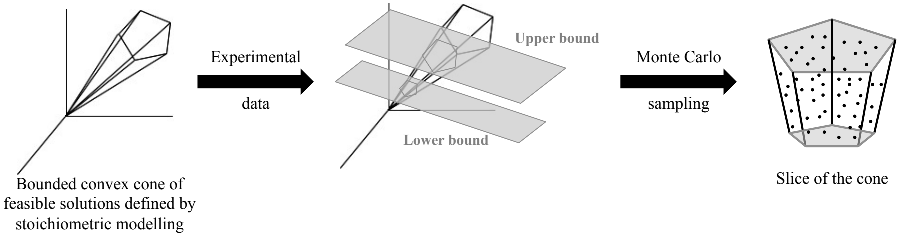

Bounded convex cone of feasible solutions defined by stoichiometric modelling
Experimental
data
Upper bound
Lower bound
Monte Carlo
sampling
Slice of the cone

Figure 5.4: MC sampling. The convex cone is obtained by Equations 3.1-3.4, the experimental measurements constrain the cone, and the sampling is performed on the resulting slice of the cone.

5.4 Multivariate modelling

5.4.1 PCA with MEDA

PCA is applied to the feasible flux solutions matrix $\mathbf{X}$ to obtain a small number of principal components (PCs) explaining a high percentage of variance of the complete data set. The idea is that the loading matrix, $\mathbf{P}$, retains information about these flux relationships; and the scores matrix, $\mathbf{T}$, describes to what extent this combination of fluxes is related to a particular observation or scenario.

After the PCA has been fitted, explaining 94.3% of total variance with five PCs, an outlier analysis is performed to the scores and the residuals of the different scenarios. The results on the statistic SPE show that scenario *C1*, classified on group glucose widely exceed the 99% control limits. Thus, the hundred observations generated for this scenario are knocked-out. Afterwards, a new PCA is fitted.

The results with the second analysis are that the first five PCs capture 95.9% of total variance in data: 42.4% for the first component, 24.2% for the second one, 19.7% for the third one, 7.0% for the fourth, and 2.5% for the last one. In these results no outliers are detected.

At this point, MEDA is applied using MEDA toolbox to enhance the interpretability of the PCs. In the MEDA method (see Section 2.3.2, the $\mathbf{Q}^2$ matrices have been built in a cumulative way, i.e. they have the variability of the first A PCs. These matrices can also be constructed by taking the information of a single PC. For this purpose, the method previously detailed has to be changed in Equations 2.10-2.13. The new equations have to consider only the a-th component for estimation. Finally, this kind of MEDA matrices, which have been used in this work, are written as $\mathbf{Q}^2_{(a)}$, where $a = 1, \ldots, A$.

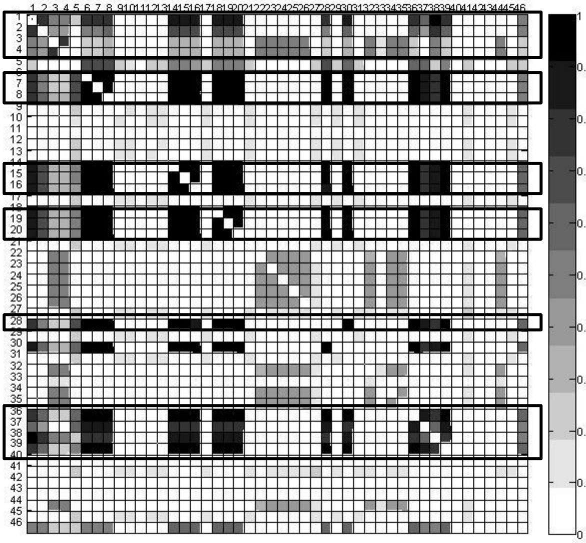


Figure 5.5: MEDA plot for the first PC. Solid line rectangles marks reactions related to this PC.

The MEDA is applied to the first five components obtained by the PCA, and the $\mathbf{Q}^2_{(a)}$, $a = 1, \ldots, 5$ matrices are obtained. Looking at the values in each matrix, the first three PCs are sufficient to explain the behaviour of the yeast, which capture 86,3% variance in data. Fourth and fifth PCs are classified as noise. The first three MEDA matrices can be seen in Figure 5.5-5.7.

If analysed from a biological standpoint, the first principal component relates protein production rate to reactions 5-8 (glycolysis), 14-16 and 18 (tricarboxylic nucelic acid (TCA) cycle), 19-20, 28, 30, 36 and 37. In Figure 5.5 these reactions are rounded by the solid line rectangle. It can be seen that these relations are indeed strongly correlated, having $Q^2_{(1),(k,l)}$ coefficients close to 1. As can be seen in Section 5.7, each of these groups is directly connected to NADH and ATP metabolism: ATP is formed in reactions 6, 8, 18 and 28, whereas NADH is formed in reactions 6, 14, 16 and 18-20. Finally, reactions 28, 30 and 36 represent the electronic transport chain, oxygen consumption and ATP dissimilation. The first PC can be then understood as the main pathway for ATP formation and dissimilation, this is, energy generation. Interestingly, protein productivity and ATP generation have been previously related in a first-principles based approach to predict recombinant protein production [150].

The second principal component is related to the biomass growth rate, which involves reactions 9-13 (fermentative pathways), 17, 21, 29 and 41 (relations shown by dashed line rectangles in Figure 5.6). Except for reaction 41, corresponding

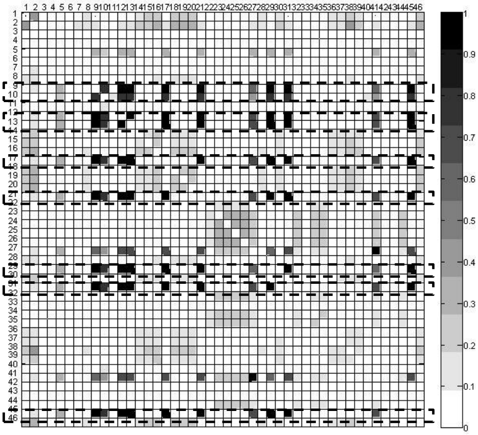


Figure 5.6: MEDA plot for the second PC. Dashed line rectangles marks reactions related to this PC.

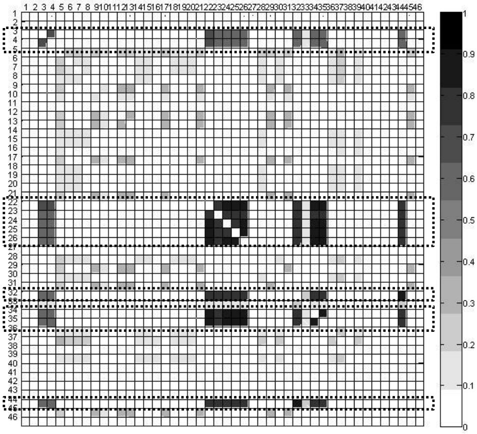


Figure 5.7: MEDA plot for the third PC. Solid line rectangles marks reactions related to this PC.

to the glycerol consumption rate, reactions 12 (around which reactions 9, 10, 11, 13 and 29 are connected), 17 and 21 share NADPH (either mitochondrial or cytosolic) production (see Section 5.7), which is, in fact, one of the major contributing precursors to biomass formation. It is worth noting that reaction 17 (corresponding to NADPH-requiring form) and not 16 (corresponding to the isoenzyme NADH-requiring) is identified.

Finally, the third principal component relates methanol consumption rate to the pentose phosphate pathway, strongly connected by reaction 34 (reactions correlated are rounded by dotted rectangles in Figure 5.7). Reactions 3-4, 22-26, 32, 35 and 44 are also related with this component.

The first three principal pathways are depicted in Figure 5.8. In this way, the reactions involved by the three first principal components seem to pinpoint specific metabolic indicators (cofactors NADH, NADPH and ATP) and their relation with protein, biomass and substrate (glycerol and methanol) consumption.

It is worth pointing out that the fit of a PCA model on the available experimental data is not feasible due to two main reasons: i) only seven out of nine external fluxes are measured for all scenarios under study, of which three have zero values mostly (see Figure 5.2), ii) the flux distributions across the metabolic network cannot be represented since no internal fluxes are considered. Actually, a PCA does not clearly relate substrates consumption to biomass and protein production, so this model is not meaningful (results not shown).

5.4.2 MCR-ALS

In this subsection, a soft modelling approach, MCR, is applied for the first time time to model flux data. Specifically, the ALS version of the algorithm is used. The reasons are its ability to provide physically more interpretable results by i) imposing some a priori knowledge through constraints on the modelling algorithm, and ii) avoiding the orthogonality restriction on the internal relationships between variables/pathways.

The idea behind MCR, traditionally applied in analytical chemistry, can be easily expanded to flux analysis by stating that a flux distribution across the metabolic network for a particular scenario is a linear combination of the different “true” pathways existing in it. This way, the spectra matrix **S** becomes the pathway matrix and **C** represents the relative contributions of pathways to each scenario.

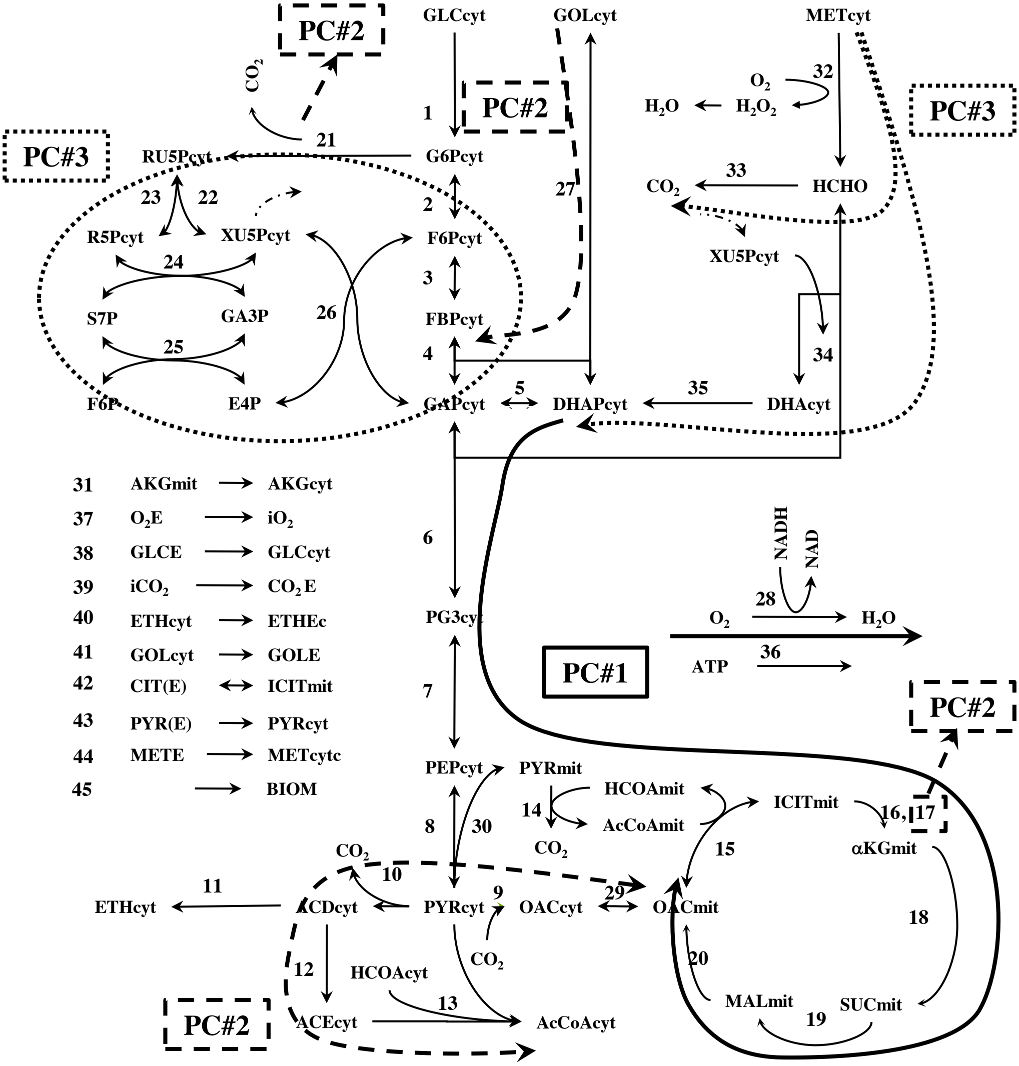


Figure 5.8: The first three PCs represent the main metabolic pathways through the yeast *P. pastoris*.

MCR-ALS: Data considerations

In order to exploit the potentiallity of the non-orthgonal components and the additional constraints in MCR, the reversible fluxes in the original dataset have been split into two irreversible ones, a usual strategy in stoichiometric modelling. Also, the positive flux values have been scaled to have values between 0 and 1. This way, when MCR model is applied in the next subsections, non-negativity and closure can be imposed on the contribution matrix.

MCR-ALS: Initial estimation

Since the MCR-ALS method is an iterative approach of MCR, it needs an initial estimation of either pathways or relative contributions matrix to start the ALS estimation of both matrices. Here, the pathways matrix is (initially) estimated using the most different scenarios in the dataset, i.e. the SIMPLISMA estimation implemented in MCR-ALS Toolbox.

MCR-ALS: solution as PCA-MEDA approach

MCR-ALS needs, unlike PCA, the number of components (or pathways) to be extracted before running the algorithm. Since our main objective is to compare the results of the PCA+MEDA approach and the results of MCR-ALS, it makes sense to start the MCR-ALS algorithm with three pathways, which is the number of PCs in the previous multivariate model. Additionally, the SVD estimation in MCR-ALS Toolbox of the number of components indicates that 3-4 components describe well the data set.

As explained above, different constraints can imposed in the MCR-ALS algorithm. The first constraint used here is that both the pathways and their relative contributions have to be positive. This is attained by the non-negativity constraint. Secondly, for each scenario, the relative contributions of pathways are forced to sum one, in order to represent a percentage of usage. This is the closure constraint, which is applied in the contributions direction.

The variance in data explained by the MCR-ALS model is 78.5%. The pathways obtained in this first approach are represented graphically in Figure 5.9. Each row represents the weights of the original variables in each pathway: the clearer is the corresponding square the higher is the weight. These pathways are represented on the metabolic network in Figure 5.10.

These results are somehow similar to the ones obtained applying PCA + MEDA: the first pathway is related to energy generation, in the form of ATP equivalents, mostly provided by glucose consumption through glycolysis and oxidative phosphorylation. The second pathway identified can be related to anabolism, and

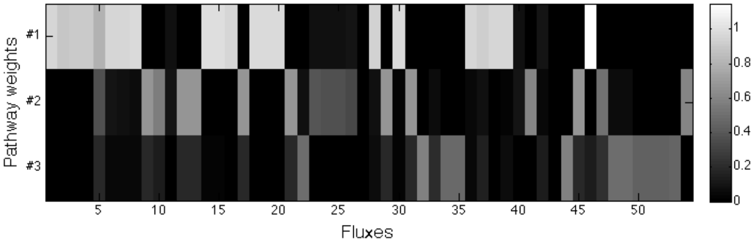


Figure 5.9: Pathways obtained extracting three components in the MCR-ALS method.

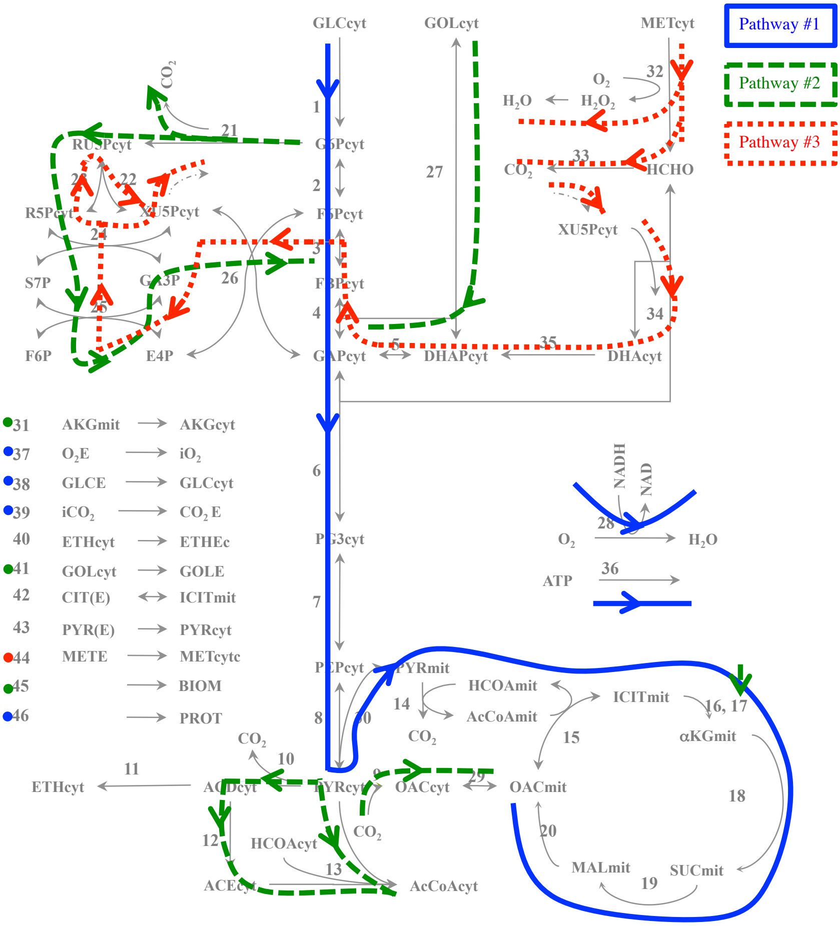


Figure 5.10: Metabolic network of P. pastoris with the three pathways obtained in the MCR-ALS method. The solid blue lines represent the first pathway, the dashed green lines the second one, and the dotted red lines the third one.

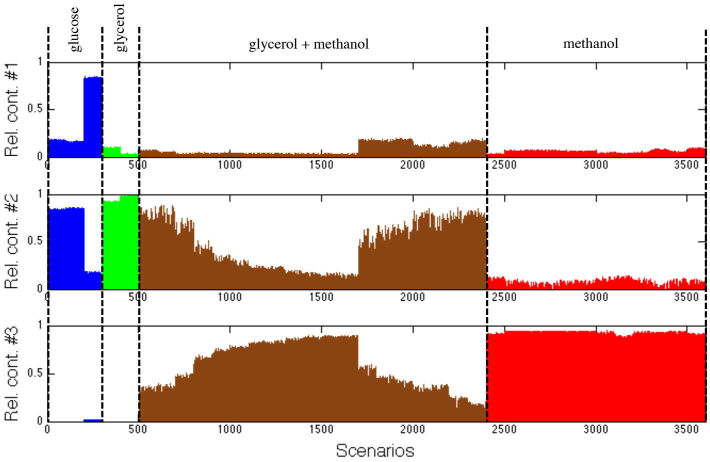


Figure 5.11: Relative contributions of the three pathways. The blue columns (scenarios 1-300) represent the percentage of usage of each pathway in glucose scenarios. The green columns (scenarios 301-500) represent the percentage of usage in glycerol scenarios. The brown ones (501-2400) are the scenarios with a glycerol-methanol mixture. The red columns (2401-3600) represent scenarios with only methanol as a substrate.

particularly to NADPH and AcCoA generation (thus indirectly to biomass growth) from glycerol. Finally, the third pathway seems to identify methanol consumption. Note that protein production is directly related to the first pathway as ATP is used as its single precursor in reaction 46. These pathways do not correspond exactly to the ones obtained in Section 5.4.1, especially in the case of the green (#2) and red (#3) pathways on the pentose phosphate route (reactions 21-26 in Figure 5.10), because the stoichiometric model is slightly different in the MCR approach (i.e. the reversible reactions are not split into two irreversible ones).

MCR-ALS can be exploited to study the relationship between each scenario and the pathways obtained. This relationship is depicted in Figure 5.11. This figure shows three plots, the first one represents the percentage of usage of the first pathway in each of the 3600 scenarios. As well, the other two plots represent the percentage of usage of the second and third pathways, respectively. The first pathway is surprisingly not strongly associated to some scenarios (1-200) in which glucose is the only carbon source. In an analogous way, the third pathway is used nearly at 100% in scenarios in which methanol is consumed. The second pathway is contributing both to scenarios in which only glucose or glycerol are used as a substrate, despite the fact that (as shown in Figure 5.10) this pathway does not consume glucose.

Once the relative contributions and the pathways have been visualised (Figures 5.9-5.11 some comments can be drawn. The second pathway depicted in Figure

5.10 does not flow in a thermodynamically feasible way through the metabolic network. The dashed green line crosses the pentose phosphate zone (reactions 22-26) and reaches reactions 3-4, where the glycerol consumption (reaction 27) ends in the opposite direction. This result, in addition with the poor contribution of the first pathway (solid blue) to the first two scenarios with glucose (1-200 in Figure 5.11), and the percentage of usage of the second pathway in glucose scenarios, indicates that the current model does not fully comprehend the behaviour of the scenarios analysed.

MCR-ALS: solution with four pathways

The results shown previously lead us to think that the actual MCR-ALS model may be improved by extracting another pathway, in order to discover if some of the pathways can be refined or if there is another hidden pattern in the data that is not explained at the moment. So a new model with four pathways is fitted. The model explains 82.4% of variance in data. The pathways obtained in this model are directly represented onto the metabolic network in Figure 5.12. The current first, third and fourth pathways are similar to the ones obtained in the previous MCR-ALS model (3 pathways). However, the second pathway represents a new metabolic route across the network.

The relative contribution of each pathway is plotted in Figure 5.13. Again, there is a plot for each pathway extracted from data. The first and second pathways seem to be associated mainly to glucose scenarios. The third pathway is widely used in the glycerol and glycerol+methanol scenarios, being the highest contribution attained in scenarios where glycerol is used as single carbon source. Finally, scenarios with only methanol use nearly at 100% the fourth pathway, and so do mixed scenarios with higher amount of this substrate.

As explained above, the flexibility of MCR-ALS method allows including different kind of constraints during the optimisation process. One of the most used constraints is selectivity. In this context, selectivity allows to constrain each pathway to not be used or "expressed" in all scenarios. By visual inspection of Figure 5.13 it seems that the first two pathways are related mainly to the glucose scenarios, the third one to glycerol and glycerol+methanol ones, and the last one to glycerol+methanol and methanol. This hypothesis is supported by the fact that *P. pastoris* cannot consume a substrate that is not present initially in the culture, so it makes sense to avoid this unrealistic metabolic behaviour through the statistical modelling.

The percentage of variance explained by including the selectivity constraint in the MCR-ALS model is 81.6%, which is only slightly lower than the percentage explained without this constraint (an admissible loss of explained variance). The variances explained by each pathway are: 11.8% (1st pathway), 9.6% (2nd path-

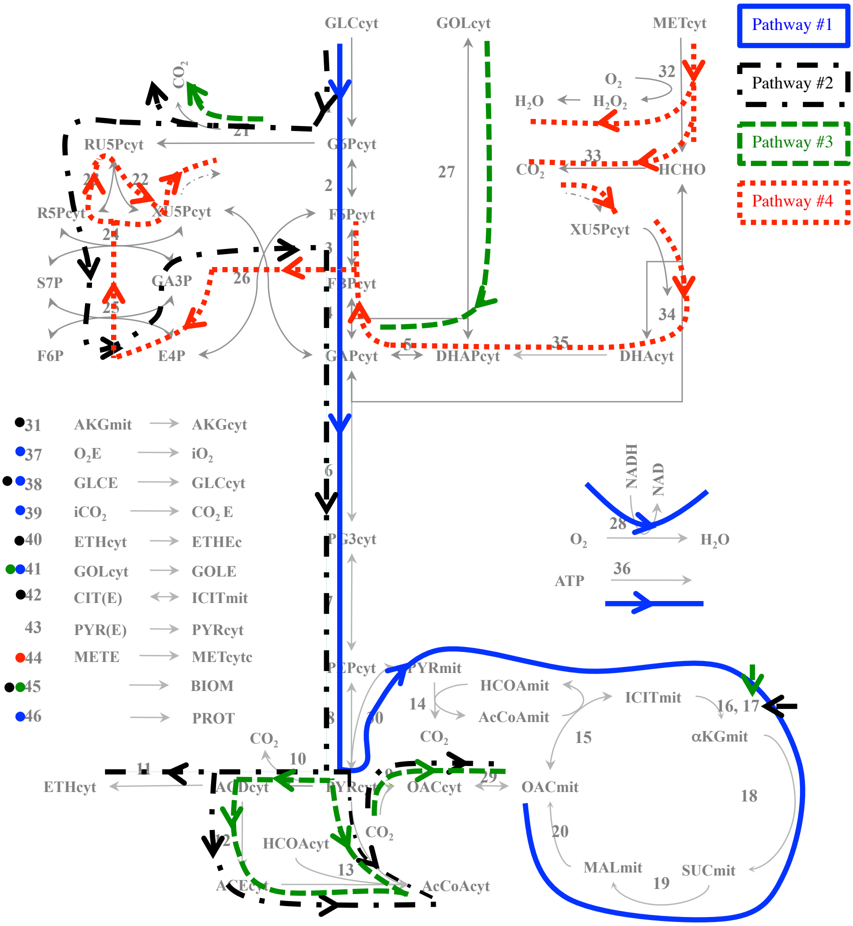


Figure 5.12: Metabolic network with four pathways. The solid blue lines represent the first pathway, the dash-dotted black lines the second pathway, the dashed green lines the third one, and the dotted red lines the fourth one.

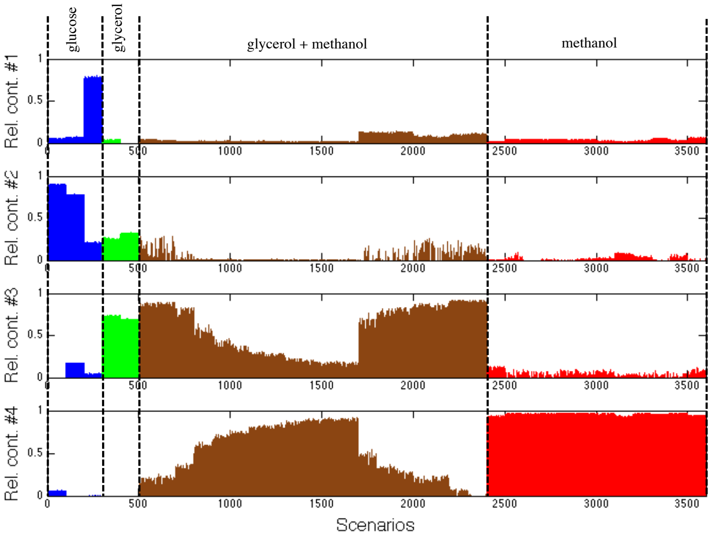


Figure 5.13: Relative contributions of the four pathways. More details in Figure 5.11.

way), 26.8% (3rd one) and 39.3% (4th one). The sum is 87.5%. Since the variance explained by the MCR model with 4 components using selectivity is 81.6%, the pathways have a degree of orthogonality of 93.2%.

The relative contributions of the pathways extracted with this model are plotted in Figure 5.14. The pathways obtained with this extra constraint in the model are basically the same as the ones represented in Figure 5.12 (results not shown).

Nevertheless, the inclusion of the selectivity constraint on the model produces a more clear usage of each pathway. In this way, the first two pathways explain the glucose scenarios, and the third and fourth pathways explain the glycerol and methanol ones, respectively, including their mixtures.

5.5 Discussion

PCA+MEDA and MCR-ALS models of *P. pastoris* deserve some discussion here. The final model of MCR-ALS includes all 36 possible experimental scenarios, while in the PCA method scenario C1 (sampled scenarios 101-200) were discarded. The reason is that this scenario, in fact the hundred simulated ones, widely exceeds the 99% control limit for the SPE. However, this scenario is clearly described in MCR-ALS by the first and second pathways. Moreover, the second pathway, which is describing scenario C1 up to 90% (Figure 5.14), consumes glucose and produces biomass. This pathway is not described by the PCA model, since biomass is only

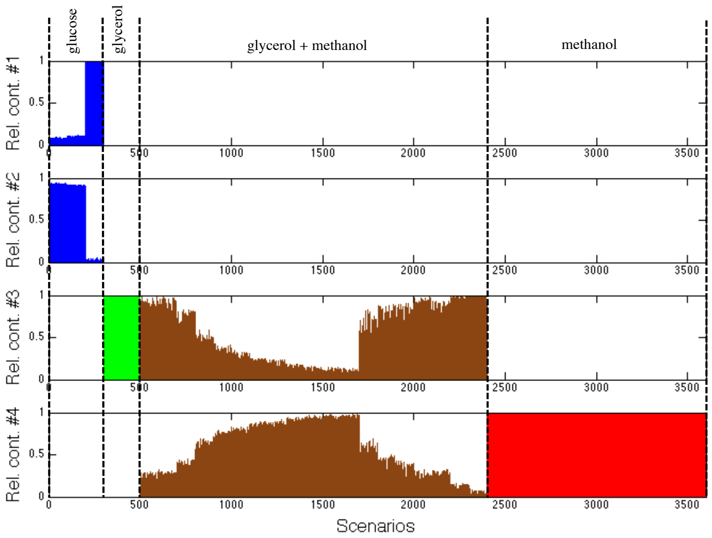


Figure 5.14: Relative contributions of the four pathways, including selectivity constraint. More details in Figure 5.11.

associated to glycerol consumption, while glucose consumption is only associated to TCA cycle, ATP and protein production. PCA associates a source of variability to a single PC, so biomass cannot be explained by two orthogonal components. However, it is obviously possible for the microorganism to grow using glucose as the only carbon source, as can be seen in Figure 5.3 (μ values of scenarios A1 and C1). Actually, this is highly desirable as the biomass yield on this substrate is the highest. This situation illustrates the main advantage of using MCR-ALS: a source of variability can be associated to more than one pathway – in the present case, biomass growth, which appears in the second pathway (associated to glucose consumption) and the third one (associated to glycerol consumption) –. This is also related to the degree of orthogonality of the MCR-ALS pathways. They are highly orthogonal (and that is the reason why some of its pathways are similar to the PCA ones), but without imposing this constraint a new biologically meaningful metabolic route (pathway 2) can be isolated.

The ability to include constraints during the optimisation is an advantage of MCR-ALS over PCA. Different types of biological knowledge can be included in a multicomponent model by using MCR-ALS. In the present case, non-negativity and closure are very useful in order to clearly identify and associate pathways to scenarios, while selectivity permits to avoid inconsistent behaviours related to known experimental conditions. The closure constraint allows us to explain the percentage of usage of each pathway in each scenario, but the total amount of flux flowing through a pathway cannot be compared between scenarios. This represents a disadvantage of the MCR-ALS model over a classical PCA, in which the more related is a scenario with a pathway the more flux is flowing through it.

5.6 Conclusions

Investigate the metabolic phenomena occurring within microorganisms is mandatory to really understand their observed behaviour. The knowledge derived from these studies is also relevant for biotechnological industries, which exploit these microbial cultures to produce top quality biochemicals. In this Chapter, the use of a grey modelling approach combining a first principles-based model with experimental information, followed by multivariate statistical techniques, such as PCA+MEDA or MCR-ALS, provides an insight on the main metabolic relationships underlying on actual *P. pastoris* cultures. In this way, the new approach presented here relates experimental substrates, metabolic pathways and biological functions of the yeast.

Both statistical modellings presented here have advantages and disadvantages. However, the flexible modelling of MCR-ALS, which permits to include many sources of biological knowledge in the model, opens a new framework of collaboration between statistical and biological modellers. This framework, which can be considered as a two-step grey modelling (first step: experimental data + constraint-based model, second step: statistical models + additional biological knowledge) can lead to a better understanding of these complex systems, and thus allows us to constrain the models into the desired direction and exploit all the available knowledge – first-principles, experimental data, etc. – in a suitable way.

5.7 Appendix. Metabolic model.

Metabolite abbreviations

Abbreviation	**Metabolite**
BIOM	Biomass (E)
Cit	Citric Acid (E)
CO2	Carbon dioxide (E)
EtH	Ethanol (E)
GLU	Glucose (E)
GOL	Glycerol (E)
Met	Methanol (E)
Pyr	Pyruvic acid (E)
O2	Oxygen (E)
ACCOAcyt	Acetyl coenzyme A

ACCOAmit	Acetyl coenzyme A (mitochondrial)
ACDcyt	Acetaldehyde
ACEcyt	Acetate
AKGcyt	2-Amino-6-ketopimelate
AKGmit	2-Amino-6-ketopimelate (mitochondrial)
DHAcyt	Dihydroxyacetone
DHAPcyt	Dihydroxyacetone phosphate
E4Pcyt	Erythrose–4-phosphate
EtOH cyt	Ethanol
F6Pcyt	Fructose-6-phosphate
FBPcyt	Fructose 1,6-biphosphate
G6Pcyt	Glucose-6-phosphate
GAPcyt	D-glyceraldehyde 3-phosphate
GLCcyt	Glucose
GOLcyt	Glycerol
HCHOcyt	Formaldehyde
ICITmit	Isocitric acid (mitochondrial)
iCO2	Carbon dioxide
iO2	Oxygen
MALmit	Malate (mitochondrial)
MeOHcyt	Methanol
NADH	Nicotinamide adenine dinucleotide phosphate
NADPHcyt	Nicotinamide adenine dinucleotide phosphate
NADPHmit	NADPH (mitochondria)
OAAcyt	Oxaloacetate
OAAmit	Oxaloacetate (mitochondrial)
PEPcyt	Phosphoenolpyruvate
PG3cyt	3 Phosphoglycerate
PYRcyt	Pyruvate
PYRmit	Pyruvate (mitochondrial)
R5Pcyt	Ribose-5-phosphate
RU5Pcyt	Ribulose-5-phosphate
S7Pcyt	Sedoheptulose-7-phosphate
SUCmit	Succinate (mitochondrial)
XU5Pcyt	Xylulose-5-phosphate

List of reactions

GLCcyt → G6Pcyt
G6Pcyt ↔ F6Pcyt
F6Pcyt ↔ FBPcyt
FBPcyt ↔ DHAPcyt + GAPcyt
DHAPcyt ↔ GAPcyt
GAPcyt + NADcyt ↔ PG3cyt + NADHcyt
PG3cyt ↔ PEPcyt + H2O
PEPcyt ↔ PYRcyt
PYRcyt + iCO2 → OAAcyt
PYRcyt ↔ ACDcyt + iCO2
ACDcyt + NADHcyt → ETHcyt + NADcyt
ACDcyt + NADPcyt → ACEcyt + NADPHcyt
ACEcyt + HCOAcyt → ACCOAcyt
PYRmit + HCOAmit + NADmit → ACCOAmit + iCO2 + NADHmit
ACCOAmit + OAAmit ↔ ICITmit + HCOAmit
ICITmit + NADmit → AKGmit + iCO2 + NADHmit
ICITmit + NADPmit → AKGmit + iCO2 + NADPHmit
AKGmit + NADmit → SUCmit + iCO2 + NADHmit
SUCmit + NADmit → MALmit + NADHmit
MALmit + NADmit → OAAmit+ NADHmit
G6Pcyt + 2 NADPcyt → RU5Pcyt + iCO2 + 2 NADPHcyt
RU5Pcyt → XU5Pcyt
RU5Pcyt → R5Pcyt
R5Pcyt + XU5Pcyt → S7Pcyt + GAPcyt
S7Pcyt + GAPcyt → E4Pcyt + F6Pcyt
E4Pcyt + XU5Pcyt → F6Pcyt + GAPcyt
DHAPcyt + NADHcyt → GOLcyt + NADcyt
NADH + 0.5 iO2 → NAD
OAAcyt ↔ OAAmit
PYRcyt → PYRmit
AKGmit → AKGcyt
O2(E) → iO2
GLC(E) → GLCcyt
iCO2 → CO2(E)
ETHcyt → ETH(E)
GOL(E) → GOLcyt
CIT(E) ↔ ICITmit
PYR(E) → PYR cyt
MET(E) → METcyt
METcyt + 1/2 O2 → HCHOcyt + H2O
HCHOcyt + 2 NADcyt → 2 NADHcyt + iCO2
HCHOcyt + XU5Pcyt ↔ DHAcyt + GAPcyt

DHAcyt → DHAPcyt
0,0033 ACCOAcyt + 0,008 ACCOAmit + 0,0266 AKGcyt + 0,0146 E4Pcyt + 0,0363 F6Pcyt + 0,0165 PG3cyt + 0,0363 G6Pcyt + 0,0000003 GOLcyt + 0,000002 iO2 + 0,0242 OAAcyt + 0,00079 OAAmit + 0,0252 PEPcyt + 0,0294 PYRmit + 0,011 R5Pcyt + 0,199 NADPHcyt + 0,056 NADPHmit + 0,0626 NAD → 1 BIOM + 0,0127 iCO2 + 0,0626 NADH + 0,0033HCCOAcyt + 0,008 HCCOAmit + 0,199 NADPcyt + 0,056 NADPmit

Chapter 6

Projection to elementary modes

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

[7] Folch-Fortuny, A., Marques, R., Isidro, I., Oliveira, R. & Ferrer, A. Principal elementary mode analysis (PEMA). *Molecular BioSystems* **12**, 737-746 (2016). 2016 Hot Article.

6.1 Introduction

In this thesis, PCA and MCR have been applied to find the main pathways in a set of experimental cultures obtained from the literature (see Chapter 5) with the aim of i) identifying which parts of the metabolism retain the main variability in flux data and ii) relating them to the behaviour of the organism, e.g. substrates consumption and protein production. Both methods build the pathways based on the relationships between fluxes and, also, the a priori knowledge introduced in the model using constrains in MCR. However, no graphical information about the inner connections in the network is introduced in the model.

Here, a new method is proposed to improve the interpretability of the components extracted by PCA and MCR-ALS, using the topology of the network to obtain the biologically relevant pathways in the model. This method is called PEMA. Its main advantage, over the previous methods, is that instead of building artificial components based on the correlation structure of the data, the components are selected from the complete set of EMs of the metabolic network. The EMs are the simplest representations of pathways across a metabolic network. The PEMA algorithm is designed to identify the most relevant set of active EMs in flux data, using a strategy akin to PCA in dimensionality reduction. The PEMA toolbox is freely available for non-commercial purposes on `http://mseg.webs.upv.es`, under GNU license.

The PEMA algorithm is quite different from previously proposed approaches using EMs. On the one hand, since PEMA is considering the whole set of EMs, instead of only the EPs, the flux data can be interpreted with fewer pathways than using the α spectrum [145]. On the other hand, PEMA finds the common set of active EMs in several flux distributions, instead of the active ones per flux distribution using optimization procedures [146, 147], thus reducing substantially the number of pathways needed to explain a complete flux data set.

This chapter is organised as follows. In the next section, the PEMA model is described in detail. Section 6.3 presents the results using simulated and actual data from *E. coli* and *P. pastoris.* The results are discussed in Section 6.4. Finally, some conclusions are drawn on Section 6.5.

6.2 Principal elementary modes analysis

PEMA uses the set of EMs as the candidates for the PCs. Let $\mathbf{X}$ be a flux data set with N observations or experiments and K fluxes. The PEMA model equation is:

$$\mathbf{X} = \mathbf{\Lambda}\mathbf{P}^{\mathrm{T}} + \mathbf{F} \tag{6.1}$$

where $\mathbf{P}$ is the $K \times E$ principal elementary mode (PEM) matrix, formed by a subset of E EMs from the entire $\mathbf{EM}$ matrix; $\mathbf{\Lambda}$ is the $N \times E$ weightings matrix; and $\mathbf{F}$ is the $N \times K$ residual matrix. It is worth noting that the values in $\mathbf{\Lambda}$ are forced to be positive, since from a network-based point of view, each possible steady-state flux distribution can be expressed as a non-negative combination of EMs [139].

In PEMA algorithm, the PEMs are chosen from the complete set of EMs in a step-wise fashion. The weightings associated to the PEMs are obtained by solving Equation 6.1:

$$\hat{\mathbf{\Lambda}} = \mathbf{XP}(\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1} \tag{6.2}$$

Unlike the loadings in PCA, the PEMs are not orthonormal, so Equation 6.2 usually requires the computation of the pseudo-inverse of $\mathbf{P}^{\mathrm{T}}\mathbf{P}$.

The first step of PEMA consists of calculating the weightings for each EM. So, initially, $\mathbf{P}$ and $\mathbf{\Lambda}$ are column vectors. Then the explained variance by each EM is obtained as follows [205]:

$$EV = \frac{\parallel \mathbf{X} \parallel^2 - \parallel \mathbf{F} \parallel^2}{\parallel \mathbf{X} \parallel^2} 100\% \tag{6.3}$$

The EMs are sorted by EV, and the EM explaining most variance becomes the first PEM, with its associated $\mathbf{\Lambda}$ values. Afterwards, the variance explained jointly by the first PEM and each of the rest of EMs is calculated, and the pairs of EMs are sorted again by EV. The EM explaining more variance (jointly with the first PEM) becomes the second PEM, with their corresponding new $\mathbf{\Lambda}$ values. This procedure is iterated until reaching the maximum number of EMs. Since the weightings are recalculated for the 1st-ith PEMs when the $(i+1)$th PEM is computed, the amount of variance explained by the current set of PEMs is maximum.

When the PEMs are extracted step-wise, selecting the EMs explaining most variance at each step, the greedy solution is obtained. This is the usual procedure in PCA. The loadings are built in such a way that they explain as much variance in data as possible, and additionally, the resulting loadings are orthonormal. However, with PEMA, the EMs are not orthonormal (neither orthogonal). Therefore, the greedy solution may not be the best subset of EMs for explaining the data, since the choice of the first PEM influences the variance in data that the following PEMs could explain. Two tuning parameters are introduced in the algorithm to cope with the previous problem. The greedy selection of the EMs is improved using a relaxation parameter R. This parameter makes the algorithm considers the best R EMs for the current PEM, and based on the variance explained extracting

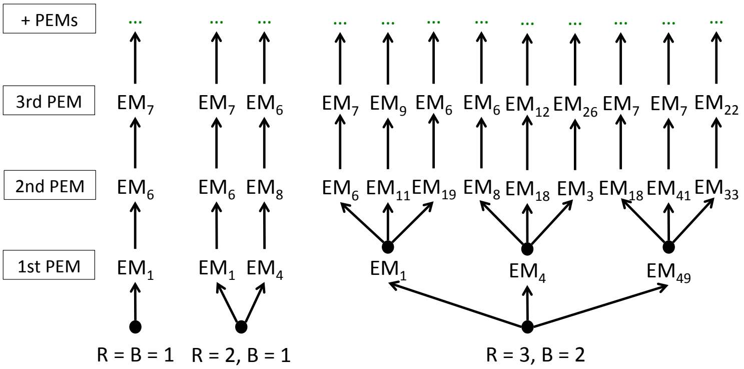

+ PEMs
3rd PEM
2nd PEM
1st PEM
...
EM7 EM7 EM6 EM7 EM9 EM6 EM6 EM12 EM26 EM7 EM7 EM22
EM6 EM6 EM8 EM6 EM11 EM19 EM8 EM18 EM3 EM18 EM41 EM33
EM1 EM1 EM4 EM1 EM4 EM49
R = B = 1
R = 2, B = 1
R = 3, B = 2

Figure 6.1: Relaxation (R) and branch point (B) parameters. When $B = R = 1$ the EM explaining more variance is chosen and fixed at each step. If these parameters change, different subsets are considered for each PEM identification.

more PEMs, the best EM from the set of R is selected. This relaxation step can be done for several consecutive selections of PEMs. The branch point number, B, marks up to which PEM the relaxed selection is performed.

Figure 6.1 shows an example of how the tuning parameters affect the selection of EMs. For instance, with $R = 3$ and $B = 2$, if one PEM is selected in the PEMA model it will be EM_1, since it is the EM explaining most variance; if two PEMs are selected it is possible that EM_1 and any of its 2nd PEM candidates (EM_6, EM_{11}, or EM_{19}) explain less variance that, for example, EM_4 and EM_8, so these last two will be the EMs selected in the PEMA model with two PEMs, and so on. The greedy approach accumulates the selected PEMs, but with $R > 1$ the EMs may change completely from one PEM to the next one, in order to explain more variance with a fixed number of PEMs.

The number of PEMs evaluations, i.e. the number of times that the algorithm solves Equation 6.1 for all EMs, can be calculated using R and B. Let A be the maximum number of PEMs to be extracted by PEMA. Then, the number of evaluations, O, has the following expression:

$$O = \sum_{i=0}^{B-1} R^i + (A - B) \cdot R^B \tag{6.4}$$

where O grows exponentially with the number of branch points B. This way, the computation time required for each possible pair (R, B) can be estimated using Equation 6.4 and the computation time of the greedy approach ($R = B = 1$ and $O_{greedy} = A$).

PEMA is an heuristic approach to solve the problem *which EMs do reconstruct the flux data?* The mathematical formulation of this problem consists of minimizing the 2-norm of $\mathbf{X} - \mathbf{\Lambda}\mathbf{P}^{\mathrm{T}}$ subject to $\mathbf{P} \subseteq \mathbf{EM}$. The problem with this formulation is that it represents a mixed integer nonlinear programming (MINLP) problem, and since the number of fluxes and EMs may be extremely high, it is justified the application of an heuristic algorithm to find a suboptimal solution to this problem. The proposed problem could be solved using genetic algorithms, however, different models have to be fit in order to get solutions with different number of PEMs. As well, the solution may change drastically depending on the initial points and the genetic operator chosen. This kind of algorithms improve an objective function, which can be the explained variance as in PEMA, but at some steps of the algorithm the search within the feasible space is performed in a random fashion, while PEMA focuses at each step in selecting the EMs explaining most variance. In this way, a single run of PEMA presents several solutions with a different number of PEMs.

6.2.1 Data preprocessing

If the original variables have strongly different means and/or variances when fitting PCA models, the PCs may focus on explaining only the variables with the highest values and/or variances, disregarding the small variance associated to the rest of variables. PEMA has the same problem as PCA, so the flux data has to be preprocessed. While in PCA it is relatively easy to scale and mean center the original flux data (see Chapter 5), in PEMA, since the EMs are fixed, this is a subtle issue. To maintain the biological meaning of the EMs, if **X** is scaled column-wise by their standard deviations, the **EM** matrix has to be modified scaling row-wise all the EMs by the same values. The scaling of the **X** and **EM** matrices gives, initially, equal importance to all fluxes in the data, since their variances are equal to 1. This preprocessing is recommended in flux data sets, since the variance of external fluxes can be exponentially greater than internal fluxes.

The mean centering of the PEMA model must not be done. When the data matrix **X** is mean centered, irreversible reactions would take negative fluxes thus the directionality of the fluxes is lost. In this way, if **X** is mean centered the PEMs are no longer able to fit the flux data. One way to overcome the mean centering problem is fitting additional PEMA models excluding the variables with the highest means. Once computed, the global and the local models can be compared in terms of EMs activation and reaction usage, to assess whether the global model is accounting for the fluxes with small values.

6.2.2 Algorithm

The PEMA algorithm consists of the following steps:

1. Scale column-wise the original flux data $\mathbf{X}$ by their standard deviations.
2. Scale row-wise the elementary modes matrix, **EM**, using the standard deviations of the original data set.
3. Choose the number of relaxations (R) and branch points (B).
4. Obtain the different PEMA models with 1 PEM, 2 PEMs, ..., A PEMs, solving Equation 6.1.
5. Select the number of EMs based on the aim of the study.
6. Recalculate the weightings $\mathbf{\Lambda}$ and the explained variance with the original flux data (without scaling).

Practitioners should start with the greedy approach ($R = B = 1$) and then, using the prediction of the computation time, select different configurations to compare the models. To span the different solutions that PEMA produces when changing the parameters, users are encouraged to follow the configurations presented in the next section (see also Table 6.1). For large datasets, e.g. genome-scale networks with millions of EMs, the computation of the greedy solution may take several hours. To avoid this long computation time, users can pre-select a subset of relevant EMs prior to applying PEMA.

Also, the number of PEMs selected in each model, as in PCA, depends on the aim of the study [19, 206]. In this way, the cumulative scree plot (see next section) may help to select the EMs explaining most variance in the flux data.

6.3 Case studies

6.3.1 *E. coli* simulated study

A simulated study is proposed here to validate the performance of PEMA. The study consists of simulating different flux data sets, using several subsets of EMs, in order to assess whether PEMA algorithm is capable of identifying them. The metabolic model (see Section 6.7 of *E. coli*, presented in [207], is used for this purpose (see Figure 6.2). The set of 255 EMs from the metabolic network of *E. coli* are obtained using EFMTOOL [165].

The simulated study is as follows: 100 different data sets are generated using from 1 to 10 EMs selected at random from the **EM** matrix. Ten different configurations

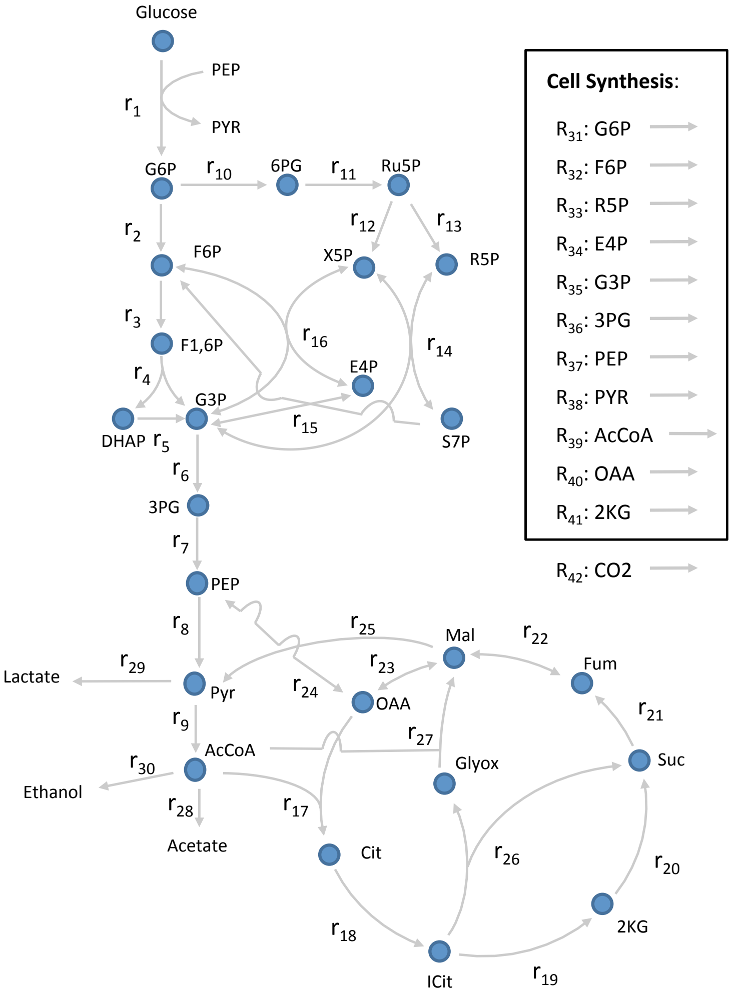


Figure 6.2: *E. coli* simulated study. Metabolic network considered in this chapter.

Table 6.1: Complete identifications of the generating elementary modes.

Configuration	Number of generating elementary modes						
(R_B)	1	2	3	4	5	6	7-10
1_1	10/10	7/10	2/10	2/10	0/10	0/10	0/10
5_1	10/10	10/10	5/10	3/10	1/10	1/10	0/10
10_1	10/10	10/10	5/10	4/10	1/10	0/10	0/10
20_1	10/10	10/10	5/10	5/10	1/10	0/10	0/10
2_2	10/10	9/10	5/10	4/10	1/10	0/10	0/10
5_2	10/10	10/10	5/10	2/10	1/10	0/10	0/10
10_2	10/10	10/10	7/10	7/10	2/10	1/10	0/10
3_3	10/10	9/10	7/10	6/10	4/10	1/10	0/10
5_3	10/10	10/10	7/10	8/10	5/10	1/10	0/10
4_4	10/10	10/10	7/10	8/10	6/10	3/10	0/10

of PEMA are applied on the present data, varying the values of the relaxations and branches $R-B$: $1-1$, $5-1$, $10-1$, $20-1$, $2-2$, $5-2$, $10-2$, $3-3$, $5-3$, $4-4$. The configurations are sorted approximately in increasing computation time.

The identifiability of each PEMA configuration can be assessed computing how many times the complete set of EMs that generated the simulated flux data is identified. This information is presented in Table 6.1. As expected, for a fixed value of B, the higher is R the better tends to be the solution. Also, the more branch points are considered the more sets of EMs tend to be completely identified.

Even though not all the EMs are identified when the number of generating ones increases, all PEMA configurations are able to detect a subset of them. The precision, P, and recall, R, are computed as:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \tag{6.5}$$

where TP are the true predicted EMs, FP the false positives, and FN the false negatives. The high precision implies that most of the EMs identified are true ones, and also the high recall implies that the method identified most of the original EMs.

Figure 6.3 shows the results of P and R of the EMs identifications. With the exception of the greedy approach, all PEMA configurations are able to identify 80-100% of the original 3-4 EMs. The most complex configurations, i.e. when $B = 3$ or $B = 4$, maintain this level of accuracy with 5-6 generating EMs.

It is also interesting to check the mean number of PEMs identified by the different configurations and the percentage of explained variance. Since there exists a high

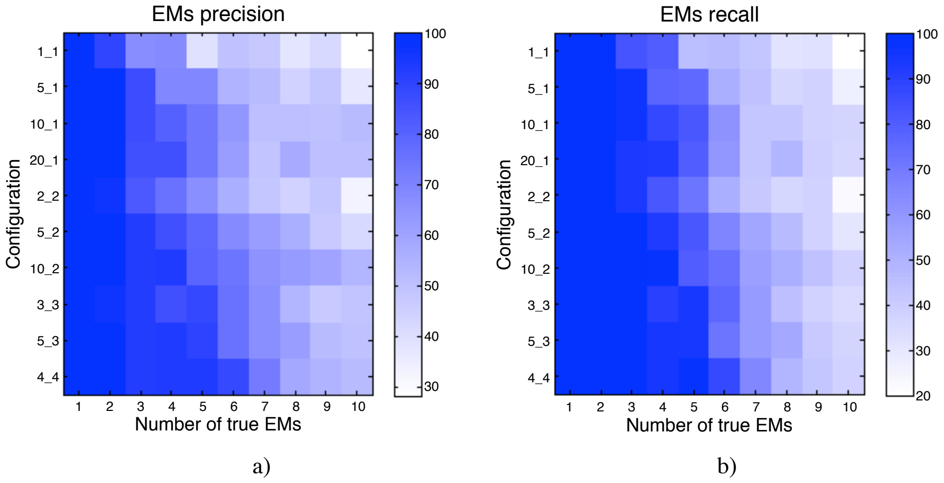


Figure 6.3: *E. coli* simulated study. Precision and recall of the different configurations. Precision is calculated by dividing the sum of the true identified EMs by the sum of the true identified plus the false identified ones. The recall is calculated by dividing the true identified EMs divided by the true ones plus the true non-identified ones.

degree of redundancy in any **EM** matrix, different linear combinations of EMs can represent a given flux distribution. This is clearly seen in Figure 6.4. Up to 5-6 generating EMs, the most complex PEMA configurations identify the same number of PEMs, matching the original ones (see Figure 6.4a). From 7 generating EMs onwards, the average number of PEMs grows slower, identifying between 7 and 8 PEMs on average with 10 generating EMs. However, the percentage of explained variance by these PEMs remains very high, more than 99% having 7-10 generating EMs (see Figure 6.4b). The reduction in the number of EMs might also be due to some of the randomly selected EMs, with a random weighting on the model, have a small contribution to the variance in comparison to the EMs with greater coefficients.

6.3.2 *E. coli* real data

The flux data of *E. coli* presented in [207] is used in this section to check the performance of PEMA with real data. Each observation in this dataset describes a flux distribution after a specifically targeted gene knock-out. The metabolic network and EMs set considered here are the same as in the simulated study (see Figure 6.2). The flux data matrix, **X** considered here has 21 observations (rows) and 42 fluxes (columns). In these 21 observations, a subset of the original 32 observations, the same set of reactions is considered.

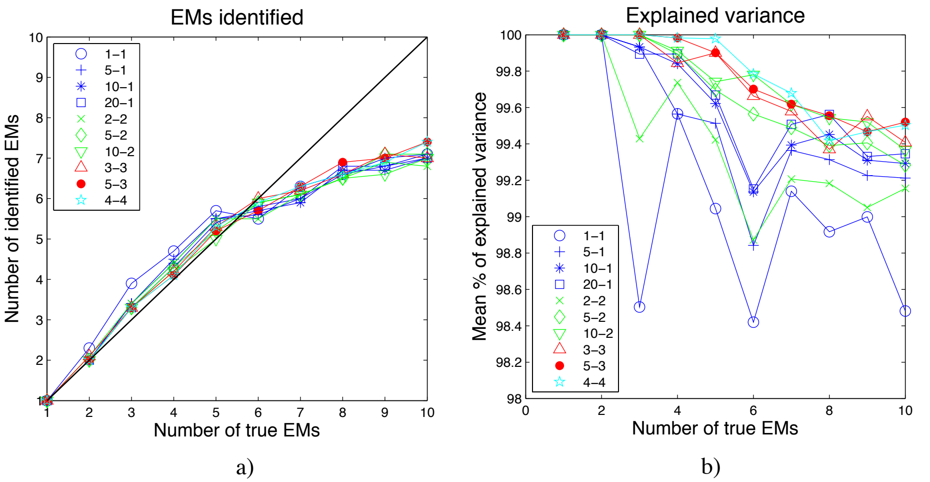


Figure 6.4: *E. coli* simulated study. a) Mean number of identified EMs. b) Mean percentage of explained variance.

Based on the results of the simulated study, the tuning parameters R and B are both set to 4, to obtain more accurate results. The computation time of PEMA in this case is 2 minutes, while the computation time of the greedy approach is less than a second. Figure 6.5a shows the cumulative scree plot of the PEMs. This kind of plot is usually employed in PCA to assess the appropriate number of PCs. Here, 8 PEMs are selected: EM_{125}, EM_{167}, EM_{254}, EM_{27}, EM_{235}, EM_{16}, EM_{143} and EM_{145}, explaining 97.8% of variance with the scaled data, and 99.4% of the real variance.

The PEMs selected can be visualized on the metabolic network in Section 6.6. As opposed to PCA, in PEMA the PEMs are usually explaining common sources of variability. This can be seen in Figure 6.5b, where the direct sum of all variances explained by the PEMs is 150%. For instance, EM_{125} explains more than 80% of variance in data, but this variance is shared with other PEMs. Nevertheless, the PEMs explaining most variance can be considered the most relevant in the model.

The degree of orthogonality of the PEMs can be obtained by dividing the variance explained by the model (99.4%) by the sum of the explained variances of each PEM. Here, the degree of orthogonality is 66.3%, way lower than MCR's in Chapter 5, implying that that the solution obtained by the PEMA is strongly non-orthogonal and, therefore, quite different from the PCA one.

To assess if some observation is not well modelled the percentage of explained variance per observation can be computed (see Figure 6.6a). Also the observed versus predicted plot can be used to visualise the differences at a datum level (see

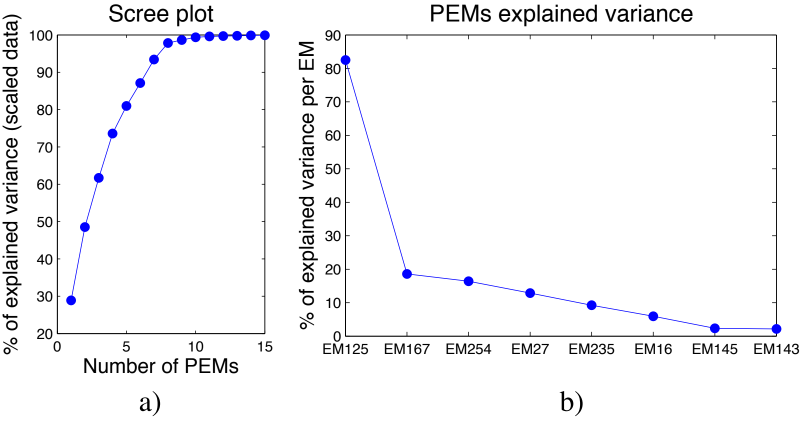


Figure 6.5: *E. coli* real case study. a) PEMA Cumulative scree plot and b) Percentage of variance explained by each PEM: 8 PEMs are selected explaining 97.4% of variance in the scaled data.

Figure 6.6b). In the present case, the percentage of explained variance is 97-99% for all observations, and the predicted values lay close to the true ones.

The PEMA model can be easily interpreted using an adaptation of the classical PCA loadings and scores plot. This way, Figures 6.7-6.8 shows the PEMs plot and the weightings plot, respectively. The PEMs plot shows which reactions are active for a specific EM, while the weightings plot represents the contribution weight of each PEM on each observation (i.e. knock-out). A first look at the selected PEMs shows that the whole set captures the formation of all metabolic requirements for cell synthesis, that is, reactions 31-41 (see Figure 6.7). EM_{125} is the PEM explaining most variance in data (see Figure 6.5b). This pathway depicts the glucose flux into glycolysis and TCA, without any exchange fluxes for cell synthesis metabolites. This leads to a high rate of NADH production, which generally is used to synthesize ATP. For this, EM_{125} can be interpreted as the cell's catabolic pathway, while the rest of PEMs capture the fluxes for cell synthesis metabolites, thus representing anabolic pathways leading to synthesis of biomass.

Since EM_{125} is related to the catabolism, it has a strong weight in each knock-out (see Figure 6.8). Nevertheless, some observations seem to have a greater impact in this PEM than others, in particular the knock-outs 2, 3, 10, 14, 15 and 16, representing the genes *glk*, *pgm*, *gpmB*, *rpiB*, *tktB* and *talB*. The *pgm* gene codifies the phosphoglucomutase that converts G6P into G1P and its deletion would likely direct the carbon flux to glycolysis or the pentose phosphate pathway. The *rpiB*, *tktB* and *talB*, also scoring a high weight, are related to pentose phosphate reactions.

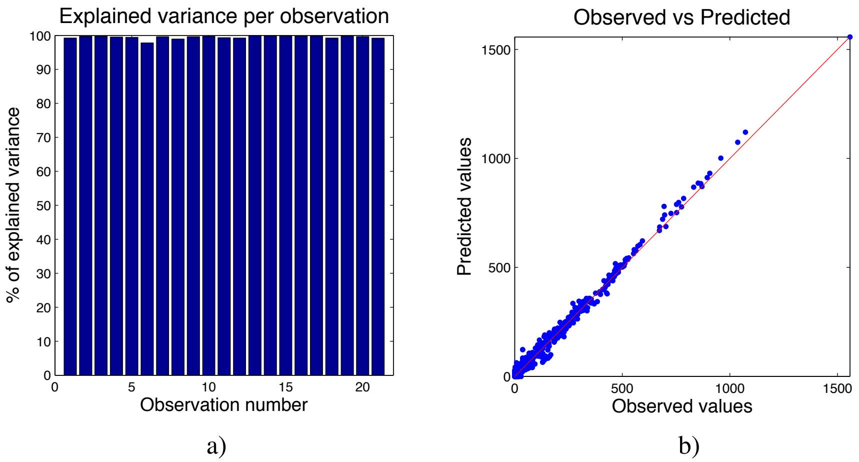


Figure 6.6: *E. coli* real case study. a) Explained variance per observation and b) Observed versus predicted plot.

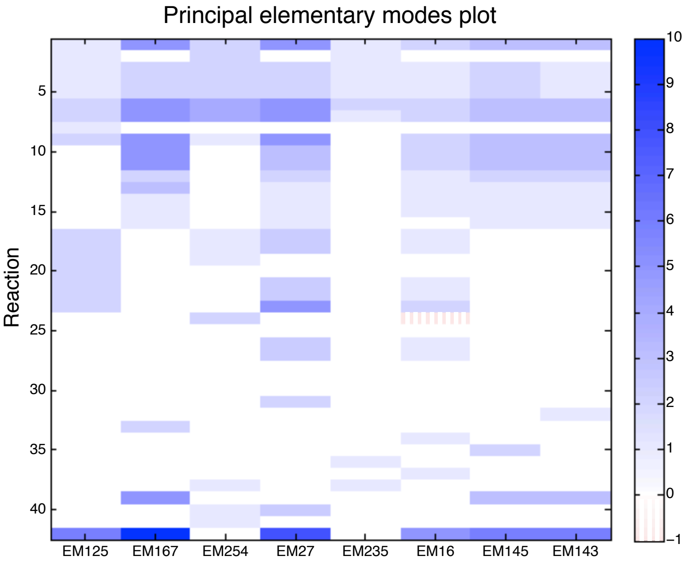


Figure 6.7: *E. coli* real case study. PEMs plot. The PEMs are represented by columns and the corresponding reactions by rows. Blue squares represent positive values, and dashed red ones the negatives. The darker the colour, the more highly positive/negative is the value.

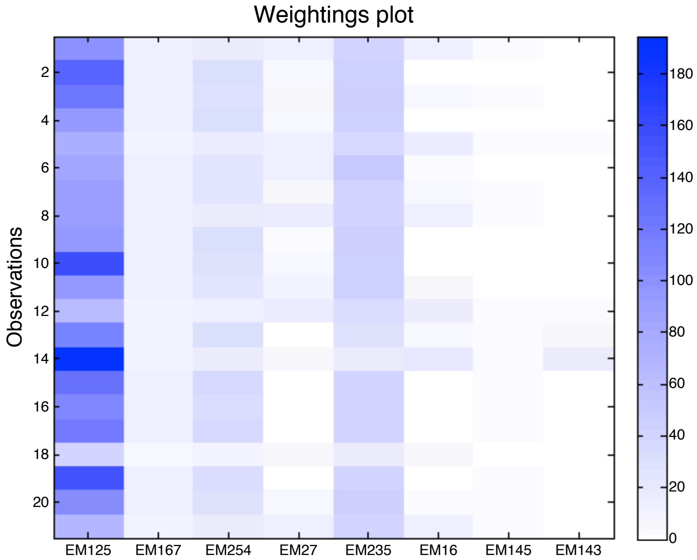


Figure 6.8: *E. coli* real case study. Weightings plot. The weightings of the PEMs are represented by columns and the observations by rows. The darker the colour, the more important is the PEM for the corresponding observation.

The EMs related to anabolic metabolism represent all the remaining exchange fluxes that produce the cell synthesis metabolites. EMs 16, 27, 143, 145 and 167 connect glucose directly to the pentose phosphate pathway, which is fundamental in the metabolism, since it generates NADPH, a reduced equivalent important in biosynthetic processes [208]. Moreover, EM_{16} and EM_{167} are responsible for balancing the metabolic fluxes towards E4P and R5P, being the sole PEMs that predict the fluxes of these metabolites to cell synthesis. With a few exceptions, the knock-out experiments have similar weight values inside each anabolic PEM. These exceptions are the observations 1, 5, 8, 12 and 14, representing the knockouts *galM*, *pfkB*, *gapC*, *pykF* and *rpiB*. This group of genes has low weightings in EM_{254} and EM_{235}, meaning that these flux modes have a minor impact in the metabolism of these mutants, that is, a lower flux in the synthesis of Pyr, 3-PG, 2-KG and OAA for biomass synthesis. Conversely higher weightings from these mutants are observed for EM_{27} and EM_{16}, that is, in the production of E4P, PEP and G6P. Another curious aspect of EM_{16} and EM_{27} is the activation of the glyoxylate bypass. This pathway is known to be active in low glucose concentrations [209], but repressed when glucose becomes available in higher concentrations [210, 211]. The observations 18 to 21 reflect *E. coli* wild-type cultured at a dilution rate of $0.2\mathrm{h}^{-1}$ used as control experiments. In these observations, positive fluxes for the gyoxylate pathway were registered, possibly due to a low glucose feed to the culture.

Finally, all the PEMs have a zero coefficient for fermentative pathways (reactions 28-30), therefore these fluxes are not being represented by the model. However, looking at the original data, all the observations have zero values for fluxes 28 and 29. Regarding flux 30, few observations (4 out of 21) have a non-zero value for it. For the latter case, since PEMA, as PCA, aims at explaining the covariance between the original variables using the PEMs, if most of the values in a variable are 0 it is difficult for PEMA to identify the EM generating these slight differences. The extraction of more PEMs may correct that, however, the risk of overfitting is higher and the model would become less parsimonious.

6.3.3 *Pichia pastoris* real data

A second real case study is analysed here: a fluxome for the growth of recombinant *P. pastoris*. This data set was based on a statistical design of experiments to test the effects of culture media factors in the flux data. The media composition was prepared according to the Invitrogen's guidelines for *P. pastoris* fermentation, and consists mainly on mineral salts. 26 shake flask experiments were performed with variations on 11 media factors selected for statistical design. Glycerol was used as carbon source in every experiment.

The metabolic network for the central carbon metabolism of *P. pastoris* used here is largely based on the network proposed in [132], with adaptations from other central carbon [212] and genome-scale networks [190]. The network consists of 43 metabolic reactions (see Section 6.7, 34 internal metabolites and 10 exchange reactions (see Figure 6.9). The main catabolic reactions are represented in this network, namely glycolysis and gluconeogenesis pathways, the TCA cycle, the pentose-phosphate pathway, anaplerotic, fermentative and phosphorylative oxidation pathways. A biomass formation reaction is also included in the model, from selected internal metabolites based on *P. pastoris* cells macromolecular compositon [190]. There exist 158 EMs in the metabolic model.

The results of PEMA with this data set are the same using either the greedy approach and the most complex approach presented here ($R = B = 4$), which takes 35 seconds. This indicates that the results are stable against the different PEMA configurations. 99.5% of the scaled data is explained using 3 PEMs, with a degree of orthogonality of 70% (i.e. the variance explained by the 3 PEMs sums 141%). As in the previous real case study, this implies that PCA cannot obtain these results using orthogonal components. The cumulative scree plot and the variance explained by each PEM are shown in Figure 6.10.

All scenarios are being represented by the selected EMs, as can be seen in the explained variance per observation plot (see Figure 6.11a); and the observed versus predicted plot (see Figure 6.11b) shows an even better fitting than with *E. coli*, which could be due to different levels of noise in the flux data sets.

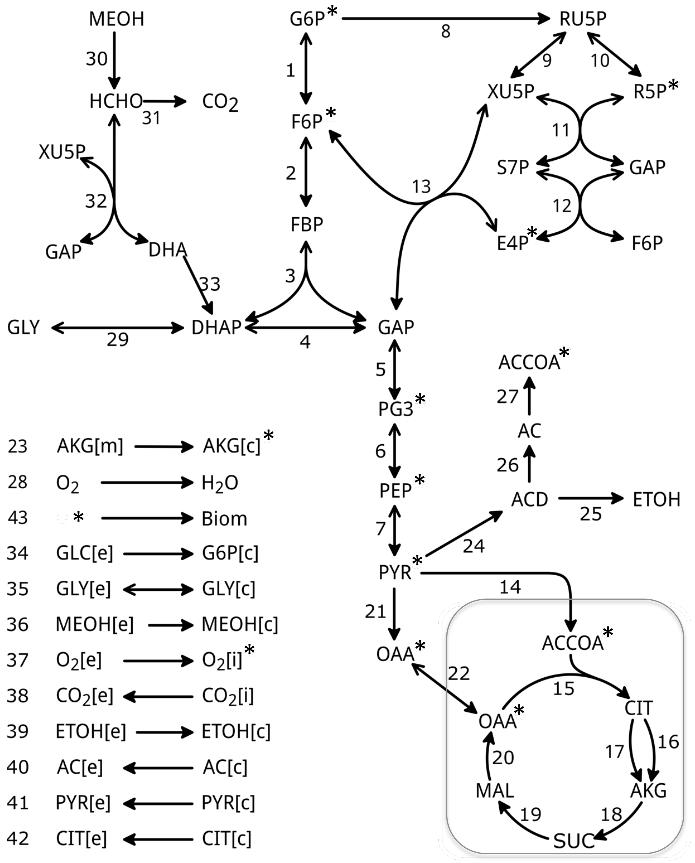


Figure 6.9: *P. pastoris* real case study. Metabolic network considered for the real case study.

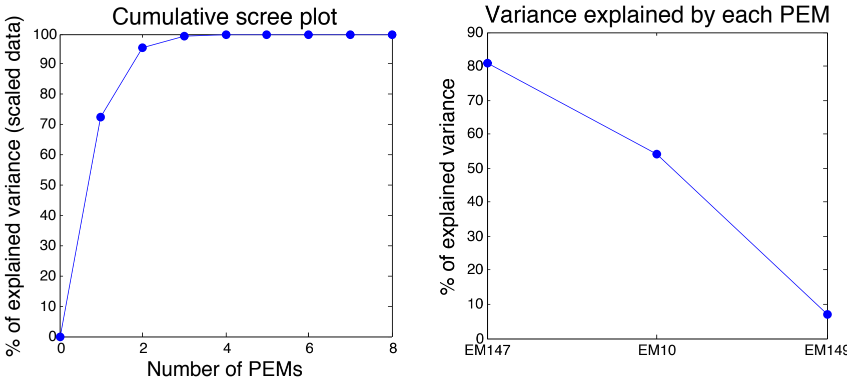


Figure 6.10: *P. pastoris* real case study. a) PEMA Cumulative scree plot and b) Percentage of variance explained by each PEM: 3 PEMs are selected explaining 99.5% of variance in the scaled data.

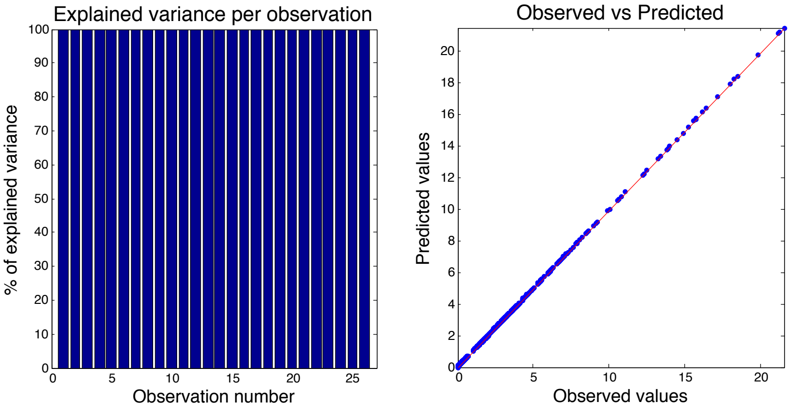


Figure 6.11: *P. pastoris* real case study. a) Explained variance per observation and b) Observed versus predicted plot.

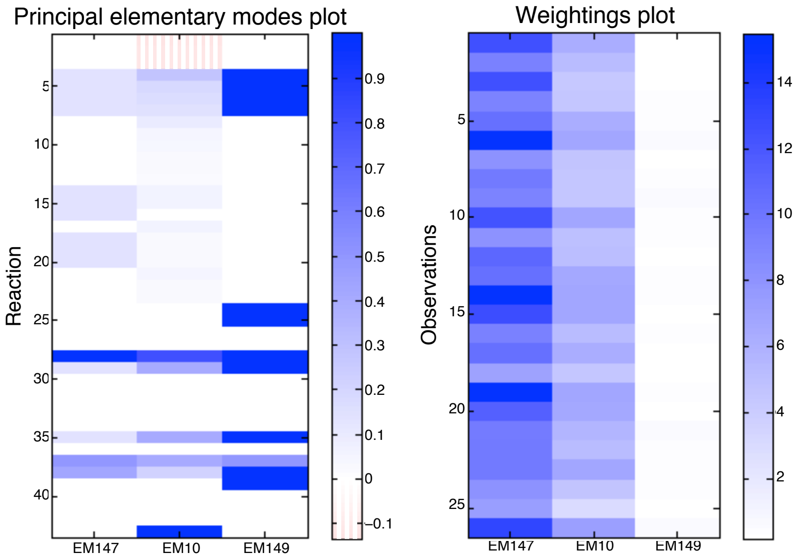


Figure 6.12: *P. pastoris* real case study. PEM and weightings plots. More details in Figures 6.7 and 6.8.

Figure 6.12 shows the PEMs and weightings plots. The PEMs identified are EM_{147}, EM_{10} and EM_{149}. They can be visualized on the metabolic network in Section 6.6.

The first PEM consumes glycerol (reactions 35 and 29) and crosses half of the glycolytic pathway (reactions 4-7) to activate the TCA cycle (reactions 15, 17-20), clearly representing the cell's catabolism. EM_{10} uses also reactions 35, 29 and 4-7 to activate the TCA cycle, but in this case reaction 16 is used instead of 17. It also activates the pentose phosphate pathway (reactions 8-13), leading to the synthesis of redox equivalents (NADPH), but also precursor metabolites for the synthesis of biomass. For this reason, this PEM groups the reactions for the cell's anabolism. At the end, this is the PEM responsible of the biomass production in all observations. The last PEM assimilates glycerol in the same way as EM_{147} and afterwards focuses on the production of ethanol (reactions 25 and 39). The occurrence of ethanol synthesis during aerobic respiration in yeast is a common feature (Crabtree effect). Nonetheless, unlike most yeasts, *P. pastoris* does not typically exhibit a significant ethanol production, favouring the aerobic metabolism. This fact is well captured by the relative lower explained variance of EM_{149} in comparison to EM_{147} (see Figures 6.10b and 6.12b).

Finally, as expected, no EM related to methanol assimilation (reactions 30-32 and 26) and final products, such as pyruvate or citrate (reactions 41 and 42, respectively), is selected, since all fluxes are 0 for these reactions.

6.4 Discussion

The simulated study on *E. coli* shows the high identifiability of PEMA. The most complex PEMA configurations are able to detect completely 1-4 generating EMs and, a high percentage of them, up to 6-7 EMs. Even though not all the EMs are identified by PEMA, the method provides always a parsimonious solution explaining more than 99% of variance. The analysis of actual flux data of the same organism confirms the tendency shown with the simulated fluxes. 8 PEMs are identified explaining 99.4% of variance in the flux data. This way, most of the PEMs identified are describing the glucose consumption, the glycolytic pathway and the TCA cycle, but afterwards, each of them has a different function in the cell synthesis. The results obtained with *P. pastoris* are coherent with *E. coli*'s. In this case 3 PEMs are selected describing accurately the metabolic pathways being activated when glycerol is used as main carbon source in aerobic conditions.

A significant number of graphical tools, all of them integrated in the PEMA toolbox, are provided in this chapter. The cumulative scree plot, the observed versus predicted plot, and the variance explained per observation plot can be used to decide the number of PEMs to extract. The plot showing the variance explained by each PEM and the PEMs and weightings plots are useful to exploit the PEMA model in terms of relevance and biological interpretation of the PEMs, and their activation among the observations.

Additionally, the theoretical estimation of the runs of PEMA algorithm when the tuning parameters change permits to establish a relatively accurate upper bound of the computation time, based on the greedy approach solution. This allows designing wisely a set of trials to compare the results of the different configurations of PEMA.

6.5 Conclusion

In this Chapter, a novel method, PEMA, is developed to explain the inherent variability on a fluxomics dataset, while preserving biological meaning. This can be regarded as an exploratory technique that allows researchers to interpret a data set by uncovering the most representative pathways operating in a cell.

There is a potential use of this methodology in bioprocess engineering applications, such as the development of structured metabolic models in cell culture fermentations. PEMA can be useful in the identification of a specific set of EMs that explains variations in cellular metabolic rates under certain operational conditions, such as temperature and pH. This would allow the improvement of the process kinetics' modelling by the incorporation of biological knowledge from the cellular system.

The PEMA toolbox is freely available for non-commercial purposes on `http://mseg.webs.upv.es`, under a GNU license.

6.6 Appendix A. PEMs.

E. coli

In this section, the PEMs identified for *E. coli* in the real case study are shown (see Figures 6.13-6.20).

P. pastoris

Here, the PEMs identified for *P. pastoris* in the real case study are shown (see Figures 6.21-6.23).

6.7 Appendix B. Metabolic models.

E. coli

Metabolite abbreviations

Abbreviation	**Metabolite**
Glucose	
G6P	Glucose-6-phosphate
F6P	Fructose-6-phosphate
F1,6P	Fructose 1,6-biphosphate
DHAP	Dihydroxyacetone phosphate
G3P	Glyceraldehydes-3-phosphate
3PG	3 Phosphoglycerate
PEP	Phosphoenolpyruvate
PYR	Pyruvate
AcCoA	Acetyl coenzyme A
CO2	Carbon dioxide
6PG	6 Phosphogluconolactonase

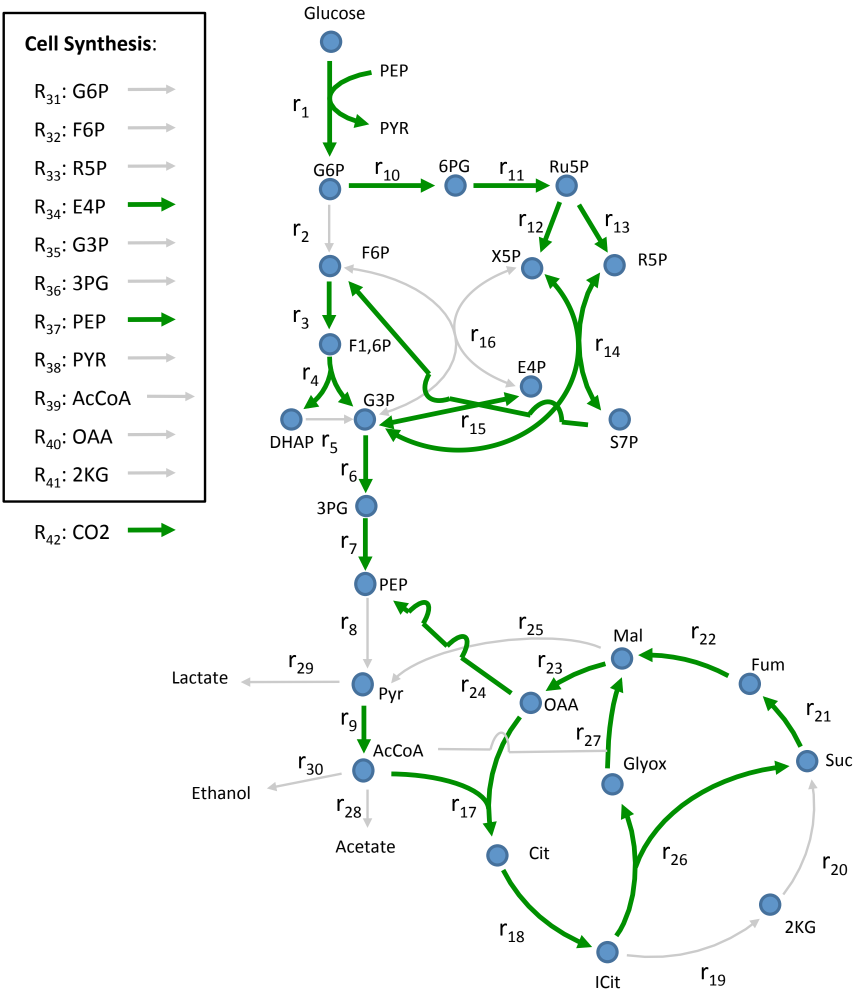


Figure 6.13: *E. coli* real case study. EM_{16} represented onto the metabolic network.

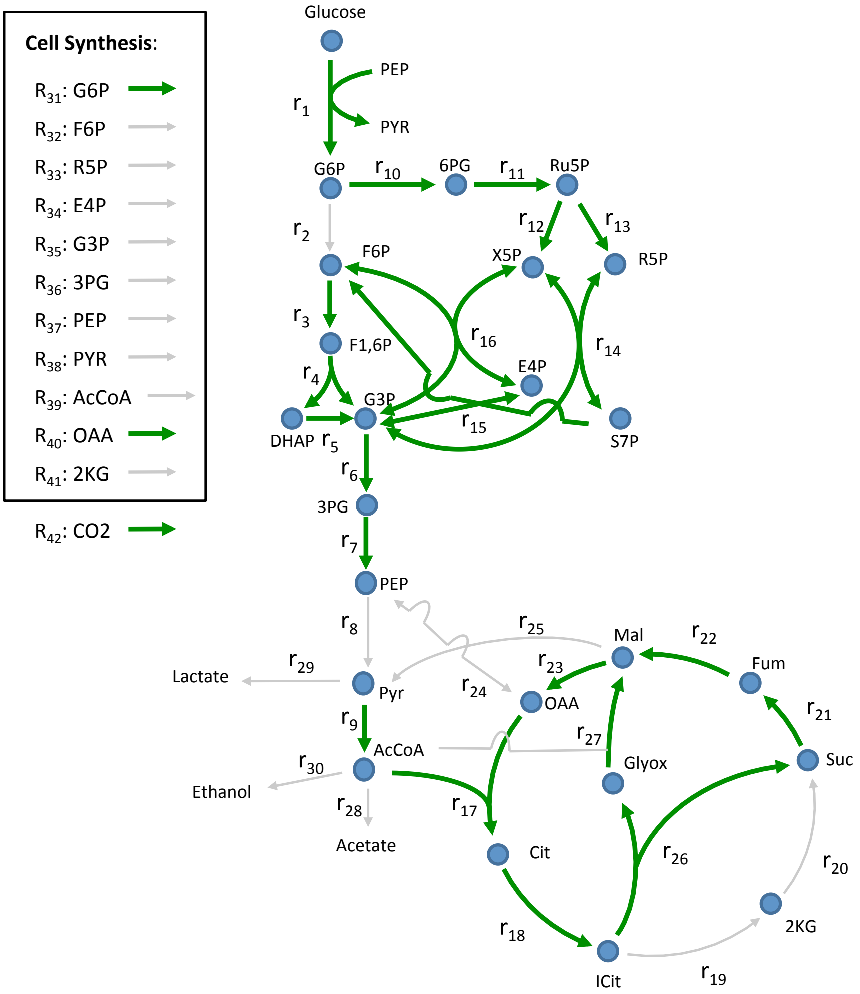


Figure 6.14: *E. coli* real case study. EM_{27} represented onto the metabolic network.

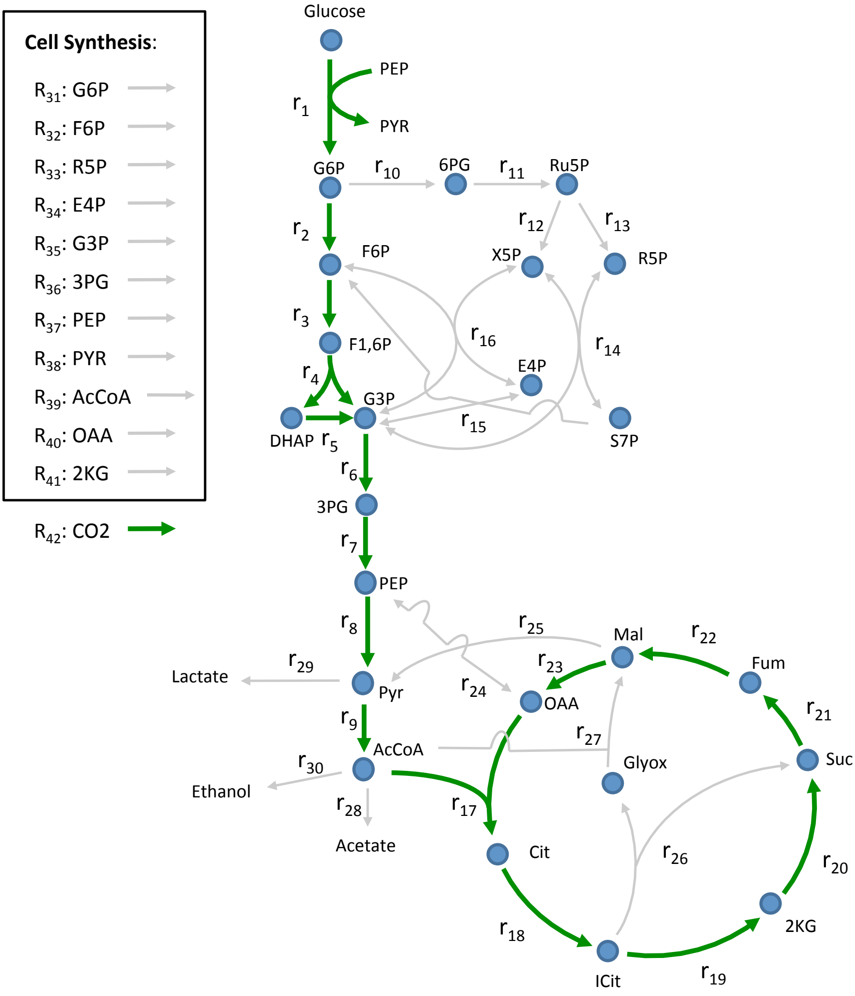


Figure 6.15: *E. coli* real case study. EM_{125} represented onto the metabolic network.

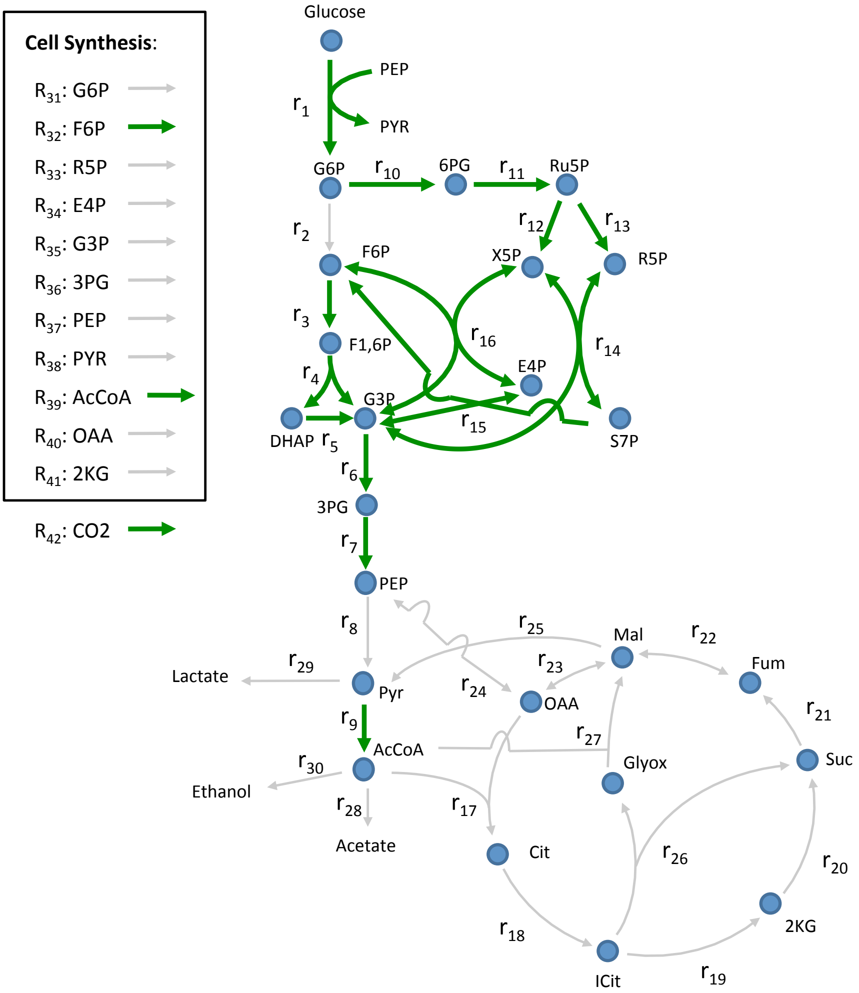


Figure 6.16: *E. coli* real case study. EM_{143} represented onto the metabolic network.

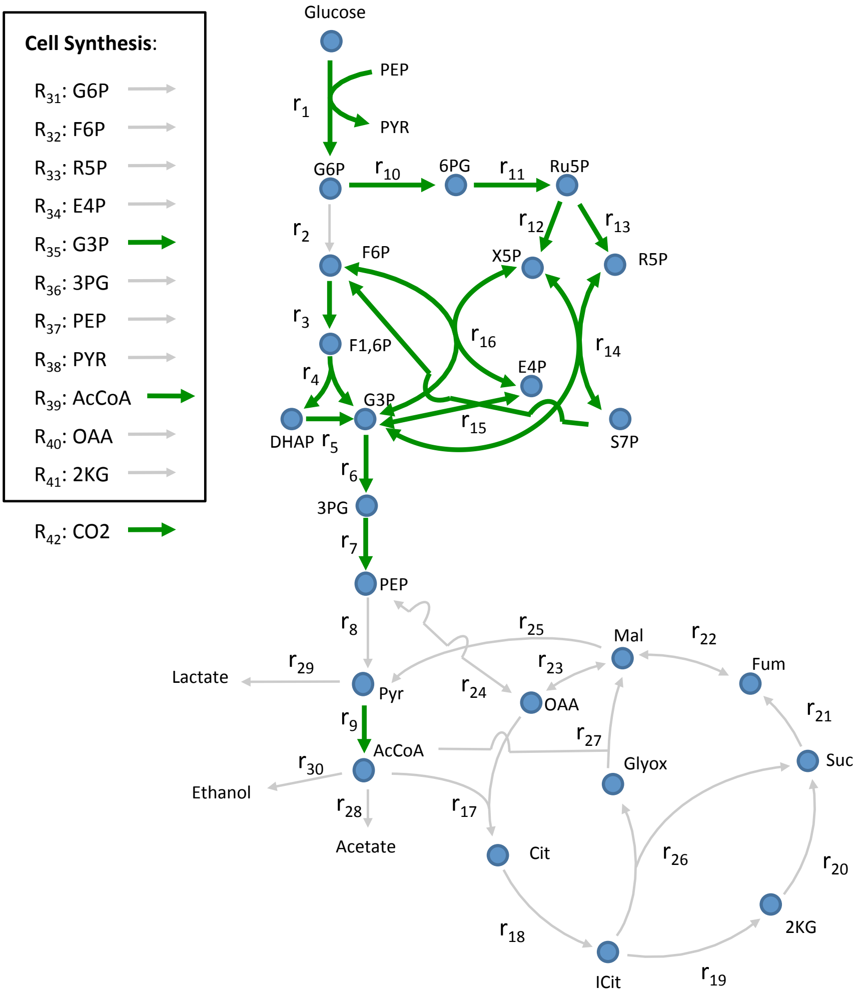


Figure 6.17: *E. coli* real case study. EM_{145} represented onto the metabolic network.

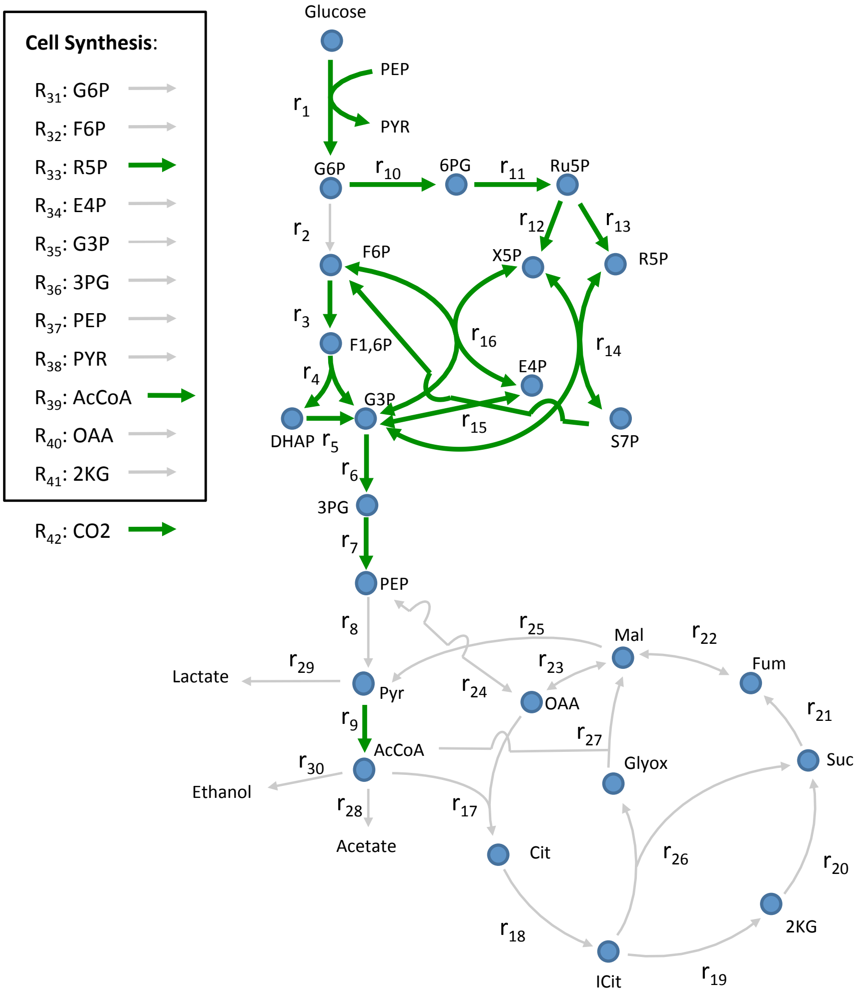


Figure 6.18: ***E. coli* real case study.** EM_{167} represented onto the metabolic network.

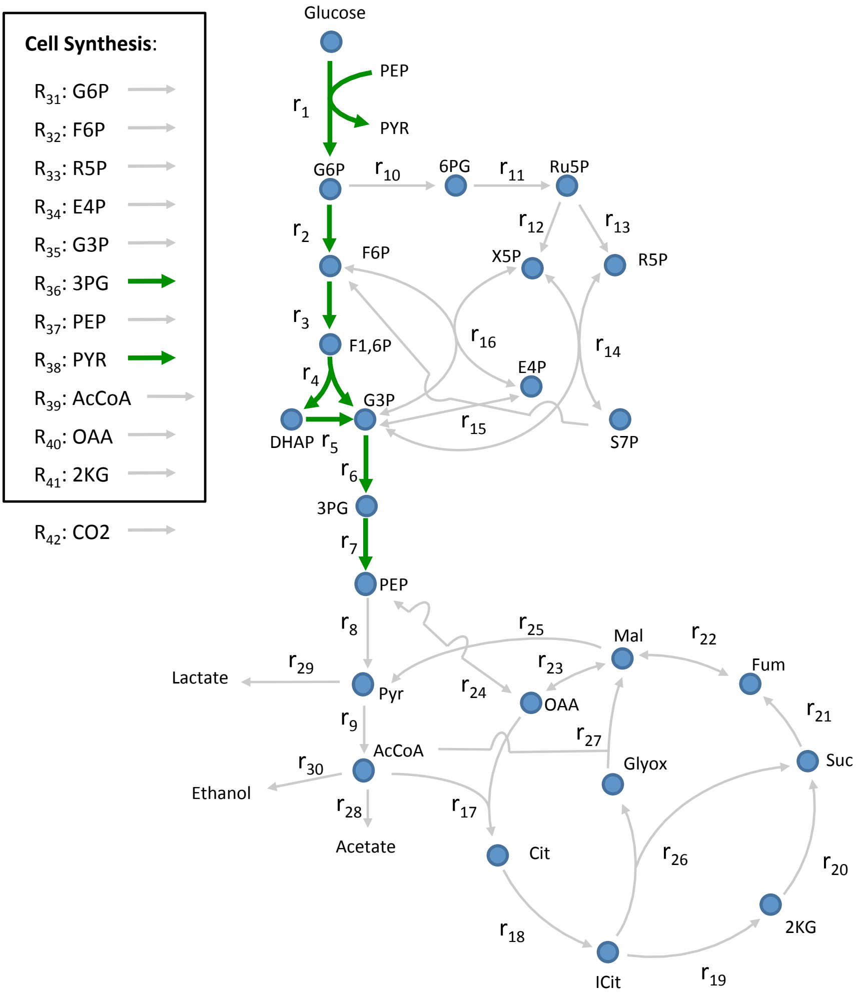


Figure 6.19: *E. coli* real case study. EM_{235} represented onto the metabolic network.

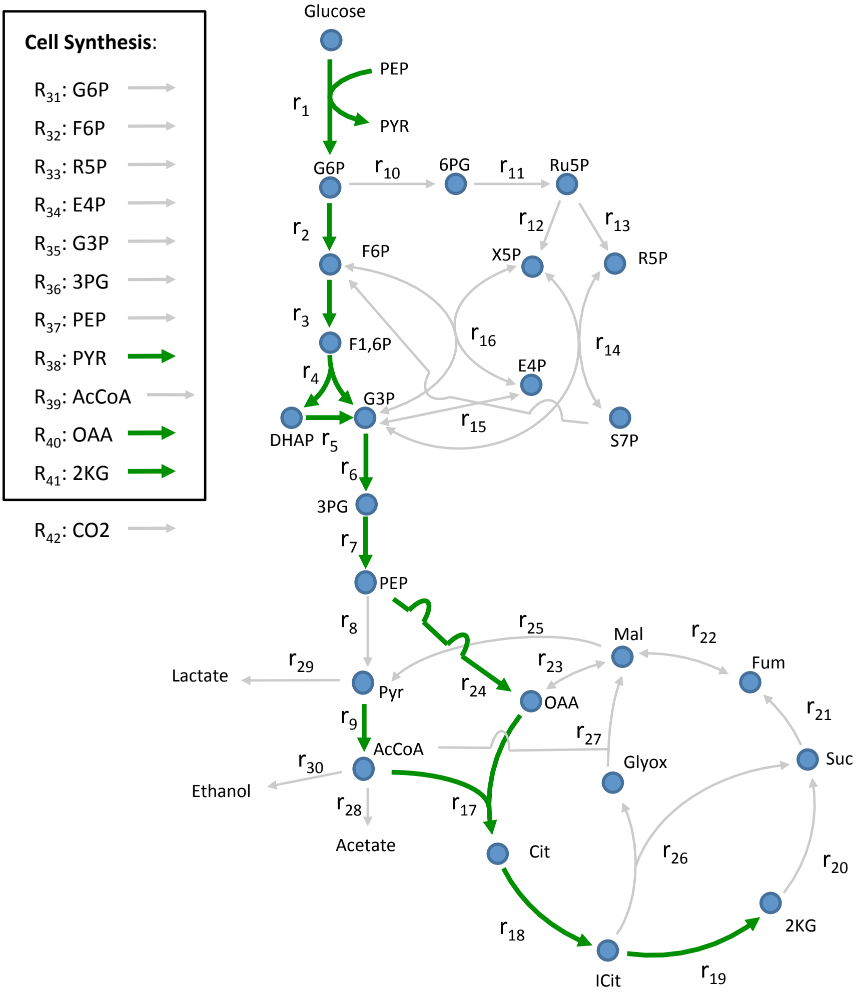


Figure 6.20: *E. coli* real case study. EM_{254} represented onto the metabolic network.

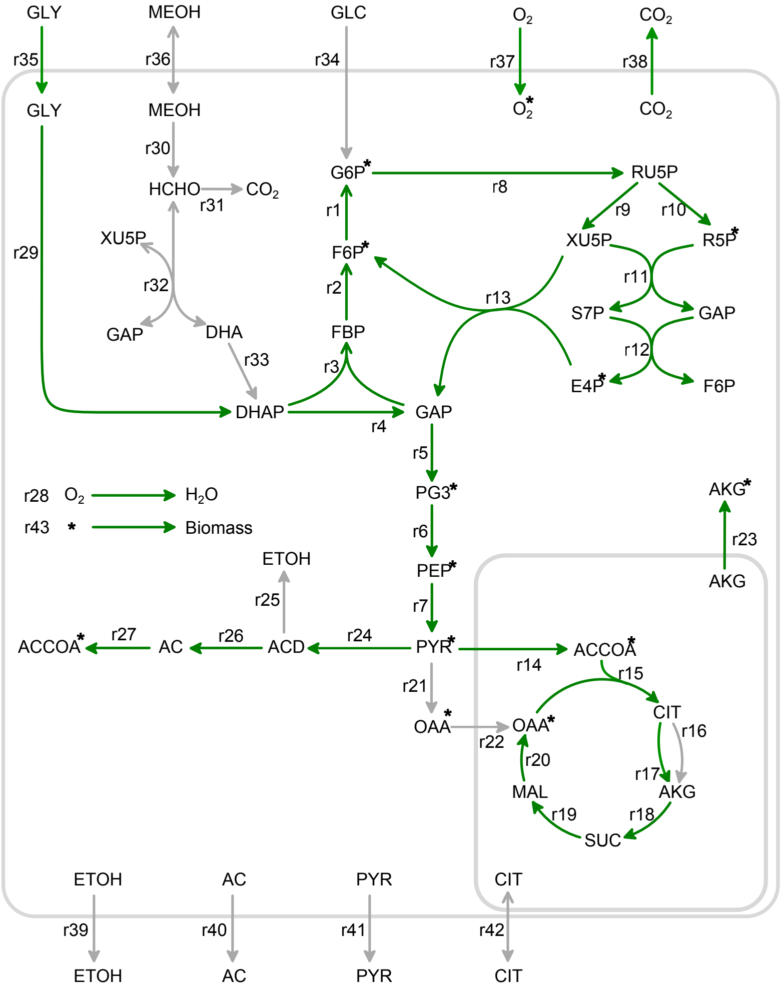


Figure 6.21: *P. pastoris* real case study. EM_{10} represented onto the metabolic network.

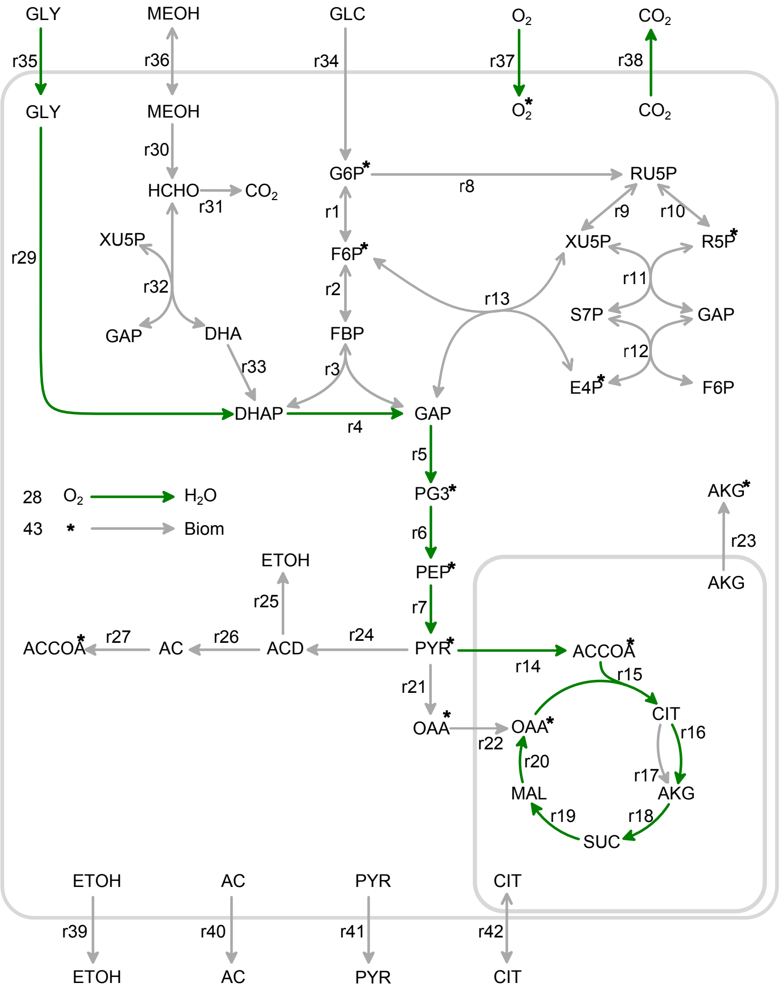


Figure 6.22: *P. pastoris* real case study. EM_{147} represented onto the metabolic network.

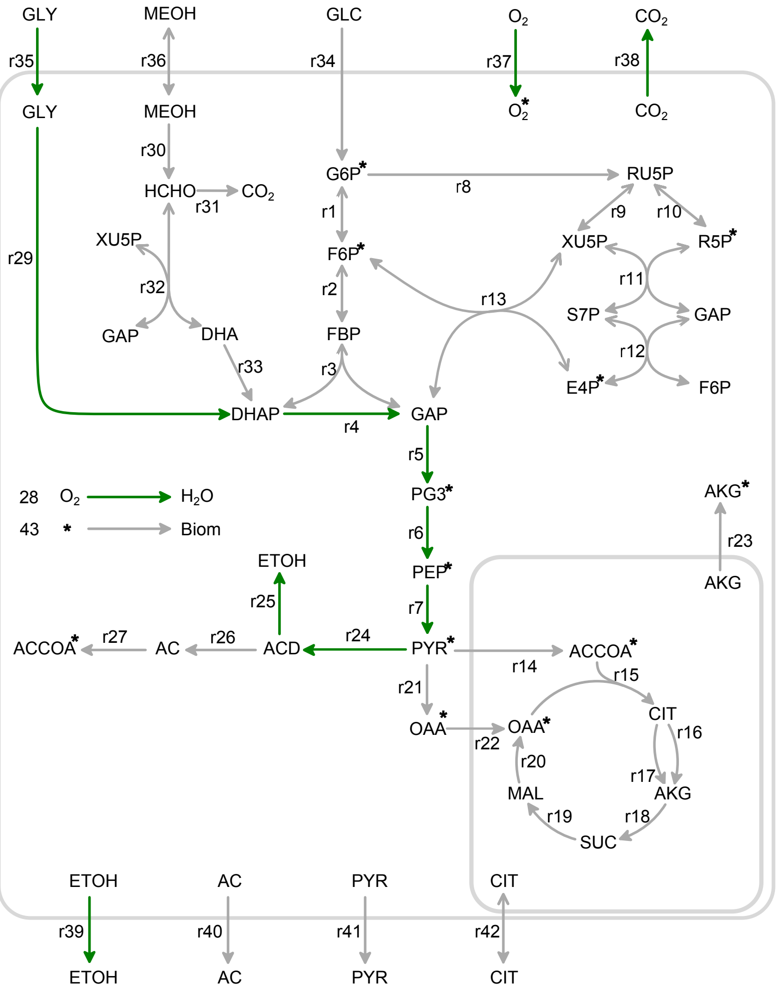


Figure 6.23: *P. pastoris* real case study. EM_{149} represented onto the metabolic network.

Ru5P	Ribulose-5-phosphate
R5P	Ribose-5-phosphate
X5P	Xylulose-5-phosphate
S7P	Sedoheptulose-7-phosphate
E4P	Erythrose–4-phosphate
AcCoA	Acetyl coenzyme A
OAA	Oxaloacetate
CIT	Citric acid
ICT	Citrullin (intracellular)
2-KG	2-Amino-6-ketopimelate
SUC	Succinate
FUM	Fumarate
MAL	Malate
OAA	Oxalate
Glyox	Glyoxylate
Acetate	
Lactate	
Ethanol	

List of reactions

Glucose + PEP → G6P + PYR

G6P ↔ F6P

F6P → F1,6P

F1,6P → DHAP + G3P

DHAP → G3P

G3P → 3PG

3PG ↔ PEP

PEP → PYR

PYR → AcCoA + CO2

G6P → 6PG

6PG → Ru5P + CO2

Ru5P → X5P

Ru5P → R5P

R5P + X5P ↔ S7P + G3P

S7P + G3P ↔ E4P + F6P

X5P + E4P ↔ F6P + G3P

AcCoA + OAA → CIT

CIT → ICT

ICT → 2-KG + CO2

2-KG → SUC + CO2

SUC → FUM

FUM → MAL
MAL ↔ OAA
PEP + CO2 ↔ OAA
MAL → PYR + CO2
ICT → Glyoxylate + SUC
Glyoxylate + AcCoA → MAL
AcCoA → Acetate
PYR → Lactate
AcCoA → Ethanol
G6P → (Cell synthesis)
F6P → (Cell synthesis)
R5P → (Cell synthesis)
E4P → (Cell synthesis)
G3P → (Cell synthesis)
3PG → (Cell synthesis)
PEP → (Cell synthesis)
PYR → (Cell synthesis)
AcCoA → (Cell synthesis)
OAA → (Cell synthesis)
2KG → (Cell synthesis)
CO2 → (Evolution)

P. pastoris

Metabolite abbreviations

Abbreviation	**Metabolite**
G6P[c]	Glucose-6-phosphate (cytosol)
F6P[c]	Fructose-6-phosphate (cytosol)
FBP[c]	Fructose 1,6-biphosphate (cytosol)
DHAP[c]	Dihydroxyacetone phosphate (cytosol)
GAP[c]	D-glyceraldehyde 3-phosphate (cytosol)
PG3[c]	3 Phosphoglycerate (cytosol)
PEP[c]	Phosphoenolpyruvate (cytosol)
PYR[c]	Pyruvate (cytosol)
RU5P[c]	Ribulose-5-phosphate (cytosol)
XU5P[c]	Xylulose-5-phosphate (cytosol)
R5P[c]	Ribose-5-phosphate (cytosol)
S7P[c]	Sedoheptulose-7-phosphate (cytosol)
E4P[c]	Erythrose–4-phosphate (cytosol)
ACCOA[m]	Acetyl coenzyme A (mitochondrial)
OAA[m]	Oxaloacetate (mitochondrial)

CIT[m]	Citric Acid (mitochondrial)
AKG[m]	2-Amino-6-ketopimelate (mitochondrial)
SUC[m]	Succinate (mitochondrial)
MAL[m]	Malate (mitochondrial)
OAA[c]	Oxaloacetate (cytosol)
AKG[c]	2-Amino-6-ketopimelate (cytosol)
ACD[c]	Acetaldehyde (cytosol)
ETOH[c]	Ethanol (cytosol)
AC[c]	Acetate (cytosol)
ACCOA[c]	Acetyl coenzyme A (cytosol)
GLY[c]	Glycerol (cytosol)
MEOH[c]	Methanol (cytosol)
HCHO[c]	Formaldehyde (cytosol)
DHA[c]	Dihydroxyacetone (cytosol)
NADH	Nicotinamide adenine dinucleotide
NADPH[c]	Nicotinamide adenine dinucleotide phosphate (cytosol)
NADPH[m]	Nicotinamide adenine dinucleotide phosphate (mitochondrial)
CO2[i]	Carbon dioxide (internal)
O2[i]	Oxygen (internal)
GLC	Glucose
GLY	Glycerol
MEOH	Methanol
O2	Oxygen
CO2	Carbon dioxide
ETOH	Ethanol
AC	Acetate
PYR	Pyruvate
CIT	Citric acid
BIOM	Biomass

List of reactions

G6P[c] $\leftrightarrow$ F6P[c]
F6P[c] $\leftrightarrow$ FBP[c]
FBP[c] $\leftrightarrow$ DHAP[c] + GAP[c]
DHAP[c] $\leftrightarrow$ GAP[c]
GAP[c] $\leftrightarrow$ PG3[c] + NADH
PG3[c] $\leftrightarrow$ PEP[c]
PEP[c] $\leftrightarrow$ PYR[c]
G6P[c] $\rightarrow$ RU5P[c] + 2 NADPH[c] + CO2[i]
RU5P[c] $\leftrightarrow$ XU5P[c]
RU5P[c] $\leftrightarrow$ R5P[c]

XU5P[c] + R5P[c] ↔ GAP[c] + S7P[c]
GAP[c] + S7P[c] ↔ F6P[c] + E4P[c]
XU5P[c] + E4P[c] ↔ F6P[c] + GAP[c]
PYR[c] → ACCOA[m] + NADH + CO2[i]
ACCOA[m] + OAA[m] → CIT[m]
CIT[m] → AKG[m] + NADH + CO2[i]
CIT[m] → AKG[m] + NADPH[m] + CO2[i]
AKG[m] → SUC[m] + NADH + CO2[i]
SUC[m] → MAL[m] + NADH
MAL[m] → OAA[m] + NADH
PYR[c] + CO2[i] → OAA[c]
OAA[c] ↔ OAA[m]
AKG[m] → AKG[c]
PYR[c] → ACD[c] + CO2[i]
ACD[c] + NADH → ETOH[c]
ACD[c] → AC[c] + NADPH[c]
AC[c] → ACCOA[c]
NADH + 0,5 O2[i] →
GLY[c] ↔ DHAP[c] + NADH
MEOH[c] + 0,5 O2[i] → HCHO[c] + DHAP[c]
HCHO[c] → 2 NADH + CO2[i]
HCHO[c] + XUP5[c] ↔ DHA[c] + GAP[c]
DHA[c] → DHAP[c]
GLC → G6P[c]
GLY ↔ GLY[c]
MEOH → MEOH[c]
O2 → O2[i]
CO2[i] → CO2
ETOH[c] → ETOH
AC[c] → AC
PYR[c] → PYR
CIT[m] → CIT
- 0,0364 G6P[c] - 0,0364 F6P[c] - 0,0165 PG3[c] - 0,0252 PEP[c] - 0.0294 PYR[c] - 0,0107 R5P[c] - 0.0146 E4P[c] - 0,008 ACCOA[m] - 0,0008OAA[m] - 0,0242 OAA[c] - 0,0267 AKG[c] - 0,0033 ACCOA[c] - 0,00000003 GLY[c] + 0,0627 NADH - 0,1995 NADPH[c] - 0,0561 NADPH[m] + 0,0177 CO2[i] - 0,000024 O2[i] → BIOM

Chapter 7

Dynamic elementary mode modelling

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

[13] Folch-Fortuny, A., Teusink, B., Kiers, H.A.L., Hoefsloot, H.C.J., Smilde, A.K. & Ferrer, A. Dynamic elementary mode analysis of non-steady state flux data. In preparation.

7.1 Introduction

In this thesis, exploratory models have been used to model non-steady state flux data. In Chapter 5, PCA is used to find a set of pathways built upon the existing relationships between fluxes. Also, MCR is proposed to model this kind of data due to its ability to include biological constraints in the multivariate model.

In chapters 5-6, different multivariate exploratory methods have been applied to model steady state flux data, including different sources of biological information in the model, depending on the approach. Regarding predictive models, PLS is widely used in metabolomics to relate a set of explanatory variables and a set of biological outputs using the latent structure of data. Especially PLS discriminant analysis (PLS-DA) is commonly applied to distinguish between biological conditions, such as a particular illness. For example, in [213] this technique was used to discriminate between non-steatotic and steatotic human liver profiles, and in [214] PLS-DA was used for the diagnosis of inherited metabolic disorders (IMDs), analysing plasma and blood samples of subjects with phenylketonuria and medium chain acyl CoA dehydrogenase deficiency. In these studies, the PLS-DA model is used for finding potential biomarkers among the pool of metabolites analised.

Despite PLS-DA is a very powerful approach to compress and interpret large amounts of data, it lacks the ability to include topological information in the model, as it happens in PCA for steady state flux data. In this chapter, a novel framework is proposed to analyse non-steady state metabolite concentrations. The methdology is based on an extention of the PEMA model presented in Chapter 6. Introducing the concept of dynamic EMs (dynEMs), i.e. EMs activated partially at each time point, the most relevant pathways activated in an experiment, or a set of experiments, can be identified. This method is called dynamic elementary mode analysis (dynEMA). Furthermore, the ultimate interest consists of identifying which metabolic routes have different performances depending on the initial conditions. Therefore, using dynamic elementary mode regression discriminant analysis (dynEMR-DA), these most discriminant pathways can be identified among large flux data sets.

This has been previously investigated [215] using the Goeman's global test and the set of pathways retrieved from the KEGG database [216–218]. The aim was to find what pathways have a different activation pattern depending on the initial conditions of the experiment. In fact, this chapter analyses the same concentration data sets. The approach presented here differs from the aforementioned in i) here the set of EMs is used instead of the KEGG pathways, which sometimes do not connect directly substrates with end-products, and ii) in dynEM modelling, all the possible pathways are analysed within a single multivariate model, instead of individual pathway testing.

The structure of this chapter is as follows. First, the metabolic models and data sets of *S. cerevisiae* used in this work are presented. In Sections 7.3-7.4, the adaptation of the PEMA model from a steady to a non-steady state environment is introduced, describing dynEMA, dynEMR-DA and the validation scheme. In the Section 7.5, the output of dynEMR-DA is analysed using simulated and actual concentration data. Finally, some conclusions are drawn in the Section 7.6.

7.2 Metabolic models of *Saccharomyces cerevisiae*

7.2.1 Metabolic networks

Two metabolic models of the well-known baker's yeast, *S. cerevisiae*, are used in this chapter to build the discriminant models (see Section 7.7 for a list of reactions). The first one was used in [219] to study the dynamics in glycolysis. The metabolic network (see Figure 7.1a) has 23 metabolites and 18 reactions. The second model was proposed in [129], and comprises 12 metabolites and 20 reactions. This second model describes the glycolysis and the TCA cycle (see Figure 7.1b). Two models are used in this chapter since the metabolites whose measurements were available in the real case study were not exactly the same as in the available simulated model. However, since both models are describing glycolysis, the results are somehow comparable.

7.2.2 Concentration data

The concentration data using the first model (Figure 7.1a) are simulated using COPASI software. The initial conditions of the metabolites match the meaurements used in the original paper [219] (see Table 7.1). In this case, COPASI is used to simulate the concentrations from 0 to 1 seconds in 20 intervals of 0.05 seconds. The fluxes and the set of EMs are also obtained directly from this software.

The interest in the simulated study consists of discriminating between scenarios using a large *versus* small amount of glucose. Therefore, 32 experiments are simulated using the data in Table 7.1 plus a 20% noise, and another set of 32 is obtained tunning the original glucose concentration from 10 to 2.5 mMol/l (also with 20% noise in data). These two values are indeed interesting, since they mimic the glucose pulses used in the real case study (see paragraph below).

In the real case, the concentrations of *S. cerevisiae* were obtained experimentally using LC-MS [220, 221] at the Kluyver Centre for Genomics of Industrial Fermentation (Biotechnology Department, TU Delft, The Netherlands), and were used afterwards in [215]. 12 different cultures are used in the present work. Regarding experiments 1 to 8, different glucose pulses in aerobic conditions were used in these

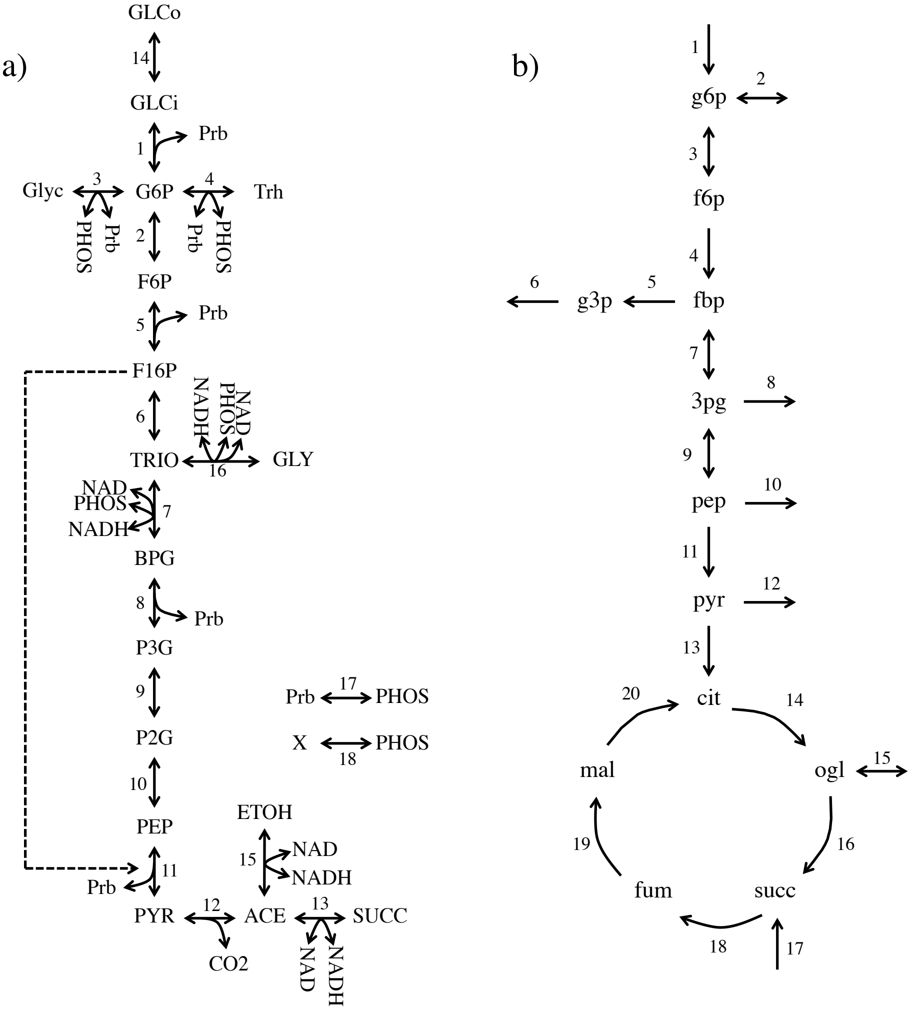


Figure 7.1: *S. cerevisiae* metabolic models. Model a) [219] (b) [215]) is used for the simulated (real) case study.

Metabolite	**Initial concentration (mMol/l)**
GLCi	0.087
Prb	5
G6P	3.085
F6P	0.75247
Glyc	0
PHOS	10
Trh	0
F16P	0.836
TRIO	0.5177
NAD	0
BPG	0.111
NADH	0.044
P3G	0.825
P2G	0.13771
PEP	0.1404
PYR	0.884031
ACE	0.0474837
CO2	1
SUCC	0
GLCo	110
ETOH	0
GLY	0.15
X	0

Table 7.1: Initial concentrations in the simulated study. Experimental conditions taken from [219].

cultures: 10 mMol of glucose were used in the first 4 experiments and 2.3-2.5 mMol in experiments 5-8. Also, 4 more cultures, experiments 9 to 12, were performed in anaerobic conditions.

The aim in the real case study consists of discriminating between i) large and small glucose pulses (i.e. experiments 1-4 vs 5-8), and ii) aerobic and anaerobic conditions (experiments 5-8 vs 9-12).

7.3 Dynamic elementary mode modelling

7.3.1 Dynamic elementary mode analysis (dynEMA)

Any steady state flux distribution $\mathbf{x} = (x_1, ..., x_K)$ can be decomposed as a positive linear combination of a set of E EMs [139]: $\mathbf{x} = \sum_{e=1}^{E} \lambda_e \mathbf{p}_e$, where K is the number of fluxes (matching the number of reactions in the network), $\mathbf{p}_e = (p_{e1}, ..., p_{eK})$ is the EM e, λ_e is the positive weighting factor of EM e, and E is the number of EMs needed to reconstruct the flux distribution $\mathbf{x}$.

When N flux distributions are considered, coming from different experiments or cultures, a PEMA model can be built: $\mathbf{X} = \mathbf{\Lambda}\mathbf{P}^{\mathrm{T}} + \mathbf{F}$, where $\mathbf{X}$ is the $N \times K$ flux data matrix, $\mathbf{P}$ is the $K \times E$ PEMs matrix, formed by a subset of E EMs; $\mathbf{\Lambda}$ is the $N \times E$ weighting matrix; and $\mathbf{F}$ is the $N \times K$ residual matrix. A schematic representation of PEMA model can be visualised in Figure 7.2.

Non-steady state flux distributions cannot be decomposed as linear combinations of EMs, as in steady state. When the biological system has not reached yet the steady state, the dynamics are unstable, and there could be flux only in some areas of the network, e.g. reactions consuming the initial substrates. However, the EMs are indeed the simplest pathways along which the non-steady state fluxes have to flow, but not in a stable or constant fashion. Following this rationale, the EMs can be modified or adapted to fit this unstability. This are the so-called dynamic elementary modes (dynEM). To adapt an EM, not a single coefficient multiplying the EM ($\mathbf{\Lambda}$ in PEMA), but a coefficient multiplying each reaction activated by the EM has to be assigned.

Thus, a single non-steady state flux distribution $\mathbf{x}$ can be decomposed as:

$$\mathbf{x} = \sum_{e=1}^{E} \boldsymbol{\alpha}_e \circ \mathbf{p}_e \tag{7.1}$$

where $\boldsymbol{\alpha}_e = (\alpha_{e1}, ..., \alpha_{eK})$ are the coefficients that adapt reactions 1 to K in the selected dynamic EM e to reproduce the fluxes in $\mathbf{x}$.

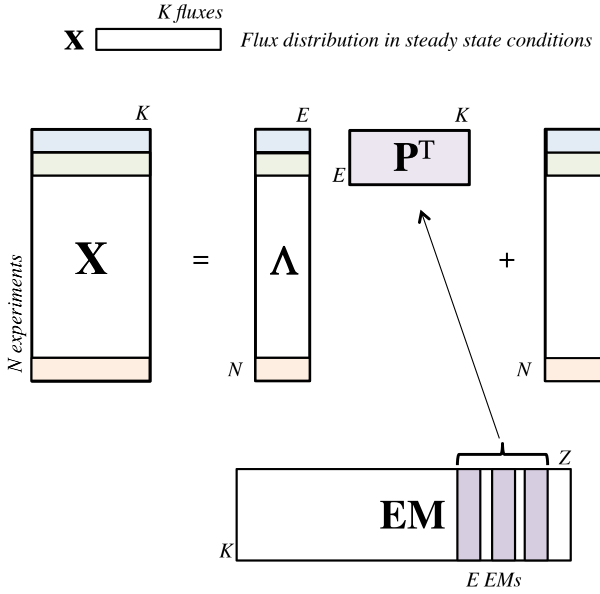


Figure 7.2: Scheme of PEMA model.

Consider now a set of non-steady state flux distributions, which can be obtained from a single experiment measuring the concentration of the metabolites at J consecutive time points. The set of active dynEMs are obtained from the dynEMA model:

$$\mathbf{X} = (\mathbf{I}_J \otimes \mathbf{1}_E^{\mathrm{T}})[\mathbf{A} \circ (\mathbf{1}_J \otimes \mathbf{P}^{\mathrm{T}})] + \mathbf{F} \tag{7.2}$$

where $\mathbf{A}$ is the $EJ \times K$ coefficients matrix and $\mathbf{I}_J$ is the $J \times J$ identify matrix. The other matrices are the same as in the PEMA model. Figure 7.3 shows a representation of dynEMA model.

The coefficients matrix $\mathbf{A}$ in the previous equation is indeed a $E \times K \times J$ three-way VWU matrix, and each entry in the matrix α_{ekj} represents the coefficient multiplying reaction k of EM e to reconstruct the flux x_k at time point j. Using this modeling it is possible to study the time evolution of a dynEM, i.e. how the dynEM is deformed or dynamically used along all measured time points.

This system of equations is solved similarly to PEMA. The candidates for first dynEM are selected from the complete $K \times Z$ **EM** matrix in a step-wise fashion. After selecting an EM, the coefficients multiplying it (thus creating the dynEM) are obtained solving 7.2 using non-negative least squares. Once all EMs are evaluated,

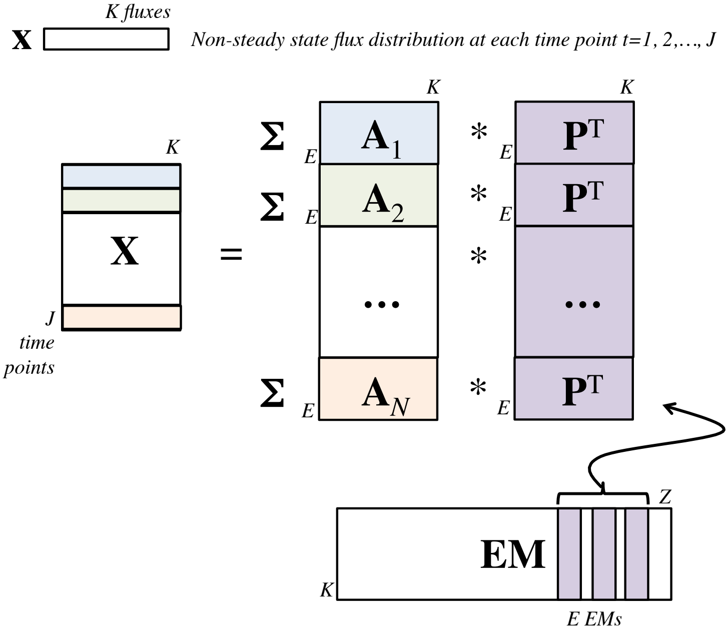

K fluxes
X
Non-steady state flux distribution at each time point t=1, 2,…, J
K
X
J
time
points
=
Σ
Σ
Σ
E
E
E
K
A1
A2
…
AN
*
*
*
*
E
E
E
K
PT
PT
…
PT
Z
EM
K
E EMs

Figure 7.3: Scheme of dynEMA.

the dynEM explaining most variance in data (as in PEMA) is classified as the first dynEM. Afterwards, this first dynEM is fixed, and the search for the second one starts, recalculating the coefficients in matrix **A** for both the first and the second dynEMs at each evaluation. In this way, the dynEMA model is built in a greedy way.

The dynEMA model is useful to identify the dynEMs active in an experiment and how each dynEM is used in the culture at different time points of the experiment.

7.3.2 Dynamic elementary mode regression discriminant analysis (dynEMR-DA)

When the aim is to establish differences between environmental or experimental conditions, e.g. presence/absence of a compound or case/control studies, a discriminant model is needed. For this, dynEMR-DA is proposed. This model focuses on finding which are the dynEMs with a strongly different time evolution or performance between conditions.

To build a dynEMR-DA model, the set of different experiments are combined in a single $\underline{\mathbf{X}}$ three-way array (see Figure 7.4). In $\underline{\mathbf{X}}$ we consider N experiments,

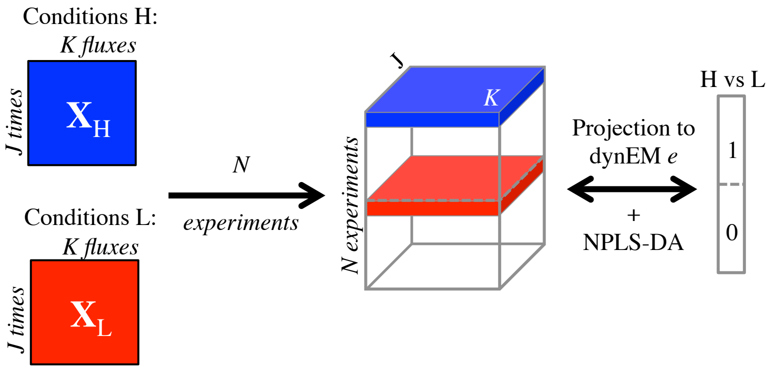


Figure 7.4: dynEMR-DA procedure.

measuring K fluxes along J time points. Therefore, it is mandatory to have the same measurement rate in all experiments.

The algorithm of dynEMR-DA is:

1. For each EM:

   (a) Unfold the $\underline{\mathbf{X}}$ matrix using VWU.

   (b) Calculate the coefficients matrix $\mathbf{A}$ using dynEMA.

   (c) Reconstruct the flux data using $\mathbf{A}$ and $\mathbf{P}$.

   (d) Fold the reconstructed data to build again a three-way data structure $\underline{\mathbf{X}}_{rec}$

   (e) Fit an NPLS-DA model between the reconstructed data and the $\mathbf{y}$ data, where $\mathbf{y}$ denotes the class of experiments (1s or 0s).

2. The dynEM whose NPLS-DA model explains most variance in $\mathbf{y}$ is classified as the first dynEM.

3. Check the predictions of NPLS-DA model. If the current model discriminates perfectly, stop. If not, fix the first dynEM and repeat steps 1-3 to extract the second dynEM.

The dynEMR-DA algorithm can select many dynEMs until attaining a perfect discrimination. However, in practice, many dynEMs are able, separatedly, to discriminate between two experimental conditions. Moreover, some dynEMs are discriminant, but some of their reactions are not used at any time point of the experiment (so the flux does not cross the metabolic pathway from the beginning to the end). These dynEMs do not represent actual metabolic pathways, so they should be removed when they are selected as discriminant.

The unfolding of $\underline{\mathbf{X}}$ flux data matrix can also be applied in dynEMA, as a prior step to the analysis. In this way, the common dynEMs activated in a pool of experiments can be identified and interpreted using visual tools (see Section 7.5).

7.4 Triple cross-validation procedure (3CV)

Proper validation of discriminant models is a subtle issue in systems biology. When enough data is available, single cross-validation procedures may lead to too optimistic models, especially when the aim is discrimination between classes. To avoid this, sometimes, spurious results in classification, double cross validation (2CV) was proposed [222]. Using this procedure, a subset of the original data is used to model fitting, another subset to decide the complexity of the model (e.g. number of components of a multivariate model), and finally, a third subset is used for validation. This kind of models are especially useful for (N)PLS-DA model validation [222, 223].

In this work, though, we need an extra round of validation. dynEMR-DA models involve the projection, as first step, of the flux data into the space defined by a single dynEM. Afterwards, an NPLS-DA model is fitted with discriminant purposes, having to determine, at the end, what dynEMs are discriminant. Therefore, we propose here a triple cross validation (3CV) scheme (see Figure 7.5). This procedure consists of the following steps:

1. Divide the data set in four groups: calibration, test, selection, and validation. The latter is left out of the analysis until the final external validation.

2. Fit a dynEMR-DA model using the calibration set, using a maximum of 15 components. In this case study, 15 are selected because this number approaches the number of reactions in both *S. cerevisiae* metabolic models.

3. Project the test set, first to the corresponding dynEM, and then to each of the 15 NPLS-DA calibration models. At this point, the minimum number of components, A, needed to classify each experiment in its corresponding class, are selected.

4. Project the selection set first to the dynEM and then to the calibration NPLS-DA model with A components. Then, the predictive power of each dynEM using these data is assessed.

5. Steps 2-4 are repeated three times, changing the roles of the subsets. That is, the models are built using, in steps 2 to 4 respectively: calibration-test-selection, test-selection-calibration and selection-calibration-test sets.

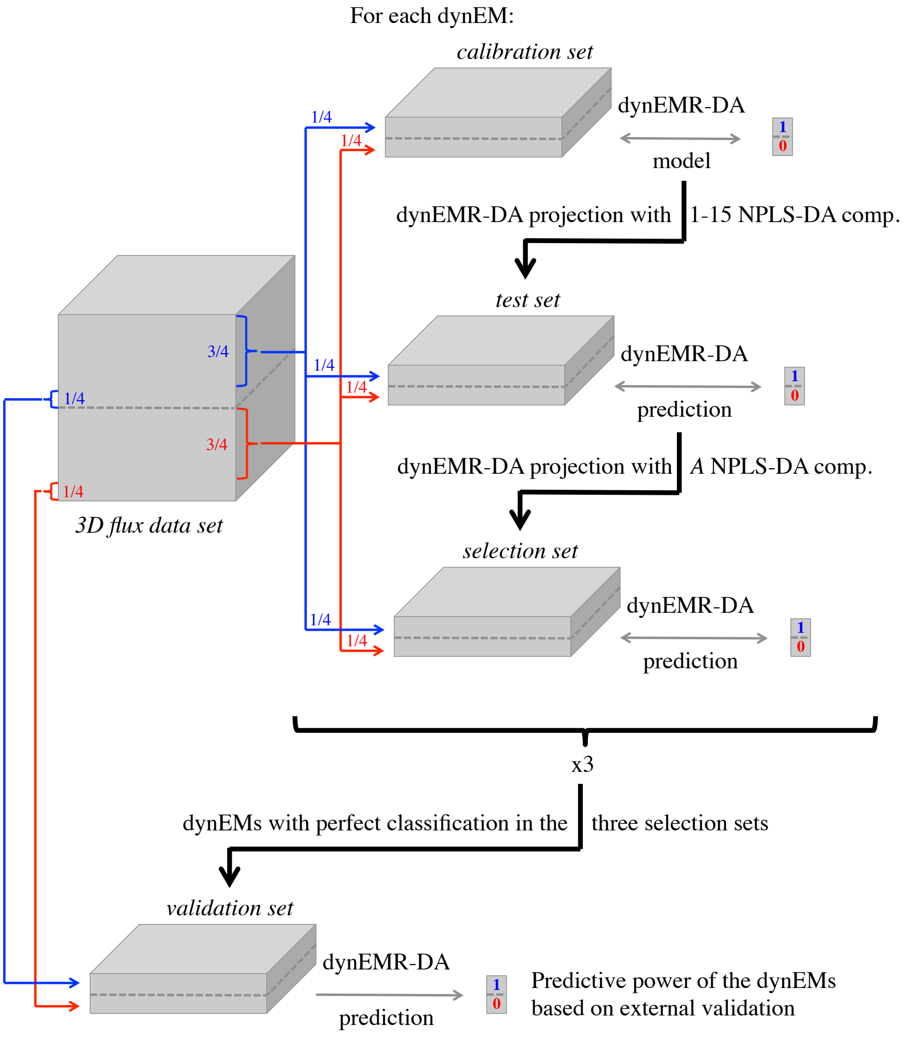

For each dynEM:
calibration set
1/4
1/4
dynEMR-DA
model
1
0
dynEMR-DA projection with 1-15 NPLS-DA comp.
test set
3/4
1/4
1/4
1/4
dynEMR-DA
prediction
1
0
3/4
1/4
dynEMR-DA projection with A NPLS-DA comp.
3D flux data set
selection set
1/4
1/4
dynEMR-DA
prediction
1
0
x3
dynEMs with perfect classification in the three selection sets
validation set
dynEMR-DA
prediction
1
0
Predictive power of the dynEMs based on external validation

Figure 7.5: 3CV procedure.

6. The dynEMs with perfect classification rates using the selection set in the three rounds are used finally for validation, so the discrimination power of each dynEM is evaluated with completely external data. This prediction is performed substituting the selection group by these validation samples in the three models previously fitted.

A 2CV strategy is used for the NPLS-DA section of the dynEMR-DA models, but an extra validation round is needed to assess the discriminant performance of the selected dynEMs. Therefore, the 3CV procedure is built basically replacing the validation step, in the original 2CV, by the selection step, and performing the external validation in the last step.

7.5 Results

7.5.1 Simulated flux data

The metabolic model of *S. cerevisiae* in Figure 7.1a is used in this section to assess the performance of dynEMR-DA on simulated data. 64 experiments are simulated using COPASI, with the initial concentrations described in Section 7.2 (see Table 7.1). Thus, 32 experiments have a high initial concentration of glucose and 32 a low concentration. The fluxes derived from the concentration data, and also the set of EMs of the metabolic model, are also obtained using the aforementioned software.

To validate the discriminant models, the 3CV scheme is used here, using N-way Toolbox for MATLAB to fit the NPLS-DA models. 8 experiments of each class selected at random (16 in total) are used as the calibration set. 16 more experiments are used to select the number of NPLS-DA components. And 16 more are used as selection samples. After repeating these procedure, changing the roles of the three groups, only one dynEM (from the whole set of 26 EMs) is able to discriminate perfectly between both experimental conditions: dynEM 8. This means that it classifies each experiment in its corresponging class in the three groups when acting with different roles (i.e. calibration, validation and selection sets). Finally, the remaining 16 cultures are used for the final validation of this dynEM. Again, all experiments are correctly classified in all dynEMR-DA models.

Figure 7.6a shows dynEM$_8$. This mode covers the whole glycolytic pathway, starting from glucose (GLCo), producing all the intermediate products until reaching pyruvate (PYR), acetate (ACE) and finally ethanol (ETOH). The coefficients multiplying the EM can be visualized in Figures 7.6b-7.6e. The system reaches the steady state at time point 5-6, so the differences between time points are minimum afterwards.

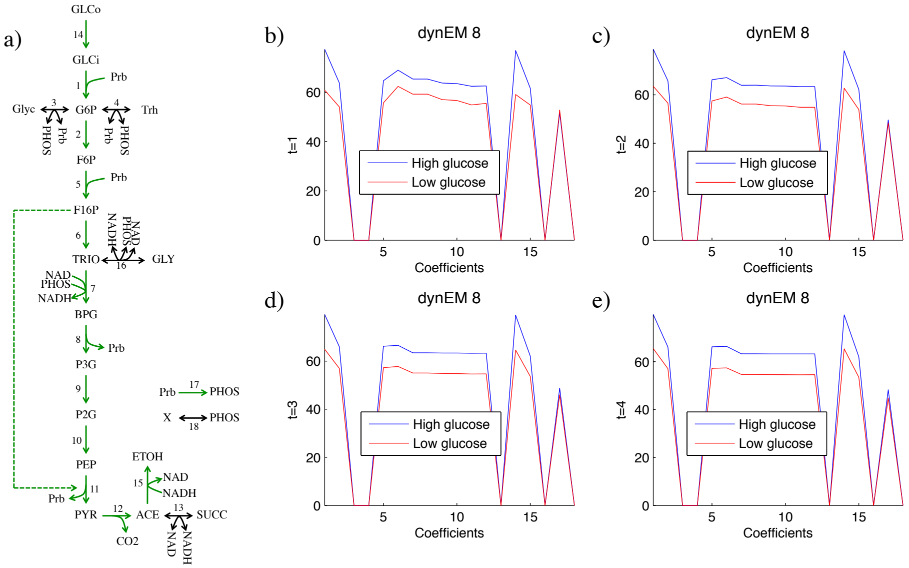


Figure 7.6: Simulated study. a) dynEM$_8$ depicted on the metabolic model. b)-e) dynEM$_8$ coefficients at time points 1-4. Blue (red) lines show the coefficients for the high (low) glucose experiments.

The differences between both experimental conditions can be seen with the naked eye in Figure 7.6. The usage of all reactions in the dynEM, i.e. the coefficients in **A** matrix, are higher in the high glucose concentration experiments than in the low glucose. This implies that these scenarios take advantage of the higher amount of glucose to carry more flux through the glycolysis until reaching the ethanol. This behaviour has been somehow commented in the literature [224–226], and in Chapter 6 with steady state flux data, and it is known as the Crabtree effect.

7.5.2 Actual flux data

High vs low glucose pulse

To assess the performance of dynEMR-DA in a real case study, a set of cultures of *S. cerevisiae* are used to discriminate between experiments using a large or a small initial glucose pulse. Unfortunately, the number of available cultures is low for this case study (4 in each class), so no 3CV, neither 2CV, is possible here. Therefore, single CV is applied here: 3+3 experiments are used for dynEMR-DA model building and selection of NPLS-DA components, and the remaining 1+1 experiments are used for validation. This procedure is repeated 4 times, leaving out a couple of cultures each time.

The dynEMR-DA model has to be built using fluxes, not concentrations. Therefore, we have to compute the fluxes based on the changes in the concentrations between two consectutive time points. To obtain the set of fluxes all at once, the following optimization problem is solved:

$$\begin{cases} \min_{x_{jk}} \sum_{j=1}^{20} \sum_{k=1}^{24} (x_{j+1,k} - x_{j,k})^2 + \sum_{j=1}^{20} \sum_{k=1}^{24} x_{j,k}^2 \\ s.t. \quad \mathbf{S}\mathbf{X}^{\mathrm{T}} = \frac{d\mathbf{C}^{\mathrm{T}}}{dj} \\ \qquad \mathbf{X} \geq \mathbf{0} \\ \qquad \mathbf{X}_0 \text{ initial solution} \end{cases} \tag{7.3}$$

where $\mathbf{X} = \{x_{jk}\}$ is the 23×20 flux data matrix, $\mathbf{X}_0$ is the initial solution for the quadratic programming problem (based on a non-negative least squares solution of $\mathbf{S}\mathbf{X}_0^{\mathrm{T}} = \frac{d\mathbf{C}^{\mathrm{T}}}{dj}$), indices k and j denote flux number and time point, respectively, $\mathbf{S}$ denote the 12×20 stoichiometric matrix, and $\mathbf{C}$ denote the 24×12 concentration matrix. There are 24 time points in the concentration data, thus 23 time points are considered for flux data, representing the fluxes between consecutive pairs of concentrations.

In this case, only dynEM$_9$ (from the set of 20 EMs) is able discriminate the left out experiments in all cases. This dynEM can be visualised, jointly with the coefficient matrix $\mathbf{A}$, in Figure 7.7. The differences between high and low glucose are also pretty clear in this example. The usage of this dynEM is stronger in scenarios with a huge glucose pulse than with a small pulse. This difference is, though, greater in the first steps of the glycolysis, which makes sense, since the effect of the higher amount of glucose is diluted when the flux is crossing the pathway. It can also be seen that the first reactions of the EM (1, 3 and 4) have higher coefficients at the first time point and lower ones at time points 3-4. The opposite happens with the subsequent reactions in the dynEM, which have low coefficients at the beggining and higher ones at time point 4. This behaviour reinforces the modelling applied in this work. The flux data cannot be modelled in the same way at the first time points than when the culture reaches the steady state, therefore it make sense to use the concept of dynEMs to model non-steady state flux data, instead of applying a PEMA model.

It is worth noting the similarity between the dynEM identified here and dynEM$_8$ of the simulated case study. Both dynEMs are describing the same phenomena, the glycolysis until reaching pyruvate. They are not exactly the same because the metabolic models are different, acetate and ethanol were not measured in experimental conditions. However, it seems that when comparing simulated and actual data, the dynEM discriminating between experimental conditions is basically the same one.

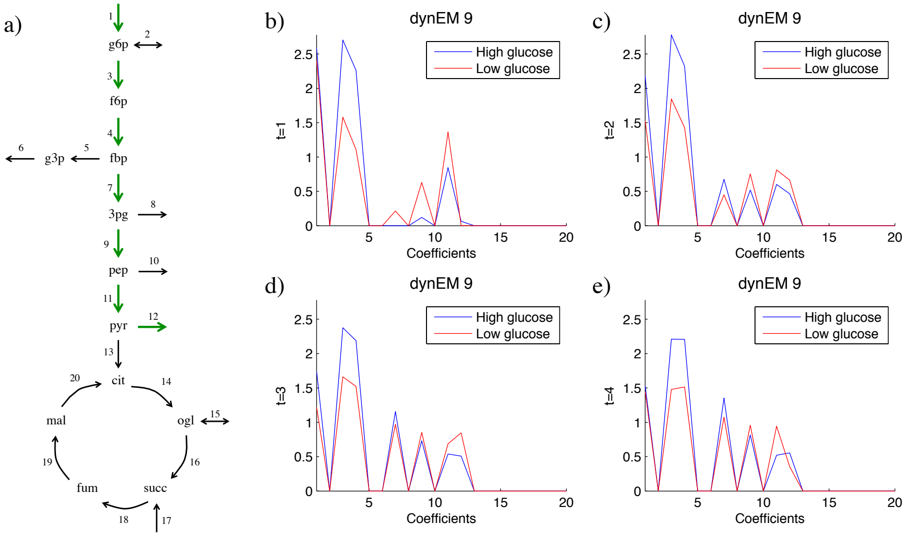


Figure 7.7: Real case study. a) dynEM$_9$ depicted on the metabolic model. b)-e) dynEM$_9$ coefficients at time points 1-4. Blue (red) lines show the coefficients for the high (low) glucose experiments.

Aerobic vs anaerobic conditions

For the second real case study, we compare 4 cultures performed in aerobic conditions versus 4 more in anaerobic conditions. As in the previous example, a single cross validation procedure is applied here.

In this case study, dynEM$_8$ is able to discriminate between both experimental conditions. The dynEM and the coefficients for the first 4 time points can be visualized in Figure 7.8. Again, the differences between both classes can be seen with the naked eye, having the anaerobic experiments higher coefficients. This behaviour has been outlined also in the literature [224, 227–229]. To satisfy the redox balances, glucose is deviated from glycolysis to the production of glycerol. The latter is produced by reduction of the glycolytic intermediate dihydroxyacetone phosphate to glycerol 3-phosphate (g3p) followed by a dephosphorylation of glycerol 3-phosphate to glycerol. Despite glycerol does not appear explicitly in the network, because this metabolite was not measured in all original experiments, it is likely that the flux flowing through g3p produce glycerol at the end, as suggested in the literature.

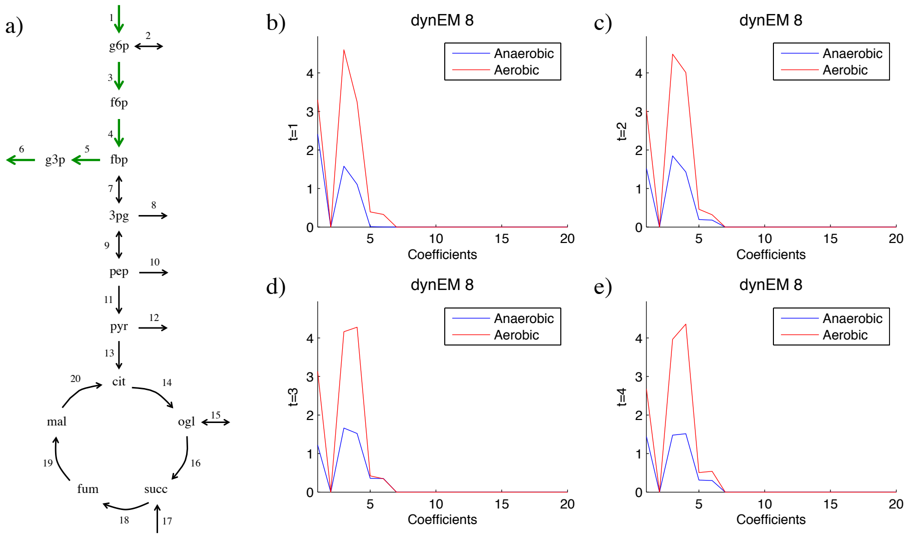


Figure 7.8: Real case study. a) dynEM$_8$ depicted on the metabolic model. b)-e) dynEM$_8$ coefficients at time points 1-4. Blue (red) lines show the coefficients for aerobic (anaerobic) experiments.

7.6 Discussion

The approach for dynEM modelling proposed here permits decomposing non-steady state flux distributions into a set of active dynEMs. This way, dynEMA can be used to study the active dynEMs in an experiment, or a set of experiments, extending the PEMA model, proposed in Chapter 6, to a dynamic environment. For discrimination purposes, dynEMR-DA permits to identify which dynEMs have different patterns of activation depending on the culture initial conditions.

Actual and simulated concentration data of *S. cerevisiae* have been used here, to evaluate dynEMR-DA. When changing the amount of glucose present in the experiment in both data sets, dynEMR-DA is able to identify that the dynEM crossing the glycolytic pathway from glucose to pyruvate is the most discriminant one. Even considering two different metabolic models, for data availability reasons, the results of dynEMR-DA seem coherent between case studies. When analysing data from aerobic *versus* anaerobic conditions, dynEMR-DA indicates that the dynEM driving the glucose pulse to the glycerol production is the most discriminant in terms of usage between both classes. Previously published research confirms the results obtained using this new methodology.

The framework presented here will serve to create reduced dynamic models of flux data while preserving biological and thermodynamical meaning, as a tool to analyse non-steady state flux distributions across experiments and to identify the hidden metabolic patterns that drive the organism from one state to another when changing the environmental conditions. dynEMA and dynEMR-DA have potential applications in bioprocess engineering to understand the small changes in cell metabolism at early stages of cultures.

7.7 Appendix. Metabolic models.

Simulated case study

Metabolite abbreviations

Abbreviation	**Metabolite**
GLCo	Glucose
GLCi	Glucose (intracelullar)
Prb	Energy status
G6P	Glucose-6-phosphate
F6P	Fructose 1,6-phosphate
Glyc	Glycogen
PHOS	Phosphate
Trh	Trehalose
F16P	Fructose-6-biphosphate
TRIO	Triose-phosphates
NAD	Nicotinamide adenine dinucleotide
BPG	Bisphosphoglycerate
NADH	Nicotinamide adenine dinucleotide phosphate
P3G	3-Phosphoglycerate
P2G	2-Phosphoglycerate
PEP	Phosphoenolpyruvate
PYR	Pyruvate
ACE	Acetate
CO2	Carbon dioxide
SUCC	Succinate
ETOH	Ethanol
X	Polyphosphates

List of reactions

GLCi + Prb ↔ G6P
G6P ↔ F6P
G6P + Prb ↔ Glyc + 2 PHOS
Prb + 2 G6P ↔ Trh + 3 PHOS
F6P + Prb ↔ F16P
F16P ↔ 2 TRIO
PHOS + TRIO + NAD ↔ BPG + NADH
BPG ↔ P3G + Prb
P3G ↔ P2G
P2G ↔ PEP
PEP ↔ Prb + PYR; F16P
PYR ↔ ACE + CO2
2 ACE + 3 NAD ↔ SUCC + 3 NADH
GLCo ↔ GLCi
ACE + NADH ↔ ETOH + NAD
NADH + TRIO ↔ PHOS + GLY + NAD
Prb ↔ PHOS
X ↔ PHOS

Real case study

Metabolite abbreviations

Abbreviation	**Metabolite**
g6p	Glucose-6-phosphate
f6p	Fructose-6-phosphate
fbp	Fructose 1,6-biphosphate
g3p	Glyceraldehydes-3-phosphate
3pg	3-Phosphoglycerate
pep	Phosphoenolpyruvate
pyr	Pyruvate
cit	Citric acid
ogl	Oxoaglutarate
succ	Succinate
fum	Fumarate
mal	Malate

List of reactions

g6p →
g6p ↔
g6p ↔ f6p
f6p → fbp
fbp → g3p
g3p →
fbp ↔ 3pg
3pg →
3pg ↔ pep
pep →
pep → pyr
pyr →
pyr → cit
cit → ogl
ogl ↔
ogl → succ
succ →
succ → fum
fum → mal
mal → cit

Chapter 8

Fusing different omics data sources

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

[2] Bosque, G., Folch-Fortuny, A., Picó, J., Ferrer, A. & Elena, S.F. Topology analysis and visualization of Potyvirus protein-protein interaction network, *BMC Systems Biology* **8**:129 (2014). Highly accessed article.

[6] Folch-Fortuny, A., Bosque, G., Picó, J., Ferrer, A. & Elena, S.F. Fusion of genomic, proteomic and phenotypic data: the case of potyviruses, *Molecular BioSystems* **12**, 253-261 (2016).

8.1 Introduction

Complex networks are widely used nowadays to model systems in many omic sciences, e.g. metabolomics, proteomics, transcriptomics or genomics [230, 231]. The case of PPINs is of special interest. Graphs are the most commonly used tool to visually represent PPINs, the nodes being the proteins of the network, and the edges their interactions. Graph theory [230, 231] is usually applied to extract statistical and topological descriptors from the PPINs as a first step. Then, other graph theory tools, usually applied on social or computer complex networks (e.g. clustering algorithms [232]), are used to identify functional modules within the network.

Since biological activity in organisms usually arises from the association or interaction of several proteins, it is crucial to relate PPINs to a biological function or a phenotype. In this study, the data are obtained from a collection of *Tobacco etch virus* (TEV) single and doule nucleotide substitution mutants. For each of these mutant genotypes, absolute fitness was evaluated in its natural host *Nicotiana tabacum var Xanthi nc* during a single infection cycle [233]. Complementarily, a PPIN inferred from empirical protein-protein interaction (PPI) data from several potyviruses is used to relate the mutations and the organismal fitness.

A mutation in a protein may change (slightly or dramatically) its ability to perform its biological functions correctly. The mutated TEV proteins establish interactions with other viral proteins according to the PPIN of potyviruses. Since viral proteins are multifunctional, and they carry out some of their functions as protein complexes, it is reasonable to assume that a part of the effect of the mutated protein on the fitness is channelled through its PPIs. In other words, mutations affect PPIs, which ultimately affect biological fitness. However, some mutations are much more harmful while others have no fitness effect. The PPIN of Potyvirus adds biological context to the mutation and allows for a deeper analysis of the importance of each protein in the virus' infectious cycle.

Some assumptions are made in the present approach. The main one is that each mutation affects all the PPIs of a mutated protein in the same way. Probably the true modifications are subtler, depending also on other factors. Proteins are highly heterogeneous structures and modifications in different parts of their sequence may have different biological consequences for different interactions. However, the lack of available data and their nature constrained the present study. The problem revolves around two issues. On the one hand, there are protein residues or domains that are much more sensitive to mutations than others. Mutations in some locations, such as the catalytic site of an enzyme, are potentially much more harmful to its function than mutations affecting other domains. Instead of relating mutants and fitness directly, the present approach relates mutants to fitness using proteins and interactions between them as a way to channel those effects and obtain useful information. On the other hand, very scarce information is available for particular interactions. One way to include variability in the influence of a particular mutation on each interaction could be carrying out a docking study. Having structural information of two proteins it would be possible to estimate the influence that any change in their sequences has on a possible docking between them. Unfortunately, none of the TEV proteins have been crystallographically determined so this analysis is not possible yet. Therefore, until no new proteomic information arises, the influence of mutations is spread equally to all the interactions that the mutated protein establishes.

In order to relate mutations, PPIN and fitness, a data integration has to be performed. The problem of relating different sources of data has been widely assessed in systems biology using data fusion. Data fusion can be defined as a statistical procedure to analyse simultaneously different sources of complex data sets [234]. This methodology has been applied to identify genes related to specific diseases [235], to PPINs and gene expression [236], to fuse gene regulatory networks, transcriptional factors and amino acid sequences [237], for metabolic profiling [238] and for biomarker search in proteomics [239]. One of the most used methods in data fusion [238–242] is PLS.

The aim in this thesis is thus to fuse the aforementioned genomic, proteomic and phenotypic data of potyviruses in a single multivariate model to understand the relationships among the different data sources. This way, the objective is to relate mutated proteins, their effect on the PPIN, and the resulting organismal fitness measured under controlled laboratory conditions. Figure 8.1 shows a scheme of the data fusion. In this case, the mutations and the PPIN are the explanatory variable data blocks, and the fitness measured for each mutant take the role of the dependent variable. Finally, a set of functional modules of the PPIN is isolated using the PLS modelling. The purpose of this approach is to gain insight into the molecular interactions that occur during the virus infection more than to construct a robust predictive model.

The rest of the chapter is organised as follows. First, *Potyvirus*, its PPIN network and the data sets, are presented in Sections 8.2-8.4, respectively. Sections 8.5 and 8.6 describe the data fusion approach, giving details about how the data is structured and related. Section 8.7 exploits the biological output of the data fusion, identifying functional modules. Finally, some conclusions are drawn on Section 8.8.

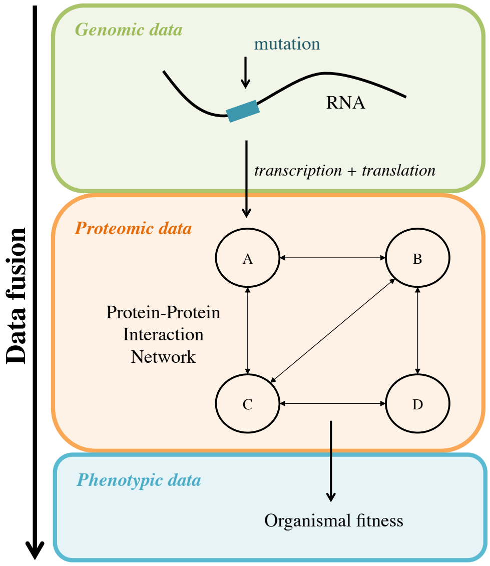


Figure 8.1: Schematic representation of the data fusion.

8.2 Potyvirus and its proteins

Potyvirus is the major genus in the Potyviridae family, accounting for 30% of all known plant viruses, with more than 180 members. Many potyviruses are important pathogens of agricultural crops. They are able to infect a wide range of mono- and dicotyledonous plant species [243], causing symptoms that severely reduce the yield and quality of crops. The economic impact of these viruses on agriculture is well-documented [244]. Some examples of potyviruses are *Plum pox virus* (PPV), *Soybean mosaic virus* (SMV), *Turnip mosaic virus* (TuMV), and *Tobacco etch virus* (TEV) [245].

Potyvirus virions are flexuous and rod-shaped, 680 to 900 nm long and 11 to 15 nm wide [246]. Potyviruses have a single-stranded, positive-sense RNA genome of approximately 10 kilobases (kb). They contain two open reading frameworks (ORF), which after translation, self-process 11 proteins: P1, HC-Pro, P3, 6K1, CI, 6K2, VPg, NIaPro, NIb, CP and P3N-PIPO (more details on these proteins can be found in [247–249]). Much research in the last two decades has focused on understanding the functions of the different potyvirus proteins during the virus life cycle. Rapid rise of academic interest in this topic followed the complete sequencing of the first two potyviruses: TEV [250] and *Tobacco vein mottling virus* (TVMV) [251]. Many excellent reviews have been published since then [243, 246, 252–256].

Table 8.1: Potyvirus interactions initial data set, containing data from 6 different studies and 8 different viruses.

Reference	Virus	Interactions		Method
		Tested	Detected	
[257]	PPV	105	54	BiFC
[258]	SMV-P	100	39	Y2H
	SYSV-O	100	45	Y2H
[259]	PVA	80	16	Y2H
	PSbMV	56	10	Y2H
[260]	PRSV-P	100	16	Y2H
[261]	SMV-G7H	100	9	Y2H
[262]	CIYW	40	5	Y2H

8.3 Protein-Protein Interaction Network (PPIN) reconstruction

All currently available *Potyvirus* PPI datasets are gathered as a first step. These data are obtained from six different articles published over the last decade [257–262]. An overview of the data is shown in Table 8.1. 681 PPIs were tested in these studies and 194 PPIs were detected among the 11 viral proteins from eight different viruses: PPV, SMV-*Pinellia ternate isolate* (SMV-P), *Shallot yellow strip virus-onion isolate* (SYSV-O), *Potato virus A* (PVA), *Pea seed-borne mosaic virus* (PSbMV), SMV-G7H strain and *Clover yellow vein virus* (CYVV).

Integrating data from different sources in a common framework requires standardization. First, each interaction tested in the original studies is collected. Some of these studies were able to test more interactions than others. In some studies it was not possible to produce enough quantity of a certain protein to test its interactions with the others. In other cases proteins had not been yet discovered when the studies took place so they are obviously absent. Additionally, not all interactions tests resulted in a positive interaction being detected.

The molecular methods used in these works to detect the interactions have an inherent directionality. Experimentally, it is common to swap the fused tags among the pair of proteins to avoid possible structural problems that may interfere with the detecting methods (e.g., Y2H and BiFC). Original studies tested all interactions in two directions, for instance P1$\sim$HC-Pro and HC-Pro$\sim$P1. This produces a problem when only one direction was detected. Since the PPI itself has no directionality (it is a molecular docking phenomenon between two molecules) the disagreement comes from the molecular methods used. Some combinations of fused and viral proteins may be less stable or may block the docking of other proteins. To overcome this, it is assumed here that an interaction is valid if it was detected in any of the two directions or in both. This produces symmetry in complementary interactions (P1∼HC-Pro and HC-Pro∼P1) representing the real process of interacting in a clearer and more truthful way.

The next step consists of determining what interactions are relevant and which ones are fair representations of the *Potyvirus* genus topology. Given the variability among studies (e.g., virus species and experimental conditions) it is not surprising that some interactions were detected only in one or few studies, while other were pervasive across the entire dataset. On the other hand, the relative scarcity of the data (only 194 interactions detected) make difficult and somewhat useless a more detailed statistical analysis. Even a confidence interval for each interaction with only eight independent values (corresponding to the eight viruses) is not reliable enough. Therefore, a relevance coefficient (RC) between the numbers of detected and tested interactions for each pair of proteins is defined. It is reasonable to assume that RC is a measure of biological importance. In other words, the more times an interaction has been detected, the higher the probability that this particular interaction is important for the virus to complete its infectious/replication cycle. However, considering the particularities of each method, percentages for Y2H and BiFC are weighted. The latter is closer and much more biologically coherent to natural conditions where potyvirus interactions take place. Therefore, the only study in which this method was used [257] is overweighted. Thus, RC takes the form:

$$RC = 100 \times (2[\mathrm{BiFC}] + [\mathrm{Y2H}])/(T + 1) \tag{8.1}$$

where T is the number of times that a particular interaction was tested (from 0 to 8), [BiFC]is the number of times that a given interaction was detected using the BiFC method (from 0 to 1 because only one study used BiFC) and [Y2H] corresponds to the number of times that an interaction was detected using the Y2H methodology (from 0 to 7). The factor of 2 multiplying the [BiFC] term is a simple way to overweight this method against the Y2H. Doubling its importance was a compromise solution between being truthful to the particularities of each method and still gathering all the relevant information.

RC can range then from 0% (the interaction was not detected in any of the studies) to 100% (was detected in every single study). A threshold for RC is established at the minim value where all nodes are part of a single connected network, which occurrs at RC = 44%. This choice has biological meaning because is based on the fact that all *Potyvirus* genomes encode for the eleven proteins and that all these proteins have been reported to interact at least once with each other. Therefore, it is only possible to study this particular system assuming only one connected network. This threshold is data-dependent and therefore can change from network to network. Even with the same dataset it may be changed to satisfy a particular research objective. For instance, setting a higher RC makes the analysis focus on the most frequent interactions, which may be interesting in a specific situa-

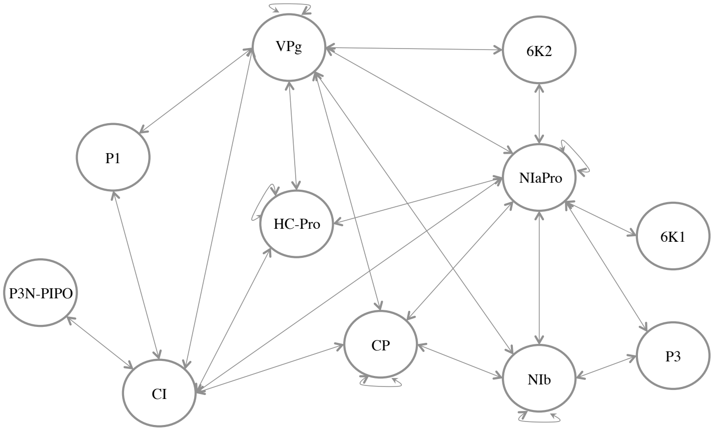


Figure 8.2: PPINs of Potyvirus. Eleven proteins (represented as circles) and their 25 detected interactions (represented as double-arrows).

tion. However, lower RC than 44% results in a disconnected network with various components.

After establishing the aforementioned threshold, only 25 out of the 66 possible interactions between the 11 proteins are considered as relevant. With those interactions the PPIN interaction matrix can be built (see Figure 8.2).

8.4 Mutations and fitness

A collection [233] of 20 TEV single nucleotide substitution mutants and 53 double mutants, resulting from the pairwise combination of the single ones [233], form the dataset analysed here. The fitness of these mutants have been previously quantified by means of growth assays in the natural host *N. tabacum.* Fitness is a measure that captures the ability of a mutant virus to grow and spread through the plant during an infection cycle relative to the ability of the unmutated wild type virus [263].

The collection of mutants was generated at random and thus it is somehow irregular, not affecting all TEV proteins: 6K1, CP and P3N-PIPO were not mutated (see Table 8.2). Moreover, some proteins like P1 and VPg were mutated more times than others such as 6K2, CI and NIb. Although a more complete collection of mutants would be very useful to further increase accuracy, the collection of 73 mutants used in this Chapter is a fair representation of the TEV genome and its 11 proteins.

Mutation	Protein	Type	# of mutants
PC2	P1	Nonsynonymous	2
PC6	P1	Nonsynonymous	7
PC7	P1	Nonsynonymous	5
PC12	P1	Nonsynonymous	4
PC19	HC-Pro	Synonymous	10
PC22	HC-Pro	Nonsynonymous	6
PC26	HC-Pro	Synonymous	4
PC40	P3	Synonymous	5
PC41	P3	Nonsynonymous	4
PC44	P3	Synonymous	5
PC49	CI	Nonsynonymous	8
PC60	CI	Synonymous	3
PC63	6K2	Nonsynonymous	10
PC67	VPg	Nonsynonymous	4
PC69	VPg	Nonsynonymous	13
PC70	VPg	Nonsynonymous	5
PC72	VPg	Nonsynonymous	3
PC76	NIaPro	Synonymous	8
PC83	NIb	Nonsynonymous	10
PC95	NIb	Nonsynonymous	10

Table 8.2: Mutations experimentally generated on the genome of TEV.

The mutant collection has some features that make it an interesting and appropriate starting point for the data fusion. 6 of the 20 single mutants correspond to synonymous mutations. In other words, the nucleotide substitution does not translate in an amino acid replacement in the protein sequence. In spite of being synonymous, some of these mutations have a significant effect on fitness [263] due to RNA stability, enhanced RNA silencing responses or improved translational efficiency, among other possibilities. Although these mutations have no effect on the protein sequence and thus no predictable effect on the PPIN either, they represent a natural source of fitness variability that is taken into account. Other particularity of the data is that lethal mutations exist, meaning those that render zero fitness for the virus bearing them, i.e. these mutations do not allow the virus to survive and grow. Nine of the double mutations are lethal. These mutations are excluded from the analysis because, if included, they will mask all the variability of non-lethal mutations varying fitness in a discrete manner.

The effect of the mutations on the proteins can be quantified using different information. Since none of the TEV proteins has been crystallized yet, to represent the biochemical similarity or the distance between the original amino acid in the sequence and the new one produced by the mutation, an empirical amino acid substitution matrix is used. These matrices describe the rate at which one amino acid changes to any other over time. These matrices are commonly used in the field of protein sequence alignment, calculating the probability that a particular amino acid changes over time to a new one through mutation. The underlying idea is that an amino acid substitution is more likely to survive to the filter of selection if it is similar to the original amino acid than if it is physically very different. Similar amino acids would then preserve a similar folding structure and activity for the protein. Thus, the information contained in the entries of these matrices to quantify the magnitude of each mutation is used.

Since the collection of mutants available is composed of single and double nucleotide mutations it seems appropriate to use the Point Accepted Mutation [264] (PAM) matrix to compute the distances generated by the mutations. These matrices were developed using observed mutations in closely related proteins. Large numbers in the PAM matrix denote substitutions very likely to be removed by purifying natural selection, thus unlikely to persist in the long-term evolutionary time. Since the mutants used for this study have almost identical sequences it seems more precise to use a low number PAM matrix. For this, the PAM2 matrix [264] is selected. It is assumed that mutations with high PAM2 values would induce a strong disruption in the protein structure and, therefore, would have a high probability to negatively affect its biological function.

Each mutation performed in this study gives a value that represents the difference between the substitution of a particular amino acid by itself (meaning no mutation at all) and the new amino acid in the sequence. For instance, mutation PC2 produces an amino acid change between F and C. The matrix establishes a score

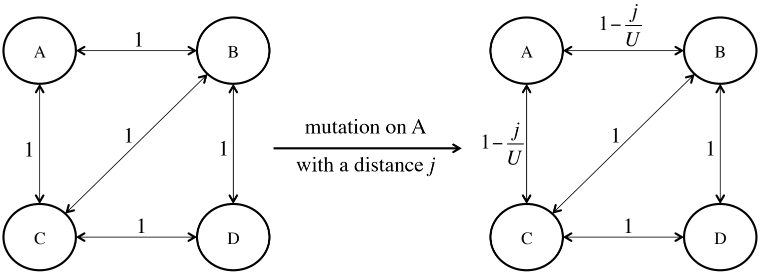


Figure 8.3: Small example of the mutation modelling.

of nine for the F to F substitution (no change) and 30 for the F to C substitution. The difference (39 in this example) between these values represents how similar (chemically and structurally) both amino acids are. Then, the value for all mutations is normalized dividing by the maximum possible value for a change among the 20 amino acids (W to E replacement, with a difference value of 47). Since, in the absence of epistatic interactions, double mutants are potentially twice as harmful as single mutants, in order to compare all mutants (single and double), a normalizing value $2 \times 47 = 94$ is chosen.

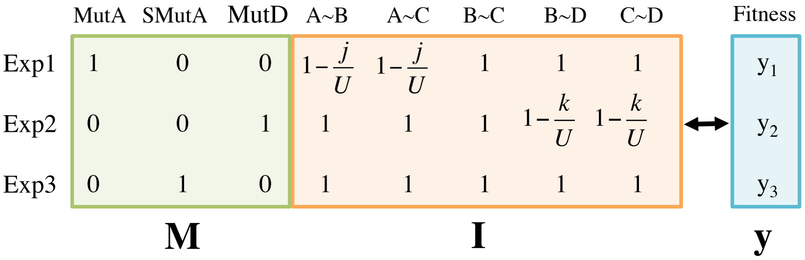
8.5 Mathematical modelling

Once the distance produced by each mutation is computed from the PAM2 matrix, the effect of the mutation on the PPIN has to be modelled. However, as commented previously, some mutations result in a zero distance (synonymous mutations). Since these mutations have no effect on the network, they may directly affect fitness without crossing the PPIN.

The distance registered for all nonsynonymous mutations is modelled as follows. The distance generated by an amino acid replacement, which affects a particular protein, weakens the existing interactions between the influenced one and its first-step neighbours in the PPIN. Figure 8.3 shows a small example of this modelling concept. If a mutation is produced on protein A, with a registered distance j, the interactions relating A to its neighbours, B and C, are weakened as follows:

$$\mathrm{A} \sim \mathrm{B} = \mathrm{A} \sim \mathrm{C} = 1 - \frac{j}{U} \tag{8.2}$$

where A∼B and A∼C mark the interaction between A and B, and A and C, respectively, and U is the aforementeioned reference value (94).



$$\begin{array}{c|ccc|ccccc|c|c}
 & \text{MutA} & \text{SMutA} & \text{MutD} & A\sim B & A\sim C & B\sim C & B\sim D & C\sim D & & \text{Fitness} \\
\text{Exp1} & 1 & 0 & 0 & 1-\frac{j}{U} & 1-\frac{j}{U} & 1 & 1 & 1 & & y_1 \\
\text{Exp2} & 0 & 0 & 1 & 1 & 1 & 1 & 1-\frac{k}{U} & 1-\frac{k}{U} & \leftrightarrow & y_2 \\
\text{Exp3} & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & & y_3 \\
 & & \mathbf{M} & & & & \mathbf{I} & & & & \mathbf{y}
\end{array}$$

Figure 8.4: Data matrices **M**, **I** and vector **y** have the information from the mutations, interactions and fitness, respectively. Three examples are presented. On Exp1 a nonsynonymous mutation is performed on A, with distance j, and fitness y_1. A nonsynonymous mutation on D is performed in Exp2, producing a distance k and fitness y_2. On Exp3 a synonymous mutation is performed in A, producing no distance (and no effect on **I**), and a fitness y_3. The colours correspond to the data sources described in Figure 8.1.

It is worth noting that the distance produced in the protein is a measure of how different is the protein after mutation. Then, this distance is translated into a strength/intensity measure in the network between the protein and its first-step neighbours.

The different data sources presented in this study must be combined properly to be analysed using a latent structure method. Since PLS, in its original form, works with two-way data matrices, the information collected on the previous subsections must be arranged in such a way that each individual (i.e. experiment) is represented by rows, and the different types of variables (i.e. mutations, interactions and fitness) by columns. So three data matrices are built: the mutation matrix **M** has the 20 different mutations as variables, the interaction matrix **I** has the intensity in each of the 25 interactions by columns, and the vector **y** has the fitness registered for each individual. All matrices have 64 rows, corresponding to the non-lethal mutants. Figure 8.4 presents an example of the matrices defined above, following the small PPIN taken as an example in Figure 8.3.

8.6 Statistical modelling

The data matrices built in the previous section could be analysed using different statistical techniques. Considering only mutations and fitness, a design of experiments (DOE) could be performed, but this approach presents some drawbacks here. There are 20 different mutations performed individually or two-by-two, across the original 73 individuals. A model including only mutations and fitness could be fitted using penalized regression (such as Lasso [265] or Elastic Net [266]) to prevent rank deficiency problems. However, it is known that the PPIs affect the fitness, so in the previous approach this effect is not considered.

The other possible approach consists of relating all the interaction strengths/intensities to the fitness, using classical linear regression. The problem is that the mutations are performed on different proteins affecting different interactions, which may not be comparable in this model.

In this work, a PLS regression is applied to fuse the genomic, proteomic and phenotypic data in a single multivariate model, the first two sources being the explanatory variable blocks and the phenotypic fitness of the dependent variable. Using a PLS model, the available data are compressed into a set of LVs that relates mutations and interactions to the observed fitness. This allows us to clarify which mutations, and also which sections of the network, increase or decrease the fitness of TEV.

The different data sources, detailed in previous sections, have to be pre-processed in order to obtain meaningful components in the PLS model. In the present case the dataset is directly autoscaled, i.e. the variables are centred and divided by their standard deviation to have mean 0 and standard deviation 1.

Regarding the statistical modelling, PLS can be strongly (and harmfully) affected by some of the mutants compiled for the present study. As commented above, lethal mutations decrease the fitness straight to zero, while for the non-lethal mutations it oscillates in a small range around the fitness of the wild-type virus. The inclusion of the lethal ones in the study will force the model to explain only the variation between the lethal and non-lethal, pointing simply to the mutations that have been lethal. To avoid this spurious result, and explain equally the positive and negative effect of the mutations and interactions on the fitness of TEV, these lethal genotypes have been removed from the datasets. This relates directly to the way in which mutation severity is quantified. PAM matrices are constructed assuming non-lethal scenarios. Even the most extreme amino acid substitution is quantified as a prerequisite of biological success. Therefore it is sensible to exclude the lethal mutations from the main analysis, since the benchmark chosen to represent mutation magnitude excludes them originally.

Once the data are prepared for the analysis, a PLS model is fitted using the software ProSensus ProMV. To decide how many components extract from the data, the CV criterion using seven groups is selected.

First, a PLS model including all variables is fitted. Later on, a reduced PLS model is obtained by deleting some mutations and interactions that have a very low influence on the fitness. These mutations are PC12, PC67, PC69, and PC72. The PPIs deleted are: HC-Pro$\sim$VPg, VPg$\sim$VPg, VPg$\sim$NIaPro and VPg$\sim$CP. Basi-

Component	R^2X cum. (%)	R^2y cum. (%)	Q^2 cum. (%)
1	11.8	57.6	39.5
2	23.4	70.0	46.7
3	30.1	78.3	56.7

Table 8.3: PLS regression results (reduced model). Cumulative variances in $\mathbf{X} = [\mathbf{M}\ \mathbf{I}]$ and $\mathbf{y}$ explained by the model ($R^2\mathbf{X}$ and $R^2\mathbf{y}$, respectively) and predictive power of the model (Q^2)

cally, these variables have a non statistically significant PLS regression coefficient in the first PLS model (95% of confidence level).

Table 8.3 shows the results of the reduced PLS model. For the analysis, matrices **M** and **I** are merged in a single matrix **X**, including all the variables collected in the study. With a 3-component model, 30.1% of the variability in **X** explains 78.3% of variance in the fitness, **y**, with a predictive ability of 56.7%. It is worth noting that although network topology is definitely a major contributor to the variance of the fitness, there are some other factors that are not included in this particular approach, harming the predictive power of the PLS model. RNA structure stability and codon usage bias are two clear examples of important contributors to fitness, as commented before, that are not included in the analysis.

Figure 8.5 shows the PLS regression coefficients of the variables in the dataset. The red bars mark the statistically significant PPIs and mutations. The relevant ones are chosen based on the 95% jackknife confidence intervals computed for their corresponding PLS regression coefficient. In this way, when the interval does not include zero, the variable has a relevant effect on the fitness, either positive or negative, with a 95% confidence level.

PC22 has a statistically significant negative effect on the resulting fitness of TEV; i.e. when this mutation is generated in the genome, the fitness lowers its value (see Figure 8.5). PC6, PC19, PC63, and PC83 also affect fitness, but in a positive direction. The fitness increases when either of these mutations is present in TEV genome. It is worth noting that a PLS model using only the mutations and the fitness identifies basically the same relevant mutations as the combined mutations-interactions model, but with less explained variance and predictive power in fitness (70.1% and 47.0%, respectively).

The PPIs P1∼CI, P1∼VPg, 6K2∼NIaPro, NIaPro∼NIb, NIb∼NIb, and NIb∼CP have a statistically significant negative effect on the fitness (see Figure 8.5). Bearing in mind the mathematical modelling, when a mutation is performed, the corresponding interactions lower their values. So, the lower is the value of the interaction, the higher is the fitness computed. Alternatively, HC-Pro∼HC-Pro and HC-Pro∼NIaPro have a statistically significant positive effect on the fitness, i.e.

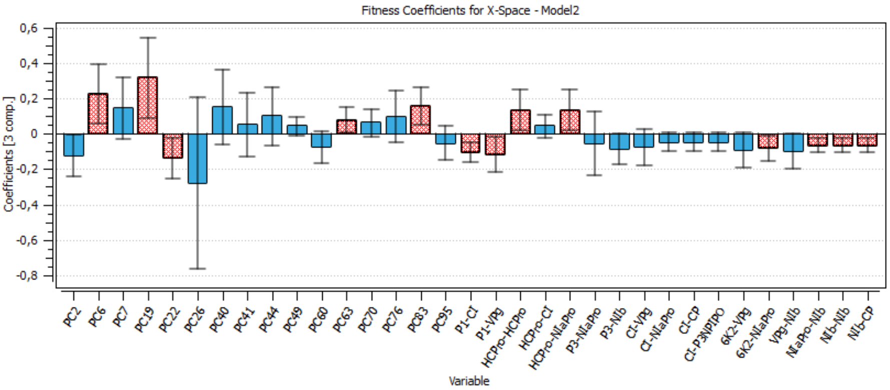


Figure 8.5: PLS regression coefficients with 95% jackknife confidence intervals. The statistically significant variables are plotted as red bars.

Mutation	Protein affected	Interactions
PC6$^+$	P1	P1∼CI$^-$, P1∼VPg$^-$
PC63$^+$	6K2	6K2∼NIaPro$^-$
PC83$^+$	NIb	NIb∼NIaPro$^-$, NIb∼NIb$^-$, NIb∼CP$^-$
PC22$^-$	HC-Pro	HC-Pro∼HC-Pro$^+$, HC-Pro∼NIaPro$^+$
PC19$^+$	HC-Pro	(Synonymous mutation)

Table 8.4: Statistically significant explanatory variables. $^+/^-$ mark the positive/negative effect of the variable on the fitness

the lower is the value of the interaction, the lower is the fitness computed. All the statistically significant variables, mutations and PPIs, are summarized in Table 8.4.

8.7 Functional modules

On the previous section, the explanatory variables, PPIs and mutations with a statistically significant effect on the organismal fitness, are identified among the rest of the variables registered. In order to finally establish the relationships among the three data sources, following the scheme proposed in Figure 8.1, the genomic-proteomic-phenotypic effect must be explained using the information in Table 8.4. If the relevant mutations and PPIs are represented on the original PPIN (see Figure 8.6) some interesting conclusions can be drawn.

Mutation PC6, affecting protein P1, is positively correlated with TEV fitness. At the same time, interactions P1∼VPg and P1∼CI are also relevant in the PLS

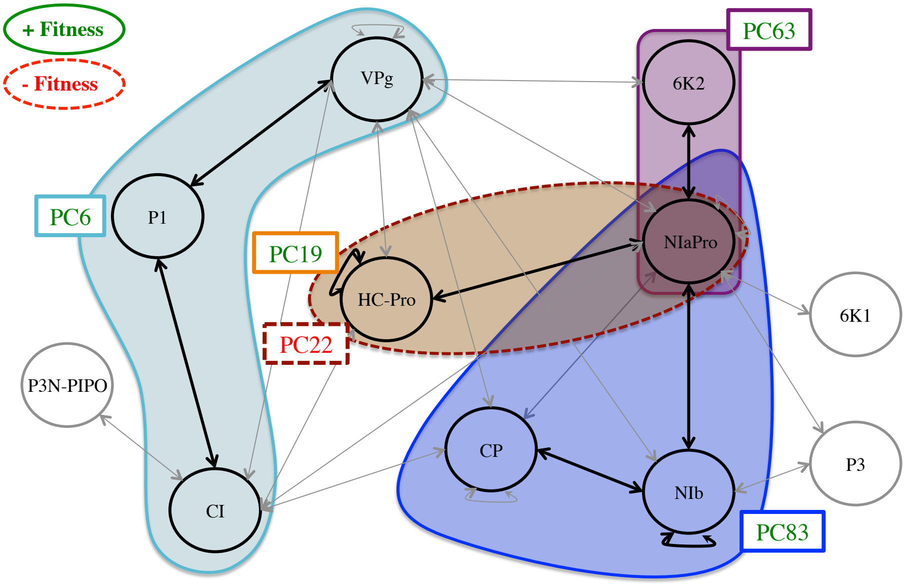


Figure 8.6: Functional modules of TEV.

model, being negatively correlated with viral fitness. These mutation-fitness effects and interaction-fitness effects represent a unified mutation-interaction-fitness effect. Figure 8.7 shows a scheme of this process: when PC6 is generated on P1, the interactions with its neighbours VPg and CI lower their values, and the fitness is increased as a result. A cyan ellipse in Figure 8.6 rounds this functional module.

This behaviour is also observed with the blue and violet modules (see Figure 8.6). The former one is activated via mutation PC83 on protein NIb, and affects NIb, NIaPro and CP. The latter starts with mutation PC63 on 6K2, affecting only its relationship with NIaPro. When these sections are activated, the fitness increases.

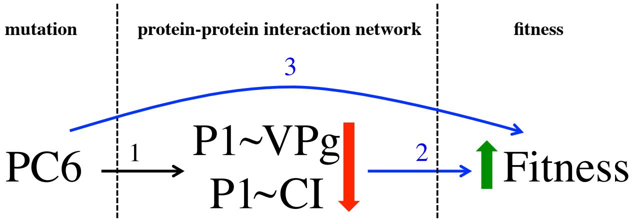


Figure 8.7: Diagram of a mutation-PPIN-fitness positive effect.

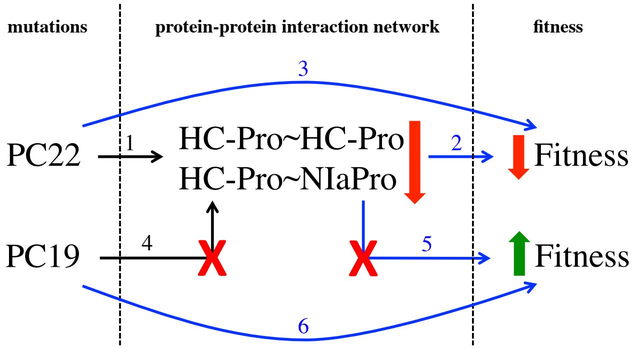


Figure 8.8: Diagram of mutations-PPIN-fitness effects in the case of multifunctional protein HC-Pro.

In this way, Figure 8.7 can also represent the behaviour observed in these modules, replacing the mutation and interaction names.

Two mutations affecting HC-Pro have a statistically significant effect. When mutation PC22 is generated, the PPIs HC-Pro∼HC-Pro and HC-Pro∼NIaPro are affected (brown module in Figure 8.6) and the phenotypic fitness decreases. Alternatively, PC19 is positively correlated with the fitness: when it is introduced in HC-Pro, the fitness increases significantly. Both mutations are compatible with the mathematical modelling because PC19 is a synonymous mutation, and therefore it has no effect on the PPIN network. Figure 8.8 shows the different effects related to HC-Pro. This modelling would be infeasible if PC19 were a nonsynonymous mutation. In this hypothetical case, since it would affect HC-Pro∼HC-Pro and HC-Pro∼NIaPro, it would be incoherent that the mutation increases the fitness and its associated interactions lower its value at the same time.

Two comments are here in due regarding the functional modules (Figure 8.6). Firstly, if an interaction between two proteins is included in a module (e.g. P1∼CI) it implies that the effect of the interaction on the fitness is statistically significant, considering that it can be activated by nonsynonymous mutations performed on both proteins (i.e. P1 and CI). However, the effect is stronger when the mutation defining the module is performed (i.e. PC6 on P1), since the mutation is activating other relevant interactions (i.e. P1∼VPg). Secondly, if an interaction activated by a key mutation is not included in the corresponding module (i.e. interaction 6K2∼VPg, activated via mutation PC63) it implies that the effect of the interaction, considering that it can be activated by nonsynonymous mutations performed on both proteins (i.e. 6K2 and VPg), is not statistically significant.

High-level and mid-level data fusion procedures obtain separate models and extract relevant features of each data matrix, respectively, to combine them in a fused model to predict the biological output [267]. In this Chapter, however, a low-level data fusion is applied, concatenating row-wise, matrices **M** and **I** because the mathematical modelling applied here establishes a direct relationship between the mutations and the PPIN, so the joint analysis of both matrices in a single PLS model leads to identify functional modules exploiting not only the mathematical modelling but also the topological interactions being affected by the different mutations.

8.8 Discussion and conclusions

The PLS modelling applied in this chapter to genomic, proteomic and phenotypic data sets allows integrating the mutations performed on viral proteins, their effects on the PPIN, and their influences on the organismal fitness experimentally quantified. In this way, three biological functional modules affecting the PPIN and influencing the fitness positively have been detected. Two additional modules are identified affecting a single protein. One influences the protein network, being negatively correlated with the organismal fitness. The other one has a positive effect on the fitness without affecting the PPIN. This implies that different mutations affecting the same protein induce different behaviours in the activity of the PPIN and the resulting fitness.

Classical clustering algorithms usually work with a standalone version of the network, detecting dense sections of the topology based solely on its interaction intensities (or basically on node degrees). In comparison to traditional clustering, the presented methodology allows working with different sources of information, combining them to squeeze the data and extract the relevant information. With this data fusion, i) the mutations are related to topological changes in the network and their subsequent influence on the fitness, and ii) the mutations not affecting the network can also be related to the fitness.

Data fusion reveals as a very powerful tool to analyse and relate different types of biological information. The larger the network and the collection of mutants, the more precise its findings are. The present study, analysing a relatively small PPIN (11 nodes and 25 interactions) and a small number of combinations of mutations (64 out of the 210 possible ones), results in a quite high-explained variability. However, there are intrinsic biological considerations that limit the scope of the method. These considerations, such as RNA stability, efficiency inducing the antiviral RNAi response of the plant and codon usage bias may be included in the model as additional sources of variability but much more data would be needed.

Chapter 9

Multivariate image analysis for fruit discrimination

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

[10] Folch-Fortuny, A., Prats-Montalbán, J.M., Cubero, S., Blasco, J. & Ferrer, A. VIS/NIR hyperspectral imaging and N-way PLS-DA models for detection of decay lesions in citrus fruits. *Chemometrics and Intelligent Laboratory Systems* **156**, 241-248 (2016).

9.1 Introduction

In previous chapters, exploratory and predictive models have been fitted to different organisms, mostly at the *micro* level, i.e. yeast, bacteria, viruses. Systems biology aims at understanding the relationships between biological levels within every kind of organism. An extention of this objective, consists of studying how the interaction between organisms can affect each other internally. In this chapter, an organism is analysed at a *macro* level, citrus fruits shortly after harvest, in order to study the effect that produces a *micro* level organism, a fungus.

Citrus production exceeded 115 million tons in 2011 [268]. They are cultivated in over one hundred countries world wide, being Spain one of the most important producer countries and the world leader in fresh citrus exports [268]. Citrus are, indeed, the most widely produced fruits for human consumption, especially oranges (62%) and mandarins (23%). To ensure product quality and reduce production losses, it is mandatory to enhance postharvest handling in food industries, e.g. citrus packinghouses. Many issues arise in this process due to pathological diseases in fruits. This problem can be potentially harmful, since a small set of rotten and sporulated fruits can contaminate the whole batch, especially during storage or transport. *Penicillium digitatum* (the cause of green mould) and *Penicillium italicum* (the cause of blue mould) are two examples of the most deleterious fungi causing fruit decay, and they affect several cultivars over the world [269, 270].

Green mould lesions at early stages cannot be detected with the naked eye because the appearance of the damage is very similar to the appearance of sound fruit. The first symptoms of this disease appear as a slightly discolored soft, water soaked around a point of injury. The spot expands rapidly to a 30-40 mm diameter. As the infection advances, a white fungal growth appears on the surface of the rot [271]. Before the sporulation, the appearance of the lesions is very similar to the sound skin being difficult for the workers to detect damaged fruit, especially when they work on an inspection table, examining fruit traveling at high speed. Therefore, the application of visual inspection or computer vision systems based on colour images is limited. Nowadays, novel machine vision technologies are being incorporated in the citrus postharvest to detect this dangerous disease, mostly based on ultraviolet (UV) induced fluorescence. Ogawa et al. [272] presented a system to detect decay lesions in citrus using fluorescence images, and Blanc et al. [273] patented an automatic machine for in-line decay detection and fruit sorting using UV illumination. However, Momin et al. [274] demonstrated that different cultivars of citrus fruits have different excitation wavelengths to produce UV induced fluorescence in the infected areas, which makes it difficult to create a system valid for all cases based only on this technology. Also, this kind of automatic detection can be potentially jeopardized by fluorescence measurements from other non-related defects [275]. Alternatively, this disease can often be observed using other techniques like image backscattering [276] or hyperspectral imaging (HSI)

[277]. In this sense, different hyperspectral sensors are being investigated to detect non-visible fruit damage [278] like decay lesions in citrus fruits [279].

Using spectral devices, a set of images is obtained at different wavelengths, capturing a huge amount of chemical information. Some works have been focused on reducing the redundant information in this procedure, compressing the high-dimensional original variable space into a low-dimensional one that preserves the main properties of the data. Gómez-Sanchis et al. [280] and Lorente et al. [281] used the features from spectral images of infected fruit as inputs for classification algorithms, in order to improve the discrimination between sound and symptomatic skin. In addition, HSI systems have also been developed to detect other dangerous diseases. Qin et al. [282] used a portable imaging spectrograph to acquire hyperspectral images of red grapefruits affected by canker and other defects. In that work, the spectral images of the different defects were analysed using PCA and spectral information divergence as classification method, detecting 97.6% of infected fruits. In Qin et al. [283], the authors exploited the bands selected using PCA and correlation analysis to obtain a system capable of detecting the canker using ratios of two bands. Afterwards, a system to detect canker lesions in-line was developed by Qin et al. [284]. Also, PCA and band ratios were used by Li et al. [285, 286] to select relevant bands for the detection of this disease among other common defects.

MIA uses a wide number of models and approaches to deal with hyperspectral images [287, 288]. PCA is probably the most used method within MIA (some examples are shown in the previous paragraph), but other two-way methods are commonly used, as PLS or MCR. In some cases, it is convenient or interesting to use three-way models such as NPLS [289] or Tucker [290].

This chapter focuses on developing multivariate models based on hyperspectral images able of discriminating between infected and sound citrus fruits while at the same time reducing as much as possible the number of wavelengths used. For this, NPLS-DA is used to build a LV-based regression model using specific features extracted from a pool of images of different orange and mandarin cultivars collected at the IVIA. This kind of models has been succesfully applied in many research works within fruit industry, e.g. for tomato [291], coffee [292], loquats [293] and apple [294] discrimination. The present study represents an attempt to implementing automatic classification procedures in fruit packinghouses to prevent the storage of infected citrus fruits, which may ultimately rot and sporulate causing contamination of packinghouse facilities and spread of the disease to healthy stored fruit.

The structure of this chapter is as follows. Section 9.2 gives specific details on the data and the image acquisition. In Section 9.3 the data preprocessing, feature extraction and latent variable modelling are described. Section 9.4 shows the

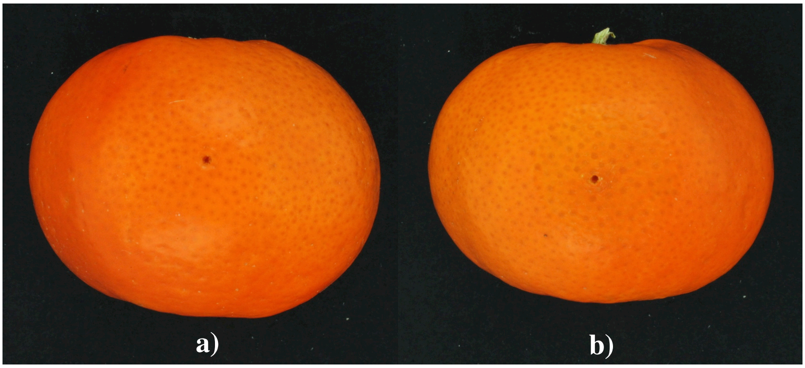


Figure 9.1: RGB images of a control (a) and an infected (b) mandarin.

results of the multivariate discriminant models. Finally, some conclusions are drawn on Section 9A.

9.2 Experiment

Eight different orange and mandarin varietes are analysed here: Clementine, Navel Lane Late, Mioro, Nadorcott, Nova, Salustiana, Blood orange, and Washington Navel. In each variety, 150 fruits were harvested from the field collection of the Citrus Germplasm Bank at the IVIA [295]. After two days of storage with controlled temperature and humidity, 100 fruits of each variety were inoculated with a concentration of 106 spores/ml of *P. Digitatum* [296]. These citrus fruits represent the fungus group. The remaining 50 fruits were inoculated with water, and they represent the control group to know if the innoculation process influences the results. Both inoculations were produced around 2 days after the fruit collection.

Between 1 and 4 days after inoculation, when the fruit started to show slight external symptons of decay, a camera coupled with a VIS/NIR liquid crystal tunable filter (LCTF) was used at IVIA to obtain a hyperspectral image from each fruit of each variety. Figure 9.1 shows the red-green-blue (RGB) images of a control and an infected mandarin, in order to illustrate how difficult is to discriminate between both classes with visual inspection. 44 wavelentghs were registered from 650 to 1080 nm with a resolution of 10 nm. Each image has 1040 times 1392 pixels per wavelentgh. Therefore, the hyperspectral images can be represented as $1040 \times 1392 \times 44$ datacubes.

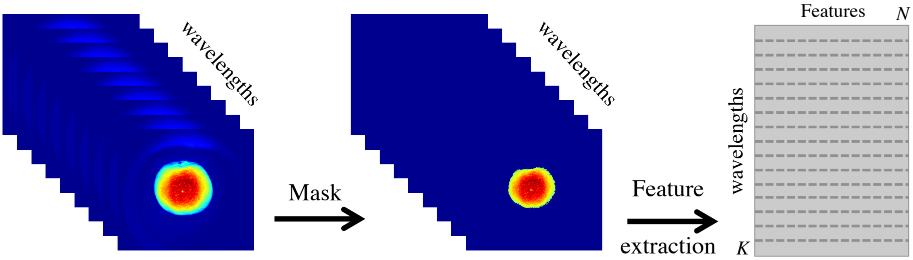


Figure 9.2: Hyperspectral image preprocessing.

9.3 Methodology

9.3.1 Data preprocessing

The citrus fruits appear centered in the images (see Figure 9.2, first block of images). The spherical shape of the fruits causes some undesirable effects in the fruit images, one of the most important being that the pixels in the borders (pale blue areas around the fruit in Figure 9.2) appear darker than those in the centre of the fruit due to the reflexion laws of the light. Therefore, it is convenient to remove the pixels near the border from the analysis, which is done in this experiment by applying a mask. After defining an intensity threshold, pixels exceeding this limit are selected, representing the inner area of the fruit (see Figure 9.2, second block of images). The pixel selection is performed at each wavelength of the image. Then, the joint area across all wavelengths is defined as the mask for the whole image. This way, if a pixel is above the threshold for, at least, one wavelength, it is guaranteed that it is included in the fruit mask. This procedure is repeated for all fruits in each variety.

Five different data preprocessings are applied in this study. The first one consists of analysing the images using the original intensities (no preprocessing), i, measured with the VIS/NIR-LCTF system. The second one consists of transforming the intensities into reflectance values, r, using black (b) and white (w) references taken with the HSI system:

$$r = 100 \times \frac{i - i_b}{i_w - i_b} \tag{9.1}$$

The third preprocessing consisted of obtaining the absorbance values, a, from the reflectance. That is:

$$a = \log_{10}(\frac{r}{100}) \tag{9.2}$$

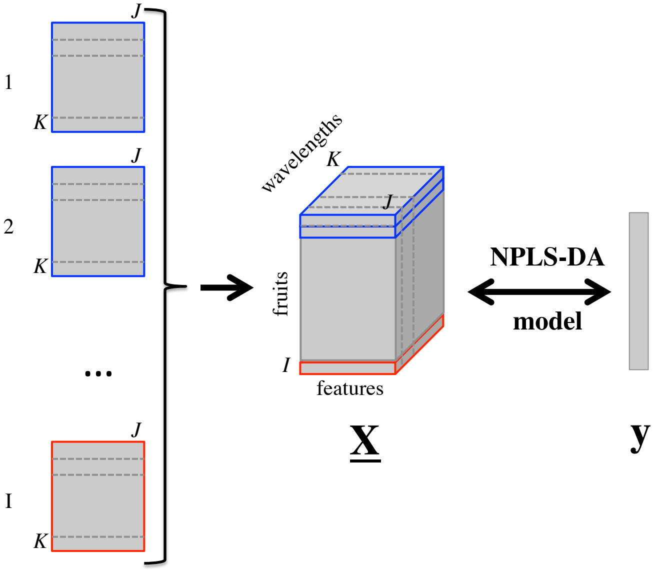


Figure 9.3: NPLS-DA modelling. The feature matrices extracted from each citrus fruit are arranged as row slices of the three-way array $\underline{\mathbf{X}}$. Then, the datacube is used jointly with the dummy variable $\mathbf{y}$, representing the fungus/control group, in the NPLS-DA model.

The fourth and fifth preprocessings consists of applying multiplicative scatter correction (MSC) and standard normal variate (SNV) methods [297] to the absorbance values, respectively. The complete study presented in Section 9.4 is reproduced using the five different preprocessings. A table with the main results is shown in Section 9.6.

9.3.2 Feature extraction

Once the mask is applied, each wavelength image is converted into a one-dimensional numerical array using an image-based approach [288] (see Figure 9.2). In each vector, a set of first order statistics are included as features describing the corresponding wavelength image. Specifically, the mean, standard deviation, and third to fifth order moments are used. After feature extraction, the data are arranged in a 3-way data cube, containing the whole set of fruits in each variety by rows, the features by columns, and the 44 wavelengths as third mode (see Figure 9.3).

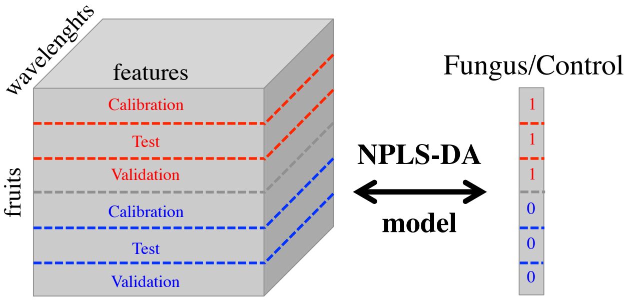


Figure 9.4: Data partition for the 2CV procedure using images of a particular variety.

9.3.3 Discriminant models, validation procedure and wavelength selection

NPLS-DA is applied in this chapter, using N-way Toolbox, to discriminate between fungus and control oranges. Specifically, the $\underline{\mathbf{X}}$ $(I \times J \times K)$ data matrix is the datacube represented in Figure 9.3. Each row slice of $\underline{\mathbf{X}}$ represents the set of features of citrus fruit i (therefore $K = 44$ wavelengths and $J = 5$ features). The dummy variable, $\mathbf{y}$ contains 1s for the fungus citrus fruits and 0s for the control ones.

Proper validation of discriminant models is a subtle issue in chemometrics and systems biology. Here, a 2CV strategy [222] is applied, similarly as in the NPLS-DA step in Chapter 7 (see Section 7.4). Using this procedure, the data from each variety and treatment (fungus/control) are split in three groups with the same number of observations in each group (16 fruits) (see Figure 9.4, using the compact 3-way array $\underline{\mathbf{X}}$). The first group is the calibration set, used to build the NPLS-DA model. The second group is the test set, used for selecting the number of components. And the third group is the validation set, used to evaluate the predictive power of the NPLS-DA.

The ultimate goal of the present study is the creation of an affordable automatic procedure to discriminate between sound and infected fruit in packinghouses. The main drawback of using the VIS/NIR-LCTF system to obtain the spectral information is the relative high price of the equipment. On the other hand, HSI-based systems in general capture a huge amount of data that is sometimes redundant and needs large time to be acquired. Hence, there is a need to reduce the dimensionality of the data by selecting only those important wavelengths that still retain most of the information. Current state of technology allows the development of multispectral cameras capable of working in production lines with three to five charge-coupled device (CCD) sensors that can be customized to capture specific wavelengths. Hence, the goal is to perform a variable selection on the third mode of the data, the spectral bands, to assess whether a few wavelengths (three to five) have enough discriminant power to classify each fruit correctly. Permutation testing is used since it is one of the most used techniques to perform variable selection in PLS-DA [213, 222, 223, 298].

The 2CV and the variable selection are performed as follows:

1. 500 different calibration, test and validation sets are built, including 32 random samples in each group: 16 fungus and 16 control citrus fruits.

2. For each of the 500 group selections:

   (a) The 32 calibration fruits are used to build NPLS-DA models, with number of components ranging from 1 to 25.

   (b) The test set was projected onto each of the 25 models to decide the number of components. The interest was in maximising first the F-score and then the parsimony of the final model. The F-score was calculated using precision, P, and recall, R, of the decay prediction of the NPLS-DA model. Parameters P and R are computed as in Equation 6.5, being now the TP the fungus citrus fruits correctly classified in the model, FP the control fruits classified as fungus, and FN the fungus citrus fruits classified as control. Henceforth, F-score is computed as:

   $$F = \frac{2PR}{P+R} \tag{9.3}$$

   Therefore the F-score is maximum when all the samples are classified correctly, both control and fungus (first criterion). If this is achieved selecting different number of LVs, the lowest number is selected following the principle of parsimony (second criterion). See Figure 9.5 for an example of this selection using Nadorcott variety. The NPLS-DA model built with the calibration fruits and the components selected using the test set is called the "real model".

   (c) The VIP values (variable importance in projection) [242, 299] of the real model were collected. The VIP value of the variable (wavelength) k was computed as:

   $$VIP_k^2 = \frac{K\sum_{a=1}^{A}[(w_{k,a}^K)^2(RSSY_{a-1} - RSSY_a)]}{RSSY_0 - RSSY_A} \tag{9.4}$$

   where $w_{k,a}^K$ is the loading value of the kth variable at the ath component, A is the number of LVs in the NPLS-DA model and $RSSY_a$ is the residual **Y**-sum of squares of the model with a components ($a = 0$ to A).

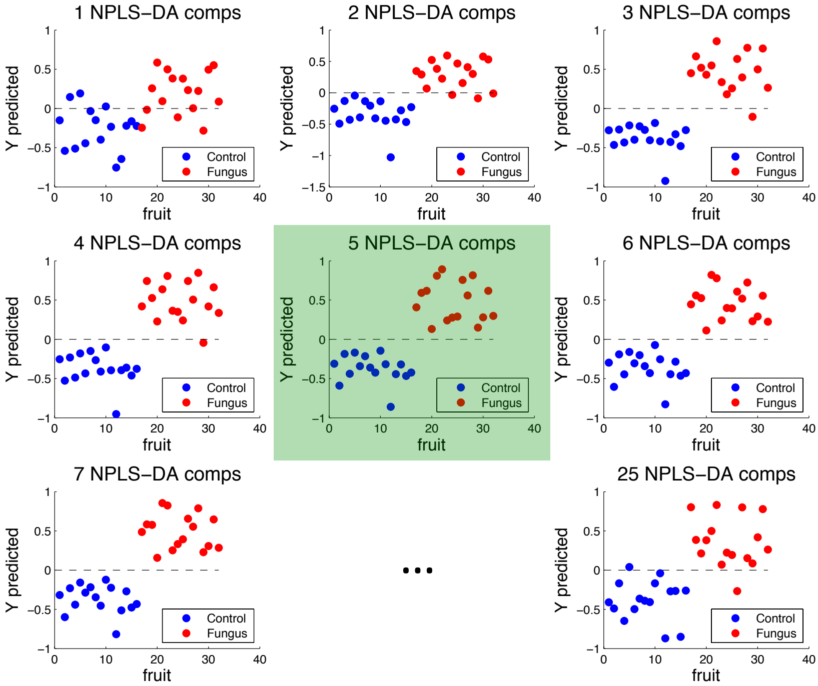


Figure 9.5: Predicted class for the test samples using different number of components in the NPLS-DA model fitted with the calibration data. 5 components are selected, since it is the model with highest F-score and parsimony.

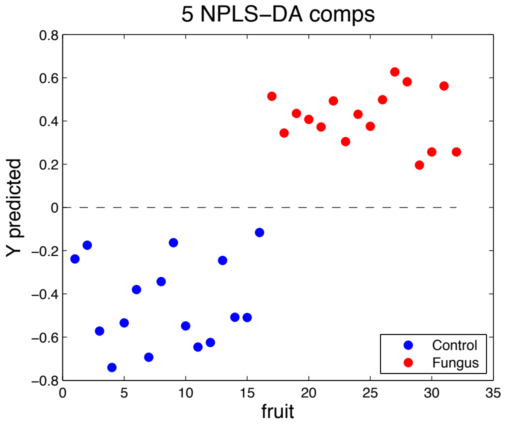


Figure 9.6: Validation samples projected onto the NPLS-DA model with 5 components built with calibration samples.

(d) Steps 2a-2c are then repeated destroying the relationships between $\underline{\mathbf{X}}$ and $\mathbf{y}$, thus creating a random model. This is done by permuting the rows of $\mathbf{y}$ before applying step 2a. Finally, the VIP values using the random model are collected after i) fitting the NPLS-DA model with the calibration set and ii) deciding the number of LVs using the test set.

(e) The remaining validation samples are projected onto the real model to obtain the correct classification rates. Figure 9.6 exemplifies the projection of the validation set in the Nadorcott variety using the model selected in step 2b (see Figure 9.5).

(f) Steps 2a-2e are repeated three times, moving the samples from group to group, that is: calibration-test-validation (first model, as in Figure 9.4), test-validation-calibration (second model), and validation-calibration-test (third model), as performed in Chapter 7 using dynEMR-DA.

(g) The VIP values of both the real and the random models were averaged among the three models.

(h) The results of the external validation using the three real models were integrated.

3. Once step 2 is performed for all group selections, the statistical significance between the real and random models is assessed. The distribution of the random VIP values represents the null distribution, so the real VIP values, or the their mean, m, can be compared with the previous distribution to

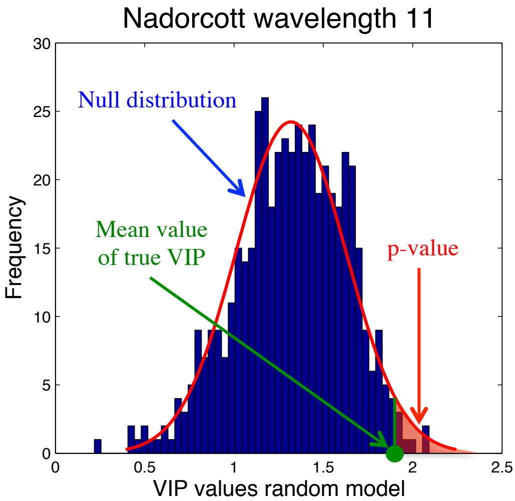


Figure 9.7: VIP values of a particular wavelength. The red line denotes the null distribution from the random models. The green dot represents the mean VIP value of the real models. The red area is the p-value associated to the green dot in the red null distribution.

compute the statistical p-value, that is the probability of obtaining at random a value equal or higher than m. Figure 9.7 shows an example using the VIP values of a particular wavelength in the Nadorcott variety.

4. The mean p-values of each wavelength across all varietes are averaged to obtain the mean p-values. Then, after sorting the p-values, the wavelengths with lower mean p-values are classified as the most discriminant variables in all fruit varieties.

5. The mean correct classification rates are obtained using the results of the validation set in the 500 models.

6. Steps 2a, 2b, 2e, 2f, 2h and 5 (2CV procedure) are repeated using the 3, 4, 5, 10, 15, 20, 25, 30, 35, and 40 most discriminant wavelengths determined in step 4. This way, the degradation of the missclassifications is evaluated in all varieties in terms of the number of wavelengths considered in the NPLS-DA model.

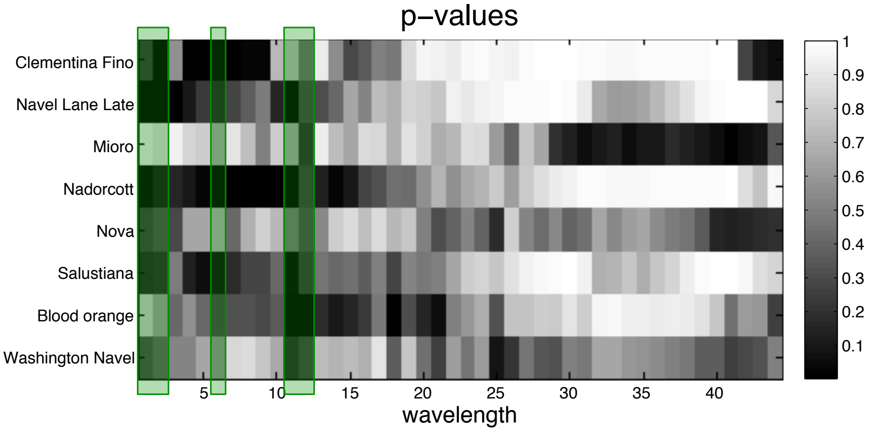


Figure 9.8: P-values computed using the random and the real VIPs. The darker the square, the lower is the corresponding p-value. Green areas mark the 5 best wavelengths attending to the highest mean across varieties: 1, 2, 6, 11 and 12.

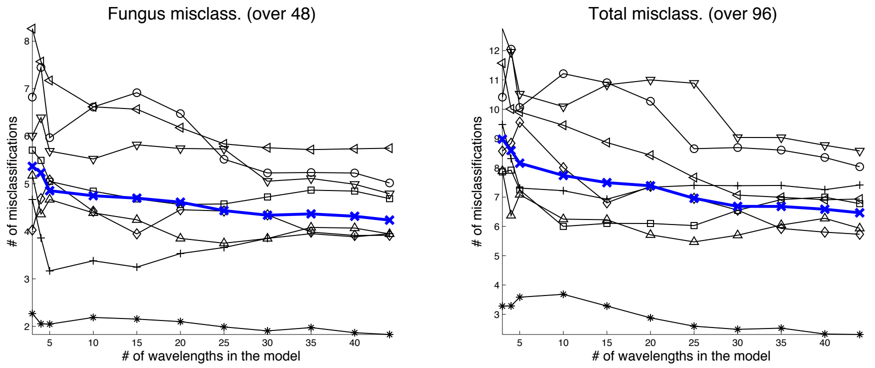
9.4 Results

Figure 9.8 shows the p-values computed using the random and real VIPs. It is clear that different distributions of p-values in the wavelengths are observed among varieties. For example, wavelengths 4-9 have the highest discriminant power (lowest p-values) in Clementine, while the best ones in Mioro are wavelengths 37-43. Despite these differences in the best bands per variety, it seems the initial 15 wavelengths, corresponding to 650-790 nm, tend to have low p-values (high discriminant power).

From a theoretical point of view, the best choice would be to fit different NPLS-DA models (step 6) in each variety including the wavelengths with the smallest p-values in that particular variety. From a practical point of view, this would imply to build different digital cameras incorporating different wavelengths depending on the variety.

Here, a compromise approach is applied, and the wavelengths according to the list of sorted mean p-values obtained in step 4 are selected. The results in terms of fungus and total missclassifications can be visualized in Figure 9.9. The average number of missclassifications in the fungus class decreases notably from 3 to 5 wavelengths, and then the values decrease slowly from 5 to 44 wavelengths. A similar behaviour is shown in the average total number of misclassifications.

The best combination of 5 wavelengths is 1, 2, 6, 11 and 12 (see Figure 9.8), corresponding to 650, 660, 700, 750 and 760 nm. Table 9.1 shows the number of correct classifications and the corresponding percentages using all wavelenghts



a) b)

Figure 9.9: Fungus (a) and total (b) number of missclassifications when varying the number of wavelengths included in the NPLS-DA model. The black lines show the mean over the 500 models of Clementine (upper triangles), Lanelate ('+' symbols), Mioro (circles), Nadorcott (asterisks), Nova (squares), Salustiana (left triangles), Blood orange (diamonds), and Washington Navel (lower triangles). Also, the mean values over all varieties are shown in bold blue lines with crosses.

and only the five most discriminant ones. The variety best discriminated in both cases is Nadorcott, having 99% and 96.8% of correct classification in control and 96.2% and 95.7% in fungus fruit, respectively. The second-best classified, in terms of disease detection, are Navel Lane Late and Clementine, with near 92% of correct classification rate using all wavelengths, and around 93% and 90% using five wavelengths, respectively. Salustiana attains the lowest classification rate. The Appendix 9A shows the results of applying different preprocessings on the original images using all wavelengths, showing that for this variety it is better to use the absorbance values or the absorbance with SNV. For the rest of varieties, it is statistically better to use the intensity values (no preprocessing). The average correct classifications rates using all wavelengths are 95.6% and 91.2% for control and fungus oranges, respectively. When using five wavelenghts the percentages decrease to 93.1% and 90.0%, respetively.

To assess the statistical differences between using 5 or 44 wavelengths, a paired t-test is applied on each variety. The results are presented also in Table 9.1. In general, the results using the selected 5 wavelengths are statistically worse than using all wavelengths. Due to the high sample size (500) used in the paired t-test, small differences in the number of correct classifications become statistically significant. However, comparing the results of both models, the mean loss in correct classification using five wavelengths instead of 44 is 0.8 and 1.1 fruits out of 48 fruits in fungus and control cases, respectively. Anyway, the dramatic reduction

Variety	All wavelengths		Five best wavelengths	
	Control	Fungus	Control	Fungus
	Corr. class. / %	Corr. class. / %	Corr. class. / %	Corr. class. / %
Clementina Fino	46.0* / 95.9%	44.1* / 91.8%	45.6 / 95.0%	43.3 / 90.3%
Navel Lane Late	44.5* / 92.8%	44.1 / 91.8%	43.9 / 91.4%	44.8* / 93.4%
Mioro	45.0* / 93.7	43.0* / 89.6%	43.9 / 91.5%	42.0 / 87.6%
Nadorcott	47.5* / 99.0%	46.2* / 96.2%	46.5 / 96.8%	46.0 / 95.7%
Nova	45.9 / 95.6%	43.3* / 90.2%	45.8 / 95.4%	43.0 / 89.5%
Salustiana	46.8* / 97.5%	42.3* / 88.0%	45.3 / 94.3%	40.8 / 85.1%
Blood orange	46.2* / 96.2%	44.1* / 91.8%	43.5 / 90.7%	42.9 / 89.4%
Washington Navel	44.2* / 92.1%	43.2* / 90.0%	43.2 / 89.9%	42.3 / 88.2%
AVERAGE	45.8 / 95.6%	43.8 / 91.2%	44.7 / 93.1%	43.1 / 90.0%
MINIMUM	44.2 / 92.1%	42.3 / 90.2%	43.2 / 89.9%	40.8 / 85.1%
MAXIMUM	47.5 / 99.0%	46.2 / 96.2%	46.5 / 96.8%	46.0 / 95.7%

Table 9.1: Correct classification results in all orange and mandarin varieties using all wavelengths and only the five most discriminant ones. * denotes a statistically better dicrimination power in the corresponding class (control or fungus) between using all wavelengths or only the five most discriminant ones.

in the price of a 5-channel camera clearly compensates for the small reduction of correct classification.

9.5 Conclusions

NPLS-DA applied on features extracted from a set of hyperspectral images reveals as a powerful tool for discrimination between infected and sound citrus fruits. This way, the methodology applied here captures the effect that the *micro* organism, fungus, produces in the *macro* organism, oranges and mandarins, and how can we detect this effect at early stages of infection. The methodology applied on several orange and mandarin varieties shows that, on average, 91% of fruit with decay lesions caused by *P. digitatum* can be detected at early stages when the damage is barely visible or even invisible and therefore cannot be detected in postharvest by manual inspection. The predictive models were properly validated using a 2CV procedure, computing up to 500 models with different fruit groupings.

Permutation testing on VIP values was used here to select a few spectral channels with the most discriminant power in all citrus fruit varieties. Despite the number of correct classifications becomes stable from five selected wavelengths onwards, there exist statistically significant differences between using five and all wavelengths captured by the VIS/NIR-LCTF system, being the latter significantly better.

Nevertheless, there is a strong cost reduction by selecting a few wavelengths, since a digital camera can be customised to capture up to five filters to reproduce the VIS/NIR-LCTF hyperspectral system. Therefore, from a practical point of view,

	Fungus oranges				
Variety	**Intensity**	**Reflectance**	**Absorbance**	**Abs. + MSC**	**Abs. + SNV**
	Corr. class. / %	**Corr. class. / %**	**Corr. class. / %**	**Corr. class. / %**	**Corr. class. / %**
Clementina Fino	44.3 / 92.3%	43.1^- / 89.7%	41.7^- / 86.9%	41.4^- / 86.2%	42.2^- / 88.0%
Lanelate	43.8 / 91.3%	42.5^- / 88.4%	40.9^- / 85.1%	34.1^- / 71.0%	34.5^- / 71.8%
Mioro Capola	43.0 / 89.7%	43.0 / 89.6%	40.1^- / 83.6%	38.0 / 79.1%	42.7^- / 88.9%
Nadorcott	46.1 / 96.0%	46.0 / 95.8%	45.1^- / 94.0%	41.5^- / 86.5%	41.7^- / 86.8%
Nova	43.5 / 90.7%	42.0^- / 87.5%	40.3^- / 84.0%	39.0^- / 81.3%	41.3^- / 86.1%
Salustiana	42.3 / 88.2%	43.0 / 89.6%	43.4^+ / 90.4%	43.8^+ / 91.2%	42.1 / 87.6%
Sanguina	44.2 / 92.0%	42.8^- / 89.2%	42.0^- / 87.5%	38.2^- / 79.5%	36.9^- / 76.8%
Washington Navel	43.2 / 90.0%	43.0 / 89.6%	41.8^- / 87.1%	35.0^- / 72.9%	38.2^- / 79.5%

Table 9.2: Correct classification rates using different preprocessing (all wavelengths). The +/- superindices mark the statistical superiority/inferiority of the results in the preprocessing compared to the raw intensity values.

the NPLS-DA models including information from the best five wavelengths are sufficient to reduce the losses in fruit warehouses due to storage of infected fruits.

The knowledge obtained in this work is a key step towards the achievement of a potential automatic fruit sorting system using these modified cameras in which fruits are photographed and instantly classified using the predictions from the NPLS-DA model. This way, suspicious fruits can be expelled from the commercial chain prior to affecting sound fruits.

9.6 Appendix. Preprocessings.

The results of the five different preprocessings (intensity, reflectance, absorbance, absorbance + MSC, absorbance + SNV) obtained using 100 models from the 500 used in Section 4, are depicted in Table 9.2. Based on the results of a paired t-test applied between the intensity values and the rest of preprocessings, it is sensible to use the intensity values to fit the models. Only in the case of Salustiana, the results of absorbance and absorbance + SNV were statistically better than the intensity values.

Part III

Missing data

Chapter 10

PCA model building with missing data

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

[4] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. PCA model building with missing data: New proposals and a comparative study. *Chemometrics and Intelligent Laboratory Systems* **146**, 77-88 (2015). #8 in TOP25 from July-September 2015.

10.1 Introduction

Incomplete data sets usually arise when dealing with experimental and process data in chemometrics and systems biology. In experimental environments, practitioners usually deal with 5-20% of missing values. In complex industrial bioprocesses, where hundreds of variables are collected per batch, 30-60% of missing data can appear in their historical data sets. Finally, with the paradigm of big data, thousands of variables are collected for huge sets of individuals, having sometimes more than 70% of missing values in their data sets.

As stated in Section 2.5.1, there are two problems associated to PCA when dealing with missing values: model explotation (ME) and model building (MB). Many methods have been proposed in both ME [27, 51–54] and MB [30, 53, 55–63, 65, 66] environments. The second problem is addressed in this chapter.

The problem of PCA-MB is studied in this chapter. Based on the good performance of the regression-based methods [27, 54] in the ME context, the main goal here is to verify if this is also true in a MB environment. In this work we propose new methods for building PCA models with MD by adapting PCA-ME methods to deal with the more general problem of PCA-MB, when the training set has missing values. The new adapted methods proposed here are PMP, KDR, KDR with PCR, KDR with PLS and TSR. They are compared against established methods (NIPALS, IA, DA and NLP) using four data sets, two simulated and two real ones, with several percentages of missing data. Also, some equivalences are established between the novel approaches and other methods proposed in the literature [62, 63, 65, 66].

This chapter is organized as follows. The new MD algorithms are introduced in Section 10.2. The data sets used as case studies and the comparative study are described in Sections 10.3-10.4. The results of the comparative study are presented in Section 10.5. Finally, Section 10.6 presents the conclusions of the study.

10.2 Methodology

We define the missing data indicator matrix $\mathbf{M}$, associated to a data matrix $\mathbf{X}$ $(N \times K)$, as the binary matrix such that $m_{nk} = 1$ if x_{nk} is missing, and $m_{nk} = 0$ if x_{nk} is known. The matrix $\bar{\mathbf{M}}$ is the complement of $\mathbf{M}$, that is, $\bar{m}_{nk} = 1 - m_{nk}$. And finally, let $\mathbf{Z}$ denote the resulting $\mathbf{X}$ matrix after filling in the unknown values with zeroes, that is, $\mathbf{Z} = \bar{\mathbf{M}} \circ \mathbf{X}$. More details on the notation can be found in Section 2.2.

A common procedure for building a PCA model from $\mathbf{X}$ is the known IA [53], that consists of filling in the missing data with initial values (usually zeros, although other imputations such as the mean of the known values of the corresponding col-

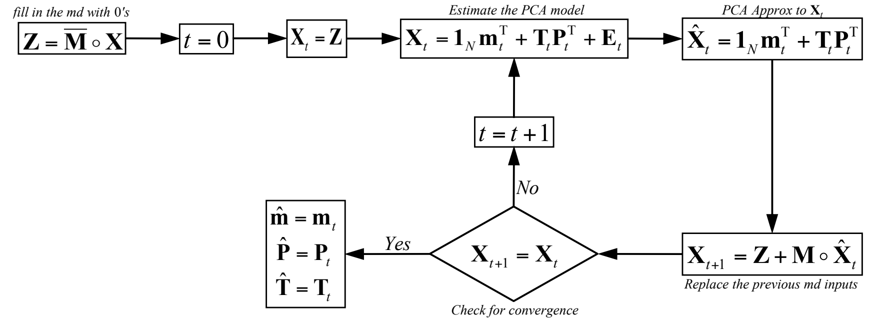

fill in the md with 0's
$\mathbf{Z} = \overline{\mathbf{M}} \circ \mathbf{X}$
$t = 0$
$\mathbf{X}_t = \mathbf{Z}$
Estimate the PCA model
$\mathbf{X}_t = \mathbf{1}_N \mathbf{m}_t^{\mathrm{T}} + \mathbf{T}_t \mathbf{P}_t^{\mathrm{T}} + \mathbf{E}_t$
PCA Approx to $\mathbf{X}_t$
$\hat{\mathbf{X}}_t = \mathbf{1}_N \mathbf{m}_t^{\mathrm{T}} + \mathbf{T}_t \mathbf{P}_t^{\mathrm{T}}$
$t = t + 1$
No
$\hat{\mathbf{m}} = \mathbf{m}_t$
$\hat{\mathbf{P}} = \mathbf{P}_t$
$\hat{\mathbf{T}} = \mathbf{T}_t$
Yes
$\mathbf{X}_{t+1} = \mathbf{X}_t$
Check for convergence
$\mathbf{X}_{t+1} = \mathbf{Z} + \mathbf{M} \circ \hat{\mathbf{X}}_t$
Replace the previous md inputs

Figure 10.1: IA method in PCA-MB with missing data

umn or the mean of the corresponding rows and columns are also used), yielding a reconstructed data set from which a PCA model is fitted. By replacing the original missing data by their predictions from this PCA model, a new reconstructed data set is obtained, and a new PCA model is fitted. This process is iterated until convergence of the predicted values for the missing data, as shown in Figure 10.1.

Walczak and Massart [53] introduce an adaptation of the IA for estimating the scores of new incomplete observation from a pre-built PCA model, fixed and known (i.e. ME). This method has the following structure: i) fill in the missing positions of the new observation with an initial estimate; ii) predict the scores for the filled-in observation, using the loadings matrix $\mathbf{P}$ from the fitted PCA model; iii) re-estimate the missing values by employing the predicted scores and the loadings of the known PCA model; and iv) iterate until convergence.

Consider that the new observation $\mathbf{x}^{\mathrm{T}}$ has some unmeasured variables and these can be taken to be the first R elements of the row vector, without loss of generality. Thus, the vector can be partitioned as $\mathbf{x}^{\mathrm{T}} = [\mathbf{x}^{\#\mathrm{T}}\ \mathbf{x}^{*\mathrm{T}}]$, where $\mathbf{x}^{\#\mathrm{T}}$ denotes the missing measurements and $\mathbf{x}^{*\mathrm{T}}$ the observed variables. This induces the following partition in $\mathbf{X}$: $\mathbf{X} = [\mathbf{X}^{\#}\ \mathbf{X}^{*}]$ where $\mathbf{X}^{\#}$ is the submatrix containing the first R columns of $\mathbf{X}$ (corresponding to the variables that are missing in $\mathbf{x}^{\mathrm{T}}$), and $\mathbf{X}^{*}$ accommodates the remaining $K - R$ columns (corresponding to the observed variables in $\mathbf{x}^{\mathrm{T}}$). Note that $\mathbf{X}^{\#}$ and $\mathbf{X}^{*}$ are different from $\mathbf{X}_{mis}$ and $\mathbf{X}_{obs}$, respectively (see Section 2.5). The first ones are square submatrices of $\mathbf{X}$, according to a reference row. The last ones denote all missing values and available measurements in all rows of $\mathbf{X}$, which may not correspond to a proper square submatrices.

Likewise, the loadings matrix $\mathbf{P}$ can be partitioned as $\mathbf{P}^{\mathrm{T}} = [\mathbf{P}^{\#\mathrm{T}}\ \mathbf{P}^{*\mathrm{T}}]$, where $\mathbf{P}^{\#}$ is the submatrix made up of the first R rows of $\mathbf{P}$, and matrix $\mathbf{P}^{*}$ contains the remaining $K - R$ rows. These induced partitions are illustrated in Figure 10.2.

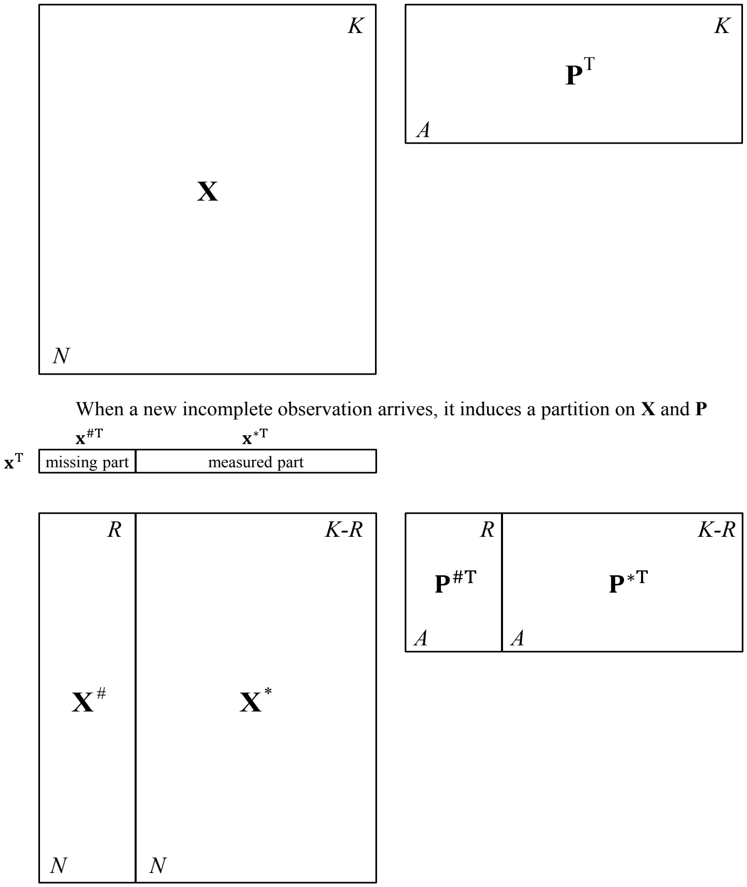

K
X
N
K
P^T
A
When a new incomplete observation arrives, it induces a partition on X and P
x^#T
x^*T
x^T
missing part
measured part
R
K-R
X^#
X^*
N
N
R
K-R
P^#T
P^*T
A
A

Figure 10.2: Data set partition induced by observation $\mathbf{x}^T$.

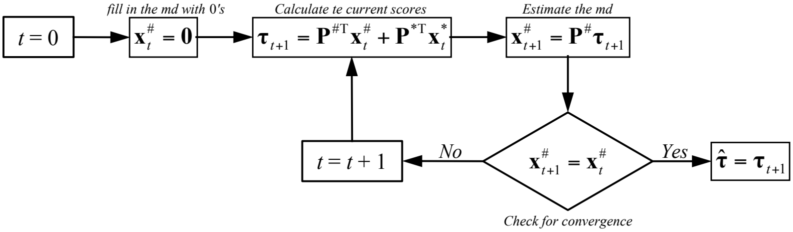

$t=0$
fill in the md with 0's
$\mathbf{x}_t^{\#}=\mathbf{0}$
Calculate te current scores
$\boldsymbol{\tau}_{t+1}=\mathbf{P}^{\#\mathrm{T}}\mathbf{x}_t^{\#}+\mathbf{P}^{*\mathrm{T}}\mathbf{x}_t^{*}$
Estimate the md
$\mathbf{x}_{t+1}^{\#}=\mathbf{P}^{\#}\boldsymbol{\tau}_{t+1}$
$t=t+1$
No
$\mathbf{x}_{t+1}^{\#}=\mathbf{x}_t^{\#}$
Yes
$\hat{\boldsymbol{\tau}}=\boldsymbol{\tau}_{t+1}$
Check for convergence

Figure 10.3: IA method in PCA-ME for a new incomplete observation.

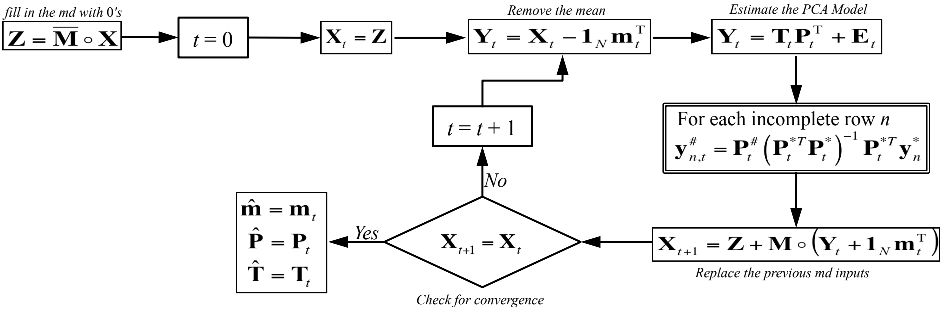

fill in the md with 0's
$\mathbf{Z}=\overline{\mathbf{M}}\circ\mathbf{X}$
$t=0$
$\mathbf{X}_t=\mathbf{Z}$
Remove the mean
$\mathbf{Y}_t=\mathbf{X}_t-\mathbf{1}_N\mathbf{m}_t^{\mathrm{T}}$
Estimate the PCA Model
$\mathbf{Y}_t=\mathbf{T}_t\mathbf{P}_t^{\mathrm{T}}+\mathbf{E}_t$
$t=t+1$
For each incomplete row n
$\mathbf{y}_{n,t}^{\#}=\mathbf{P}_t^{\#}\left(\mathbf{P}_t^{*T}\mathbf{P}_t^{*}\right)^{-1}\mathbf{P}_t^{*T}\mathbf{y}_n^{*}$
No
$\hat{\mathbf{m}}=\mathbf{m}_t$
$\hat{\mathbf{P}}=\mathbf{P}_t$
$\hat{\mathbf{T}}=\mathbf{T}_t$
Yes
$\mathbf{X}_{t+1}=\mathbf{X}_t$
$\mathbf{X}_{t+1}=\mathbf{Z}+\mathbf{M}\circ\left(\mathbf{Y}_t+\mathbf{1}_N\mathbf{m}_t^{\mathrm{T}}\right)$
Replace the previous md inputs
Check for convergence

Figure 10.4: PMP method adapted for PCA-MB with missing data.

Arteaga and Ferrer [27] show that, under general conditions, the IA adaptation from Walczack and Massart [53] is equivalent to the projection to the model plane (PMP) estimator, studied by Nelson et al. [52], that is, the least square estimator based on the observed variables: $\hat{\boldsymbol{\tau}}=(\mathbf{P}^{*\mathrm{T}}\mathbf{P}^{*})^{-1}\mathbf{P}^{*\mathrm{T}}\mathbf{x}^{*}$, where $\hat{\boldsymbol{\tau}}$ is the estimated vector of scores for observation $\mathbf{x}^{\mathrm{T}}$. Figure 10.3 shows a flow diagram of this IA adaptation. Note that the known part of the new individual, $\mathbf{x}^{*\mathrm{T}}$, is assumed to be centred with the mean of the corresponding columns of the $\mathbf{X}$ matrix, $\mathbf{X}^{*}$. Note also that in this adaptation of the IA the PCA model does not change, and thus loadings in matrix $\mathbf{P}$ are fixed. From this equivalence we can state that, in model exploitation, the PMP estimator summarises the iterations of the IA method in one step.

Based on these results, in this chapter we propose to adapt the IA method in PCA-MB from Figure 10.1, by replacing the prediction of missing values from the PCA model to that resulting when we treat each incomplete row in the data set as a new observation with missing values, and apply the PMP method for PCA-ME. This is illustrated in Figure 10.4. The PMP adaptation to PCA-MB uses, in fact, the same regression coefficients for each incomplete observation as the t-EM algorithm [66]. Note that in this case the scores and loadings matrices $\mathbf{T}$ and $\mathbf{P}$, respectively, change at each iteration.

The aforementioned equivalence between PMP and t-EM and other equivalences and similarities between other approaches and methods presented in this chapter are made in Section 10.7.

Arteaga and Ferrer [27] also present two new regression-based methods for estimating the scores of a new incomplete observation: KDR and TSR. The KDR method in PCA-ME, when a new incomplete observation $\mathbf{x}$ is registered, consists of the following steps:

1. Fit the regression model

$$\mathbf{X}^{\#} = \mathbf{X}^{*}\mathbf{B} + \mathbf{U} \tag{10.1}$$

   yielding:

$$\hat{\mathbf{B}} = (\mathbf{X}^{*\mathrm{T}}\mathbf{X}^{*})^{-1}\mathbf{X}^{*\mathrm{T}}\mathbf{X}^{\#} \tag{10.2}$$

2. Estimate the missing part $\mathbf{x}^{\#\mathrm{T}}$ as :

$$\hat{\mathbf{x}}^{\#} = \mathbf{X}^{\#\mathrm{T}}\mathbf{X}^{*}(\mathbf{X}^{*\mathrm{T}}\mathbf{X}^{*})^{-1}\mathbf{x}^{*} = \mathbf{S}^{\#*}(\mathbf{S}^{**})^{-1}\mathbf{x}^{*} \tag{10.3}$$

   where $\mathbf{S}^{**}$ is the estimated covariance matrix of $\mathbf{X}^{*}$, $\mathbf{S}^{**} = (\mathbf{X}^{*\mathrm{T}}\mathbf{X}^{*})/N-1$, and $\mathbf{S}^{\#*}$ is a R by $K-R$ matrix containing the estimated covariances of the combinations of columns of $\mathbf{X}^{\#}$ and columns of $\mathbf{X}^{*}$, $\mathbf{S}^{\#*} = (\mathbf{X}^{\#\mathrm{T}}\mathbf{X}^{*})/N-1$.

In the same conditions, the TSR method can be summarized as:

1. Fit the regression model:

$$\mathbf{X}^{\#} = (\mathbf{X}^{*}\mathbf{P}^{*})\mathbf{B} + \mathbf{U} \tag{10.4}$$

   where $\mathbf{X}^{*}\mathbf{P}^{*}$ is the trimmed scores matrix, i.e. the score matrix that corresponds only to the known variables and their associated loadings (note that $\mathbf{T} = \mathbf{X}\mathbf{P} = \mathbf{X}^{*}\mathbf{P}^{*} + \mathbf{X}^{\#}\mathbf{P}^{\#}$), yielding:

$$\hat{\mathbf{B}} = (\mathbf{P}^{*\mathrm{T}}\mathbf{X}^{*\mathrm{T}}\mathbf{X}^{*}\mathbf{P}^{*})^{-1}\mathbf{P}^{*\mathrm{T}}\mathbf{X}^{*\mathrm{T}}\mathbf{X}^{\#} \tag{10.5}$$

2. Estimate the missing part $\mathbf{x}^{\#\mathrm{T}}$ as :

$$\hat{\mathbf{x}}^{\#} = \mathbf{X}^{\#\mathrm{T}}\mathbf{X}^{*}\mathbf{P}^{*}(\mathbf{P}^{*\mathrm{T}}\mathbf{X}^{*\mathrm{T}}\mathbf{X}^{*}\mathbf{P}^{*})^{-1}\mathbf{P}^{*\mathrm{T}}\mathbf{x}^{*} = \mathbf{S}^{\#*}\mathbf{P}^{*}(\mathbf{P}^{*\mathrm{T}}\mathbf{S}^{**}\mathbf{P}^{*})^{-1}\mathbf{P}^{*\mathrm{T}}\mathbf{x}^{*} \tag{10.6}$$

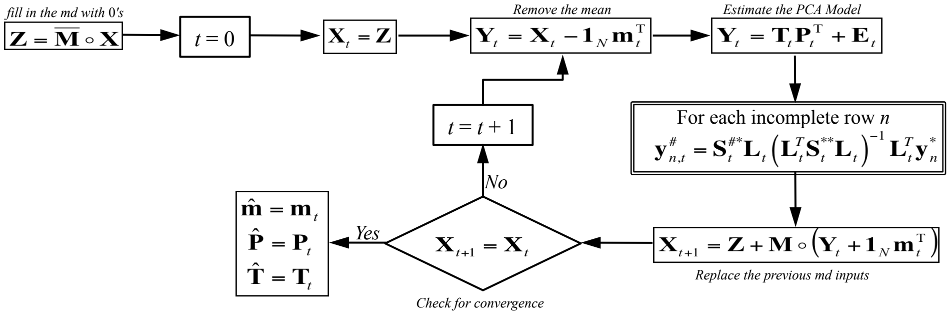

fill in the md with 0's
$\mathbf{Z} = \overline{\mathbf{M}} \circ \mathbf{X}$
$t = 0$
$\mathbf{X}_t = \mathbf{Z}$
Remove the mean
$\mathbf{Y}_t = \mathbf{X}_t - \mathbf{1}_N \mathbf{m}_t^{\mathrm{T}}$
Estimate the PCA Model
$\mathbf{Y}_t = \mathbf{T}_t \mathbf{P}_t^{\mathrm{T}} + \mathbf{E}_t$
For each incomplete row n
$\mathbf{y}_{n,t}^{\#} = \mathbf{S}_t^{\#*} \mathbf{L}_t \left(\mathbf{L}_t^T \mathbf{S}_t^{**} \mathbf{L}_t \right)^{-1} \mathbf{L}_t^T \mathbf{y}_n^*$
$t = t + 1$
No
Yes
$\mathbf{X}_{t+1} = \mathbf{X}_t$
Check for convergence
$\hat{\mathbf{m}} = \mathbf{m}_t$
$\hat{\mathbf{P}} = \mathbf{P}_t$
$\hat{\mathbf{T}} = \mathbf{T}_t$
$\mathbf{X}_{t+1} = \mathbf{Z} + \mathbf{M} \circ \left(\mathbf{Y}_t + \mathbf{1}_N \mathbf{m}_t^{\mathrm{T}} \right)$
Replace the previous md inputs

Figure 10.5: Regression-based framework adapted for PCA-MB with missing data.

Arteaga and Ferrer [54] show that TSR and KDR methods are particular cases of a general framework of methods derived from the generalized regression model $\mathbf{X}^{\#} = (\mathbf{X}^{*}\mathbf{L})\mathbf{B} + \mathbf{U}$, where the key matrix $\mathbf{L}$ takes different expressions depending on which method is applied. The key matrix for the KDR method is the identity matrix, $\mathbf{L} = \mathbf{I}_{K-R}$; in KDR with PCR, $\mathbf{L} = \mathbf{V}_{1:\rho}$, where $\mathbf{V}$ is the eigenvector matrix of $\mathbf{S}^{**}$ and $\rho \leq rank(\mathbf{S}^{**})$; in KDR with PLS, $\mathbf{L} = \mathbf{R}$, where $\mathbf{R}$ is the normalized weights matrix that allows writing the PLS scores $\mathbf{T}_{\mathrm{PLS}}$ as $\mathbf{T}_{\mathrm{PLS}} = \mathbf{X}^{*}\mathbf{R} = \mathbf{X}^{*}\mathbf{W}(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}$ in the PLS model for estimating $\mathbf{X}^{\#}$ from $\mathbf{X}^{*}$. Note that the PLS normalized weights matrix, usually denoted as $\mathbf{W}$ with a star superindex (see Chapter 2) is replaced here by $\mathbf{R}$, to avoid misunderstandings with the available part of the weights. Finally, for the TSR method, $\mathbf{L} = \mathbf{P}^{*}$.

To adapt the PCA-ME framework methods for PCA-MB we only need to substitute, in the IA adaptation (Figure 10.4), the estimation of the missing part of each incomplete observation:

$$\mathbf{y}_{n,t}^{\#} = \mathbf{P}_t^{\#}(\mathbf{P}_t^{*\mathrm{T}}\mathbf{P}_t^{*})^{-1}\mathbf{P}_t^{*\mathrm{T}}\mathbf{y}_n^{*} \tag{10.7}$$

by the expression:

$$\mathbf{y}_{n,t}^{\#} = \mathbf{S}_t^{\#*}\mathbf{L}_t(\mathbf{L}_t^{\mathrm{T}}\mathbf{S}_t^{**}\mathbf{L}_t)^{-1}\mathbf{L}_t^{\mathrm{T}}\mathbf{y}_n^{*} \tag{10.8}$$

at step t. This is illustrated in Figure 10.5.

Assuming data follows a multivariate normal distribution, the adaptation of the KDR method results in the known E-M algorithm [56–58]. The other adapted framework members (i.e. KDR with PCR, KDR with PLS and TSR) are approximations to KDR method that are useful when the covariance matrix $\mathbf{S}^{**}$ is ill-conditioned or singular, because the matrix $\mathbf{L}$ makes $\mathbf{L}^{\mathrm{T}}\mathbf{S}^{**}\mathbf{L}$ to have best conditioning properties than $\mathbf{S}^{**}$.

Note that in all the cited approximations, the key matrix $\mathbf{L}_t$ depends on the missing data combination. This implies that, at each iteration t, two incomplete observations with different missing data combinations require two different $\mathbf{L}_t$ matrices. In KDR and TSR methods this is not a problem because in KDR $\mathbf{L}_t$ is the identity matrix, and in TSR $\mathbf{L}_t$ is $\mathbf{P}^*$, that is, a submatrix of $\mathbf{P}$. Nevertheless, in KDR with PCR an SVD for each missing data combination at each iteration is needed, and in KDR with PLS, a PLS regression for each missing data combination at each iteration has to be fitted. This causes PCR and PLS adaptations to be more computing time demanding than KDR and TSR adaptations. Nevertheless, as commented before, KDR may not be useful in practice due to ill-conditioning or singular problems of the covariance matrix $\mathbf{S}^{**}$.

The previously studied imputation methods impute a unique number for each missing value. These single imputation methods permit to estimate the parameter's values, but ignore the variability of the estimates, leading to underestimation of standard errors and confidence intervals for the estimated parameters. That is, the single value being imputed cannot reflect the sampling variability around the actual value. Multiple imputation [50, 58] overcomes this disadvantage. Multiple imputation, basically, creates several (M) values for each missing value representing a distribution capable of reflecting the sampling variability. Then, we have M complete data sets that we can analyse with the standard statistical techniques to estimate their parameters of interest. This allows calculation of variances of the parameters by combining the variability of estimates from within each imputed data set with the variability of the estimates across M imputed data sets. Rubin [50] shows how to combine both sources of variability in order to obtain confidence intervals for the estimated parameters.

Multiple imputation is based on three main assumptions: a probability model on complete data (observed and missing values), a prior distribution reflecting the uncertainty of the parameters for the imputation model, and that the missing data mechanism is ignorable (i.e. MAR or MCAR).

Multiple imputation can be made in several manners, but the most popular is the data augmentation (DA) algorithm [59]. This is an iterative process that alternatively fills in the missing data and makes inferences about the unknown parameters, but unlike the E-M algorithm, this is made in a stochastic or random fashion.

DA first performs a random imputation of missing data under assumed values of the parameters, and then draws new parameters from a Bayesian posterior distribution based on the observed and imputed data. DA starts with some value of the set of parameters $\boldsymbol{\Theta}$, usually these initial estimates are obtained with the E-M algorithm, and in iteration t alternates between two steps:

- I step (Imputation), draws $\mathbf{X}_t^{\#}$ from their conditional distribution given $\mathbf{X}^*$ and $\mathbf{\Theta}_{t-1}$.
- P step (Posterior), draws $\mathbf{\Theta}_t$ from their posterior distribution given $\mathbf{X}^*$ and $\mathbf{X}_t^{\#}$.

The procedure of alternatively simulating missing data and parameters creates a Markov chain that eventually stabilizes or converges in distribution [57]. In practice, the length of the Markov chain should be long enough to assure stability (convergence) and thus no dependency on initial values. The E-M algorithm is recommended in practice to provide initial estimates of model parameters $\mathbf{\Theta}$ and number of iterations that DA algorithm needs to converge.

The convergence in the E-M algorithm to a single set of values can be easily assessed by checking the change in the parameter estimations from one iteration to the next. For DA, the algorithm converges to a probability distribution, not a single set of values. This makes it rather difficult to determine whether convergence has, in fact, been achieved [58]. The rate of convergence of the E-M algorithm is a useful indication of the rate of convergence for DA. A good rule of thumb is that the number of iterations for DA should be at least as large as the number of iterations required for E-M and then it is useful to run the E-M algorithm before DA.

Finally, López-Negrete de la Fuente et al. [60] present a new approach for PCA-MB with incomplete data sets. This methodology solves a nonlinear programming problem (NLP) (see Equation 10.9) using the IPOPT solver [300].

$$Min\|\bar{\mathbf{M}} \circ (\mathbf{X} - \mathbf{T}\mathbf{P}^{\mathrm{T}})\|_F^2 \quad s.t. \begin{cases} \mathbf{p}_a^{\mathrm{T}}\mathbf{p}_b = \delta_{a,b} \quad a,b = 1,\ldots,A \\ \mathbf{t}_a^{\mathrm{T}}\mathbf{t}_b = 0 \quad a \neq b \quad a,b = 1,\ldots,A \\ \mathbf{1}^{\mathrm{T}}\mathbf{t}_a = 0 \quad a = 1,\ldots,A \end{cases} \tag{10.9}$$

where $\delta_{a,b}$ is the Kronecker delta and A is the number of PCs extracted by the PCA model. The objective function minimises the squared error between the known values and their estimations from the PCA model, subject to the constraints defined by the PCA assumptions: the loadings have to be orthonormal, and the scores have to be orthogonal with zero mean.

10.3 Data sets

The first data set analysed here consists of the percentage composition of eight fatty acids: palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic and eicosenoic, found in the lipid fraction of 572 Italian olive oils. In the original data set [301] there are nine collection areas from three different regions of Italy. In order to reduce the computation time, 75 randomly chosen wines from South Apulia are included in the dataset analysed here. One PC is extracted from these data, explaining 59% of variance.

The second data set is obtained from the Eigenvector Research Inc. data library (`http://www.eigenvector.com/data/index.html`). The data set contains the NIR spectra of several diesel fuels ($N = 40$) obtained at the Southwest Research Institute (SWRI) on a project sponsored by the U.S. Army [302]. The fuels were originally scanned from the wavelength 750 nm to 1550 nm in 2 nm increments ($K = 401$ variables). Two PCs are extracted explaining 60% and 25% of variance in data, respectively.

The third case study is a three-component multivariate data set is generated [303, 304] to compare the performance of the different methods. This data set has ten variables ($K = 10$) and a hundred samples ($N = 100$), and follows a multivariate normal distribution with zero means and unit variances. The highest eigenvalue of the correlation matrix is 4.0, the second one 3.0 and the last one 2.0, explaining 40%, 30% and 20% of the variance from the ten variables, respectively.

Finally, an additional data set, from the big data perspective [305], is analysed in this chapter. Using again [303, 304] a simulated data set is generated with 10 million data entries (100000 observations × 100 variables). 4 PCs are used to generate the data, explaining 35%, 25%, 15% and 15% of the variance, respectively. In this dataset, the methods are forced to deal with even more missing values than in the previous datasets. Here, 80% and 90% of missing values are also simulated.

10.4 Comparative study

In this section, the performance of different methods for PCA-MB with MD are compared. The methods under study are the standard methods for PCA-MB: NIPALS and IA, and the proposed PCA-ME adapted methods: PMP and the regression-based framework adapted methods (KDR, PCR, PLS and TSR). KDR with PCR and KDR with PLS are referred in this section simply as PCR and PLS, to ease the reading.

In order to improve the convergence properties, in the implementation of the above methods, the pseudoinverse is used to calculate $(\mathbf{P}_t^{*\mathrm{T}}\mathbf{P}_t^{*})^{-1}$ in PMP, and $(\mathbf{L}_t^{\mathrm{T}}\mathbf{S}_t^{**}\mathbf{L}_t)^{-1}$ in the framework methods.

The MATLAB implementations for PMP, KDR, PCR, PLS and TSR have been developed for this thesis. The codes for IA and DA have been reproduced from the original papers (see [53] for IA and [57, 59] for DA). For the NLP method, the implementation in the Phi toolbox (version 1.7) [60] is used.

On the following subsections four data sets with missing values are analysed. Two of them are simulated and the other two are taken from the literature. The strategy to generate the MD is the same in all of them. Nine incremental levels of MD are considered in each data matrix (5%, 10%, 15%, 20%, 30%, 40%, 50%, 60% and 70%). 80% and 90% of missing values are also included in the last example. And for each data set and percentage, 50 possible data sets are simulated, following missing completely at random (MCAR) mechanism [47].

The principal performance criterion for each method is the mean squared prediction error MSPE (Equation 10.10).

$$MSPE(Method) = \frac{\sum_{n=1}^{N}\sum_{k=1}^{K}(\hat{\mathbf{x}}_{nk} - \hat{\mathbf{x}}_{nk}^{Method})^2}{NK} \tag{10.10}$$

where $\hat{\mathbf{x}}_{nk}$ is the predicted value for the kth variable of the nth observation in the prediction matrix $\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}^{\mathrm{T}}$ obtained from the complete data set; and $\hat{\mathbf{x}}_{nk}^{Method}$ the analogous prediction obtained after applying the corresponding method on the incomplete data set.

In order to assess whether the differences among methods, in terms of MSPE, are statistically significant, a mixed-effect ANOVA model is fitted per each case study. The factors considered are method, percentage of missing values, and simulated data set, being the latter nested to the percentage factor. Method and percentage are fixed-effect factors; the data set is a random-effect factor. Given the positive skewness of MSPE, a logarithmic transformation is used. This transformation also expands the differences for low percentages of MD, easing the visualization of the plots. In case any effect or interaction is statistically significant, the 95% LSD (least significance difference) intervals are calculated to assess which groups are different from others.

In order to understand the degradation in the PCA model due to missing values the cosine between each loading vector obtained using the full data matrix and its corresponding from the incomplete data set is calculated. The closer to one it is, the more similar are both loadings for a particular component. However, if more than one PC is extracted from data, the cosines of further PCs are being strongly influenced by the previous ones, since they have to be orthogonal to the estimated first PC.

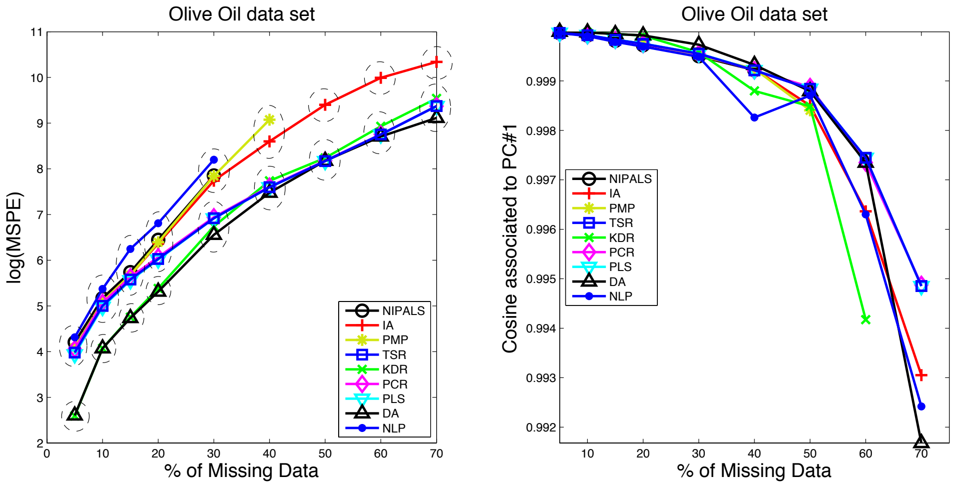


Figure 10.6: Olive Oil data set. log(MSPE) (left) and cosines associated to the first PC (right) for the reconstructed data from a PCA model with different methods: NIPALS, IA, PMP, TSR, KDR, PCR, PLS, DA and NLP. The missing values in the figure correspond to NaNs (which implies convergence problems for the method). Some other higher values are not shown (especially for the highest %) in order to appreciate the differences among the most accurate methods. Dashed ellipses mark the statistically significant differences between groups of methods, i.e. the differences exist between-groups, not within-groups.

10.5 Results

10.5.1 Olive Oil data set

As shown in Figure 10.6, DA and KDR are statistically superior to all other methods from 5-20% of missing data. There exist no significant differences between DA, KDR, PCR, PLS and TSR for further percentages. From 30% of MD onwards, NLP, PMP, IA and NIPALS perform statistically worse than the other methods. From 40% upwards, NLP is unstable for some combinations of missing values, i.e. some missing values are poorly imputed, and therefore the MSPE value and the cosine are strongly affected by them. NIPALS and PMP are unable to converge for high percentages of missing values (60-70%). Some MSPEs are not shown on Figure 10.6 up to some percentage, e.g. NLP's, in order to appreciate the differences among the most accurate methods. Figure 10.6 also shows that the degradation in cosine associated to the first PC matches the increment in MSPE.

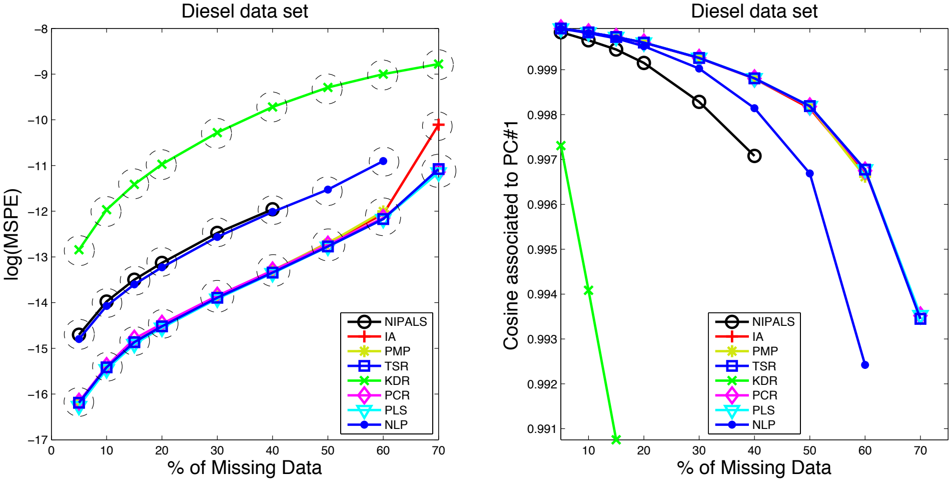


Figure 10.7: Diesel data set. log(MSPE) (left) and cosines associated to the first PC (right) for each method. DA is not applicable in this data set. See Figure 10.6 caption for more details.

10.5.2 Diesel data set

DA is the only method that is unable to analyse the present data set, regardless the percentage of missing values. The main reason is the singularity of the $\mathbf{S}^{**}$ matrix for the different combinations of missing values. This also affects the KDR method, which is statistically the worst in terms of MSPE (see Figure 10.7).

NLP and NIPALS offer better results than KDR, but they are statistically worse than the other methods for all percentages of missing values. NIPALS does not converge with 70% of MD, and neither does NLP. TSR, PCR, PLS, PMP and IA show the best performances for 10-60% of missing data. For 70% of MD, IA and PMP are statistically worse than the previous regression-based methods. These results are coherent with the degradation of the cosine of the first and second loadings (see Figure 10.7).

10.5.3 Simulated data set

In the simulated data set, PMP, NLP and NIPALS have again problems with the convergence (see Figure 10.8), the first one does not converge with 40% of MD, the second one with 50%, and the last one with 60%. Now, at early stages, there exist statistically significant differences between methods. KDR, jointly with DA, is statistically superior to all other methods with 5%-15% of missing values, being NIPALS the worst method. For medium and high percentages of MD, DA and

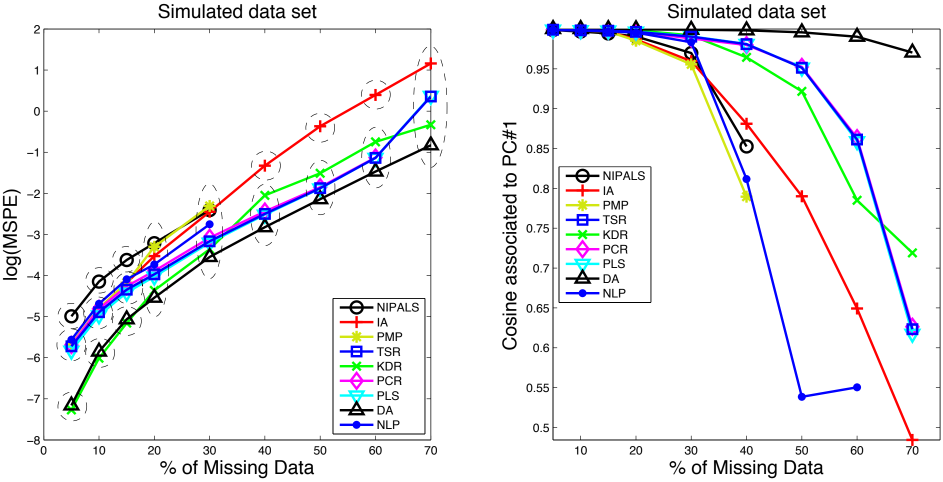


Figure 10.8: Simulated data set. log(MSPE) (left) and cosines associated to the first PC (right) for each method. See Figure 10.6 caption for more details.

KDR are not significantly superior to PCR, PLS and TSR. The performance of IA in this data set is coherent with the previous data sets, i.e. with low percentages of missing data its performance is similar to the regression-based methods, for higher percentages IA performs significantly worse. Again, the cosine degradations (Figure 10.8) agree with the results observed in MSPE.

A comment regarding the cosine values is here in due. A value of 0.9 implies a deviation of 25 degrees between the imputed and the actual PC, which is, in fact, a huge rotation of the basis of the PCs. However, even when the cosines are below 0.9, the imputations of the best methods are still useful, based on their MSPE values.

10.5.4 Big data set

Based on the results of the previous subsections, methods having problems with convergence and/or instability, such as NIPALS, PMP and NLP, are not used for the big data set. IA is applied here, since it is a fast imputation method and showed no problems with convergence in the previous data sets. DA was initially applied but since this method, jointly with KDR, PCR and PLS, are more time consuming than the rest of methods the imputation is not obtained in a reasonable time period. Therefore, only IA and TSR are applied to this data set.

The MSPE results and the 1st PC cosines are depicted in Figure 10.9. TSR offers a statistically significant lower MSPE than IA for all simulations; however, this

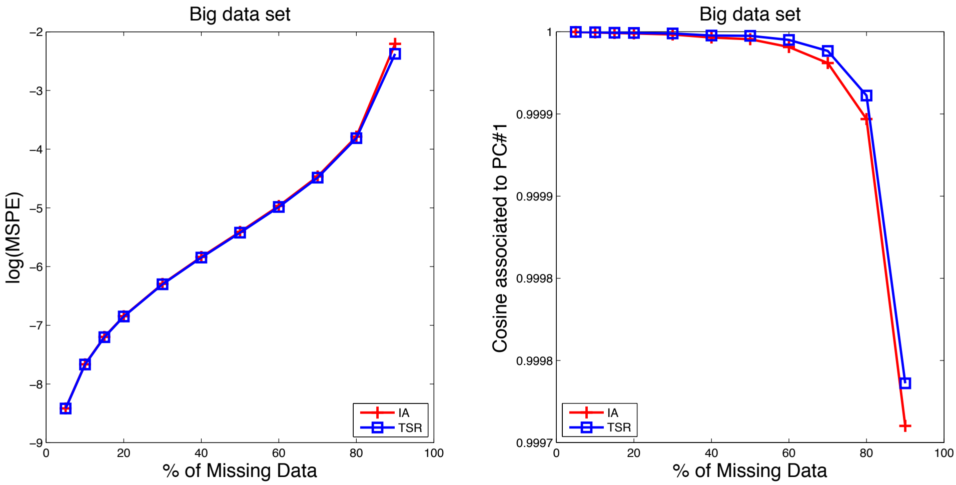


Figure 10.9: Big simulated data set. log(MSPE) (left) and cosines associated to the first PC (right) for each method. See Figure 10.6 caption for more details.

cannot be appreciated in Figure 10.9. It is interesting that the results with this huge data set are better than in the previous simulated one. This is due to the more individuals has the dataset, the more accurate are the estimations of the covariance matrices that TSR and IA perform internally, which implies that the estimation is more coherent that with less observations. It is worth noting that the imputation in the most extreme case, 90% of missing data, is indeed difficult, taking around an hour.

10.6 Discussion and conclusions

TSR method performed extraordinarily well in all the data structures and missing data percentages analysed throughout this chapter. This MD imputation method, adapted here from the PCA-ME context to MB, represents the best compromise solution among prediction quality, robustness against data structure and computation time. From the other regression-based methods adapted here, the KDR methods with PCR and PLS offer also good solutions, however, they are more time-consuming, since they fit additional PCR and PLS models.

From the rest of the methods analysed here, DA and KDR have excellent performances with thin data sets (i.e. more observations than variables). Nevertheless, they have two important drawbacks. The first one is that both methods, especially DA, since it is a multiple imputation method, are strongly more time consuming than, e.g. TSR. The second drawback is that with fat data sets (i.e. more variables than observations, typical in batch processes or with spectral data) DA is unfeasible, and KDR has the worst performance among the rest of the methods.

The NIPALS method for MD imputation, a procedure implemented in many commercial statistical packages such as ProMV, SIMCA-P [306] and PLS Toolbox, is unable to deal with most of the missing data scenarios analysed in this study. Regarding the rest of the methods, PMP and NLP have also convergence problems when high percentages of missing data are generated in all datasets. IA is the only method, jointly with TSR, applied to the big data set, due to its fast performance in the previous data sets and its robustness against high percentages of missing data. However, its performance level in all four data sets is statistically worse than TSR's.

10.7 Appendix. Methods equivalences.

In this section, the equivalence between some methods described in [61] and the ones presented here are proven. The GIP method [61, 63] is equivalent to IA [53] extracting one principal component. The GIP steps are the following:

1. An initial guess for the missing data is imputed.

2. The correlation matrix $\mathbf{R}$ is obtained using the available data, and then the largest eigenvalue λ, and its associated eigenvector $\mathbf{v}$, is obtained. In this way, the first principal component score for sample n is $\tau_n = \sum_{k=1}^{K} v_k x_{nk}$.

3. The missing elements are replaced by their projection using the score, i.e. if $\bar{m}_{nk} = 0$, then $x_{nk} = \tau_n v_k$.

4. Steps 3-4 are repeated until the consecutive imputed values are within the specified tolerance.

The PC scores are calculated in the same way as the estimation of the PCA model in IA (box 4 in Figure 10.1). And the projection for the missing values is basically the PCA approximation to $\mathbf{X}_t$ matrix (box 5 in Figure 10.1). Finally, the process is iterated using the same condition as in IA. Therefore, GIP method is mathematically equivalent to IA when one principal component is extracted.

The main difference between the so-called MICE method defined in [61, 64] and KDR is that the former works variable-wise and the latter observation-wise. The steps of MICE algorithm are detailed here:

1. Initial guesses for all missing elements are provided.

2. For each variable with missing elements, $\mathbf{x}_k$, the data are split into two sub-vectors: $\mathbf{x}_k^*$ a sub-vector that contains all available data, and $\mathbf{x}_k^{\#}$ a sub-vector

that contains all missing data. The available sub-vector $\mathbf{x}_k^*$ is regressed on all other variables, which are restricted to the samples in $\mathbf{x}_k^*$; that is $\mathbf{x}_k^* = f(\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_K)$.

3. The missing sub-vector $\mathbf{x}_k^{\#}$ is then predicted from the regression and its missing entries are replaced with the predictions from the regression. The regression procedure is repeated for all variables with missing elements.

4. After all missing elements are imputed, the regressions and predictions are repeated until consecutive iterates are within the specified tolerance for each of the imputed values.

In MICE the regression model is performed within each column, predicting the observations with the missing values from the observations with available data. KDR follows the same algorithm as MICE, but the data is split based on the missing and available values of an observation (see Figure 10.5, taking $\mathbf{L}$ as the identity matrix). Then, the calibration is performed between the submatrix of $\mathbf{X}$ corresponding to the missing elements in row n, $\mathbf{X}^{\#}$, and the submatrix of available measurements, $\mathbf{X}^*$, following the expression $\mathbf{X}^{\#} = \mathbf{X}^*\mathbf{B} + \mathbf{U}$, where $\mathbf{B}$ is the regression coefficients matrix and $\mathbf{U}$ is the residual matrix. The prediction step is also performed observation-wise: using the model between the missing and the available submatrices, the missing elements in row n are predicted based on its available measurements (see Figure 10.5). The same notation, including $\mathbf{X}^*$ and $\mathbf{X}^{\#}$ partitions could be used in the case of MICE, bearing in mind that this partition is performed taking a variable as reference.

Finally, the equivalence between the regularised t-EM method [61, 66] and PMP [52] is drawn. t-EM algorithm is defined as follows:

1. Estimate the covariance matrix, $\hat{\mathbf{S}}$.

2. Calculate the singular value decomposition of the covariance matrix $\hat{\mathbf{S}} = \mathbf{V}\mathbf{D}\mathbf{V}^{\mathrm{T}}$.

3. Build the regression model:

$$\hat{\mathbf{x}}^{\#\mathrm{T}} = \hat{\mathbf{m}}^{\#\mathrm{T}} + (\hat{\mathbf{x}}^{*\mathrm{T}} - \hat{\mathbf{m}}^{*\mathrm{T}})\mathbf{B} \tag{10.11}$$

   being $\mathbf{B} = \mathbf{V}_q^*(\mathbf{V}_q^{*\mathrm{T}}\mathbf{V}_q^*)^{-1}\mathbf{V}_q^{\#\mathrm{T}}$, where the row vectors $\hat{\mathbf{x}}^{\#\mathrm{T}}$, $\hat{\mathbf{m}}^{\#\mathrm{T}}$, $\hat{\mathbf{x}}^{*\mathrm{T}}$ and $\hat{\mathbf{m}}^{*\mathrm{T}}$ are the estimated missing part of row $\mathbf{x}^{\mathrm{T}}$, the estimated mean of the missing part, the available measurements of $\mathbf{x}^{\mathrm{T}}$ and its estimated mean vector, respectively.

4. Iterate the process until convergence.

Since in Figure 10.4, the rows of matrix $\mathbf{X}$ are represented by columns, the row-wise representation of the box 6 is $\mathbf{y}_n^{\#\mathrm{T}} = \mathbf{y}_n^{*\mathrm{T}}\mathbf{P}^*(\mathbf{P}^{*\mathrm{T}}\mathbf{P}^*)^{-1}\mathbf{P}^{\#\mathrm{T}}$. This equation, bearing in mind that the data in Figure 10.4 is previously mean-centred, is exactly the same as Step 3 in t-EM, being $\mathbf{P}^*$ the first q loadings (significant ones) of the available values of the covariance matrix (which are the same as in $\mathbf{X}$ matrix) and $\mathbf{P}^{\#}$ the first q loadings corresponding to the variables of the missing part.

Chapter 11

Network inference with missing data and outliers

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

[5] Folch-Fortuny, A., Villaverde, A.F., Banga, J.R. & Ferrer, A. Enabling network inference methods to handle missing data and outliers. *BMC Bioinformatics* **16**:283 (2015).

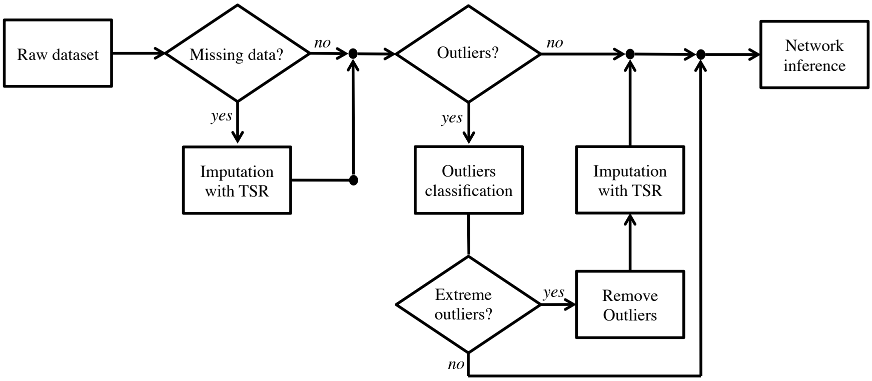


Figure 11.1: Missing data and outlier detection and correction modules.

11.1 Introduction

Network inference methods rely on estimating quantities such as correlation or mutual information, whose calculation requires simultaneous measurements of several variables. When the data collection in a time point fails for a particular variable, resulting in an unmeasured value, the scientist has to decide whether to discard the information regarding the entire experiment at this time point or to impute an appropriate value. Sometimes the data is not missing but is faulty. This case is possibly more dangerous, since the data point is taken as a true measurement and it can i) distort the relationships among entities, ii) generate false links in the network, and iii) hide true connections.

To avoid this problems, in this chapter, two new functional modules for curating the data are provided, which will be used as input to network inference procedures. The first module is devoted to handle MD using TSR in its PCA-MB version (proposed in Chapter 10). The second module detects extreme outliers in the raw dataset, and if there exist, they are first replaced by missing values and then recalculated using TSR.

The way both preprocessing modules act can be visualized in Figure 11.1. To evaluate the performance of TSR a comparison with other missing data methods commonly used by practitioners is carried out using several network inference benchmark problems. Likewise, several univariate and multivariate outliers are included in the datasets in order to check the ability of the outlier detection module.

Additionally, to illustrate how a network inference method can be augmented with these functionalities, they are used in combination with a state of the art technique called MIDER. MIDER is used here for demonstration purposes, but any other network inference method - such as those mentioned in Section 3.5.1 - could be used as well. The data curation modules presented here are of general purpose, and work on the data independently of the reverse engineering procedure.

To facilitate the joint use of MIDER and the new functional modules, a MATLAB/Octave implementation of the two data curation modules is included in a new version of the MIDER toolbox, MIDERv2 (`http://www.iim.csic.es/~gingproc/mider.html`)

Network inference studies that take into account the missing data imputation problem have been more common in the social sciences than in the biological sciences; however, some examples of the latter type can also be found. Thus, the works [307–311] report network inference results obtained with datasets with missing values. Wu et al. [307] presented a network inference method with an interpolation controller, providing three selections of data interpolation approaches. In [308, 309] MD is handled with a weighted k-nearest neighbor method. Hurley *et al* [311] illustrated the use of a suite of gene GRN analysis tools imputing MD using the LSImpute missing value estimation method. It should be noted that the aforementioned methods [307–309, 311] are specific for GRN inference with gene expression data, and the approaches chosen to handle MD are not justified nor compared to other alternatives. In [310] a network inference tool designed for GRNs, NetGenerator, is extended in several ways in order to predict pathogen-host interactions. Remarkably, NetGenerator is applicable to datasets with missing values, although it requires complete data for the last time point, which means that MD imputation procedures may still be needed in some cases. There are also studies that have addressed the issue of missing data in biological applications, but outside the context of network inference, e.g. [312].

The present contribution differs from the aforementioned works in that it i) presents a general method for handling MD and outliers, ii) compares its performance with that of other common approaches, iii) provides an implementation of the methodology, and iv) combines it with a freely available general-purpose network inference method. In this respect, there may be more resemblances with a recently published paper [313], which presents a framework for network inference in Cytoscape and includes the possibility of using three built-in MD “apps”: row average, zero imputation, and Bayesian principal component analysis. The contribution of this Chapter is complementary to [313] since i) the missing data imputation methods are different, and ii) here, outlier detection and correction is also taken into account.

The chapter is structured as follows. Section 11.2 describes how the functional modules work. Section 11.3 shows the results of the MD methods comparative study and the outliers study. Finally, some conclusions are drawn in Section 11.4.

11.2 Methods

11.2.1 Missing data methods

Two projection to latent structure methods are used in this chapter for imputing missing data: IA and TSR, both described in Chapter 10. There are other approaches to impute missing data that are commonly used by practitioners, commented in Section 2.5, such as CC and MI. Despite these methods are discouraged [47], since these methods are commonly used as a first (and fast) attempt to "solve" the problem of MD, their results are included in the comparative study. Additionally, two other methods are tested. The first one consists of imputing the average value of a linear interpolation (LI) between the previous and the posterior values of the missing datum. The second fills the missing values of an observation with the ones of its nearest neighbor (NN) using the k-nearest neighbors algorithm.

11.2.2 Outlier detection and correction

Projection to latent structure methods are commonly used within (bio)industries to monitor huge amounts of variables during (bio)processes. The concept of abnormal situation (or fault) in this context can be extrapolated here to errors during the data acquisition or mistakes during the data compilation. With this objective, PCA can be applied to find the latent structure of the original data, before reconstructing the network, and then identify these uncommon measurements.

When a PCA model is fitted, two different types of outliers can appear [20]: Hotelling-T^2 and SPE (more details in Section 2.3.1). While the first type of outlier usually represents a change in the process, but still coherent with the actual correlation structure of the data, the second one breaks the correlation structure, which is usually associated to failures in the process. The latter type of outliers should be removed in order to better understand the true structure of data. This idea can be extrapolated to network inference, being the SPE outliers the possible failures when the dataset is built. According to the previous rationale, these outliers have to be removed in order to study the relationships between species; otherwise these anomalous values could mask the true structure of data. Once an observation is classified as an outlier, contribution plots can be used to isolate the original variable responsible of this abnormal behavior [314]. Outliers in the T^2 statistic are not evaluated in the present approach, since they are not as harmful as SPE's for inferring links between species in a network.

The present chapter proposes a methodology for automatically detecting and correcting outliers. Some related approaches have been published in the past focusing on fitting PCA models with missing data and/or outliers [315]. Here, since the goal is to cure the dataset, we use TSR to correct for the outliers, thus allowing the exploitation of all the data available for the network inference task.

The outlier detection and correction procedure works as follows. The first step consists of calculating the PCA model of data. As in TSR, the number of PCs is here determined automatically, extracting all the PCs whose associated eigenvalue, in the SVD, is higher than one [316]. It is worth noting that the number of PCs determined here may be different than the number of PCs extracted by TSR, since in the outlier scheme we want to detect deviations from the main directions of variability, i.e. deviations from the PCs with the highest eigenvalues.

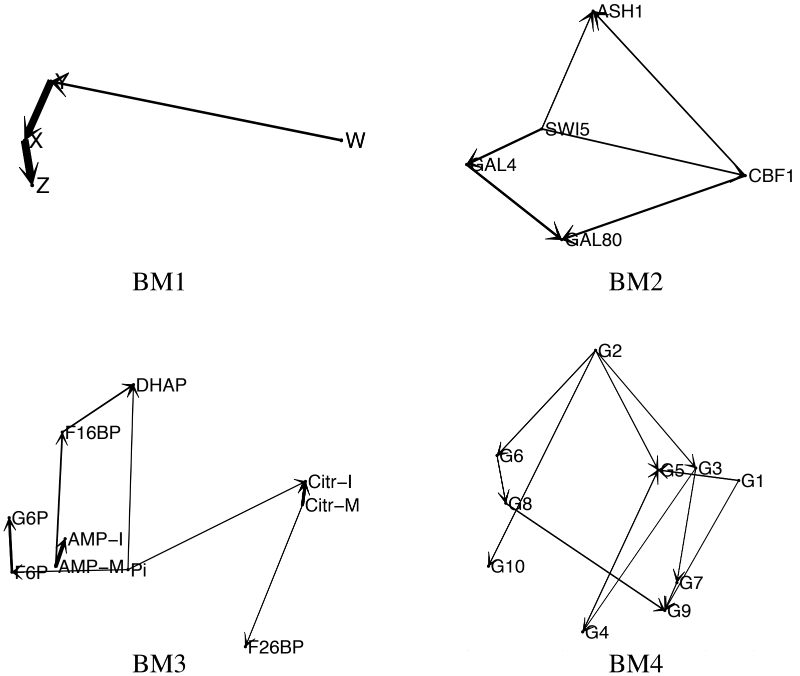
Once the PCA model is fitted, the next step consists of calculating 95% upper control limit of SPE in order to detect possible faults. Different ways of computing control limits have been proposed in the literature [314]. However, when dealing with real data, which do not fulfill the theoretical constraints, the outliers may not be correctly detected in this way. Hence, in this study, the control limits are computed using a CV scheme [317] as follows: a thousand data subsets, using 95% of the observations, are randomly selected to compute "real" 95% control limits, i.e. leaving 5% of the observations above the limit. Then, the median of all the limits is computed to take it as the reference limit. Values above this limit are considered outliers.

From the set of outliers, another classification has to be done. Any control limit based on a confidence interval has intrinsically a false alarm rate, corresponding to the confidence level. In the present case, 5% of the observations are likely to fall above the control limit without necessarily being outliers. If more observations appear, they can be considered as faults. To distinguish between both types of "outliers", an outlier classification recently proposed in [318] is followed. Firstly, the outliers are ordered in decreasing order. Secondly, the admissible false alarm rate, corresponding to 5% of the observations, are no longer considered as outliers. Finally, each value above two times the control limit is considered an outlier. Additionally, a data point is considered an outlier if its distance to the control limit exceeds 10 times the distance between the lowest false alarm and the control limit.

Once the extreme outliers have been identified it is needed to determine the variable responsible for the fault. For this purpose, contribution plots can be used to determine which variable k of the nth observation has the highest SPE [21]:

$$Cont(SPE, x_{nk}) = e_{nk}^2 \tag{11.1}$$

Once the responsible variable is identified, a missing value is generated for that value. And finally, TSR is again used to reconstruct the faulty observation following the latent structure of data.



BM1

BM2

BM3

BM4

Figure 11.2: Case studies. MIDER reconstructed networks with the original data for benchmark problems BM1 (small chain of 4 reactions), BM2 (In vivo Reverse-engineering and Modeling Assessment), BM3 (first steps of a glycolytic pathway) and BM4 (DREAM4 in silico network challenge)

11.2.3 Case studies

In order to validate the usefulness of the proposed methodology five well-known benchmark problems are selected, four from [98] and one new test case from the DREAM5 Network Inference challenge[1]. Additionally, to test the performance of the TSR method against the other approaches, a comparative study is also performed. The first benchmark problem (BM1) is a small chain of three reactions between four species (W, Y, X and Z) proposed in [319, 320] (see BM1 in Figure 11.2). In this network the reaction between W and Y is much weaker than the other reactions $Y - X$ and $X - Z$.

The second benchmark problem (BM2) is the so-called IRMA (In vivo Reverse-engineering and Modeling Assessment) [321]. It corresponds to a yeast synthetic network for benchmarking reverse-engineering approaches. IRMA consists of five genes that regulate each other through several interactions. It is particularly

[1] http://wiki.c2b2.columbia.edu/dream/index.php/D5c4

interesting as a benchmark because it is an engineered system, which means that the true network is known, and at the same time the system outputs can be measured in vivo, instead of just simulated in silico. A dataset consisting of time series and steady-state expression data after multiple perturbations is available; for network inference purposes the time-series data was used. Figure 11.2 shows the reconstruction of the network by MIDER.

BM3 models the first steps of a glycolytic pathway. The reconstruction of this network is shown in Figure 11.2. The problem of reverse-engineering this system was chosen in [322] as a way of demonstrating the feasibility of the Correlation Metric Method (CMC) . With that aim, an experiment was carried out in a continuous-flow, stirred-tank reactor. Experimental time-series data were obtained for the concentrations of ten chemical species: Pi, G6P, F6P, F16BP, F26BP, and DHAP, as well as the input and reactor concentrations of citrate and AMP. The sampling period was 13 minutes, and the overall number of sampling instants was 57. The data is publicly available online[2] as part of the Deduce software package.

The fourth benchmark problem (BM4) was generated for the DREAM4 in silico network challenge[3]. This challenge aimed at reverse engineering genetic networks. The artificial network presented here was generated as reported in [323, 324]. It consists of a 10 nodes and 13 links. The MIDER reconstruction of this network is shown in Figure 11.2.

Finally, the last benchmark problem (BM5) is an in-silico network produced for the DREAM5 network inference challenge. Specifically, we used the problem referred to as *Network 1*, which is an in silico network with 1643 nodes (genes), of which 195 are transcription factors. The challenge consisted in reporting an ordered list of 100000 predicted interactions. The reconstruction of the network is not shown here.

11.3 Results

11.3.1 Missing data: comparative study

In this section, the results of the tests of the missing data module are shown. The performance of the TSR method for the imputation of MD is compared against another multivariate projection method, IA, and other fast approaches used by practitioners, like the CC, MI, LI and NN. The study is performed as follows. The five benchmark problems (BM1–BM5) described in the previous section are chosen as case studies. In BM1–BM4, 7 different percentages of missing data are generated, from 5% to 35%, and for each percentage, 100 datasets are simulated.

[2] http://genomics.lbl.gov/?page id=44

[3] http:// wiki.c2b2.columbia.edu/dream/index.php/D4c2

For BM5 the same percentages of MD are generated; however, only one dataset was simulated for each of them, due to the much higher computational cost of reverse engineering such a large-scale network compared to BM1–BM4.

In Chapter 10, the MSPE was used to assess the performance of each imputation method. However, in the present study the application is different: the aim is to use the imputed data for network inference. Hence, since different imputations may lead to the same inferred network, we use here instead the precision (P) and recall (R) of each method as the performance criteria to evaluate the results. These statistics have been introduced in Chapter 6, 7 and 9. Here, TP are the true predicted links with respect to the reconstruction of the network without missing values, FP the false positives, and FN the false negatives.

The mean results of P and R, for each BM problem, are shown in color scale maps in Figure 11.3. An absolute zero value, on either P or R, in Figure 11.3 implies that the method is unable to impute the missing values for that particular percentage. To determine whether the differences in P and R obtained among TSR and the other methods, for each percentage, are significant or not, a mixed-effects three-way ANOVA, using LSD intervals for the statistically significant differences, is applied on the results, in the same way as presented in Chapter 10.

Figure 11.3 shows that CC, LI and NN are not able to impute the missing data for medium-high percentages. CC is the worst method: for small networks (BM1 and BM2) CC can deal with up to 10%-15% of missing data, but for more complex networks (BM3 and BM4) it can only perform an imputation with 5% of missing values. LI and NN are slightly better, but they fail to reach 30% of missing data for all BMs. The TSR performance in all BMs is statistically better than CC, LI and NN. TSR results are superior to MI in most of the percentages of all BMs. Regarding IA, TSR attains statistically better results for most percentages in BMs 2-4. In the large network (BM5) no statistical differences can be computed, however, none of the aforementioned methods seems to outperform TSR in this case study. CC and NN are not applicable in BM5 since, even considering only 5% of missing values, all rows in the data set have missing data. Furthermore, the results in this DREAM5 network are similar to the mean values obtained in DREAM4 network (BM4). Since no single method is statistically better than TSR for any percentage of any dataset, it is sensible to choose this method to implement the missing data module of network inference procedures.

The MD imputation is achieved in around 1-2 seconds in BM1-BM4, for all methods; however, most of them are unable to impute with medium-high percentages of missing data. In BM5, LI maintains the computation time of BM1-BM4, the mean imputation (MI) is achieved in 48 seconds, and the most accurate methods in terms of P and R (IA and TSR) perform the imputation in 1 hour by truncating the number of PCs to 50 (to accelerate convergence).

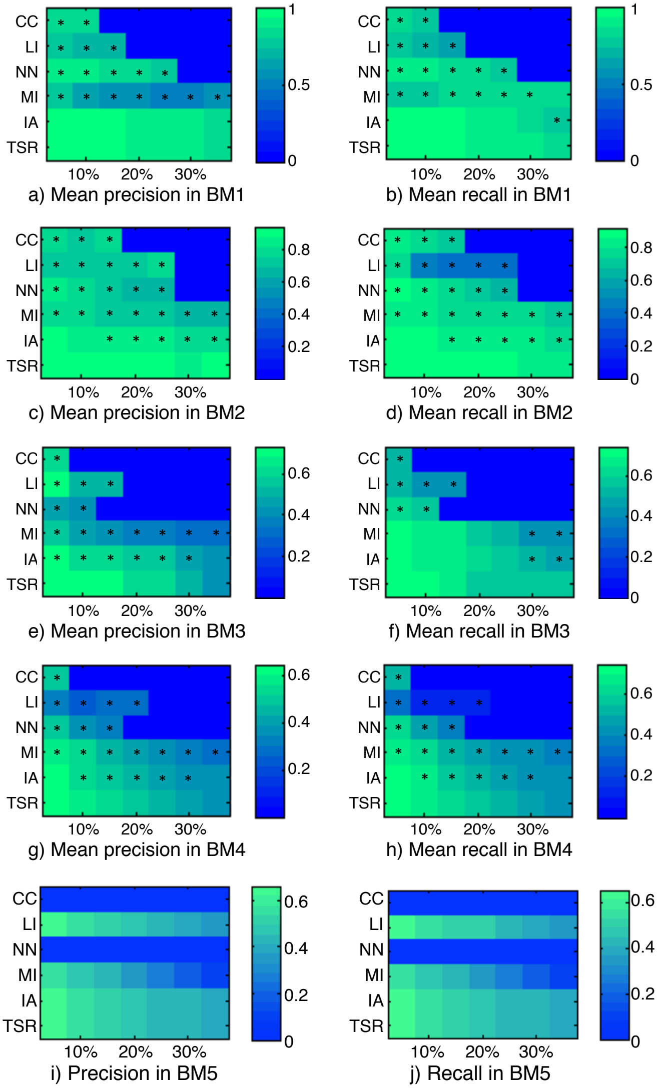


Figure 11.3: Results of the MD comparative study. The first (second) column shows P (R). Rows correspond to BM1 (a-b), BM2 (c-d), BM3 (e-f), BM4 (g-h) and BM5 (i-j). The asterisks on the color maps in Figure 11.3 mark the statistically significant differences (p-values < 0.05) with respect to TSR.

Regarding the reconstruction of the network, TSR, when used in combination with MIDER, is able to recover more than 90% of the links inferred with the complete dataset in small networks with low percentages of missing data. For higher percentages, 30-35% of MD, it can infer 80% of the links. Furthermore, for more complex networks with low percentages of MD, it is able to reconstruct nearly 70% of the links inferred with complete data. When the % of MD increases in these networks, it can recover 40-50% of the links.

11.3.2 Outlier detection and correction: a simulated study

A simulated study is described here to assess the performance of the outlier detection and correction scheme. Two types of outliers are simulated on the benchmark problems: univariate and multivariate outliers. The first group involves outliers in the usual sense, i.e. values 3 times the interquartile range above (below) the 3rd (1st) quartile. The second group involves outliers that do not satisfy the aforementioned condition, i.e. they are not outliers in a univariate sense, but nonetheless alter the data correlation structure. This means that values above (below) the mean plus (minus) 1.5 times the standard deviation are moved to the other side of the mean value, e.g. if a variable has mean 0 and standard deviation 1, a value of 2 is moved to -2. This latter group of outliers requires a multivariate approach to be detected since they are not univariate outliers, i.e. they can not usually be detected using, for example, a box-whiskers plot.

Three percentages of outliers are simulated in each dataset: 1% (with a minimum of 2 outliers), 5% and 10%. 100 rounds of univariate and multivariate outliers are simulated for each benchmark problem. A paired t-test with α Type-I risk = 0.05 is used to determine if the solution provided by MIDER is significantly improved by the inclusion of the detection and correction module.

Table 11.1 shows the results of the simulated study using univariate outliers. P and R results of the network reconstruction of MIDER and MIDER + outlier scheme are shown by rows. It is worth noting that the MIDER reconstruction of BM1 with univariate outliers has exactly the same links as MIDER with the original data set. However, the inclusion of the outlier detection and correction scheme presents no significant differences among the results. In BM2-BM3, the performance of MIDER + the outlier scheme is statistically superior to the results of MIDER on the faulty data. In this way, the network reconstructed correcting the detected outliers is more similar to the one inferred by MIDER using the original data. The results for BM4 are not statistically significant. Regarding BM5, no statistical differences can be computed among methods, since only one simulation per case is performed. However, the P and R results are coherent with the mean values obtained for BM4. The outlier detection and posterior imputation in BM1-BM4 is performed in 2-3 seconds. This procedure takes 2 minutes in the large network (BM5).

Table 11.2 shows the results of the simulated study using multivariate outliers. In this case, MIDER + the outlier scheme performed statistically better than MIDER in BM1-BM4 for all percentages of outliers. The differences are bigger when the network is simpler, e.g. BM1-BM2, however there is also a significant improvement in the quality of the solution in BM3-BM4 when MIDER is used with this module. Finally, as in the univariate outliers study, no statistical differences can be computed in BM5.

11.3.3 Remark on computation times

The computational cost associated to these procedures is relatively modest, compared to that of performing network inference. For the benchmark problems used in this work, the cost of the TSR-based approach is of only a few seconds for networks of moderate size (BM1-BM4), while for the very large one (BM5, including thousands of genes) it is of two minutes for outlier detection and correction and one hour for missing data imputation. These values do not represent a significant increase in the computation times of the network inference procedure, which means that the proposed methodology is appropriate for problems of realistic size.

11.4 Discussion

This chapter presents an enhancement of network inference methods consisting of two preprocessing modules for handling incomplete and faulty datasets. The first one is capable of imputing values for lost measurements coherently with the latent structure of data. The second module detects univariate and multivariate outliers and replaces the faulty measurements with new values coherently with the available data.

A comparison of different methodologies for handling MD has led to two main conclusions. First, traditional approaches used by practitioners, like CC, LI and NN, have problems when the datasets have high percentages of missing values and when the complexity of the network increases. Second, since TSR performs significantly better than the previous methods in BM1-BM4, including the MI and IA, it represents the best approach to deal with missing values in network inference.

Likewise, the module for detecting and correcting outliers proposed here also applies TSR for replacing faulty observations. The good performance of this approach has been shown by means of simulations with five benchmark problems. The results of the network inference in four benchmarks with simulated multivariate outliers are statistically better when the new module is used as a preprocessing step; results obtained with univariate outliers in the fourth benchmark are not statistically significant.

Outliers	Method	BM1		BM2		BM3		BM4		BM5	
		P	R	P	R	P	R	P	R		
1%	MIDER	100%	100%	84.4%	89.7%	89.3%	85.22%	92.0%	91.3%	99.9%	99.9%
	MIDER+OS	100%	100%	96.9%*	97.0%*	94.6%*	91.3%*	90.6%	90.8%	99.8%	99.8%
5%	MIDER	100%	100%	85.1%	87.7%	90.7%	84.6%	88.7%	88.3%	99.8%	99.8%
	MIDER+OS	99.7%	99.7%	91.0%*	92.3%*	93.1%*	88.1%*	87.6%	87.9%	98.9%	98.9%
10%	MIDER	100%	100%	85.7%	88.0%	86.9%	79.8%	84.4%	86.4%	99.4%	99.5%
	MIDER+OS	99.7%	99.7%	88.2%*	89.7%*	90.2%*	81.8%*	82.7%	85.4%	99.4%	99.5%

Table 11.1: Univariate outliers simulated study. *P* and *R* results of the network inference of MIDER and MIDER + the outlier scheme (MIDER+OS). The asterisks mark the statistically better results of MIDER+OS (p-value < 0.05).

Outliers	Method	BM1		BM2		BM3		BM4		BM5	
		P	R	P	R	P	R	P	R	P	R
1%	MIDER	66.7%	66.7%	84.9%	88.2%	86.9%	82.0%	88.3%	88.8%	99.9%	99.9%
	MIDER+OS	100%*	100%*	91.6%*	94.6%*	90.8%*	88.6%*	91.5%*	91.2%*	98.6%	98.6%
5%	MIDER	65.7%	65.7%	83.7%	86.8%	83.0%	74.7%	79.6%	83.1%	98.6%	98.6%
	MIDER+OS	91.0%*	91.0%*	86.9%*	90.5%*	85.6%*	80.2%*	82.8%*	85.1%*	97.3%	97.3%
10%	MIDER	64.0%	64.0%	80.6%	83.2%	70.1%	69.7%	70.4%	77.3%	98.8%	98.8%
	MIDER+OS	73.0%*	73.0%*	82.6%*	85.8%*	72.8%*	71.1%*	73.0%*	79.2%*	98.2%	98.2%

Table 11.2: Multivariate outliers simulated study. *P* and *R* results of the network inference of MIDER and MIDER + the outlier scheme (MIDER+OS). The asterisks mark the statistically better results of MIDER+OS (p-value < 0.05).

Due to the long computation time required to analyse the fifth benchmark, a single simulation is performed in this large-scale network. Therefore, no statistical differences are computed in this case study. However, the results of TSR in the comparative study and the outlier detection and correction are coherent with the results of BM4.

By extending MIDER with these new functionalities, two different approaches for data analysis: information-theoretic and variance-based, are combined. The joint use of both methodologies increases significantly the number of datasets that can be used for network inference.

Crucially, both MIDER and the new preprocessing modules are general-purpose methods, which may be applied to networks of any kind – not only biological, but also from other areas of science – without requiring prior knowledge from the user. Furthermore, the missing data and outlier detection and correction modules can be used as a preprocessing step for other network inference methods.

Chapter 12

Missing data imputation toolbox

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

[8] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Missing Data Imputation Toolbox for MATLAB, *Chemometrics and Intelligent Laboratory Systems* **154**, 93-100 (2016).

12.1 Introduction

The problem of missing data arises in several research areas, such as chemometrics (see Chapter 10), genomics [325], network inference (see Chapter 11), meteorology [326], engineering [327], informatics [328], and chemical [47], biochemical [318] and pharmaceutical [329] industries. To help scientists across these research areas, we present here a GUI in MATLAB, called MDI Toolbox, devoted to fulfil incomplete data sets following MCAR. The MDI toolbox is freely available for academic purposes at `http://mseg.webs.upv.es`, under a GNU license.

The missing values are imputed applying PCA-MB methods with missing data. The different methods implemented in MDI Toolbox are: TSR, KDR, KDR with PCR, KDR with PLS, PMP, IA, NIPALS and DA. The main outputs of MDI Toolbox are the PCA model of the incomplete data set, the estimated covariance matrix and the original missing data set with the imputed missing values.

This chapter is organised as follows. Some comments on the software requirements for the toolbox are made in Section 12.2. The data sets included as examples in the toolbox are presented in Section 12.3. An example of analysis using MDI Toolbox is proposed in Section 12.4, explaining in detail the steps via the GUI to obtain the missing data imputation. Finally, some concluding remarks are made on Section 12.5.

12.2 Software specifications and requirements

MDI Toolbox has been built in MATLAB R2013a, and it has been tested in many different previous and posterior MATLAB versions (2010-2015). The toolbox consists of a set of `.m` files, with the source code of the menus and the imputation methods; a set of `.fig` files, with the GUI; and a `.mat` file (`MDI_Examples.mat`) with few examples to run the toolbox. The toolbox is launched introducing `MDIgui` in the MATLAB command window. Afterwards, the main function calls other auxiliary routines (`SelectData`, `SelectExample`, `DataOverview`, `NumberComponents` and `ShowResults`) until performing the imputation. The output of the toolbox is a data structure, whose fields are described in Table 12.1.

The methods implemented in the MDI Toolbox are described in detail in Chapter 10.

Field	Type	Content
Dataset	string	Name of the data set
X_MD	array	Data set with missing values
Percentage_MD	double	Percentage of missing values
X_imputed	array	Data set with the imputed values
PCs	integer	Number of principal components selected
Mean	arary	Estimation of the mean vector
Covariances	array	Estimated covariance matrix
Iterations	integer	Iterations required (not available when DA is applied)
Tolerance	integer	Threshold for convergence (not available when DA is applied)
X_reconstructed	array	Predictions of the PCA model
Method	string	Method applied
Computationtime	double	Computation time measured in seconds
Ini_est_cum_R2	array	Initial estimation of the cumulative explained variance in data (R^2)
Ini_est_eig	array	Initial estimation of the eigenvalues of the covariance data matrix
Loadings	array	Loadings matrix of the PCA model of `X_imputed`
Scores	array	Scores matrix of the PCA model of `X_imputed`
Num_Markov_chains	integer	Number of Markov chains computed when DA is applied
Chain_Length	integer	Length of each Markov chain computed when DA is applied

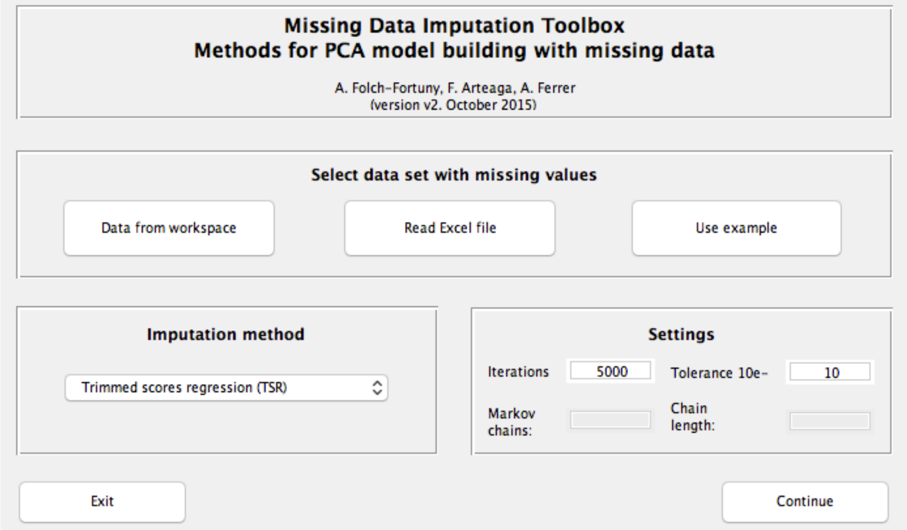
Table 12.1: Data fields within the MDI Toolbox results structure.

12.3 Data sets

Three data sets are included in the MDI Toolbox. These datasets correspond to the case studies used in Chapter 10: the Olive Oil data set [301], the Diesel NIR data set [302] and the Simulated data set with 3 PCs [303, 304]. For each data set the toolbox includes the complete data, and data sets with 10%, 30% and 60% of missing data following MCAR patterns.

12.4 Operating procedure

MDI Toolbox is launched introducing `MDIgui` in the MATLAB command window. Figure 12.1 shows the initial window of the graphical interface. The first step consists of selecting the data set. The button `Data from workspace` permits loading a data set with missing values from the MATLAB workspace. A data set previously stored in Excel can be loaded clicking at `Read Excel File` (more details on the Excel data can be found in Section 12.6). The button `Use example` opens a new window with example data (see Figure 12.2). This way 3 different data sets can be selected with three percentages of missing values: 10%, 30% and 60%. For this tutorial the Simulated data set with 30% of missing values is selected. The MD imputation method is also selected in the `MDIgui` window. Among the available methods, the recommended one is TSR, since it represents a good com-



Missing Data Imputation Toolbox
Methods for PCA model building with missing data

A. Folch-Fortuny, F. Arteaga, A. Ferrer
(version v2. October 2015)

Select data set with missing values

Imputation method

Trimmed scores regression (TSR)

Settings

Iterations 5000

Tolerance 10e- 10

Markov chains:

Chain length:

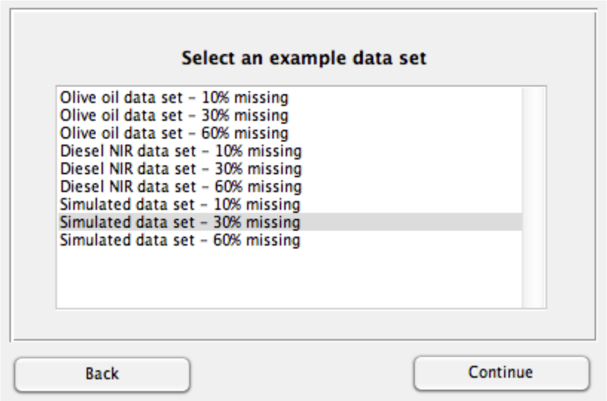
Figure 12.1: MDI Toolbox GUI for data, method and settings selection.

promise solution between prediction quality, robustness against data structure and computation time, as seen in Chapter 10.

The MDI interface allows also changing the settings of the different methods. In this way, the number of maximum iterations performed by the method and the tolerance for the convergence can be modified from their default values: 5000 iterations and a tolerance of 10^{-10}. These parameters are active when the regression-based methods, IA and NIPALS are active. If DA is selected as the imputation method, these settings are disabled and the Number of Markov chains and Chain Length are enabled. So the user can modify the default 100-iteration 10 Markov chains.

Once the data, method and settings have been introduced, the `DataOverview` window appears. The pattern of missing values and its percentage can be visualised here (see Figure 12.3). The red squares represent the missing entries in the data set, and the white ones the available values. After clicking `Continue` two progress bars appear one after the other. The first one shows the calculation progress of the variances and the second one the calculation progress of the covariances.

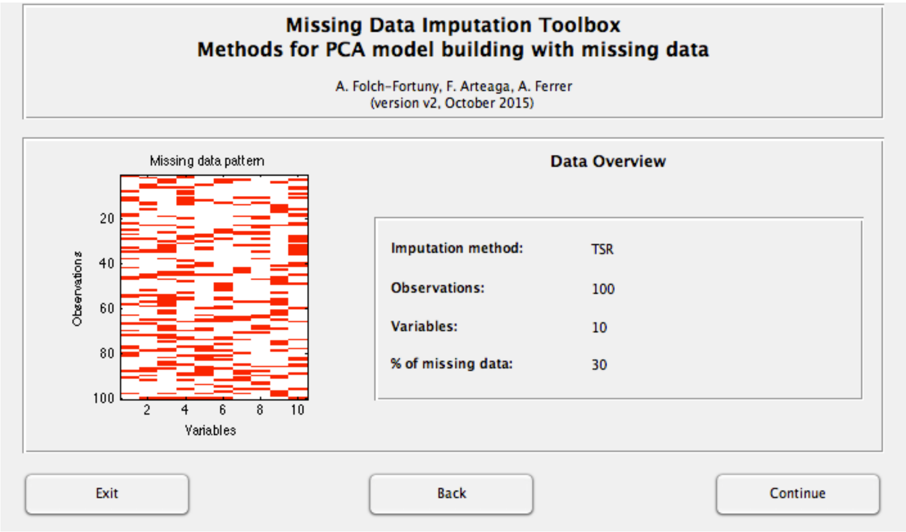
The next window, `NumberComponents`, allows the user to select the appropriate number of PCs for the PCA model. Three plots are presented here to assess this number (see Figure 12.4). On the left side the classical scree plot, with the eigenvalues of the estimated covariance matrix of $\mathbf{X}$. On the center, the cumulative percentage of explained variance. It is worth noting that both plots are obtained based on a pairwise estimation of the covariance matrix of the data set with missing values,



Select an example data set

- Olive oil data set - 10% missing
- Olive oil data set - 30% missing
- Olive oil data set - 60% missing
- Diesel NIR data set - 10% missing
- Diesel NIR data set - 30% missing
- Diesel NIR data set - 60% missing
- Simulated data set - 10% missing
- Simulated data set - 30% missing
- Simulated data set - 60% missing

Figure 12.2: Example data selection window.



Missing Data Imputation Toolbox
Methods for PCA model building with missing data

A. Folch-Fortuny, F. Arteaga, A. Ferrer
(version v2, October 2015)

Data Overview

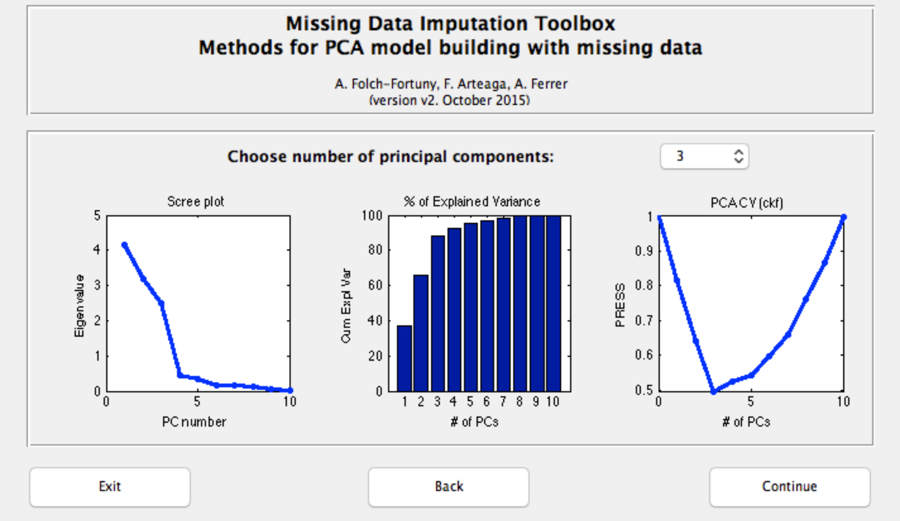
Imputation method: TSR

Observations: 100

Variables: 10

% of missing data: 30

Figure 12.3: GUI for data overview.



Missing Data Imputation Toolbox
Methods for PCA model building with missing data

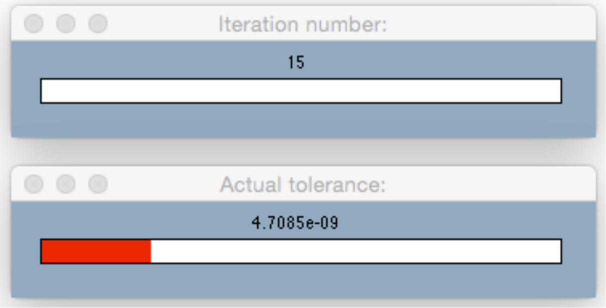
A. Folch-Fortuny, F. Arteaga, A. Ferrer
(version v2. October 2015)

Choose number of principal components: 3

Figure 12.4: Selection of the number of principal components, based on the scree plot (left) and the cumulative explained variance bar plot (center), and the PCA cross validation using the *ckf* algorithm.

i.e. the covariance between each pair of variables is computed using only rows with non-missing values in both variables. Using this procedure, pseudo-covariance matrices are obtained, i.e. they may be non-positive semidefinite. Since this matrix is used only to determine the number of PCs, corresponding to the highest eigenvalues, it is not important whether some negative eigenvalues are obtained by its SVD. A third plot is included at the right side. This plot corresponds to the results of the column-wise k-fold (*ckf*) algorithm to estimate the number of PCs in PCA, recently proposed in [330]. This algorithm is an efficient adaptation of the previously proposed element-wise k-fold (*ekf*) algorithm, which is based on the capability of PCA to recover missing data [19]. Here, since our data set has, originally, missing values, the *ckf* algorithm can be used to select the number of components with the lowest sum of squares of the prediction error (PRESS). More details on the *ckf* algorithm can be found in [330]. The code for *ckf* algorithm has been taken from the MEDA Toolbox for MATLAB, and it can be downloaded separately from `https://github.com/josecamachop/MEDA-Toolbox/releases/tag/v1.0`. These three plots are included in the MDI Toolbox to give the practitioner different criteria to select the number of PCs, which is a critical issue even with complete data [19].

In this case study the information provided by the three plots in Figure 12.4 is coherent, so three PCs are selected, since i) there is a huge difference in the eigenvalues between 3 and 4 components in the scree plot, and the differences are small between 4 and more components; ii) the cumulative explained variance with


Iteration number:

15

Actual tolerance:

4.7085e-09

Figure 12.5: Progress bars reflecting the missing data imputation procedure. In this example 15 out of the 5000 iterations have been computed (top), and the mean squared difference between the imputed values in iterations 14 and 15 is 4.7089×10^{-9} (bottom).

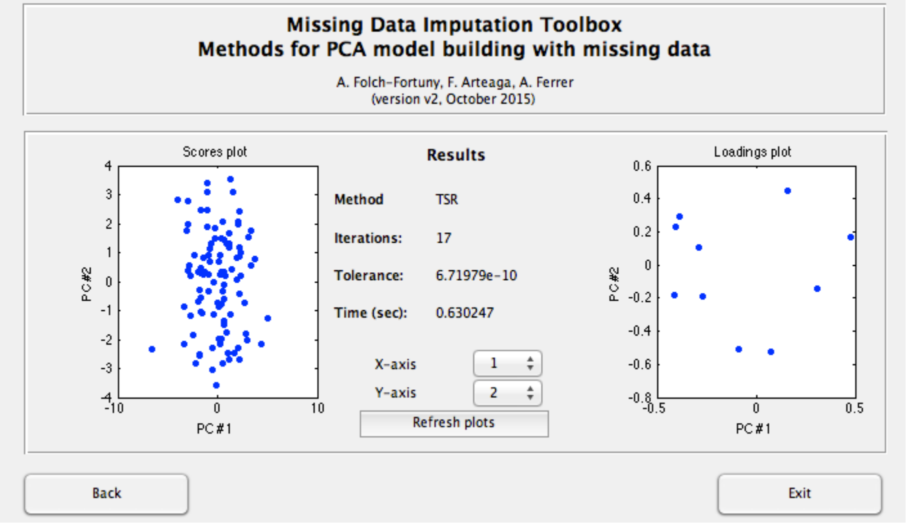
three components is around 90%, and the variance explained with 4 components is similar; and iii) the PRESS is minimum using three components.

Once the number of PCs is selected, MDI Toolbox runs the selected MD method to impute the missing values. The computation time depends on the method selected. Usually TSR and IA are the fastest methods, and DA and KDR are the slowest ones.

Two progress bars appear simultaneously (see Figure 12.5) while the toolbox is performing the iterative imputations. The top bar shows the current iteration number, and runs until reaches the maximum number of iterations specified in the `MDIgui` initial window (see Figure 12.1). The bottom bar gives an idea of how far is the difference between consecutive iterations from the tolerance defined for convergence. This is calculated as $1 - \frac{d-l}{d}$ where d is the mean squared difference between the imputed values in consecutive iterations and l is the specified tolerance. The first progress bar that is fulfilled stops the calculations, therefore, if the iterations bar reaches the maximum, it implies that the established convergence criterion is not achieved.

The last window of MDI Toolbox is `ShowResults` (see Figure 12.6). Here, the details of the data imputation are summarised: imputation method, iterations, tolerance and computation time (in seconds). Also, two figures with the loadings and scores plots are shown to ease the graphical interpretation of the model. The axis of both plots can be changed via the pop-up menus.

Finally, MDI Toolbox returns automatically a data structure to the MATLAB workspace with all the information of the data imputation (see Table 12.1). Among other parameters related to the number of iterations, computation time, *etc.* the original data set with imputed values are stored in the field `X_imputed` of the MATLAB structure `MDIToolbox_results`. Additionally, the resulting PCA



Missing Data Imputation Toolbox
Methods for PCA model building with missing data

A. Folch-Fortuny, F. Arteaga, A. Ferrer
(version v2, October 2015)

Results

Method TSR

Iterations: 17

Tolerance: 6.71979e-10

Time (sec): 0.630247

X-axis 1

Y-axis 2

Figure 12.6: Scores and loadings plots from the PCA model fitted on the imputed data set.

model fitted on this data is stored in the fields `Loadings` and `Scores`, as well as the mean and the covariances of the variables. In this way, the data is reproduced as: $\texttt{X\_reconstructed} = \texttt{Mean} + \texttt{Scores} \times \texttt{Loadings}^{\mathrm{T}}$.

12.5 Concluding remarks

In this chapter a new MATLAB toolbox is presented devoted to impute MD. MDI Toolbox includes PCA model building methods with missing data that are able to reconstruct the missing values coherently with the latent structure of the available data. Several methods from the literature are included in this toolbox: TSR, KDR, KDR-PCR, KDR-PLS, PMP, IA, modified NIPALS and DA. TSR is presented as the default method for its good performance with all data structures, as commented in Chapter 10.

A GUI is provided with the toolbox to ease its use. In this way, several windows guide the user step by step: from the data loading and settings to the results exploitation via interactive loadings and scores plots.

The purpose of MDI Toolbox is two-fold. On one hand, this toolbox permits to fit PCA models when there exist missing values in the original data set, obtaining as a result the loadings and the scores matrices. On the other hand, this toolbox can be used as a preprocessing step of other methodologies, since one of the outputs is simply the original data matrix with the imputed missing data.

The MDI toolbox is freely available for academic purposes at `http://mseg.webs.upv.es`, under a GNU license.

12.6 Appendix A. Excel files.

To read Excel files with MDI Toolbox there have to be no headers nor observations names in the sheet. Also, there has to be only one sheet in the Excel file, containing the data set to analyse. The missing values have to appear as blank cells.

12.7 Appendix B. Using MATLAB command window.

The missing data imputation can be obtained typing the specific functions directly on the MATLAB command window, without using the GUI windows. This way:

$$[\mathtt{X}, \mathtt{m}, \mathtt{S}, \mathtt{It}, \mathtt{diff}, \mathtt{Xrec}] = \mathtt{pcambtsr}(\mathtt{X\_MD}, \mathtt{A}, \mathtt{M}, \mathtt{f}) \tag{12.1}$$

imputes, using TSR, the missing values in matrix `X_MD` using `A` components, a maximum of `M` iterations, and a tolerance `f`. The outputs are the original data matrix with the imputed values (`X`), the mean and covariance estimations (`m` and `S`, respectively), the number of iterations (`It`) and the tolerance value (`diff`). Also, the reconstructed matrix **X** using the final PCA model is obtained (`Xrec`).

To impute using the other regression-based methods, IA and modified NIPALS, the user only has to change the name of the function in Command 12.1 `pcambkdr` for KDR, `pcambpcr` for KDR-PCR, `pcambpls` for KDR-PLS, `pcambpmp` for PMP, `pcambia` for IA, `pcambnipals` for modified NIPALS algorithm.

For DA, the user has to type: $[\mathtt{X}, \mathtt{m}, \mathtt{S}, \mathtt{Xrec}] = \mathtt{pcambda}(\mathtt{X\_MD}, \mathtt{M}, \mathtt{CL}, \mathtt{A})$, where `X_MD` and `A` are the matrix with missing values and the number of components of the final PCA model, and `M` and `CL` are the number of Markov chains and the chain length, respectively.

Chapter 13

Framework for MLPCA missing data imputation

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

[9] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Assessment of maximum likelihood PCA missing data imputation. *Journal of Chemometrics* **30**, 386-393 (2016).

13.1 Introduction

MLPCA was originally proposed to incorporate measurement error variance information in principal component analysis (PCA) models [29]. MLPCA has been widely applied in several works within chemometrics and systems biology, *e.g.* to analyse reflectance FTIR microspectroscopic data [331] and ion mass spectroscopic data [332], to fault detection in process industry [333], to the characterization of measurement errors in nuclear magnetic resonance (NMR) data [334] and gene expression data [335], to determine the appropriate number of reactions in stoichiometric modelling [336], and as a useful preprocessing tool for metabolomic, proteomic, transcriptomic [337] and environmental [338] data analysis.

Shortly after the publication of the original MLPCA algorithm, an application of this method was proposed addressing the MD problem in PCA-MB [339]. MLPCA deals with the missing values by assigning them large variances prior to implementing the method, which guides the algorithm to fit a PCA model disregarding these data points. The MLPCA approach for MD has been applied successfully in the literature to fluorescent, chromatographic, near-infrared spectroscopic [339], spectrophotometric [340], and environmental [341] data.

Nelson [30] showed the equivalence between the scores calculation by columns in MLPCA and the PMP algorithm for PCA-ME. Also, the PMP algorithm has been adapted in Chapter 10) to a MB environment. Here, the equivalence between the imputation step observation-wise in MLPCA algorithm and the adapted PMP method for PCA-MB. is proven.

The aim of this chapter is, thus, to answer three questions that arise from the aforementioned equivalence:

1. Once the MD algorithms converge, are the imputed values of MLPCA and PMP for PCA-MB equal?

2. Since TSR outperforms PMP, if the imputation step in MLPCA is substituted by a TSR-based imputation, does the imputation outperform the original MLPCA?

3. In any case, does MLPCA, or its adapted version with TSR, outperform the original TSR algorithm?

To answer these research questions, the regression-based methods presented in Chapter 10 (KDR, KDR with PCR, KDR with PLS and TSR) are here adapted to work as different imputation steps within the MLPCA algorithm, providing a framework for maximum likelihood MD imputation. The performance of these methods is compared to PMP and TSR methods using six data sets from different environments, actual and simulated ones, taken from systems biology, chemometrics and food industry.

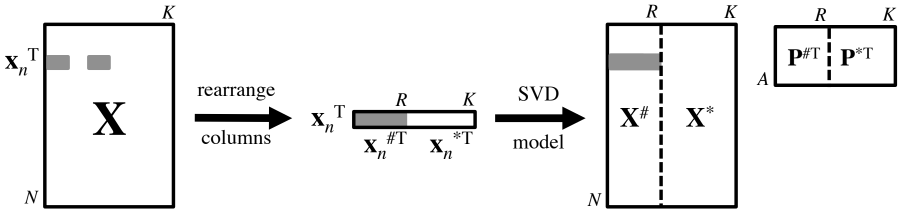

K
$\mathbf{x}_n^{\mathrm{T}}$
X
N
rearrange
columns
$\mathbf{x}_n^{\mathrm{T}}$
R
K
$\mathbf{x}_n^{\#\mathrm{T}}$
$\mathbf{x}_n^{*\mathrm{T}}$
SVD
model
R
K
$\mathbf{X}^{\#}$
$\mathbf{X}^{*}$
N
R
K
A
$\mathbf{P}^{\#\mathrm{T}}$
$\mathbf{P}^{*\mathrm{T}}$

Figure 13.1: Partition induced in $\mathbf{X}$ matrix by the missing data in its nth row. Grey squares denote missing positions in the data set.

The rest of the chapter is organised as follows. Section 13.2 proves the equivalence between the imputation step observation-wise of MLPCA and the PMP method for PCA-MB, and describes how the regression-based methods are adapted to its ML version. Sections 13.3-13.4 describe the data sets used in this study, as well as how the comparative study is performed. Section 13.5 shows the results of the ML regression-based methods, jointly with the original PMP and TSR algorithms. Finally, the conclusions are highlighted in Section 13.6.

13.2 Maximum likelihood regression-based methods

The original MLPCA algorithm has been described in Section 2.3.3. The adaptation of MLPCA to MB with MD assumes uncorrelated errors for both objects (rows) $\mathbf{x}_n^{\mathrm{T}}$ and variables (columns) $\mathbf{x}_k$ of the $(N \times K)$ data matrix $\mathbf{X}$. Therefore matrices $\mathbf{\Sigma}_n$ and $\mathbf{\Psi}_k$ in Equation 2.14, the measurement errors associated to objects and variables, respectively, are diagonal [24, 30]. In this algorithm, large variances (10^{10}) are assigned to the missing measurements, and ones to the available ones. Therefore, the inversion of matrices $\mathbf{\Sigma}_n$ and $\mathbf{\Psi}_k$ produces diagonal matrices with 1s and 0s. The ones serve to fit these specific measurements in the PCA and the 0s to disregard the missing measurements in the multivariate model.

Assuming that row $\mathbf{x}_n^{\mathrm{T}}$ has MD. The values in this vector can be rearranged to have the missing entries in its first R_n positions without loss of generality, and the remaining $K - R_n$ available values at the end. This partition in $\mathbf{x}_n^{\mathrm{T}}$, induces a partition in the $\mathbf{X}$ data set, being $\mathbf{X}^{\#}$ $(N \times R_n)$ the missing part and $\mathbf{X}^{*}$ $(N \times (K - R_n))$ the available part, according to row n. Additionally, this partition can be transferred into a SVD (or PCA) model, $\mathbf{X} = \mathbf{UDP}^{\mathrm{T}}$, being $\mathbf{P}^{\#}$ $(R_n \times A)$ the missing part of the loadings matrix, and $\mathbf{P}^{*}$ $((K - R_n) \times A)$ the available part. Figure 13.1 shows a scheme of this notation.

Using this partition, the inverse of matrix $\mathbf{\Sigma}_n$ can be written as:

$$\mathbf{\Sigma}_n^{-1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{K-R_n} \end{bmatrix} \tag{13.1}$$

where $\mathbf{I}_{K-R_n}$ is the identity matrix with $K - R_n$ rows/columns, according to the missing data pattern in $\mathbf{x}_n^{\mathrm{T}}$.

Substituting this expression in Equation 2.15, observation $\mathbf{x}_n^{\mathrm{T}}$ can be computed as:

$$\begin{aligned}\hat{\mathbf{x}}_n = \begin{bmatrix} \hat{\mathbf{x}}_n^{\#} \\ \hat{\mathbf{x}}_n^{*} \end{bmatrix} = \\ = \begin{bmatrix} \hat{\mathbf{P}}^{\#} \\ \hat{\mathbf{P}}^{*} \end{bmatrix} ([\hat{\mathbf{P}}^{\#\mathrm{T}} \hat{\mathbf{P}}^{*\mathrm{T}}] \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{K-R_n} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{P}}^{\#} \\ \hat{\mathbf{P}}^{*} \end{bmatrix})^{-1} [\hat{\mathbf{P}}^{\#\mathrm{T}} \hat{\mathbf{P}}^{*\mathrm{T}}] \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{K-R_n} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_n^{*} \end{bmatrix}) = \\ = \begin{bmatrix} \hat{\mathbf{P}}^{\#} \\ \hat{\mathbf{P}}^{*} \end{bmatrix} (\hat{\mathbf{P}}^{*\mathrm{T}} \hat{\mathbf{P}}^{*})^{-1} \hat{\mathbf{P}}^{*\mathrm{T}} \mathbf{x}_n^{*} = \begin{bmatrix} \hat{\mathbf{P}}^{\#} (\hat{\mathbf{P}}^{*\mathrm{T}} \hat{\mathbf{P}}^{*})^{-1} \hat{\mathbf{P}}^{*\mathrm{T}} \mathbf{x}_n^{*} \\ \hat{\mathbf{P}}^{*} (\hat{\mathbf{P}}^{*\mathrm{T}} \hat{\mathbf{P}}^{*})^{-1} \hat{\mathbf{P}}^{*\mathrm{T}} \mathbf{x}_n^{*} \end{bmatrix}\end{aligned} \tag{13.2}$$

Alternatively, the inverse of matrix $\mathbf{\Psi}_k$ can be written as:

$$\mathbf{\Psi}_k^{-1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N-R_k} \end{bmatrix} \tag{13.3}$$

where $\mathbf{I}_{N-R_k}$ is the identity matrix with $N - R_k$ rows/columns, according to column $\mathbf{x}_k$. Following Equation 2.16, $\hat{\mathbf{x}}_k$ is therefore computed as:

$$\begin{aligned}\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{x}}_k^{\#} \\ \hat{\mathbf{x}}_k^{*} \end{bmatrix} = \\ = \begin{bmatrix} \hat{\mathbf{U}}^{\#} \\ \hat{\mathbf{U}}^{*} \end{bmatrix} ([\hat{\mathbf{U}}^{\#\mathrm{T}} \hat{\mathbf{U}}^{*\mathrm{T}}] \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N-R_k} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}}^{\#} \\ \hat{\mathbf{U}}^{*} \end{bmatrix})^{-1} [\hat{\mathbf{U}}^{\#\mathrm{T}} \hat{\mathbf{U}}^{*\mathrm{T}}] \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N-R_k} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_k^{*} \end{bmatrix}) = \\ = \begin{bmatrix} \hat{\mathbf{U}}^{\#} \\ \hat{\mathbf{U}}^{*} \end{bmatrix} (\hat{\mathbf{U}}^{*\mathrm{T}} \hat{\mathbf{U}}^{*})^{-1} \hat{\mathbf{U}}^{*\mathrm{T}} \mathbf{x}_k^{*} = \begin{bmatrix} \hat{\mathbf{U}}^{\#} (\hat{\mathbf{U}}^{*\mathrm{T}} \hat{\mathbf{U}}^{*})^{-1} \hat{\mathbf{U}}^{*\mathrm{T}} \mathbf{x}_k^{*} \\ \hat{\mathbf{U}}^{*} (\hat{\mathbf{U}}^{*\mathrm{T}} \hat{\mathbf{U}}^{*})^{-1} \hat{\mathbf{U}}^{*\mathrm{T}} \mathbf{x}_k^{*} \end{bmatrix}\end{aligned} \tag{13.4}$$

where $\hat{\mathbf{U}}^{\#}$ $(R_k \times A)$ and $\hat{\mathbf{U}}^{*}$ $((N - R_k) \times A)$ are the missing and available parts of $\hat{\mathbf{U}}$.

The MLPCA imputation step of the missing values $\mathbf{x}_n^{\#\mathrm{T}}$ is the same as the PMP method for PCA-MB presented in Chapter 10. The main difference between MLPCA algorithm and PMP is that the former performs the imputation iteratively first by observations and then by variables, instead of only by observations, as PMP does. And additionally, the convergence in PMP is achieved based only on the imputed missing values, instead of the imputation of the available measurements, as it is in MLPCA.

As shown in Chapter 10, the imputation step in the adapted PMP algorithm for PCA-MB can be substituted by the regression-based methods presented in [27] (KDR and its variants, and TSR). Most of these methods showed a superior performance than PMP across several case studies. So, the idea here consists of adapting the alternating imputation of MLPCA algorithm to include the imputation step of the regression-based methods, thus proposing a ML framework: ML-KDR, ML-KDR with PCR, ML-KDR with PLS and ML-TSR.

The imputation step of the regression-based missing data methods is:

$$\hat{\mathbf{x}}_n = \begin{bmatrix} \hat{\mathbf{x}}_n^{\#} \\ \hat{\mathbf{x}}_n^{*} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{S}}^{\#*}\hat{\mathbf{L}}_n(\hat{\mathbf{L}}_n^{\mathrm{T}}\hat{\mathbf{S}}^{**}\hat{\mathbf{L}}_n)^{-1}\hat{\mathbf{L}}_n^{\mathrm{T}}\mathbf{x}_n^{*} \\ \hat{\mathbf{S}}^{**}\hat{\mathbf{L}}_n(\hat{\mathbf{L}}_n^{\mathrm{T}}\hat{\mathbf{S}}^{**}\hat{\mathbf{L}}_n)^{-1}\hat{\mathbf{L}}_n^{\mathrm{T}}\mathbf{x}_n^{*} \end{bmatrix} \tag{13.5}$$

where $\hat{\mathbf{S}}$ is the covariance matrix of $\hat{\mathbf{X}}$, and:

$$\hat{\mathbf{S}} = [\hat{\mathbf{X}}^{\#}\hat{\mathbf{X}}^{*}]^{\mathrm{T}}[\hat{\mathbf{X}}^{\#}\hat{\mathbf{X}}^{*}]/(N-1) = \begin{bmatrix} \hat{\mathbf{X}}^{\#\mathrm{T}}\hat{\mathbf{X}}^{\#} & \hat{\mathbf{X}}^{\#\mathrm{T}}\hat{\mathbf{X}}^{*} \\ \hat{\mathbf{X}}^{*\mathrm{T}}\hat{\mathbf{X}}^{\#} & \hat{\mathbf{X}}^{*\mathrm{T}}\hat{\mathbf{X}}^{*} \end{bmatrix}/(N-1) = \begin{bmatrix} \hat{\mathbf{S}}^{\#\#} & \hat{\mathbf{S}}^{\#*} \\ \hat{\mathbf{S}}^{*\#} & \hat{\mathbf{S}}^{**} \end{bmatrix} \tag{13.6}$$

The key matrix $\mathbf{L}$ in Equation 13.5 particularises which method of the ML framework is being used for the imputation: $\mathbf{L} = \mathbf{I}$ for KDR; $\mathbf{L} = \hat{\mathbf{V}}_{1:\rho}$ for KDR with PCR, where $\hat{\mathbf{V}}_{1:\rho}$ is the eigenvector matrix of $\hat{\mathbf{S}}^{**}$ and $\rho \leq rank(\hat{\mathbf{S}}^{**})$; $\mathbf{L} = \hat{\mathbf{W}}^{*}$ for KDR with PLS, where $\hat{\mathbf{R}}$ is the normalized weights matrix of the PLS model $\hat{\mathbf{T}}_{\mathrm{PLS}} = \hat{\mathbf{X}}^{*\mathrm{T}}\hat{\mathbf{R}} = \hat{\mathbf{X}}^{*\mathrm{T}}\hat{\mathbf{W}}(\hat{\mathbf{P}}^{\mathrm{T}}\hat{\mathbf{W}})^{-1}$; and $\mathbf{L} = \hat{\mathbf{P}}^{*}$ for TSR.

Therefore, to adapt the MLPCA original algorithm for MD [339] to use the regression-based methods, the imputation step (Equations 2.15-2.16) has to be substituted by:

$$\hat{\mathbf{x}}_n = \hat{\mathbf{S}}\mathbf{\Lambda}_n\hat{\mathbf{L}}_n(\hat{\mathbf{L}}_n^{\mathrm{T}}\mathbf{\Lambda}_n^{\mathrm{T}}\hat{\mathbf{S}}\mathbf{\Lambda}_n\hat{\mathbf{L}}_n)^{-1}\hat{\mathbf{L}}_n^{\mathrm{T}}\mathbf{\Lambda}_n^{\mathrm{T}}\mathbf{x}_n \tag{13.7}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{S}}\mathbf{\Phi}_k\hat{\mathbf{L}}_k(\hat{\mathbf{L}}_k^{\mathrm{T}}\mathbf{\Phi}_k^{\mathrm{T}}\hat{\mathbf{S}}\mathbf{\Phi}_k\hat{\mathbf{L}}_k)^{-1}\hat{\mathbf{L}}_k^{\mathrm{T}}\mathbf{\Phi}_k^{\mathrm{T}}\mathbf{x}_k \tag{13.8}$$

where $\mathbf{L}$ matrix is the same as in the regression-based framework, particularising for the missing data pattern in row n or column k. And:

$$\mathbf{\Lambda}_n = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{K-R_n} \end{bmatrix} \tag{13.9}$$

$$\mathbf{\Phi}_k = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{N-R_k} \end{bmatrix} \tag{13.10}$$

The equivalence between Equations 13.7 and 13.5 is shown in Section 13.7.

13.3 Data sets

Six data sets are used in the present chapter to compare the results of the different imputation methods included in the framework. The first data set contains FTIR miscroscopy spectra of a polymer laminate consisting of three layers: polyethylene, isophtalic polyester, and polyethylene terephthalate. The polymer was scanned in a seventeen point transect across the different layers, obtaining measurements from 81 wavelengths [342–344]. The second case study is a systems biology data set consisting of a set of measured and inferred fluxes from *P. pastoris* cultures on heterogeneous culture media, used in previous chapters. From the original data set with 3600 scenarios and 45 fluxes, a representative sample of 105 individuals is selected for the present comparative study. This data set has 3 biologically relevant PCs. Finally, a simulated data set, with 100 observations and 10 variables, is used to compare the performance of the different maximum likelihood methods [303, 304]. This data set has 4 eigenvalues (3, 2.5, 2 and 1.5) explaining 90% of the variance in data.

Three additional data sets are analysed here, taken from Chapter 10, where the adaptation of the regression based methods to PCA-MB was proposed: the Olive Oil data set [301], the Diesel NIR data set [302] and the simulated data set with 3 PCs [303, 304].

13.4 Comparative study

The same performance criteria applied in Chapter 10 are used here, i.e. the MSPE and the cosine between the first loading vector obtained using the full data matrix and its corresponding one from the incomplete data set. The original regression-based framework methods use, as convergence criterion, the difference between consecutive imputations of missing values. Instead, MLPCA use the difference between the available measurements and their predictions from the current PCA model. For this, the MSPEs for the available measurements and the missing ones separately, using Equation 10.10, are shown.

Six different levels of missing values are generated for all data sets, ranging from 10 to 60% of MD. Also, 50 different MD patterns are generated for each percentage of missing data, in order to build CI for the MSPEs.

Finally, to compare the different approaches the same ANOVA applied in Chapter 10 is used here, using the LSD intervals to assess the statistically significant differences.

13.5 Results

In this section the results of the comparative study are presented. However, we decided to exclude the results of ML-KDR, ML-KDR with PCR and ML-KDR with PLS due to large computation times, something already observed in Chapter 10, and due to the instability of some of them, especially ML-KDR (also observed before with KDR) and ML-KDR with PLS. Therefore, the results of MLPCA, ML-TSR, TSR and PMP are shown, in order to answer the three research questions posed in the Section 13.1.

13.5.1 FTIR microspectroscopy

Regarding Figure 13.2A, there exist no statistical differences between MLPCA and ML-TSR in all percentages of MD. TSR and PMP statistically outperform both ML approaches for low percentages of MD (10-20%). From 50% onwards, TSR is superior to PMP, MLPCA and ML-TSR. The cosines shown in Figure 13.2B are coherent with the results of the MSPEs, having TSR the highest cosines from 30% to 60%.

The results in Figure 13.2C show that TSR and PMP are superior to the ML approaches in terms of the measured values, which implies that the PCA model fitted once the data is imputed with these methods is closer to the original one than using ML estimations. Figure 13.2D is indeed very similar to Figure 13.2A, due to the fact that the errors in the imputed values between the true PCA model and the imputed one are way larger than in the measured values, as expected.

13.5.2 *P. pastoris* cultures on heterogeneous culture media

The results with the *P. pastoris* data set are similar to the previous ones, both in MSPEs and cosines (see Figure 13.3A-13.3B). TSR and PMP achieve the statistically best performance from 20%-40% of MD; and again, from 50% onwards, TSR becomes the best approach, being PMP superior to MLPCA and ML-TSR. The performances of TSR and PMP are indeed coherent with the results observed in Chapter 10. The cosines shown in Figure 13.3B are coherent with the MSPE

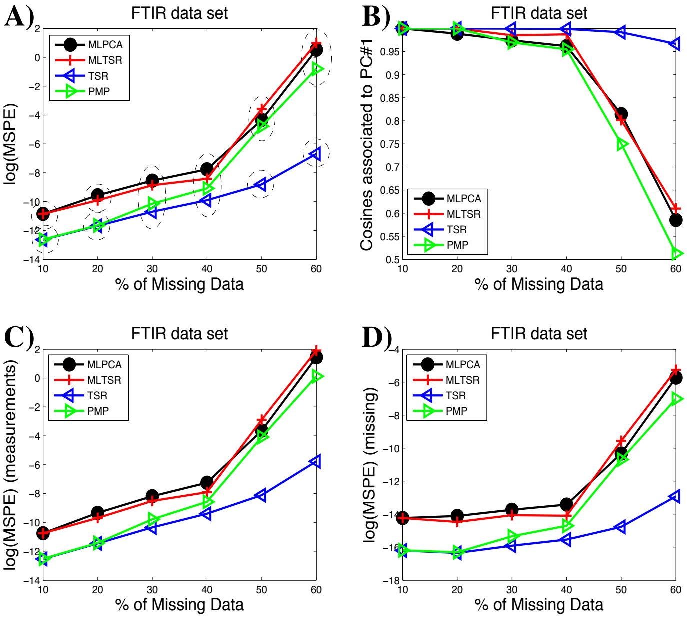


Figure 13.2: FTIR data set results. A) Logarithm of the MSPE for all measurements. B) Cosines associated to the first PC. C) Logarithm of the MSPE for the available measurements. D) Logarithm of the MSPE for the missing data. The dashed ellipses in a) mark the statistically significant differences between groups of methods. In A) TSR is statistically superior to MLPCA with 30-40% of MD. However, since there is no method statistically significant from all the rest, a single dashed ellipse encloses all of them.

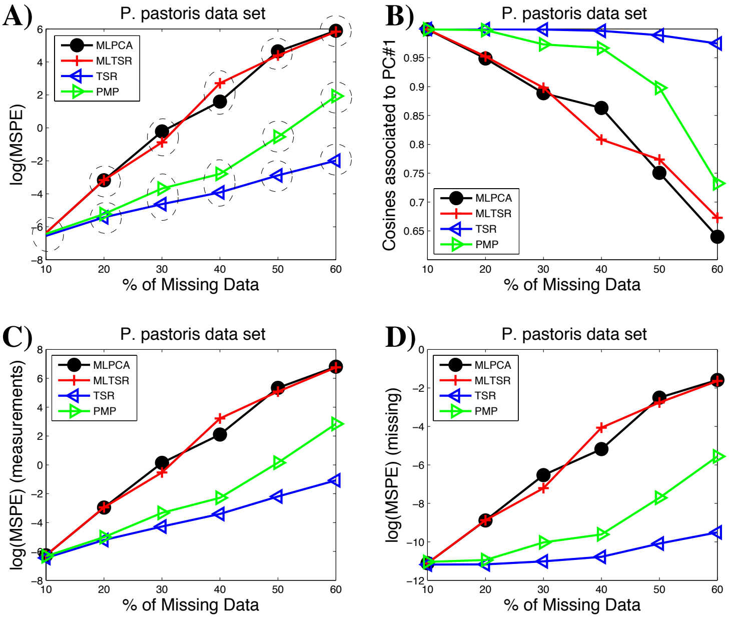


Figure 13.3: *P. pastoris* data set results. More details in Figure 13.2.

values. The lower is the logarithm of the MSPE, the closer are the loading vectors of the reconstructed matrix to the actual ones.

Regarding Figures 13.3C-13.3D, the performance of all methods is also similar to the first example. For low percentages of MD, the differences among methods are smaller in the measured values, but from 30% of MD onwards, the PCA model obtained with TSR imputation resembles more to the real one.

13.5.3 Simulated data set

In the Simulated data set with 4 PCs, the differences among TSR, ML-TSR, PMP and MLPCA are not statistically significant for low percentages of missing values (10-20%) (see Figure 13.4A). With 30% of MD, TSR becomes statistically the best method and PMP the worst one. This is something that was observed in Chapter 10, also using a simulated data set [303, 304]. The higher is the percentage of missing data, the more difficult is to impute properly for PMP. For higher percentages (30-60%), there are statistical differences among all methods: TSR maintains the best performance, followed by ML-TSR, MLPCA and PMP. There exist differences between MLPCA and ML-TSR, being the latter statistically superior. These differences in the MSPEs can also be seen in Figure 13.4B, where

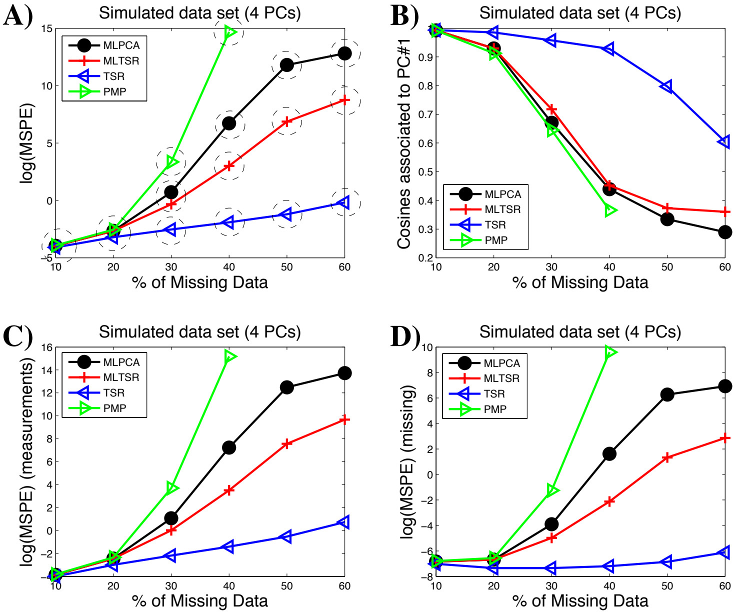


Figure 13.4: Simulated data set results. More details in Figure 13.2.

all methods but TSR show huge deviations from the true principal coordinate of the data with low-medium percentages of MD (10-40%).

In this third example the differences among methods regarding the measured values are narrower (see Figure 13.4C), but still showing the superiority of TSR.

13.5.4 Additional data sets

Three more data sets are used to compare the performance of the ML-based methods against PMP and TSR in its original form: the olive oil data set, the diesel NIR data set, and a 3-component simulated data set. The figures containing the logarithm of the MSPEs and the cosines associated to the first component are available in Section 13.8.

Summarizing the results, in these data sets the performance of TSR is statistically superior to PMP (as proven in Chapter 10), and to MLPCA and ML-TSR for medium-high percentages (30-60%) and also for low percentages (10-20%) in the olive oil and diesel NIR data set. Also, the reconstruction of the available measurements with TSR is more similar to the PCA on complete data than the ML-based approaches in both data sets. These results are coherent with sections 13.5.1-13.5.2. Comparing ML-TSR and MLPCA, the former yields better results than MLPCA for high percentages of missing data (50-60%) in the 3-component simulated data set, as happened in section 13.5.3 with the 4-component simulated data set.

13.6 Conclusions

To conclude, it is worth to remember the research questions posed at the beginning of the chapter:

- Are the imputed values of MLPCA and PMP for MB equal? The answer is no. The PMP imputation step performed alternatively by rows and columns in MLPCA drives the imputation in a different direction than performing it only observation-wise, as PMP does. Based on the six data sets analysed here, PMP, if converges, has better results than MLPCA. However, PMP suffered from convergence problems in some case studies, while MLPCA converge in all data sets and all MD percentages.

- Does ML-TSR outperform the imputation of MLPCA? The answer, based on the case studies analysed here, is that when the latent structure is complex, and the percentage of missing data is high, ML-TSR may outperform MLPCA. In other cases, the overall results have no statistically significant differences. However, MLPCA tends to be between 2-5 times faster than ML-TSR.

- Does MLPCA or ML-TSR outperform the original TSR algorithm? The answer is no. TSR outperforms the ML approaches for medium-high percentages of missing data. For low percentages, depending on the case study analysed, it is statistically superior or there exist no statistical difference compared to the other methods.

Finally, the use of TSR over MLPCA for PCA-MB with MD is recommended, since both the reconstruction of the available and imputed values is statistically more accurate than using MLPCA or ML-TSR.

13.7 Appendix A. Regression-based imputation step in MLPCA.

The equivalence between Equations 13.7 and 13.5 is proven here. Assuming that the values in row $\mathbf{x}_n^{\mathrm{T}}$ are rearranged to have the R_n missing values, $\mathbf{x}_n^{\#\mathrm{T}}$, at the first positions, and the remaining $K-R_n$ available ones, $\mathbf{x}_n^{\#\mathrm{T}}$, at the end.Equation 13.1 can be used in Equation 13.7 to introduce the extension of the missing data partition, $\hat{\mathbf{X}} = [\hat{\mathbf{X}}^{\#}\hat{\mathbf{X}}^{*}]$. Bearing in mind that the decomposition of the covariance matrix of $\hat{\mathbf{X}}$ (see Equation 13.6), and matrix $\mathbf{\Lambda}_n$ (Equation 13.9), Equation 13.7 can be written as:

$$
\begin{aligned}
\hat{\mathbf{x}}_n = \hat{\mathbf{S}}\mathbf{\Lambda}_n\hat{\mathbf{L}}_n(\hat{\mathbf{L}}_n^{\mathrm{T}}\mathbf{\Lambda}_n^{\mathrm{T}}\hat{\mathbf{S}}\mathbf{\Lambda}_n\hat{\mathbf{L}}_n)^{-1}\hat{\mathbf{L}}_n^{\mathrm{T}}\mathbf{\Lambda}_n^{\mathrm{T}}\hat{\mathbf{x}}_n = \\
= \begin{bmatrix} \hat{\mathbf{S}}^{\#\#} & \hat{\mathbf{S}}^{\#*} \\ \hat{\mathbf{S}}^{*\#} & \hat{\mathbf{S}}^{**} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{K-R_n} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{K-R_n} \end{bmatrix} \mathbf{L}_n \\
(\mathbf{L}_n^{\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{I}_{K-R_n} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{S}}^{\#\#} & \hat{\mathbf{S}}^{\#*} \\ \hat{\mathbf{S}}^{*\#} & \hat{\mathbf{S}}^{**} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{K-R_n} \end{bmatrix} \mathbf{L}_n)^{-1} \\
\mathbf{L}_n^{\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{I}_{K-R_n} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_n^* \end{bmatrix} = \\
= \begin{bmatrix} \hat{\mathbf{S}}^{\#\#} & \hat{\mathbf{S}}^{\#*} \\ \hat{\mathbf{S}}^{*\#} & \hat{\mathbf{S}}^{**} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{L}_n \end{bmatrix} (\begin{bmatrix} \mathbf{0} & \mathbf{L}_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{S}}^{\#\#} & \hat{\mathbf{S}}^{\#*} \\ \hat{\mathbf{S}}^{*\#} & \hat{\mathbf{S}}^{**} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{L}_n \end{bmatrix})^{-1} \begin{bmatrix} \mathbf{0} & \mathbf{L}_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_n^* \end{bmatrix} = \\
= \begin{bmatrix} \hat{\mathbf{S}}^{\#*}\hat{\mathbf{L}}_n(\hat{\mathbf{L}}_n^{\mathrm{T}}\hat{\mathbf{S}}^{**}\hat{\mathbf{L}}_n)^{-1}\hat{\mathbf{L}}_n^{\mathrm{T}}\mathbf{x}_n^* \\ \hat{\mathbf{S}}^{**}\hat{\mathbf{L}}_n(\hat{\mathbf{L}}_n^{\mathrm{T}}\hat{\mathbf{S}}^{**}\hat{\mathbf{L}}_n)^{-1}\hat{\mathbf{L}}_n^{\mathrm{T}}\mathbf{x}_n^* \end{bmatrix}
\end{aligned}
\tag{13.11}
$$

The proof using Equation 13.8 is analogous; substituting $\hat{\mathbf{x}}_n$ by $\hat{\mathbf{x}}_k$, changing the subindices n by k and the matrix $\mathbf{\Lambda}_n$ by $\mathbf{\Phi}_k$, and bearing in mind that the $\mathbf{L}_k$ matrix is obtained using the MD pattern of $\hat{\mathbf{x}}_k$.

13.8 Appendix B. Additional figures.

Figures 13.5-13.7 show the results of the comparative study proposed in this chapter with the Olive Oil, the Diesel NIR and the 3-component Simulated data set.

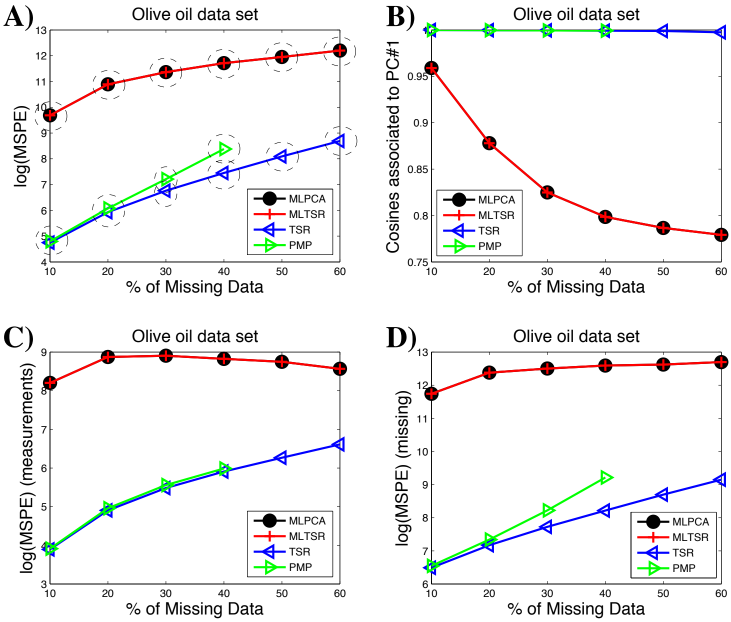


Figure 13.5: Olive Oil data set results. More details in Figure 13.2.

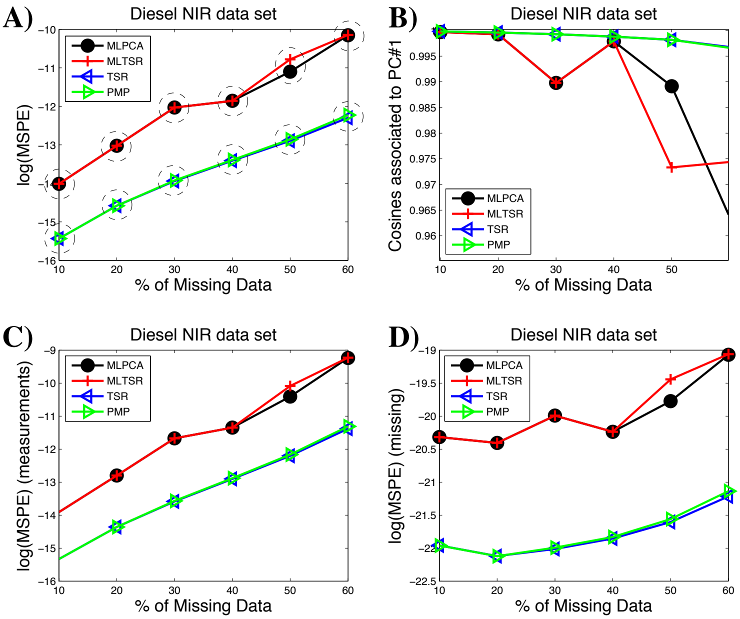


Figure 13.6: Diesel NIR data set results. More details in Figure 13.2.

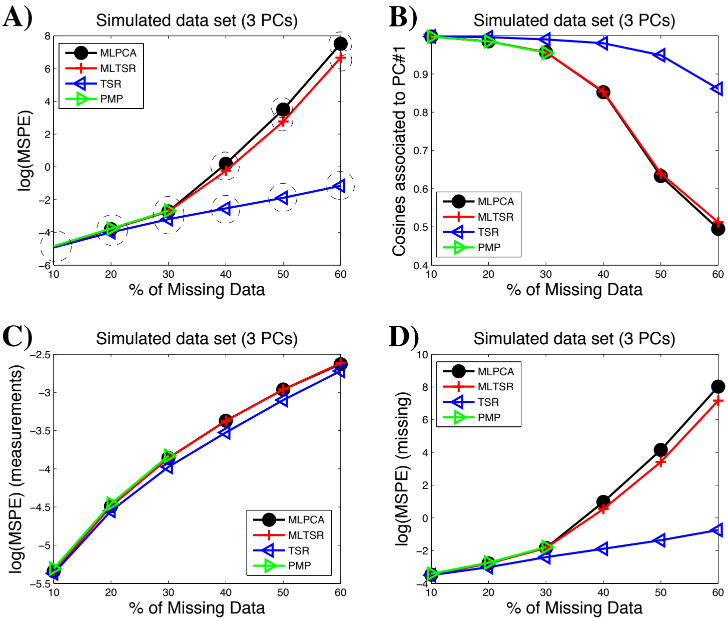


Figure 13.7: 3-component simulated data set results. More details in Figure 13.2.

Chapter 14

Calibration transfer between near infrared instruments

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

[11] Folch-Fortuny, A., Vitale, R., de Noord, O.E. & Ferrer, A. Calibration transfer between NIR spectrometers: new proposals and a comparative study. *Journal of Chemometrics*, accepted.

14.1 Introduction

Multivariate calibration is of crucial importance when interpretable information needs to be extracted from complex spectroscopic signals in chemometrics and systems biology. Numerous applications of e.g. PCR or PLS aimed at this end have been reported over the past decades [345, 346]. Nevertheless, a practical limitation to the employment of such techniques shows up when existing calibration models are applied to measurements recorded by a new instrument and/or in different environmental conditions. In fact, even very similar spectrometers generally exhibit strong variations in their responses, which may jeopardise this so-called calibration transfer.

Several methods have been proposed to overcome this subtle issue and avoid at the same time an expensive and time-consuming full recalibration, using the newly acquired spectral profiles. One of these approaches consists in updating the calibration model by merging measurements collected by both the first or *master* and the second or *slave* spectrometer. However, that is commonly effective only when the two sets of spectral profiles are rather similar [347]. Among all the other strategies proposed in the scientific literature, piecewise direct standardisation (PDS) [348] is considered as the reference for novel techniques due to its local and multivariate nature [347, 349–351]. PDS basically transforms the spectra recorded by the slave instrument so that its spectral response matches the one of the master instrument. This allows any calibration model, built on the data resulting from the master spectrometer, to be used for the analysis of those acquired by the slave apparatus.

From a slightly different perspective, the transfer of a calibration model from a NIR spectrometer to another can be looked at as a missing data imputation problem. In this circumstance, all the information contained in the available master and slave spectra can be exploited to entirely reconstruct the profiles associated to those samples that were not analysed by the slave instrument. These profiles can be then utilised for fitting an improved predictive model, suitable for the assessment of future incoming recordings. MLPCA [339] has been the first computational methodology to be applied for solving the calibration transfer issue in this peculiar fashion.

In this chapter, two innovative strategies to perform calibration transfer based on TSR algorithm for PCA-MB with MD and JYPLS are proposed. Specifically, their performance is be assessed and compared here to MLPCA and PDS in two real case-studies, in which the same set of samples were characterised by two different NIR spectrometers.

The chapter is structured as follows. Section 14.2 presents the spectral measurements used in this chapter. Sections 14.3-14.4 describe PDS, the adaptation of MLPCA, TSR and JYPLS to solve the calibration transfer problem and the comparative study. The results shown in Section 14.5 are then discussed in Section 14.6. Finally, some conclusions are drawn on Section 14.7.

14.2 Materials

The first dataset analysed here contains 60 spectra measured on 30 pseudo-gasoline samples within 800 and 1600 nm (401 scanned wavelengths, 30 spectra per instrument). Heptane, iso-octane, toluene, xylene and decane concentration are the properties of interest to be predicted. The second relates to 80 corn samples, whose spectral profiles were registered within 1100 and 2498 nm (700 scanned wavelengths for a total number of 160 spectra, 80 per each spectrometer). The response variables are moisture, oil, protein and starch content. Both datasets have been widely used to compare calibration transfer methods [352–354]. The gasoline dataset is included in the PLS Toolbox for MATLAB [169], the corn dataset can be downloaded from `http://www.eigenvector.com/data`.

14.3 Methods

14.3.1 Piecewise direct standardisation (PDS)

PDS executes a series of local linear transformation of the spectra collected by the slave instrument to subsequently allow the calibration model built for the master spectrometer to be exploited for prediction purposes. Specifically, at each kth wavelength, the whole set of absorbance values registered by the master instrument ($\mathbf{x}_{\mathrm{a},k}$) are related by PCR to a specific spectral window of the profiles of the same samples collected by the slave spectrometer $\mathbf{X}_{\mathrm{b},k}$ ($N \times K$):

$$\mathbf{x}_{\mathrm{a},k} = \mathbf{1}_N b_k + \mathbf{X}_{\mathrm{b},k}\mathbf{f}_k \tag{14.1}$$

Incoming slave instrument data are then adjusted through the estimated standardisation parameters, $\mathbf{f}_k$ and b_k. In this chapter, PDS is applied as coded in the PLS Toolbox [169]: all the PCs, whose eigenvalue (divided by the first one) are larger than 0.0001, are included in each local regression model. On the other hand, the spectral window width is automatically optimised within the modelling procedure.

14.3.2 MLPCA and TSR

To transfer a calibration model using MLPCA, the complete set of N_a master spectra, $\mathbf{X}_\mathrm{a}$ ($N_\mathrm{a} \times K_\mathrm{a}$), has to be concatenated with the N_b measurements collected by the slave instrument, $\mathbf{X}_\mathrm{b}$ ($N_\mathrm{b} \times K_\mathrm{b}$). An augmented data matrix $\mathbf{X}_\mathrm{ab}$ ($N_\mathrm{a} \times (K_\mathrm{a} + K_\mathrm{b})$) is then constructed, where the unrecorded slave profiles are missing (see Figure 14.1, *Imputation* box). In other words, if the sample associated to the nth row of $\mathbf{X}_\mathrm{ab}$ has not been analysed by the slave spectrometer, the available part of the row, $\mathbf{x}_n^{*\mathrm{T}}$, denotes the available master spectrum, while the missing part, $\mathbf{x}_n^{\#\mathrm{T}}$, denotes the missing slave spectrum. $\mathbf{X}_\mathrm{ab}$ is finally subjected to MLPCA.

Calibration transfer by TSR is achieved in the same way as for MLPCA, that is building the augmented array $\mathbf{X}_\mathrm{ab}$ and inputing it to the computational procedure described before. The TSR version for PCA-MB with MD described in Chapter 10 is used here.

14.3.3 JYPLS

Until now, JYPLS has been mainly resorted to for product transfer between different production sites, but here its application is extended to cases in which the common sources of variation underlying measurements resulting from multiple instruments and mostly related to specific properties of interest need to be modelled (i.e. calibration transfer). To this end, two possible JYPLS-based computing strategies are implemented, namely JYPLS-noinv and JYPLS-inv.

- JYPLS-noinv - Let $\mathbf{X}_\mathrm{a}$ contain the spectra collected by the master spectrometer and $\mathbf{X}_\mathrm{b}$ those registered by the slave one. Let $\mathbf{Y}_\mathrm{a}$ and $\mathbf{Y}_\mathrm{b}$ be the matrices including the measured dependent variables, noticing that the rows of $\mathbf{Y}_\mathrm{b}$ are also contained in $\mathbf{Y}_\mathrm{a}$, as they relate to samples analysed by both the slave and the master instrument. Once built a JYPLS model as in Equations 2.24-2.28 (see Figure 14.1, *Model transfer* box), the responses for new samples characterised by the second apparatus, $\mathbf{Y}_\mathrm{b,new}$, can be predicted from their spectral profiles, $\mathbf{X}_\mathrm{b,new}$, as (see Figure 14.1, *External validation II* box):

$$\mathbf{Y}_\mathrm{b,new} = \mathbf{X}_\mathrm{b,new}\mathbf{W}_\mathrm{b}^{*}\mathbf{Q}^\mathrm{T} \tag{14.2}$$

  where $\mathbf{Q}$ and $\mathbf{W}^*$ are obtained from JYPLS algorithm.

- JYPLS-inv - On the other hand, as for TSR, spectra unrecorded by the slave instrument can be reconstructed, provided they are associated to samples analysed by the master one and whose response values ($\mathbf{Y}_\mathrm{b,unrecorded}$) are then present in $\mathbf{Y}_\mathrm{a}$, by the following inversion (see Figure 14.1, *Model inversion* box):

$$\mathbf{X}_{\mathrm{b,unrecorded}} = \mathbf{Y}_{\mathrm{b,unrecorded}}(\mathbf{Q}\mathbf{Q}^{\mathrm{T}})^{\dagger}\mathbf{Q}\mathbf{P}_{\mathrm{b}}^{\mathrm{T}} \tag{14.3}$$

where † denotes the pseudoinverse [355]. Such *imputed* spectra, fused to $\mathbf{X}_{\mathrm{b}}$, are then appealed to for fitting an improved PLS predictive model (see Figure 14.1, *Model calibration* box), suitable for the assessment of future incoming data (see Figure 14.1, *External validation I* box).

14.4 Modelling procedure

The comparative study among PDS, MLPCA, TSR and JYPLS is carried out according to a 5-step procedure (see Figure 14.1):

1. Both the master and slave instrument data blocks are randomly split into calibration (2 thirds of the original spectra) and validation (1 third of the original spectra) sets (see Figure 14.1, *Data split* box). 20 split rounds are conducted to prevent spurious results from being yielded.

2. Slave instrument calibration subsets of increasing size are generated to determine the minimum number of measurements needed to be collected for accomplishing an accurate calibration transfer. The samples belonging to each one of these subsets are selected by the Kennard-Stone (KS) algorithm [356], probably the most popular computational procedure for data-representative object identification [357, 358] (see Figure 14.1, *Sample selection* box). Here, KS is run on the scores of a PLS model resulting from the master spectrometer calibration data.

3. The four methods under study are then applied in the following fashion:

   - When TSR, MLPCA and JYPLS-inv are handled, the slave instrument calibration spectra left out of each subset are consecutively reconstructed as described before (see Figure 14.1, *Imputation*, *Model transfer* and *Model inversion* boxes). They are then merged with those belonging to the calibration subset to fit a new PLS regression model (see Figure 14.1, *Model calibration* box).

   - By JYPLS-noinv, predictive JYPLS models are constructed fusing both the master spectrometer calibration set and the different slave spectrometer calibration subsets (see Figure 14.1, *Model transfer* box).

   - The PDS standardisation is performed relating the slave instrument calibration subsets of spectra to their corresponding profiles registered by the master spectrometer (see Figure 14.1, *Parameter fitting* and *Standardisation* boxes). Notice that the properties of interest of the

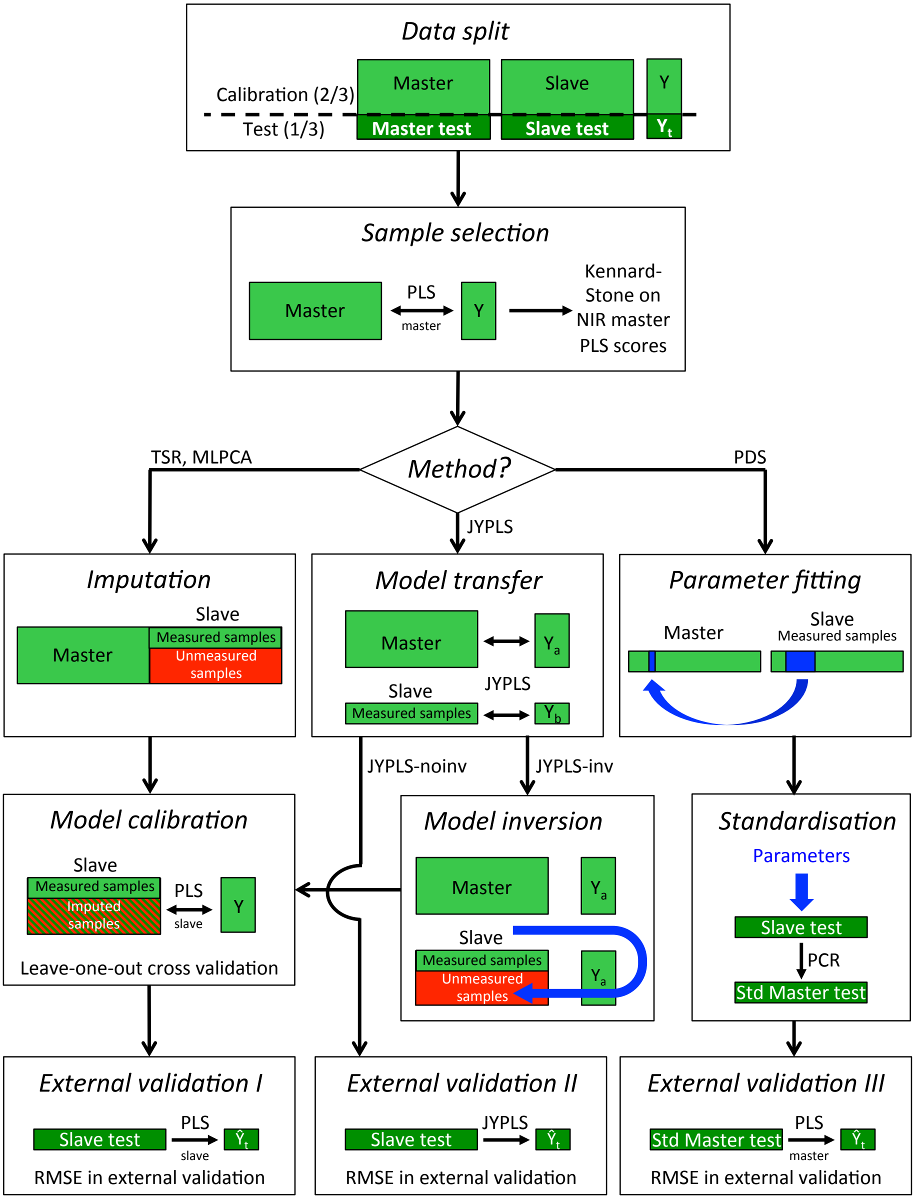


Figure 14.1: Flow-chart of the comparative study. Std stands for standardised. ^ refers to predicted values. Notice that part of the whole slave instrument calibration set is assumed to be unmeasured when addressing the calibration transfer.

corrected spectra are thereafter predicted by a PLS regression model built on the whole master instrument calibration set (see Figure 14.1, *External validation III* box).

In the various cases, the parameters to be optimised (number of components of the imputation model, number of components of the regression model, PDS spectral window width) are adjusted in order to minimise the average root mean square error in CV (RMSECV), defined as:

$$RMSECV = \frac{\sum_{m=1}^{M} \sqrt{\frac{\sum_{n=1}^{N}(y_{n,m}-\hat{y}_{n,m})^2}{N}}}{M} \tag{14.4}$$

where $y_{n,m}$ represents the actual value of the mth response variable associated to the nth calibration sample and $\hat{y}_{n,m}$ is its final prediction.

4. The performance of PDS, MLPCA, TSR and JYPLS were finally assessed according to the average root mean square error in prediction (RMSEP):

$$RMSEP = \frac{\sum_{m=1}^{M} \sqrt{\frac{\sum_{n'=1}^{N'}(y_{n',m}-\hat{y}_{n',m})^2}{N'}}}{M} \tag{14.5}$$

where $y_{n',m}$ represents the actual value of the k-th response variable associated to the n'th validation sample and $\hat{y}_{n',m}$ is its final prediction, while N' equals the total number of spectra included in the validation (see Figure 14.1, *External validation I*, *External validation II* and *External validation III* boxes). The reported RMSECV and RMSEP values concern autoscaled response variables owing to the differences in their original units of measurements.

5. Statistically significant differences among the considered approaches were finally evaluated via a mixed-effect ANOVA, as used in previous chapters of this thesis: calibration transfer technique, size of the slave instrument calibration subset and their interaction are fixed-effect factors, and split round is a random-effect factor, nested to the size of the slave instrument calibration subset). In case any effect or interaction was statistically significant, the 95% LSD (least significance difference) intervals are used.

14.5 Results

14.5.1 Gasoline dataset

For each spectrometer, 20 pseudo-gasoline samples are assigned to the calibration set and the remaining 10 to the validation set. 15 slave instrument calibration subsets, containing from 5 to 19 spectral profiles, are generated.

MD imputation

As TSR, JYPLS-inv and MLPCA rely on a preliminary MD imputation step, it is worth assessing the accuracy of the reconstruction of the unmeasured spectra, since they will be then resorted to for building the final predictive PLS model.

Figure 14.2 permits to compare original and imputed profiles for one of the split rounds. Their correlation and $\chi 2$ distance are represented in Figures 14.2A, 14.2D, 14.2G and 14.2B, 14.2E, 14.2H, respectively. Each line refers to the best model selected for one specific slave instrument calibration subset. The correlation was always higher than 0.9999 and the $\chi 2$ distance smaller than 0.001 for TSR and JYPLS-inv, while several issues appear when dealing with MLPCA. First, it often suffers from convergence problems (as already pointed out by Feudale *et al.* [347]), which dramatically slows the computational procedure down. Consequently, the reconstructed spectra are found to be considerably different from their actual profiles (see Figure 14.2F). For these reasons, MLPCA is not taken into account in the final study.

Comparative study

Figure 14.3A allows the performance of the different calibration transfer techniques under study to be examined. Each point in the plot represents the average RMSEP value, estimated from the 10-sample external validation set, across the 20 split rounds (for 5- to 19-sample slave instrument calibration subsets). As expected, for all the approaches, the higher the size of the slave instrument calibration subset, the lower the RMSEP.

As the effect of all the factors included in the ANOVA model was found to be statistically significant ($p-$value < 0.05), the 95% LSD intervals were calculated to point out existing differences among methods. For the sake of an easy visualisation, dashed-line ellipses are used in Figure 14.3A to highlight them. Specifically, methods embraced by the same ellipse show no statistical difference. On the other hand, methods embraced by different ellipses are statistically different.

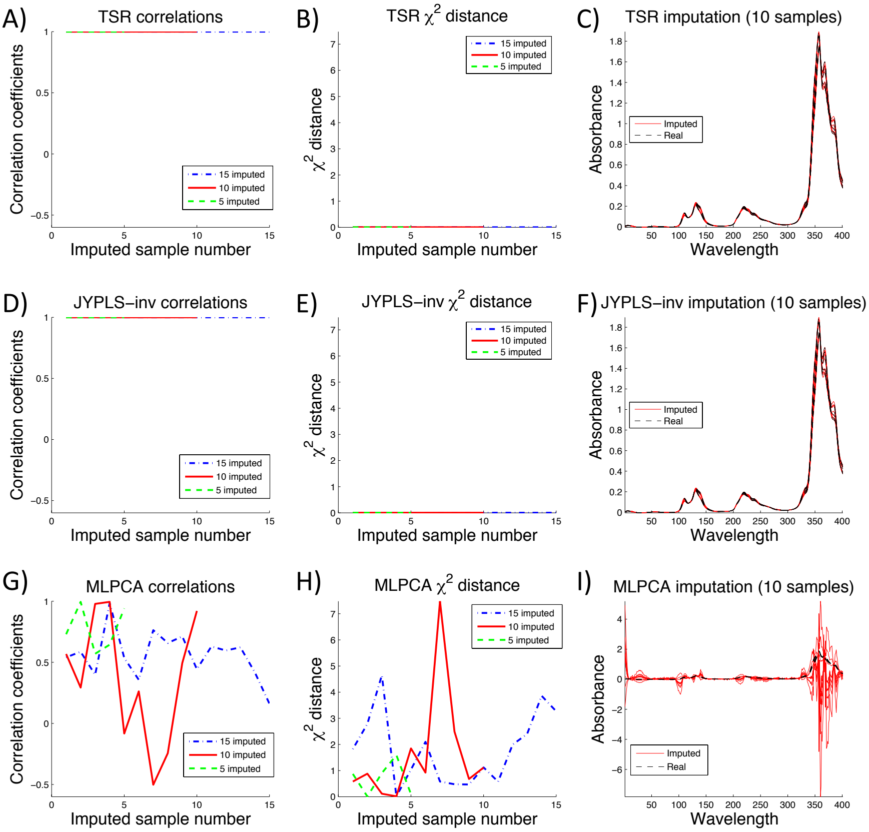


Figure 14.2: Gasoline dataset. A), D) and G) show the correlation coe cients between the original spectra and those imputed by TSR, JYPLS-inv and MLPCA, respectively. B), E) and H) represent their corresponding $\chi 2$ distance values. The dotted-dashed blue lines refer to the case in which the slave instrument calibration subset was constituted by 5 samples and 15 spectra were imputed. The solid red lines refer to the case in which the slave instrument calibration subset was constituted by 10 samples and 10 spectra were imputed. The dashed green lines refer to the case in which the slave instrument calibration subset was constituted by 15 samples and 5 spectra were imputed. C), F) and I) display the original and reconstructed profiles in the second of these three cases.

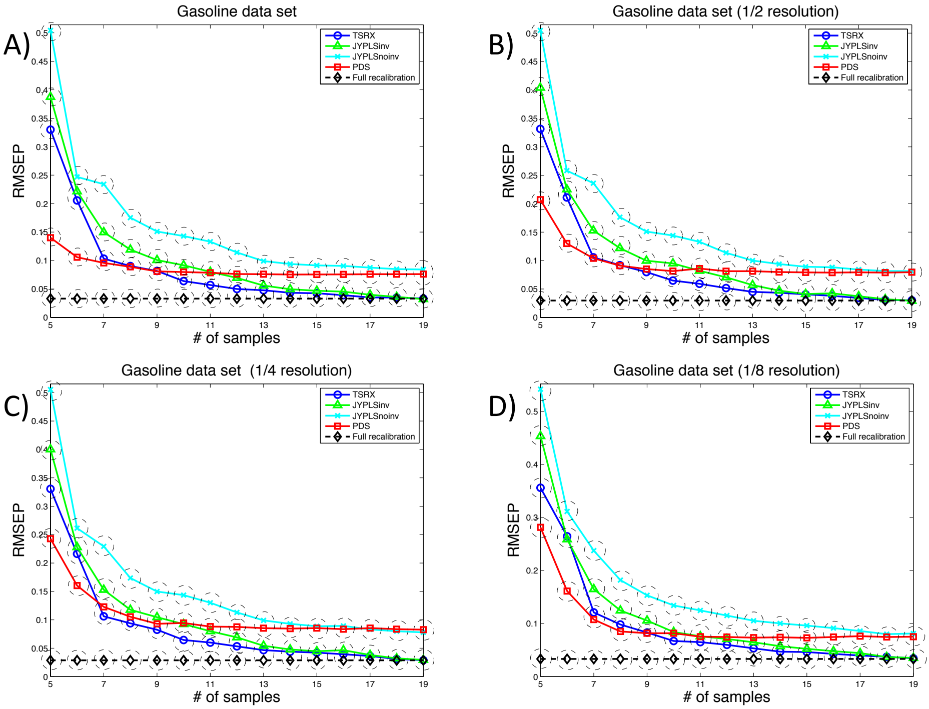


Figure 14.3: Gasoline dataset. RMSEP values obtained with A) the same spectral resolution for both instruments, B) $\frac{1}{2}$, C) $\frac{1}{4}$ and D) $\frac{1}{8}$ of the master instrument spectral resolution for the slave spectrometer. Dashed ellipses mark the statistically significant differences among groups of methods ($p-$value < 0.05)

Clearly, PDS guarantees the lowest RMSEP when the slave instrument calibration subset consisted of 5 or 6 samples. No statistically significant differences are detected between PDS and TSR for the slave instrument calibration subset of 7 samples and between PDS and JYPLS-inv when a 10-sample slave instrument calibration subset is considered. From 10 samples onwards, the RMSEP stabilizes around 0.09 for PDS, but it continuously decreases for TSR and JYPLS-inv, until reaching values around 0.05-0.06 (12-13 to 19 samples). The straight line in Figure 14.3 indicates the RMSEP value obtained when a full recalibration is performed, i.e. when the whole set of 20 slave instrument calibration samples is used to build a new predictive model. Although it cannot be directly compared to the outcomes resulting from PDS, TSR, JYPLS-inv and JYPLS-noinv, it eases the determination of the number of spectra needed to be collected by the slave spectrometer for generating no statistically significant differences with respect to full recalibration. TSR requires 12 spectra out of 20, while JYPLS-inv 13. On the other hand, PDS and JYPLS-noinv always show a statistically worse performance.

Instruments with different resolutions

A common situation faced by practitioners in industrial environments is transferring calibration models between instruments with diverse spectral resolution. This problem has already been addressed in [359], where the authors propose a novel PLS-based approach resulting in similar results as PDS.

Figures 14.3B-14.3D show the results of the whole analysis, conducted gradually reducing the spectral resolution of the slave instrument. The performance of the methods is basically the same as in the full resolution case described in the previous section. However, for PDS, a gradual decrease in the quality of the calibration transfer can be noticed. This effect will be much more evident in the corn example.

Sample selection effect

The effect of the slave spectrometer calibration subset sample selection is here assessed. 10 random selections are performed for one particular split round and the final RMSEP values are then compared to those obtained by preliminarily running KS. It is clear from Figure 14.4 that KS generally returned a lower RMSEP, very close to that achievable through a full recalibration. It then enabled a better calibration transfer plausibly due to the fact that it permits to choose a subset of samples, which is statistically representative of the experimental domain related to the spectral data collected by the master instrument. This is not necessarily the case when such a selection is carried out at random.

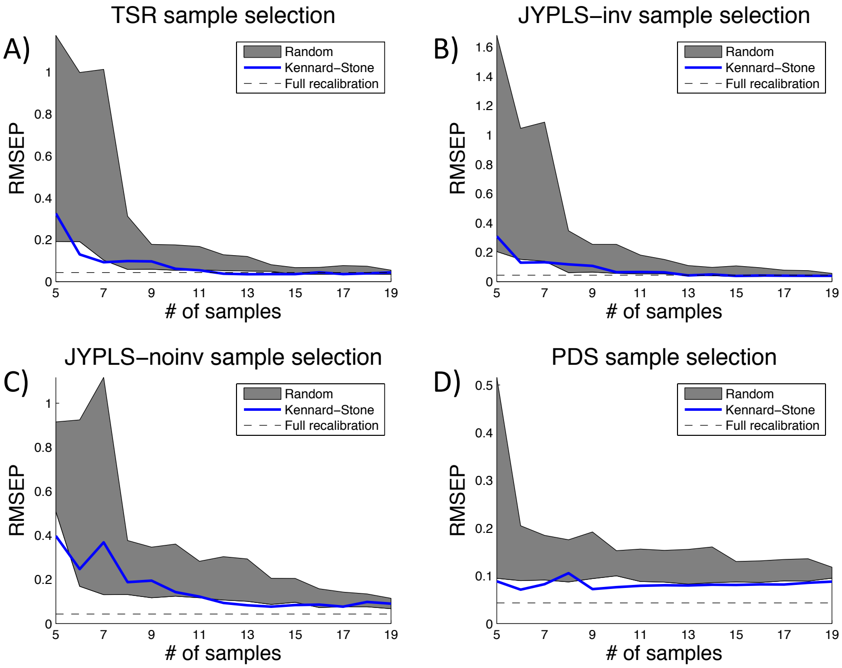


Figure 14.4: Gasoline dataset. Effect of the KS algorithm-based sample selection on the performance of the calibration transfer methods under study

14.5.2 Corn dataset

For each spectrometer, 54 corn samples are assigned to the calibration set and the remaining 26 to the validation set. 10 slave instrument calibration subsets, containing from 5 to 50 spectral profiles (5-spectra intervals), are generated.

Missing data imputation

Figure 14.5 permits to compare original and imputed corn sample spectral profiles for one split round. TSR preserves its reconstruction ability and MLPCA suffers from the same problems observed for the gasoline dataset. Regarding JYPLS-inv, the correlation coefficients/$\chi 2$ distance values were rather high/low, but the imputed spectra showed less variability than the real ones (see e.g. Figure 14.5F). This happens because the large difference in their offset is scarcely related to the properties to be predicted. As the imputation here involves the joint-**Y** loadings matrix **Q**, such difference is not transferred to the reconstructed spectra (see

Equation 2.24). Thus, one can think of JYPLS-inv as filtering spectral variations, which is uninteresting from a predictive point of view.

Comparative study

Existing differences among methods were investigated as in the previous case-study (also here the effect of all the ANOVA factors was found to be statistically significant). Figure 14.6A displays the results of the comparative study conducted on the corn dataset. Again, PDS shows a better performance for small slave calibration subsets (5-10 samples). For 20-25 samples, there are no statistical differences between PDS, TSR and the two JYPLS algorithms. Finally, from 30 samples onwards, the novel approaches outperform PDS, as happened in the gasoline data set. From 40 samples onwards, TSR, JYPLS-inv and JYPLS-noinv guarantee no significant differences with respect to full recalibration.

Instruments with different resolutions

In this case, the reduction of the spectral resolution of the slave instrument strongly affects the quality of the PDS-based calibration transfer. In fact, when the resolution of the slave spectrometer is decreased to $\frac{1}{8}$, even for small slave calibration subsets, the performance of PDS is statistically worse than the other compared approaches. On the other hand, TSR, JYPLS-inv and JYPLS-noinv are quite robust towards such change (see Figures 14.6B-14.6D).

Sample selection effect

The effect of the slave spectrometer calibration subset sample selection can be evaluated by looking at Figure 14.7. Here, especially when the size of such calibration subset is not particularly large, some random selection runs permit to obtain better results in terms of RMSEP. This might be related to the aforementioned weak relationship between spectral variations and properties of interest. However, when the number of calibration spectra recorded by the slave instrument increases, KS-based selection enabled better prediction than random ordering.

14.6 Discussion

When carrying out a calibration transfer with a very small slave instrument calibration subset (around 5-10 samples), PDS shows better or equal results, but its performance is far from being comparable to that guaranteed by a full recalibration. Nevertheless, when the size of the slave instrument calibration subset is enlarged, TSR and JYPLS-inv clearly outperform PDS. No evident conclusions

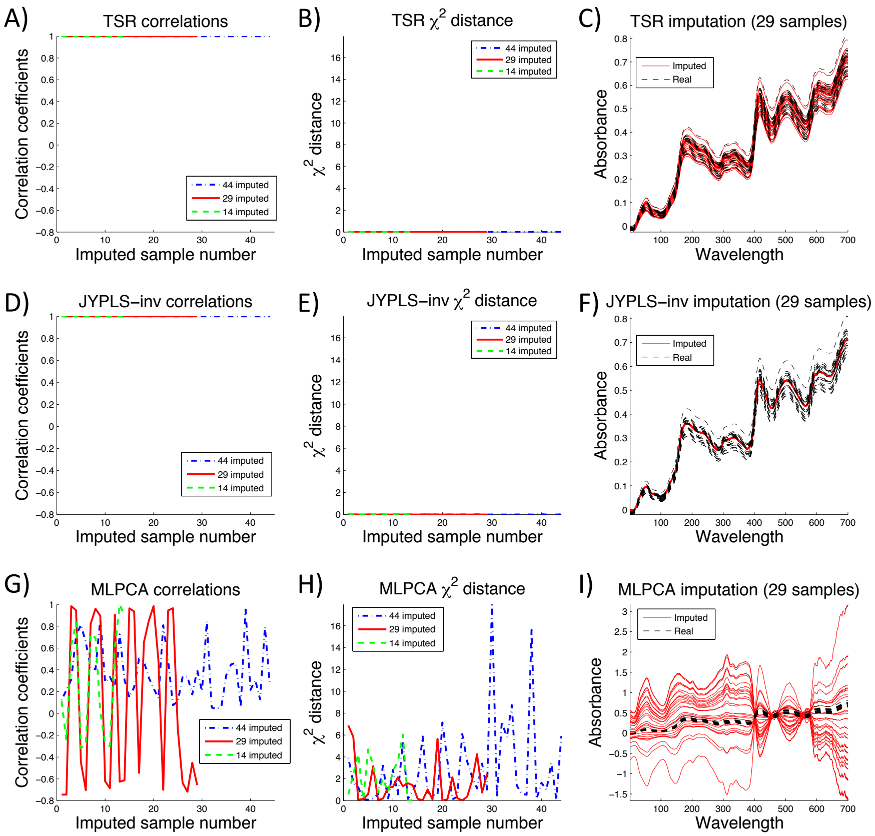


Figure 14.5: Corn dataset. A), D) and G) show the correlation coefficients between the original spectra and those imputed by TSR, JYPLS-inv and MLPCA, respectively. B), E) and H) represent their corresponding $\chi 2$ distance values. The dotted-dashed blue lines refer to the case in which the slave instrument calibration subset was constituted by 10 samples and 44 spectra were imputed. The solid red lines refer to the case in which the slave instrument calibration subset was constituted by 25 samples and 29 spectra were imputed. The dashed green lines refer to the case in which the slave instrument calibration subset was constituted by 40 samples and 14 spectra were imputed. C), F) and I) display the original and reconstructed profiles in the second of these three cases.

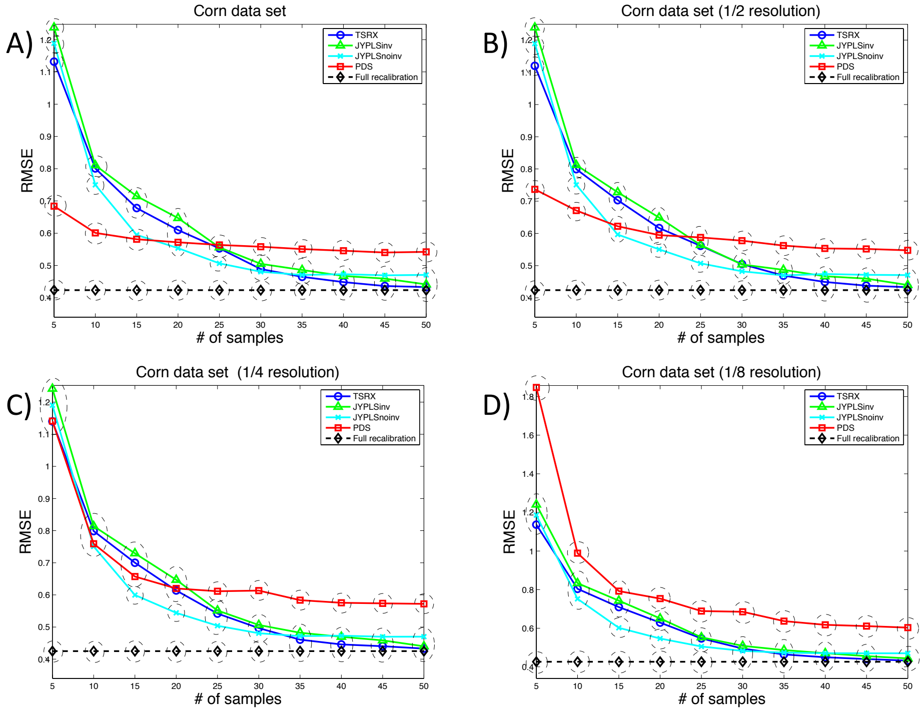


Figure 14.6: Corn dataset. RMSEP values obtained with A) the same spectral resolution for both instruments, B) $\frac{1}{2}$, C) $\frac{1}{4}$ and D) $\frac{1}{8}$ of the master instrument spectral resolution for the slave spectrometer. Dashed ellipses mark the statistically significant differences among groups of methods ($p-$value < 0.05)

can be drawn regarding the differences between PDS and JYPLS-noinv, as the quality of their outcomes changes depending on the analysed dataset.

The number of spectra to be collected by the slave spectrometer for a precise calibration transfer is also assessed. TSR and JYPLS-inv yield very similar results to full recalibration even if a part of the total number of the available spectra (around 10-40 samples) are included in the corresponding calibration subset. On the other hand, PDS never reaches such degree of accuracy. Concerning JYPLS-noinv, it is found to be, in general, as reliable as TSR and JYPLS-inv when the corn dataset is dealt with, but statistically worse in the gasoline case study.

PDS is strongly affected by the reduction of the spectral resolution of the slave instrument in the corn dataset, while TSR, JYPLS-inv and JYPLS-noinv seem not to suffer from the same issue.

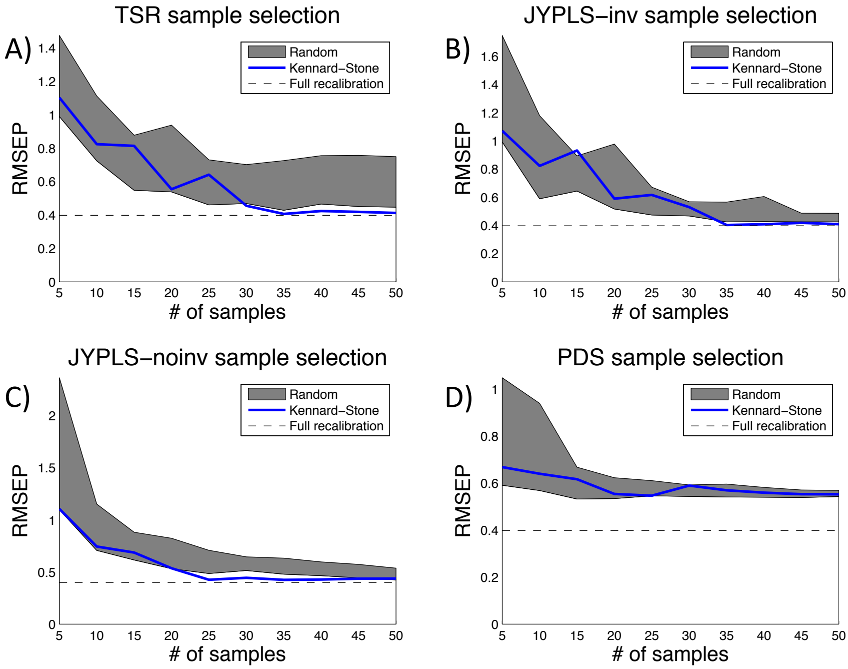


Figure 14.7: Corn dataset. Effect of the Kennard-Stone algorithm-based sample selection on the performance of the calibration transfer methods under study

In terms of unmeasured spectra reconstruction, TSR results in the best performance. In contrast, JYPLS-inv acts as a sort of filter removing the variations in the spectra not related to the properties to be predicted, consequently producing deviations from their original shape.

Finally, it is shown that selecting the samples using KS generally permits to achieve better results, regardless the calibration transfer technique.

14.7 Conclusions

Two novel methods to perform calibration transfer between NIR spectrometers, based on TSR and JYPLS, respectively, are proposed in this chapter. They outperform PDS and guarantee a very similar performance to that resulting from a full recalibration, when enough spectra collected by the slave instrument are available. Both approaches also show a sufficient robustness towards the reduction of its spectral resolution. In addition, TSR allows unmeasured spectra to be accurately imputed, while the inversion of the JYPLS models yields reconstructed spectral profiles filtered of all the variation not of interest from a predictive point of view.

Chapter 15

PLS model building with missing data

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

[12] Folch-Fortuny, A., Arteaga, F. & Ferrer, A. PLS model building with missing data: New algorithms and a comparative study. *Journal of Chemometrics*, submitted.

15.1 Introduction

To conclude Part III: Missing data, this chapter investigates how to build PLS models with MD. This problem is pervasive both in bioindustries, when missing values appear in historical batch or continuous data, and in research, when MD appear in predictors (independent variables) and responses (dependent variables) when collecting data, via e.g. experiments, for fitting regression models. This problem has been addressed in the literature using different approaches.

Probably the most used methods for PLS-MB with MD are the aforementioned IA and NIPALS, in their PLS versions. Being the default imputation procedures in many commercial software packages, such as ProMV, SIMCA-P, The Unscrambler [360] and PLS Toolbox. Another method has been recently proposed in the literature to address missing values in PLS-MB [361]. This method is based on an optimization procedure using an undeflated PLS algorithm (OUPLS). Regarding ME, the original TSR algorithm for PCA-ME was adapted to a PLS-ME environment in [70], with the aim of predicting the uncoming measurements and the future quality variables while the batch is still being processed. Also, as commented in [47], IA and NIPALS can be also used for PLS-ME.

After the good performance of TSR in PCA-MB, PCA-ME and PLS-ME, two novel versions of TSR are proposed in this chapter for PLS-MB with MD . Thus, TSR can be applied, from now on, to solve both MD problems (MB and ME) in exploratory and predictive models, as IA and NIPALS. For these methods, as with methods proposed in chapters 10 and 13, MCAR or MAR mechanisms are assumed for the MD. The first version of TSR presented here, TSR-1, is a direct adaptation of the algorithm for PCA-MB to PLS-MB, changing the data preprocessing within the algorithm. The second one, TSR-2, is an adaptation of the TSR algorithm for PLS-ME to PLS-MB, using the same rationale developed in Chapter 10 to adapt the regression-based framework methods from PCA-ME to PCA-MB. The other regression-based methods, KDR and its variants, are not adapted to a PLS-MB environment, since TSR has been proved a more efficient approach in chapters 10 and 13.

To test the novel TSR algorithms, a comparative study is presented here against other state-of-the-art methods commonly used by practitioners: NIPALS and IA. OUPLS is not used in the comparative for software availability problems. Other missing data imputation methods used in the literature for predictive modelling [61], such as the algorithm of Krzanowski based on SVD[62], GIP [63], MICE [64], and the regularized versions of the E-M algorithm: r-EM [65] and t-EM [66], are also not included here, since they consider only MD in the predictor variables, and thus, its comparison would be more appropriate with PCA-MB methods, as commented in Chapter 10.

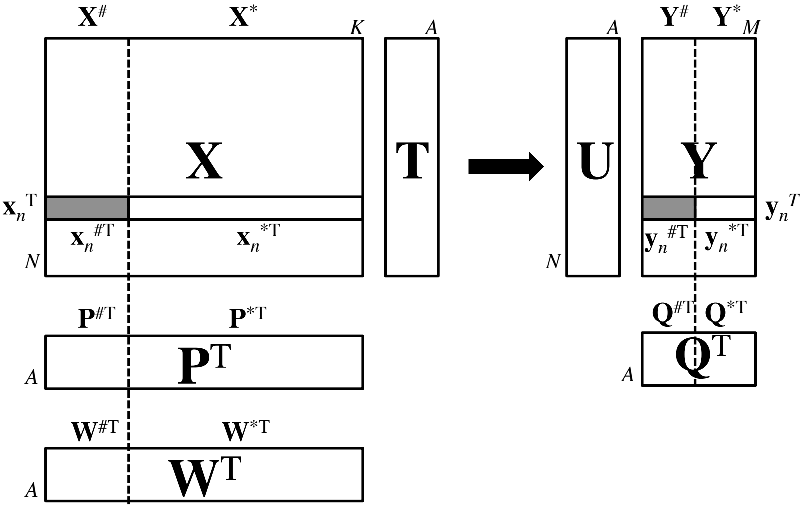


Figure 15.1: MD partition in PLS data matrices.

The aim of this chapter consists of providing researchers and practitioners with a ready-to-use MATLAB code to impute missing values in a regression environment, that is, using not only information of predictor and response matrices separatedly but exploiting the relationships among them. This way, the algorithms provided here can be used for fitting PLS models with MD or for imputing MD as a previous step of any other methodology (predictive or not). The TSR algorithms proposed here are freely available at `http://mseg.webs.upv.es`, under a GNU license.

The structure of this chapter is as follows. Section 15.2 explains how the two TSR algorithms for PLS-MB are built. Sections 15.3-15.4 describe the data sets and the performance criteria used in the comparative study. After showing the resuts in Section 15.5, Section 15.6 discusses on the methods performances.

15.2 Adaptations of TSR for PLS-MB

This chapter considers the same notation proposed in Chapter 10, that is, the missing data indicator matrix $\mathbf{M}$ and its complementary $\bar{\mathbf{M}}$. Missing and available values in data matrices are denoted with superindices * and $^\#$, respectively. Finally, the PLS normalized weights matrix is denoted as $\mathbf{R} = \mathbf{W}(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}$.

Figure 15.1 presents the data matrices involved in PLS-MB with missing values. MD is assumed to appear in the first positions of row n of $\mathbf{X}$ and $\mathbf{Y}$. The partitions of $\mathbf{x}_n^{\mathrm{T}}$ and $\mathbf{y}_n^{\mathrm{T}}$ are then transferred to matrices $\mathbf{P}$, $\mathbf{W}$ and $\mathbf{Q}$.

15.2.1 From PCA model building (TSR-1)

PLS aims at finding the latent space of $\mathbf{X}$ that better explains $\mathbf{Y}$ by maximising the covariance between both data matrices. Thus, one could argue that one way of meeting this objective consists of augmenting the $\mathbf{X}$ data set with the $\mathbf{Y}$ matrix and fit a PCA model, which in this case would maximise the covariance of matrix $[\mathbf{X}\ \mathbf{Y}]$:

$$[\mathbf{X}\ \mathbf{Y}]^{\mathrm{T}}[\mathbf{X}\ \mathbf{Y}] = \begin{bmatrix}\mathbf{X}^{\mathrm{T}}\\ \mathbf{Y}^{\mathrm{T}}\end{bmatrix}[\mathbf{X}\ \mathbf{Y}] = \begin{bmatrix}\mathbf{X}^{\mathrm{T}}\mathbf{X} & \mathbf{X}^{\mathrm{T}}\mathbf{Y}\\ \mathbf{Y}^{\mathrm{T}}\mathbf{X} & \mathbf{Y}^{\mathrm{T}}\mathbf{Y}\end{bmatrix} \tag{15.1}$$

Following this idea, the TSR algorithm for PCA-MB with MD can be used directly for PLS-MB purposes simply by using the aforementioned augmented matrix as input.

Figure 15.2 shows a scheme of the adapted TSR algorithm. The modifications with respect to Figure 10.5 in Chapter 10 are: i) the input data is now the augmented matrix $[\mathbf{X}\ \mathbf{Y}]$, ii) the data is now autoscaled at each step t, and iii) the last step of the algorithm consists of fitting a PLS model to obtain matrices $\mathbf{T}$, $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$. This TSR version for PLS-MB is from now on denoted as TSR-1. This iterative imputation procedure stops when the imputed values stabilize. At this step, the data is decomposed again in $\mathbf{X}$ and $\mathbf{Y}$ matrices in order to compute the PLS model.

15.2.2 From PLS model exploitation (TSR-2)

Two issues arise in the straightforward adaptation of TSR-1. Firstly, even pursuing a similar objective, a PCA on $[\mathbf{X}\ \mathbf{Y}]$ gives a different solution than a PLS, so a PCA-based model for MD may offer a different imputation than using a method fitting inner PLS models in the algorithm, as NIPALS and IA do. Secondly, the number of components may be different in PCA than in PLS. Therefore, if the number of PLS components are used to fit a PCA model using the augmented matrix, overfitting or underfitting problems may appear.

A TSR version for PLS-ME, using PLS as the core model, can be derived from the original idea of the algorithm for PCA-ME. This was done in [70] to propose a model to estimate the missing values in real-time batch monitoring. TSR for PLS-ME aims at estimating the complete scores of new observations using the information contained in the scores of the submatrix of $\mathbf{X}$ corresponding to the available data in the reference observation. Using matrix $\mathbf{T}$ from the complete model, this can be expressed as:

$$\mathbf{T} = \mathbf{T}^{*}\mathbf{B} + \mathbf{E} \tag{15.2}$$

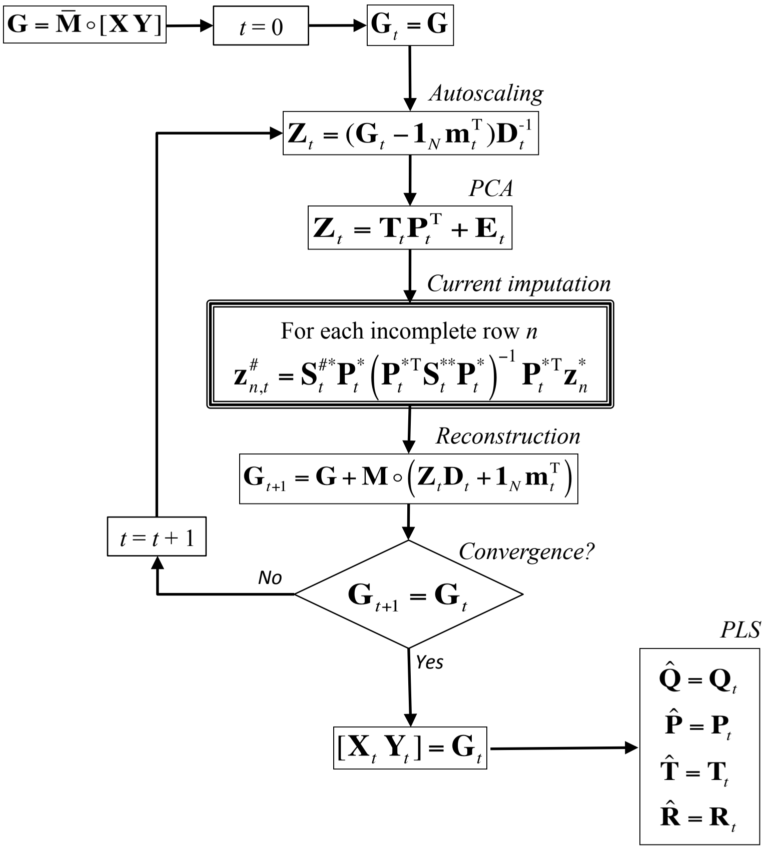

$\mathbf{G} = \bar{\mathbf{M}} \circ [\mathbf{X}\,\mathbf{Y}]$
$t = 0$
$\mathbf{G}_t = \mathbf{G}$
Autoscaling
$\mathbf{Z}_t = (\mathbf{G}_t - \mathbf{1}_N \mathbf{m}_t^{\mathrm{T}})\mathbf{D}_t^{-1}$
PCA
$\mathbf{Z}_t = \mathbf{T}_t\mathbf{P}_t^{\mathrm{T}} + \mathbf{E}_t$
Current imputation
For each incomplete row n
$\mathbf{z}^{\#}_{n,t} = \mathbf{S}_t^{\#*}\mathbf{P}_t^{*}\left(\mathbf{P}_t^{*\mathrm{T}}\mathbf{S}_t^{**}\mathbf{P}_t^{*}\right)^{-1}\mathbf{P}_t^{*\mathrm{T}}\mathbf{z}_n^{*}$
Reconstruction
$\mathbf{G}_{t+1} = \mathbf{G} + \mathbf{M}\circ\left(\mathbf{Z}_t\mathbf{D}_t + \mathbf{1}_N\mathbf{m}_t^{\mathrm{T}}\right)$
$t = t + 1$
Convergence?
No
$\mathbf{G}_{t+1} = \mathbf{G}_t$
Yes
$[\mathbf{X}_t\,\mathbf{Y}_t] = \mathbf{G}_t$
PLS
$\hat{\mathbf{Q}} = \mathbf{Q}_t$
$\hat{\mathbf{P}} = \mathbf{P}_t$
$\hat{\mathbf{T}} = \mathbf{T}_t$
$\hat{\mathbf{R}} = \mathbf{R}_t$

Figure 15.2: TSR-1 procedure for PLS-MB. $\mathbf{M}$ denotes here the MD indicator matrix of the augmented data $[\mathbf{X}\ \mathbf{Y}]$. $\bar{\mathbf{M}}$ is the complementary of the indicator matrix.

where $\mathbf{T} = \mathbf{XR}$, and:

$$\mathbf{T}^* = \mathbf{X}^*\mathbf{R}^* = \mathbf{X}^*\mathbf{W}^*(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1} \tag{15.3}$$

Matrices $\mathbf{P}$ and $\mathbf{W}$ are used to obtain $\mathbf{R}^*$ in order to improve prediction of the missing values using information from the complete PLS model, and to avoid problems of invertibility [70].

From Equations 15.2-15.3, the regression matrix $\mathbf{B}$ can be estimated as:

$$\hat{\mathbf{B}} = (\mathbf{T}^{*\mathrm{T}}\mathbf{T}^*)^{-1}\mathbf{T}^{*\mathrm{T}}\mathbf{T} = (\mathbf{R}^{*\mathrm{T}}\mathbf{X}^{\mathrm{T}*}\mathbf{X}^*\mathbf{R}^*)^{-1}\mathbf{R}^{*\mathrm{T}}\mathbf{X}^{*\mathrm{T}}\mathbf{T} \tag{15.4}$$

And since $\mathbf{X}^* = \mathbf{TP}^{*\mathrm{T}}$:

$$\hat{\mathbf{B}} = (\mathbf{R}^{*\mathrm{T}}\mathbf{X}^{\mathrm{T}*}\mathbf{X}^*\mathbf{R}^*)^{-1}\mathbf{R}^{*\mathrm{T}}\mathbf{P}^*\mathbf{T}^{\mathrm{T}}\mathbf{T} = (\mathbf{R}^{*\mathrm{T}}\mathbf{S}^{**}\mathbf{R}^*)^{-1}\mathbf{R}^{*\mathrm{T}}\mathbf{P}^*\mathbf{\Theta} \tag{15.5}$$

where $\mathbf{\Theta} = \frac{\mathbf{T}^{\mathrm{T}}\mathbf{T}}{N-1}$ is the covariance matrix of the scores. Finally, using the previous estimation, the scores of the PLS can be estimated in the last step of TSR for PLS-ME [70], that is, combining Equations 15.2-15.3:

$$\mathbf{T} = \mathbf{X}^*\mathbf{R}^*\mathbf{B} + \mathbf{E} \tag{15.6}$$

we get:

$$\hat{\boldsymbol{\tau}} = \hat{\mathbf{B}}^{\mathrm{T}}\mathbf{R}^{*\mathrm{T}}\mathbf{x}^* \tag{15.7}$$

being $\mathbf{x}^*$ the available part of the measurements of the new observation $\mathbf{x}$.

To adapt TSR from PLS-ME to PLS-MB, the same rationale presented in Chapter 10 is followed here. That is, the TSR version for ME is applied in each of the n rows with missing values of the data matrices at each step t of the iterative procedure, using the PLS model of the previous imputation step as the complete model.

Additionally, as a final step in TSR for PLS-MB, not only the PLS scores are needed, but the values for the MD imputation. These are obtained, from Equation 15.7, as:

$$\mathbf{x}_n^{\#} = \mathbf{P}^{\#}\hat{\boldsymbol{\tau}}_n = \mathbf{P}^{\#}\hat{\mathbf{B}}^{\mathrm{T}}\mathbf{R}^{*\mathrm{T}}\mathbf{x}_n^{*} = \mathbf{P}^{\#}\boldsymbol{\Theta}\mathbf{P}^{*\mathrm{T}}\mathbf{R}^{*}(\mathbf{R}^{*\mathrm{T}}\mathbf{S}^{**}\mathbf{R}^{*})^{-1}\mathbf{R}^{*\mathrm{T}}\mathbf{x}_n^{*} =$$
$$= \mathbf{P}^{\#}\frac{\mathbf{T}^{\mathrm{T}}\mathbf{T}}{N-1}\mathbf{P}^{*\mathrm{T}}\mathbf{R}^{*}(\mathbf{R}^{*\mathrm{T}}\mathbf{S}^{**}\mathbf{R}^{*})^{-1}\mathbf{R}^{*\mathrm{T}}\mathbf{x}_n^{*} =$$
$$= \frac{\mathbf{X}^{\#\mathrm{T}}\mathbf{X}^{*}}{N-1}\mathbf{R}^{*}(\mathbf{R}^{*\mathrm{T}}\mathbf{S}^{**}\mathbf{R}^{*})^{-1}\mathbf{R}^{*\mathrm{T}}\mathbf{x}_n^{*} = \mathbf{S}^{\#*}\mathbf{R}^{*}(\mathbf{R}^{*\mathrm{T}}\mathbf{S}^{**}\mathbf{R}^{*})^{-1}\mathbf{R}^{*\mathrm{T}}\mathbf{x}_n^{*} \quad (15.8)$$

It is worth noting that Equation 15.8 gives, in fact, a similar estimation for the missing measurements in **X** as presented in Chapter 10 for PCA-MB, that is, substituting $\mathbf{P}^*$ by $\mathbf{R}^*$ in Equation 10.6.

Finally, the estimation for the **Y** missing values is obtained as:

$$\mathbf{y}_n^{\#} = \mathbf{Q}^{\#}\hat{\boldsymbol{\tau}}_n = \mathbf{Q}^{\#}\boldsymbol{\Theta}\mathbf{P}^{*\mathrm{T}}\mathbf{R}^{*}(\mathbf{R}^{*\mathrm{T}}\mathbf{S}^{**}\mathbf{R}^{*})^{-1}\mathbf{R}^{*\mathrm{T}}\mathbf{x}_n^{*} \quad (15.9)$$

Unfortunately, Equation 15.9 cannot be expressed in a more simplified way, since the matrix establishing the relationship between **X** and **Y**, **R**, has the dimensions of the loading matrix in **X**, not in **Y**.

The previous methodology can be transferred to a similar diagram as presented in Figure 15.2. Thus, Figure 15.3 shows the adapted TSR version from PLS-ME [70] to PLS-MB, from now on denoted as TSR-2. This algorithm is indeed similar to TSR-1 (see Figure 15.2) with some differences: i) since data matrices are processed separatedly, each step is applied on both matrices, ii) MD indicator matrices are defined, each one associated to the data partition in one of the matrices, and iii) a PLS model is fitted on both autoscaled data matrices, instead of PCA. The iterative procedure stops again when the imputation values stabilize in both data matrices, so the PLS matrices are estimated using the last round of imputation.

15.3 Data sets

Four data sets are used in this chapter to compare the results of the new TSR algorithms against state-of-the-art approaches. The data sets have been selected exploring data matrices of different sizes and several latent structures.

The first case study is the Hald data set, widely used as an example for regression purposes [362, 363]. This data set has 13 observations of 4 ingredients of Portland cement and a single response variable equal to the number of calories of heat generated in the hardening process. One single LV is extracted, explaining 55% of **X** and 96% of **Y**.

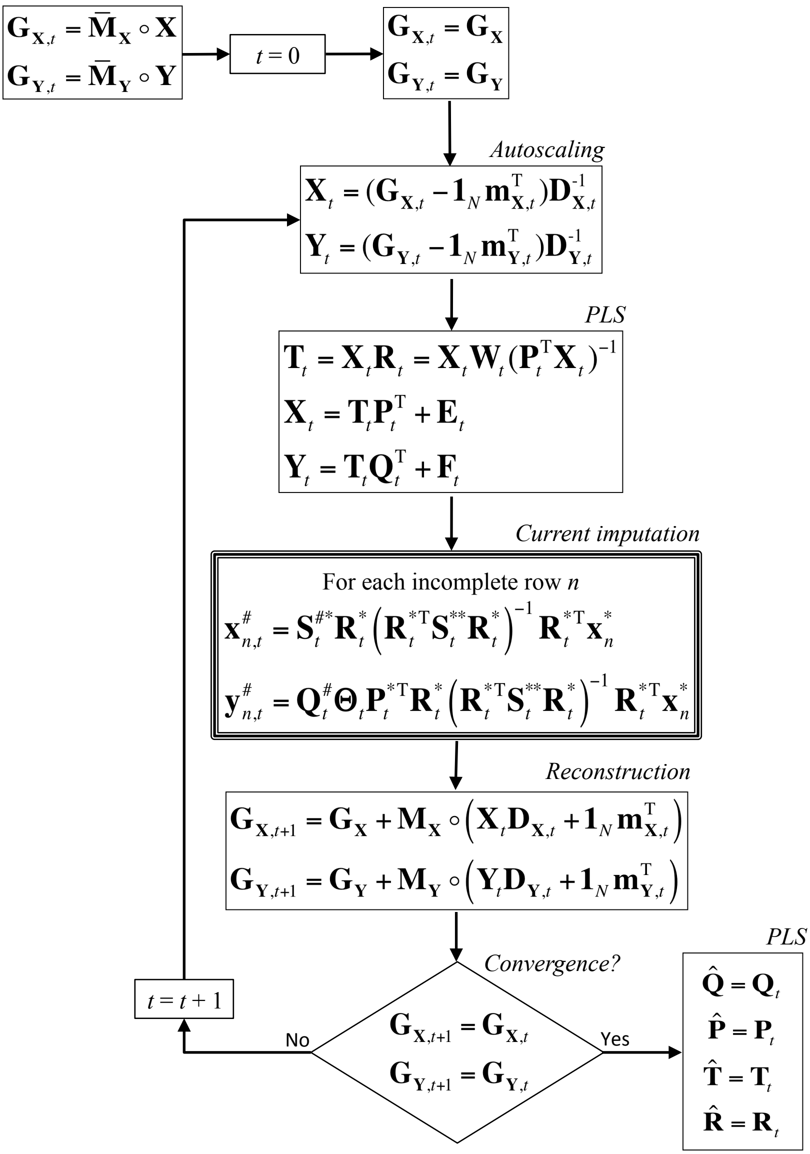

$\mathbf{G}_{\mathbf{X},t} = \bar{\mathbf{M}}_{\mathbf{X}} \circ \mathbf{X}$
$\mathbf{G}_{\mathbf{Y},t} = \bar{\mathbf{M}}_{\mathbf{Y}} \circ \mathbf{Y}$
$t = 0$
$\mathbf{G}_{\mathbf{X},t} = \mathbf{G}_{\mathbf{X}}$
$\mathbf{G}_{\mathbf{Y},t} = \mathbf{G}_{\mathbf{Y}}$
Autoscaling
$\mathbf{X}_t = (\mathbf{G}_{\mathbf{X},t} - \mathbf{1}_N \mathbf{m}_{\mathbf{X},t}^{\mathrm{T}})\mathbf{D}_{\mathbf{X},t}^{-1}$
$\mathbf{Y}_t = (\mathbf{G}_{\mathbf{Y},t} - \mathbf{1}_N \mathbf{m}_{\mathbf{Y},t}^{\mathrm{T}})\mathbf{D}_{\mathbf{Y},t}^{-1}$
PLS
$\mathbf{T}_t = \mathbf{X}_t\mathbf{R}_t = \mathbf{X}_t\mathbf{W}_t(\mathbf{P}_t^{\mathrm{T}}\mathbf{X}_t)^{-1}$
$\mathbf{X}_t = \mathbf{T}_t\mathbf{P}_t^{\mathrm{T}} + \mathbf{E}_t$
$\mathbf{Y}_t = \mathbf{T}_t\mathbf{Q}_t^{\mathrm{T}} + \mathbf{F}_t$
Current imputation
For each incomplete row n
$\mathbf{x}_{n,t}^{\#} = \mathbf{S}_t^{\#*}\mathbf{R}_t^{*}\left(\mathbf{R}_t^{*\mathrm{T}}\mathbf{S}_t^{**}\mathbf{R}_t^{*}\right)^{-1}\mathbf{R}_t^{*\mathrm{T}}\mathbf{x}_n^{*}$
$\mathbf{y}_{n,t}^{\#} = \mathbf{Q}_t^{\#}\mathbf{\Theta}_t\mathbf{P}_t^{*\mathrm{T}}\mathbf{R}_t^{*}\left(\mathbf{R}_t^{*\mathrm{T}}\mathbf{S}_t^{**}\mathbf{R}_t^{*}\right)^{-1}\mathbf{R}_t^{*\mathrm{T}}\mathbf{x}_n^{*}$
Reconstruction
$\mathbf{G}_{\mathbf{X},t+1} = \mathbf{G}_{\mathbf{X}} + \mathbf{M}_{\mathbf{X}} \circ \left(\mathbf{X}_t\mathbf{D}_{\mathbf{X},t} + \mathbf{1}_N\mathbf{m}_{\mathbf{X},t}^{\mathrm{T}}\right)$
$\mathbf{G}_{\mathbf{Y},t+1} = \mathbf{G}_{\mathbf{Y}} + \mathbf{M}_{\mathbf{Y}} \circ \left(\mathbf{Y}_t\mathbf{D}_{\mathbf{Y},t} + \mathbf{1}_N\mathbf{m}_{\mathbf{Y},t}^{\mathrm{T}}\right)$
Convergence?
$\mathbf{G}_{\mathbf{X},t+1} = \mathbf{G}_{\mathbf{X},t}$
$\mathbf{G}_{\mathbf{Y},t+1} = \mathbf{G}_{\mathbf{Y},t}$
No
$t = t + 1$
Yes
PLS
$\hat{\mathbf{Q}} = \mathbf{Q}_t$
$\hat{\mathbf{P}} = \mathbf{P}_t$
$\hat{\mathbf{T}} = \mathbf{T}_t$
$\hat{\mathbf{R}} = \mathbf{R}_t$

Figure 15.3: TSR-2 procedure for PLS-MB.

The second data set is taken from systems biology, and corresponds to 36 cultures from the flux data set used in chapters 5 and 13. The 44 fluxes measured in each experiment, excluding biomass, are considered as predictors, and the protein produced as the response. 3 LVs are selected in the PLS, explaining 76.5% of variance in **X** and 71.5% in **Y**.

The third data set comes from chemometrics, and has been used in Chapter 14. It corresponds to a set of measurements of pseudo-gasoline samples using an spectrometer capturing wavelengths from 800 nm to 1600 nm in 2 nm intervals. The first (master) spectrometer is used here. 6 LV are used in the PLS model, explaining 99.9% and 99.8% of variance in **X** and **Y**, respectively.

Finally, a simulated data set including 10 variables and 100 observations is simulated [303, 304] as in Chapter 13, using 4 PCs with eigenvalues equal to 3.5, 2.5, 2 and 1.5. The original data matrix is split afterwards: the first 6 variables are assigned to the **X** data set, the remaining 4 to **Y**. When fitting a PLS model, 3 LVs are chosen, explaining 87.9% of variance in **X** and 75.2% in **Y**.

15.4 Comparative study

In the next section, the performances of TSR-1, TSR-2, IA and NIPALS are compared. The previous data sets are used here as case studies. The strategy to generate the MD is the same as proposed in Chapters 10, 11 and 13: 6 incremental levels of MD are considered in each data set, ranging from 10% to 60%, and for each data set and percentage, 50 possible data sets are simulated, following the MCAR mechanism.

The principal performance criterion for each method is the MSPE (see Equation 10.10). Since the missing values are being imputed both in **X** and **Y**, MSPE-X and MSPE-Y denote the *MSPE* values in each data matrix, respectively. The second performance criterion is the cosine between the normalized weight vector of the first PLS, $\mathbf{r}_1$ obtained using the full data matrix and its corresponding from the imputed data set.

In order to assess whether the differences among methods, in terms of MSPE, are statistically significant, a mixed-effect ANOVA model is fitted per each case study. Now instead of a three-factor, as described in Chapter 10, a four-factor mixed-effect ANOVA model is applied: method (4 levels), **X**-MD percentage (6 levels), **Y**-MD percentage (6 levels), and their interactions are fixed-effect factors, and the data set, nested to the combination of **X**-MD and **Y**-MD percentages, is a random-effect factor. Also, a logarithmic transformation is used for MSPE-X and MSPE-Y. This transformation also expands the differences for low percentages of MD, easing the visualization of the plots. In case any effect or interaction is statistically significant (p-value<0.05), the 95% LSD intervals are computed to establish differences among methods.

15.5 Results

15.5.1 Hald data

As expected, the more missing values are considered in both **X** and **Y** the more difficult is for all methods to reconstruct accurately the MD. This can be seen in the first and second column of plots in Figure 15.4, corresponding to MSPE-X and MSPE-Y values. Each plot in these two columns show the evolution of the MSPEs when increasing the **X**-MD percentage for a fixed **Y**-MD percentage. In the third column of plots, representing the cosines of the normalized weigths of the first LV, this effect can also be appreciated in the degradation of the cosine values.

NIPALS has problems in imputing MD in this first data set from 40% of **X**-MD onwards, and when converges, it has, in general, a statistically worse peformance than the other methods in imputing MD in **X** (see first column of plots in Figure 15.4). Regarding the MSPE-Y, its performance is clearly the worst (see second column of plots in Figure 15.4).

The performance of TSR-2 and IA is similar in MSPE-X, having TSR-2 a better performance for some percentages of MD (see first column of plots in Figure 15.4). Regarding MD in **Y**, IA attains a statistically better results for low **X**-MD (10-30%), otherwise their results are similar (see second column of plots in Figure 15.4).

The performance of TSR-1 is, in general, statistically superior to all the other methods up to 40% of **Y**-MD, and there tend to be no statistically significant differences for some **X**-MD percentages among the other methods (except NIPALS) for 50-60% of **Y**-MD.

15.5.2 *P. pastoris data*

NIPALS has also problems in the *P. pastoris* data set (see Figure 15.5). Even having results on MSPE-Y statistically as good as TSR-1 for low percentages of **X**-MD and **Y**-MD, and statistically better than IA and TSR-2 (see Figure 15.5e, h and k), it fails to converge when more than 40% of missing data is considered in **Y**.

TSR-1 obtains here the best performance both in MSPE-X and MSPE-Y, with very few exceptions, in which its results are statistically equal to other approaches

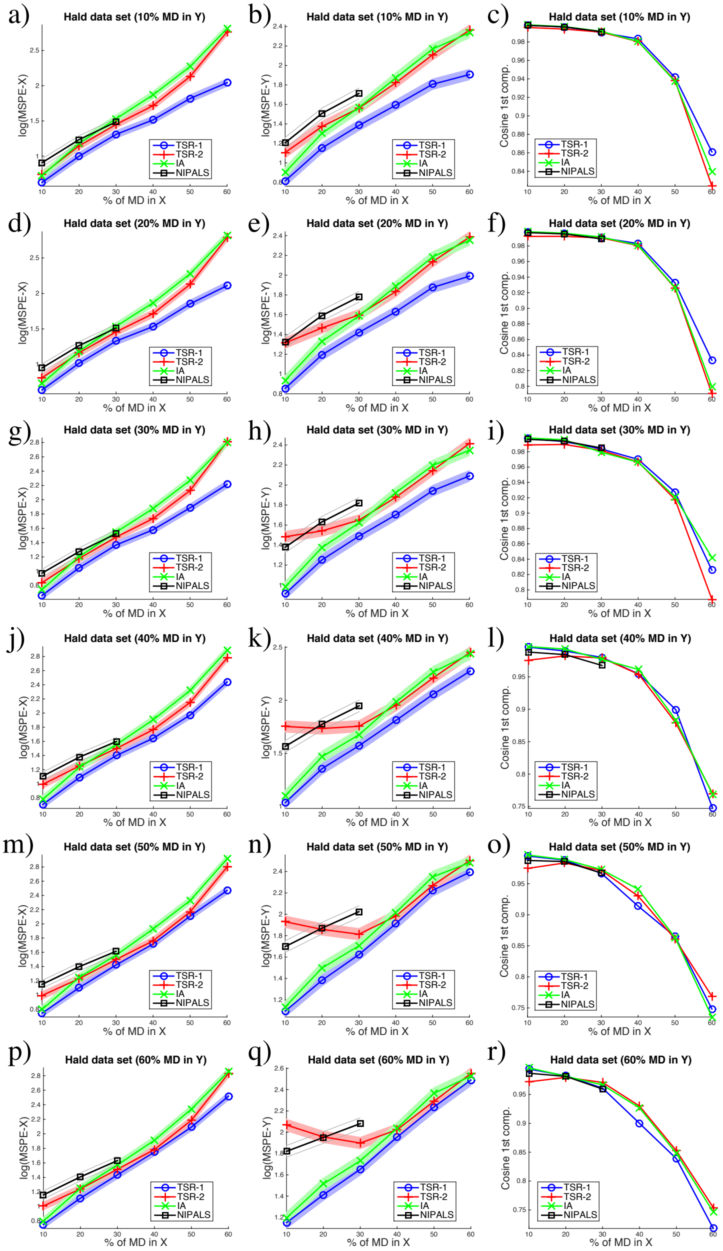


Figure 15.4: Hald data set results. The first (second) column of plots show the MSPE-X (MSPE-Y) results and the last column shows the cosines of the normalized weights of the first LV. The x-axis of each plot denotes the **X**-MD percentage. The differences regarding **Y**-MD percentages can be seen comparing rows of plots. The shaded bands represent the LSD 95% confidence intervals for the MSPE results of each method.

(see first and second column of plots in Figure 15.5). Mainly, the second-best method in this data set is TSR-2, followed by IA.

The MSPE-Y for high percentages of **Y**-MD show an oscillatory performance of all methods, e.g. Figure 15.5k, n and q. This effect is probably due to the fact that the PLS does not explain approximately 29% of the variability in **Y**, and this lack of explained variance is causing artifacts depending on the combination of percentages of MD in **X** and **Y** considered for the imputation.

15.5.3 NIR data

The performance of NIPALS in this third case study is even poorer than in previous examples. Here, it is only able to impute up to 40% of **X** and 20% of **Y**-MD. And, when available, its results are statistically worse than the other iterative approaches.

Regarding MSPE-X, TSR-1 and TSR-2 have a similar performance, being both statistically superior to IA for 40%-60% of **X**-MD percentages (see first column of plots in 15.6). However, TSR-1 is indisputably the best method when checking the MSPE-Y results, followed by TSR-2, which gets statistically better or equal results than IA (second column of plots in Figure 15.6).

The performance in MSPE-Y of IA for high **Y**-MD percentages (see Figures 15.6k, n and q) improves when changing from 10 to 30% of MD in **X**. This is probably due to IA is being affected by overfitting in the imputation, since the percentage of variance explained of both **X** and **Y** in the PLS model is very high (see Section 15.3) when using 6 LV in the model. TSR-1 and TSR-2 seem to be not influenced by this problem.

The difference in the performances of TSR-based algorithms and IA can also be appreciated in the third column of plots in Figure 15.6, where, even getting very high cosines, the values of IA appear below TSRs' when the percentages of MD in **X** and **Y** increase.

15.5.4 Simulated data

In the last case study analysed here, NIPALS is unable to analyse any combination of **X** and **Y**-MD percentages, even including only 10%-**X** and 10%-**Y** MD. TSR-1 shows again a clear statistically better performance in both MSPE-X and MSPE-Y for all MD percentages than its competitors, with few exceptions where its results are as accurate as TSR-2's. Between TSR-2 and IA there are again some cases in which they get statistically equal results, but in general the performance of TSR-2 outperforms IA. These significant differences match the results obtained in the

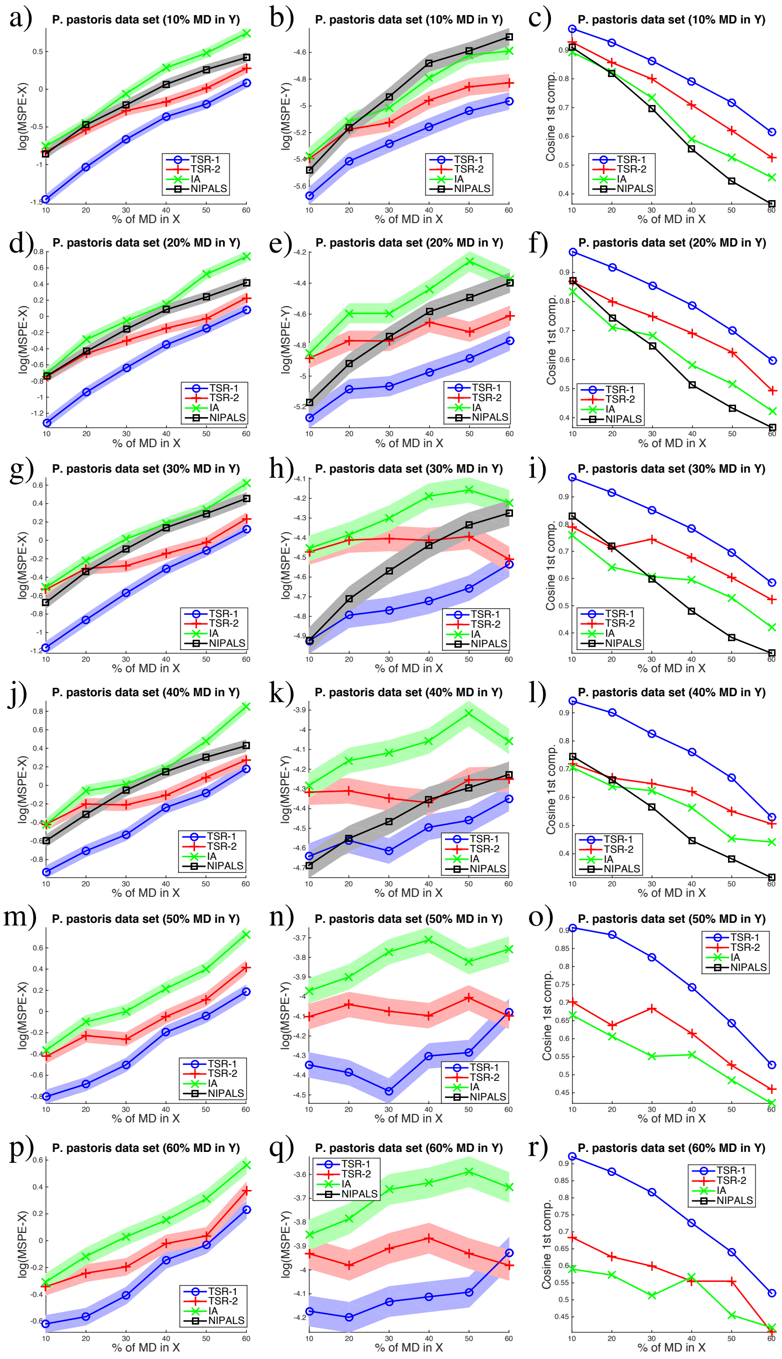


Figure 15.5: ***P. pastoris*** **data set results.** More details in caption of Figure 15.4.

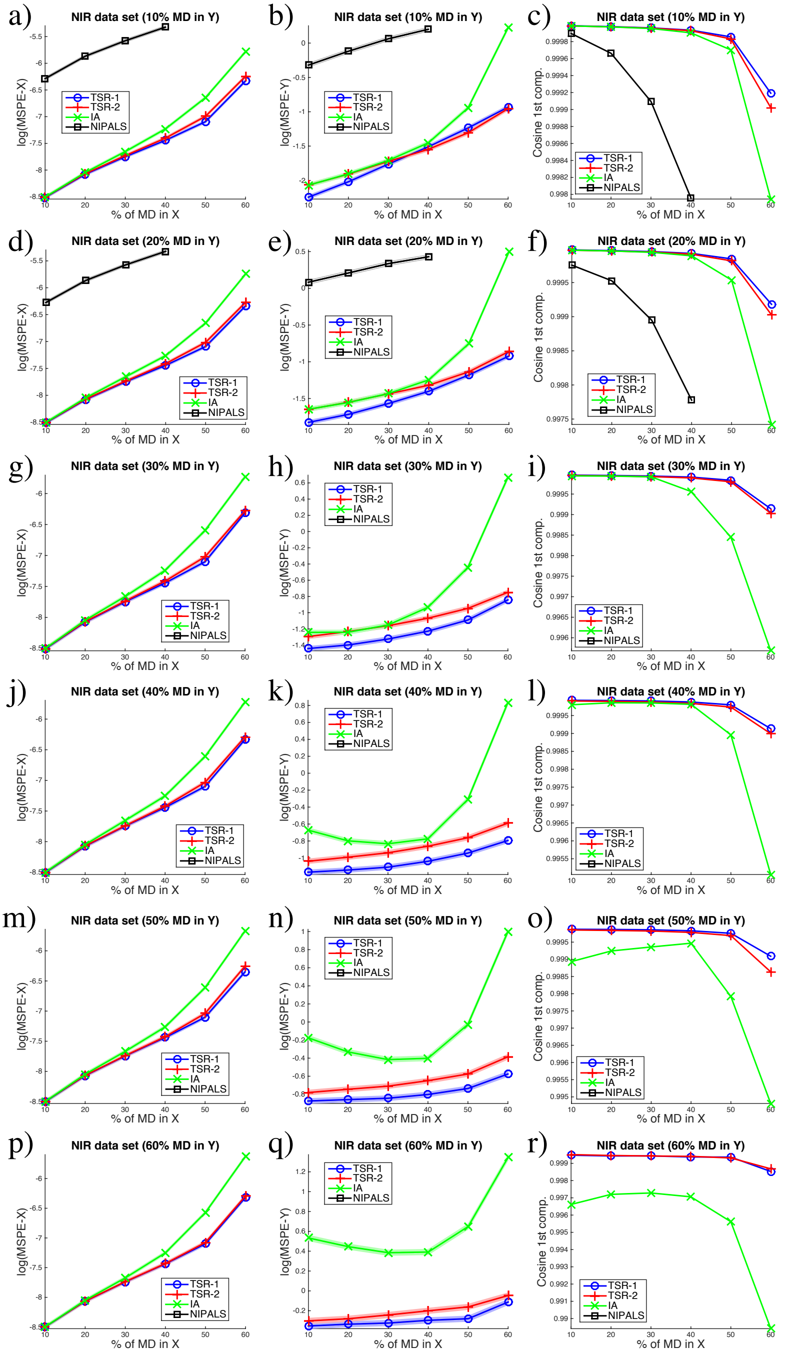


Figure 15.6: NIR data set results. More details in caption of Figure 15.4.

third column of plots, corresponding to the cosines of the weight vector of the first LV.

In this dataset, IA shows an erratic performance, especially in MSPE-Y (see Figures 15.7n and q). This happened also in the *P. pastoris* case study, and reinforces the hypothesis that it is due to the lack of variance explained in **Y**, in this case similar to the aforementioned example (25%). However, TSR-1 seems to be not affected by this problem in any case study.

15.6 Discussion and conclusion

Two TSR algorithms have been proposed in this chapter: TSR-1 consists of an adaptation of the TSR algorithm from PCA-MB to PLS-MB, and TSR-2 is an adaptation of TSR from PLS-ME to PLS-MB. The case studies analysed here show that TSR-1 is an excellent approach regardless the latent structure of the data. Its performance is, in general, statistically superior to TSR-2, with few exceptions for some combinations of MD percentages in **X** and **Y**.

The TSR approaches proposed have been compared to other state-of-the-art methods: IA and NIPALS. IA shows generally a statistically worse performance than the TSR-based approaches, being its results in few cases closer to TSR-2's. NIPALS, a method implemented in many commercial statistical packages (such as ProSensus MultiVariate, The Unscrambler, SIMCA-P and PLS Toolbox), is clearly the statistically worst method compared here, since for most MD combinations is not able to converge and when it converges, its results are significantly worse than IA and TSR-based methods.

TSR-1 performed extraordinarily well for PLS-MB with MD. As commented in the Introduction, the ability of TSR to reconstruct the covariance matrix of incomplete data sets, which ultimately determines the relationships among variables in most multivariate models, makes the final PLS fitted on imputed data resemble more the actual model than specific methodologies developed for PLS-MB with MD. This way, if practitioners find MD when fitting other covariance matrix-dependent methodologies, such as principal component regression or multiple regression models, they can use directly TSR-1 to impute the MD and then use the complete matrices for obtaining the desired model.

On the other hand, TSR-1 uses the number of components specified for the PLS model at hand to build the PCA-based model for the MD imputation. This may generate a problem if the covariance structure of the augmented data matrix [**X Y**] is strongly different to the latent structure of a PLS model between **X** and **Y**, thus provoking over or underfitting. However, one way to overcome this hypothetic situation consists of using an algorithm to select the appropriate number of PCs using the augmented matrix. In Chapter 12, the *ckf* algorithm [330] was used to

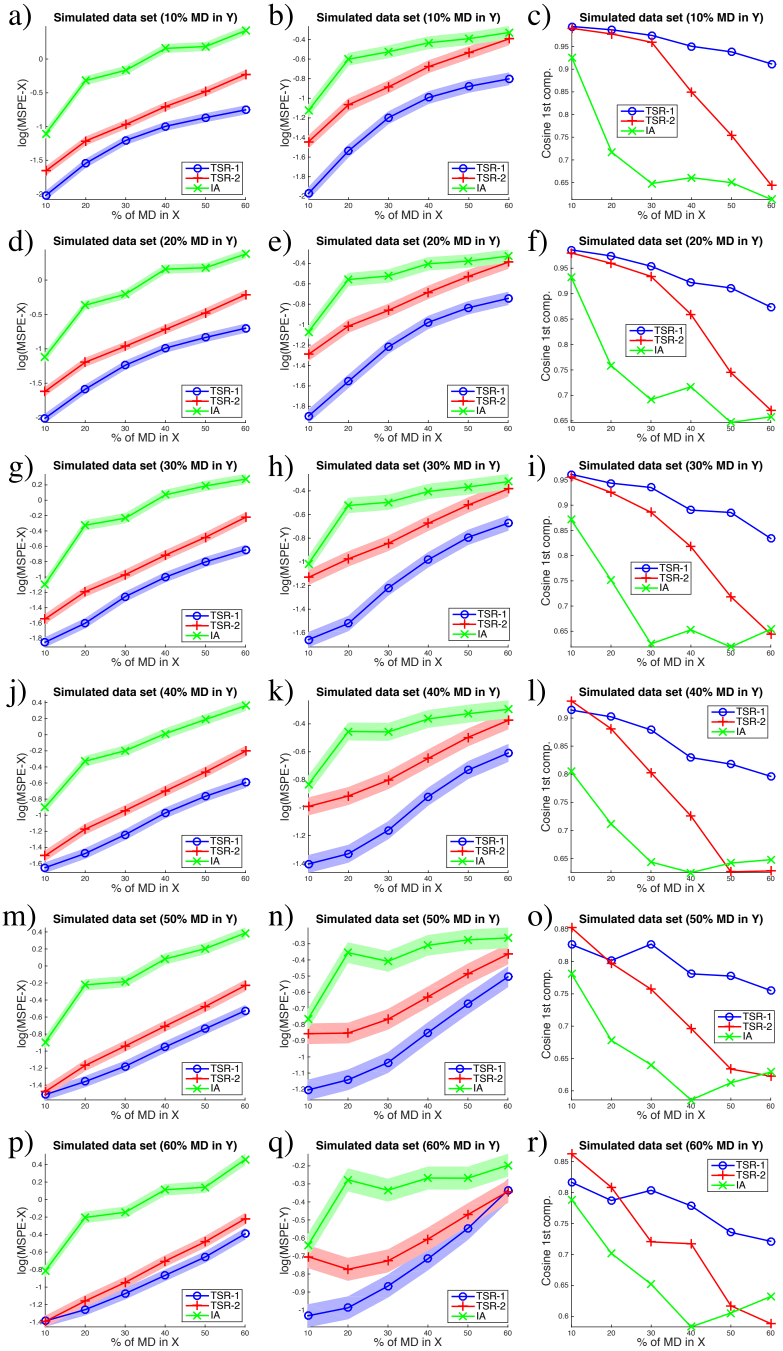


Figure 15.7: Simulated data set results. More details in caption of Figure 15.4.

decide the number of components in the MDI toolbox for PCA-MB. This procedure could solve the aforementioned problem.

Both TSR algorithms proposed here are freely available at `http://mseg.webs.upv.es`, under a GNU license.

Part IV

Epilogue

Chapter 16

Conclusions

In this thesis several procedures commonly applied within the chemometrics community were used to model and solve problems in systems biology. Furthermore, new methodologies were developed to cope with specific problems within the different omic areas. After an initial introduction of both the existing chemometric techniques and the basics of omic sciences and systems biology, the different contributions were presented within two parts. In Part II: Modelling biological organisms, metabolic, fluxomic, proteomic and phenotypic data were analysed using exploratory and predictive models. In Part III: Missing data, the problem of missing data and outliers in systems biology and bioprocesses was studied, solving problems as model building with missing data, data cleaning, and calibration transfer. In this last part, the conclusions, relevance and future lines of the thesis are outlined.

16.1 Meeting the objectives

In this section, the main conclusions in this document are summarized, in order to demonstrate that the objectives have been met.

Objective 1: Build models integrating information from different biological levels

Grey models revealed as a powerful approaches to combine first principle knowledge of an organism, i.e. the metabolic model and stoichiometry, and a set of experimental results. This way, when steady state flux measurements were coherent with the theoretical model, a MC sampling could be performed on the convex cone of feasible flux solutions to capture the variability associated to the enviornmental conditions. To understand the flux distributions two methodologies were tested in this thesis: an enhanced PCA using MEDA, and MCR. The output of the first approach was a set of orthogonal pathways or modules in the network relating enviornmental conditions, such as substrate consumption, with end-products of interest, such as biomass and protein production. However, no biological information could be included in the results, since it is a hard model. In comparison, MCR allowed to include constraints in the algorithm, driving the pathways to a biologically more meanginful solution. This ability, jointly with the non-orthogonality of pathways, permitted to describe all experimental conditions tested in Chapter 5, including a scenario not described using PCA+MEDA. Both approaches permit to understand what was happening from the substrates to the end-products, however, MCR offered the possibility to model the flux data including constraints capturing the available information from the experiment.

Chapters 6 and 7 presented a novel framework to model steady state flux data, in order to include topological information of the metabolic network in the multivariate models evaluated in the previous paragraph. This way, PEMA built a PCA-like model using the EMs of the network as the candidates for the components. Using this approach, the principal pathways describing the flux data were directly a combination of fluxes flowing in the metabolic network in thermodynamically feasible way from substrates to end-products. The simulated study presented in Chapter 6 confirmed that the algorithm identified most of the EMs used to simulate the data, up to the point where a particular flux distribution could be represented by different combinations of EMs. The results in the real case studies show the EMs identified in *P. pastoris* and *E. coli* and how the PEMA model could be exploited using a set of visual plots, all of them included in a freely available in MATLAB code within the so-called PEMA toolbox. This EM-based model could be extended to non-steady state data, as presented in Chapter 7: dynEMA was proposed as a direct extention of PEMA, i.e., to fit an exploratory model, and dynEMR-DA was proposed to discrimante between experimental conditions. A simulated and a real case study proved the ability of this modelling to find which dynEMs were the most variable pathways when changing the glucose amount present as a substrate or when switching from aerobic to anaerobic conditions. The results between simulated and actual data were coherent, strenghthening the validity of the modelling and justifying the need of model non-steady state data using dynamic models rather than applyting a PEMA-based model directly.

From this thesis, it is concluded that when dealing with steady state flux data sets, PEMA should be applied in order to extract the relevant actual metabolic pathways active in a metabolic network. When the network size increases, and it is desired to combine this pathways creating pseudo-pathways (not necessarily from inputs to outputs) or functional modules, MCR should be applied, being also able of including more biological constraints in the multivariate model. When dealing with non-steady state flux data, dynamic methods have to be used, as dynEMA or dynEMR-DA, since the coefficients multiplying the pathways can change strongly depending on the phase of the experiment.

To complete this first objective, three additional biological layers of organisms were investigated: the genome, the proteome and the phenotype. After a bibliography review on PPIs detected in potyviruses, a PPIN was built as a meta-analysis of these references. This network was the starting point to relating the effect that mutations in the RNA of the viruses provoke in the PPIN and how this effect was transferred to the physical state of the organism. To model this different sources of information, a data fusion was applied, specifically using PLS. This methodology allowed to identify functional modules in the PPIN, activated by particular mutations and contributing positively or negatively to the performance of the organism. This way, the mutations-protein-fitness and the mutations-fitness effects were modelled, giving clues to researchers about what regions and mutations are the most influencing ones on potyviruses.

Objective 2: Develop missing data methods and outlier detection and correction procedures in systems biology and bioprocesses

The first contribution devoted to accomplish objective 2 was the adaptation of the regression-based framework methods from PCA-ME with MD to the MB context. This way TSR, PMP and KDR methods are now able to impute missing values in multivariate data sets. These methods were compared to other state-of-the-art methods (NIPALS, IA, NLP and DA) in several data sets. When comparing the performances, TSR attained excellent results when imputing MD with different data structures and LVs, even with big data sets. Most of these methods shown problems when dealing with datasets with complex latent structures.

The regresion-based methods, presented in Chapter 10 were afterwards adapted to impute within the MLPCA algorithm (Chapter 13), in order to check whether a ML-based imputation using the regression-based methods improve the accuracy of the reconstruction of missing values. This adaptation arose from a similarity between the imputation step of MLPCA and PMP, therefore it made sense to test the other regression-based methods in this step. The conclusions were: i) even though PMP and MLPCA have similarities, the imputation is completely different, ii) MLPCA using TSR as imputation step may outperform the original MLPCA, and iii) the original TSR outperform any other ML-based algorithm. All methods in these two chapter were tested using datasets coming from different areas within chemometrics and systems biology, such as NIR and FTIR measurements and flux data sets.

After evaluating the good performance of TSR it was decided to make the algorithm ready-to-use for the scientific community. This was done through two contributions. In the first one, the algorithm was implemented within a module of data cleaning for network inference purposes in systems biology. This way, the data cleaning module have two steps: i) an initial MD imputation, as performed in Chapter 10, and afterwards an outlier detection and correction. The final step of the methodology consisted of inferring the biological network, based on the cleaned data, using MIDER method. When compared to other imputation methods commonly used within omic sciences (such as the k-nearest neighbor algorithm), TSR showed again a statistically superior performance, and the outlier detection scheme, based on contribution plots to the squared prediction error (SPE) of a PCA, permit to correct errors in datasets prior to inferring the desired network. At the same time, this chapter studied the effect that the MD imputation has when further analysis are applied on the imputed data sets, something that was not evaluated in Chapter 10.

The second contribution presented a MATLAB GUI for MD imputation, the MDI toolbox, able to guide the practitioner from the missing pattern visualisation in the data set to the imputation of missing values and the visual exploitation of the resulting PCA model. Several external validators outlined the benefits of having this freely distributed GUI able for the scientific community.

The main conclusions of this thesis about MD imputation methods is that TSR is a great methodology to solve MD problems regardless the data structure and the aim of the imputation, i.e. fitting PCA models or use the imputed values as an input for further analysis. When exploring other state-of-the-art proposals, including ML-based algorithms and methods implemented in commercial software packages, it is concluded that TSR is an outstanding approach.

Finally, two versions of TSR were proposed for PLS-MB: one based on an adaptation from the PCA-MB to PLS-MB (TSR-1), and another one from PLS-ME to PLS-MB (TSR-2). The first version, using first a PCA model to impute the data in the augmented matrix, including both the predictors and the responses, yielded better results than the second version, imputing using the PLS algorithm. When both methods were compared to other state-of-the-art methods, such as IA and NIPALS, they obtained better results in general.

Objective 3: Address near infrared (NIR) and image analysis problems in bioprocesses.

Two projects were launched to fulfill the third objective, involving two entities not included in the MultiScaleS/SynBioFactory projects, as commented in Chapter 1. The first project consisted in addressing the so-called calibration transfer problem between NIR instruments. In Chapter 14 three new methods were presented for performing calibration transfer: TSR and two JYPLS-based methods. Applying TSR and JYPLS with model inversion, the unmeasured spectra collected in the slave instrument were imputed using the relationships between samples measured in both instruments and the available measurements of the master instrument. JYPLS without inversion used the original algorithm to transfer a model developed in the master instrument to the slave one. A comparative study was performed using NIR samples from chemometrics and systems biology. The conclusion of the study was that TSR and JYPLS with inversion offer better results than PDS, the reference method, when enough samples were included in the model, that is, when results as good as a full recalibration are desired. Also, the performance of these methods was not affected when the resolution of the slave instrument was reduced, something very common in (bio)industries, when a high-resolution spectrometer is used for specific measurements and a low-resolution one is used on-line.

The second project aimed at detecting rottenness in oranges using image analysis. Specifically, the idea was to study the effect of the virus *P. digitatum* on oranges some days after the harvest, trying to catch this differences in the inner part of the fruit using hyperspectral images. After feature extraction of the three-dimensional datacubes, *N*way PLS discriminant analysis (NPLS-DA) was applied to discriminate between infected and sound oranges. Discriminant models are prone to overfitting, so a double cross validation procedure was applied in this study, in order to avoid spurious results. The interest in this project was not only to be able to discriminate but also to find what wavelengths were the most discriminant ones, since CCD cameras can include filters reproducing these wavelengths. Permutation testing on VIP revealed as a powerful tool to select this discriminant wavelengths from the complete set, loosing only around 2% of correct classification.

16.2 Relevance

The relevance of the present PhD Thesis is highlighted in the following points:

- This thesis has been developed within the framework of two research projects from the Spanish Ministry of Economy, coordinated among different Spanish sites. Two projects have been completed with CSIC and IVIA. Two international research stays have been carried out (Lisbon and Amsterdam. And two companies, Biopolis S.L. and Shell Global Solutions B.V. have been interested in the results provided here. Therefore, the methodologies developed and applied here have been disseminated across many research groups to analyse data coming from projects and companies in different countries.

- The exploratory methods applied here for metabolic flux understanding (PCA, MCR, PEMA and dynEMR-DA) can be used to indentify desirable metabolic states in organisms. This knowledge can be used in bioindustries to drive biofermentations to the desired state, as for example, the production of a protein of interest.

- Two free (open use) toolboxes were presented in this thesis: PEMA and MDI, which can be used by the scientific and the industrial community to identify active pathways in fluxomics and to impute missing values in the most proper way, respectively.

- The expected output of the wavelength selection using NPLS-DA is the development of an automatic procedure to discriminate between sound and infected oranges in fruit warehouses, which will potentially save money due to i) the relatively cheap filters that can be incorporated to CCD cameras to mimic the results of expensive hyperspectral cameras, and ii) the reduction of the losses for fruit contamination after their storage.

- The knowledge obtained about *Potyvirus* will help the scientific community in i) the creation of more resistant strains of the virus, for research purposes, and ii) understanding the way in which they have to attack the virus.

- With the methods proposed in this document, the transfer of calibration models between near infrared spectrometers will be performed both in research and bio-based industries in a more proper way, avoiding time-consuming recalibrations and fostering advances in other research areas.

- Other research areas, such as social sciences, will benefit from the advances in this thesis, due to MD and outlier problems are an intrinsic issues to data-driven network inference not only in systems biology and chemometrics.

16.3 Future lines

This PhD dissertation opens some future lines:

- Test the results obtained with *P. pastoris*, *E. coli* and *S. cerevisiae* using data from fermentations online. Also, new multivariate models, taking into account the properties of raw materials, could be applied in this data, to tune the bioproduction systems, depending on the suppliers.

- Investigate whether PEMA models can be used for intracellular fluxes prediction. The set of active EMs can be obtained applying PEMA on a subset of the fluxes, say the extracelullar ones. In this case, however, the redundancy in the EMs is even higher, since they represent only a subset of the network. But if only extracellular measurements are available, it would be very useful to fit a PEMA model, obtain the active EMs, and then, based on the coefficients multiplying the EM to fit the extracellular fluxes, infer the intracellular ones.

- Develop a new version of the PEMA toolbox including i) the dynamic modelling of non-steady state flux data (dynEM models) and ii) a GUI, as presented in Chapter 12 with MDI toolbox.

- Test the data fusion approach with a larger dataset containing more mutants, and extend the analysis to larger PPINs.

- Design and test the CCD camera able to discriminate between infected and sound oranges based on the wavelengths selected in Chapter 9, and solve the problems of its online implementation.

- Test the NPLS-DA models for hyperspectral imaging in other fruits of the valencian region, as lemons and khakis.

- Combine the knowledge acquired about fermentations, near infrared spectroscopy and image analysis, in order to monitor big fermentations using these different information sources.

- Extend TSR from PLS model building with missing values to PLS model exploitation, as performed in Chapter 10 with PCA. This way, missing values in the quality variables could be imputed within industrial (bio)processes.

- Build a new version of the MDI toolbox addressing not only PCA-MB with missing values, but incorporating i) PLS-MB and ii) PCA/PLS-ME with missing data.

- Incorporate the outliers detection and possible correction within the TSR algorithm. In Chapter 11, this was done at two different steps: first imputation, then outlier analysis. It could be interesting to integrate the outlier detection at each iteration of the TSR algorithm, since outliers corresponding to dirty data may influence the solution when used as true values.

- Create a toolbox in MATLAB for calibration transfer between near infrared spectrometers.

- Investigate whether the calibration transfer procedures proposed in this thesis, based on TSR and JYPLS with inverse, can be applied in other spectrometers, such as mid-infrared or NMR. As well, it would be interesting to check whether the imputation of measurements in a device can be performed using measurements from a different one, once proven that the relationships between both measurements are strong enough.

- Compare the results of TSR-1 and TSR-2 algorithm for PLS-MB when the covariance structure of the augmented matrix is strongly different than the latent structure of a PLS between predictors and responses. Also, investigate how to chose the appropriate number of components in these cases, e.g. using the ckf algorithm. As well, a procedure to select the appropriate number of components in TSR-2 must be investigated.

- Finally, as commented for PCA-MB purposes, it would be nice to address not only MD within TSR algorithms for PLS-MB but also outlier detection and correction.

Bibliography

1. González-Martínez, J. M. *et al.* Metabolic flux understanding of Pichia pastoris grown on heterogenous culture media. *Chemometrics and Intelligent Laboratory Systems* **134,** 89–99 (2014).

2. Bosque, G., Folch-Fortuny, A., Picó, J., Ferrer, A. & Elena, S. F. Topology analysis and visualization of Potyvirus protein-protein interaction network. *BMC Systems Biology* **8,** 129 (2014).

3. Folch-Fortuny, A. *et al.* MCR-ALS on metabolic networks: Obtaining more meaningful pathways. *Chemometrics and Intelligent Laboratory Systems* **142,** 293–303 (2015).

4. Folch-Fortuny, A., Arteaga, F. & Ferrer, A. PCA model building with missing data: New proposals and a comparative study. *Chemometrics and Intelligent Laboratory Systems* **146,** 77–88 (2015).

5. Folch-Fortuny, A., Villaverde, A. F., Ferrer, A. & Banga, J. R. Enabling network inference methods to handle missing data and outliers. *BMC Bioinformatics* **16,** 283 (2015).

6. Folch-Fortuny, A., Bosque, G., Picó, J., Ferrer, A. & Elena, S. F. Fusion of genomic, proteomic and phenotypic data: the case of potyviruses. *Molecular BioSystems* **12,** 253–261 (2016).

7. Folch-Fortuny, A., Marques, R., Isidro, I. A., Oliveira, R. & Ferrer, A. Principal elementary mode analysis (PEMA). *Molecular BioSystems* **12,** 737–746 (2016).

8. Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Missing Data Imputation Toolbox for MATLAB. *Chemometrics and Intelligent Laboratory Systems* **154,** 93–100 (2016).

9. Folch-Fortuny, A., Arteaga, F. & Ferrer, A. Assessment of maximum likelihood PCA missing data imputation. *Journal of Chemometrics* **30,** 386–393 (2016).

10. Folch-Fortuny, A., Prats-Montalbán, J., Cubero, S., Blasco, J. & Ferrer, A. VIS/NIR hyperspectral imaging and N-way PLS-DA models for detection of decay lesions in citrus fruits. *Chemometrics and Intelligent Laboratory Systems* **156,** 241–248 (2016).

11. Folch-Fortuny, A., Vitale, R., de Noord, O. & Ferrer, A. Calibration transfer between NIR spectrometers: new proposals and a comparative study. *Journal of Chemometrics,* accepted.

12. Folch-Fortuny, A., Arteaga, F. & Ferrer, A. PLS model building with missing data: New algorithms and a comparative study. *Journal of Chemometrics,* submitted.

13. Folch-Fortuny, A. *et al.* Dynamic elementary mode analysis of non-steady state flux data. In preparation.

14. MacGregor, J. Using On-Line Process Data to Improve Quality. Is there a Role for Statisticians?. Are They Up for the Challenge? *ASQC Statistics Division Newsletter* **16,** 6–13 (1996).

15. Ferrer, A. *Control Estadístico MegaVariante para los Procesos del Siglo XXI* in *27 Congreso Nacional de Estadística e Investigación Operativa* (Lleida, 2003), 24–38.

16. Brown, S. D, Sarabia, L. A & Trygg, J. *Comprehensive chemometrics chemical and biochemical data analysis* (Elsevier, Amsterdam, 2009).

17. Jackson, J. E. *A User's Guide to Principal Components* (Wiley Series in Probability and Statistics, 2004).

18. Camacho, J., Picó, J. & Ferrer, A. Data understanding with PCA: Structural and Variance Information plots. *Chemometrics and Intelligent Laboratory Systems* **100,** 48–56 (2010).

19. Camacho, J. & Ferrer, A. Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: theoretical aspects. *Journal of Chemometrics* **26,** 361–373 (2012).

20. Kourti, T. & MacGregor, J. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems* **28,** 3–21 (1995).

21. Ferrer, A. Multivariate Statistical Process Control Based on Principal Component Analysis (MSPC-PCA): Some Reflections and a Case Study in an Autobody Assembly Process. *Quality Engineering* **19,** 311–325 (2007).

22. Jackson, J. & Mudholkar, G. S. Control procedures for residuals associated with Principal Component Analysis. *Technometrics* **21,** 341–349 (1979).

23. Mahalanobis, P. C. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India* **2,** 49–55 (1936).

24. Arteaga, F. *Control Estadístico Multivariante de Procesos con datos faltantes mediante Análisis de Componentes Principales* PhD thesis (Universidad Politécnica de Valencia, 2003).

25. Tracy, N. D., Young, J. C. & Mason, R. L. Multivariate Control Charts for Individual Observations. *Journal of Quality Technology* **24,** 88–95 (1992).

26. Camacho, J. Missing-data theory in the context of exploratory data analysis. *Chemometrics and Intelligent Laboratory Systems* **103,** 8–18 (2010).

27. Arteaga, F. & Ferrer, A. Dealing with missing data in MSPC: Several methods, different interpretations, some examples. *Journal of Chemometrics* **16,** 408–418 (2002).

28. Wold, S. Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models. *Technometrics* **20,** 397–405 (1978).

29. Wentzell, P. D., Andrews, D. T., Hamilton, D. C., Faber, K. & Kowalski, B. R. Maximum likelihood principal component analysis. *Journal of Chemometrics* **11,** 339–366 (1997).

30. Nelson, P. R. *The treatment of missing measurements in PCA and PLS models* PhD thesis (MacMaster University, Hamilton, Ontario, Canada, 2002).

31. Tauler, R. Multivariate curve resolution applied to second order data. *Chemometrics and Intelligent Laboratory Systems* **30,** 133–146 (1995).

32. Tauler, R., Smilde, A. & Kowalski, B. Selectivity, local rank, three-way data analysis and ambiguity in multivariate curve resolution. *Journal of Chemometrics* **9,** 31–58 (1995).

33. Jaumot, J., Tauler, R. & Gargallo, R. Exploratory data analysis of DNA microarrays by multivariate curve resolution. *Analytical Biochemistry* **358,** 76–89 (2006).

34. De Juan, A., Rutan, S. C. & Tauler, R. *Two-Way Data Analysis: Multivariate Curve Resolution – Iterative Resolution Methods* in *Comprehensive Chemometrics* **3** (Elsevier, Oxford, 2009), 325–344.

35. Jaumot, J., Gargallo, R., De Juan, A. & Tauler, R. A graphical user-friendly interface for MCR-ALS: A new tool for multivariate curve resolution in MATLAB. *Chemometrics and Intelligent Laboratory Systems* **76,** 101–110 (2005).

36. Jaumot, J., de Juan, A. & Tauler, R. MCR-ALS GUI 2.0: New features and applications. *Chemometrics and Intelligent Laboratory Systems* **140,** 1–12 (2015).

37. Windig, W. & Guilment, J. Interactive self-modeling mixture analysis. *Analytical Chemistry* **63,** 1425–1432 (1991).

38. Grande, B.-V. & Manne, R. Use of convexity for finding pure variables in two-way data from mixtures. *Chemometrics and Intelligent Laboratory Systems* **50,** 19–33 (2000).

39. Windig, W., Gallagher, N. B., Shaver, J. M. & Wise, B. M. A new approach for interactive self-modeling mixture analysis. *Chemometrics and Intelligent Laboratory Systems. Festschrift Honouring Professor D.L. Massart S.I.* **77,** 85–96 (2005).

40. Maeder, M. Evolving factor analysis for the resolution of overlapping chromatographic peaks. *Analytical Chemistry* **59,** 527–530 (1987).

41. Gampp, H., Maeder, M., Meyer, C. J. & Zuberbühler, A. D. Calculation of equilibrium constants from multiwavelength spectroscopic data-III Model-free analysis of spectrophotometric and ESR titrations. *Talanta* **32,** 1133–1139 (1985).

42. Geladi, P. & Kowalski, B. Partial least-squares regression: a tutorial. *Analytica Chimica Acta* **185,** 1–17 (1986).

43. Belsley, D. A., Kuh, E. & Welsch, R. E. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity* (Wiley-Interscience, Hoboken, N.J, 2013).

44. Wold, S., Sjöström, M. & Eriksson, L. PLS-regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems* **58,** 109–130 (2001).

45. Efron, B. Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrika* **68,** 589–599 (1981).

46. Grung, B. & Manne, R. Missing values in principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **42,** 125–139 (1998).

47. Arteaga, F. & Ferrer, A. *Missing data* in *Comprehensive chemometrics chemical and biochemical data analysis* **3** (Elsevier, Amsterdam, 2009), 285–314.

48. Little, R. J. A. & Rubin, D. B. *Statistical Analysis with Missing Data* 2 edition (Wiley-Interscience, Hoboken, N.J, 2002).

49. Muteki, K., MacGregor, J. F. & Ueda, T. Estimation of missing data using latent variable methods with auxiliary information. *Chemometrics and Intelligent Laboratory Systems* **78,** 41–50 (2005).

50. Rubin, D. B. *Multiple Imputation for Nonresponse in Surveys* (John Wiley & Sons, 2004).

51. Wise, B. & Ricker, N. *Recent Advances in Multivariate Statistical Process Control: Improving Robustness and Sensitivity* in *Proceedings of the IFAC International Symposium* (Toulouse, France, 1991), 125–130.

52. Nelson, P. R., Taylor, P. A. & MacGregor, J. F. Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and Intelligent Laboratory Systems* **35,** 45–65 (1996).

53. Walczak, B. & Massart, D. Dealing with missing data: Part II. *Chemometrics and Intelligent Laboratory Systems* **58,** 29–42 (2001).

54. Arteaga, F. & Ferrer, A. Framework for regression-based missing data imputation methods in on-line MSPC. *Journal of Chemometrics* **19,** 439–447 (2005).

55. Wold, S. *et al. Pattern recognition: Finding and using regularities in multivariate data* in *Food Research and Data Analysis* (Elsevier Applied Science, London, UK, 1983), 147–188.

56. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **39,** 1–38 (1977).

57. Schafer, J. L. *Analysis of Incomplete Multivariate Data* 1 edition (Chapman and Hall/CRC, Boca Raton, 1997).

58. Allison, P. D. *Missing Data* (SAGE Publications, 2002).

59. Tanner, M. & Wong, W. The calculation of posterior distribution by data augmentation (with discussion). *Journal of the American Statistical Association* **82,** 528–550 (1987).

60. López-Negrete de la Fuente, R. L.-N., García-Muñoz, S. & Biegler, L. T. An efficient nonlinear programming strategy for PCA models with incomplete data sets. *Journal of Chemometrics* **24,** 301–311 (2010).

61. Liu, Y. & Brown, S. D. Comparison of five iterative imputation methods for multivariate classification. *Chemometrics and Intelligent Laboratory Systems* **120,** 106–115 (2013).

62. Krzanowski, W. Missing value imputation in multivariate data using the singular value decomposition of a matrix. *Biometrical Letters* **XXV,** 31–39 (1988).

63. Dear, R. E. *A principal-component missing-data method for multiple regression models* (System Development Corp, 1959).

64. White, I. R., Royston, P. & Wood, A. M. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine* **30,** 377–399 (2011).

65. Schneider, T. Analysis of Incomplete Climate Data: Estimation of Mean Values and Covariance Matrices and Imputation of Missing Values. *Journal of Climate* **14,** 853–871 (2001).

66. Fierro, R., Golub, G., Hansen, P. & O'Leary, D. Regularization by Truncated Total Least Squares. *SIAM Journal on Scientific Computing* **18,** 1223–1241 (1997).

67. Gómez-Carracedo, M. P., Andrade, J. M., López-Mahía, P., Muniategui, S. & Prada, D. A practical comparison of single and multiple imputation methods to handle complex missing data in air quality datasets. *Chemometrics and Intelligent Laboratory Systems* **134,** 23–33 (2014).

68. Karhunen, J. Robust PCA methods for complete and missing data. *Neural Network World* **21** (2011).

69. Walczak, B. & Massart, D. Dealing with missing data. *Chemometrics and Intelligent Laboratory Systems* **58,** 15–27 (2001).

70. Camacho, J., Picó, J. & Ferrer, A. Bilinear modelling of batch processes. Part II: a comparison of PLS soft-sensors. *Journal of Chemometrics* **22,** 533–547 (2008).

71. Smilde, A., Bro, R. & Geladi, P. *Multi-way Analysis: Applications in the Chemical Sciences* (John Wiley & Sons, 2004).

72. Bro, R. Multiway calibration. Multilinear PLS. *Journal of Chemometrics* **10,** 47 –61 (1998).

73. Barker, M. & Rayens, W. Partial least squares for discrimination. *Journal of Chemometrics* **17,** 166–173 (2003).

74. Bro, R., Smilde, A. K. & de Jong, S. On the difference between low-rank and subspace approximation: improved model for multi-linear PLS regression. *Chemometrics and Intelligent Laboratory Systems* **58,** 3–13 (2001).

75. Gu, X. *Systems biology approaches to the computational modelling of trypanothione metabolism in Trypanosoma brucei* PhD thesis (University of Galsgow, Glasgow, 2010).

76. Nueda, M. J. *Statistical methods for Time Course Microarray data* PhD thesis (Universidad Politécnica de Valencia, Valencia, 2009).

77. Papini, M. *Metabolic Engineering of Central Carbon Metabolism in Saccharomyces cerevisiae* PhD thesis (Chalmers University of Technology, Göteborg, 2012).

78. Tarazona, S. *Statistical methods for transcriptomics: From microarrays to RNA-seq* PhD thesis (Universidad Politécnica de Valencia, Valencia, 2014).

79. Anderson, P. W. More Is Different. *Science* **177,** 393–396 (1972).

80. Kitano, H. Systems biology: A brief overview. *Science* **295,** 1662–1664 (2002).

81. Regenmortel, M. H. V. Reductionism and complexity in molecular biology. *EMBO Reports* **5,** 1016–1020 (2004).

82. Visconti, A. *Systems Biology: Knowledge Discovery and Reverse Engineering* PhD thesis (University of Torino, Italy, 2012).

83. Ge, H., Walhout, A. J. M. & Vidal, M. Integrating 'omic' information: a bridge between genomics and systems biology. *Trends in genetics: TIG* **19,** 551–560 (2003).

84. Westerhoff, H. & Palsson, B. The evolution of molecular biology into systems biology. *Nature Biotechnology* **22,** 1249–1252 (2004).

85. Folch-Fortuny, A. *A grey modelling approach of Pichia pastoris metabolic network* PhD thesis (Universitat de València, Valencia, 2013).

86. Smith, H., Tomb, J.-F., Dougherty, B., Fleischmann, R. & Venter, J. Frequency and distribution of DNA uptake signal sequences in the Haemophilus influenzae Rd genome. *Science* **269,** 538–540 (1995).

87. Craig Venter, J. *et al.* The sequence of the human genome. *Science* **291,** 1304–1351 (2001).

88. Hood, L. Systems biology: Integrating technology, biology, and computation. *Mechanisms of Ageing and Development* **124,** 9–16 (2003).

89. Kitano, H. *Foundations of Systems Biology. Chapter 1: Toward System-Level Understanding of Biological Systems* (The MIT Press, 2001).

90. Isidro, I. A. *et al. Design of Pathway-Level Bioprocess Monitoring and Control Strategies Supported by Metabolic Networks* in *Measurement, Monitoring, Modelling and Control of Bioprocesses* **132** (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), 193–215.

91. Otero, J. & Nielsen, J. Industrial systems biology. *Biotechnology and Bioengineering* **105,** 439–460 (2010).

92. Carrondo, M. J. T. *et al.* How can measurement, monitoring, modeling and control advance cell culture in industrial biotechnology? *Biotechnology Journal* **7,** 1522–1529 (2012).

93. Watson, J. D. & Crick, F. H. C. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature* **171,** 737–738 (1953).

94. Sebastián-León, P. *Understanding disease mechanisms with statistical models of signaling pathway activities* PhD thesis (Universitat de València, Valencia, 2016).

95. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489,** 57–74 (2012).

96. Zhang, Y., Xuan, J., de los Reyes, B. G., Clarke, R. & Ressom, H. W. Reconstruction of gene regulatory modules in cancer cell cycle by multisource data integration. *PloS One* **5,** e10268 (2010).

97. Villaverde, A. F., Ross, J. & Banga, J. R. Reverse Engineering Cellular Networks with Information Theoretic Methods. *Cells* **2,** 306–329 (2013).

98. Villaverde, A. F., Ross, J., Morán, F. & Banga, J. R. MIDER: Network Inference with Mutual Information Distance and Entropy Reduction. *PLoS ONE* **9,** e96732 (2014).

99. Wilkins, M. R. *et al.* Progress with proteome projects: why all proteins expressed by a genome should be identified and how to do it. *Biotechnology & Genetic Engineering Reviews* **13,** 19–50 (1996).

100. Kellner, R. Proteomics. Concepts and perspectives. *Fresenius' Journal of Analytical Chemistry* **366,** 517–524 (2000).

101. Fu, Y., Jarboe, L. R. & Dickerson, J. A. Reconstructing genome-wide regulatory network of E. coli using transcriptome data and predicted transcription factor activities. *BMC bioinformatics* **12,** 233 (2011).

102. Hartemink, A. J., Gifford, D. K., Jaakkola, T. S. & Young, R. A. Combining location and expression data for principled discovery of genetic regulatory network models. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing,* 437–449 (2002).

103. Savage, R. S., Ghahramani, Z., Griffin, J. E., Cruz, B. J. d. l. & Wild, D. L. Discovering transcriptional modules by Bayesian data integration. *Bioinformatics* **26,** i158–i167 (2010).

104. Tamada, Y. *et al.* Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics* **19 Suppl 2,** ii227–236 (2003).

105. Patterson, S. & Aebersold, R. Proteomics: the first decade and beyond. *Nature Genetics* **33,** 311–323 (2003).

106. Culver, J. N. & Padmanabhan, M. S. Virus-induced disease: altering host physiology one interaction at a time. *Annual Review of Phytopathology* **45,** 221–243 (2007).

107. De Las Rivas, J. & Fontanillo, C. Protein-Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks. *PLoS Comput Biol* **6,** e1000807 (2010).

108. Börnke, F. *Protein Interaction Networks* in *Analysis of Biological Networks* (John Wiley & Sons, Inc., 2008), 207–232.

109. Chandrasekhar, K., Dileep, A., Lebonah, D. E. & Kumari, J. P. A Short Review on Proteomics and its Applications. *International Letters of Natural Sciences* **17,** 77–84 (2014).

110. Phizicky, E. M. & Fields, S. Protein-protein interactions: methods for detection and analysis. *Microbiological Reviews* **59,** 94–123 (1995).

111. Brückner, A., Polge, C., Lentze, N., Auerbach, D. & Schlattner, U. Yeast two-hybrid, a powerful tool for systems biology. *International Journal of Molecular Sciences* **10,** 2763–2788 (2009).

112. Fields, S. & Song, O. A novel genetic system to detect protein-protein interactions. *Nature* **340,** 245–246 (1989).

113. Ho, Y. *et al.* Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature* **415,** 180–183 (2002).

114. Hu, C.-D., Chinenov, Y. & Kerppola, T. K. Visualization of Interactions among bZIP and Rel Family Proteins in Living Cells Using Bimolecular Fluorescence Complementation. *Molecular Cell* **9,** 789–798 (2002).

115. Kodama, Y. & Hu, C.-D. An improved bimolecular fluorescence complementation assay with a high signal-to-noise ratio. *BioTechniques* **49,** 793–805 (2010).

116. Rual, J.-F. *et al.* Towards a proteome-scale map of the human protein-protein interaction network. *Nature* **437,** 1173–1178 (2005).

117. Venkatesan, K. *et al.* An empirical framework for binary interactome mapping. *Nature Methods* **6,** 83–90 (2009).

118. Uetz, P. *et al.* A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature* **403,** 623–627 (2000).

119. Ito, T. *et al.* A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America* **98,** 4569–4574 (2001).

120. Uetz, P. *et al.* Herpesviral protein networks and their interaction with the human proteome. *Science (New York, N.Y.)* **311,** 239–242 (2006).

121. Fossum, E. *et al.* Evolutionarily conserved herpesviral protein interaction networks. *PLoS pathogens* **5,** e1000570 (2009).

122. Palsson, B. *Properties of Reconstructed Networks* (Cambridge University Press, 2006).

123. Nielsen, J. It is all about metabolic fluxes. *Journal of Bacteriology* **185,** 7031–7035 (2003).

124. Heinrich, R., Rapoport, S. M. & Rapoport, T. A. Metabolic regulation and mathematical models. *Progress in Biophysics and Molecular Biology* **32,** 1–82 (1977).

125. Schuster, S., Dandekar, T. & Fell, D. A. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends in Biotechnology* **17,** 53–60 (1999).

126. Teusink, B. *et al.* Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. *European journal of biochemistry / FEBS* **267,** 5313–5329 (2000).

127. Wiechert, W. 13C metabolic flux analysis. *Metabolic Engineering* **3,** 195–206 (2001).

128. Mahadevan, R., Edwards, J. S. & Doyle, F. J. Dynamic flux balance analysis of diauxic growth in Escherichia coli. *Biophysical Journal* **83,** 1331–1340 (2002).

129. Willemsen, A. M. *et al.* MetDFBA: incorporating time-resolved metabolomics measurements into dynamic flux balance analysis. *Molecular bioSystems* **11,** 137–145 (2015).

130. Llaneras, F. *Interval and Possibilistic Methods for Constraint-Based Metabolic Models* PhD thesis (Universidad Politécnica de Valencia, 2010).

131. Kayser, A., Weber, J., Hecht, V. & Rinas, U. Metabolic flux analysis of Escherichia coli in glucose-limited continuous culture. I. Growth-rate-dependent metabolic efficiency at steady state. *Microbiology* **151,** 693–706 (2005).

132. Tortajada, M., Llaneras, F. & Pico, J. Validation of a constraint-based model of Pichia pastoris metabolism under data scarcity. *BMC Systems Biology* **4,** 1–11 (2010).

133. Benyamini, T., Folger, O., Ruppin, E. & Shlomi, T. Flux balance analysis accounting for metabolite dilution. *Genome Biology* **11** (2010).

134. Covert, M., Knight, E., Reed, J., Herrgard, M. & Palsson, B. Integrating high-throughput and computational data elucidates bacterial networks. *Nature* **429,** 92–96 (2004).

135. Covert, M., Schilling, C. & Palsson, B. Regulation of gene expression in flux balance models of metabolism. *Journal of Theoretical Biology* **213,** 73–88 (2001).

136. Åkesson, M., Förster, J. & Nielsen, J. Integration of gene expression data into genome-scale metabolic models. *Metabolic Engineering* **6,** 285–293 (2004).

137. Stephanopulos, G. N., Aristidou, A. A. & Nielsen, J. *Metabolic engineering: principles and methodologies* (Academic Press, San Diego, 1998).

138. Price, N., Papin, J., Schilling, C. & Palsson, B. Genome-scale microbial in silico models: The constraints-based approach. *Trends in Biotechnology* **21,** 162–169 (2003).

139. Llaneras, F. & Picó, J. Stoichiometric modelling of cell metabolism. *Journal of Bioscience and Bioengineering* **105,** 1–11 (2008).

140. Schuster, S., Fell, D. A. & Dandekar, T. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology* **18,** 326–332 (2000).

141. Schuster, S., Hilgetag, C., Woods, J. H. & Fell, D. A. Reaction routes in biochemical reaction systems: algebraic properties, validated calculation procedure and example from nucleotide metabolism. *Journal of Mathematical Biology* **45,** 153–181 (2002).

142. Klamt, S. & Stelling, J. Combinatorial complexity of pathway analysis in metabolic networks. *Molecular Biology Reports* **29,** 233–236 (2002).

143. Quek, L.-E. & Nielsen, L. K. A depth-first search algorithm to compute elementary flux modes by linear programming. *BMC Systems Biology* **8,** 94 (2014).

144. Badsha, M. B., Tsuboi, R. & Kurata, H. Complementary elementary modes for fast and efficient analysis of metabolic networks. *Biochemical Engineering Journal* **90,** 121–130 (2014).

145. Wiback, S. J., Mahadevan, R. & Palsson, B. O. Reconstructing metabolic flux vectors from extreme pathways: defining the α-spectrum. *Journal of Theoretical Biology* **224,** 313–324 (2003).

146. Schwartz, J.-M. & Kanehisa, M. A quadratic programming approach for decomposing steady-state metabolic flux distributions onto elementary modes. *Bioinformatics* **21,** ii204–ii205 (2005).

147. Song, H.-S. & Ramkrishna, D. Reduction of a set of elementary modes using yield analysis. *Biotechnology and Bioengineering* **102,** 554–568 (2009).

148. Llaneras, F., Sala, A. & Picó, J. A possibilistic framework for constraint-based metabolic flux analysis. *BMC Systems Biology* **3,** 1–22 (2009).

149. Zadeh, L. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* **1,** 3–28 (1978).

150. Tortajada, M., Llaneras, F., Ramón, D. & Picó, J. Estimation of recombinant protein production in Pichia pastoris based on a constraint-based model. *Journal of Process Control* **22,** 1139–1151 (2012).

151. De Smet, R. & Marchal, K. Advantages and limitations of current network inference methods. *Nature Reviews Microbiology* **8,** 717–729 (2010).

152. Marbach, D. *et al.* Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences* **107,** 6286–6291 (2010).

153. Prill, R. J., Saez-Rodriguez, J., Alexopoulos, L. G., Sorger, P. K. & Stolovitzky, G. Crowdsourcing network inference: the DREAM predictive signaling network challenge. *Science Signaling* **4,** mr7 (2011).

154. Lecca, P. & Priami, C. Biological network inference for drug discovery. *Drug Discovery Today* **18,** 256–264 (2013).

155. Maetschke, S. R., Madhamshettiwar, P. B., Davis, M. J. & Ragan, M. A. Supervised, semi-supervised and unsupervised inference of gene regulatory networks. *Briefings in Bioinformatics,* bbt034 (2013).

156. Shannon, C. E. A Mathematical Theory of Communication. *Bell System Technical Journal* **27,** 379–423 (1948).

157. Cover, T. M. & Thomas, J. A. *Elements of Information Theory* 99 edition (Wiley-Interscience, New York, 1991).

158. Faith, J. J. *et al.* Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. *PLoS Biol* **5,** e8 (2007).

159. Margolin, A. A. *et al.* ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics* **7,** S7 (2006).

160. Zoppoli, P., Morganella, S. & Ceccarelli, M. TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC bioinformatics* **11,** 154 (2010).

161. Meyer, P. E., Kontos, K., Lafitte, F. & Bontempi, G. Information-Theoretic Inference of Large Transcriptional Regulatory Networks. *EURASIP Journal on Bioinformatics and Systems Biology* **2007,** 79879 (2007).

162. Luo, W., Hankenson, K. D. & Woolf, P. J. Learning transcriptional regulatory networks from high throughput gene expression data using continuous three-way mutual information. *BMC Bioinformatics* **9,** 467 (2008).

163. Becker, S. *et al.* Quantitative prediction of cellular metabolism with constraint-based models: The COBRA Toolbox. *Nature Protocols* **2,** 727–738 (2007).

164. Camacho, J., Pérez-Villegas, A., Rodríguez-Gómez, R. A. & Jiménez-Mañas, E. Multivariate Exploratory Data Analysis (MEDA) Toolbox for Matlab. *Chemometrics and Intelligent Laboratory Systems* **143,** 49–57 (2015).

165. Terzer, M. & Stelling, J. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics* **24,** 2229–2235 (2008).

166. Hoops, S. *et al.* COPASI–a COmplex PAthway SImulator. *Bioinformatics* **22,** 3067–3074 (2006).

167. Andersson, C. A. & Bro, R. The N-way Toolbox for MATLAB. *Chemometrics and Intelligent Laboratory Systems* **52,** 1–4 (2000).

168. *ProSensus Multivariate release 15.02* Ancaster, Ontario, Canada, 2015.

169. *PLS_ Toolbox Release 7.0.2* Manson, Washington, USA, 2012.

170. Mesarovic, M., Sreenath, S. & Keene, J. Search for organising principles: understanding in systems biology. *Systems biology* **1,** 19–27 (2004).

171. Kauffman, K. J., Prakash, P. & Edwards, J. S. Advances in flux balance analysis. *Current Opinion in Biotechnology* **14,** 491–496 (2003).

172. Feyo De Azevedo, S., Dahm, B. & Oliveira, F. Hybrid modelling of biochemical processes: A comparison with the conventional approach. *Computers and Chemical Engineering* **21,** S751–S756 (1997).

173. Ramaker, H., Van Sprang, E., Gurden, S., Westerhuis, J. & Smilde, A. Improved monitoring of batch processes by incorporating external information. *Journal of Process Control* **12,** 569–576 (2002).

174. Takane, Y. & Shibayama, T. Principal component analysis with external information on both subjects and variables. *Psychometrika* **56,** 97–120 (1991).

175. Takane, Y., Kiers, H. & de Leeuw, J. Component analysis with different sets of constraints on different dimensions. *Psychometrika* **60,** 259–280 (1995).

176. Sariyar, B., Perk, S., Akman, U. & Hortaçsu, A. Monte Carlo sampling and principal component analysis of flux distributions yield topological and modular information on metabolic networks. *Journal of Theoretical Biology* **242,** 389–400 (2006).

177. Barrett, C., Herrgard, M. & Palsson, B. Decomposing complex reaction networks using random sampling, principal component analysis and basis rotation. *BMC Systems Biology* **3,** 1–8 (2009).

178. Van Dien, S., Iwatani, S., Usuda, Y. & Matsui, K. Theoretical analysis of amino acid-producing Escherichia coli using a stoichiometric model and multivariate linear regression. *Journal of Bioscience and Bioengineering* **102,** 34–40 (2006).

179. Bro, R. PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems* **38,** 149–171 (1997).

180. Verouden, M. *et al.* Multi-way analysis of flux distributions across multiple conditions. *Journal of Chemometrics* **23,** 406–420 (2009).

181. Carinhas, N. *et al.* Hybrid metabolic flux analysis: Combining stoichiometric and statistical constraints to model the formation of complex recombinant products. *BMC Systems Biology* **5,** 1–13 (2011).

182. Dieterle, F., Ross, A., Schlotterbeck, G. & Senn, H. Metabolite projection analysis for fast identification of metabolites in metabonomics. Application in an amiodarone study. *Analytical Chemistry* **78,** 3551–3561 (2006).

183. Tondel, K. *et al.* Hierarchical Cluster-based Partial Least Squares Regression (HC-PLSR) is an efficient tool for metamodelling of nonlinear dynamic models. *BMC Systems Biology* **5,** 1–17 (2011).

184. Westerhuis, J., Derks, E., Hoefsloot, H. & Smilde, A. Grey component analysis. *Journal of Chemometrics* **21,** 474–485 (2007).

185. Ng, C. & Hussain, M. Hybrid neural network - prior knowledge model in temperature control of a semi-batch polymerization process. *Chemical Engineering and Processing: Process Intensification* **43,** 559–570 (2004).

186. Bechmann, H., Madsen, H., Poulsen, N. & Nielsen, M. Grey box modeling of first flush and incoming wastewater at a wastewater treatment plant. *Environmetrics* **11,** 1–12 (2000).

187. Sohlberg, B. Grey box modelling for model predictive control of a heating process. *Journal of Process Control* **13,** 225–238 (2003).

188. Ten Berge, J. & Smilde, A. Non-triviality and identification of a constrained Tucker3 analysis. *Journal of Chemometrics* **16,** 609–612 (2002).

189. Teixeira, A. *et al.* Cell functional enviromics: Unravelling the function of environmental factors. *BMC Systems Biology* **5,** 1–16 (2011).

190. Dragosits, M. *et al.* The effect of temperature on the proteome of recombinant Pichia pastoris. *Journal of Proteome Research* **8,** 1380–1392 (2009).

191. Solà, A. *et al.* Metabolic flux profiling of Pichia pastoris grown on glycerol/methanol mixtures in chemostat cultures at low and high dilution rates. *Microbiology* **153,** 281–290 (2007).

192. Solà, A. *Estudi del metabolisme central del carboni de Pichia pastoris* PhD thesis (Universitat Autònoma de Barcelona, 2004).

193. Jungo, C., Marison, I. & von Stockar, U. Mixed feeds of glycerol and methanol can improve the performance of Pichia pastoris cultures: A quantitative study based on concentration gradients in transient continuous cultures. *Journal of Biotechnology* **128,** 824–837 (2007).

194. Ren, H., Yuan, J. & Bellgardt, K.-H. Macrokinetic model for methylotrophic Pichia pastoris based on stoichiometric balance. *Journal of Biotechnology* **106,** 53–68 (2003).

195. d'Anjou, M. C. & Daugulis, A. J. A rational approach to improving productivity in recombinant Pichia pastoris fermentation. *Biotechnology and bioengineering* **72,** 1–11 (2001).

196. Curvers, S., Linnemann, J., Klauser, T., Wandrey, C. & Takors, R. Recombinant protein production with Pichia pastoris in continuous fermentation - Kinetic analysis of growth and product formation. *Chemical Engineering and Technology* **25,** 229–235 (2002).

197. Zhang, W., Liu, C.-P., Inan, M. & Meagher, M. Optimization of cell density and dilution rate in Pichia pastoris continuous fermentations for production of recombinant proteins. *Journal of Industrial Microbiology and Biotechnology* **31,** 330–334 (2004).

198. Schilling, B., Goodrick, J. & Wan, N. Scale-up of a high cell-density continuous culture with Pichia pastoris X-33 for the constitutive expression of rh-chitinase. *Biotechnology Progress* **17,** 629–633 (2001).

199. Benito, M. *et al.* Adjustment of systematic microarray data biases. *Bioinformatics* **20,** 105–114 (2004).

200. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8,** 118–127 (2007).

201. Luo, J *et al.* A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *The Pharmacogenomics Journal* **10,** 278–291 (2010).

202. Reese, S. E. *et al.* A New Statistic for Identifying Batch Effects in High-Throughput Genomic Data that uses Guided Principal Components Analysis. *Bioinformatics,* btt480 (2013).

203. Hadlich, F., Nöh, K. & Wiechert, W. Determination of flux directions by thermodynamic network analysis: Computing informative metabolite pools. *Mathematics and Computers in Simulation* **82,** 460–470 (2011).

204. Machado, D., Costa, R., Ferreira, E., Rocha, I. & Tidor, B. Exploring the gap between dynamic and constraint-based models of metabolism. *Metabolic Engineering* **14,** 112–119 (2012).

205. Bro, R. & Smilde, A. K. Principal component analysis. *Analytical Methods* **6,** 2812–2831 (2014).

206. Camacho, J. & Ferrer, A. Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Practical aspects. *Chemometrics and Intelligent Laboratory Systems* **131,** 37–50 (2014).

207. Ishii, N. *et al.* Multiple High-Throughput Analyses Monitor the Response of E. coli to Perturbations. *Science* **316,** 593–597 (2007).

208. Madigan, M., Martinko, J. & Parker, J. *Brock Biology of Microorganisms* 10th Edit. (Pearson Education, Inc., New Jersey, 2003).

209. Nanchen, A., Schicker, A., Revelles, O. & Sauer, U. Cyclic AMP-dependent catabolite repression is the dominant control mechanism of metabolic fluxes under glucose limitation in Escherichia coli. *Journal of Bacteriology* **190,** 2323–2330 (2008).

210. Nanchen, A., Schicker, A. & Sauer, U. Nonlinear Dependency of Intracellular Fluxes on Growth Rate in Miniaturized Continuous Cultures of Escherichia coli. *Applied and Environmental Microbiology* **72,** 1164–1172 (2006).

211. Carlson, R. & Srienc, F. Fundamental Escherichia coli biochemical pathways for biomass and energy production: identification of reactions. *Biotechnology and Bioengineering* **85,** 1–19 (2004).

212. Chung, B. K. *et al.* Genome-scale metabolic reconstruction and in silico analysis of methylotrophic yeast Pichia pastoris for strain improvement. *Microbial Cell Factories* **9,** 50 (2010).

213. Quintás, G. *et al.* Chemometric approaches to improve PLSDA model outcome for predicting human non-alcoholic fatty liver disease using UPLC-MS as a metabolic profiling tool. *Metabolomics* **8,** 86–98 (2012).

214. Kalivodová, A. *et al.* PLS-DA for compositional data with application to metabolomics. *Journal of Chemometrics* **29,** 21–28 (2015).

215. Hendrickx, D. M., Hoefsloot, H. C. J., Hendriks, M. M. W. B., Canelas, A. B. & Smilde, A. K. Global test for metabolic pathway differences between conditions. *Analytica Chimica Acta* **719,** 8–15 (2012).

216. Kanehisa, M. *et al.* From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research* **34,** D354–357 (2006).

217. Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* **28,** 27–30 (2000).

218. Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M. & Hirakawa, M. KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Research* **38,** D355–D360 (2010).

219. Heerden, J. H. v. *et al.* Lost in Transition: Start-Up of Glycolysis Yields Subpopulations of Nongrowing Cells. *Science* **343,** 1245114 (2014).

220. Canelas, A. B., van Gulik, W. M. & Heijnen, J. J. Determination of the cytosolic free NAD/NADH ratio in Saccharomyces cerevisiae under steady-state and highly dynamic conditions. *Biotechnology and Bioengineering* **100,** 734–743 (2008).

221. Nikerel, I. E., Canelas, A. B., Jol, S. J., Verheijen, P. J. T. & Heijnen, J. J. Construction of kinetic models for metabolic reaction networks: Lessons learned in analysing short-term stimulus response data. *Mathematical and Computer Modelling of Dynamical Systems* **17,** 243–260 (2011).

222. Westerhuis, J. A. *et al.* Assessment of PLSDA cross validation. *Metabolomics* **4,** 81–89 (2008).

223. Szymańska, E., Saccenti, E., Smilde, A. K. & Westerhuis, J. A. Double-check: validation of diagnostic statistics for PLS-DA models in metabolomics studies. *Metabolomics* **8,** 3–16 (2012).

224. Rodrigues, F., Ludovico, P. & Leão, C. *Sugar Metabolism in Yeasts: an Overview of Aerobic and Anaerobic Glucose Catabolism* in *Biodiversity and Ecophysiology of Yeasts* (Springer Berlin Heidelberg, 2006), 101–121.

225. Rigoulet, M. *et al.* Organization and regulation of the cytosolic NADH metabolism in the yeast Saccharomyces cerevisiae. *Molecular and Cellular Biochemistry* **256-257,** 73–81 (2004).

226. Pronk, J. T., Yde Steensma, H. & Van Dijken, J. P. Pyruvate Metabolism in Saccharomyces cerevisiae. *Yeast* **12,** 1607–1633 (1996).

227. Larsson, K., Ansell, R., Eriksson, P. & Adler, L. A gene encoding sn-glycerol 3-phosphate dehydrogenase (NAD+) complements an osmosensitive mutant of Saccharomyces cerevisiae. *Molecular Microbiology* **10,** 1101–1111 (1993).

228. Eriksson, P., André, L., Ansell, R., Blomberg, A. & Adler, L. Cloning and characterization of GPD2, a second gene encoding sn-glycerol 3-phosphate dehydrogenase (NAD+) in Saccharomyces cerevisiae, and its comparison with GPD1. *Molecular Microbiology* **17,** 95–107 (1995).

229. Norbeck, J., Pâhlman, A. K., Akhtar, N., Blomberg, A. & Adler, L. Purification and characterization of two isoenzymes of DL-glycerol-3-phosphatase from Saccharomyces cerevisiae. Identification of the corresponding GPP1 and GPP2 genes and evidence for osmotic regulation of Gpp2p expression by the osmosensing mitogen-activated protein kinase signal transduction pathway. *The Journal of Biological Chemistry* **271,** 13875–13881 (1996).

230. Albert, R. & Barabási, A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics* **74,** 47–97 (2002).

231. Newman, M. E. J. The Structure and Function of Complex Networks. *SIAM Review* **45,** 167–256 (2003).

232. Wang, Z. & Zhang, J. In Search of the Biological Significance of Modular Structures in Protein Networks. *PLoS Comput Biol* **3,** e107 (2007).

233. Lalić, J. & Elena, S. F. Magnitude and sign epistasis among deleterious mutations in a positive-sense plant RNA virus. *Heredity* **109,** 71–77 (2012).

234. Van Mechelen, I. & Smilde, A. K. A generic linked-mode decomposition model for data fusion. *Chemometrics and Intelligent Laboratory Systems. OMICS* **104,** 83–94 (2010).

235. Aerts, S. *et al.* Gene prioritization through genomic data fusion. *Nature Biotechnology* **24,** 537–544 (2006).

236. Nie, Y. & Yu, J. Mining breast cancer genes with a network based noise-tolerant approach. *BMC Systems Biology* **7,** 49 (2013).

237. Baumbach, J., Rahmann, S. & Tauch, A. Reliable transfer of transcriptional gene regulatory networks between taxonomically related organisms. *BMC Systems Biology* **3,** 8 (2009).

238. Xu, Y., Correa, E. & Goodacre, R. Integrating multiple analytical platforms and chemometrics for comprehensive metabolic profiling: Application to meat spoilage detection. *Analytical and Bioanalytical Chemistry* **405,** 5063–5074 (2013).

239. Forshed, J. *et al.* Proteomic data analysis workflow for discovery of candidate biomarker peaks predictive of clinical outcome for patients with acute myeloid leukemia. *Journal of Proteome Research* **7,** 2332–2341 (2008).

240. Lee, H., Christie, A., Xu, J. & Yoon, S. Data fusion-based assessment of raw materials in mammalian cell culture. *Biotechnology and Bioengineering* **109,** 2819–2828 (2012).

241. Rojas, J. *et al.* Chemometric analysis of screen-printed biosensor chronoamperometric responses. *Sensors and Actuators, B: Chemical* **102,** 284–290 (2004).

242. Conesa, A., Prats-Montalbán, J. M., Tarazona, S., Nueda, M. J. & Ferrer, A. A multiway approach to data integration in systems biology based on Tucker3 and N-PLS. *Chemometrics and Intelligent Laboratory Systems. OMICS* **104,** 101–111 (2010).

243. Gibbs, A. & Ohshima, K. Potyviruses and the digital revolution. *Annual Review of Phytopathology* **48,** 205–223 (2010).

244. Spence, N. J. *et al.* Economic impact of Turnip mosaic virus, Cauliflower mosaic virus and Beet mosaic virus in three Kenyan vegetables. *Plant Pathology* **56,** 317–323 (2007).

245. Ward, C. W. & Shukla, D. D. Taxonomy of potyviruses: current problems and some solutions. *Intervirology* **32,** 269–296 (1991).

246. Riechmann, J. L., Laín, S. & García, J. A. Highlights and prospects of potyvirus molecular biology. *The Journal of General Virology* **73 (Pt 1),** 1–16 (1992).

247. Elena, S. F. & Rodrigo, G. Towards an integrated molecular model of plant-virus interactions. *Current Opinion in Virology* **2,** 719–724 (2012).

248. Wei, T. *et al.* Formation of complexes at plasmodesmata for potyvirus intercellular movement is mediated by the viral protein P3N-PIPO. *PLoS pathogens* **6,** e1000962 (2010).

249. Chung, B. Y.-W., Miller, W. A., Atkins, J. F. & Firth, A. E. An overlapping essential gene in the Potyviridae. *Proceedings of the National Academy of Sciences of the United States of America* **105,** 5897–5902 (2008).

250. Allison, R., Johnston, R. E. & Dougherty, W. G. The nucleotide sequence of the coding region of tobacco etch virus genomic RNA: evidence for the synthesis of a single polyprotein. *Virology* **154,** 9–20 (1986).

251. Domier, L. L. *et al.* The nucleotide sequence of tobacco vein mottling virus RNA. *Nucleic Acids Research* **14,** 5417–5430 (1986).

252. Revers, F., Le Gall, O., Candresse, T. & Maule, A. J. New Advances in Understanding the Molecular Biology of Plant/Potyvirus Interactions. *Molecular Plant-Microbe Interactions* **12,** 367–376 (1999).

253. Urcuqui-Inchima, S., Haenni, A. L. & Bernardi, F. Potyvirus proteins: a wealth of functions. *Virus Research* **74,** 157–175 (2001).

254. Merits, A. *et al.* Proteolytic processing of potyviral proteins and polyprotein processing intermediates in insect and plant cells. *The Journal of General Virology* **83,** 1211–1221 (2002).

255. Adams, M. J., Antoniw, J. F. & Beaudoin, F. Overview and analysis of the polyprotein cleavage sites in the family Potyviridae. *Molecular Plant Pathology* **6,** 471–487 (2005).

256. Zheng, H. *et al.* Mapping the self-interacting domains of TuMV HC-Pro and the subcellular localization of the protein. *Virus Genes* **42,** 110–116 (2011).

257. Zilian, E. & Maiss, E. Detection of plum pox potyviral protein-protein interactions in planta using an optimized mRFP-based bimolecular fluorescence complementation system. *The Journal of General Virology* **92,** 2711–2723 (2011).

258. Lin, L. *et al.* Protein-protein interactions in two potyviruses using the yeast two-hybrid system. *Virus Research* **142,** 36–40 (2009).

259. Guo, D., Rajamäki, M.-L., Saarma, M. & Valkonen, J. P. T. Towards a protein interaction map of potyviruses: protein interaction matrixes of two potyviruses based on the yeast two-hybrid system. *Journal of General Virology* **82,** 935–939 (2001).

260. Shen, W. T., Wang, M. Q., Yan, P., Gao, L. & Zhou, P. Protein interaction matrix of Papaya ringspot virus type P based on a yeast two-hybrid system. *Acta Virologica* **54,** 49–54 (2010).

261. Kang, S.-H., Lim, W.-S. & Kim, K.-H. A protein interaction map of soybean mosaic virus strain G7H based on the yeast two-hybrid system. *Molecules and Cells* **18,** 122–126 (2004).

262. Yambao, M. L. M., Masuta, C., Nakahara, K. & Uyeda, I. The central and C-terminal domains of VPg of Clover yellow vein virus are important for VPg-HCPro and VPg-VPg interactions. *The Journal of General Virology* **84,** 2861–2869 (2003).

263. Carrasco, P., de la Iglesia, F. & Elena, S. F. Distribution of Fitness and Virulence Effects Caused by Single-Nucleotide Substitutions in Tobacco Etch Virus. *Journal of Virology* **81,** 12979–12984 (2007).

264. Dayhoff, M. O. & Schwartz, R. M. *Chapter 22: A model of evolutionary change in proteins* in *In Atlas of Protein Sequence and Structure* **5** (1978), 345–358.

265. Rasmussen, M. A. & Bro, R. A tutorial on the Lasso approach to sparse modeling. *Chemometrics and Intelligent Laboratory Systems* **119,** 21–31 (2012).

266. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67,** 301–320 (2005).

267. Buydens, L. M. C. Towards tsunami-resistant chemometrics. *The analytical Scientist* **813,** 24–30 (2013).

268. *Citrus Fruit, Fresh and Processed, Annual Statistics 2012* tech. rep. (Food and Agriculture Organization of the United Nations (FAO), 2012).

269. Palou, L. *Penicillium digitatum, Penicillium italicum (Green Mold, Blue Mold)* in *Postharvest Decay* (Elsevier, 2014), 45–102.

270. Holmes, G. J. & Eckert, J. W. Sensitivity of Penicillium digitatum and P. italicum to Postharvest Citrus Fungicides in California. *Phytopathology* **89,** 716–721 (1999).

271. Fullerton, R., Tyson, J. & Sale, P. *Citrus Diseases* in *Growing Citrus in New Zealand* (New Zealand Citrus Growers Inc, Wellington, 2001).

272. Ogawa, Y. *et al.* Detection of Rotten Citrus Fruit Using Fluorescent Images. *The Review of Laser Engineering* **39,** 255–261 (2011).

273. Blanc, P. G. R., Blasco, I. J., Moltó, G. E., Gómez, S. J. & Cubero, G. S. *pat.* EP2133157 B1. International Classification B07C5/342; Cooperative Classification G01N2021/845, G01N21/85, G01N2021/8466, B07C5/363, B07C5/3422 (2013).

274. Momin, M. A. *et al.* Investigation of Excitation Wavelength for Fluorescence Emission of Citrus Peels based on UV-VIS Spectra. *Engineering in Agriculture, Environment and Food* **5,** 126–132 (2012).

275. Obenland, D., Margosan, D., Smilanick, J. L. & Mackey, B. Ultraviolet Fluorescence to Identify Navel Oranges with Poor Peel Quality and Decay. *HortTechnology* **20,** 991–995 (2010).

276. Lorente, D., Zude, M., Idler, C., Gómez-Sanchis, J. & Blasco, J. Laser-light backscattering imaging for early decay detection in citrus fruit using both a statistical and a physical model. *Journal of Food Engineering* **154,** 76–85 (2015).

277. Gómez, J., Blasco, J., Moltó, E. & Camps-Valls, G. Hyperspectral detection of citrus damage with Mahalanobis kernel classifier. *Electronics Letters* **43,** 1082–1084 (2007).

278. Lorente, D. *et al.* Recent Advances and Applications of Hyperspectral Imaging for Fruit and Vegetable Quality Assessment. *Food and Bioprocess Technology* **5,** 1121–1142 (2011).

279. Gómez-Sanchis, J. *et al.* Development of a Hyperspectral Computer Vision System Based on Two Liquid Crystal Tuneable Filters for Fruit Inspection. Application to Detect Citrus Fruits Decay. *Food and Bioprocess Technology* **7,** 1047–1056 (2013).

280. Gómez-Sanchis, J. *et al.* Hyperspectral LCTF-based system for classification of decay in mandarins caused by Penicillium digitatum and Penicillium italicum using the most relevant bands and non-linear classifiers. *Postharvest Biology and Technology* **82,** 76–86 (2013).

281. Lorente, D. *et al.* Comparison of ROC Feature Selection Method for the Detection of Decay in Citrus Fruit Using Hyperspectral Images. *Food and Bioprocess Technology* **6,** 3613–3619 (2012).

282. Qin, J., Burks, T. F., Ritenour, M. A. & Bonn, W. G. Detection of citrus canker using hyperspectral reflectance imaging with spectral information divergence. *Journal of Food Engineering* **93,** 183–191 (2009).

283. Qin, J., Burks, T. F., X. Zhao, N. Niphadkar & M. A. Ritenour. Multispectral Detection of Citrus Canker Using Hyperspectral Band Selection. *Transactions of the ASABE* **54,** 2331–2341 (2011).

284. Qin, J., Burks, T. F., Zhao, X., Niphadkar, N. & Ritenour, M. A. Development of a two-band spectral imaging system for real-time citrus canker detection. *Journal of Food Engineering* **108,** 87–93 (2012).

285. Li, J., Rao, X., Ying, Y. & Wang, D. Detection of navel oranges canker based on hyperspectral imaging technology. *Nongye Gongcheng Xuebao / Transactions of the Chinese Society of Agricultural Engineering* **26,** 222–228 (2010).

286. Li, J., Rao, X. & Ying, Y. Development of algorithms for detecting citrus canker based on hyperspectral reflectance imaging. *Journal of the Science of Food and Agriculture* **92,** 125–134 (2012).

287. Geladi, P. & Grahn, H. *Multivariate Image Analysis* 1 edition (Wiley, Chichester ; New York, 1997).

288. Prats-Montalbán, J., de Juan, A. & Ferrer, A. Multivariate image analysis: A review with applications. *Chemometrics and Intelligent Laboratory Systems* **107,** 1–23 (2011).

289. Prats-Montalbán, J. M. & Ferrer, A. Integration of colour and textural information in multivariate image analysis: defect detection and classification issues. *Journal of Chemometrics* **21,** 10–23 (2007).

290. Prats-Montalbán, J. M., Cocchi, M. & Ferrer, A. N-way modeling for wavelet filter determination in multivariate image analysis. *Journal of Chemometrics* **29,** 379–388 (2015).

291. Shrestha, S., Deleuran, L. C., Olesen, M. H. & Gislum, R. Use of Multispectral Imaging in Varietal Identification of Tomato. *Sensors (Basel, Switzerland)* **15,** 4496–4512 (2015).

292. Calvini, R., Ulrici, A. & Amigo, J. M. Practical comparison of sparse methods for classification of Arabica and Robusta coffee species using near infrared hyperspectral imaging. *Chemometrics and Intelligent Laboratory Systems* **146,** 503–511 (2015).

293. Yu, K.-Q. *et al.* Application of Visible and Near-Infrared Hyperspectral Imaging for Detection of Defective Features in Loquat. *Food and Bioprocess Technology* **7,** 3077–3087 (2014).

294. Gao, J.-F., Zhang, H.-L., Kong, W.-W. & He, Y. Nondestructive discrimination of waxed apples based on hyperspectral imaging technology. chi. *Spectroscopy and Spectral Analysis* **33,** 1922–1926 (2013).

295. Navarro, L. *et al. The citrus variety improvement program in Spain in the period 1975-2000* in *7th IOCV Conference* (Riverside, 2002), 306–316.

296. Palou, L., Smilanick, J. L., Usall, J. & Viñas, I. Control of Postharvest Blue and Green Molds of Oranges by Hot Water, Sodium Carbonate, and Sodium Bicarbonate. *Plant Disease* **85,** 371–376 (2001).

297. Fearn, T., Riccioli, C., Garrido-Varo, A. & Guerrero-Ginel, J. E. On the geometry of SNV and MSC. *Chemometrics and Intelligent Laboratory Systems* **96,** 22–26 (2009).

298. Bijlsma, S. *et al.* Large-scale human metabolomics studies: a strategy for data (pre-) processing and validation. *Analytical Chemistry* **78,** 567–574 (2006).

299. Favilla, S., Durante, C., Vigni, M. L. & Cocchi, M. Assessing feature relevance in NPLS models by VIP. *Chemometrics and Intelligent Laboratory Systems. Multiway and Multiset Methods* **129,** 76–86 (2013).

300. Wächter, A. & Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* **106,** 25–57 (2005).

301. Forina, M., Armanino, C., Lanteri, S. & Tiscornia, E. *Classification of Olive Oils from their Fatty Acid Composition* in *Food Research and Data Analysis* (Elsevier Applied Science, London, UK, 1983), 189–214.

302. Scott A. Hutzler, G. B. B. *Remote Near-Infrared Fuel Monitoring System* tech. rep. (U.S. Army TARDEC Fuels and Lubricants Research Facility, Southwest Research Institute, San Antonio, USA, 1997), 33.

303. Arteaga, F. & Ferrer, A. How to simulate normal data sets with the desired correlation structure. *Chemometrics and Intelligent Laboratory Systems* **101,** 38–42 (2010).

304. Arteaga, F. & Ferrer, A. Building covariance matrices with the desired structure. *Chemometrics and Intelligent Laboratory Systems* **127,** 80–88 (2013).

305. Camacho, J. Visualizing Big data with Compressed Score Plots: Approach and research challenges. *Chemometrics and Intelligent Laboratory Systems* **135,** 110–125 (2014).

306. *SIMCA release 14* Umea, Sweden, 2015.

307. Wu, C.-C., Huang, H.-C., Juan, H.-F. & Chen, S.-T. GeneNetwork: an interactive tool for reconstruction of genetic networks using microarray data. *Bioinformatics* **20,** 3691–3693 (2004).

308. Gustafsson, M., Hörnquist, M. & Lombardi, A. Constructing and analyzing a large-scale gene-to-gene regulatory network–lasso-constrained inference and biological validation. *IEEE/ACM transactions on computational biology and bioinformatics* **2,** 254–261 (2005).

309. Guthke, R., Möller, U., Hoffmann, M., Thies, F. & Töpfer, S. Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection. *Bioinformatics* **21,** 1626–1634 (2005).

310. Schulze, S., Henkel, S. G., Driesch, D., Guthke, R. & Linde, J. Computational prediction of molecular pathogen-host interactions based on dual transcriptome data. *Frontiers in Microbiology* **6,** 65 (2015).

311. Hurley, D. *et al.* Gene network inference and visualization tools for biologists: application to new human transcriptome datasets. *Nucleic Acids Research* **40,** 2377–2398 (2012).

312. Souto, M. C. d., Jaskowiak, P. A. & Costa, I. G. Impact of missing data imputation methods on gene expression clustering and classification. *BMC Bioinformatics* **16,** 64 (2015).

313. Guitart-Pla, O., Kustagi, M., Rügheimer, F., Califano, A. & Schwikowski, B. The Cyni framework for network inference in Cytoscape. *Bioinformatics* **31,** 1499–1501 (2015).

314. MacGregor, J. & Kourti, T. Statistical process control of multivariate processes. *Control Engineering Practice* **3,** 403–414 (1995).

315. Stanimirova, I., Daszykowski, M. & Walczak, B. Dealing with missing values and outliers in principal component analysis. *Talanta* **72,** 172–178 (2007).

316. Abdi, H. & Williams, L. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* **2,** 433–459 (2010).

317. Camacho, J., Picó, J. & Ferrer, A. The best approaches in the on-line monitoring of batch processes based on PCA: Does the modelling structure matter? *Analytica Chimica Acta* **642,** 59–68 (2009).

318. González-Martínez, J., de Noord, O. & Ferrer, A. Multisynchro: A novel approach for batch synchronization in scenarios of multiple asynchronisms. *Journal of Chemometrics* **28,** 462–475 (2014).

319. Samoilov, M. S. *Reconstruction and functional analysis of general chemical reactions and reaction networks* (Stanford University, 1997).

320. Samoilov, M., Arkin, A. & Ross, J. On the deduction of chemical reaction pathways from measurements of time series of concentrations. *Chaos (Woodbury, N.Y.)* **11,** 108–114 (2001).

321. Cantone, I. *et al.* A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell* **137,** 172–181 (2009).

322. Arkin, A., Shen, P. & Ross, J. A Test Case of Correlation Metric Construction of a Reaction Pathway from Measurements. *Science* **277,** 1275–1279 (1997).

323. Schaffter, T., Marbach, D. & Floreano, D. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics* **27,** 2263–2270 (2011).

324. Marbach, D., Schaffter, T., Mattiussi, C. & Floreano, D. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology* **16,** 229–239 (2009).

325. Brás, L. & Menezes, J. Dealing with gene expression missing data. *IEE Proceedings: Systems Biology* **153,** 105–119 (2006).

326. Zarzo, M. & Martí, P. Modeling the variability of solar radiation data among weather stations by means of principal components analysis. *Applied Energy* **88,** 2775–2784 (2011).

327. Quevedo, J. *et al.* Estimating missing and false data in flow meters of a water distribution network. **6,** 1181–1186 (2006).

328. Magán-Carrión, R., Pulido-Pulido, F., Camacho, J. & García-Teodoro, P. Tampered data recovery in WSNs through dynamic PCA and variable routing strategies. *Journal of Communications* **8,** 738–750 (2013).

329. Visky, D. *et al.* Characterisation of reversed-phase liquid chromatographic columns by chromatographic tests: Rational column classification by a minimal number of column test parameters. *Journal of Chromatography A* **1012,** 11–29 (2003).

330. Saccenti, E. & Camacho, J. On the use of the observation-wise k-fold operation in PCA cross-validation. *Journal of Chemometrics* **29,** 467–478 (2015).

331. Ristolainen, M., Alén, R., Malkavaara, P. & Pere, J. Reflectance FTIR Microspectroscopy for Studying Effect of Xylan Removal on Unbleached and Bleached Birch Kraft Pulps. *Holzforschung* **56,** 513–521 (2005).

332. Keenan, M. R. Maximum likelihood principal component analysis of time-of-flight secondary ion mass spectrometry spectral images. *Journal of Vacuum Science & Technology A* **23,** 746–750 (2005).

333. Choi, S. W., Martin, E. B., Morris, A. J. & Lee, I.-B. Fault Detection Based on a Maximum-Likelihood Principal Component Analysis (PCA) Mixture. *Industrial & Engineering Chemistry Research* **44,** 2316–2327 (2005).

334. Karakach, T. K., Wentzell, P. D. & Walter, J. A. Characterization of the measurement error structure in 1D 1H NMR data for metabolomics studies. *Analytica Chimica Acta* **636,** 163–174 (2009).

335. Wentzell, P. D. & Hou, S. Exploratory data analysis with noisy measurements. *Journal of Chemometrics* **26,** 264–281 (2012).

336. Mailier, J., Remy, M. & Vande Wouwer, A. Stoichiometric identification with maximum likelihood principal component analysis. *Journal of Mathematical Biology* **67,** 739–765 (2013).

337. Hoefsloot, H. C. J., Verouden, M. P. H., Westerhuis, J. A. & Smilde, A. K. Maximum likelihood scaling (MALS). *Journal of Chemometrics* **20,** 120–127 (2006).

338. Dadashi, M., Abdollahi, H. & Tauler, R. Maximum Likelihood Principal Component Analysis as initial projection step in Multivariate Curve Resolution analysis of noisy data. *Chemometrics and Intelligent Laboratory Systems* **118,** 33–40 (2012).

339. Andrews, D. T. & Wentzell, P. D. Applications of maximum likelihood principal component analysis: incomplete data sets and calibration transfer. *Analytica Chimica Acta* **350,** 341–352 (1997).

340. Ho, P., Silva, M. & Hogg, T. Multiple imputation and maximum likelihood principal component analysis of incomplete multivariate data from a study of the ageing of port. *Chemometrics and Intelligent Laboratory Systems* **55,** 1–11 (2001).

341. Stanimirova, I. Practical approaches to principal component analysis for simultaneously dealing with missing and censored elements in chemical data. *Analytica Chimica Acta* **796,** 27–37 (2013).

342. Guilment, J., Markel, S. & Windig, W. Infrared Chemical Micro-Imaging Assisted by Interactive Self-Modeling Multivariate Analysis. *Applied Spectroscopy* **48,** 320–326 (1994).

343. Windig, W. & Markel, S. Simple-to-use interactive self-modeling mixture analysis of FTIR microscopy data. *Journal of Molecular Structure* **292,** 161–170 (1993).

344. Windig, W. Spectral data files for self-modeling curve resolution with examples using the Simplisma approach. *Chemometrics and Intelligent Laboratory Systems* **36,** 3–16 (1997).

345. Martens, H. & Næs, T. *Multivariate Calibration* First Edition (John Wiley & Sons Ltd., 1989).

346. Esbensen, K. *Multivariate Data Analysis - in practice* Fifth Edition (CAMO Process AS, 2002).

347. Feudale, R. N. *et al.* Transfer of multivariate calibration models: a review. *Chemometrics and Intelligent Laboratory Systems* **64,** 181–192 (2002).

348. Wang, Y., D.J., V. & Kowalski, B. Multivariate instrument standardization. *Anal. Chem.* **63,** 2750–2756 (1991).

349. Bouveresse, E. & Massart, D. Standardisation of near-infrared spectrometric instruments: a review. *Vib. Spectrosc.* **11,** 3–15 (1996).

350. Fearn, T. Standardisation and calibration transfer for near infrared instruments: a review. *Journal of Near Infrared Spectroscopy* **9,** 229–244 (2001).

351. De Noord, O. Multivariate calibration standardisation. *Chemometr. Intell. Lab.* **25,** 85–97 (1994).

352. Wise, B., Martens, H., Martin, H., Bro, R. & Brackhoff, P. *Calibration transfer by Generalized Least Squares* tech. rep. (Eigenvector Research Inc.).

353. Bouveresse, E. & Massart, D. Improvement of the piecewise direct standardisation procedure for the transfer of NIR spectra for multivariate calibration. *Chemometr. Intell. Lab.* **2,** 201–213 (1996).

354. Wang, Y. & Kowalski, B. Calibration transfer and measurement stability of near-infrared spectrometers. *Appl. Spectrosc.* **46,** 764–771 (1992).

355. Alves Barata, J. & Hussein, M. The Moore-Penrose pseudoinverse: a tutorial review of the theory. *Braz. J. Phys.* **42,** 146–165 (2012).

356. Kennard, R. & Stone, L. Computer aided design of experiments. *Technometrics* **11,** 137–148 (1969).

357. Daszykowski, M., Walczak, B. & Massart, D. Representative subset selection. *Anal. Chim. Acta* **468,** 91–103 (2002).

358. Vitale, R. *et al.* A rapid and non-invasive method for authenticating the origin of pistachio samples by NIR spectroscopy and chemometrics. *Chemometrics and Intelligent Laboratory Systems* **121,** 90–99 (2013).

359. Fernández-Pierna, J., Vermeulen, P., Lecler, B., Baeten, V. & Dardenne, P. Calibration transfer from dispersive instruments to handheld spectrometers. *Appl. Spectrosc.* **64,** 644–647 (2010).

360. *The Unscrambler X Release 10.4* As, Norway, 2016.

361. Puwakkatiya-Kankanamage, E. H., García-Muñoz, S. & Biegler, L. T. An optimization-based undeflated PLS (OUPLS) method to handle missing data in the training set. *Journal of Chemometrics* **28,** 575–584 (2014).

362. Hald, A. *Statistical Theory with Engineering Applications* 1St Edition edition (John Wiley & Sons Inc, New York etc., 1952).

363. Kubinyi, H. Evolutionary variable selection in regression and PLS analyses. *Journal of Chemometrics* **10,** 119–133 (1996).

Abbreviations and acronyms

2CV: double cross validation, 126
3CV: triple cross validation, 126
ALS: alternating least squares, 23
ANN: artificial neural network, 58
ANOVA: analysis of variance, 17
AP-MS: affinity purification with mass spectrometry, 39
ARACNE: algorithm for the reconstruction of accurate cellular networks, 47
BPEG: BioProcess Engineering Group, 3
BWU: batch-wise unfolding, 31
BiFC: bimolecular fluorescence complementation, 39
CCD: charge-coupled sensors, 161
CC: complete case analysis, 30
CI: confidence interval, 26
CLR: context likelihood relatedness, 47
CMC: correlation metric method, 197
CMR: conditional mean replacement, 29
COBRA: constraints based reconstruction and analysis, 52
COPASI: complex pathway simulation, 52
CV: cross validation, 26
CYVV: *Clover yellow vein virus*, 141
DA: data augmentation, 30
DA: variable-wise unfolding, 31
DCS: digital control system, 27
DNA: deoxyribonucleic acid, 34
DOE: design of experiments, 147
E-M: expectation maximization, 30
EFA: evolving factor analysis, 23
EM: elementary mode, 7, 43
EP: extreme pathway, 44
FBA: flux balance analysis, 40
FPI: research personnel formation, 3
FTIR: Fourier transformed infrared, 8

GC-MS: gas chromatography - gas spectrometry, 40
GCA: grey component analysis, 58
GCSC: Group of Control of Complex Systems, 3
GIEM: Multivariate Statistical Engineering Group, 3
GIP: general iterative principal component imputation, 30, 188
GPR: gene-protein-reaction, 39
GRN: gene regulatory network, 38
GSSP: Group of Statistics and Stochastic Processes, 3
GUI: graphical user-friendly interface, 8
HC-PLSR: hierarchical clustering partial least squares regression, 58
HSI: hyperspectral imaging, 156
IA: iterative algorithm, 29
IBMCP-CSIC: Institute of Cellular and Molecular Plant Biology - Spanish Research Council, 8
IIM-CSIC: Marine Research Institute - Spanish Research Council, 3
IVIA: Valencian Institute for Agricultural Research, 9
JYPLS: joint-Y partial least squares, 26
KDR: known data regression, 29
KS: Kennard-Stone, 233
LC-MS: liquid chromatography - gas spectrometry, 40
LCTF: liquid crystal tunable filter, 158
LI: linear interpolation, 194
LSD: least significance intervals, 183
LV: latent variable, 24
MAR: missing at random, 28
MB: model building, 29
MCAR: missing completely at random, 28
MCR-ALS: multivariate curve resolution alternating least squares, 23
MCR: multivariate curve resolution, 7, 23
MC: monte carlo, 7
MDI: Missing Data Imputation, 8, 206
MD: missing data, 6
MEDA Toolbox: Multivariate Exploratory Data Analysis Toolbox, 52
MEDA: missing data methods in the context of exploratory data analysis, 7, 21
ME: model exploitation, 29
MFA: metabolic flux analysis, 40
MIA: multivariate image analysis, 53
MICE: multiple imputation by chained equations, 30, 188
MIDER: mutual information distance and entropy reduction, 8, 47
MINLP: mixed integer nonlinear programming, 87
MI: mean imputation, 30
ML-KDR: maximum likelihood known data regression, 219
ML-TSR: maximum likelihood trimmed score regression, 219
MLPCA: maximum likelihood principal component analysis, 8, 22

MLR: multivariate linear regression, 24

ML: maximum likelihood, 8

MRMR: maximum relevance/minimum redundancy, 47

MSC: multivariate scatter correction, 160

MetDFBA: time-resolved metabolomics combined with dynamic flux balance analysis, 40

MultiScaleS: multi-scale inference, monitoring, optimization and control: from engineered cells to bioreactors, 3

NIPALS: nonlinear iterative partial least squares, 26, 29

NIR: near infrared, 6

NI: non-ignorable, 28

NLP: nonlinear programming, 30

NMAR: not missing at random, 28

NMR: nuclear magnetic resonance, 216

NN: nearest neighbor, 194

NPLS-DA: N-way partial least squares discriminant analysis, 32

NPLS: N-way partial least squares, 32

ODE: ordinary differential equation, 40

ORF: open reading framework, 140

PAM: point accepted matrix, 145

PARAFAC: parallel factor analysis, 58

PCA-MB: principal component analysis model building, 29

PCA-ME: principal component analysis model exploitation, 29

PCA: principal component analysis, 7, 19

PCR: principal component regression, 24

PC: principal component, 19

PDS: piecewise direct standardisation, 52, 231

PEMA: principal elementary mode analysis, 7, 84

PEM: principal elementary mode, 85

PLP: projection to latent pathways, 59

PLS: partial least squares, 8, 24

PMP: projection to the model plane, 29

PPIN: protein-protein interaction network, 39

PPI: protein-protein, 38

PPV: *Plum pox virus*, 140

PRESS: sum of squares of the predicted error, 210

PSbMV: *Pea seed-borne mosaic virus*, 141

PVA: *Potato virus A*, 141

QSAR: quantitative structure-activity relationship, 27

RC: relevance coefficient, 142

RGB: red-green-blue, 158

RMSECV: root mean square error in CV, 235

RMSEP: root mean square error in prediction, 235

RNA: ribonucleic acid, 8

SCP: single component projection, 29
SIMPLISMA: simple-to-use interactive self-modeling analysis, 23
SMV-G7H: *Soybean mosaic virus - GH7*, 141
SMV-P: *Soybean mosaic virus - Pinellia ternate isolate*, 141
SMV: *Soybean mosaic virus*, 140
SNV: standard normal variate, 160
SPE: squared prediction error, 19
SVD: singular value decomposition, 23
SWRI: Southwest Research Institute, 182
SYSV-O: *Shallot yellow strip virus-onion isolate*, 141
SynBioFactory: synthetic biology for bioproduction enhancement: design, optimization, monitoring and control, 3
SynBio: synthetic biology, 5
TCA: tricarboxylic nucelic acid, 67
TEV: *Tobacco etch virus*, 138
TRI: trimmed score method, 29
TSR: trimmed score regression, 29
TVMV: *Tobacco vein mottling virus*, 140
TuMV: *Turnip mosaic virus*, 140
UCV: Catholic University of Valencia, 8
UNL: New University of Lisbon, 8
UPCT: Technical University of Cartagena, 3
UPV: Technical University of Valencia, 3
UV: ultraviolet, 156
UvA: University of Amsterdam, 8
VIS: visible, 9
Y2H: yeast two-hybrid, 39
ckf: column-wise fold, 210
dFBA: dynamic flux balance analysis, 40
dynEMA: dynamic elementary mode analysis, 7, 123
dynEMR-DA: dynamic elementary mode regression discriminant analysis, 7, 124
dynEM: dynamic elementary mode, 122
ekf: element-wise fold, 210
r-EM: expectation maximization based on ridge regression, 30
t-EM: expectation maximization based on truncated total least squares regression, 30, 189