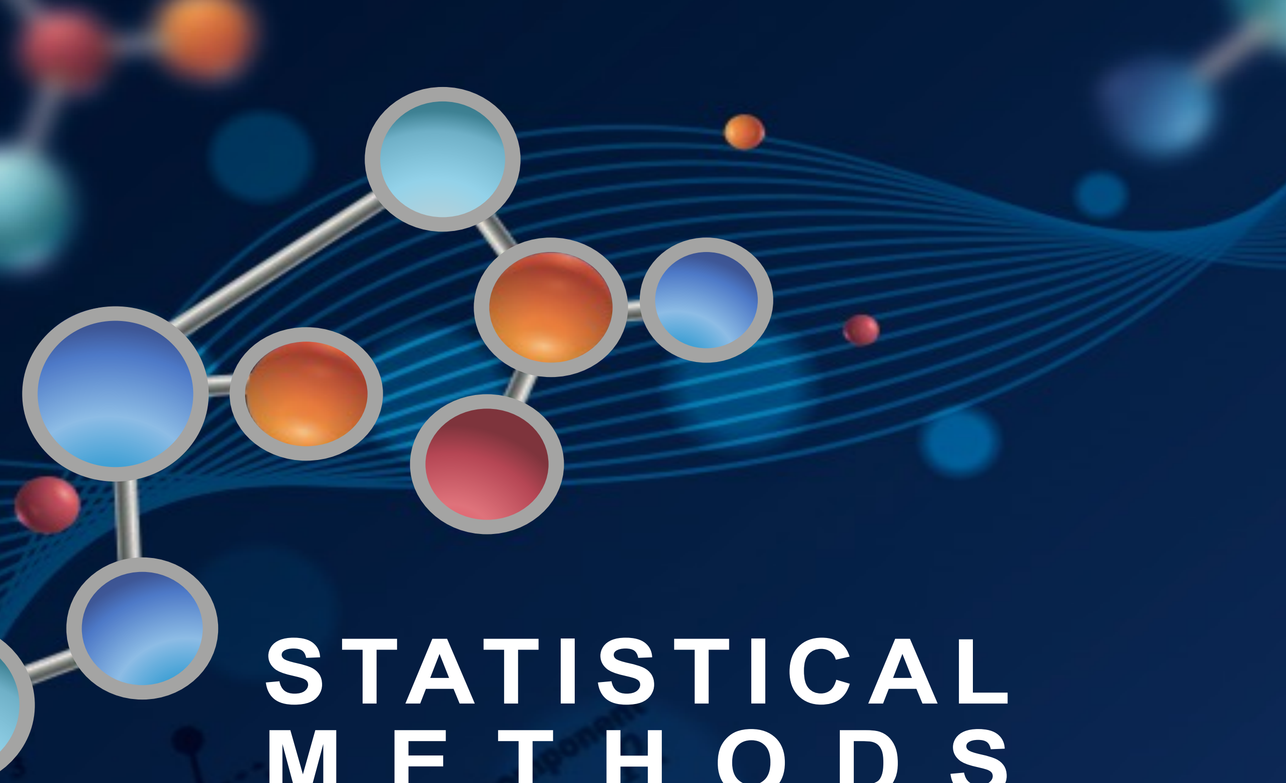


STATISTICAL METHODS DEVELOPMENT

FOR THE MULTIOMIC SYSTEMS BIOLOGY

MANUEL UGIDOS GUERRERO

PHD THESIS

SUPERVISORS:

DR. SONIA TARAZONA CAMPOS

DR. ANA CONESA CEGARRA

DR. ALBERTO J. FERRER RIQUELME

JANUARY, 2023

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Statistical methods development for the multiomic systems biology

Author:	Manuel Ugidos Guerrero
Supervisors:	Dr. Sonia Tarazona Campos Dr. Ana Conesa Cegarra Dr. Alberto J. Ferrer Riquelme

Abstract

Systems Biology research has expanded over the last years together with the development of omic technologies. The combination and simultaneous analysis of different kind of omic data allows the study of the connections and relationships between different cellular layers. Indeed, multiomic integration strategies provides a key source of knowledge about the cell as a system. The present Ph.D. thesis aims to study, develop and apply multiomic integration approaches to the field of systems biology.

The still high cost of omics technologies makes it difficult for most laboratories to afford a complete multiomic study. However, the wide availability of omic data in public repositories allows the use of these already generated data. Unfortunately, the combination of omic data from different sources provokes the appearance of unwanted noise in data, known as batch effect. Batch effect impairs the correct integrative analysis of the data. Therefore, the use of so-called Batch Effect Correction Algorithms is necessary. As of today, there is a large number of such algorithms based on different statistical models and methods that correct batch effect and are part of the data pre-processing steps. However, the existing methods are not intended for multi-omics designs as they only allow the correction of the same type of omic data that must be measured across all batches. For this reason, we developed MultiBaC algorithm, which removes batch effect in multiomic

designs, allowing the correction of data that are not measured across all batches. MultiBaC is based on PLS regression and ANOVA-SCA models and was validated and evaluated on different datasets. We also present MultiBaC as an R package to facilitate the use of this tool.

Most existing multiomic integration approaches are multivariate methods based on latent space analysis. These methods are known as data-driven as they are based on the search for correlations to determine the relationships between the different variables. Data-driven methods require a large number of observations or samples to find robust and/or significant correlations among features. Unfortunately, in the molecular biology field, data sets with a large number of samples are not very common, again due to the high cost of generating omic data. As an alternative to data-driven methods, some multiomic integration strategies are based on model-driven approaches. These methods can be fitted with a smaller number of observations and are very useful for finding mechanistic relationships between different cellular components. However, model-driven methods require a priori information, which is usually a metabolic model of the organism under study. Currently, only transcriptomics and quantitative metabolomics have been successfully integrated using model-driven methods. Nonetheless, quantitative metabolomics is not very widespread and most laboratories generate non-quantitative or semi-quantitative metabolomics, which cannot be integrated with current methods. To address this issue, we developed MAMBA, a model-driven multiomic integration method

that relies on mathematical optimization problems and is able to jointly analyze non-quantitative or semi-quantitative metabolomics with other types of gene-centric omic data, such as transcriptomics. MAMBA was compared to other existing methods in terms of metabolite prediction accuracy and was applied to a multiomic dataset generated within the PROMETEO project, in which this thesis is framed. MAMBA proved to capture the known biology of our experimental design and was useful for deriving new findings and biological hypotheses.

Altogether, this thesis presents useful tools for the field of systems biology, covering both the pre-processing of multiomic datasets and their subsequent statistical integrative analysis.

Resumen

La investigación en Biología de Sistemas se ha expandido en los últimos años junto con el desarrollo de las tecnologías ómicas. La combinación y el análisis simultáneo de diferentes tipos de datos ómicos permite el estudio de las conexiones y relaciones entre los diferentes niveles de organización celular, lo cuál permite una visión a nivel de sistema del organismo estudiado. La presente tesis doctoral tiene como objetivo estudiar, desarrollar y aplicar estrategias de integración multiómica al campo de la biología de sistemas.

El todavía elevado coste de las tecnologías ómicas, dificulta que la mayoría de laboratorios puedan abordar un estudio multiómico completo. No obstante, la gran disponibilidad de datos ómicos en repositorios públicos, permite el uso de estos datos ya generados. Desafortunadamente, la combinación de datos ómicos provenientes de diferentes orígenes, da lugar a la aparición de un ruido no deseado en los datos, lo que se conoce como efecto lote o *"batch effect"* en inglés. El efecto lote impide el correcto análisis conjunto de los datos y, por lo tanto, es necesario el uso de los llamados Algoritmos de Corrección de Efecto Lote para eliminarlo. En la actualidad, existe un gran número de éstos algoritmos que corrigen el efecto lote que se basan en diferentes métodos y modelos estadísticos, y que forman parte del paso de preprocesado de los datos. Sin embargo, los métodos existentes no están pensados para los diseños multiómicos ya que solo permiten la cor-

rección de un mismo tipo de dato ómico que debe haber sido medido en todos los lotes o batches. Por esta razón desarrollamos nuestra herramienta MultiBaC basada en la regresión PLS y modelos ANOVA-SCA, que permite la corrección del efecto lote en diseños multiómicos, permitiendo la corrección de datos que no hayan sido medidos en todos los lotes. En este trabajo, MultiBaC fué validado y evaluado en diferentes conjuntos de datos, además presentamos MultiBaC como paquete de R para facilitar el uso de nuestra herramienta.

La mayoría de métodos existentes de integración multiómica son métodos multivariantes basados en el análisis del espacio latente. Estos métodos se conocen como "dirigidos por datos" o *"data-driven"* en inglés. Este tipo de métodos se basan en la búsqueda de correlaciones para determinar las relaciones entre las distintas variables. Los métodos dirigidos por datos necesitan de gran cantidad de observaciones o muestras para poder encontrar correlaciones robustas y/o significativas entre las variables. Lamentablemente, en el mundo de la biología molecular, los conjuntos de datos con un gran número de muestras no suelen ser muy habituales, debido de nuevo al elevado coste de generación de los datos ómicos. Como alternativa a los métodos dirigidos por datos, algunas estrategias de integración multiómicas se basan en métodos "dirigidos por modelos" o *"model-driven"* en inglés. Estos métodos pueden ajustarse con un menor número de observaciones y son muy útiles para encontrar relaciones mecanísticas entre los diferentes componentes celulares. Sin embargo, los métodos dirigidos por modelos necesitan de una información a priori, el modelo, que normalmente es un modelo metabólico del organismo estudiado. Actualmente, únicamente transcriptómica y metabolómica cuantitativa, han sido los dos tipos de dato ómico que se han integrado con éxito usando métodos dirigidos por modelos. No obstante, la metabolómica cuantitativa no está muy extendida y la mayoría de laboratorios generan metabolómica no cuantitativa o semi-cuantitativa, las cuáles no pueden integrarse con los métodos actuales. Para contribuir en esta cuestión, desarrollamos MAMBA, una herramienta de integración multiómica dirigida por modelos y basada en métodología de optimización matemática, que es capaz de analizar conjuntamente metabolómica no cuantitativa o semi-cuantitativa con otro tipo de ómica asociada a genes, como por ejemplo la trascriptómica. MAMBA fue comparado con otros métodos existentes en cuanto a la capacidad de predcción de metabolitos y fué aplicado al conjunto interno de datos multiómicos. Este conjunto de datos multiómicos fue generado dentro del proyecto PROMETEO, en el cuál está enmarcada esta tesis. MAMBA demostró capturar la biología conocida sobre nuestro diseño experimental, además de ser útil para derivar nuevas observaciones e hipótesis biológicas.

En conjunto, esta tesis presenta herramientas útiles para el campo de la biología de sistemas, y que cubren tanto el preprocesado de conjunto de datos multiómicos como su posterior análisis estadístico integrativo.

Resum

La investigació en Biologia de Sistemes s'ha expandit els darrers anys juntament amb el desenvolupament de les tecnologies òmiques. La combinació i l'anàlisi simultània de diferents tipus de dades òmiques permet l'estudi de les connexions i les relacions entre els diferents nivells d'organització celůlular, la qual cosa permet una visió a nivell de sistema de l'organisme estudiat. Aquesta tesi doctoral té com a objectiu estudiar, desenvolupar i aplicar estratègies dintegració multiòmica al camp de la biologia de sistemes.

L'encara elevat cost de les tecnologies òmiques dificulta que la majoria de laboratoris puguin abordar un estudi multiòmic complet. Això no obstant, la gran disponibilitat de dades òmiques en repositoris públics permet l'ús d'aquestes dades ja generades. Malauradament, la combinació de dades òmiques provinents de diferents orígens, dóna lloc a l'aparició d'un soroll no desitjat en les dades, cosa que es coneix com a efecte lot o *batch effect* en anglès. L'efecte lot impedeix la correcta anàlisi conjunta de les dades i, per tant, cal utilitzar els anomenats algorismes de correcció d'Efecte lot (*Batch Effect Correction Algorithms*, BECAs) per eliminar-lo. Actualment hi ha un gran nombre d'aquests algorismes que corregeixen l'efecte lot que es basen en diferents mètodes i models estadístics i que formen part del pas de preprocessament de les dades. Tot i això, els mètodes existents no estan pensats per als dissenys multiòmics ja que només permeten la correcció d'un mateix

tipus de dada òmica que ha d'haver estat mesurada en tots els lots o *batches*. Per això desenvolupem la nostra eina MultiBaC basada en la regressió PLS i models ANOVA-SCA, que pot corregir l'efecte lot en dissenys multiòmics, permetent la correcció de dades que no hagin estat mesurades a tots els lots. En aquest treball, MultiBaC ha sigut validat i avaluat en diferents conjunts de dades, a més a més, presentem MultiBaC com a paquet de R per facilitar l'ús de la nostra eina.

La majoria de mètodes d'integració multiòmica existents són mètodes multivariants basats en l'anàlisi de l'espai latent. Aquests mètodes es coneixen com a "dirigits per dades" o *"data-driven"* en anglès. Aquest tipus de mètodes es basen en la cerca de correlacions per determinar les relacions entre les diferents variables. Els mètodes dirigits per dades necessiten gran quantitat d'observacions o mostres per poder trobar correlacions robustes i/o significatives entre les variables. Lamentablement, al món de la biologia molecular, els conjunts de dades amb un gran nombre de mostres no solen ser molt habituals, degut a l'elevat cost de generació de les dades òmiques. Com a alternativa als mètodes dirigits per dades, algunes estratègies d'integració multiòmiques es basen en mètodes dirigits per models o *model-driven* en anglès. Aquests mètodes poden ajustar-se amb un nombre menor d'observacions i són molt útils per trobar relacions mecanístiques entre els diferents components celůlulars. Tot i això, els mètodes dirigits per models necessiten una informació a priori, el model, que normalment és un model

metabòlic de l'organisme estudiat. Actualment, únicament transcriptòmica i metabolòmica quantitativa, han estat els dos tipus de dada òmica que s'han integrat amb èxit usant mètodes dirigits per models. No obstant això, la metabolòmica quantitativa no està gaire estesa i la majoria de laboratoris generen metabolòmica no quantitativa o semi-quantitativa, les quals no es poden integrar amb els mètodes actuals. Per contribuir en aquesta qüestió, hem desenvolupat MAMBA, una eina d'integració multiòmica dirigida per models i basada en la metodologia d'optimització matemàtica, que és capaç d'analitzar conjuntament metabolòmica no quantitativa o semi-quantitativa amb un altre tipus d'òmica associada a gens, com per exemple la trascriptòmica. MAMBA va ser comparat amb altres mètodes existents quant a la capacitat de predcció de metabòlits i va ser aplicat al conjunt intern de dades multiòmiques. Aquest conjunt de dades multiòmiques va ser generat dins del projecte PROMETEO, en el qual està emmarcada aquesta tesi. Es demostra que MAMBA capturar la biologia coneguda sobre el nostre disseny experimental, a més de ser útil per derivar noves observacions i hipòtesis biològiques.

En conjunt, aquesta tesi presenta eines útils per al camp de la biologia de sistemes, i que cobreixen tant el preprocessament de conjunt de dades multiòmiques com la seua posterior anàlisi estadística integrativa.

Agradecimientos

A pesar de aparecer como único autor de este trabajo, cuando me pregunto a mí mismo: ¿cómo he llegado hasta aquí?, son muchas las personas que me vienen a la mente. Familia, amigos y compañeros que, no solo durante los años de la tesis, han ido acompañándome e influenciándome en mi vida hasta llegar a hoy.

Es bien sabido, por aquellos que me conocen más de cerca, que no estudié la carrera de mis sueños, pero gracias a algunos profesores (o incluso mentores) conseguí encontrar mi lugar dentro del campo de las ciencias biológicas. Carlos, con quién me inicié como investigador gracias a que me aceptara como alumno colaborador en su laboratorio, Tito, Ismael y Martín son, aunque no todos sean conscientes, los que despertaron en mí el interés por el uso de la computación para modelar, analizar y, en definitiva, estudiar organismos vivos. Fue así como decidí continuar mis estudios en Bioinformática en la Universidad de Valencia, donde conocí a otro profesor que marcó el futuro devenir de mi carrera académica/profesional, Guillermo Ayala. Con él descubrí lo emocionante de sacar la señal de los datos biológicos, y también la, no tan emocionante, diferencia entre los modelos y los datos reales.

Continué con mi habilidad de encontrar buenos mentores y, cuando decidí hacer el doctorado, conocí a los tres grandes culpables de que acabara escribiendo este trabajo y con los que quiero compartir el mérito del mismo, Sonia, Ana y Alberto. Gracias por vuestro tiempo, por haber confiado en mí a pesar de que no tenía el background ideal para el proyecto y, sobre todo, gracias por el manual de "cómo ser científico" que me habéis ido enseñando todos estos años. Sonia, gracias en especial por la paciencia y la ayuda del día a día. Nunca sabré si llegaste a estar harta de mí, porque siempre respondías con dedicación y buen humor cada vez que te pedía ayuda (no lo sabré, pero espero que no). Tampoco habría sido capaz de terminar este trabajo sin la ayuda y colaboración de nuestras inmejorables "wet-lab partners", Susana y Carme. Gracias a las dos por las discusiones en las que tanto he aprendido de biología y, en general, por toda vuestra ayuda relacionada o no con el trabajo de investigación. A Igor, con quien tuve el placer de trabajar durante mi estancia de doctorado (y más allá), quiero también dedicarle un agradecimiento especial, por su incansable ayuda en hacerme entender un mundo que era nuevo para mí, y por volcarse con el desarrollo de nuestro trabajo en común por encima de toda expectativa.

Durante este tiempo en el CIPF, IBV y UPV he tenido la gran suerte de contar con un grupo de compañeros y compañeras con los/las que he compartido grandes momentos que, sin duda, han hecho del doctorado una etapa inolvidable. Lorena, Víctor, Salva, Carlos, Fran, Teresa, Pedro, Ángeles, Cristina ... Aunque no lo diga muy a menudo, echo mucho de menos nuestras charlas durante las comidas en el CIPF, como seguro que vosotros echáis de menos mi habilidad con el Photoshop.

Como bien conocerán todos los que hayan sobrevivido a una tesis, no sólo el apoyo en el trabajo contribuye al resultado final de todo este esfuerzo. Agradecer a mis padres todo lo que han hecho por mí, mi educación y mis valores, que son los pilares de todo lo que soy hoy. También a mi hermano y mis amigos, mi "team", por apoyarme y ayudarme a desconectar y desahogarme en los momentos de agobio. A Marta, con quién parece que compartiré mi vida, gracias, no solo por tu apoyo, si no también por ser un ejemplo para mí como científica y por aguantar mis largas charlas sobre mi trabajo y ayudarme con tus consejos, sin ti esto tampoco habría sido posible. Vivimos toda esta etapa juntos y, al final, ambos lo hemos conseguido. Podemos decir que todo, separarnos de nuestra familia y amigos, los largos viajes en coche escuchando sevillanas . . . , todo ha merecido la pena!

Soy consciente de que dejo por nombrar a muchas personas, así pues, muchas gracias a todos los que me habéis acompañado este tiempo, por vuestro tiempo, vuestra ayuda y vuestro apoyo. Todos sois parte de este trabajo y a todos os estaré eternamente agradecido por ello. Es bien sabido, por aquellos que me conocen más de cerca, que no podía terminar de otra forma:
"Salud, Juancarlismo y Libertad."

Contents

Chapter 1

Introduction

1.1 The concept of Systems Biology

The application of systems theory to biology dates back to the early 1960s, when Ludwig von Bertalanffy referred to an open system as the characteristic state of living organisms [1]. Then in the 1990s, more specific terms were used to enunciate that the future of biology would depend on the analysis of systems and complex networks [2]. However, there is a certain lack of consensus about the exact moment when the concept was stated as a perfectly defined term.

Systems biology emerged from the linking of a large amount of data from different cellular layers and different molecular nature, the increase in computational performance, the improvement of data generation technologies and the collaboration between different disciplines unable to give large scale understanding about biology on their own [3]. The field of systems biology was developed through several stages. First, traditional molecular biology turned into molecular systems biology without the use of mathematical models. This fact started when the structure and function of genes were discovered and the human genome was decoded. However, on the post-genomic era the study of biological pathways became a central item for the research community. Then, the convergence with systems theory took place which allowed the origin of the systems biology field based on mathematical models [2, 4].

The vision of systems biology about biology as an integrative discipline is opposed to the molecular reductionist vision. Molecular biology considered the different cellular components as static and isolated, ignoring the interaction between them in biological processes. Without the intention of detracting the reductionist approach, many authors have stated the need of integrative approaches since biological systems have properties that cannot be evaluated analysing their parts in an exclusive manner [5, 6]. Thus, the integration of different omic data is the main moving force of systems biology as it combines experimental assays with model building approaches.

Currently, there is a huge diversity of omic data types (e.g. transcriptomics, metabolomics, proteomics, etc.) and new experimental methods that allow measuring a bunch of different biomolecules. Moreover, the powerful computational performance has increased permitting the application of more complex mathematical models to more complex biological systems [5–9].

In the last years, the interest of the scientific community on systems biology has increased since the number of conferences, publications and/or projects about this topic is wider than ever before and interdisciplinary research combining traditional molecular biology with the most advance mathematical and computational developments has made systems biology one of the most interesting research areas. However, although large-scale omic datasets are becoming more accessible, and multiomics studies are becoming much more frequent, real multiomic integration remains very challenging [10, 11]. Nonetheless, it has become increasingly evident that cellular processes are multi-layered and hence multiomic integration approaches are crucial to understand the functioning of living cells, e.g. changes in chromatin status affect gene expression profiles, stability and translation of transcripts, which in turn causes changes in the cellular metabolism.

1.2 Omic technologies

High-throughput technologies allow the large scale study and quantification of the different cell constituents, increasing the amount, quality and variety of molecular data (the so-called omic data) [12]. From genotype to phenotype, omic technologies permit to analyze biomolecules at different layers, providing information for almost all biochemical transformation/regulation steps (Figure 1.1). Some of the most widely used omics in systems biology are: epigenomics, transcriptomics, metabolomics and fluxomics.

The study of the chromatin and transcriptome is based on Next Generation Sequencing (NGS), a high-throughput genomic technology that consists on sequencing DNA and RNA [12]. Chromatin is dynamic and the specific set of modifications across the genome regulates the final synthesis of the mRNA [13]. ChIP-seq (Chromatin Immunoprecipitation Sequencing) is a methodology that detects binding sites of DNA-binding proteins, either transcription factors, histones or other proteins by sequencing bound DNA [14]. In a different way, ATAC-seq measures chromatin accessibility by sequencing open DNA not protected (bound) by any kind of DNA-binding protein [15]. These methodologies allow to build the map of the chromatin status and monitoring its evolution under certain conditions.

Transcription rates can also be profiled by NGS. GRO-seq [16] and NET-seq [17] combine the precipitation of the RNA-polymerase protein to the sequencing of the bound transcripts, giving information of the nascent RNA. These newly synthetized RNA molecules are then exported to the nucleus where they can be degraded by the mRNA decay machinery, stored in specific loci or translated into proteins [18, 19]. Throughout this journey, RNA molecules are guided by RNA-binding proteins that control their fate [20]. Methods such as PAR-CLIP [21] allow the study of mRNA bound to proteins in the cytoplasm. Moreover, steady-state levels of mRNA are profiled by RNA-seq which is the most popular NGS technology in computational biology. Finally, translation process can also be monitored via Ribo-seq that measures the mRNA in active translation (bound to ribosomes) [22].

The large scale study of proteins (proteomics) is much more intricate than genomics or transcriptomics, mostly because the total protein expression profile is highly heterogeneous in terms of physical and structural properties [23]. Proteomics relies on two basic technological cornerstones: a method to fractionate complex protein or peptide mixtures and a technology to acquire the data necessary to identify

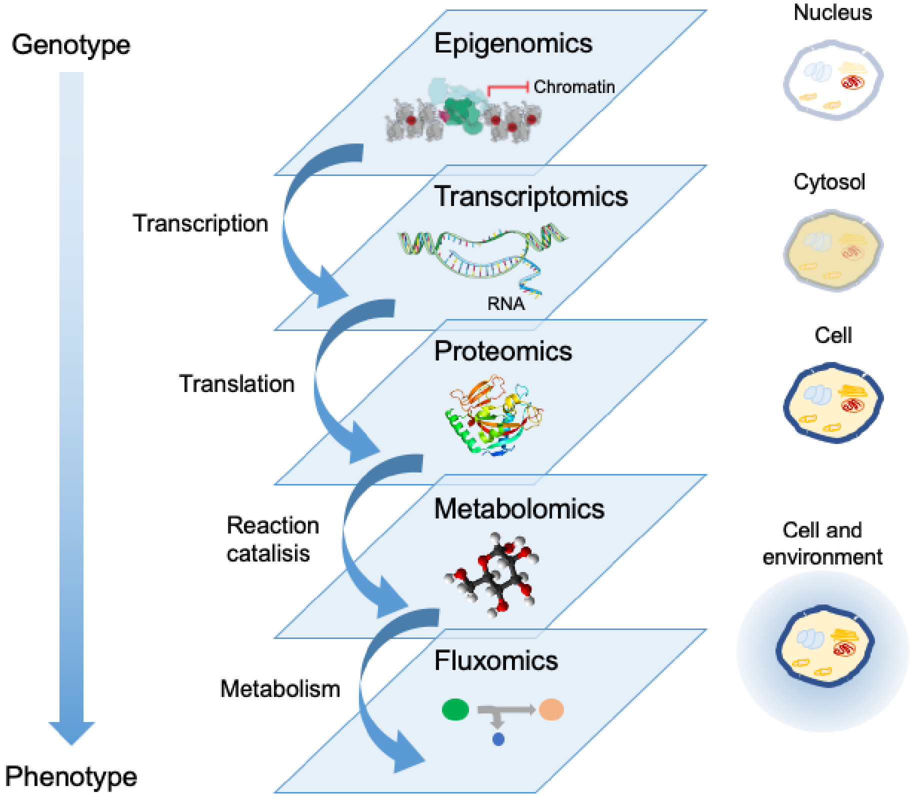


Figure 1.1: The most relevant omic data types in integrative omic studies, represented as different layers of biological information.

individual proteins. Mass spectrometry (MS) has been widely used in the identification of proteins and is usually performed in combination with a protein separation procedure. These procedures are roughly divided into gel-based [e.g. differential in-gel electrophoresis (DIGE)] or gel-free [e.g. liquid or gas chromatography (LC and GC, respectively)] [24]. MS is a technique that allows the detection of compounds by separating ions by their unique mass (mass-to-charge ratios) using a mass spectrometer [23]. Although protein identification and quantification have improved in the last years, this technique still has a low signal/noise ratio which makes proteomic data less sensitive than other omic technologies and hence is strongly biased towards abundant proteins. Targeted proteomics (determining the presence and quantity of particular proteins or peptides) improves sensitivity in contrast to untargeted studies (quantitative and qualitative study of all proteins present in a sample). In addition, missing values are also a relevant issue in proteomics studies. The occurrence of missing data results from different biological and/or technical reasons: a peptide is either not detected, below the detection limit or simply not present in the sample [24, 25].

Lastly, metabolomics provides a global characterization of the metabolic profile of a given sample, providing a closer picture of the metabolic processes. Metabolomics is intricate due to three main reasons: i) thousands of metabolites could be present at the same time in a sample and some of them are almost identical in terms of molecular composition and abundance, ii) the concentration of two metabolites present in the same sample may differ in several orders of magnitude and iii) among the biological constituents of the cell, metabolites present the largest heterogeneity in terms of chemical and physical properties [26]. Metabolomics analyses are typically performed by three principal analytical techniques: Gas Chromatography coupled to Mass Spectometry (GC-MS), Liquid Chromatography coupled with single-stage Mass Spectometry (LC-MS) and Nuclear Magnetic Resonance (NMR). NMR measures the magnetic response of the atomic nucleus of a sample to an external magnetic field. NMR has become the preferred platform for long-term or large-scale clinical metabolomic research due to its relative ease of sample preparation, capacity to measure metabolite levels, high level of experimental reproducibility, and nondestructive nature [27]. However, NMR is less sensitive than MS techniques [27]. In contrast to transcriptomics or proteomics, the availability of published metabolomic data is still scarce and limited.

All metabolic processes in living cells are performed by interactions of the different types of biomolecules presented above. These processes are constituted by a concatenation of metabolic reactions that modifies cell metabolome. Fluxes through metabolic reactions are the final result of the interplay of proper chromatin modification, gene expression, protein activity and metabolite concentrations. By analogy to other omic modalities, measuring the activity of metabolic reactions was termed “fluxomics”. This omic integrates experimental measurements of metabolic fluxes with mathematical models to determine the flux through a metabolic network. While the overall rate of nutrients consumption or production can be easily measured (e.g., consumption of Glucose or secretion of Ethanol), intracellular fluxes are more difficult to characterize [28]. The complete set of reactions that consume or produce metabolites is known as metabolism. Metabolic reactions are catalyzed by enzymes (proteins) and are quantified in terms of metabolic fluxes (units of substrate metabolized per unit of time). Metabolism allows organisms to perform their vital functions (e.g. grow and reproduce, respond to the environment, etc.). Due to its close relationship with cellular phenotype, the study of cellular metabolism has allowed the diagnosis of diseases [29], novel drug target discovery [30] and improvement of biotechnological processes [31] among others.

Metabolism has been extensively studied by gathering the metabolic reactions that serve a specific biological demand. A group of interconnected metabolic reactions that operate together to satisfy a certain biological role, is commonly known as metabolic pathway. In the literature, there are several well characterized metabolic pathways: Glycoysis, Tricarboxylic acid (TCA) cycle, lactic acid or alcoholic fermentation, etc. However, metabolic pathways do not operate in isolation and therefore, the metabolism of living organism is characterized by the interaction between different metabolic pathways.

The comprehensive study of the phenotype arises from the use of different omic technologies that allow to measure the set of biomolecules of the cell massively and address biological questions that are otherwise unattainable using conventional methods [32].

1.3 Nature and structure of omic data

In the previous section, several omic technologies have been introduced. For the scope of this work, only NGS and NMR omic data are being discussed. The nature and the type of data generated are different between omic technologies and therefore the processing and analysis of different omic data vary. Nevertheless, data from different omic technologies have a similar structure. Particularly, a huge number of features are observed on (usually) a small number of samples ($Number\ of\ features >> Number\ of\ samples$). Each feature measures the abundance of biomolecules.

Omic data are commonly represented as matrices: $\mathbf{X}$, where x_{ij} indicates the quantification of feature j in sample i. For most omic data types and prior to any normalization/transformation step, the value of x_{ij} is usually positive and a higher value represents more abundance of a given feature.

NGS omic technologies differ on how biomolecules are captured/extracted from organisms or cells. However, the pipelines used to obtain NGS data are rather similar to each other. Extracted biomolecules (RNA or DNA) are sheared and the fragments are sequenced by a high-throughput platform. The sequencing process generates millions of short reads with associated quality scores. Those reads are mapped onto a reference genome to identify the genomic location of each read. Reference genomes are usually annotated, i.e., gene coordinates are known. Thus, in the case of gene-centric omics as RNA-seq, the mapped reads corresponding to each gene can be quantified and the number of reads mapped to a given gene is an estimation of the expression level of that gene. Regarding region-based omics as ChIP-seq or ATAC-seq, a process called peak-calling is performed. Roughly, genomic regions with a high concentration of mapped genes are declared as "peaks" and reads mapping those regions are pooled to quantify the signal of the peaks. Peaks' signals can be assigned to genes (if needed). Regardless the omic, the resulting NGS data consist of discrete values (the number of reads mapping a certain region), the so-called counts. These data are not directly comparable and have to be corrected/normalized to remove technical biases and allow between-sample comparisons. Initial attempts to model NGS omic data assumed the Poisson distribution [33]. However, due to the variability between biological replicates, currently the negative binomial distribution is generally accepted for modeling NGS data as it incorporates over-dispersion in the definition of the variance [34, 35].

Contrary to NGS, NMR generates continuous values in the form of spectra. Researchers can record NMR spectra for multiple different nuclei (^{1}H, ^{13}C, ^{15}N, and ^{31}P) either separately or simultaneously to study different metabolite classes [27]. Once metabolites are extracted from cells, a suitable solvent suppression before performing the NMR assay is required [27, 36]. Figure 1.2 shows an example of an NMR spectra. Letters represent different compounds that are separated along the x-axis according to their chemical composition regarding the target nuclei. Y-axis indicates the abundance of the target nuclei in each compound. Therefore, compound concentrations are not directly reflected in NMR spectra.

The next step in the NMR data processing pipeline is the identification of the compounds in the spectra. Historically, NMR databases of most chemical compounds were compiled and kept in books. Thanks to the efforts of a number of metabolomics labs from around the world, there are now several high-quality, web-based NMR spectral databases containing reference NMR spectra for hundreds of metabolites. Particularly, The Human Metabolome Database (HMDB www.hmdb.ca) [37] is becoming the standard reference for most metabolomic studies. Moreover, alternative software tools exist that can both identify compounds and estimate metabolite concentrations from NMR spectra [38, 39]. To quantify the compound, the areas of the peaks are used and resulting quantification values are modeled using a Normal distribution [36, 40]. Similarly to NGS data, adequate normalization is often required to account for technical variability among samples.

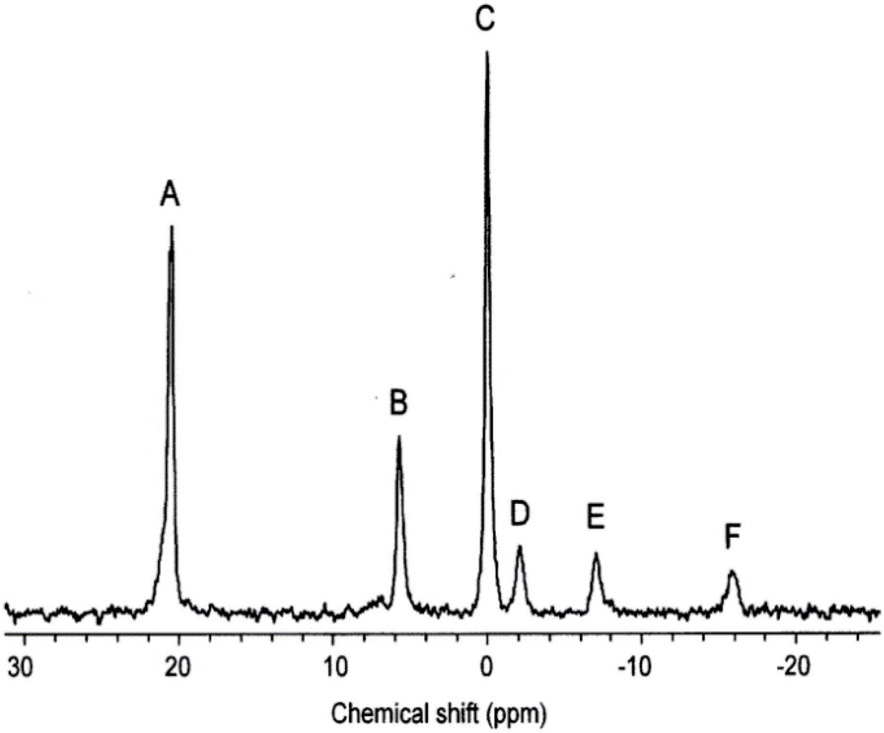


Figure 1.2: Example of NMR spectra for metabolomics. Source: mdpi.com

1.4 Statistics for systems biology

In the last decades, the use of high-throughput technologies has transformed molecular biology into a data-rich discipline. The study of biological systems at single-layer level, where all molecular components (e.g. RNA, proteins, etc.) were studied separately, has been substituted for the combination of multiple holistic approaches that allow to understand how biological entities establish highly interconnected networks where biological functions cannot be characterized by individual actors [41]. In this context, Systems Biology has evolved into an integrative discipline for the study of complex interactions between biological systems components [42]. However, extraction of knowledge from this wealth of omic data is not trivial [10, 11] and a wide range of statistical methods has been developed or adapted to cope with this challenging task. These methodologies comprise algorithms that range from data pre-processing and normalization strategies to multiomic integration approaches.

1.4.1 Single-omic data analysis

Available statistical methods for the analysis of omic data are mostly developed for finding statistically significant differences between two groups of biological interest (e.g. treated and control) [43]. Since high throughput transcriptomics is one of the most widely used technique to profile biological samples, a vast number of approaches exist for the analysis of gene expression data. The most relevant and widely tested data-driven methods used in computational biology are introduced below and classified regarding the task or biological question they address.

1.4.1.1 Omic data pre-processing and normalization

Once the quantification of omic features is obtained, between- and within-sample normalization methods are applied to remove or mitigate technical biases and make samples and features comparable [44, 45]. In particular, most of the normalization strategies intend to eliminate systematic differences among samples. Samples with higher sequencing depth or total NMR signal (i.e. sum of the values for all the features) complicate feature-wise analyses since they introduce artificial high values compared to the rest of samples. In NMR omic data, this issue is usually corrected by dividing the area of each peak by the sum of total spectrum intensity [36]. Similarly, samples from NGS data are corrected by their library size or sequencing depth, i.e., total number of counts. In addition to sequencing depth, NGS data are affected by other specific biases:

- **RNA/DNA composition.** RNA-seq or ChIP-seq techniques measure the relative abundance of each gene (or region) in a given biological sample. To illustrate RNA/DNA composition problem, let us consider that only a small fraction of genes is highly expressed in only one biological sample (or group of samples) compared to other biological samples. This fraction of genes will capture an important part of the reads sequenced and hence the rest of genes (equally expressed among samples) will receive a low number of reads resulting into low expression values. Therefore, the expression of those genes where a low amount of reads are assigned will be underestimated leading to wrong between-sample comparisons. The most popular method for correcting RNA composition bias is the Trimmed Mean of M-values (TMM, detailed in Chapter 4) [46].

- **GC content.** GC content [nucleotides: Guanine (G) and Cytosine (C)] of genes does not change between samples and therefore should not affect between-sample differences. However, GC content affects the sequencing reaction and, as a result gene expression depends on the percentage of gene GC content [47, 48]. In Risso et al., 2011 [48] the authors propose the Full-Quantile Normalization for correcting GC content bias. In full-quantile (FQ) normalization, genes are stratified according to GC content. The quantiles of the read count distributions are then matched between GC bins (genes are previously stratified into equally-sized bins based on GC content), by sorting counts within bins and then taking the median of quantiles across bins.
- **Gene length.** Longer genes generate more fragments during the library preparation and therefore they produce more sequencing reads, which, in turn, results into more counts compared to short

transcripts of similar expression. This is a within-sample bias and some methods have been proposed to correct it. Reads per Kilobase Million (RPKM) [49] is an extended method for correcting gene-length bias. The normalized counts for gene i and sample s (y_{is}) are defined as: $y_{is} = 10^9 x_{is}/(N_s l_i)$ where x_{is} is the original count value, N_s is the total number of counts in sample s, and l_i is the length of gene i. However, other authors question the usefulness of gene-length bias correction. Oshlack and Wakefield, 2009 discussed this issue in [50]. Under some basic assumptions, they demonstrate that statistical power for comparing two biological groups depends on gene-length and hence longer genes are more likely to be declared significantly expressed. However, this issue is usually not fully solved after gene-length bias correction.

These technical biases are not always present in NGS omic data at the same time. Therefore, an adequate quality control step is critical for assessing the existing biases and applying the correct normalization methods [51].

1.4.1.2 Batch effect correction

Batch effects appear when omic data are not generated under the exact same conditions (e.g. time point, laboratory, reagents, etc.). Both economic and time costs of omic data generation may motivate the combination of omic data from different sources, i.e. different batches (e.g.: leverage omic data from public repositories). Batches are often an important source of noise in data that confound the biological signal and impairs statistical analyses. Therefore, batch effects have to be removed.

Batch effect correction has been addressed by many authors and consequently different approaches have been proposed. The removal of batch effects is possible as long as no other covariate (biological condition) is confounded with batch effect. ComBat [52] is the most popular method, which uses either parametric or non-parametric empirical Bayes frameworks for estimating the batch effect. Other examples are Limma package [53] that estimates the batch effect using linear models, and ARSyN [54], which implements ANOVA-SCA (ASCA) decomposition [55]. These methods differ on the way they estimate the batch effect, but the final goal for all of them is to extract the estimated batch effect from the data (methods detailed in Chapter 3).

In the multiomic scenario, these methods have to be applied in an omic-wise manner (intra-omic batch effect correction), but when the different omic data types come from different sources, the currently available batch-effect correction methods are not prepared to correct the inter-omic batch effect.

1.4.1.3 Identification of differentially expressed/quantified features

Identifying differentially expressed genes (or biological features in general) is a pivotal task in computational biology and there are a lot of contributions to address this problem. A classification of differential expression methods can roughly be done considering the underlying model imposed to the data.

Most of the methods proposed for NGS omic data are parametric and work directly on the count data using a negative binomial distribution. Among this group, the most popular methods are edgeR [35] and DESeq2 [56]. A critical part of the inference procedure is to obtain a reliable estimate of the dispersion parameter for each gene which is limited by the sample size. Both DESeq and edgeR share information across genes to estimate a gene-wise dispersion parameter. In fact, the way these methods implement information sharing accounts for the main difference between them. edgeR uses a two-step estimation. First, a common dispersion parameter for all the genes is computed using a conditional maximum likelihood approach. Then, gene-wise dispersion is estimated, but the individual estimates are squeezed towards the common one using a weighted likelihood approach [57]. On the other hand, DESeq2 obtain the dispersion estimates by modeling the observed mean-dispersion relationship for the genes using local regression. Both methods test for significant differential expression using either a variant of an exact test (two-group comparisons) or a generalized linear model (GLMs, for more complex designs).

Regarding non-parametric methods, NOISeq [51] explores the distribution of absolute expression differences and fold-changes between two biological conditions. Then, this distribution is compared to a null-distribution generated by permutation techniques to compute the probability of differential expression for each gene.

Finally, other methods assume omic data follow a normal distribution. This is true for NMR data but NGS omic data need a previous transformation to meet normality assumptions [58]. Among methods that assume normal distribution for omic data, Limma R package (Linear models for microarrays) [53] is the most popular one (detailed in Chapter 4).

Other methods can model either raw counts or normalized data via GLMs. That is the case of maSigPro R package [59] which is specially deigned for time-series data (detailed in Chapter 4).

So far, there is no general consensus regarding which method performs best and new methods are continuously being presented [60]. Moreover, all these methods incorporate multiple testing correction, since as many tests are performed as the number of existing omic features.

1.4.1.4 Identifying differentially activated pathways by functional enrichment analyses

Gene expression analysis methodologies are often focused on identifying particular genes that differ significantly between two states of interest. These approaches, while useful, are unable to detect biological processes such as metabolic pathways, transcriptional programs, and stress responses, which are spread across a network of genes and inconspicuous at the level of individual genes. Functional enrichment analysis (FEA) aims to find those pathways or gene sets that are enriched in those features (genes, metabolites or proteins) changing between conditions. Therefore, FEA is often performed after a differential expression/quantification analysis. Pathways or gene sets are defined a priori and there are powerful databases to obtain the list of pathways or gene sets for a given organism (KEGG www.kegg.jp, REACTOME www.reactome.org, Gene Ontology www.geneontology.org or MSigDB www.gsea-msigdb.org). FEA can be performed by three main approaches:

- **Over-representation analysis (ORA).** ORA relies on independence tests such as Fisher's exact test [61]. ORA determines whether features from pre-defined sets (pathways or gene sets) are present more than would be expected (over-represented) in a subset of your data (differentially expressed features). Therefore, this question can be expressed as a contingency table to be analyzed with an independence test:

	Features not of interest	Features of interest
Annotated	M-k	k
Not annotated	N-M - (n-k)	n-k

where N is the total number of features (universe), M is the number of features within the universe that are annotated to the gene set, n is the size of the list of features of interest (differentially expressed), and k is the number of genes within that list which are annotated to the gene set.

- **Gene-set Enrichment Analysis (GSEA).** GSEA determines whether a set of features (pathways, gene sets) shows statistically significant, concordant differences between two biological conditions. GSEA utilizes the output from a differential expression analysis (DEA), i.e., an effect size measure (typically log2FC) and the associated p-values. The output metric is called Enrichment Score (ES) that reflects the degree to which a given gene set S is overrepresented at the extremes (top or bottom) of the entire ranked list of genes L (based on DEA output). Although there are different approaches to perform a GSEA analysis, here we introduce the one proposed by Subramanian et al., 2005 [62] as this is the method that will be used in this work. In this approach, ES is obtained by walking down the list L evaluating the fraction of genes in S weighted by their effect size measures $(r(g_j))$ and the fraction of genes not in S present up to a given position i in L:

$$\begin{aligned} Phit(S,i) &= \sum_{g_j \in S}^{j \leq i} \frac{|r(g_j)|^p}{N_R}, \text{where } N_R = \sum_{g_j \in S} |r(g_j)|^p \\ Pmiss(S,i) &= \sum_{g_j \notin S}^{j \leq i} \frac{1}{(N - N_H)}, \text{where } N_H = \text{genes in } S \end{aligned} \tag{1.1}$$

The enrichment score is the maximum deviation from zero of $Phit - Pmiss$. When $p = 0$, ES reduces to the standard Kolmogorov-Smirnov statistic; otherwise, features are weighted by their effect size values and it is the common way of using GSEA $(p = 1)$. In addition, a statistical test for significance is performed by condition-based permutation test procedure and a p-value is also returned after multiple testing correction [62].

- **Gene-Set Variation Analysis (GSVA).** GSVA can be used for functional enrichment analysis. In this case, GSVA is performed before differential feature analysis. GSVA computes a score per gene set that somehow summarizes the expression of all the genes contained. Then the association between GSVA scores and biological conditions under study is evaluated, for instance using limma or classic linear regression. According to the authors, GSVA provides increased power to detect subtle pathway activity changes over a sample population in comparison to GSEA. However, GSVA has been designed and tested only for gene expression data [63].

These methods are useful to extract deeper biological insights from omic data beyond differential feature analysis. Yet, mechanistic interpretation of the results and the connection among resulting significant gene sets are not easy tasks.

1.4.2 Multiomic integration

Separate analysis of omic data modalities allows answering targeted biological questions regarding a wide variety of biological processes, e.g. the expression of genes (transcriptomics), abundance of proteins (proteomics), or dynamics of metabolites (metabolomics and fluxomics), independently. Multiomic integration aims to combine, model, and interpret data sets that contain several of these data types. Cavill and collaborators [64] elaborated an extended description of multiomic integration strategies. They described three levels of data integration: conceptual integration, statistical integration and model-based integration. Conceptual integration means that "data sets are analyzed separately and the conclusions are compared and integrated". Thus, single-omic methodologies explained above are used in conceptual multiomic integration. Statistical integration combines data sets and analyzes them jointly, "reaching conclusions supported by all data and potentially finding signals that are not observable with the conceptual approach" [65]. Model-based integration indicates the joint analysis of the data in combination of training of a model, "which itself incorporates prior beliefs of the system" [66]. Model-based integration was later reclassified to distinguish qualitative reconstruction of biological pathways or systematic regulatory pathways from quantitative, mathematical evaluation [67].

In this document, multiomic integration approaches are presented according to the definition proposed by Jamil et al., 2020 [68]. The authors re-defined the multiomic integration workflow into three levels with increasing complexity. Level 1 is called element-based integration with two main subclasses: correlation and multivariate analyses. Level 2 is the knowledge-based pathway integration. Finally, level 3 is the model-driven integration or genome-scale analysis.

1.4.2.1 Level 1: Element-based integration

Correlation analysis

The first level of multiomic integration is an element-based approach using correlation of features. The basic approach is correlative association between two omic data sets. Correlation analysis is focused on the study of a broad class of statistical relationships involving dependence between two random variables or two sets of data. The most familiar measure of dependence between two quantities is the Pearson product-moment correlation coefficient, or simply "correlation coefficient" [69]. Multiple linear regression models have been also applied for multiomic integration to model the coordination of different omics to regulate the response of another molecule [70].

Multiomic integration strategies based on correlation are useful because they can indicate a predictive relationship that can be exploited in practice. However, two potential issues may limit this approach. Firstly, multicollinearity is a common problem when considering several explanatory features as predictors in the same model. Secondly, sample size in multiomic studies is generally small. Both together make it enormously difficult to perform a robust statistical inference and find significant regulators. To solve, or at least mitigate these issues, variable selection strategies such as ElasticNet [71] or Lasso regularization [72] (or Group Lasso [73]) methods have been used in the field, but these approaches do not consider the biological meaning of omic features and hence interesting findings may be missed.

Multivariate statistics

In the biological research area, traditional statistical techniques (e.g, linear regression, bi-variate correlations, analysis of the variance or Fisher's exact test) have been applied to solve many kind of problems. However, these techniques became insufficient to exploit the data-rich environment found in modern biology when omic data emerged, since these new technologies allow registering a wide range of variables. Moreover, the relative high cost of omic assays lead to small sample-sized experiments which is an important issue for traditional statistics. Thus, in the 70s and 80s new methods were developed to deal with such high dimensional datasets coming from chemistry and process industry, which are the basis of what it is known today as multivariate statistics.

Multivariate methods (MVA methods) applied in computational biology can be divided into two main groups: data exploration and feature-association. The first group aims to understand high dimensional datasets by extracting the most relevant signals from data via dimension reduction and selection of the most relevant features in data. The second group seeks to relate different types of data being the most common situation an explanatory omic type and another response omic information or an outcome variable (numerical or categorical). These methods are used mainly for classification among classes, discrimination and prediction.

Multiomic integration methods that have been developed in this context enforce to solve three important issues: i) Dimensionality reduction. The analysis has to identify true signals from a noisy background. ii) The relationships within and between datasets (omics). iii) Easy interpretation and visualization of the results.

Principal Component Analysis (PCA): A Principal Component Analysis (PCA) [74] finds a variable subspace that explains most of the variability of data. The original variables are transformed into a lower number of non-correlated latent variables, the so-called principal components (PCs). A PCA model summarizes the variability structure of high dimensional data and is very helpful to cluster the samples according to their observed features. A PCA model has the following expression:

$$\mathbf{X} = \mathbf{TP^t} + \mathbf{E} \tag{1.2}$$

where $\mathbf{X}$ is a $N \times M$ data matrix, $\mathbf{T}$ is the $N \times L$ matrix of scores which are the projection of the observations over the new subspace. $\mathbf{P}$ is the $M \times L$ loading matrix containing the linear combination of the variables represented in each PC.N and M are the number of observations and features, respectively, and L is the number of latent variables (PCs) of the model. Figure 1.3 shows an example to illustrate the PCA model of a 2-dimensional matrix turned into a PCA model with one latent variable.

Partial Least Squares regression (PLS): Partial Least Squares regression (PLS) [75] is a projection method used to model the relationship between a multivariate $\mathbf{X}$ $(N \times M)$ predictor or explanatory matrix and a set of response variables $\mathbf{Y}$ $(N \times K)$. Thus, N is the number of observations or samples, M represent the number of explanatory variables, and K is the number of response variables. The aim of a PLS model is to find a variable subspace (a set of L latent variables) for both matrices that maximize the covariance between them. The PLS model can be expressed as $\mathbf{Y} = \mathbf{TC^t} + E = \mathbf{XB} + \mathbf{E}$, where $\mathbf{B}$ is the regression coefficient matrix and $\mathbf{E}$ is the residuals matrix. $\mathbf{B}$ can be estimated as:

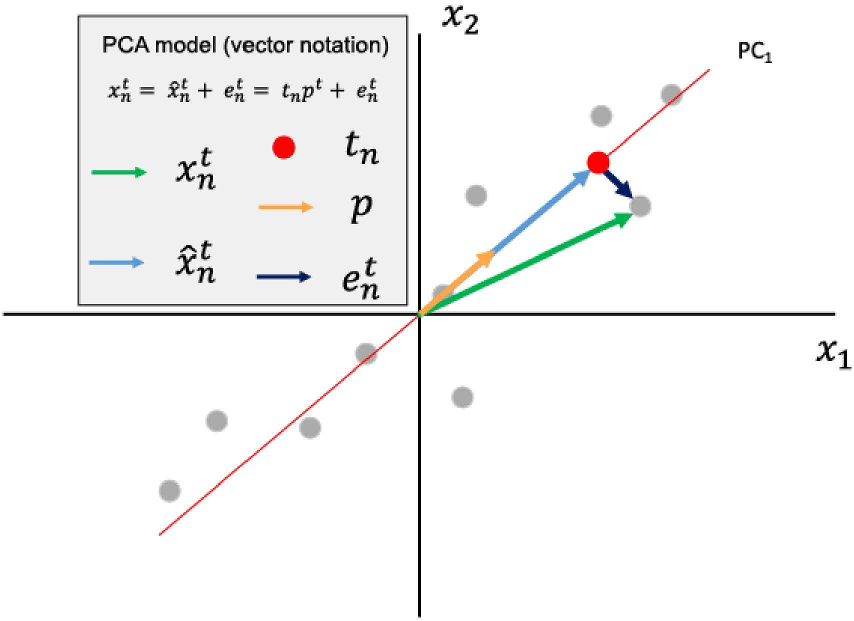

PCA model (vector notation)
$x_n^t = \hat{x}_n^t + e_n^t = t_n p^t + e_n^t$
x_n^t
t_n
p
$\hat{x}_n^t$
e_n^t
x_2
x_1
PC_1

Figure 1.3: PCA model example of a 2-dimensional observation. Vectors are represented by arrows and scalars as dots.

$$\mathbf{B} = \mathbf{W}^{*}\mathbf{C}^{\mathbf{t}} = \mathbf{W}(\mathbf{P}^{\mathbf{t}}\mathbf{W})^{-\mathbf{1}}\mathbf{C}^{\mathbf{t}} \tag{1.3}$$

where $\mathbf{T}$ is the $\mathbf{X}$-scores matrix, $\mathbf{W}$ is the $\mathbf{X}$-weight matrix, $\mathbf{P}$ is the $\mathbf{X}$-loading matrix and $\mathbf{C}$ is the $\mathbf{Y}$-weight matrix. While the loadings, define the subspace that best reconstruct the $\mathbf{X}$ matrix given the scores $\mathbf{T}$, the weightings define the subspace that maximize the covariance between $\mathbf{X}$ and $\mathbf{Y}$. A PLS model can be used as a predictor model by its equation, $\hat{\mathbf{Y}} = \mathbf{TC}^{\mathbf{t}} = \mathbf{XB}$. The new set of observations must be projected into the model, $\mathbf{T} = \mathbf{XW}^{*}$, then the PLS coefficients can be computed using Equation 1.3 and the prediction is performed.

MVA methods in computational biology: A vast universe

Due to the characteristics of omic datasets (n^{o} of variables $>> n^{o}$ of observations) classical statistical methods are hardly applicable, even impossible in most cases. To solve this, most of the studies in multi-omic integration use MVA methods based on dimension reduction to summarize the variability of datasets. There is a wide diversity of these methods covering almost all experimental designs and analysis possibilities. Multivariate methods are usually classified into two groups: unsupervised and supervised methods. PCA (unsupervised) and PLS (supervised) are the most used MVA methods in computational biology and are also the basis for many other methods that have been developed over the last years to cover more data structures. In the context of multiomic analysis, multiblock PCA and PLS (MB-PCA and MB-PLS) allow the use of multiple explanatory and/or response matrices by weighting the different modalities and creating super latent spaces [76, 77]. Also applied to PCA and PLS, variable selection has a great interest in the field. Regularized linear regression principles (Lasso penalization, Ridge regression and Elastic net) have been adapted to PCA and PLS to obtain the key features in explaining data variability [78, 79]. These method have been called as sparse versions, sparse PLS (S-PLS) for instance. In the case of supervised analysis, all of these approaches include the discriminant analysis version (PLS-DA, MB-PLS-DA, etc.) for modeling categorical outcomes (classification) [80, 81]. Moreover, O2-PLS and other approaches were conceived for separating common and distinctive variability among different matrices (omics) [82–84]. In addition to genes and samples, sometimes a third dimension is added to an experimental setting. Specially, time series data is the most common situation where, to represent data, we need to use three-dimension arrays (samples x genes x time) instead of matrices. The above methods can be applied in these cases by unfolding the three-way array into a 2-way matrix (samples x ((genes x time)), however there are specific approaches for N-dimensional structures. PARAFAC [85] and Tucker3 [86] are the N-way version of PCA, and N-PLS adapts PLS regression for N-way data. These approaches have been successfully applied to biological data [87].

In conclusion, multivariate statistics has been extensively and successfully applied in the multiomic integration field. The high dimensionality of omic data makes these methods powerful for extracting meaningful signals from biological data. However, they lack on mechanistic interpretation of data as they are purely data-driven approaches and no a priori information that relates different features is used. Nevertheless, there is also a dimension reduction method that uses prior information which is the PLS Path Modelling (PLS-PM) where path models (association among features) need to be defined a priori [88].

1.4.2.2 Level 2: Pathway-based integration

Pathway-based integration is aimed to map omics data sets, either transcriptome, proteome or metabolome to existing metabolic pathway databases, e.g. Kyoto Encyclopedia of Genes and Genomes (KEGG; https://www.genome.jp/kegg/). There are many software tools available for multiomic integration at pathway level [89]. These methods often require a prior feature differential analysis and translate omic-wise results into a pathway-based joint-omic output. Paintomics web resource [90] is an example of these approaches and allows multiomic functional enrichment analysis combining the p-values from omic-wise ORAs (detailed in Chapter 4). Pathway-based integration was beyond the scope of this thesis work and hence no further examples or details are provided. For a comprehensive review of these approaches we recommend the following references: Ili Nadhirah Jamil et al., 2020 [68], Rafael Hernández-de-Diego et al., 2018 [90] and Hernández-de-Diego et al., 2017 [91].

1.4.2.3 Level 3: Model driven approaches

Data-driven methods are extremely efficient to extract knowledge from a massive amount of data. However, they require a considerable amount of experimental data to determine patterns, correlations and mechanisms. In some cases, it is necessary to add new knowledge extracted from the literature in order to constraint the space of solutions to be analyzed. Typically, this information is introduced in the form of models, which are abstractions of a biological system representing the interactions and inter-dependencies among the components of the system (metabolites, genes, proteins, etc.) [92].

These models can be represented as graphs. However these representations are not accurate, often fail to capture the dynamic behavior of biological systems, and are ineffective when dealing with vast networks. Computational models, on the other hand, give a precise mathematical representation of information that may be used to understand and assess observed data, analyze system behavior (e.g., identify critical pieces for a specific behavior), and generate and test hypotheses [92]. As a result, mathematical modeling has become an indispensable tool for fully understand cell metabolism and its interactions with the environment conditions [92]. Next, some of the most relevant model-driven approaches to study the metabolism are briefly summarized:

Flux Balance Analysis (FBA)

Flux Balance Analysis (FBA) [93] is one of the most used constraint-based modeling (CBM) methods in systems biology. In CBM, the fluxes through metabolic reactions conform the decision variables. The outcome of each decision in CBMs is constrained by a minimum and maximum range of limits. Thus, CBMs calculate flux distributions that satisfy three fundamental types of constraints [94] : i) steady-state mass-balance constraint, which sets the total production and consumption rates for each metabolite to be equal; ii) thermodynamics (reaction reversibility), i.e., non feasible metabolic transformations are not allowed; iii) capacity constraints, i.e., upper and lower bounds for fluxes can be imposed. CBMs are most commonly used with optimization techniques, such as the use of linear and mixed-integer programming to maximize an objective function (OF) and find a space of feasible flux solutions that is consistent with the stoichiometric, thermodynamic and capacity constraints imposed by a given metabolic model. The OF ($f(\mathbf{x})$) is the formal way to parameterize the biological objective, the phenotype, and defines how much each reaction contributes to the phenotype of interest. This objective function usually reflects the growth rate which is defined by an artificial biomass production reaction, although it can be defined differently as needed [92]. FBA is mathematically formulated as follows:

$$\begin{aligned} &\text{max} f(\mathbf{x}) && (1.4)\\ \text{subject to} & \\ &\mathbf{Sx} = 0 && (a)\\ &\mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} && (b) \end{aligned}$$

where $\mathbf{x}$ is the solution or flux vector, $\mathbf{S}$ is the stochiometric matrix that contains both stochiometric and thermodynamic constraints, and $\mathbf{lb}$ and $\mathbf{ub}$ are vectors that contains the lower and upper bound, respectively, which represent the capacity constraints.

Constraint-based methods and Genome-scale Metabolic Models

The basic FBA can be enriched with transcriptomic data to determine an optimal metabolic flux solution that fits with the given phenotype which in this case is given by gene expression profile. The integration of transcriptomic data into CBMs is based on the inference of reaction activity by using the expression of the associated genes. Overall, existing integration methods seek to build a metabolic network that satisfies the thermodynamic and stoichiometric constraints, while maximizing the concordance between reaction activities and the expression of their associated genes. However, the way in which gene expression data is translated from genes to reactions is different between the available integration methods.

- Gimme [66, 95] uses transcriptomic data to penalize the activity of reactions catalyzed by low expressed genes. Low expression is determined by an user-defined threshold. Setting an optimal gene expression threshold is critical and testing with several values is recommended. Gimme algorithm classifies reactions as active or inactive regarding the expression of their associated genes whether it is over or below the threshold, respectively. Penalty coefficients are calculated for each gene on the basis of log transformed expression:

  $\tau_g = E_{max} - E_g$

  Here, τ_g indicates the gene-associated penalty for gene g. E_g indicates the log-expression value of gene g and E_{max} the maximum log-expression for all genes in the metabolic network. These penalties are then mapped to model reactions to obtain a vector of reaction penalty coefficients, ϕ. The objective function is then defined as $min(\phi x)$.

- iMat method [96] establishes a discrete state for metabolic reactions (low, intermediate or high) based on the expression of the associated genes. Then, it maximizes the number of active (high) reactions associated to highly expressed genes and minimizes the number of active reactions associated to lowly expressed genes. Since iMat models a single condition at a time, it is necessary to define an upper and a lower expression threshold to determine whether a gene is over- or under-expressed.

- The method proposed in Gonçalves et al. 2012 [97] was developed for comparing two conditions (case vs. control) and uses relative gene expression values. Gene expression levels from one experimental condition relative to a control are used to re-define the upper and lower bounds of metabolic reactions. That is modifying the capacity constraints (lower and upper bounds) of metabolic reactions. Let us consider p as the relative value for a given reaction flux (based on gene expression), if $p > 1$, the reaction is over-expressed constraining its flux to be larger than the reference value (reaction flux in control condition) multiplied by p: $p \cdot x_c < x_i \leq ub$, where indexes c and i indicates reaction flux in control and case conditions, respectively. Similarly, if $p < 1$, the reaction is under-expressed and its flux is constrained to be lower than the reference multiplied by p ($lb \leq x_i < p \cdot x_c$).

- MADE algorithm [98] also uses relative gene expression values. In this case, metabolic reactions are classified into up-, down-regulated or constant in one condition relative to a control according to the relative expression of the associated genes. In other words, reaction activities are discretized between conditions. For instance, if a certain reaction r is associated to gene g which is significantly down-regulated in the condition compared to the control, the gene state of gene g is active in control (1) and inactive in the condition (0) and in turn the reaction r has zero flux in the condition and a positive flux in control. These discrete levels are set by using the log fold change of the associated genes and the corresponding p-value (p). MADE algorithm finds a solution that is consistent with maximum number of relative discrete levels. The optimization process is mathematically defined as:

$$\operatorname*{argmin}_{x \in X} \sum_{1=1}^{n-1} w(p_{i \to i+1}) |d_{i \to i+1} - x_{i \to i+1}| \tag{1.5}$$

where n is the number of conditions, $i \to i+1$ represents a transition from condition n_i to n_{i+1}, **x** is the solution vector containing the binary gene states, p contains the associated p-values for each transition, $w()$ is the weighting function (usually defined as -log10(p)) and **d** is the vector of observed gene state differences being: 1 for over-expressed genes, -1 for under-expressed genes and 0 for genes that do not change at a given $i \to i+1$ transition. Thus, the expression above aims to minimize the differences between the observed gene expression changes and the ones predicted by the model.

Metabolomic data have also been integrated into CBM [66, 99, 100]. Metabolomics are typically integrated into GEM reconstruction analyses as a set of capacity constraints that limit the flux through a given reaction/s. To this aim, metabolomic data from experiments using isotopically labeled subtrates or absolute quantification from label-free experiments can be used [28]. Nevertheless, the second version of

Gimme algorithm [66], allows the use of non-quantitative metabolomics data only for ensure that the detected species are used in the calculated network operating states as it creates condition-specific models. Therefore, the inclusion of non-quantitave metabolomics combining different conditions simultaneously into CBM still remains challenging. Moreover, only transcriptomics data as a gene-centric information have been widely utilized to built constraints. However, other gene-centric omic data, such as histone modifications or chromatin accessibility, have been found to be linked to cellular metabolism [101–106]. Consequently, the adaptation of current model-driven methodology framework to these omics data and specially their integrative analysis, remain uncovered by existing CBM approaches.

Chapter 2

Motivation, Objectives and Contributions

2.1 Motivation

The work in this thesis is framed within a research project fully funded by Generalitat Valenciana via PROMETEO plan, from 2017 to 2020. The project was entitled *The new system biology: development of statistical methods for multiomic system biology*, and it was coordinated between different research groups in Valencia (Spain): Genomic of Gene Expression Lab from Príncipe Felipe Research Center (CIPF), Gene Expression and RNA metabolism Lab from Biomedical Institute of Valencia (IBV) and Multivariate Statistical Engineering Group (GIEM) from the Technical University of Valencia (UPV). This research project combined both multiomic data generation and multiomic data analysis.

In the Introduction section we have reviewed the different omic technologies and the currently available methods and approaches to combine them in multiomic analyses. Particularly, data-driven correlation-based methods have been vastly used to address multiomic integration thus far. Dimension-reduction-based approaches have been particularly exploited for such purpose [107]. These methods, such as Partial Least Squares regression (PLS) [75] and derivatives, are really useful to extract global patterns from data and to find bulk relationships between features among the different omic data types considered [65]. Many methods have been formulated to address different biological questions, including, but not limited to, outcome prediction [65, 108], variable selection [108, 109] and regulatory network inference [110]. However, data-driven correlation-based methodologies have insurmountable caveats. Since correlation does not always mean causation, sometimes these multivariate approaches can lead us to the wrong interpretation of the system. These approaches also typically need a huge amount of data to successfully provide both meaningful and significant results. Those are the reasons why model-driven constraint-based approaches have also been explored for multiomic integrative analysis. In these approaches, prior biological knowledge is required, which is mostly the metabolic network model of the studied organism. Existing methodologies in this context are extensions/adaptations of the Flux Balance Analysis (FBA) [95, 96, 98]. However, only gene expression data have been successfully integrated for the dynamic modeling of a system and hence there is a window of opportunity to develop novel strategies that integrates other omic data types. Therefore, there is space for formulating novel approaches that extent the use of different types of gene-centric omic data beyond transcriptomics. Moreover, the simultaneous modeling of different conditions that characterize the metabolic network at different states, remains challenging.

Regarding data pre-processing steps, there has not been much effort in the development of suitable multiomic data harmonization methods. Particularly, batch effect is known to affect omic data. This is due to the fact that obtaining data for all the cellular layers requires several experimental technologies that are barely easily and simultaneously accessible in many research groups as omic data generation is costly.

As a matter of fact, research groups very often use data from public repositories, e.g. GEO, in order to combine those public datasets with their own information. Unfortunately, when combining different sources, data will almost unavoidably be affected by an unwanted noise effect. This unwanted source of variation is commonly known as batch effect. While single omic-wise batch effect correction is really straightforward, in multiomic analysis designs, batch effect correction has not been properly addressed thus far. Different omics can be corrected separately but only if they have been measured across all the batches. In the multiomic scenario, each omic modality may be measured by a different lab or at a different moment in time, and so it is obtained within a different batch. When this is the case, the batch effect will be confounded with the omic type effect and impossible to remove from the data.

In order to surmount these challenges concerning multiomic integration, there is a need of tools for removing batch effects on a multiomic scenario and for integrating mutiomic data to track biological signals across the different molecular layers to gain an in-depth comprehensive understanding of the cellular system. The development and application of such mathematical, statistical and computational methods are the basis of the objectives for this thesis.

2.2 Objectives of this thesis

The general aim of the PROMETEO project, in which this thesis is framed, was to develop statistical methods for the system biology that integrate multiomic data and can lead to novel hypothesis about biological processes at different layers of cellular organization. In order to accomplish that general goal, this project is divided in four different secondary objectives:

- **Objective 1: To develop specific preprocessing pipelines for each omic data type.** Three different omic data types were generated within PROMETEO project: ChIP-seq of the histone mark H4 Acetylated (H4ac), RNA-seq and metabolomics. Specific aims were:

  – Perform quality control and normalization on the different omic data.

  – Assess technical validation and biological reproducibility of the different omic data.

  – Perform omic-wise data analysis including differential feature expression/quantification and functional enrichment.

- **Objective 2: To develop a batch effect correction algorithm for multiomic integration strategies.**

  Specific aims were:

  - Study the existing batch effect correction methods.
  - Analyze boundaries of batch effect magnitudes in real data.
  - Formulate a batch effect correction alternative that handles multiomic batch effect.
  - Simulate different multiomic batch effect scenarios to validate and test the limitations of our model.
  - Create an open source R package implementing the new approach.

- **Objective 3: To develop novel multiomic integration approaches.**

  Specific aims were:

  - Revision of the state of the art of model-driven multiomic integration and identify flaws of existing methodologies.
  - Develop novel Constraint Based Modeling (CBM) approach that integrates semi-quantitative metabolomic data and other gene-regulatory omic data.
  - Perform technical and biological validation of the new approach.
  - Use the new method to derive novel biological insights about the experimental setting under study.

2.3 Contributions

I arrived at Genomics of Gene Expression Lab in 2017 to work on the PROMETEO research project focused on multiomic integration analysis. One of my first tasks was to study, define and test processing pipelines for the different omic data types that were going to be generated under PROMETEO. Since internal project data were not generated at the beginning of the project, I used public data for this purpose and during this process, I came across an important issue involving the combination of omic data from different sources, which is the batch effect. Therefore, I studied the existing BECAs and found that there were not available methods for multiomic data batch effect correction when different omic data types come from different sources. This observation led to the development of MultiBaC method that would be published later [111] and coded into an R package [112], and will be presented in Chapter 3. This piece of work has been also presented at several national and international conferences at different states of development. At Mini-Arctic conference (November, 2017) I presented the method and the first validation steps. One year later, at Symposium on Bioinformatics (November, 2018) full method validation and its application to real data were exposed. Lastly, at ISMB/ECCB congress (July, 2019) MultiBaC R package was introduced.

When PROMETEO data were generated, previously defined and tested pipelines were utilized to process these data [113] which is explained in Chapter 4. To analyze our internal data in an integrative manner, I did a research stay at Quantitative Modelling of Cell Metabolism Lab (Danish Technical University) where I studied Contrainst Based Modelling (CBM) and collaborated to develop MAMBA method which is presented in Chapter 5. MAMBA was used to analyze our internal multiomic dataset and extract biological knowledge from them.

Moreover, during MultiBaC development I delved into multivariate statistical methodologies and their application to omic data, and attended to "Multivariate Process Analysis, Monitoring and Diagnosis" (MAMD) course that was part of the Master's Degree in Data Analysis, Process Improvement and Decision Support Engineering (Universitat Politèctica de València). As a result I collaborated in one publication [103] -advising in the application and interpretation of three-way PLS-, and taught multivariate methods at MIAGE course (2018). In addition to MAMD, I also attended a "Data Mining" course as part of the same Master's degree program and other courses to improve my skills as a scientist: i) Research dissemination strategies for researchers, ii) High standard for scientific production and communication, and iii) Document composition and high-quality presentations with LaTeX.

2.3.1 Articles in peer-reviewed journals

1 Víctor Sánchez-Gaya, Salvador Casaní-Galdón, Manuel Ugidos, Zheng Kuang, Jane Mellor, Ana Conesa and Sonia Tarazona. Elucidating the Role of Chromatin State and Transcription Factors

on the Regulation of the Yeast Metabolic Cycle: A Multi-Omic Integrative Approach. *Frontiers in Genetics*, 9:578, 2018.

2 Manuel Ugidos, Sonia Tarazona, José M. Prats-Montalbán, Alberto Ferrer and Ana Conesa. MultiBaC: a strategy to remove batch effects between different omic data types. *Statistical Methods in Medical Research*, 2020.

3 Carme Nuño-Cabanes, Manuel Ugidos, Sonia Tarazona, Manuel Martín-Expósito, Alberto Ferrer, Susana Rodríguez-Navarro and Ana Conesa. A multi-omics dataset of heat-shock response in the yeast RNA binding protein Mip6. *Scientific Data*, 7:69, 2020.

4 Manuel Ugidos, María J. Nueda, José M. Prats-Montalbán, Alberto Ferrer, Ana Conesa and Sonia Tarazona. MultiBaC: an R package to remove batch effects in multi-omic experiments. *Bioinformatics*, btac132, 2022.

5 Manuel Ugidos, Igor Marín de Mas, Sonia Tarazona, Carme Nuño-Cabanes, Alberto Ferrer, Lars Keld Nielsen, Susana Rodríguez-Navarro and Ana Conesa. MAMBA: a model-driven, constraint-based multiomic integration approach. *(In preparation)*.

2.3.2 Conference contributions

1. Manuel Ugidos, Sonia Tarazona, José M. Prats-Montalbán, Alberto Ferrer and Ana Conesa. MultiBaC: a strategy to remove batch effects between different omic data types. *Mini Arctic Conference*, Valencia, Spain, 2017.

2. Manuel Ugidos, Sonia Tarazona, José M. Prats-Montalbán, Alberto Ferrer and Ana Conesa. MultiBaC: a strategy to remove batch effects between different omic data types. *Jornadas de Bioinformática*, Granada, Spain, 2018.

3. Manuel Ugidos, Sonia Tarazona, José M. Prats-Montalbán, Alberto Ferrer and Ana Conesa. MultiBaC: a strategy to remove batch effects between different omic data types. *27th Conference on Intelligent Systems for Molecular Biology and the 18th European Conference on Computational Biology (ISMB/ECCB)*, Basel, Switzerland, 2019.

2.3.3 Software

1. MultiBaC R package. Available at https://bioconductor.org /packages/MultiBaC

2. MAMBA toolbox for matlab. Built in Python (10%) and MATLAB (90%). Available at https://github.com/ConesaLab/MAMBA

2.3.4 Teaching

1. MIAGE 2018 edition. Multiomic Integrative Analysis of Gene Expression (Centro de Investigación Príncipe Felipe, Valencia).

Chapter 3

Understanding and removing batch effects on a multiomic scenario

[1] Ugidos M, Tarazona S, Prats-Montalbán JM, Ferrer A, Conesa A. MultiBaC: A strategy to remove batch effects between different omic data types. Stat Methods Med Res. 2020 Oct;29(10):2851-2864.

[2] Ugidos M, Prats-Montalbán JM, Ferrer A, Conesa A, Tarazona S. MultiBaC: an R package to remove batch effects in multi-omic experiments. Bioinformatics, btac132, 2022.

3.1 Introduction

Over the last decade, high-throughput omic technologies such as transcriptomics, metabolomics, proteomics or epigenomics have become routine assays in many biological research laboratories. Increasingly, combinations of these methods are proposed to address complex questions about the molecular regulation of genomes and the physiology of cellular systems. As different omic assays target different biomolecules or chemical modifications, the combined study of these various molecular layers has the potential to provide insights into the complex regulatory networks that operate in living cells. However, simultaneously generating multiple omic measurements of the same molecular system for one particular study might be difficult. Challenges arise due to budgetary restrictions, time and sample limitations, or simply because of the convenience of a sequential analysis of the data in order to make informed decisions for follow up experiments. At the same time, researchers are no longer restricted to their own experimental capacities in order to obtain multiomic information, as facilities offer these assays on a commercial basis. Widespread editorial policies requiring omic data deposition in public repositories before publication of results have created a wealth of molecular data available to researchers for reuse. As a consequence, scientists have the opportunity to combine compatible data generated in other labs to compose a suitable multiomic dataset without the need of repeating experiments already performed by somebody else. Unfortunately, combining data obtained by different people and/or at different moments in time has an important drawback. Data will almost unavoidably be affected by an unwanted effect associated to the experimentation event that, especially for high throughput molecular assays, may result in important levels of noise contaminating the biological signal. This unwanted source of variation is commonly known as batch effect and is very frequently seen as the first component of variability in the omic dataset, standing out over the experimental conditions under study.

Batch effects significantly impair the power of statistical algorithms to detect significant true effects as they increase measurement errors and data variability. Removing batch effects becomes then necessary in order to obtain meaningful results from statistical analyses [52, 114]. Provided that the omic experiment has been designed in such a way that batch effects are not confounded with the effects of interest (e.g. treatment, disease, cell type, etc.), the so-called Batch Effect Correction Algorithms (BECAs) can be used to remove, or at least mitigate, systematic biases. Therefore these methods are extremely useful to combine data from different laboratories or measured at different times.

Several BECAs for omic data have been proposed. Limma [53] applies linear models while the ComBat method [115] from sva R package [116] estimates batch effects as the sum of an additive and a multiplicative effect with an empirical Bayes approach. RUV [117] estimates the unwanted variation from negative control genes that are known a priori to be unaffected by the biological factor of interest. We proposed the ARSyN approach [54], that relies on the ANOVA-Simultaneous Components Analysis (ASCA) framework [118, 119] to decompose the omic signal into experimental effects, the batch effect and residuals. ARSyN applies Principal Component Analysis (PCA) to estimate the systematic variation due to batch effect and then removes it from the original data. More recent research includes the commercial software Partek Genomic Suite; the ber R package [120], which assumes a model similar to ComBat; the exploBATCH [121] and guided PCA (gPCA) [122] R packages, both based on the study of the latent subspace to estimate the batch effect, as ARSyN does; and the BatchI R package [123], which removes batch effects of unknown sources using dynamic programming.

These methods have been traditionally applied to remove batch effects from omic data of the same type, as for example gene expression, and have been instrumental for the combination of data from the public domain into meta-analyses to reveal novel biological insights that cannot be discovered with small sample sizes [124–128]. However, while removing batch effects from a single omic data type with an appropriate experimental design is relatively straightforward, it can become unapproachable when dealing with multiomic datasets. In the multiomic scenario, each omic modality may be measured by a different lab or at a different moment in time, and so it is obtained within a different batch. When this is the case, the batch effect will be confounded with the omic type effect and impossible to remove from the data.

In this chapter, most popular single-omic BECAs are tested and compared. Moreover, MultiBaC method is presented, which is the first BECA dealing with batch effect correction in multiomic datasets. MultiBaC is able to remove batch effects across different omics generated within separate batches provided that at least one common omic data type is included in all the batches considered. Although this may seem a strong requirement, in practice there are many studies that include at least gene expression or popular histone marks as part of their multiomic design and hence provide opportunities for data combination across omic modalities. For example, stress response in yeast has been studied at the transcriptional rate [129–131], translational rate [132] and RNA-binding of global proteins [133], in three different studies that also included RNA-seq profiling. A method that corrects batch effects across omics will allow for the integration of these data in one single analysis that jointly evaluates different layers of transcriptional regulation by leveraging public resources and without the need of generating additional data. MultiBaC is effective in removing batch effects without introducing additional biases and outperforms adaptation of existing strategies to the multiomic batch problem. MultiBaC is therefore an effective tool to reuse existing datasets to perform meta-analysis across omics technologies.

3.2 Data

3.2.1 A yeast multiomic dataset obtained at different laboratories

We collected data from Gene Expression Omnibus (GEO) database pertaining to three different studies that analyzed the effects of glucose starvation in yeast. These studies used equivalent yeast strains and experimental conditions, but differed in the types of omic technologies profiled. Study A (Department of Biochemistry and Molecular Biology, Universitat de València) collected gene expression (RNA, with accession number GSE11521) and transcription rates (GRO, with accession number GSE1002) [129–131]. Study B (Department of Molecular and Cellular Biology, Harvard University) obtained gene expression (RNA) and translation rates (RIBO), with accession number GSE56622 [132]. Finally, Study C (Department of Biology, Johns Hopkins University) measured gene expression (RNA) and global PAR-CLIP data (gPAR-CLIP) with accession number GSE43747 [133]. Therefore, labs had one shared (RNA) and one distinct (GRO, RIBO and PAR-CLIP, respectively) data types. This distributed multiomic scenario represents the type of correction problem MultiBaC addresses. RNA-seq processed data from studies B and C studies were obtained from the GEO database and used without any further pre-processing since no technical biases were found using NOISeq package [51].

However the voom transformation from limma R package [53] was required to make data normally distributed as in study A. GRO-seq data from study B followed the same steps as gene expression data, including the voom transformation. For gPAR-CLIP data, genomic region-based quantification was downloaded and the translation to gene-based data was performed to analyze both data types with the same PCA model. To link regions with their closest genes we used RGmatch [134] with default parameters. We considered associations in which regions felt into the gene body or 100 bases upstream the transcription start site. Finally, the voom transformation was again applied. In contrast, raw data was downloaded for the study A and normalized as described in the original publications (code provided at Appendix 2). Briefly, this normalization procedure consists of using a genomic DNA hybridization signal (also available at the same GEO accession number) to correct the intensity of the mRNA and GRO hybridization. When all datasets were independently normalized (if required), a final TMM (Trimmed Mean of M values) normalization [35] using NOISeq R package was applied to make all samples have the same dynamic range.

3.2.2 Proof of concept data

We validated MultiBaC on two multiomic datasets that shared all omics modalities (GEO accession numbers GSE24488 [135] and GSE33136 [136]). In both GEO studies transcription rates and gene expression data were available and the experimental conditions compared were room temperature versus heat-shock stress in yeast. We denote these datasets "proof of concept" data because both omic data types are available from both studies and, hence, traditional BECAs for a single omic can be applied and compared to MultiBaC correction. Each of the two laboratories considered applied a different technology to obtain omic measurements: study 1 (GSE24488) used microarrays while study 2 (GSE33136) used sequencing techniques. Data from study 1 share the source with study A in the first section, thus the previously described pre-processing steps were performed. On the other hand, data available for study 2 at GEO were not suitable for our analysis. Hence, we downloaded the fastq files of the study (RNA-seq and GRO-seq files) and performed the complete pipeline from single-end reads to counts. First, mapping was done with TopHat2 [137] and the sacCer3 reference genome obtained from the University of California Santa Cruz (https://genome.ucsc.edu). Next, HTSEQ [34] with default parameters was used to obtain the gene counts. Once the gene counts were obtained, the quality control of NOISeq R package [51] detected library size and RNA composition biases that were corrected using the tmm function in this package. Finally, the voom transformation from limma R package [53] was applied on both data types. Again, once all datasets were independently normalized, a final TMM normalization using NOISeq R package [51] was applied to make all samples have the same dynamic range.

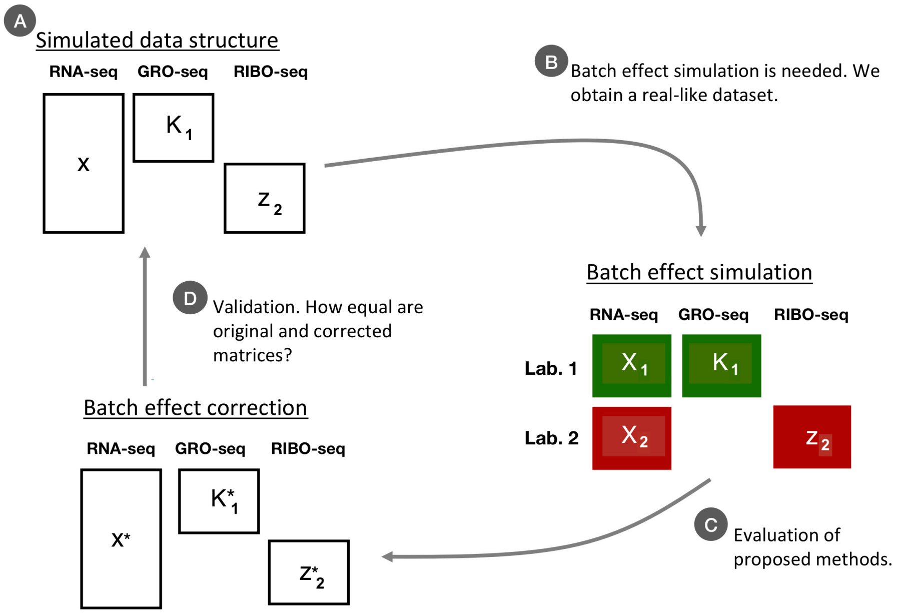


Figure 3.1: Scheme of MultiBaC testing and validation with simulated data [112].

3.2.3 Simulated data

A synthetic multiomic dataset was created that reproduces the scenario described in the yeast example. Figure 3.1 shows an overview of the whole process for MultiBaC testing and validation with simulated data.

A) We generated batch-free omic data with MOSim [138], a simulation tool that generates different omic data types together with the regulatory network between omic features, although we did not use this last information, but just the omic datasets. The data obtained with MOSim had the experimental design shown in Figure 3.1, i.e., a common omic with 2x observations and two additional non-common omics with 1x observations. We considered two conditions and 10 replicates per condition which results in 20 observations for each batch (1x = 20). Moreover, the number of features (gene-centric information) was the same for every omic data type and was set to five thousand.

B) We had to simulate the batch effect to get a design with two different laboratories, as in Figure 3.1. For that, we studied the batch effect behavior in the real datasets presented in this work (see Section 3.2.1 and 3.2.2) to know more about its magnitude and how to simulate it. The following multiple linear model was estimated to assess the dependence between the common data $\mathbf{X}$, containing K variables in columns and M samples in rows, and the batch and treatment factors:

$$\mathbf{X} = 1\bar{x}^t + \mathbf{B}\beta_1 + \mathbf{T}\beta_2 + \mathbf{BT}\beta_3 + \mathbf{E} = 1\bar{x}^t + \mathbf{CD} + \mathbf{E} \tag{3.1}$$

where 1 is a M size vector of ones, $\bar{x}^t$ is the K size vector of means for all the omic features, $\mathbf{B}$ is the batch design matrix with M rows and as many columns as the number of batches minus one (J), $\mathbf{T}$ is the treatment design matrix with dimensions M x N, being N the number of experimental conditions minus one, and $\mathbf{BT}$ is the interaction matrix of batch and treatment effects with dimensions M x $N \dots J$. $\mathbf{X}$ is the matrix that concatenates $\mathbf{X_1}$ and $\mathbf{X_2}$ by rows (observations) and omic features being in columns and E the M x K matrix of residuals. β_i coefficients were estimated by the least squares approach: $\hat{\mathbf{C}} = (\mathbf{D^tD})^{-1}\mathbf{D^tX}$, where $\hat{\mathbf{C}}$ is the matrix with the estimated coeffi-

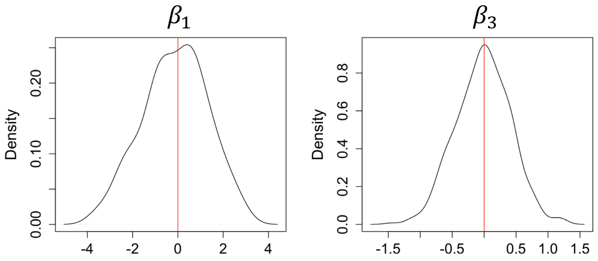


Figure 3.2: β_1 and β_3 distributions from batch effect estimation using mRNA expression data from Proof of concept Dataset .

cients ($\hat{\beta_1}$, $\hat{\beta_2}$ and $\hat{\beta_3}$) and $\mathbf{D}$ is the design matrix containing $\mathbf{B}$, $\mathbf{T}$ and $\mathbf{BT}$ dummy variables. Thus, $\hat{\beta_1}$ and $\hat{\beta_3}$ vectors contain the batch effect information, that is, the magnitudes of the batch and batch-treatment interaction effects. As shown in Figure 3.2, we can assume a normal distribution with $\mu = 0$ for β_1 and β_3. We also verified this assumption with a Shapiro-Wilk test [139, 140] (p-value $>$ 0.1 in both cases). Therefore, we can simulate batch effects from a normal distribution ($\beta_i \sim N(0, sd)$). The standard deviation sd will determine the magnitude of the effect. Table 3.1 shows the values for standard deviation of β_1 and β_3 in three real datasets. It was different in each case but always higher for batch (β_1) than for interaction (β_3) effect.

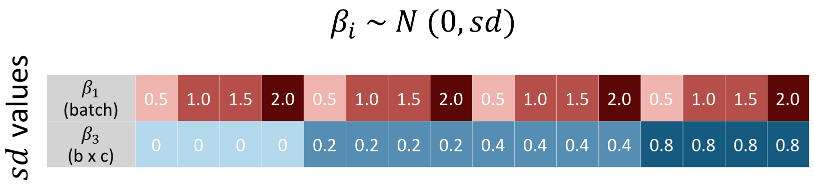
C) Once the normal distribution was assumed for β_1 and β_3 coefficients, we used this information to generate β_1^* and β_3^* values for each omic feature and used them to simulate batch effects as follows:

$$\mathbf{X}_{Batch} = \mathbf{X} + \mathbf{B}\beta_1^* + \mathbf{BT}\beta_3^* \tag{3.2}$$

Table 3.1: Batch/interaction effect simulation: Standard deviation values for β_1 (batch) and β_3 (interaction). Effect is referring to batch/interaction effect. First row: gene expression data from Lab A and Lab B ("A motivating example" data). Second row: gene expression data from Lab B and Lab C ("A motivating example" data). Third row: gene expression data from "Proof of concept data". Fourth row: transcription rates from "Proof of concept data"

		Mean	SD
GSE11521 and GSE566 (gene expression)	β_1	0	1.11
	β_3	0	0.57
GSE566 and GSE43747 (gene expression)	β_1	0	1.01
	β_3	0	0.25
GSE33136 and GSE24488 (gene expression)	β_1	0	1.27
	β_3	0	0.39
GSE33136 and GSE24488 (transcription rates)	β_1	0	1.5
	β_3	0	0.43

where $\mathbf{X}_{Batch}$ is the resulting matrix containing the simulated batch effect, and β_1^* and β_3^* are the simulated values for the coefficients randomly taken from a normal distribution with mean equal to zero and different standard deviation values as indicated in Figure 3.3. In total, sixteen different scenarios were simulated by modifying batch and/or interaction effect magnitudes. We distinguish three magnitude levels: low, moderate and high, being magnitudes low and moderate present in real experimental data. Regarding the batch effect magnitudes, *sd* values are considered low, moderate or high as follows: 0.5 is low, 1.0 and 1.5 are moderate and 2.0 is high. As for the interaction magnitude, 0.2 is considered as a low magnitude, 0.4 as a moderate magnitude and 0.8 as high.



$\beta_i \sim N\,(0, sd)$

sd values																
β_1 (batch)	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0
β_3 (b x c)	0	0	0	0	0.2	0.2	0.2	0.2	0.4	0.4	0.4	0.4	0.8	0.8	0.8	0.8

Figure 3.3: Standard deviations for batch effect and interaction simulated distributions. Sixteen different scenarios are generated .

D) Finally, the simulated data with batch effect are corrected to remove this effect and then compared with the batch-free simulated data to evaluate the performance of MultiBaC (or other methods).

3.3 BECAs usage: estimating the batch effect

As it will be explained later on, MultiBaC requires the use of a traditional BECA in its last step, so one of the previously mentioned algorithms had to be adapted and implemented to work together with MultiBaC. Three of the most common BECAs were compared: limma [53], ComBat [116, 141] and ARSyN [54]

3.3.1 Methodology behind BECAs

3.3.1.1 Limma

Limma method [53] uses linear models to estimate batch effect. Being y_{ijk} the information from gene y in batch i ,sample j and condition k, it can be expressed as follows:

$$y_{ijk} = \mu + \alpha_i + \gamma_k + \epsilon_{ijk} \tag{3.3}$$

where μ is the mean of gene y across all samples, α is the batch effect and γ is the treatment or condition effect. Translating this idea to a linear model expression, limma function computes batch effect by estimating β coefficients from the expression bellow:

$$y = \beta_0 + \beta_1 \cdot Batch + \beta_2 \cdot Treatment + \epsilon \tag{3.4}$$

In this case batch effect (α) has been translated to $\beta_1 \cdot Batch$ where batch is a dummy variable defining batch groups for samples. Similarly, γ has been substituted by $\beta_2 \cdot Treatment$. The inclusion of treatment effect in the model is not mandatory. The way limma corrects batch is subtracting its effect from original gene value following this formula:

$$y^* = y - \beta_1 \cdot Batch \tag{3.5}$$

Note that the correction value $\beta_1 \cdot Batch$ is common for all samples in the same batch.

3.3.1.2 ComBat

ComBat method from sva R package [116] described in W. Evan Johnson et al. 2007 [141] uses an empirical Bayes approach to estimate batch effect. For ComBat model, let Y_{ijg} be the expression value for gene g in sample j and batch i. The following model is defined:

$$Y_{ijg} = \alpha_g + X\beta_g + \gamma_{ig} + \delta_{ig}\epsilon_{ijg} \tag{3.6}$$

where α_g is the overall gene expression, X is a design matrix for treatments, and β_g is the vector of regression coefficients corresponding to X. The error terms, ϵ_{ijg}, can be assumed to follow a normal distribution. The γ_{ig} and δ_{ig} represent the additive and multiplicative batch effects of batch i for gene g, respectively. The batch-effect corrected data, Y^*_{ijg}, is given by

$$Y^*_{ijg} = \hat{\alpha_g} + X\hat{\beta_g} + \frac{Y_{ijg} - \hat{\alpha_g} - X\hat{\beta_g} - \hat{\gamma_{ig}}}{\hat{\delta_{ig}}} \tag{3.7}$$

where $\hat{\alpha_g}$, $\hat{\beta_g}$, $\hat{\gamma_{ig}}$ and $\hat{\delta_{ig}}$ are estimators (using empirical Bayes procedures) of α_g, β_g, γ_{ig} and δ_{ig} ,respectively.

3.3.1.3 ARSyN method

ARSyN (ASCA Removal of Systematic Noise) was presented by Nueda et al. [54] and is a batch effect correction approach that relies on the ANOVA-Simultaneous Component Analysis (ASCA) framework. Let

x_{ijr} be the gene expression of gene x, measured at treatment i, under batch j and for replicate r, which can be decomposed as in any ANOVA model as:

$$x_{ijr} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + (\alpha\beta\gamma)_{ijr} \tag{3.8}$$

where μ is an offset term, α_i the treatment effect, β_j the batch effect, $(\alpha\beta)_{ij}$ the interaction effect between batch and treatment, and $(\alpha\beta\gamma)_{ijr}$ the individual variation (residuals). If our omic data matrix X contains N genes in columns and M samples in rows, the previous equation can be expressed using matrix notation as:

$$\mathbf{X} = 1m^t + \mathbf{X_a} + \mathbf{X_b} + \mathbf{X_{ab}} + \mathbf{X_{abg}} \tag{3.9}$$

where 1 is a M size vector of ones, m is an N size vector containing the estimations of μ for each gene, matrices $\mathbf{X_a}$, $\mathbf{X_b}$ and $\mathbf{X_{ab}}$ contain the estimations of parameters α_i, β_j and $(\alpha\beta)_{ij}$ respectively, and $\mathbf{X_{abg}}$ contains the residuals $(\alpha\beta\gamma)_{ijr}$. Once this ANOVA-like decomposition is obtained, a PCA is applied on each submatrix and the number of principal components is determined for each case based on the required level of explained variability (see details at Appendix 3). The resulting ASCA model is:

$$\mathbf{X} = 1m^t + \overbrace{\mathbf{T_a}\mathbf{P_a^t} + \mathbf{E_a}}^{\mathbf{X_a}} + \overbrace{\mathbf{T_b}\mathbf{P_b^t} + \mathbf{E_b}}^{\mathbf{X_b}} + \overbrace{\mathbf{T_{ab}}\mathbf{P_{ab}^t} + \mathbf{E_{ab}}}^{\mathbf{X_{ab}}} + \overbrace{\mathbf{T_{abg}}\mathbf{P_{abg}^t} + \mathbf{E_{abg}}}^{\mathbf{X_{abg}}} \tag{3.10}$$

where $\mathbf{T_i}$ and $\mathbf{P_i}$ are the scores and loadings matrices from the PCA on each matrix $\mathbf{X_i}$, respectively; and $\mathbf{E_i}$ represents the residuals of PCA models. After estimating the effects with ASCA, ARSyN corrects the batch effect by subtracting undesirable effects from the original data according to the following equation:

$$\mathbf{X}^* = \mathbf{X} - \overbrace{(\mathbf{T_b}\mathbf{P_b^t} + \mathbf{T_{ab}}\mathbf{P_{ab}^t})}^{\text{Batch and interaction effects}} \tag{3.11}$$

where $\mathbf{X}^*$ is the corrected matrix without batch or interaction batch-treatment effects. Batch effect correction can be also performed without the interaction term.

3.3.2 Comparison of BECAs' performance

In order to compare these three BECAs, the gene expression information described in Section 3.2.3 was used since these traditional BECAs are only able to perform a single omic batch effect correction. Figure 3.4 shows the comparison of different BECAs' performance including the original non-corrected data. Althought Limma uses linear models, it does not take advantage of the interaction between batch and condition and the correction is not found to be as good as ARSyN or ComBat are. Nevertheless, linear models can be customized to include an interaction effect. However, there is an important concern about batch effect removal when including interactions bewteen the batch and other factors, which is the model overfitting. This issue could modify original omic values and even affect the biological signal of interest. Thus, the possibility to customize the model and the magnitude of the effect to be extracted from original data, is strongly desirable and that is exactly what ARSyN does. The highest performance of ARSyN, including the estimation of the batch effect and its interactions, is the most powerful method to correct the batch effect as shown in Figure 3.4.

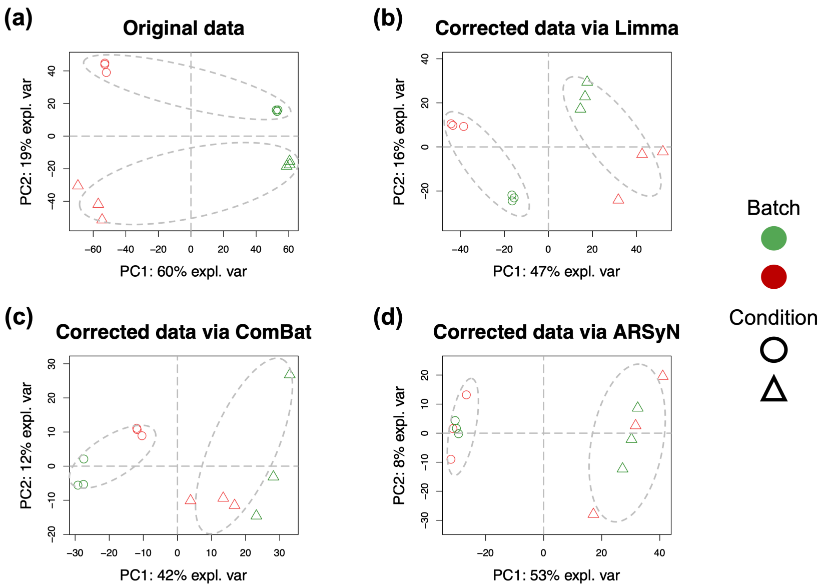


Figure 3.4: Comparison of BECAs' performance on "proof of concept" data by PCA score plots. (a) Original non-corrected data. (b) Batch effect correction using limma (linear models). (c) Batch effect correction using ComBat. (d) Batch effect correction using ARSyN. Gray dashed lines circle samples that should be clustered together since they belong to the same experimental condition .

Moreover, ARSyN allows a complete customization of the magnitudes of the effects removed by modifying the PCA models in Equation 3.10. In contrast, even though ComBat is a powerfull and very popular BECA, customizing the model performance is not an easy task and the interpretation of the resulting parameters is confusing compared to ARSyN or linear models.

While it should be acknowledge that, although by considering an interaction effect in ARSyN the biological signal may be diluted, the method does not impose a condition term, and instead, the model can be completely customized to accomodate any experimental design. Therefore, ARSyN method is the most versatile and powerful approach and those are the reasons why we implemented ARSyN inside MultiBaC method.

3.4 BECAs for multiomic data

3.4.1 MultiBaC: A multiomic batch effect correction strategy

MultiBaC (**Multi**omic **Ba**tch **C**orrection) method was conceived to correct batch effects across different omic data types provided that at least one omic modality is repeated in all the batches. In the formulation of the MultiBac method we consider that batch effect arises from different labs generating data, although the method is generally applicable to any other batch sources such as time or lab technician. Let us consider a minimal size problem example with two labs, each one of them measuring two different omic data types, one of them in common (Figure 3.5.a). We denote $\mathbf{X_1}$ as the common data type from lab 1, $\mathbf{X_2}$ as the common data type from lab 2, $\mathbf{K_1}$ as the non-common data type from lab 1 and $\mathbf{Z_2}$ as the non-common data type from lab 2. One important feature of MultiBaC is that the different omics studied in each lab do not have to share the variable space. This allows to combine gene-related omics (e.g. RNA-seq) with other technologies such as proteomics or metabolomics. However, MultiBaC requires that the same samples are measured for the different omic technologies obtained within the same batch. MultiBaC also assumes that each omic data matrix (X_1, X_2, K_1, Z_2) has been independently normalized to remove technical biases. We also recommend to transform sequencing count data to make them approximately follow a normal distribution (e.g. with log or voom transformations).

MultiBaC assumes that there exists a relationship between two different omic data types that does not depend on the laboratory. Basically, MultiBaC applies a multivariate PLS regression [75] to model the non-common omic data matrix as a function of the common omic measurements. The models are then used to predict the missing measurements what results in complete multiomic datasets in all laboratories. Next, traditional BECA methods are applied to correct the batch effect from the original matrices. MultiBaC proceeds through three steps (Figure 3.5.b):

In the *Modelling step*, PLS models are built for each lab, where the common omic data type is used as the explanatory matrix **X** and the non-common omic is used as the response matrix **Y**. The PLS model can be expressed as $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}$, where $\mathbf{B}$ is the regression coefficient matrix and $\mathbf{E}$ is the residuals matrix. $\mathbf{B}$ can be estimated as:

$$\mathbf{B} = \mathbf{W}^{*}\mathbf{C}^{\mathbf{t}} = \mathbf{W}(\mathbf{P}^{\mathbf{t}}\mathbf{W})^{-\mathbf{1}}\mathbf{C}^{\mathbf{t}} \tag{3.12}$$

where $\mathbf{W}$ is the $\mathbf{X}$-weight matrix, $\mathbf{P}$ is the $\mathbf{X}$-loading matrix and $\mathbf{C}$ is the $\mathbf{Y}$-weight matrix.

Therefore, considering a PLS model for each lab, for our minimal size problem, we will have the following PLS models:

$$PLS_1 : \mathbf{K_1} = \mathbf{X_1}\mathbf{B_1} + \mathbf{E_1} \tag{3.13}$$

$$PLS_2 : \mathbf{Z_2} = \mathbf{X_2}\mathbf{B_2} + \mathbf{E_2} \tag{3.14}$$

Q^2-based cross-validation (CV) optimization, proposed by Tenenhaus [142], is applied to select the optimal number of components for the PLS models, since Q^2 measures the marginal contribution of each component to the predictive power of the model. A good Q^2 value $>$ 0.7 is required to ensure that the model has a good prediction performance and can be used to infer the missing data modality. In the *Prediction step*, MultiBaC will estimate the missing omic data type for each lab by using the previously obtained PLS coefficient matrices:

$$\mathbf{\hat{Z}_1} = \mathbf{X_1 B_2} \tag{3.15}$$

$$\mathbf{\hat{K}_2} = \mathbf{X_2 B_1} \tag{3.16}$$

Note that, for predicting $\mathbf{\hat{Z}_1}$, the coefficient matrix relating $\mathbf{X_2}$ and $\mathbf{Z_2}$ is used, that is, $\mathbf{B_2}$. And the procedure is analogous for $\mathbf{\hat{K}_2}$. Remember that we aim to predict the omic information that was not initially available for each lab. This will allow us to remove the batch effect on non-common information with traditional methods using the original and the predicted information, i.e., $\mathbf{K_1}$ and $\mathbf{\hat{K}_2}$ for instance.

Finally, in the *Correction step* MultiBaC applies ARSyN to remove batch effect from every omic data type. Available data are used for the common omic, while predicted data must be used for the rest of omics.

$$\mathbf{X}^* = ARSyN(\mathbf{X_1},\ \mathbf{X_2}) \tag{3.17}$$

$$\mathbf{K}^* = ARSyN(\mathbf{K_1},\ \mathbf{\hat{K}_2}) \tag{3.18}$$

$$\mathbf{Z}^* = ARSyN(\mathbf{\hat{Z}_1},\ \mathbf{Z_2}) \tag{3.19}$$

where * means corrected matrix. Typically, we will discard now the predicted and corrected omic matrices $\mathbf{\hat{K}_2^*}$ and $\mathbf{\hat{Z}_1^*}$, and use the original and corrected matrices, $\mathbf{K_1^*}$ and $\mathbf{Z_2^*}$, for further statistical analyses.

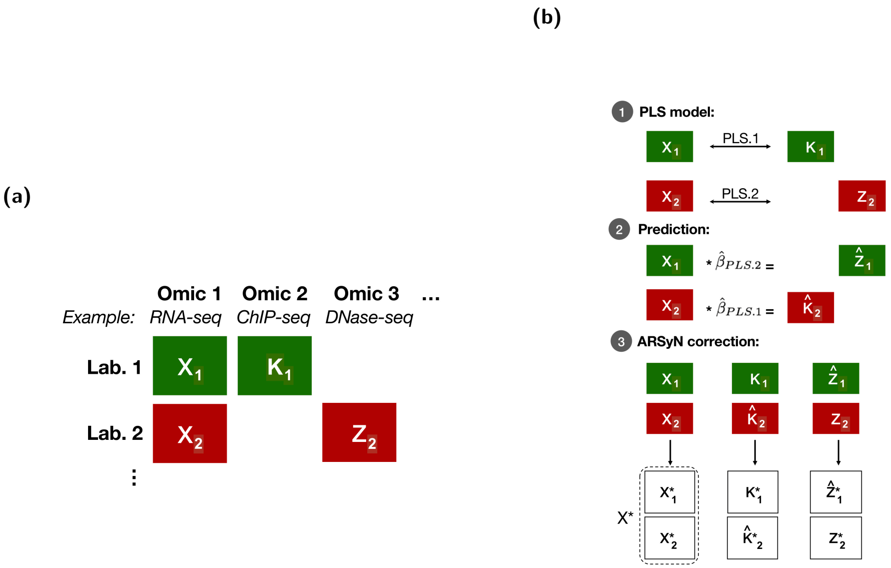


Figure 3.5: Description of MultiBaC method to correct batch effects in multiomic data from different laboratories. (a) Minimal size problem example in which one omic data type is shared by both laboratories and each laboratory may have other omic data types in an exclusive manner. (b) Overview of MultiBaC strategy, which combines PLS regression with conventional ARSyN batch effect correction. 1: A PLS model is built per laboratory to explain the non-common omic with the shared one. 2: For each laboratory, the initially missing omic is predicted. 3: ARSyN correction is applied on each omic data type by using predicted data .

3.4.2 Other multiomic batch effect correction approaches

In addition to MultiBaC strategy, we also adapted two other existing and conceptually different methodologies that theoretically could be applicable for solving the multiomic batch effect problem. These strategies were compared with the MultiBaC method.

3.4.2.1 *Missing data imputation strategy:*

In this approach, the values for the non-common omic data types are considered missing values for the laboratories where these omic data types are not available. Imputation of missing values is carried on with the multivariate method **T**rimmed **S**cores **R**egression (TSR) [143, 144], and then a BECA is applied (e.g. ARSyN). TSR models the structure in Figure 3.6.a containing missing values (NA) as a unique matrix ($\mathbf{X}$):

$$\mathbf{X} = [\mathbf{X}^{NA}\mathbf{X}^{*}] \tag{3.20}$$

where $\mathbf{X}^{NA}$ denotes the missing measurements and $\mathbf{X}^{*}$ the observed elements. TSR employs the latent space of the whole matrix, $\mathbf{T} = \mathbf{XP} = \mathbf{X^{NA}P^{NA}} + \mathbf{X}^{*}\mathbf{P}^{*}$, to impute missing data according to the relation between observed variables in each batch by using the common information as an inner reference, i.e, it reconstructs $\mathbf{T}$ from $\mathbf{T}^{*}$ using the model: $\mathbf{T} = \mathbf{T}^{*}\mathbf{B} + \mathbf{U}$, where the least squares estimator of matrix $\mathbf{B}$ is $\hat{\mathbf{B}} = (\mathbf{T^{*t}T^{*}})^{-1}\mathbf{T^{*t}T}$.

3.4.2.2 *Product transfer model:*

The Joint-Y PLS (JY-PLS) methodology presented by García Muñoz et al. [145] is based on PLS regression and assumes that both PLS response matrices ($\mathbf{X_1}$ and $\mathbf{X_2}$ in Figure 3.6.b) share the same latent structure. Note that response matrices in this model are $\mathbf{X_1}$ and $\mathbf{X_2}$

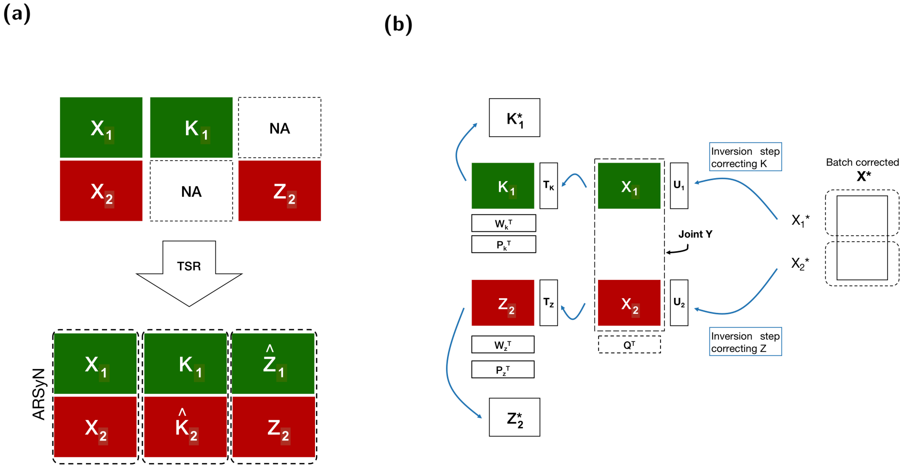


Figure 3.6: Outline of alternative methods for multiomic batch correction. The matrix notation used is the same as in Figure 3.5.a. (a) TSR: After TSR, a traditional BECA (e.g. ARSyN) can be applied. (b) JY-PLS: $\mathbf{W_i}$, $\mathbf{P_i}$ and $\mathbf{T_i}$ are weights, loadings and scores of $\mathbf{K_1}$ and $\mathbf{Z_2}$ matrices, respectively. $\mathbf{U_i}$ are the scores for X matrices. $\mathbf{Q^t}$ is the matrix of common weights of $\mathbf{X}$ matrices. $\mathbf{X}^*$ is used in the JY-PLS inversion step to obtain $\mathbf{K_1^*}$ and $\mathbf{Z_2^*}$.

(the common data type). Basically, JY-PLS builds a PLS model between $\mathbf{K_1}$ and $\mathbf{X_1}$ and another PLS model between $\mathbf{Z_2}$ and $\mathbf{X_2}$ by forcing $\mathbf{X_1}$ and $\mathbf{X_2}$ to share the same weight matrix (Q^t), i.e the same latent space. The ARSyN batch effect corrected common data type ($\mathbf{X}^*$) is used for the JY-PLS inversion step in order to obtain $\mathbf{K_1^*}$ and $\mathbf{Z_2^*}$, that is, the non-common batch effect corrected matrices. In brief, the inversion step tries to transfer a new set of responses which are the corrected data (e.g $\mathbf{X_1^*}$), in order to obtain which observations of the non-common omic could be in agreement with that set of responses (i.e $\mathbf{K_1^*}$).

3.5 MultiBaC validation

3.5.1 Validation of MultiBaC PLS models

Since MultiBaC uses PLS to model the relationships between omic datasets, the validation of the PLS model is critical to ensure a good MultiBaC performance. We investigated the influence of PLS prediction accuracy, determined by the Q^2 value, on MultiBaC performance using our simulated dataset. In this analysis, MultiBaC performance was evaluated using the conservation of differential expression calls. Assuming that batch effects impair combination of different experiments but do not affect the inner information structure of one experiment, we consider that differential expression (DE) analysis applied to individual omic matrices should give the same or very similar results before and after batch effect correction. In order to assess the concordance of such DE results, we considered the original data as the true values and the corrected data as the predicted values, and we used three different scores based on the number of DE genes obtained from each dataset: Sensitivity (SE), False Discovery Rate (FDR) and Specificity (SP). FDR measures the false positive rate, i.e. the percentage of genes declared as DE after correction but non-DE in the original matrix. SE assesses the ability to detect, after correction, all the DE genes obtained from the original data. Finally, SP appraises the ability to detect, after correction, all the initially non-DE genes. Differential expression analysis were performed using limma R package

[53]. (Figure 3.7). We obtained different values of Q^2 by adding different magnitudes of random noise to X (explanatory omic type) and Y (response omic type) matrices, obtaining two new noisy matrices X^N and Y^N, respectively (Figure 3.7.a). The relationship between Q^2 of the prediction model and MultiBaC performance results is shown in Figure 3.7.b. We observed that both FDR and SP were hardly impacted by the magnitude of the Q^2 value, but the SE was importantly reduced with lower Q^2s, i.e. the power for differential expression analysis was lower than in the original dataset. This result indicates that, when the prediction value of the PLS model is reduced, the resulting MultiBaC corrected datasets are compromised in their capacity for preserving their original biological signal, although they do not acquire false differences between experimental groups. MultiBaC returns Q^2 of the PLS model and we recommend that the method is only applied when Q^2 values are 0.7 and greater.

Another important aspect of PLS validation is the linearity. The inner relationship between the scores of response and predictor matrices for each component must be linear. We checked this requirement for MultiBaC PLS models when applied to experimental data (Figure 3.8 and 3.9) and observed a strong linear relationship in all cases, with correlation coefficients varying from 0.919 to 0.999.

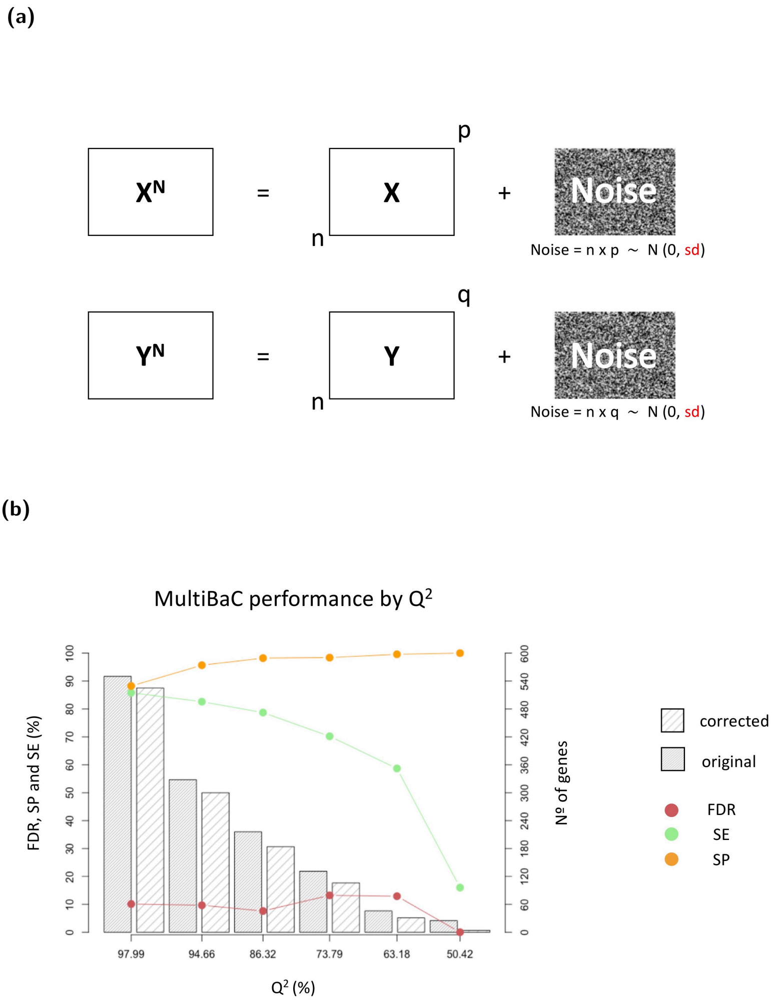


Figure 3.7: Evaluation of MultiBaC performance for different Q^2 values. a) Design scheme used to create analysis scenarios with different Q^2 values. This is done by adding different magnitudes of random noise to $\mathbf{X}$ (explanatory omic type) and $\mathbf{Y}$ (response omic type) matrices, obtaining two new noisy matrices $\mathbf{X}^{\mathbf{N}}$ and $\mathbf{Y}^{\mathbf{N}}$, respectively. b) Relationship between MultiBaC performance (FDR, SE and SP) and Q^2 values of the PLS model (first Q^2 value was obtained without adding noise). Results for original (before MultiBaC correction) and corrected (after MultiBaC correction) matrices are compared. Left Y axis are performance values (dotted lines). Right Y axis indicates the number of differentially expressed genes (bars) .

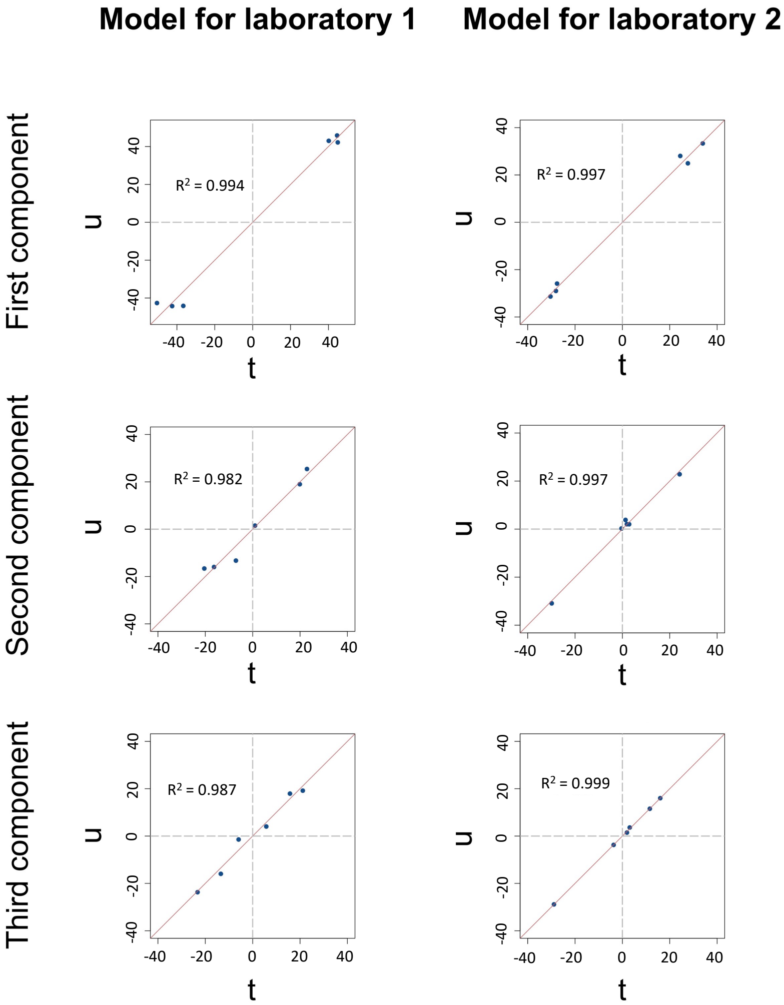


Figure 3.8: PLS models created from "Proof of concept" datasets. Inner relation between scores from response (u) and predictor (t) matrices for components 1-3. Red line is the diagonal, i.e. where t = u .

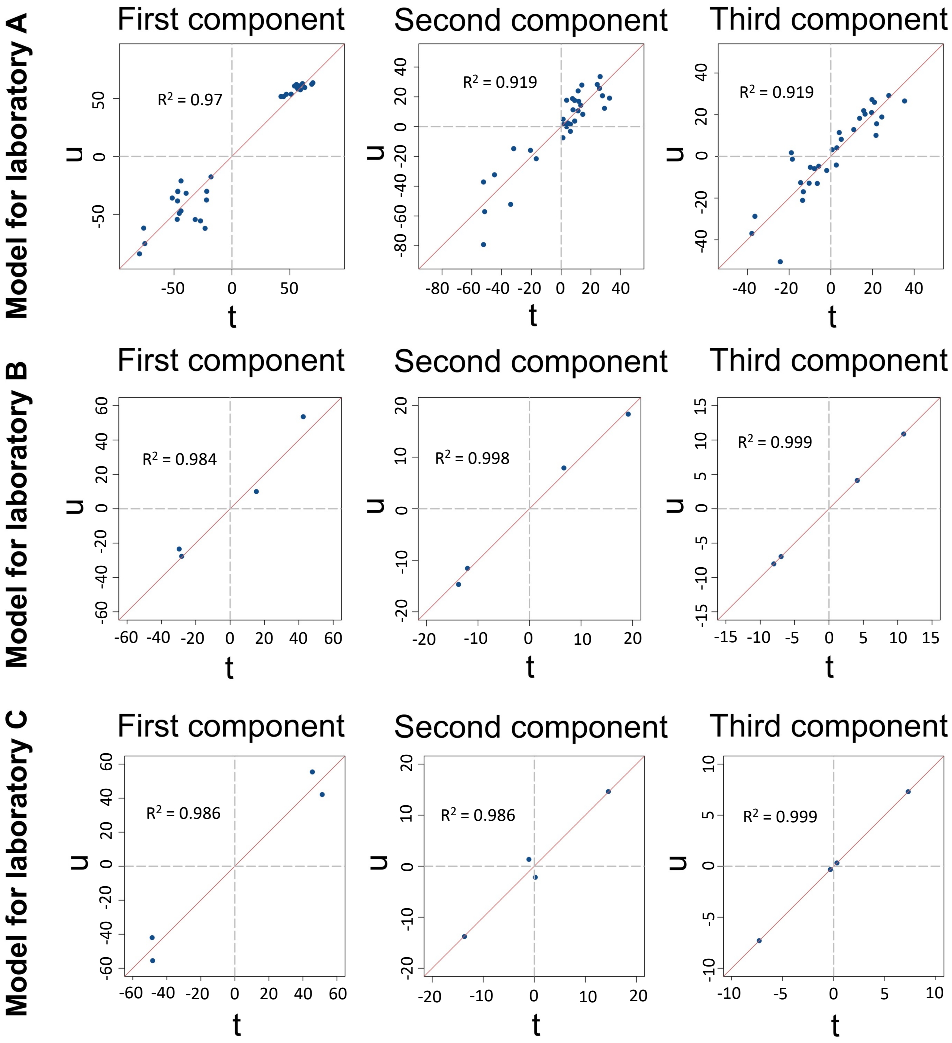


Figure 3.9: PLS models created from "Real problem" datasets. Inner relation between scores from response (u) and predictor (t) matrices for components 1-3. Red line is the diagonal, i.e. where t = u .

3.5.2 Multiomic BECAs comparison on simulated data

Simulated datasets (Section 3.2.3) were used to test the performance of MultiBaC method at removing batch effects and preserving the structure of the original data. Results of this part are summarized in Figure 3.10.

3.5.2.1 Latent space concordance.

This validation strategy was used to assess the performance of the methods on simulated data by evaluating if original data (before batch effect addition) and batch effect-corrected data shared the latent space in a PCA model. Latent space concordance (R^2) measures how well the variability structure of originally simulated matrices is able to explain the variability of corrected matrices, and the higher the R^2 the better the concordance. Considering the PCA model formula [$\mathbf{X} = \mathbf{TP^t} + \mathbf{E}$ (Equation 1.2)], R^2 represents how well $\mathbf{TP^t}$ models $\mathbf{X}$ matrix, and it is calculated as: $R^2 = 1-(SCR/SCT)$, where $SCR = \sum_{i=1}^{M}\sum_{j=1}^{N} e_{ij}^2$ and $SCT = \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}^2$ (M x N is the dimension of $\mathbf{X}$ matrix). We computed latent structure concordance by estimating a PCA model with the original data and computing R^2 for the corrected data after projection onto that PCA model Figure 3.10.a. In order to remove rotation effect differences, which could decrease the R^2, the PROCRUSTES algorithm [146, 147] was applied in this step. R^2 was high ($>$ 0.7) and very similar for the three tested methods at all batch magnitudes except for the highest values. Moreover, the intensity of the batch-condition interaction had little effect on the R^2 values. These results indicate that tested batch correction methods successfully recovered the latent structure of the unbiased data when batch effects were within limits observed in real datasets.

3.5.2.2 Differential expression analysis.

As described in the previous section, in order to assess the concordance of DE results between original and corrected data, we considered the original data as the true values and the corrected data as the predicted values, and we used three different scores based on the number of DE genes obtained from each dataset: Sensitivity (SE; Figure 3.10.b), False Discovery Rate (FDR; Figure 3.10.c) and Specificity (SP; Figure 3.10.d). Differential expression analysis were performed using limma R package [53].

The performance of the compared methods regarding these three indicators was greatly affected by the magnitude of the interaction effect between the batch and the experimental condition, while the batch effect magnitude did not seem to have an important effect. FDR is lower for MultiBaC than for the other two methods in all cases. In general, this indicator varies from 0 to 20%, while it reaches more than 50% in some cases for TSR or JY-PLS. In addition, MultiBaC FDR was less affected by the effect of the interaction when compared to the other methods. The increase in false positives caused SP rate to generally decrease at high interaction magnitudes, but JY-PLS and

MultiBaC performances were very similar, with scores above 80% in all cases, including at high interaction levels. Regarding SE results, MultiBaC was once again the best method, with SE above 95% in all simulations. This means that MultiBaC recovers all the originally differentially expressed genes, regardless the magnitude of the interaction effect. Altogether we conclude that MultiBaC outperforms compared methods and results in batch corrected data where no apparent additional biases have been introduced.

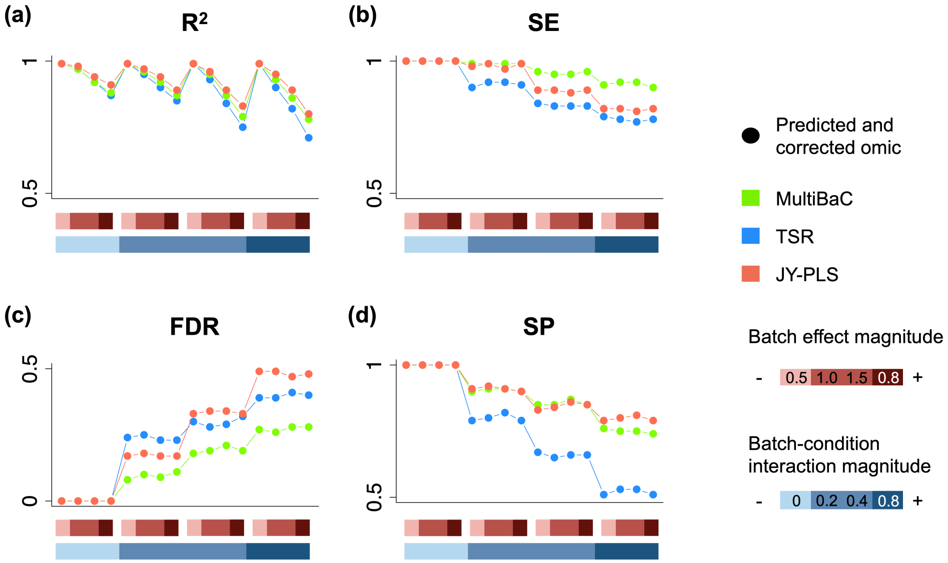


Figure 3.10: Performance of MultiBaC correction. Simulated data results. (a) Latent space concordance (R^2). (b) Sensitivity (SE) (c) False Discovery rate (FDR). (d) Specificity (SP). Rectangles at the bottom represent the batch (top) and interaction (bottom) magnitudes as explained in Section 3.2.2 .

3.5.3 MultiBaC validation using "Proof of Concept" data

Next, MultiBaC was further validated with proof of concept data, where the same two omics (gene expression and transcriptional rates) had been measured by two different laboratories. Consequently, traditional BECAs can be applied on each omic data type to remove the laboratory effect and results can be compared to methods correcting batch across omics. ARSyN and ComBat were used as BECA methods. For MultiBaC, JY-PLS and TSR, we assumed that gene expression was the common omic and transcriptional rates were non-common between labs. We evaluated these results by comparing PCA plots from the original and corrected data (Figure 3.11). Matrices with different omic information were merged by genes to compute PCA. As expected, the PCA of the original data (Figure 3.11.a) showed a strong effect of the laboratory, captured by the first principal component (PC). This effect should be removed if correction was successful. PCA plots showed that the best batch effect correction is performed by ARSyN (Figure 3.11.b) where batch effect is completely removed form PC 1 and 2, followed by MultiBaC (Figure 3.11.c), which still maintains a residual lab separation for GRO in the second PC. This separation is much larger both for JY-PLS (Figure 3.11.d) and TSR (Figure 3.11.e). Interestingly, ComBat (Figure 3.11.f) did not appropriately correct the batch effects although it performs a omic-wise correction, since labs are still separated by omics and conditions in the first or second PCs.

After MultiBaC correction, the strongest effect (first PC) is related to the experimental condition, similarly to ARSyN correction. The separation between omic data types (second PC) is due to the fact that each one of them provides information about different biological aspects of the system under study and not to bath effects. Therefore, this example illustrates that MultiBaC performance on experimental data is equivalent or superior to established BECAs with the advantage that MultiBaC can be applied when specific omic data types are not included in all batches.

3.5.4 MultiBaC application to a real problem

Lastly, we applied MultiBaC to the real distributed multiomic dataset, with three labs having gene expression (RNA) as common omic data type and a second omic assay as non-common (namely GRO, RIBO and PAR-CLIP). These data showed a pronounced batch effect (Figure 3.12.a (left-panel)) that stood out above omic methodology and experimental condition. MultiBaC was successful at correcting these biases (Figure 3.12.a (right-panel)). After correction, PCA clustered samples by omic type rather than by laboratory and, within each technology, separation of samples from the two experimental conditions was observed, suggesting that technical noise was removed to reveal biological information. Since no separation is observed between labs for the common omic, we now expect that separation between the rest of omic data types is mostly due to the different biological informa-

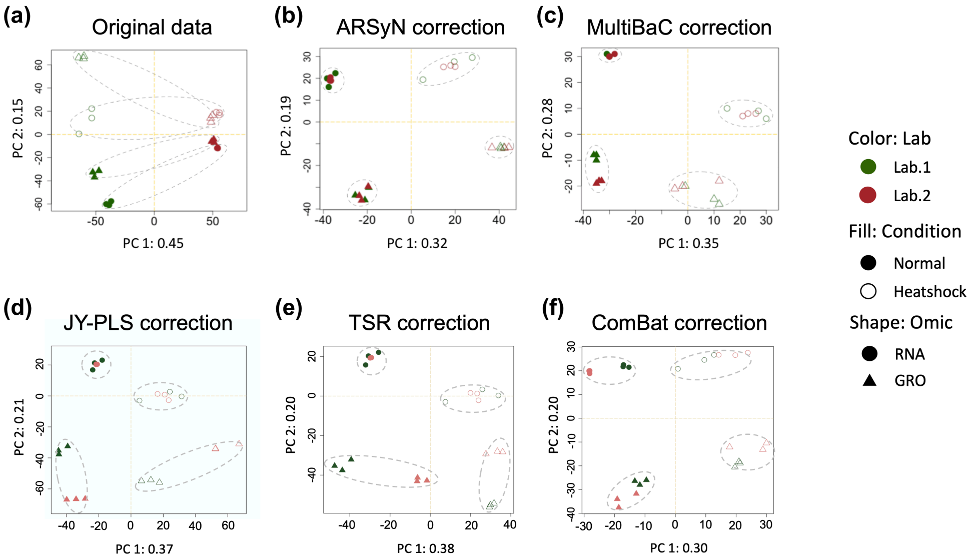


Figure 3.11: Performance of batch effect correction on Proof of concept data. (a) PCA score plot for original data. First principal component (main source of variability) groups samples by lab instead of by omic or treatment. (b) PCA score plot for ARSyN batch-corrected data. (c) PCA score plot for MultiBaC batch-corrected data. (d) PCA score plot for JY-PLS batch-corrected data. (e) PCA score plot for TSR batch-corrected data. (f) PCA score plot for ComBat batch-corrected data. ComBat corrects batch effect at a single omic level as ARSyN does, thus this result is only comparable to ARSyN pannel. Dashed line ellipses are grouping samples from different batches by omic-condition factor.

tion they provide. We further evaluated that MultiBac preserved the biological information between experimental conditions by comparing differential expression calls between corrected and non-corrected data (Table 3.2), as well as the number of common genes in both analyses. We computed FDR, SE and SP by taking the original data as the true reference. Although original data do not represent a real true refer-

Table 3.2: Differential expression results for the yeast multiomic dataset obtained at different labs. First column (Original) contains the number of differentially expressed genes (DEG) for each omic computed from original data. Second column (Corrected) contains the same results but computed from corrected data. Third column (Common) displays the number of DEG that are common to both analyses. FDR, SE and SP (columns 4-6) were calculated in percentage by assuming original results as true. Differential expression for omics with the symbol $*$ was computed without adjusting p-values. Last row (TOTAL) shows the number of DEG obtained in at least one omic.

	Original	Corrected	Common	FDR	SP	SE
	n^o of genes			%		
GRO	3075	2616	2615	0.038	99.950	85.041
RNA	2440	2487	2440	1.889	98.253	100
RIBO*	109	87	87	0	100	79.817
PAR*	653	607	601	0.988	99.089	92.037
TOTAL (unique)	4135	4445	3906			

ence without batch effect as happened in simulated data, these results are still useful to compare the effect of MultiBaC correction with ARSyN performance (only applied on RNA data) in terms of differential expression results.

The sensitivity to detect true positives (SE) was high, around 80% in the worst case (RIBO-seq), while the specificity exceeded 98% in all cases and FDR was always below 2%. RNA measurements can be considered as a control since the correction was made with the ARSyN method. In this case, a small increase in RNA number of DEGs revealed that correction slightly affected differential expression results, even when traditional BECAs and MultiBaC were applied. This is expected as the removal of batch effects reduces the variability within experimental conditions and hence improves the differential expression results. Even so, most DEGs were recovered after correction and we can state that MultiBaC preserves most of the biological information in the original data, as happens with any other traditional BECA.

Genes declared as differentially expressed in at least one of the omics (4135 for the original set and 4445 for the corrected set) were selected for clustering analysis in order to check if gene profiles across omics and conditions changed after correction. K-means algorithm [148, 149] was applied for clustering analysis and each cluster was labeled by its pattern of change (Table 3.3) across omic data types.

Table 3.3: Clusters characterization. Each cluster obtained is characterized by a differential behavior shared by all genes in that cluster. Up or down means up- or down-regulated genes in treatment condition versus control condition.

Cluster	Pattern
1	GRO down
2	GRO and RNA down
3	GRO up and PAR down
4	RNA down
5	PAR down
6	GRO up
7	RNA up
8	GRO and PAR down
9	PAR up

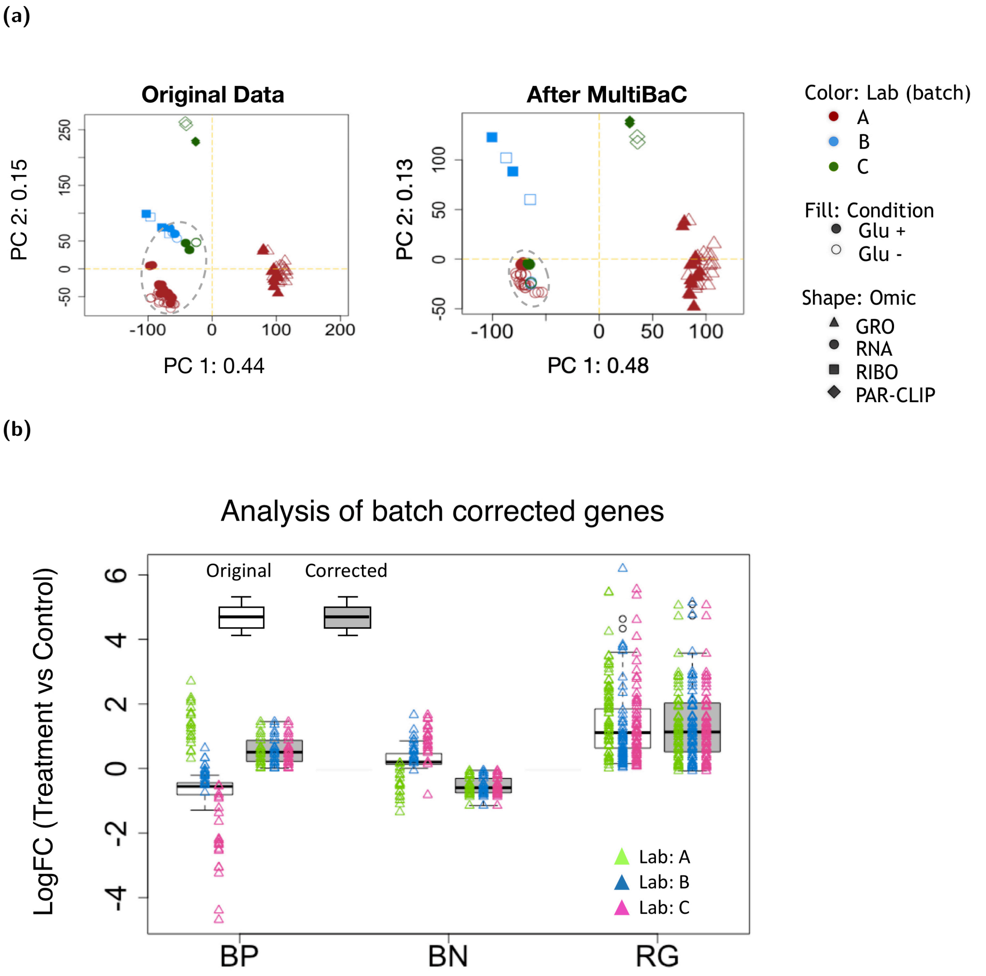


Figure 3.12: MultiBaC results on the "distributed yeast multiomics dataset" data. (a) PCA score plot of the global matrix with all the omic data types (merged by genes) after MultiBaC correction. Dashed line ellipses are grouping samples from different batches by omic-condition factor. (b) RNA values of 42 genes that have changed the sign of their logFC after correction. Become Positive Genes (BP) are genes that were down-regulated in the original data (white boxes) but up-regulated after correction (gray boxes). Become Negative genes (BN) had the opposite behavior. Random Genes (RG) are 100 up-regulated genes randomly selected. Triangles show the logFC value for each single gene in each lab.

The number of genes in each cluster before and after MultiBaC correction was compared and only 42 genes inverted their trend from up to down regulation or viceversa and just for RNA while the other omic data types conserved their gene trends. Among these 42 genes, 21 were classified as become positive (BP) genes, since they were initially down-regulated and after correction they became up-regulated. The other 21 become negative (BN) genes followed the opposite behavior, that is, they were initially up-regulated and after the correction they were down-regulated.

A functional enrichment analysis of these 42 genes did not return any significant result, which means that these genes are involved in many different functions but their change in trend when correcting batch effect is not related to any specific functional category. In order to further understand why these genes changed their trend, we compared their expression values to those of 100 randomly selected up-regulated genes for RNA (RG) that did not change their trend after correction (Figure 3.12.b). We found that BP genes were originally up-regulated in Lab A despite of being down regulated when performing the average between labs. The same happens for BN genes, they were initially down regulated in one lab. Interestingly, for RG randomly selected genes, the mean value was the same for all labs and there was no discordant information. This result suggests that MultiBaC corrects genes with a true laboratory associated bias. For other genes MultiBaC slightly modified the value of the fold-change without introducing a switch in the direction (sign) of the change.

Finally, we compared MultiBaC results with those from P. L. Nagy et al., 2003 [132] (Lab B in our example). They focused their analysis on two groups of genes: RNA & Ribosome Occupancy (RO) up-regulated (G1) and RNA up- & RO down-regulated (G2) (see Figure 3.13). In [132], RO denotes the ratio between RIBO and RNA values, while the ratio between GRO and RNA is named as Polymerase Occupancy (PO) and we used here the same notation. Regarding RO ratios, there are no large differences between the original and the corrected state. However, the PO ratio is greater after correction. This result agrees and improves the conclusions of the cited paper, where PO values were approximately the half of RNA values. This means that MultiBaC correction improved the relationship between omics improving accuracy and in agreement with previous studies.

3.6 MultiBaC implementation as an R package

MultiBaC algorithm is available at Bioconductor repository under the same name (https://bioconductor.org/packages/MultiBaC). The MultiBaC R package integrates two different batch effect correction methods: MultiBaC, which deals with batch effect correction in multiomic designs, and ARSyN (previously described [54]), a flexible method for single omic type batch effect correction applicable to different data

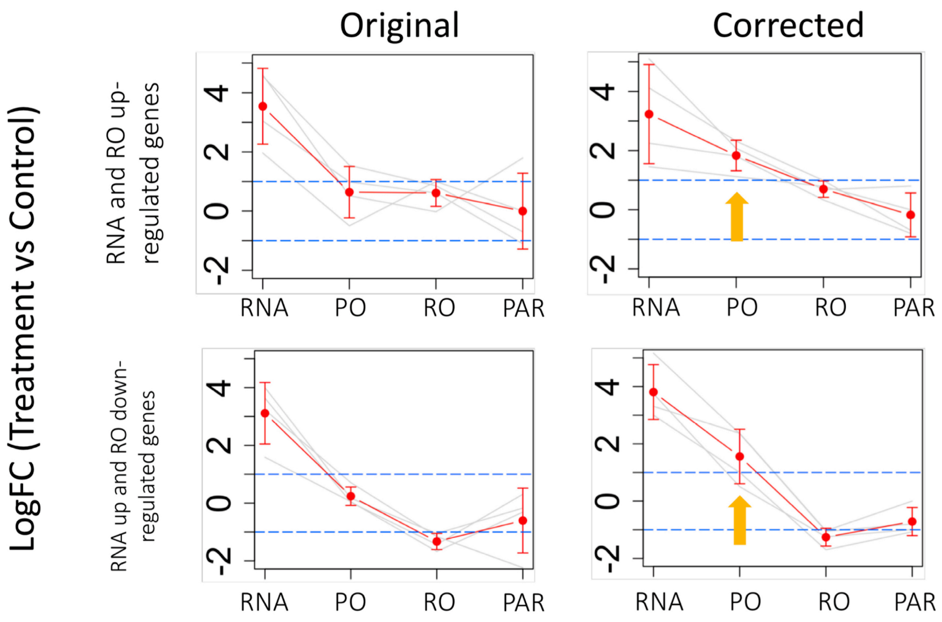


Figure 3.13: LogFC values per omic before and after MultiBaC correction. First row: RNA and Ribosome Occupancy (RO) up-regulated genes. Second row: RNA up-regulated but RO down-regulated genes. Each line corresponds to the profile of a gene in the corresponding group. The doted central line is the average profile of all the genes in the group, and the segment at each point represents the mean value ± the standard deviation. Yellow arrows indicate the increase in Polymerase Occupancy (PO) logFC values after correction .

modalities (Figure 3.14). In this section, the most relevant functions and objects of MultiBaC R package are presented and a more detailed description is provided in the package's vignette (Appendix 3).

The MultiBaC package uses MultiAssayExperiment objects, a type of Bioconductor container for multiomic studies [150], that can be created from a list of matrices or data.frame objects. These matrices must have features in rows and samples in columns. It is important that all data matrices share the sample space. In addition, common omic matrices across different batches must share the variable space. Thus, if the number of omic variables and order are not the same, the createMbac function will select the common variables. Hence, it is mandatory that rows are named with the same type of identifiers. A MultiAssayExperiment object needs to be created for each batch. The mbac new data structure is a S3 list class of MultiAssayExperiment objects and can be easily generated with the createMbac function in the package. The resulting mbac object will be the ARSyNbac or MultiBaC input.

These are the arguments for the `createMbac` function:

- **inputOmics** A list containing all the matrices or data.frame objects to be analysed. MultiAssayExperiment objects can alternatively be provided.
- **batchFactor** Either a vector or a factor indicating the batch were each input matrix belongs to (i.e. study, lab, time point, etc.). If NULL (default) no batch is considered and just ARSyNbac noise reduction mode could be applied.
- **experimentalDesign** A list with as many elements as batches. Each element can be a factor, a character vector or a data.frame indicating the experimental conditions for each sample in that batch. When being a data.frame with more than one column (multi-factorial experimental designs), the different columns will

be combined into a single one to be used by MultiBaC functions. In any case, the experimental setting must be the same for all batches. In addition, the names of the elements in this list must be the same as declared in `batches` argument. If not (or if NULL), names are forced to be the same in as in `batches` argument and in the same order.

- **omicNames** Vector of names for each input matrix. The common omic is required to have the same name across batches.
- **commonOmic** Name of the common omic between the batches. It must be one of the names in omicNames argument. If NULL (default), the omic name which is common to all batches is selected as commonOmic.

The `mbac` R structure generated by the `createMbac` function is an S3 object that contains just one slot, the `ListOfBatches` object. However, the `mbac` structure may contain more elements that are created when running the `ARSyNbac`: `CorrectedData` and `ARSyNmodels`. Moreover, after applying `MultiBaC` (explained in next sections) this object can incorporate two more slots: `PLSmodels` and `InnerRelation`. The five slots contained in the `mbac` object are next described:

- **ListOfBatches**: A list of MultiAssayExperiment objects (one per batch).
- **CorrectedData**: Same structure than ListOfBatches but with the corrected data matrices instead of the original ones.

- **PLSmodels**: PLS models created by MultiBaC method (one model per non-common omic data type). Only available for MultiBaC method.
- **ARSyNmodels**: ARSyN models created either by ARSyNbac or MultiBaC functions.
- **InnerRelation**: Table of class `data.frame` containing the inner correlation (i.e. correlation between the scores of X (t) and Y (u) matrices) for each PLS model across all components, for model validation purposes. Only available for MultiBaC method.
- **commonOmic** Name of the common omic between batches.

3.6.1 ARSyN batch effect correction

The ARSyN method is implemented into the ARSyNbac function in MultiBaC package. The arguments of ARSyNbac function are:

ARSyNbac (mbac, batchEstimation = TRUE, filterNoise = TRUE, Interaction=FALSE, Variability = 0.90, beta = 2, modelName = "Model 1", showplot = TRUE)

- **mbac**: mbac object generated by `createMbac`.
- **batchEstimation**: Logical. If TRUE (default) the batch effect is estimated and used to correct the data. If batch effect is unknown or it is not the main source of noise, this argument must be

set to FALSE and ARSyNbac will extract unwanted effects from residuals.

- **Interaction**: Logical. Whether to model the interaction between factors or not (FALSE by default).
- **Variability**: From 0 to 1. Minimum percent of data variability that must be explained by each model. Used in batch correction mode. By default, 0.90.
- **filterNoise**: Logical. If TRUE (default) structured noise is removed form residuals. Use this option when there is an unknown source of batch effect in data.
- **beta**: Numeric. Components that represent more than beta times the average variability are identified as systematic noise in residuals. Used in noise reduction mode. By default, 2.
- **modelName**: Name of the model created. This name will be showed if you use the explained_varPlot function. By default, "Model 1".
- **showplot**: Logical. If TRUE (default), the explained_varPlot is showed. This plot represents the number of components selected for the ARSyN model.

When the batch is identified in the `batchFactor` argument of the `mbac` input object (known source of batch effect), its effect can be estimated and removed by choosing `batchEstimation = TRUE`. Moreover, a possible interaction between the experimental factors and the batch factor can be studied by setting `interaction=TRUE`. In addition, ARSyNbac can also correct data when the source of batch effect is unknown and in turn cannot be estimated (`batchEstimation = FALSE` and `filterNoise = TRUE`). Finally, when both, known and unknown batch effect, are present, ARSyNbac is able to correct both sources of unwanted variation (`batchEstimation = TRUE` and `filterNoise = TRUE`).

3.6.2 MultiBaC correction

Once the `mbac` object has been created with a multiomic design, it is used as the input data for MultiBaC function (`mbac` argument), which is the wrapper function for the correction of multiomic batch effects.

MultiBaC (mbac, test.comp = NULL, scale = FALSE, center = TRUE, crossval = NULL, Interaction = FALSE, Variability = 0.90, showplot = TRUE, showinfo = TRUE)

The arguments of the MultiBaC function correspond to the different steps of the MultiBaC method:

- **mbac**: mbac object generated by `createMbac`.
- **test.comp**: Maximum number of components allowed for PLS models. If NULL (default), the minimal effective rank of the matrices is used as the maximum number of components.

- **scale**: Logical. Whether X and Y matrices must be scaled. By default, FALSE.
- **center**: Logical. Whether X and Y matrices must be centered. By default, TRUE.
- **crossval**: Integer: number of cross-validation segments. The number of samples (rows of 'x') must be at least >= crossvall. If NULL (default), a leave-one-out crossvalidation is performed.
- **Interaction**: Logical. Whether to model the interaction between experimental factors and bacth factor in ARSyN models. By default, FALSE.
- **Variability**: From 0 to 1. Minimum percent of data variability that must be explained for each ARSyN model. By default, 0.90.
- **showplot**: Logical. If TRUE (default), the Q2 and the explained variance plots are shown.
- **showinfo**: Logical. If TRUE (default), the information about the function progress is shown.

3.6.3 Visualization of results

As mentioned before, `ARSyNbac` and `MultiBaC` outputs are `mbac` type objects. Since the `mbac` class incorporates a plotting method, the `plot` function can by applied on `mbac` objects to graphically display additional information about the performance of the methods or the data

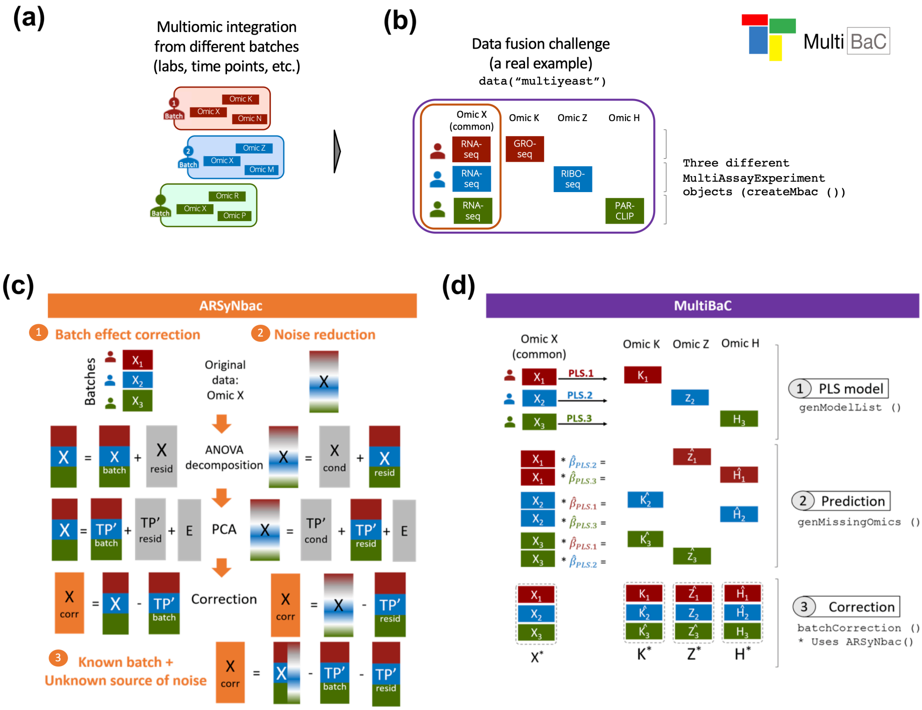


Figure 3.14: Graphical abstract of MultiBaC R package. (a) Multiomic integration challenge where the batch effect is present and confunded in several omic data. (b) Structure of data on a real case problem where at least one omic has been measured in all the batches. All the matrices represented in this figure have variables in columns and samples in rows. (c) ARSyNbac overview for three different options: 1) Batch effect correction, 2) Noise reduction for unknown batches, and 3) Correction when both types of unwanted effects are present in the data. In all cases, an initial ANOVA-like decomposition is performed and followed by PCA for the estimation of unwanted effects. (d) Overview of MultiBaC strategy, which combines PLS regression with conventional ARSyN batch effect correction .

characteristics. The `plot` function for `mbac` objects accepts several additional arguments:

plot (x, typeP = "def", col.by.batch = TRUE, col.per.group = NULL, comp2plot = c(1,2), legend.text = NULL, args.legend = NULL, ...)

Description of the arguments:

- **x**: mbac object generated by `createMbac`, `ARSyNbac` or `MultiBaC`.
- **typeP**: The type of plot to be displayed. Options are: "def" (default option, "Q2 plot" and "Explained variance plot" for MultiBaC and "Explained variance plot" for ARSyNbac), "inner" (inner correlation plots for each PLS model across the components of MultiBaC output), "pca.org" (PCA plot of original data), "pca.cor" (PCA plot of corrected data for MultiBaC or ARSyNbac outputs), "pca.both" (PCA plots for both original and corrected data for MultiBaC or ARSyNbac outputs), and "batch" ("Batch effect estimation" plot for all the outputs). PCA plots can only be generated when all data matrices share the same variable space.
- **col.by.batch**: Argument for PCA plots. Logical. If TRUE (default), samples are colored according to the batch factor. If FALSE, samples are colored according to the experimental conditions.

- **col.per.group**: Argument for PCA plots. Color for each group (given by batches or experimental conditions). If NULL (default), the colors are taken from a predefined pallete.
- **comp2plot**: Argument for PCA or InnerRel plot. It indicates which components are to be plotted. The default is c(1,2), which means that, in PCA plots, component 1 is plotted in "x" axis and component 2 in "y" axis, and for InnerRel plots, the inner relation plots of components 1 and 2 are shown. If more than two components are given, the function will return as many plots as needed to show all the components.
- **legend.text**: Argument for PCA plot. A vector of text used to construct a legend for the plot. If NULL (default) batch or conditions names included in the mbac object are used.
- **args.legend**: List of additional arguments to pass to legend(). Names of the list are used as argument names. Only used if legend.text is supplied.
- **...**: Other graphical arguments.

While the `plot` function can generate all the plot types described above, each plot can also be independently generated by its corresponding function (Figure 3.15): `Q2_plot(mbac)`, `explained_varPlot(mbac)`, `plot_pca(mbac)`, `batchEstPlot(mbac)`, or `inner_relPlot(mbac, comp2plot = c(1,2))`. All these plots are useful to validate or understand `ARSyNbac` or `MultiBaC` performance.

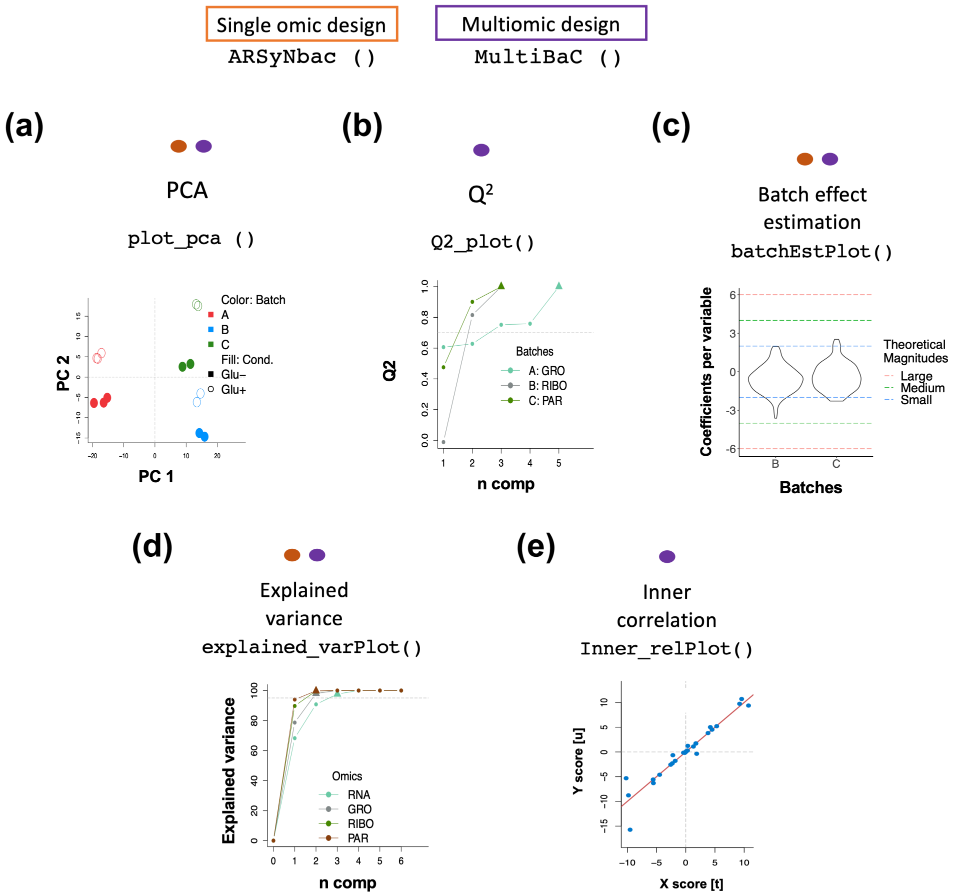


Figure 3.15: Collection of visualization possibilities that MultiBaC provides to the users. Colors represent the output needed for each plot (orange: ARSyNbac, purple: MultiBaC). (a) PCA plot. (b) Q2 plot. (c) Batch effect estimation plot. (d) Explained variance plot. (e) Inner correlation plot .

3.7 Discussion

Many methods have been proposed to efficiently remove unwanted effects from omic data, such as effects related to lab, machine, protocol, etc., which are known in general as batch effects. These approaches (BECAs) deal with just one omic data type at a time and, to the best of our knowledge, no strategy has been suggested yet for the multi-omic context, where each omic may have been produced in a different lab, by a different person or at a different period. Obviously, when two different omics have been generated in two different batches, it is difficult, if not impossible, to distinguish between the effect of the batch and the effect of the omic type itself. However, it is possible to estimate the batch effect between different omics when there is at least one common omic data type in all the batches. In this work we introduce MultiBaC, a new methodology to correct batch effects when integrating multiomic datasets in this scenario. Thus, the only requisite to apply MultiBaC is that one omic data type must be shared by all the batches to allow batch effect estimation and removal.

We show the application of MultiBaC to integrate different omic technologies obtained for the same biological system at different labs. However, MultiBaC could be in principle applied in other situations such as experiments where the same omic data type has been generated by two different techniques or protocols. One example could be metabolomics obtained with Gas Chromatography (GC) and High-Pressure Liquid Chromatography (HPLC), where a few metabolites are shared by both protocols but the rest of metabolites are specific of each protocol. The common metabolites would constitute the common information and MultiBaC can be applied to remove the protocol effect so both datasets can be joined in a single analysis.

To prove the ability of MultiBaC to correct batch effect, we applied the method on simulated multiomic scenarios. As there are not established multiomic batch correction methods, we adapted and applied two suitable existing algorithms (JY-PLS and TSR) and compared them to our MultiBaC approach. The performance of MultiBac and the other methods naturally depends on the magnitude of the batch effect and on how much this effect interacts with the effect of the experimental factor of interest. MultiBaC correction worked extremely well at batch levels expected for these technologies. Batch magnitude affected the latent structure similarity between original and corrected data but it did not affect differentially expressed genes (DEG). With extreme interaction magnitudes MultiBaC performance was compromised although it was still the best approach. We concluded that our results under the moderate interaction scenario represent very well the MultiBaC performance with real interaction effects. All in all, our analyses showed a good performance of the correction methods in realistic scenarios with MultiBaC outperforming in all simulated scenarios when correcting real experimental datasets with a strong laboratory effect. In the "proof of concept" dataset, where traditional BECAs could also be applied, results obtained with ARSyN and MultiBaC were very similar according to the PCA. MultiBaC performance was slightly less powerful than ARSyN method since MultiBaC does not estimate the batch and interaction effects from the non-common omic, while ARSyN does. Thus, the estimation and correction of the unwanted variation is not the same and should be more accurate for ARSyN. Nonetheless, MultiBaC almost completely removed the batch effect. Finally, in our "real yeast multiomic dataset", differential expression together with clustering analysis proved that lab effect was removed while the effects of experimental factors were preserved in all the omics. Few genes changed their trend after correction but the comparison with previously published results showed that results after correction were more meaningful, reliable and concordant with such studies.

MultiBaC models the relationship between omic datasets with PLS and uses this to infer the batch-corrected data. PLS is a powerful predictive multivariate technique based on the linear combination of predictive variables. While the relationship between molecular layers measured by multiomic methods may or may not be linear, our results show that the linear approach is effective in modeling relationships for the purpose of batch correction. However, MultiBaC could be easily adapted to include, for example, kernel PLS [151, 152] to allow for non linear relationships. In any case, the method returns the Q^2 of the PLS model to provide control over the accuracy of the predicted batch correction.

In conclusion, MultiBaC is effective at removing non-biological noise from multiomic data collected at different studies, and makes these datasets comparable. We anticipate MultiBac will be a useful tool for the reutilisation of existing data for multiomic integration analyses and in facilitating experimental designs that involved the generation of multiple and diverse omic assays.

Chapter 4

Generating a multiomic dataset

[1] Nuño-Cabanes, C., Ugidos, M., Tarazona, S. et al. A multiomics dataset of heat-shock response in the yeast RNA binding protein Mip6. Sci Data 7, 69 (2020).

4.1 Introduction

Eukaryotic gene expression is a complex process in which genetic information is converted into functions that sustain living cells. Different cellular components are involved in this process, which perform a series of interconnected steps in different cellular compartments [153, 154]. One of the earlier steps consists of setting up the appropriate epigenetic modifications to allow the expression or repression of specific gene programs [155, 156]. These modifications take place mostly on DNA and histones, ensuring access to the proper transcriptional machinery. Methylation and acetylation are the most estudied histone marks as they have a higher impact on gene expression [157]. This specific set of modifications across the genome regulates the final synthesis of the mRNA [13]. Newly synthetized RNA molecules are extensively modified prior to their export to the cytoplasm, where they can be degraded by the mRNA decay machinery, stored in specific organelles or translated into proteins [18, 19]. Finally, the encoded protein products participate in numerous processes, including cellular metabolism where organic compounds are transformed and/or stored. A number of these compounds, such as Acetyl-CoA, glucose or methyl groups, participate, in turn, in chromatin modifications. Thus, by profiling chromatin modifications, steady-state mRNA and metabolites, we have a view of the whole process from genotype to phenotype.

In this chapter, the generation of a yeast multiomic dataset is explained. This dataset features three basic layers of the transcriptional circuit, measured in the same set of samples. These include one epigenetic modification (ChIP-seq) - H4K12ac, a mark for active promoters-, gene expression (RNA-seq) and targeted metabolomics (NMR quantification). Moreover, data were obtained for two different yeast strains, wild type (WT) and a $mip6\Delta$ mutant, in control and heat-shock induced conditions. Mip6 is an RNA-binding protein that participates in RNA export under stress [106] and consequently is informative of the contribution of post-transcriptional regulations to the adaptation of RNA levels to environmental changes.

To obtain the final useful molecular information, raw omic data require different computational steps depending on the technology [158]. Regarding RNA-seq and ChIP-seq data, sequencing output consists of a huge number of short reads that need to be mapped to a reference genome to get gene or transcript associated quantification. Before that, Quality Control (QC) checking is needed to ensure data reliability and get rid of residual sequencing adapter sequences. Once mapped, the gene expressions are given by the number of reads detected for each gene. In addition to gene-wise quantification, ChIP-seq data can also be processed as peaks, which are groups of mapped reads that represent genomic regions close to modified histones [159]. Sequencing data may have some potential biases that include gene length (longer genes accumulate more reads) and library size or sequencing depth (different total number of reads for different samples impairs direct sample comparisons). Removal or mitigation of these biases are needed to perform further analysis [160]. Many methods have been developed to correct these caveats [e.g, Reads Per Kilobase Million (RPKM) [49] and Trimmed Mean of M values (TMM) [35]].

Regarding NMR spectra raw data, between sample normalization is needed. Peaks are relative quantification of metabolites and are normalized by the total area of the spectra to make samples comparable between each other. However, within sample comparison of different metabolites is not possible since peak areas do not correlate with actual metabolite concentration. NMR metabolomics is a targeted assay which means that peak-metabolite association is made based on prior knowledge [36].

In conclusion, in this chapter the analysis and exploration of these data is performed on a single-omic basis as a preliminary step for the multi-omic integrative analysis explained in Chapter 5. Therefore, the work presented in this chapter addresses: quality control of omic data, normalization procedures, identification of differentially expressed/quantified features and functional enrichment analysis.

4.2 Experimental design

Figure 4.1 illustrates the experimental design of our dataset. A single culture (either for WT or $mip6\Delta$ strains) was grown at $30°$C until the exponential growth phase, and was then split across three flasks. One flask was maintained at $30°$C and labeled as time point 0. The other two flasks were incubated at $39°$C for 20 minutes and 120 minutes, respectively. These last two flasks capture the heat-shock response, while the $30°$C flask serves as a control representing non-stress condition. Then, for each of the flasks described above three aliquotes were extracted for RNA-seq, NMR metabolomics, and ChIP-seq analyses. Therefore, the three omics assays were performed on the same cell culture. The process described in Figure 4.1 was repeated 4 times to generate four biological replicates. Due to hardware limitations, these 4 replicates were generated and processed at two different time points which might lead to a batch effect that will be evaluated. Nevertheless, following properly experimental design guidelines, samples were randomly distributed between days so that the batch is not a confounding factor for any other covariate (Table 4.1). Note that the time variable for control samples ($30°$C 20 min.) was set as 0 minutes for subsequent time-series data analysis.

Table 4.1: Experimental design.

Strain	Temp (°C)	Time (min.)	Replicate	Day (Batch)	Sample ID

Strain	Temp (°C)	Time (min.)	Replicate	Day (Batch)	Sample ID
WT	30	0	1	1	wt.0.30.1
WT	39	20	1	1	wt.20.39.1
WT	39	120	1	1	wt.120.39.1
WT	30	0	2	2	wt.0.30.2
WT	39	20	2	2	wt.20.39.2
WT	39	120	2	2	wt.120.39.2
WT	30	0	3	1	wt.0.30.3
WT	39	20	3	1	wt.20.39.3
WT	39	120	3	1	wt.120.39.3
WT	30	0	4	2	wt.0.30.4
WT	39	20	4	2	wt.20.39.4
WT	39	120	4	2	wt.120.39.4
$mip6\Delta$	30	0	1	1	mip6.0.30.1
$mip6\Delta$	39	20	1	1	mip6. 20.39.1
$mip6\Delta$	39	120	1	1	mip6.120.39.1
$mip6\Delta$	30	0	2	2	mip6. 0.30.2
$mip6\Delta$	39	20	2	2	mip6. 20.39.2
$mip6\Delta$	39	120	2	2	mip6.120.39.2
$mip6\Delta$	30	0	3	1	mip6.0.30.3
$mip6\Delta$	39	20	3	1	mip6.20.39.3
$mip6\Delta$	39	120	3	1	mip6.120.39.3
$mip6\Delta$	30	0	4	2	mip6.0.30.4

Strain	Temp (°C)	Time (min.)	Replicate	Day (Batch)	Sample ID
$mip6\Delta$	39	20	4	2	mip6.20.39.4
$mip6\Delta$	39	120	4	2	mip6.120.39.4

4.3 Statistical methods

4.3.1 Sequencing data pre-processing

Removing low-count features

The removal of low-count features improves the result of statistical analyses. In RNA-seq, the expression of low-count features is noisier since read counts could have been assigned by chance [161, 162] causing background noise and between-sample variability. In this work, CPM (Counts per Million) method implemented in NOISeq package [51] was used. This method uses a transformation of expression data to perform the filtering, the counts per million (CPM). CPM for gene g in sample s is defined as:

$$CPM_g^s = 10^6 \frac{x_g^s}{\sum_g x_g^s} \tag{4.1}$$

where x is the number of raw counts. CPM takes into account sequencing depths (total number of counts) of individual samples to

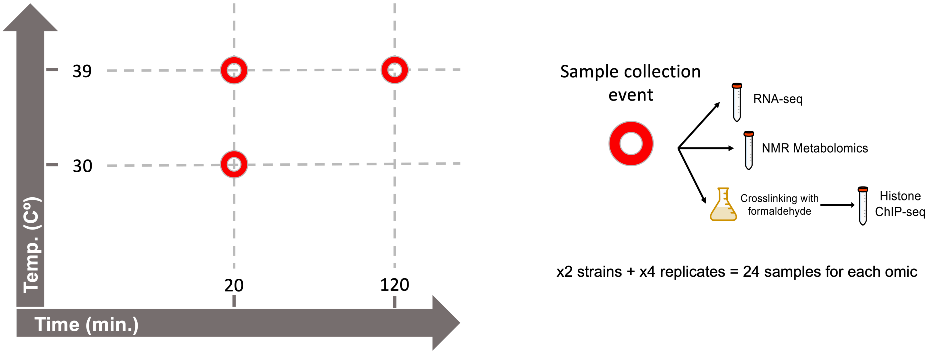


Figure 4.1: Experimental design and sample management. For each strain and replicate, a yeast culture flask was grown at $30°$C until the exponential phase, then split into three flasks, each of them receiving a different treatment. From the same treatment flask, aliquots were collected for RNA-seq, metabolomics and ChIP-seq.

avoid removing genes with relatively high expression in at least one experimental condition. The method needs a user-defined value for CPM (cpm) under which a feature is considered to have low counts. Let us consider S samples in a given condition, a gene g is removed if the sum of CPM values across all the samples in the same experimental condition is below the condition cutoff ($\sum_s CPM_g^s < cpm \times S$). For this work, default cpm value was used: $cpm = 1$.

Trimmed mean of M-values normalization

Trimmed mean of M-values (TMM) [46] normalization is used when systematic differences among samples are present in sequencing data. These systematic biases can often be due to differences in the library size (total number of counts) and the RNA composition (distribution of counts across features). The rationale behind this idea can be explained with the following hypothetical example. Let us consider two RNA sequencing experiments (samples), A and B, and two different sets of genes with the same number of elements: S_1 and S_2. The first set, S_1, is equally expressed in A and B, whereas S_2 is only expressed in sample A and not in sample B. If the library size (total number of reads/counts) is the same in both samples, the number of transcripts observed for an S_1 gene will be half as many in A as in B even though it is known that S_1 is equally expressed in both samples. In other words, the probability of observing transcripts for a given gene depends on its frequency and the total number of transcripts observed. Thus, it depends on the expression of the rest of genes. In conclusion, it depends on the RNA composition. This issue impairs further statistical analysis and therefore a proper correction of this bias is pivotal in high-throughput data analysis. Roughly, first the sample/observation that have the closest average expressions to mean of all samples is considered as reference sample, and all others are test samples. For each test sample, the scaling factor is calculated based on weighted mean of log ratios between the test and reference. The following framework is used to provide a more formal explanation for this normalization. Define Y_{gk} as the raw observed count for gene g in sample k, and N_k as the total number of counts for sample k. The normalization factor ($\log_2(TMM_k^{(r)})$) for sample k based on reference sample r is derived as follows:

$$\log_2(TMM_k^{(r)}) = \frac{\sum_{G^*} w_{gk}^r M_{gk}^r}{\sum_{G^*} w_{gk}^r} \tag{4.2}$$

The cases where $Y_{gk} = 0$ or $Y_{gk} = 0$ are trimmed in advance of this calculation since log-fold-changes cannot be calculated and G^* represents the set of genes not trimmed. M_{gk}^r represents the log-ratios between test (k) and reference (r) samples and it is defined as:

$$M_{gk}^r = \frac{(\log_2 \frac{Y_{gk}}{N_k})}{\log_2(\frac{Y_{gr}}{N_r})} \tag{4.3}$$

The weights, w_{gk}^r, are obtained by the following expression:

$$w_{gk}^r = \frac{N_k - Y_{gk}}{N_k Y_{gk}} + \frac{N_r - Y_{gr}}{N_r Y_{gr}} \tag{4.4}$$

TMM normalization is a simple and effective method for estimating relative RNA production levels from RNA-seq data. However, TMM normalization assume that the majority of genes are not differentially expressed.

Batch effect correction

ARSyN (ASCA Removal of Systematic Noise) [54] was used for correcting the batch effect. This approach is based on the ANOVA-

Simultaneous Component Analysis (ASCA) framework and has been already presented in this document (Chapter 3).

4.3.2 Differential expression/quantification analysis

In this analysis, Time was encoded as a categorical variable and both Strains were analyzed separately. The aim of this analysis was to produce the required information for the method further explained in the next chapter. Therefore, two comparisons were performed for each Strain: i) $39°$C 20 min. (t-20) vs $30°$C 20 min. (t-0), and ii) $39°$C 120 min. (t-120) vs $39°$C 20 min. In this chapter, comparison (i) is named as First Transition (FT) and comparison (ii) is named as Second Transition (ST). Among the variety of available methods for performing differential expression, limma R package [53] was used in this analysis as it showed overall robust results in multiple scenarios when compared with other approaches [60] and it is one of the most extended methods in the bioinformatics community. Limma (Linear models for microarrays) was originally developed for the analysis of microarray data and hence linear models are the basis of limma modeling. Following the explanation in Smyth et al., 2004 [163], limma fits a single linear model for each gene (or omic feature) where gene expression is the response vector $\mathbf{y}_g = (y_{g1}, \ldots, y_{gs})^T$ (S being the number of samples). In a simplistic way, it is assumed that: i) $E(\mathbf{y}_g) \approx \mathbf{X}\beta_g$ and ii) $var(\mathbf{y}_g) \approx \sigma_g^2$, where **X** is the design matrix. Given the large number of gene-wise linear model fits, limma takes advantage of the parallel structure to estimate the unknown parameters β_g and σ_g^2 as the same model is fitted to every gene. This is done by assuming prior distributions for these set of parameters. Prior distribution for σ_g^2 is assumed based on a prior estimator s_0^2 with d_0 degrees of freedom:

$$\frac{1}{\sigma_g^2} \sim \frac{1}{d_0 s_0^2}\chi_{d_0}^2 \tag{4.5}$$

This describes how the variances are expected to vary across genes. Then, the expected distribution of regression coefficients is defined as:

$$\beta_{gj}|\sigma_g^2 \sim N(0, \sigma_g^2) \tag{4.6}$$

where j represents a given experimental condition (model covariate). This equations describe a conjugate prior. Under this hierarchical model, the posterior mean of σ_g^2 given the actual observed residual sample variance (s_g^2) is:

$$\tilde{s}_g^2 = E(\sigma_g^2|s_g^2) = \frac{d_0 s_0^2 + d_g s_g^2}{d_0 + d_g} \tag{4.7}$$

Basically, the observed variances shrinkage towards the prior values redefine the t-statistic. This new moderated t-statistic is defined as:

$$\tilde{t_{gj}} \approx \frac{\hat{\beta_{gj}}}{\tilde{s}_g} \tag{4.8}$$

This statistic represents a hybrid frequentist/Bayesian approach in which the posterior variances substitute the usual sample variance in the classical t-statistic. Both, d_0 and s_0^2 are estimated from the data as described in Smyth et al., 2004 [163]. The fact that the method shares information across genes to estimate gene-wise residual variances, makes limma specially powerful for datasets with small sample sizes which unfortunately is an extended scenario in the computational biology field.

Additionally and before applying limma, sequencing data need to be transformed to approxymately follow a normal distribution. This step is done with the so called $voom$ transformation [58], implemented also in the limma R package. Briefly, the aim of this transformation is to correct the mean-variance relationship present in count data. First, counts are transformed as log-cpm values ($log_2(CPM); CPM$ defined in (eq.4.4)). Next, the function estimates the mean-variance trend for log-counts (LOWESS fit) and assigns a weight to each observation based on its predicted variance which are then used in the linear modelling process to adjust for heteroscedasticity.

4.3.3 Time-series data modeling

To obtain the genes/metabolites with significant differential time-profile between Strains, we used maSigPro [59, 164, 165]. maSigPro is a two-regression step approach. First, it adjusts a gene-wise regression model with all the defined covariates (Time and Strain in our case) to identify differentially expressed genes. maSigPro uses polynomial regression to model response variables (e.g. gene expression values or metabolite quantifications) and applies least-squares to estimate the parameters. Considering our time-course series dataset with $T = 3$ time points and $S = 2$ experimental groups ($i = 1, 2$ and $j = 1, 2, 3$), the model of response y_{ij} at time t_j is

$$y_{ij} = \beta_0 + \beta_1 t_j + \beta_2 t_j^2 + \beta_3 z_i + \beta_4 t_j z_i + \beta_5 t_j^2 z_i + \epsilon_{ij} \tag{4.9}$$

where z_i represents the dummy variable for the experimental group, i.e. $z \in (0, 1)$. To find genes with statistically significant changes, gene-wise ANOVA is performed testing the null hypothesis that all coefficients (except β_0) are equal to zero versus the alternative hypothesis where at least one coefficient is different form zero:

$$\begin{aligned} H_0 &: \beta_1 = \beta_2 = \beta_3 = \beta 4 = \beta_5 = 0 \\ H_1 &: \exists i / \beta_i \neq 0, (i = 1, 2, 3, 4, 5) \end{aligned} \tag{4.10}$$

maSigPro supports Linear Models (LMs) and Generalized Linear Models (GLMs). Therefore, sequencing data as RNA-seq experiments can be modelled with maSigPro without any count data transformation. When analysing sequencing data, maSigPro applies GLMs with Negative Binomial (NB) distribution:

$$Y_i \sim NB(\mu_i, \theta),\ where\ E(Y_i) = \mu_i\ and\ var(Y_i) = \mu_i + \frac{\mu_i^2}{\theta} \tag{4.11}$$

The authors state that maSigPro results do not change much by using different values of θ and hence it is recommended to use the default value ($\theta = 10$). However, θ can be estimated form data using available software (edgeR, [35]).

In a second step, a variable selection strategy (stepwise regression) is performed to find the conditions for which genes shows statistically significant profiles changes.

4.3.4 Multiple testing correction

In differential expression/quantification analysis, a single model is fitted for each feature (e.g., gene, metabolite, etc.) and therefore the number of statistical tests is considerably large. The repeated application of a test, for a given level of significance, may lead to a large number of rejections of the null hypothesis even though no real differences exist. False Discovery Rate (FDR) is the most common measure of the error associated to hypothesis testing. FDR is defined as the expected proportion of null hypothesis that are true among the ones declared as significant. Thus, considering R as the total number of significant tests and V as the number significant tests when the null hypothesis is true, FDR is defined as: $FDR = E(\frac{V}{R})$, where $E()$ represents the expected value or statistical expectation. One of the most extended approaches to control FDR is the Benjamini-Hochberj (BH) correction [166]. Let us consider a series of independent null hypothesis $H_0^1, \dots, H_0^m$, from which an equal number of p-values have been obtained $p_1, \dots, p_m$. After ordering p-values by $p_1 \leq, \dots, \leq p_m$, we obtain:

$$i^* = max(i : p_i \leq \frac{i}{m}\alpha) \tag{4.12}$$

and we reject H_0^i for $i = 1,\ dots, i^*$. FDR parameter, α, is usually set as 0.05. Adjusted p-values p^* are obtained as $p_i^* = min\frac{m}{i}p_i, 1$.

4.3.5 Multiomic integration

Two types of multiomic data integration were utilized for this chapter: pathway enrichment analysis (conceptual multiomic integration) and Multi-Block PLS (statistical multiomic integration).

Pathway enrichment analysis (PEA)

PaintOmics web tool [http://www.paintomics.org/; [167]] was used for PEA. PaintOmics uses the pathways from KEGG database [168]. The tool identifies the subset of features (genes, proteins or metabolites) that participate in a particular pathway for the input. Then, it evaluates the fraction of those biological features that overlaps with the significant set of features provided. Thus, PEA is performed after differential expression/quantification analysis. Finally, PaintOmics computes the significance of the overlap using Fisher's exact test [169].

To combine the information across different omic data types, PaintOmics applies the Fisher's combined probability test [61], which allows the results from several independent tests for similar null hypotheses to be combined. Thus, this method combines the p-values for the test of each omic into one test statistic (X) using the formula:

$$X = -2\sum_{i=1}^{k} log(p_i) \tag{4.13}$$

where k is the number of tests being combined. X follows a χ^2 distribution with $2k$ degrees of freedom. PEA from PaintOmics was used to find those gene sets that have a different behavior between strains during heat stress.

Multi-Block PLS (MB-PLS)

mixOmics R package [81] was used for MB-PLS analysis [108]. MB-PLS is an extension of the PLS framework. In MB-PLS, the PLS-components of each group (omic type) are constraint to be built based on the same loading vectors in X and Y. These global loading vectors thus allow the samples from each group or study to be projected in the same common space spanned by the PLS-components. For each dimension $h = 1, \ldots, H$, MB-PLS seeks to maximize:

$$\max_{||\mathbf{a_h}||=||\mathbf{b_h}||=1} \sum_{j=1}^{J} n_j cov(\mathbf{X_h^{(j)}}\mathbf{a_h}, \mathbf{Y_h^{(j)}}\mathbf{b_h}) \tag{4.14}$$

where $\mathbf{a_h}$ and $\mathbf{b_h}$ are the global loadings vectors common to all groups (omic types, J), and n_j is the number of individuals in each group. Group-specific PLS-components can also be obtained as $\mathbf{t_h^{(j)}} = \mathbf{X_h^{(j)}}\mathbf{a_h}$ and $\mathbf{u_h^{(j)}} = \mathbf{Y_h^{(j)}}\mathbf{b_h}$. The global loadings vectors ($\mathbf{a_h}$, $\mathbf{b_h}$) and global components can be used to assess overall classification accuracy, whereas the group-specific loadings and components can be used to analyze individual block contributions to the model. In this work, MB-PLS was performed to evaluate which metabolic changes (Y) are driven jointly by RNA-seq ($\mathbf{X^{(1)}}$) and ChIP-seq data ($\mathbf{X^{(2)}}$).

4.4 Data acquisition and preprocessing

4.4.1 RNA-seq

RNA extraction protocol is detailed in the original manuscript [113]. Sequencing was done with Illumina using the TruSeq protocol. Between 5060 million reads of 100 bp paired data were obtained from each sample. Raw sequencing data quality was checked by fastQC and good overall quality (Figure 4.2.a) was observed in all cases. No adapter trimming was deemed necessary. Reads were mapped to the yeast saccer3 genome with Tophat2 [137] and genes were quantified with HTSEQ [34], intersection-option. The NOISeq [51] R package was used to perform the quality control of count data. We observed most of reads mapped onto protein-coding genes (>80%), as expected (Figure 4.2.b). Low count filtering was applied with the NOISeq cpm method (with cpm = 1). Cpm stands for counts per million and it represents theoretical gene counts considering that the sum of all the gene counts for a given sample is 1 million. NOISeq cpm methods removes features with an average cpm per condition below a certain threshold (1 in this case) in all conditions. Systematic differences among samples were detected and hence raw counts were normalized via TMM method [33]. Principal Component Analysis (PCA) indicated a slight batch effect for the day of culture growth (Figure 4.2.c [left panel]) that was removed by ARSyN function from NOISeq package as well [51, 54] (Figure 4.2.c [right panel]). In total, we obtained gene expression values for 6,379 genes.

4.4.2 Metabolomics

Metabolomics measurements were performed on an NMR platform as described in Palomino-Schätzlein ([36]). Signal peaks of spectra were normalized considering that the sum of peak areas across all metabolites was constant for every sample, and values for each metabolite were given as a fraction of the total area. This targeted NMR technique implies metabolomic values are not absolute, i.e., metabolites cannot be compared within a given sample. A total of 45 compounds were detected, that included 5 sugars, 17 amino-acids, 4 alcohols, 3 vitamin-

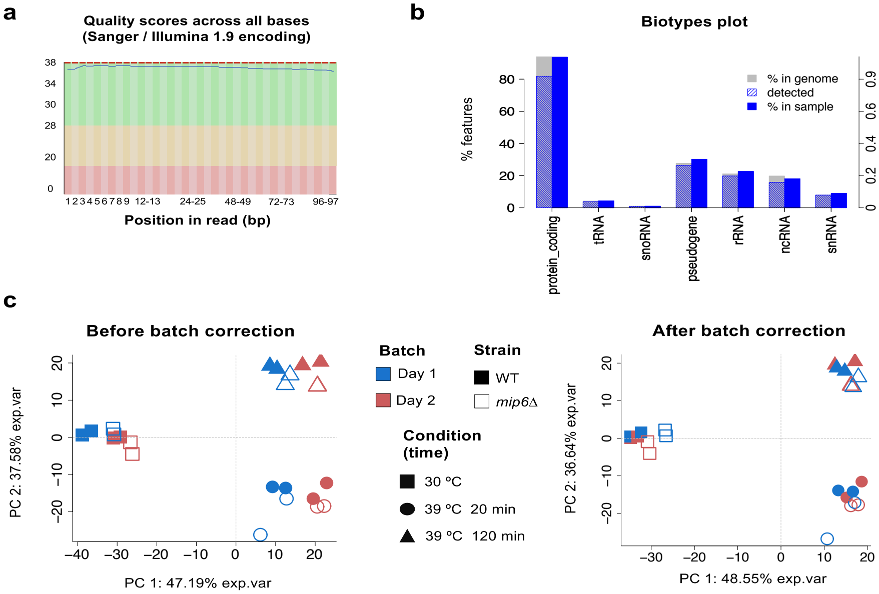


Figure 4.2: RNA-seq data preprocessing. (a) Example of base quality scores across all reads obtained by fastQC analysis, showing uniform read quality. (b) Biotype plot of NOISeq package, which indicates that the vast majority of detected features are protein-coding genes. (c) Batch effect correction. PCA score plots for the first two principal components are represented. The left panel shows raw data where a day-of-culture batch effect is observed. The right plot shows the corrected data where this batch effect has been removed .

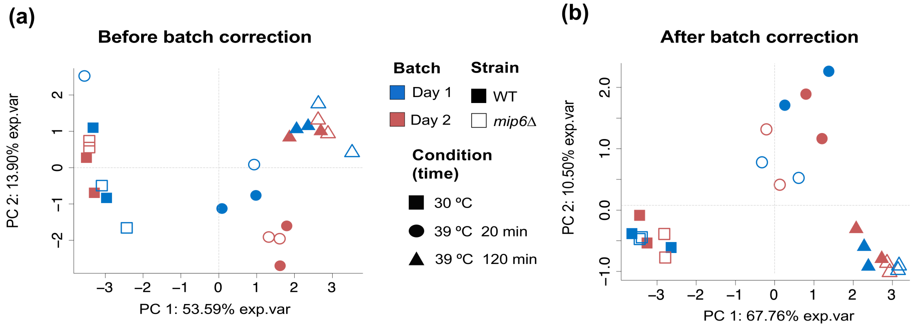


Figure 4.3: Batch effect correction of metabolomics data. PCA score plots for the two first principal components are represented. (a) Raw data shows a slight day of culture batch effect for the 20 min $39°$C condition. (b) Batch effect corrected data .

derivated compounds, 5 carboxylic acids, and other compounds (CMP, NAD, Glutathione, ATP and GMP), plus 3 unidentified metabolites. Raw data were log2 transformed and compounds with non-positive measure across all samples were removed, as they were considered to be below the reliable limit of detection. PCA analysis indicated a small batch effect (Figure 4.3.a), that was removed by ARSyN method (Figure 4.3.b).

4.4.3 Histone modifications

A comprehensive protocol for ChIP-seq data generation is detailed in the original publication [113]. Two ChIP-seq data files were obtained for each sample: H4 and H4K12ac. H4 files contain the reads after purification of total H4 histone and H4K12ac files contain the data associated to acetylation of Lysine 12 of H4 histones. Raw sequencing data quality was checked by fastQC and good overall quality (Figure 4.4.a) was observed in all cases. In this case, trimming of Illumina adapters was needed and performed using Cutadapt software [170]. Then, reads were mapped to the yeast saccer3 genome with Bowtie2 [171]. Macs2 software [172] was used to call Histone 4 acetylation peaks on the H4K12ac samples alone. Next, a consensus file was generated by merging peaks across all samples using the merge command from bedtools software [173] with default parameters. These consensus regions were used to map back reads of all samples, including H4 samples. Peaks were quantified with HTSEQ [34], intersection-option. NOISeq [51] R package was used to perform a quality control of count data. Moreover, coverage per base was obtained for both, H4 and H4K12ac samples, using the genomecov command from bedtools [173]. Batch effect was also checked but no correction was needed (Figure 4.4.b). To obtain the final H4K12ac signal, H4K12ac samples are divided by H4 values as they represent the total amount of H4 histone. Therefore, H4K12ac signal is treated as a relative quantification of acetylated histones against total amount. Finally, to use ChIP-seq data information as a gene-based omic, RGmatch [134] was used considering regions around gene TSS (Transcription Start Site) ± 200 base pairs.

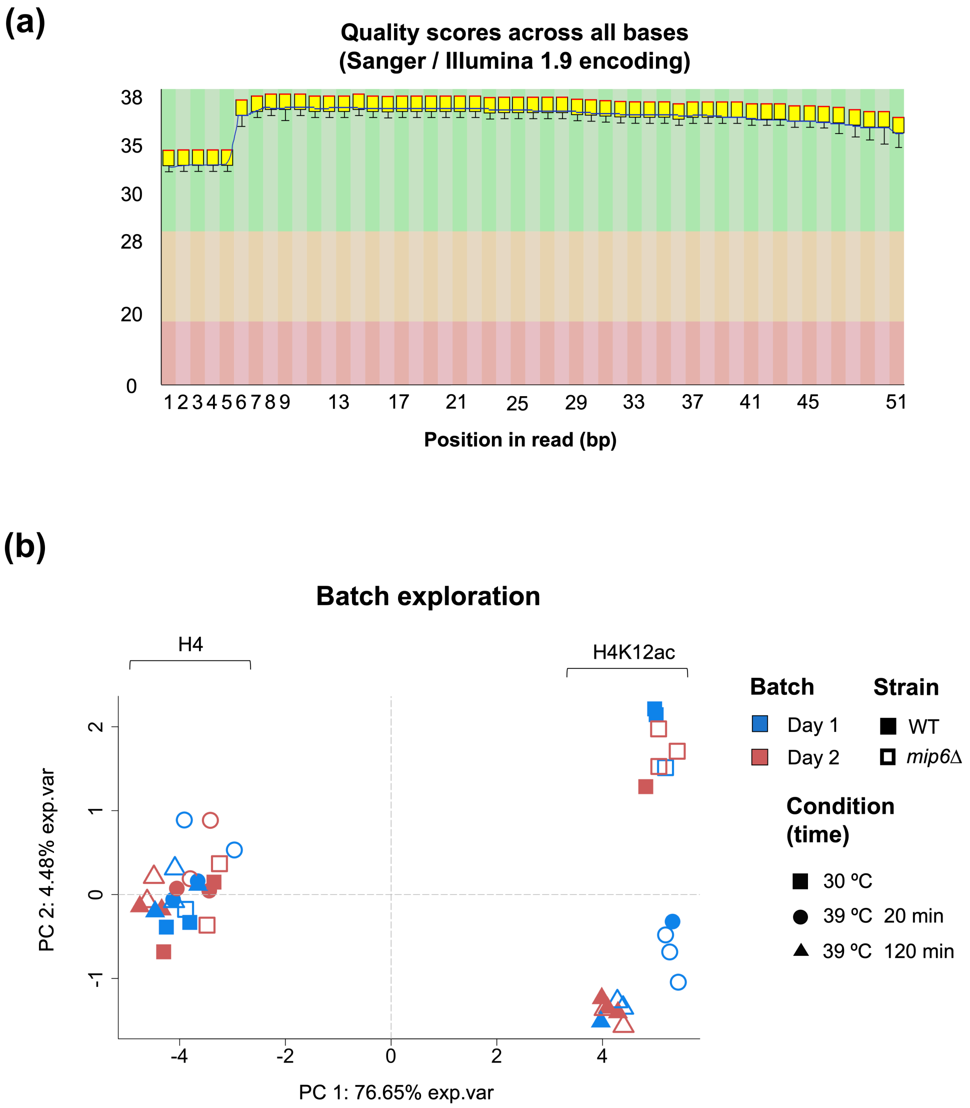


Figure 4.4: ChIP-seq data preprocessing. (a) Example of base quality scores across all reads obtained by fastQC analysis. (b) PCA of H4 and H4K12ac data. The first PC indicates the type of ChIP-seq assay, while the second PC reflects the heat treatment in the H4K12ac samples. No batch effect is observed .

4.5 Technical validation

In order to assess data replicability, pairwise scatter plots were obtained for RNA-seq data (Figure 4.5.a), metabolomics data (Figure 4.5.b) and ChIP-seq data (Figure 4.5.c, d). Only WT strain replicates are shown as $mip6\Delta$ strain data behaved similarly. Replicates were highly and equally correlated with each other, and no experimental outliers were detected.

4.6 Omic-wise differential expression/quantification analysis

For each omic, two types of differential expression analysis were performed (see Methods section). First, within-strain differential expression analysis was performed using limma [53]. Figure 4.6.a and Figure 4.7.a and b show the results of this analysis for RNA-seq, Metabolomics and ChIP-seq data, respectively. The number of differentially expressed features (DEF) is represented for each strain-wise comparison: FT (First Transition, 39○C 20 min. vs 30○C 20 min.) and ST (Second Transition, 39○C 120 min. vs 39○C 20 min.). The total amount of DEF is represented as an horizontal barplot with gray bars at the bottom-left part of the panels. Intersections between different comparisons (yellow dots) are represented as blue bars in the center of the panels. Regarding gene expression (Figure 4.6.a), around half of the changes that occur immediately after heat-shock (FT) are common between strains

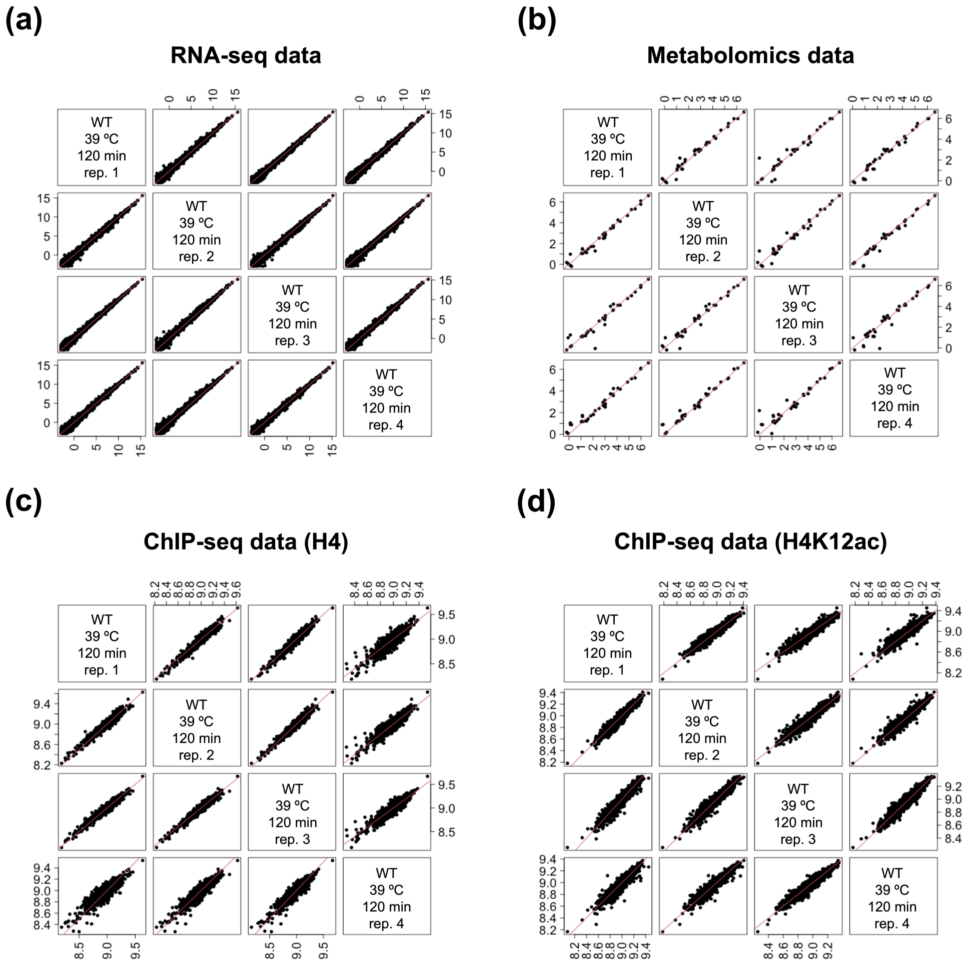


Figure 4.5: Replicability of processed data. Wild type 39°C 120 minutes sample is selected as an example. Log2 transformed data are shown. (a) RNA-seq. (b) Metabolomics. (c) ChIP-seq (H4). (d) ChIP-seq (H4K12ac). Red diagonal line indicates perfect correlation between samples .

(380/815 for WT and 380/765 for $mip6\Delta$). This indicates that an important part of the heat-shock response is common to both strains (also shown in Figure 4.6). However, the number of non-common changes is also relevant for both FT and ST and therefore there are big differences between strains in terms of heat-shock adaptation. Attending metabolic changes (Figure 4.8.a), only a few metabolites are significant and most of them are common between strains and between transitions which could suggest adaptation to an initial status. Interestingly, ST comparisons present a higher number of significant changes that could indicate a slower metabolic response compared to changes in gene expression. Lastly, results from ChIP-seq data (Figure 4.8.b) show a big difference between strains in terms of the number of significant changes. $mip6\Delta$ mutant seems less capable to modify the chromatin compared to the WT. In addition, the number of common changes between strains is much lower compared to metabolomics and RNA-seq analyses.

In order to analyze between-strain differences, maSigPro [59] was used to extract all DEF between strains considering baseline status and changes in time. Figures 4.6.b and c represent the modeling of two different genes using maSigPro. These two panels correspond to the analysis of gene expression data. Figure 4.6.b shows a gene expression profile where the interaction strain x time is significant. On the other hand, in Figure 4.6.c the interaction strain x time^2 is significant but not the coefficient for strain x time. Therefore, all features that have one significant strain-associated coefficient in maSigPro models (see Equation 4.9), were selected as DEF. This analysis was also performed for metabolomics and ChIP-seq data (results not shown).

4.7 Multiomic data integration

4.7.1 PEA identifies significant differences between strains supported by all omics

Significant DEF returned by maSigPro were uploaded to PaintOmics for PEA. Two different analyses were run: i) integration of RNA-seq and metabolomics data, ii) integration of RNA-seq, metabolomics and ChIP-seq data. Significant gene sets or pathways are shown in Table 4.2 and Table 4.3 for runs (i) and (ii), respectively. Only significant pathways (combined p-value < 0.05) are shown.

Table 4.2: PEA analysis with all gene expression and metabolomics data.

Pathway	**Combined p-value**
Protein processing in endoplasmic reticulum	0.000
Proteasome	0.000
Starch and sucrose metabolism	0.001
Glycolysis / Gluconeogenesis	0.001
Biosynthesis of secondary metabolites	0.002
Pentose phosphate pathway	0.005
Terpenoid backbone biosynthesis	0.012

Base excision repair	0.012
Homologous recombination	0.016
Longevity regulating pathway - multiple species	0.016
Carbon metabolism	0.016
Sulfur metabolism	0.018
Ether lipid metabolism	0.025
Glycerophospholipid metabolism	0.028
DNA replication	0.045

Table 4.3: PEA analysis with all omic data

Pathway	**Combined p-value**
Protein processing in endoplasmic reticulum	0.000
Proteasome	0.003
Glycolysis / Gluconeogenesis	0.003
Starch and sucrose metabolism	0.005
Biosynthesis of secondary metabolites	0.006
Terpenoid backbone biosynthesis	0.019
Ether lipid metabolism	0.025
Carbon metabolism	0.030
Pentose phosphate pathway	0.034

Regarding analysis (i), central carbon metabolism routes (e.g. Glycolysis, Pentose phosphate) and carbon metabolism in general are signifi-

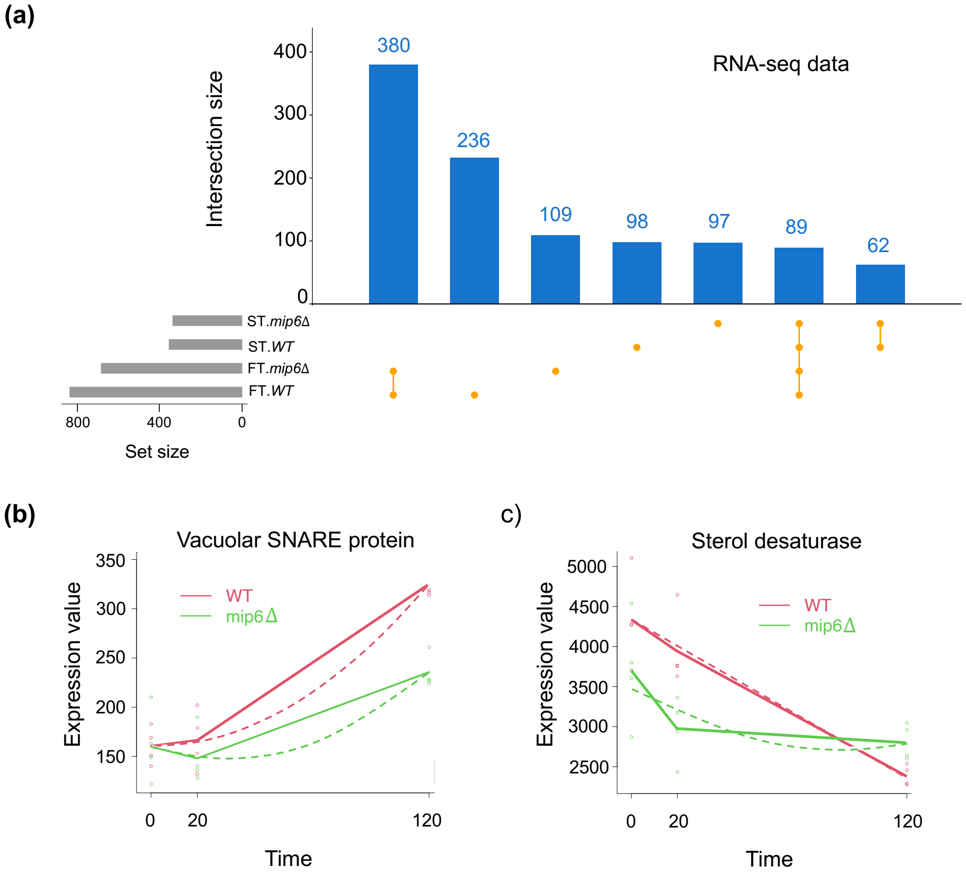


Figure 4.6: Differential expression analysis of RNA-seq data. a) Strain-wise comparisons using limma. FT (First Transition, 39°C 20 min. vs 30°C 20 min.) and ST (Second Transition, 39°C 120 min. vs 39°C 20 min.). The total amount of DEF are represented as an horizontal barplot with gray bars at the bottom-left part of the panels. Intersections between different comparisons (yellow dots) are represented as blue bars in the center of the panels. b) and c) maSigPro model two randomly selected genes. Dots and straight lines represent observed data and average value across time points. Dashed lines represent maSigPro models. WT is shown in red and $mip6\Delta$ in green.

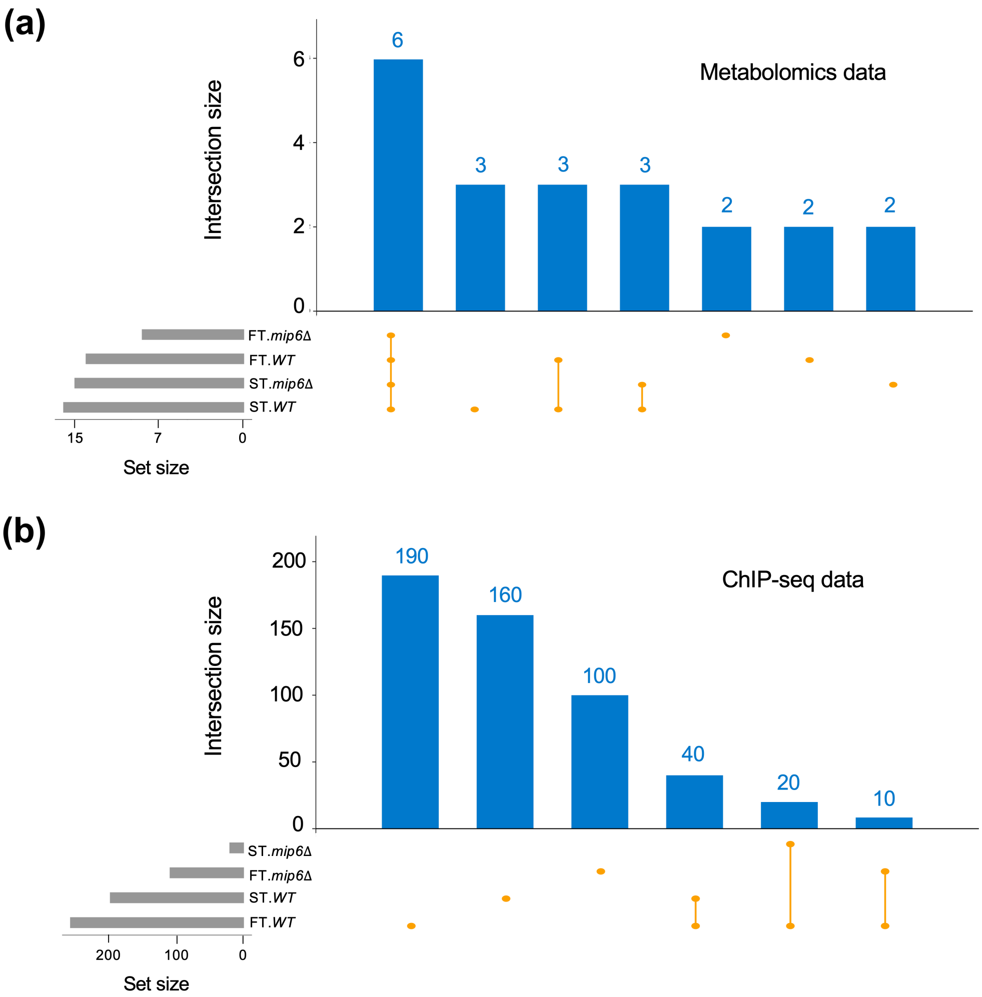


Figure 4.7: Differential feature analysis of metabolomics (a) and ChIP-seq data (b). Strain-wise comparisons using limma. FT (First Transition, 39°C 20 min. vs 30°C 20 min.) and ST (Second Transition, 39°C 120 min. vs 39°C 20 min.). The total amount of DEF are represented as an horizontal barplot with gray bars at the bottom-left part of the panels. Intersections between different comparisons (yellow dots) are represented as blue bars in the center of the panels.

cant. All these pathways are more active in the $mip6\Delta$ mutant during heat-shock treatment. Additionally, DNA replication is also more active in the mutant. When ChIP-seq data is incorporated (ii), the same central carbon metabolism pathways stay as significant although a considerable lower number of gene sets are significant.

4.7.2 Multi-Block PLS finds consistent changes across omics

Compared to PEA, with MB-PLS we aimed to find those metabolic changes supported by RNA-seq and ChIP-seq data. Therefore, in this analysis, metabolomics data is the response matrix (Y) while the other two omics are the explanatory matrices (X_1 and X_2). Figure 4.8 shows the latent space of the MB-PLS model for every omic type. All omic data types show similar results. Component 1 separates baseline status from heat-shock time points (specially 120 min.) and component 2 separates 39°C 20 min. and 39°C 120 min. samples. Thus, time or heat-shock effect is stronger than strain differences.

Lastly, component 3 separates strains (Figure 4.8.d). Therefore, this analysis shows that experimental conditions (strain and time after heat stress) explain data variability of the three omic data types. Further analyses on the relationship among omic data types and the differences between both strains will be adressed in the next chapter.

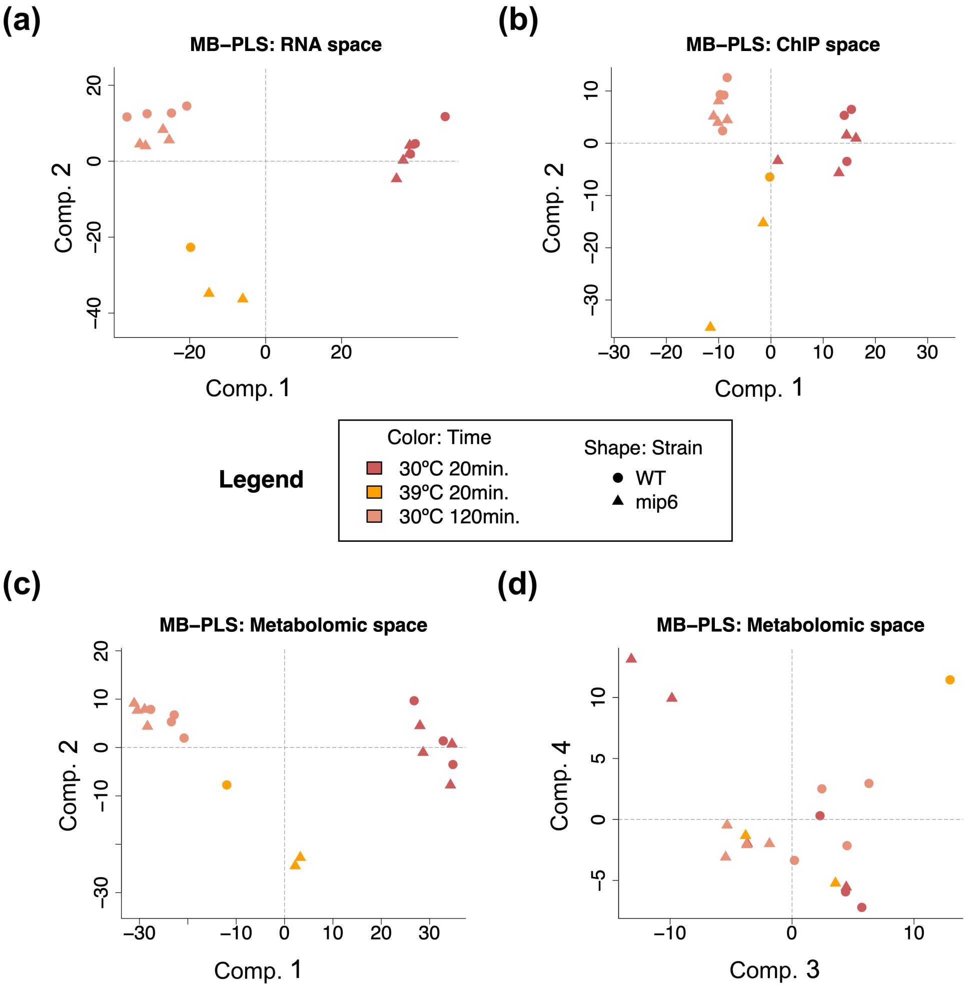


Figure 4.8: Score plot of MB-PLS analysis. (a) RNA-seq space for PLS Components 1 and 2. (b) ChIP-seq space for components 1 and 2. (c) and (d) Metabolomics (response) for components 1, 2, 3 and 4.

4.8 Discussion

Gene transcription and transcriptional regulation are among the most studied mechanisms in molecular biology. Histone acetylation is generally associated to active transcription of genes around the modified histones. However, ChIP-seq signal and RNA-seq data are not always 100% concordant as many post-trancriptional regulation processes also take place in living cells. Histone modifications require cellular compounds that are in turn produced inside the cell through metabolic reactions. Thus, to have a complete understanding of biological process regulation, different layers of information are needed. The data we are releasing is a robust multiomic dataset. It contains three layers that goes from genotype, ChIP-seq data, to phenotype (metabolomics) through transcriptome (RNA-seq). This represents an unique opportunity to study RNA-metabolism in yeast and to assist the development of multiomic integration tools. Due to sample management limitations, samples were generated on two different days which led to a batch effect. Nonetheless, samples were randomized and thus the batch effect can be modelled and in turn removed from data.

The quality of sequencing and NMR assays were optimal and samples show high replicability. PCA analyses after batch effect correction separate samples according to time in the first place. This means that the heat stress is the predominant effect which represent most of the variability of the data. These data are reliable as they reflect the state of the art knowledge about heat response in yeast. Global transcriptional shut down and trehalose production increase are two main impact of heat-shock in yeast and these processes encompasses the three layer of information studied.

Within-strain differential expression/quantification analysis confirmed that an important fraction of heat-shock response was common in both strains. However, we have also found differences between strains at every omic data type and thus they also responded differently to heat stress in some extent. Contrary to RNA-seq data analysis, in the analysis of metabolomics data the number of significant changes in the first transition was lower compared to second transition. This could indicate that transcriptomic changes occur faster and precede metabolomic changes. Finally, the analysis with ChIP-seq data showed that the number of significant changes in WT samples was higher compared to mutant cells. In other words, $mip6\Delta$ mutant was less efficient in modifying histone acetylation status after heat stress.

Delving into between-strains differences, Pathway Enrichment Analysis (PEA) supported by three omic data types revealed that part of carbon metabolism (Glycolysis, Pentose phosphate pathway, Starch and sucrose metabolism and Biosynthesis of secondary metabolites) was overactivated in the $mip6\Delta$ mutant compared to WT samples during heat stress. Finally, Multiblock PLS (MB-PLS) analysis confirmed that metabolic changes were in the same direction as transcriptomic and histone acetylation signals.

All of the above taken together, confirmed the usefulness of these data for multiomic integration purposes, either method development or biological insight discovering. Yeast strains included in this study showed signals of differential behavior regarding heat-shock adaptation. However, a key part of heat stress response was common for both strains. Data-driven multiomic integration approaches have been useful for finding between-strain differences although they lack of mechanistic interpretation. The fact that the heat-shock effect has been shown to be stronger than the strain effect makes the development of novel multiomics integration approaches challenging, as a high level of sensitivity will be required to uncover differences between yeast strains.

Chapter 5

Development of a model-driven multiomic integration approach

[1] Manuel Ugidos, Carme Nuño-Cabanes, Sonia Tarazona, Alberto Ferrer, Lars K. Nielsen, Susana Rodríguez-Navarro, Igor Marín de Mas and Ana Conesa. MAMBA: a model-driven, constraint-based multiomic integration method. BioRxiv, 2022.10.09.511458.

5.1 Introduction

The availability of sequenced genomes, together with the annotation of genes and their functions has facilitated the reconstruction of high-quality genome-wide metabolic networks, the so-called, genome-scale metabolic models or GEMs [174]. These metabolic networks gather all the known metabolic reactions identified in an organisms genome and incorporate information about stoichiometry, thermodynamics and optionally the associations between the reactions and proteins/genes involved. Constraints-based modeling (CBM) is a family of mathematical methods suitable for the analysis of these large metabolic networks. When CBM is applied to GEMs, the fluxes through metabolic reactions represent the model variables to be estimated. The computation of each variable in CBMs is constrained by a minimum and maximum range of values. CBMs calculate flux distribution through the metabolic network that satisfies two fundamental types of constraints [9] i) steady-state mass-balance, which sets the total production and consumption rates for each metabolite to be equal; ii) capacity, i.e., upper and lower bounds for fluxes can be imposed. One of the most widely used CBMs is Flux Balance Analysis (FBA). This method describes the phenotype of an organism making use of an objective function (OF) that needs to be optimized (i.e. biomass maximization) [175]. The OF, together with the stoichiometric and thermodynamic parameters -embedded in the metabolic model- and the imposed constraints are formalized as numerical matrices that can be solved by a number of mathematical optimization algorithms and software developed for this purpose to define tissue/organism-specific metabolic network flux profile [93].

In order to improve the metabolic network characterization by CBM methods, additional biological data can be included into GEMs. One of the first extended CBM method was the GIMME algorithm [95], that proposed the incorporation of gene expression data into the FBA model. GIMME connects a given metabolic reaction to the genes encoding the enzymes that carry out the reaction. To determine the output of reaction fluxes, GIMME minimizes the usage of reactions where lowly-expressed genes participate, while keeping the OF above a certain value. Reactions and genes are connected by a set of Boolean rules that associate reactions to the expression state of the involved genes using a binary representation. These rules are known as gene-protein-reaction associations or GPRs and they indicate the collection of proteins (isozymes, enzymatic subunits, etc.) required for the reaction to carry flux. To determine the expression state of genes, GIMME and other CBMs require a set of user-supplied expression thresholds for classifying genes, and in turn reactions, generally as on or off [96], although more states can also be used. These arbitrary thresholds have a strong impact in the output of the method and hence must be carefully selected, which is an important caveat of such CBMs. Notably, other approaches that do not require user-supplied expression threshold values have also been developed, e.g. MADE [98]. MADE relies on expression data from two or more experimental conditions and uses the results of a differential expression analysis to determine gene states across conditions. The inclusion of transcriptomics data into CBMs has been successfully used for predicting associations between silencing or expression of genes and the metabolic capabilities of an organism [174, 176]. These approaches rely on the proven existence of a correlation between transcript levels and reaction activity [177] and it has been widely demonstrated that GEMs become more powerful by integrating additional molecular information [178].

Metabolomic data have also been included into CBMs to improve metabolic network characterization [66, 99, 100, 179]. Metabolomics are typically integrated into GEM reconstruction analyses as a set of capacity constraints that limit the flux through a given reaction/s. To this aim, metabolomic data from experiments using isotopically labeled subtrates or absolute quantification from label-free experiments can be used [28]. However, the inclusion of semi-quantitative (relative quantification) metabolomics into CBM still remains challenging. Furthermore, there is growing evidence of a significant contribution of epigenomics features such as chromatin modifications to the regulation of the metabolism and vice versa [101, 103, 105]. However, the incorporation of chromatin modification data into the CBM framework has not been yet reported. Finally, to the best of our knowledge, methods enabling the integration of non-quantitative metabolomics and transcriptomics data combining different time points and conditions are still not available. All together, these considerations imply that despite the power of CBM for modelling metabolic fluxes, there is a significant number of applications for which these methods have not been adapted yet. This represents a gap in our understanding of metabolic regulation at large, and a missed opportunity for the analysis of a wealth of multi-omics data that include metabolomics measurements together with other molecular layers and/or a variety of experimental designs.

In this chapter we present MAMBA (Metabolic Adjustment via Multiomic Block Aggregation), a constraint-based genome-scale metabolic reconstruction model that integrates gene-centric omics measurements such as gene expression or histone modifications without requiring arbitrary expression thresholds, and semi-quantitative metabolomic data. Any experimental design can be modeled with MAMBA. When analyzing time-series data, MAMBA considers the dynamics of the system by including all time-points in the same model. MAMBA has been tested with a multi-omic time-series yeast dataset consisting of two strains, $mip6\Delta$ mutant and Wild Type (WT) subject to heat-shock treatment. Samples were obtained at three different time points: baseline, 20 min. and 120 min., after heat-shock and profiled for metabolomics, RNA-seq and ChIP-seq. MAMBA showed a better performance than other CBMs to predict metabolite changes, found key differences between strains regarding dynamic adaptation to heat stress and revealed differences between the transcriptional and chromatin control of metabolic fluxes. MAMBA was implemented in MATLAB and it is freely available at https://github.com/ConesaLab/MAMBA.git.

5.2 Data and computational details

In this work we have used a multiomic dataset from yeast previously generated in our lab and deeply explained in the previous chapter and detailed at Nuno-Cabanes et al., 2020 [113]. Briefly, this dataset consists of gene expression RNA-seq, NMR targeted metabolomic data and H4K12ac ChIP-seq data generated from the same samples. Yeast cells were subjected to heat stress and measurements were taken at three different conditions: $30°$C 20 min., $39°$C 20 min. and $39°$C 120 min. Two different strains were analyzed: Wild Type (WT) yeast cells and a strain lacking the mip6 gene ($mip6\Delta$), which is a factor involved in mRNA metabolism [106]. The improved iMM904 *Saccharomyces cerevisiae* GEM model [180] was used as the yeast metabolic network. This metabolic model contains 1226 metabolites, 1577 reactions and 905 genes. RNA-seq data covered the totality of metabolic genes in the GEM, however only 30 metabolites of the metabolic model were present in the NMR data, that included 4 sugars, 12 amino-acids, 4 alcohols, 5 carboxylic acids, and other compounds (CMP, NAD, Glutathione, ATP and GMP).

RNA-seq, ChIP-seq and metabolomics data were pre-processed as described in Nuno-Cabanes et al., 2020 [113]. Differential gene expression and H4K12ac levels were calculated using the limma package [53] and FDR corrected p-values were used (see Chapter 4 for more details). The MAMBA method was implemented in Matlab, and Gurobi library [181] was used as the default solver for linear programming problems.

5.3 Description of the approach

5.3.1 Flux Balance Analysis

Given a metabolic network model containing M metabolites and L reactions, FBA formalizes the mass balance of internal metabolites as a set of linear equations that satisfy the condition:

$$\sum_{j=1}^{L} s_{ij} x_j = 0,\ i = 1, \ldots, M, \tag{5.1}$$

where x_j corresponds to the flux of reaction j, and s_{ij} stands for stoichiometric coefficient of metabolite i in reaction j. This condition represents the steady state or null neat mass balance, i.e. that for all cellular metabolites in the network the sum of all productions and consumptions equals zero. FBA calculates $\mathbf{x}$ (vector of fluxes for all reactions) given $\mathbf{S}$ (an M x L stoichiometric matrix), which is known from the metabolic network model. Capacity constraints can be added to the equation system as inequalities to delimit the solution space:

$$lb_j \leq x_j \leq ub_j, \; j = 1, \ldots, L \tag{5.2}$$

where lb_j contains the lower bounds of the reaction j, i.e., minimum flux value that reaction j is allowed to carry; and ub_j indicates upper bound of reaction j.

For most metabolic networks, this results in an undetermined system of equations. Thus, the FBA method makes use of linear programming (e.g.: mixed- integer linear programming or milp) to determine the optimal flux distributions given the objective function, OF ($f(\mathbf{x})$):

$$\begin{gathered} \text{max} f(x) \\ \text{subject to} \\ \mathbf{Sx} = 0 \\ \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \end{gathered} \tag{5.3}$$

The OF, $f(\mathbf{x})$, contains coefficients associated to the values of $\mathbf{x}$ to satisfy the restrictions imposed to the model. In regular FBA, the OF is usually the maximization of the biomass production reaction and therefore, only the coefficient of the reaction of biomass production biomass is set to be distinct from zero. However, the coefficients in $f(\mathbf{x})$ can be customized to maximize any reaction in the metabolic network or combination of reactions.

Figure 5.1 describes FBA model construction for a small toy metabolic network, where biomass production is obtained through reaction R_5 (x_5). The metabolic network model provides all required objects except $\mathbf{x}$, which is the model solution. FBA can only be applied to a single condition or biological status.

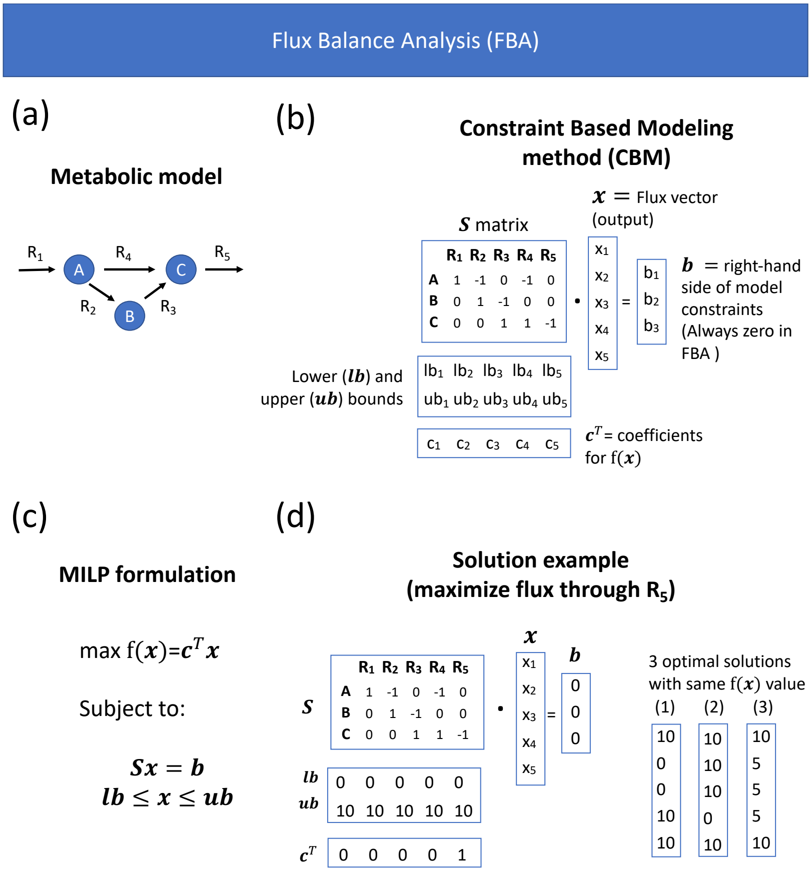


Figure 5.1: Illustration of FBA. a) Representation of the metabolic network. b) Mathematical representation of FBA elements. c) FBA formulated as a linear programming problem. d) Example of a FBA solution for the toy network.

5.3.2 Integration of gene/protein associated information into GEMs

Gene-protein-reaction rules (GPRs) describe the associations between genes/proteins and reactions by using Boolean definitions that represent the gene(s) encoding the protein(s) required to catalyze a given reaction. Therefore, GPRs unlock the integration of gene/protein associated omic data into metabolic models resulting into the so-called Genome-scale metabolic models or GEM, that contains both the metabolic network and the gene-reaction associations. GPRs are used to constraint the fluxes of metabolic reactions based on gene/protein-associated omic data. As shown in Figure 5.2, a metabolic reaction is allowed to be active if and only if the associated GPR is TRUE, which is determined by the expression state of the gene(s) involved. These GPR-dependent constraints are included within the capacity constraints into CBM. Additionally, MAMBA introduces stoichiometric GPRs (S-GPRs) into the model which provide the information about the subunits resulting from the transcription of each gene required to have a catalytically active unit [182].

5.3.3 Formalizing multiomic-based constraints in MAMBA

MAMBA uses omic data on top of the metabolic model to constraint the FBA solution. In particular, a gene-centric omic data type (e.g. gene expression) and metabolomic data are used to formulate new

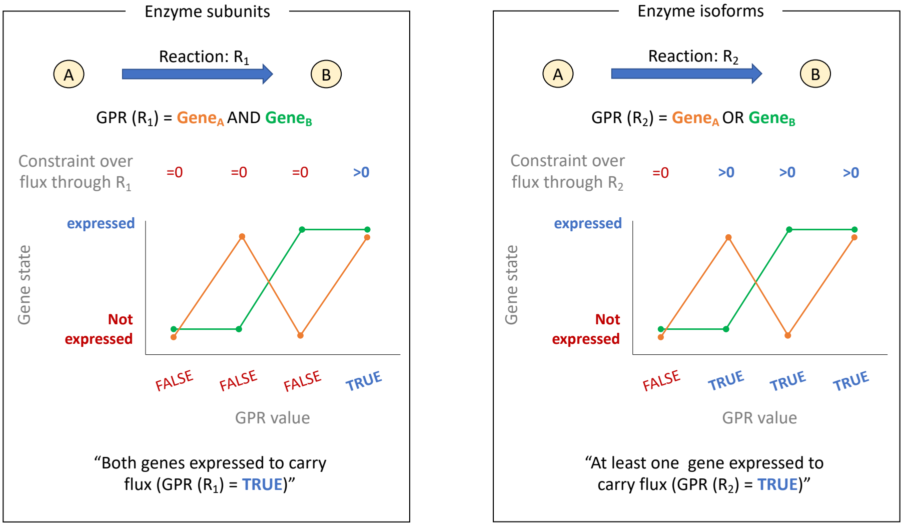

Enzyme subunits
Reaction: R_1
A
B
GPR (R_1) = $Gene_A$ AND $Gene_B$
Constraint over flux through R_1
=0
=0
=0
>0
Gene state
expressed
Not expressed
FALSE
FALSE
FALSE
TRUE
GPR value
"Both genes expressed to carry flux (GPR (R_1) = TRUE)"
Enzyme isoforms
Reaction: R_2
A
B
GPR (R_2) = $Gene_A$ OR $Gene_B$
Constraint over flux through R_2
=0
>0
>0
>0
Gene state
expressed
Not expressed
FALSE
TRUE
TRUE
TRUE
GPR value
"At least one gene expressed to carry flux (GPR (R_2) = TRUE)"

Figure 5.2: Graphical representation of GPRs. Left panel represents a GPR for enzyme subunits that have an "AND" relationship where both genes need to be expressed for the reaction to be active. Right panel represents a GPR for enzyme isoforms, which are modelled by "OR" relationships.

model constraints. Therefore, MAMBA requires several input data in addition to the metabolic model. MAMBA works by simultaneously modeling all experimental conditions under study (two at least) as it uses relative feature quantification. Regarding gene-centric omic data, the algorithm needs the output of a differential feature analysis, i.e., an effect-size measure and the associated p-values. Similarly, metabolite ratios (or an effect-size measure as well) between conditions must be also given to the model as input data. MAMBA modifies the elements of FBA model (Equation 5.3) according to the input data as described below (Figure 5.3).

5.3.3.1 Gene/protein associated data

MAMBA works with differential values between two comparing conditions -the usual design of omics experiments- rather than with absolute values. By using differential data, MAMBA bypasses the difficult task of defining absolute threshold values to determine different levels of gene expression. More importantly, MAMBA allows the incorporation in the model of any type of omics data, as long as a gene/protein associated value can be computed, thereby unlocking the utilization of multi-omics data in FBA. In the reminder of the model formulation, we will use the notation gene to generally represent any omics measurement that can be associated to a gene ID such as expression, protein or chromatin data. Consequently, constraints over reactions can be generated based on different omic data types and, by comparing the resulting models, inferences can be done about the control of the metabolic network by different molecular regulatory layers. In this work, we demonstrate MAMBA using gene expression (RNA-seq) and histone modification (ChIP-seq) data, but other omics modalities, such as DNA methylation or chromatin accessibility could also be used. To incorporate gene-associated omics measurements, MAMBA adapts the MADE model [98] developed for gene expression data. The algorithm requires an effect size measure (typically, log2 Fold-change) or a statistic that compares gene values across conditions, together with the associated p-values. Based on the omics data, genes are classified into three categories: UP (gene activity increases significantly), DOWN (gene activity decreases) and CONSTANT (non-significant change). The algorithm finds a sequence of binary expression states (one gene state for each comparing condition) that best fits the differential expression data. Formally, the sequence of binary gene states returned by the algorithm is expressed as:

$$\underset{x \in X}{\text{argmin}} \sum_{i=1}^{N-1} w(p_{i \to i+1}) |d_{i \to i+1} - x_{i \to i+1}| \tag{5.4}$$

where N is the number of conditions, $i \to i+1$ represents a transition from condition n_i to n_{i+1}, $\mathbf{x}$ is the solution vector containing the predicted binary gene states, $\mathbf{p}$ contains the associated p-values for each transition, $w()$ is a weighted function used to prioritize gene state calculations and $\mathbf{d}$ is the vector of observed differences being: 1 for UP genes, -1 for DOWN genes and 0 for CONSTANT genes. Typically the weighted function $w()$ is the -log10(p-value) of the differential expression, implying that when two gene transitions (gene states) are not simultaneously feasible, p-values are used to prioritize and the transition associated to a lower p-value is reflected in the output of the model. MAMBA uses equation 5.4 to find a solution of gene states (i.e., 0 or 1) that minimizes the differences between the observed and predicted gene state changes. However, in the formulation of the stoichiometric matrix, vector of differences ($\mathbf{d}$) cannot be included and hence an equivalent expression of equation 5.4 is used. Therefore, for a given transition $i \to i+1$, the OF for MAMBA can be defined as the following weighted sum:

$$\begin{aligned} f_{i \to i+1}(\mathbf{x}) = & \sum_{x \in U} w(p_{i \to i+1})(x_{i+1} - x_i) \\ & + \sum_{x \in D} w(p_{i \to i+1})(x_i - x_{i+1}) \\ & - \sum_{x \in C} w(p_{i \to i+1}) \Delta_{x_i, x_{i+1}} \end{aligned} \tag{5.5}$$

where U is the list of UP genes, D is the list of DOWN genes and C is the list of CONSTANT genes. Regarding UP genes, if the solution matches the expected output ($x_i = 0$ and $x_{i+1} = 1$), the first element of this equation will be positive and therefore contributes to maximize $f(\mathbf{x})$. The same occurs for DOWN genes as the position of x_i and x_{i+1} is swapped in the formula. Finally, the third element controls the contribution of constant genes to the OF by defining $\Delta_{xi,xi+1}$ as a binary variable that takes the value 0 when $x_i = x_{i+1}$ and 1 otherwise. Thus, the third element is negative when the solution does not match the expected result, thereby penalizing $f(\mathbf{x})$.

5.3.3.2 Metabolomics data

Next to gene-associated omics data, MAMBA also incorporates into the model semi-quantitative metabolomics data obtained from comparing experimental conditions. To add metabolomics data to the MAMBA model, we incorporated the concept of sink reaction presented in Schmidt et al., 2013 [66]. This implementation consists of creating a new set of artificial metabolic reactions, the sink reactions, that connect measured metabolites to the GEM through a two-step process. First, reversible reactions are transformed into two irreversible reactions. Next, for each measured metabolite, a turnover metabolite is added to the model and connected to every reaction that produce or consume the actual measured metabolite. Finally, a sink reaction is included having the turnover metabolite as unique reactant and no products (Figure 5.3.a). To ensure that all detected metabolites have associated sink reactions with non-zero flux in the solution, the lower bound of sink reactions is set to a small positive value limited by the linear programming solvers numerical tolerance [10^{-8} [181]].

Once turnover metabolites and their corresponding sink reactions are included in the model, constraints can be formulated. Basically, the observed metabolite ratios between conditions are modeled. For instance, let us consider a metabolite m measured in two conditions, A and B with values $m_A = 6$ and $m_B = 12$. Since the quantification of m is two times greater in condition B than in A, the flux of m through the sink reaction in condition B should be twice the flux through the sink reaction in condition A. This ratio requirement can be added to the model by imposing both lower and upper bounds of sink reactions to have the same ratio relationship. However, such modification of the capacity (hard) constraints dramatically decreases the number of feasible solutions. Instead, a penalty on the OF, which is a soft model constraint, was implemented as the parameter r, which reflects the difference between the observed and the predicted ratio. In order to deal with positive and negative deviations of the solution with respect to the expected ratio, two different penalty scores are created, r^+ and r^- (Figure 5.3.b and c). The difference between both is that r^+ is constrained to be greater than or equal to zero and r^- is constrained to be lower than or equal to zero. To illustrate the meaning of these two penalty scores, let us consider v as the flux through the sink reaction associated to metabolite m and only two conditions (A and B), the penalty imposed is derived from:

$$v_A \times \frac{m_B}{m_A} - v_B + r_m^+ + r_m^- = 0 \tag{5.6}$$

From the expression above, one can derive that the lower the values of r_m^+ and r_m^- , the higher the concordance between observed and modeled metabolite ratios. Therefore, the model is forced to minimize both penalty values:

$$\underset{m \in M}{\text{argmin}} \sum_{i=1}^{N-1} r^+_{m,i \to i+1} + |r^-_{m,i \to i+1}| \tag{5.7}$$

where N is the number of conditions and M is the number of metabolites (or number of sink reactions). Finally, the new component is included in the OF (Equation 5.5) and its final form is:

$$\begin{aligned}
f_{i\to i+1}(x) = &\sum_{x\in U} w(p_{i\to i+1})(x_{i+1} - x_i) \\
&+ \sum_{x\in D} w(p_{i\to i+1})(x_i - x_{i+1}) \\
&- \sum_{x\in C} w(p_{i\to i+1})\Delta_{xi,xi+1} \\
&- \sum_{m\in M} \; g(m)\; (r^{+}_{m,i\to i+1} + |r^{-}_{m,i\to i+1}|)
\end{aligned} \tag{5.8}$$

The expression above is the final OF that MAMBA maximizes to find the optimal solution for $i = 1, \ldots, N$, being N the number of conditions. Note that an additional weight function for metabolomic constraints, $g(m)$, has been included. This function has the purpose of balancing gene expression and metabolomics constraints to give them similar weights in the OF. Since constraints from gene expression include a weight function, $w()$ based on p-values, these constraints may be very strong for very low p-values, e.g. $w(1 \times 10^{-100}) = 100$ (considering a p-value $= 1 \times 10^{-100}$). Additionally, without considering any weight function for metabolomic constraints, every deviation from the actual ratio has the same penalty. For instance, let us consider two metabolites m_1 and m_2 having observed ratios between 2 conditions A and B defined as: $m_{1,A \to B} = 3$ and $m_{2,A \to B} = 6$, while their predicted ratios are $m_{1,\hat{A} \to B} = 2$ and $m_{2,\hat{A} \to B} = 5$. Both metabolites would contribute to the OF with the same magnitude, -1, which is the penalty associated to the difference between observed and predicted ratios according to Equation 5.6. However, the prediction for metabolite m_2 is more accurate than for metabolite m_1 $(5/6 > 2/3)$. To account for this issue, ratios of metabolites are standardized between 0 and 1, being 1 the highest observed ratio. The metabolite weight function, $g(m)$ captures these two considerations by defining $g(m)$ as:

$$g(m) = 1 \times 10^{(p - s_m)} \tag{5.9}$$

where p is the value of the highest p-value of the gene expression data after $w()$ (-log10) transformation (considering only genes included into the GEM) and s_m is the standardized ratio of metabolite m. Considering the previous example with two metabolites, the standardized ratio for metabolite m_1 is 0 and for m_2 is 1. Hence, the penalty for m_1 is $g(m_1) = 1 \times 10^{(p-0)}$, while the penalty for m_2 is $g(m_2) = 1 \times 10^{(p-1)}$, and therefore, penalty of m_2 is lower than penalty of m_1 (as p is a common value).

Finally, MAMBA representation as a linear programming optimization problem is defined as follows:

$$\begin{aligned} &\text{max} f(\mathbf{x}) \\ \text{subject to} \quad & \end{aligned} \tag{5.10}$$

$$\sum_{j=1}^{L} a_{ij} x_j \leq b_i, \; i = 1, \ldots, M, \tag{a}$$

$$lb_j \times z_j \leq x_j \leq ub_j \times z_j \tag{b}$$

where $\mathbf{A}$ (with general element a_{ij}) contains original stoichiometric matrix and transcriptomics and metabolomics constraints, vector $\mathbf{b}$ sets the boundaries of the constraints that can be different from zero, vector $\mathbf{x}$ is the optimal solution to be found and vector $\mathbf{z}$ contains the S-GPR constraints, i.e, z_j is 1 if the S-GPR is TRUE and 0 otherwise. Finally, the final OF including all transitions is represented as $f(\mathbf{x}) = \sum_{i=1}^{N-1} f_{i \to i+1}(\mathbf{x})$

5.3.4 Evaluation of MAMBA method

5.3.4.1 Sensitivity and Robustness analysis

The sensitivity analysis consists in evaluating the impact of each reaction state on the optimization function. Each reaction, one by one, was forced to be active (A) or inactive (I) at every condition by modifying capacity constraints, i.e. a given reaction is forced to be active by setting its lower bound > 0 and it is forced to be inactive by setting both, lower and upper bounds, equal to zero. The optimization function was evaluated in each case and compared to the MAMBA model without forced reactions. In case the result equals the unforced model, the reaction is deemed not to be part of the solution, while if the result is substantially different, the reaction is considered critical in the model. Therefore, sensitivity analysis identifies both the reactions that most affect the optimization function and the reactions with an undetermined state, i.e., the optimization result is the same whether they are active or inactive. Robustness analysis identifies the consistency of predictions as a function of changes in method parameters and identifies reaction states (Active or Inactive) that are unambiguously predicted by the model. The varying parameter in robustness analysis is the logFC threshold to call a gene differentially expressed, as this parameter is regularly set by the user. Tested logFC values were set according to the overall distribution of logFC values between all conditions in the experiment. In particular, quantiles 25, 50 and 75 of the logFC distribution were selected. In addition, logFC equal to 1 was also tested and consequently four different logFC thresholds were evaluated in the robustness analysis. After performing both analyses, highly confidence reaction states were determined consisting of those reactions that pass both the sensitivity and robustness analyses. A reaction has a highly confident or unambiguous state if i) a change of its state reduces the result of the OF, and ii) its state does not change when using different logFC thresholds.

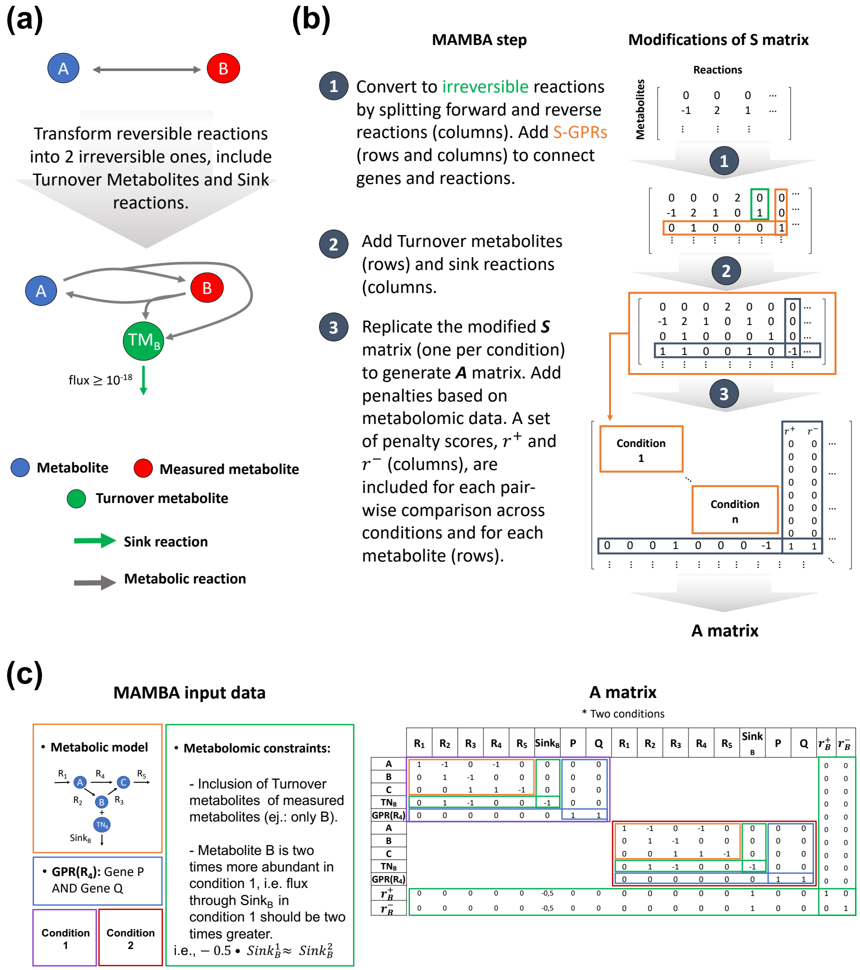
5.3.4.2 Evaluation of metabolite prediction accuracy

Since the MAMBA model can be used to predict metabolite levels, we used metabolite prediction accuracy to evaluate MAMBA and compare it with MADE, a related approach that incorporates gene expression but no metabolomics data into the GEM. We evaluated the impact of including metabolomic data into the model prediction error using a leave-X-out strategy. Basically, we calculated the error of the MAMBA model in predicting measured metabolites as an increasing number of metabolites were incorporated in the model, i.e., we first fit a model including measurements for one metabolite and compute metabolite prediction error for the remaining metabolite dataset, next we fit the model including measurements for two metabolites and calculate again the error estimates. This was repeated increasing by one the number of included metabolites to reach the complete MAMBA model that includes all available metabolite data and the whole process is repeated one thousand times. The Root Mean Square Error of Prediction (RMSEP) [183] across all predicted metabolites at each leave-X-out iteration was used as error metric.

5.4 Results

5.4.1 MAMBA model

The MAMBA model unlocks the utilization of time-course non-quantitative or relative metabolomics data from different analytic platforms such as MS or NMR as well as widely adopted multi-omics approaches that generate sequencing within the CBM framework for the study of metabolic networks. A number of approaches address partially the integration of these omics into a CBM framework. In this sense D-MFA enables the integration of absolute metabolomics concentration from time-course experiments into a metabolic models, however this approach is limited to quantitative measurements and restricted to networks with low degrees of freedom imposing a strong limitation on the size of the analyzed network. Other methods like MADE are limited to the integration of transcriptomic data from time-course experiments into GEMs. MAMBA addresses this limitation by using relative values whereby the semi-quantitative omics data are modeled as transitions (comparison between two conditions) [98]. Hence, the MAMBA framework uses differential expression/quantification data to characterize the metabolic network across N conditions enabling the dynamic modeling of the system. Compared to a basic FBA, MAMBA includes two set of constraints: gene/protein associated constraints and metabolomics associated constraints.



1. Convert to irreversible reactions by splitting forward and reverse reactions (columns). Add S-GPRs (rows and columns) to connect genes and reactions.

2. Add Turnover metabolites (rows) and sink reactions (columns.

3. Replicate the modified ***S*** matrix (one per condition) to generate ***A*** matrix. Add penalties based on metabolomic data. A set of penalty scores, r^+ and r^- (columns), are included for each pair-wise comparison across conditions and for each metabolite (rows).

- **GPR(R_4):** Gene P AND Gene Q

- **Metabolomic constraints:**

  - Inclusion of Turnover metabolites of measured metabolites (ej.: only B).

  - Metabolite B is two times more abundant in condition 1, i.e. flux through Sink_B in condition 1 should be two times greater.

  i.e., $-0.5 \bullet Sink_B^1 \approx Sink_B^2$

A matrix

* Two conditions

	R_1	R_2	R_3	R_4	R_5	Sink_B	P	Q	R_1	R_2	R_3	R_4	R_5	Sink B	P	Q	r_B^+	r_B^-
A	1	-1	0	-1	0	0	0	0									0	0
B	0	1	-1	0	0	0	0	0									0	0
C	0	0	1	1	-1	0	0	0									0	0
TN_B	0	1	-1	0	0	-1	0	0									0	0
GPR(R_4)	0	0	0	0	0	0	1	1									0	0
A									1	-1	0	-1	0	0	0	0	0	0
B									0	1	-1	0	0	0	0	0	0	0
C									0	0	1	1	-1	0	0	0	0	0
TN_B									0	1	-1	0	0	-1	0	0	0	0
GPR(R_4)									0	0	0	0	0	0	1	1	0	0
r_B^+	0	0	0	0	0	-0,5	0	0	0	0	0	0	0	1	0	0	1	0
r_B^-	0	0	0	0	0	-0,5	0	0	0	0	0	0	0	1	0	0	0	1

Figure 5.3: Graphical representation of model modifications performed by MAMBA. a) Modifications of the genome-scale metabolic model. b) MAMBA model is contained in $\mathbf{A}$ matrix resulting from the modification of the standard stoichiometric matrix ($\mathbf{S}$). c) Structure of MAMBA model and input data require to construct $\mathbf{A}$ matrix.

This reduces the solution space of the model and the characterization of the metabolic network is more robust and accurate. A basic FBA model is defined by (Figure 5.1): the stoichiometric matrix, $\mathbf{S}$, that indicates the association metabolites (rows) with reactions (columns); vectors $\mathbf{lb}$ and $\mathbf{ub}$ that contain the minimal and maximal fluxes allowed for reactions, respectively; vector $\mathbf{b}$ is the right side of the mass balance equations and is equal to 0 at all its elements to meet the zero mass balance condition; and vector $\mathbf{c}$ contains the reaction coefficients for the optimization function. The output is a vector $\mathbf{x}$ with the same size as $\mathbf{c}$, i.e., columns of $\mathbf{S}$. For MAMBA, gene/protein (via S-GPRs) and metabolomic (via Sink reactions) associated constraints are included alongside $\mathbf{S}$, resulting into a new larger matrix, $\mathbf{A}$. Figure 5.3.c shows the design of $\mathbf{A}$ matrix with a toy metabolic network and considering only two conditions.

5.4.2 Application of MAMBA to a yeast heat-shock dataset improves metabolic prediction accuracy

The time-series multi-omic yeast data (see Section 5.2) was used to test MAMBA. This is a multifactorial experimental design with two factors: strain (2 levels: WT and $mip6\Delta$) and temperature-time (3 levels: baseline or $30°$C 20 minutes, $39°$C 20 minutes and $39°$C 120 minutes) where three omic modalities (RNA-seq, H4K12ac ChIP-seq and metabolomics) were measured. A separate MAMBA model was obtained for each strain, each of them including the three time points of the heat-shock treatment. Consequently, each MAMBA model contains three conditions and two transitions, that is: i) First transition $= 30^{\circ}$C 20 min. $\rightarrow 39^{\circ}$C 20 min., and ii) Second transition $= 39^{\circ}$C 20 min. $\rightarrow 39^{\circ}$C 120 min.. The two MAMBA outputs were then compared to evaluate the differences between strains regarding heat-shock adaptation. Table 5.1 shows the number of genes, H4K12ac peaks and metabolites measured in our experiment and the number of differentially expressed features (presented in Chapter 4) that were used to feed the MAMBA models.

Table 5.1: Number of genes, H4K12ac peaks and metabolites measured in our experiment and the number of differentially expressed features (see Methods for details) that were used to feed the MAMBA models.

	WT			$mip6\Delta$	
	Total	First transition	Second transition	First transition	Second transition
Transcripts	6379	815	386	765	379
H4K12ac peaks (gene-associated)	6379	248	196	114	29
Metabolites	42	8	16	14	15
Number of reactions					
MAMBA model	1577	371	186	293	125

We used MAMBA results with the WT strain to validate the consistency of the novel approach. First, predicted metabolite ratios were compared with experimental measurements and prediction error was calculated as Root Mean Square Error of Prediction, RMSEP (see section 5.3.4.2 for details). In addition, MAMBA was compared with

MADE algorithm, a similar method that only integrates transcriptomics data [98]. MAMBA and MADE performances were contrasted to evaluate whether the inclusion of a new layer of omic information improves the prediction accuracy of metabolites. We found that MAMBA outperformed MADE in terms of metabolite ratios prediction accuracy with lower RMSEP values in both transitions (Figure 5.4.a). Moreover, the prediction error decreased when each new metabolite data was incorporated into the model (Figure 5.4.b). These results demonstrate that the inclusion of metabolite measurements into the GEM boosts network characterization. We also observed that metabolites with other measured metabolites close in the metabolic network (e.g. amino acids), had better predictions than isolated metabolites (Figure 5.4.c), which adds to the consistency of the MAMBA model when incorporating metabolic information. This analysis was performed with the set of metabolites in the model that were measured with experimental data (30).

MAMBA results also recapitulated the known biology. We delved into the Trehalose pathway, which is a well-known process affected by heat stress in yeast cells [184]. MAMBA predicted the activation of Trehalose production pathway after heat-shock and its maintenance at 120 minutes (Figure 5.5.a). However, trehalose metabolism genes were down-regulated after 120 minutes compared to 20 minutes of heat stress (Figure 5.5.b) which was not readily consistent with high trehalose production at 120 minutes. Nonetheless, our metabolomic data show trehalose rises steadily during heat stress (Figure 5.5.c). Indeed,

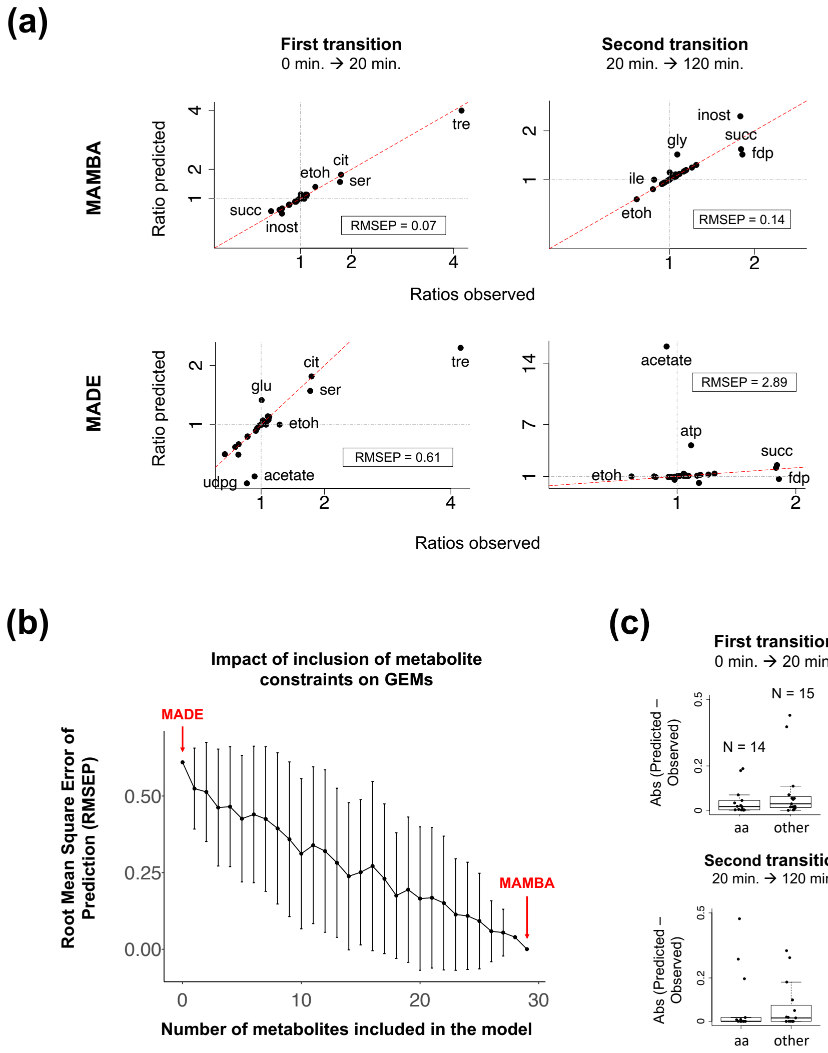


Figure 5.4: MAMBA validation of prediction accuracy. a) Metabolite prediction accuracy. Predicted vs observed metabolite ratios are shown for both transitions (comparisons) in the WT model. MAMBA is compared to MADE which only uses transcriptomic data. b) The effect of the number of metabolites included in the model on the prediction error (RMSEP) only for the first transition. Mean (dots) ± sd (error bars) are represented. c) Prediction error of MAMBA by metabolite type separating between amino acids (aa) and other metabolites.

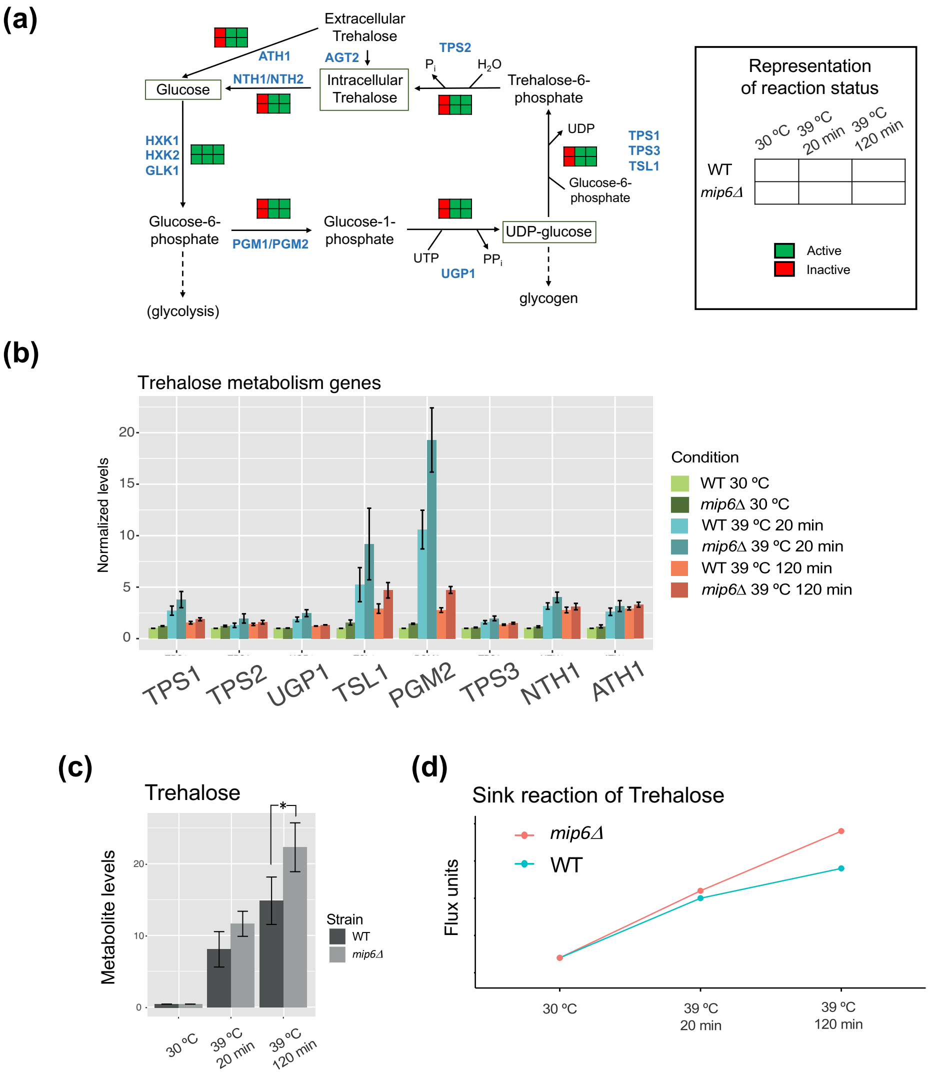


Figure 5.5: MAMBA validation of biological consistency. a) Trehalose pathway: Predicted reaction status. Trehalose pathway activity is predicted to increase at 20 min and to remain active allowing trehalose accumulation. b) Expression of genes involved in trehalose pathway. c) Trehalose quantification from NMR metabolomic data. d) MAMBA prediction of Trehalose quantification.

MAMBA predicted Trehalose over-production (Figure 5.5.d) regardless of gene expression data. Again, this is consistent with the inclusion of metabolomics data improving network characterization as only gene expression is not always completely representative of metabolic changes.

5.4.3 Deciphering differential behavior between strains

We next used MAMBA to study metabolic differences between strains. Reaction states (active or inactive) were compared, and we found 211 reactions that had a different state between strains at one time point at least (list of reactions at Appendix 4). To understand the biological meaning of these differences, we performed a pathway-level analysis where yeast pathways from KEGG database were used. Given a metabolic pathway P containing n reactions, we defined a pathway enrichment score (PES) as the percentage of reactions of P contained in the list of differential reactions identified by MAMBA (Q):

$$PES = 100 \times \frac{|P \cap Q|}{n} \tag{5.11}$$

In order to set a relevance threshold, we compared each pathway PES to the PES in the general KEGG pathway, “Metabolic Pathways” -that contains all metabolic reactions-, which was 17%. Thus, those pathways with a PES higher than 17% were selected as relevant pathways to describe strain differences. This resulted into a list of 25 pathways showed in Figure 5.6.a. We observed that this relevant pathways recapitulate main biological changes across conditions using Gene Set Variation Analysis (GSVA) method [63], that computes a sample-wise score for each pathway that summarizes the expression of the genes contained in it. An unsupervised clustering performed on the GSVA scores Figure 5.6.a showed that samples (columns) were separated by both experimental conditions (strain and time), indicating that the list of relevant pathways summarized the main biological signal across conditions.

To better represent the predicted activity for these relevant pathways, we computed the pathway activity scores (PAS) that is the percentage of active reactions (binary reaction status equal to 1):

$$PAS = 100 \times \frac{\sum_{i=1}^{n} \delta(p_i, 1)}{n}, \text{ where } \delta(i,j) = 1 \text{ if } i = j \text{ and } 0 \text{ if } i \neq j \tag{5.12}$$

Figure 5.6.b represents PAS of two key processes within core carbon metabolism, Glycolysis and TCA cycle. Activity scores for the rest of relevant pathways can be found in Appendix 5. We observed that $mip6\Delta$ mutant over-activates Glycolysis and TCA cycle while the WT decreases the activity of TCA cycle and maintain the status of Glycolysis. Similar to TCA, Pyruvate metabolism is down-regulated in WT but over-activated in the mutant during heat stress. In fact, global "Carbon metabolism" pathway presented the same pattern, and "Peroxisome metabolism" and "Pentose phosphate pathway" also showed similar profile. "Fatty Acid (FA) degradation" profile was also different between strains. WT maintains “FA degradation” over time while the mutant showed a huge increase of FA degradation activity. Interestingly, “FA degradation” and “Peroxisome metabolism” shared most of the mapped differential reactions that were involve in Peroxisomal FA degradation (PFAD). Analyzing the average flux through reaction involved in PFAD we observed a consistent pattern where the average flux decreases in the WT and stay flat in the mutant (Figure 5.6.c). Lastly, activation scores of “Purine” and “Pyrimidine” metabolism pathways slightly decrease in the WT and remains flat in the mutant.

The disconnection of TCA cycle and Glycolysis in the WT can be explained by the increase in Ethanol (ETOH) production after heat-shock (Figure 5.6.d). MAMBA predicts this differential behavior as the modeled ethanol ratios for both transitions are 1.4 (time 20 vs time 0) and 0.6 (time 120 vs time 20) for the WT (Figure 5.4.a).

Based on the list of differential reactions we then evaluated which reactants and products were predominant. Table 5.2 shows that NADH/NADPH were enriched as products of differential reactions, and therefore NAD/NADP were enriched as reactants. Moreover, most of the reactions that have either NADH or NADPH as products (12/15) were down-regulated in the WT after heat-shock but not in the mutant. This result suggested an imbalance in the generation of reducing power between strains towards $mip6\Delta$ mutant.

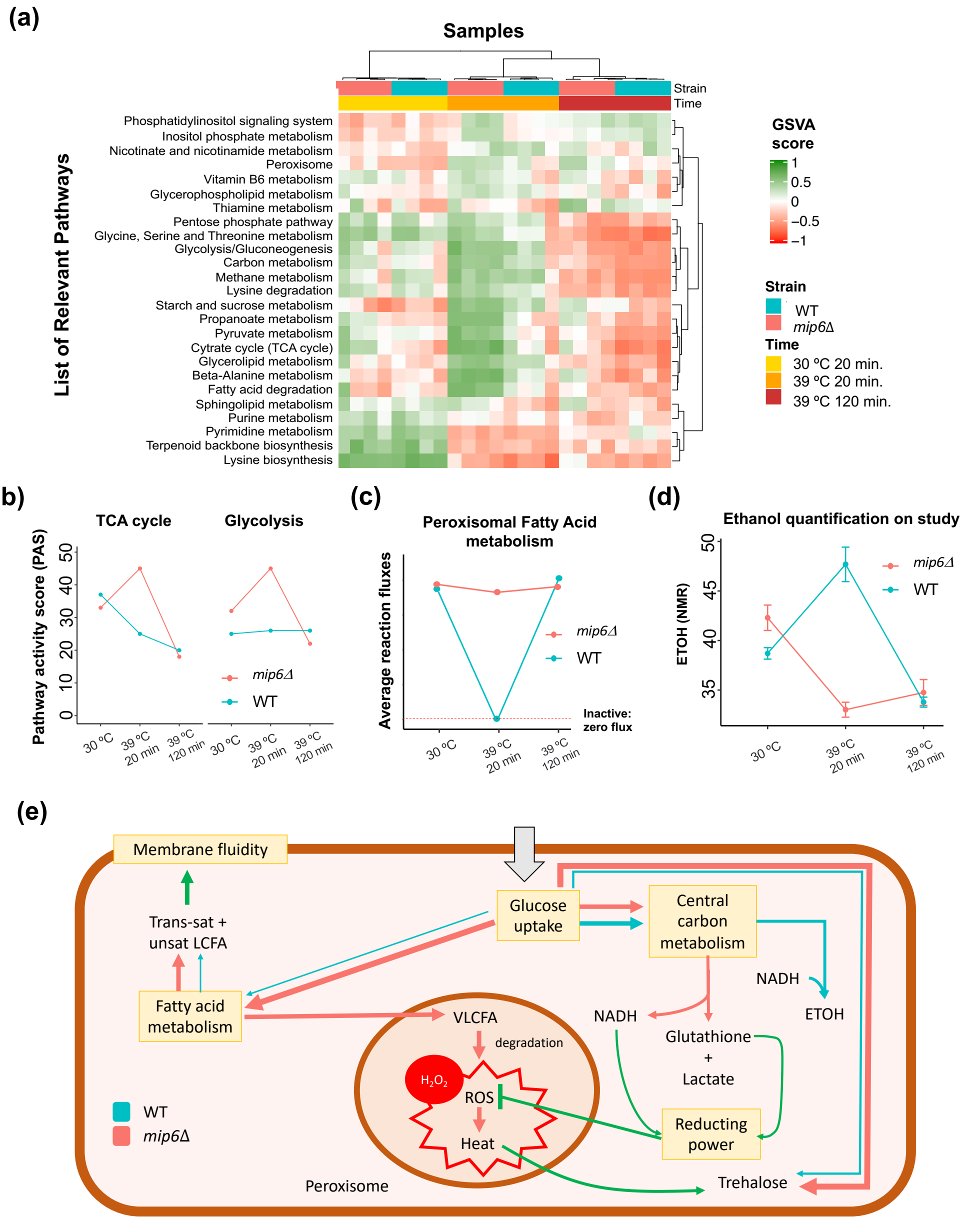

(a)
Samples
Strain
Time
List of Relevant Pathways
Phosphatidylinositol signaling system
Inositol phosphate metabolism
Nicotinate and nicotinamide metabolism
Peroxisome
Vitamin B6 metabolism
Glycerophospholipid metabolism
Thiamine metabolism
Pentose phosphate pathway
Glycine, Serine and Threonine metabolism
Glycolysis/Gluconeogenesis
Carbon metabolism
Methane metabolism
Lysine degradation
Starch and sucrose metabolism
Propanoate metabolism
Pyruvate metabolism
Cytrate cycle (TCA cycle)
Glycerolipid metabolism
Beta-Alanine metabolism
Fatty acid degradation
Sphingolipid metabolism
Purine metabolism
Pyrimidine metabolism
Terpenoid backbone biosynthesis
Lysine biosynthesis
GSVA score
1
0.5
0
−0.5
−1
Strain
WT
mip6Δ
Time
30 ºC 20 min.
39 ºC 20 min.
39 ºC 120 min.
(b)
TCA cycle
Glycolysis
Pathway activity score (PAS)
0
10
20
30
40
50
30 ºC
39 ºC 20 min
39 ºC 120 min
mip6Δ
WT
(c)
Peroxisomal Fatty Acid metabolism
Average reaction fluxes
mip6Δ
WT
Inactive: zero flux
30 ºC
39 ºC 20 min
39 ºC 120 min
(d)
Ethanol quantification on study
ETOH (NMR)
35
40
45
50
mip6Δ
WT
30 ºC
39 ºC 20 min
39 ºC 120 min
(e)
Membrane fluidity
Trans-sat + unsat LCFA
Fatty acid metabolism
Glucose uptake
Central carbon metabolism
NADH
ETOH
NADH
VLCFA
degradation
H_2O_2
ROS
Heat
Glutathione + Lactate
Reducting power
Trehalose
Peroxisome
WT
mip6Δ

Figure 5.6: Model of differential adaptation to heat-shock between strains revealed by MAMBA analysis. a) Relevant pathways heatmap. Those pathways containing reactions with differential activity state profile between strains were considered. Samples are fully separated according to experimental conditions. b) Pathway activity score of Glycolisis and TCA cycle, calculated as the percentage of active reactions. c) MAMBA predicted fluxes through peroxisomal fatty acid metabolism reactions. MAMBA predicts a shutdown of this pathway in WT while it remains active in the mutant. d) Ethanol quantification from NMR metabolomics data. e) Explanatory model of metabolic differences found between strains. From the carbon source (glucose), the WT produces ethanol, trehalose and LCFAs (Long-chain Fatty acids), that contributes to membrane stabilization. On the contrary, $mip6\Delta$ mutant produces lactate + glutathione instead of ethanol, a higher amount of trehalose and LCFAs, which are metabolized in the peroxisome to produce H_2O_2 resulting in oxidative and heat stress for the mutant cells.

Table 5.2: Number of reactions that have NADH or NADPH as products (left) and NAD or NADP as reactants (right) separated by whether they are in the list of 211 differential reactions between strains. Significant enrichment assessed by Fisher's exact test [61]

	NADH/NADPH as products		NAD/NADP as reactants	
	Yes	No	Yes	No
Differential Reactions	15	198	15	198
Rest	31	1333	35	1329
Fisher's exact test p-val:	0.0006		0.002	

An explanatory model of the main metabolic differences between strains found by MAMBA is represented in Figure 5.6.e. Starting from the carbon source (glucose), the WT produces ethanol, trehalose and LCFAs (Long-chain Fatty acids), that contributes to membrane stabilization [185]. On the contrary, $mip6\Delta$ mutant produces lactate + glutathione instead of ethanol, a higher amount of trehalose and LCFAs, which are metabolized in the peroxisome to produce H_2O_2 resulting in oxidative and heat stress for the mutant cells.

5.4.4 Evaluating the effect of mip6 affinity on metabolic changes

We seized PAR-CLIP data, a technology that profiles RNA-protein interactions, to assess to which extent metabolic differences between strains might be caused by direct interaction with mip6. Mip6 is involved in RNA export from the nucleus and in stabilizing mRNAs in the cytosol through direct protein-RNA interactions [106]. PAR-CLIP data was available for WT yeast strain on the same heat stress conditions, indicating a total of 6685 mRNAs bound by mip6, with a small fraction of them (488) showing differential mip6 affinity after heat stress [106]. We compared this list of mip6-bound RNAs to the 150 genes involed in the differential reactions identified by MAMBA. We found that this set of genes showed higher mip6 affinity both at normal growth condition (30°C) and after heat-shock (39°C) (Figure 5.7.a and b) than genes not involved in MAMBA-detected reactions. However, only 10 genes showed differential mip6 affinity at 39°C compared to 30°C (Table 5.3).

Table 5.3: Genes involved in differential reactions identified by MAMBA and showing a significant increasing mip6 affinity after heat-shock according to Martin-Exposito et al., 2019 [106].

Gene Name	Description

Table 5.3: Genes involved in differential reactions identified by MAMBA and showing a significant increasing mip6 affinity after heat-shock according to Martin-Exposito et al., 2019 [106].

Gene Name	Description
AGP1	Low-affinity amino acid permease with broad substrate range; involved in uptake of asparagine, glutamine, and other amino acids; expression regulated by SPS plasma membrane amino acid sensor system (Ssy1p-Ptr3p-Ssy5p); AGP1 has a paralog, GNP1, that arose from the whole genome duplication
SFA1	Bifunctional alcohol dehydrogenase and formaldehyde dehydrogenase; formaldehyde dehydrogenase activity is glutathione-dependent; functions in formaldehyde detoxification and formation of long chain and complex alcohols, regulated by Hog1p-Sko1p; protein abundance increases in response to DNA replication stress
PDE1	Low-affinity cyclic AMP phosphodiesterase; controls glucose and intracellular acidification-induced cAMP signaling, target of the cAMP-protein kinase A (PKA) pathway; glucose induces transcription and inhibits translation
GRE3	Aldose reductase; involved in methylglyoxal, d-xylose, arabinose, and galactose metabolism; stress induced (osmotic, ionic, oxidative, heat-shock, starvation and heavy metals); regulated by the HOG pathway; protein abundance increases in response to DNA replication stress
TPO1	Polyamine transporter of the major facilitator superfamily; member of the 12-spanner drug:H(+) antiporter DHA1 family; recognizes spermine, putrescine, and spermidine; catalyzes uptake of polyamines at alkaline pH and excretion at acidic pH; during oxidative stress exports spermine, spermidine from the cell, which controls timing of expression of stress-responsive genes; phosphorylation enhances activity and sorting to the plasma membrane
ADH2	Glucose-repressible alcohol dehydrogenase II; catalyzes the conversion of ethanol to acetaldehyde; involved in the production of certain carboxylate esters; regulated by ADR1

Table 5.3: Genes involved in differential reactions identified by MAMBA and showing a significant increasing mip6 affinity after heat-shock according to Martin-Exposito et al., 2019 [106].

Gene Name	Description
ADH1	Alcohol dehydrogenase; fermentative isozyme active as homo- or heterotetramers; required for the reduction of acetaldehyde to ethanol, the last step in the glycolytic pathway; ADH1 has a paralog, ADH5, that arose from the whole genome duplication
GRE2	3-methylbutanal reductase and NADPH-dependent methylglyoxal reductase; stress induced (osmotic, ionic, oxidative, heat-shock and heavy metals); regulated by the HOG pathway; restores resistance to glycolaldehyde by coupling reduction of glycolaldehyde to ethylene glycol and oxidation of NADPH to NADP+; protein abundance increases in response to DNA replication stress; methylglyoxal reductase (NADPH-dependent) is also known as D-lactaldehyde dehydrogenase
TPO4	Polyamine transporter of the major facilitator superfamily; member of the 12-spanner drug:H(+) antiporter DHA1 family; recognizes spermine, putrescine, and spermidine; localizes to the plasma membrane

One of the genes identified by the MAMBA analysis which showed a differential mip6 affinity upon heat-shock was ADH1. This gene, responsible for the reduction of acetaldehyde to ethanol, showed a strong mip6 affinity increase at $39°$C (Figure 5.7.c) what might explain its stabilization in the WT and therefore the increase in ETOH production. Interestingly, ADH1 expression did not change between strains after heat stress which indicates that the lower ETOH production in the mutant is not caused by differences in gene expression (Figure 5.7.c).

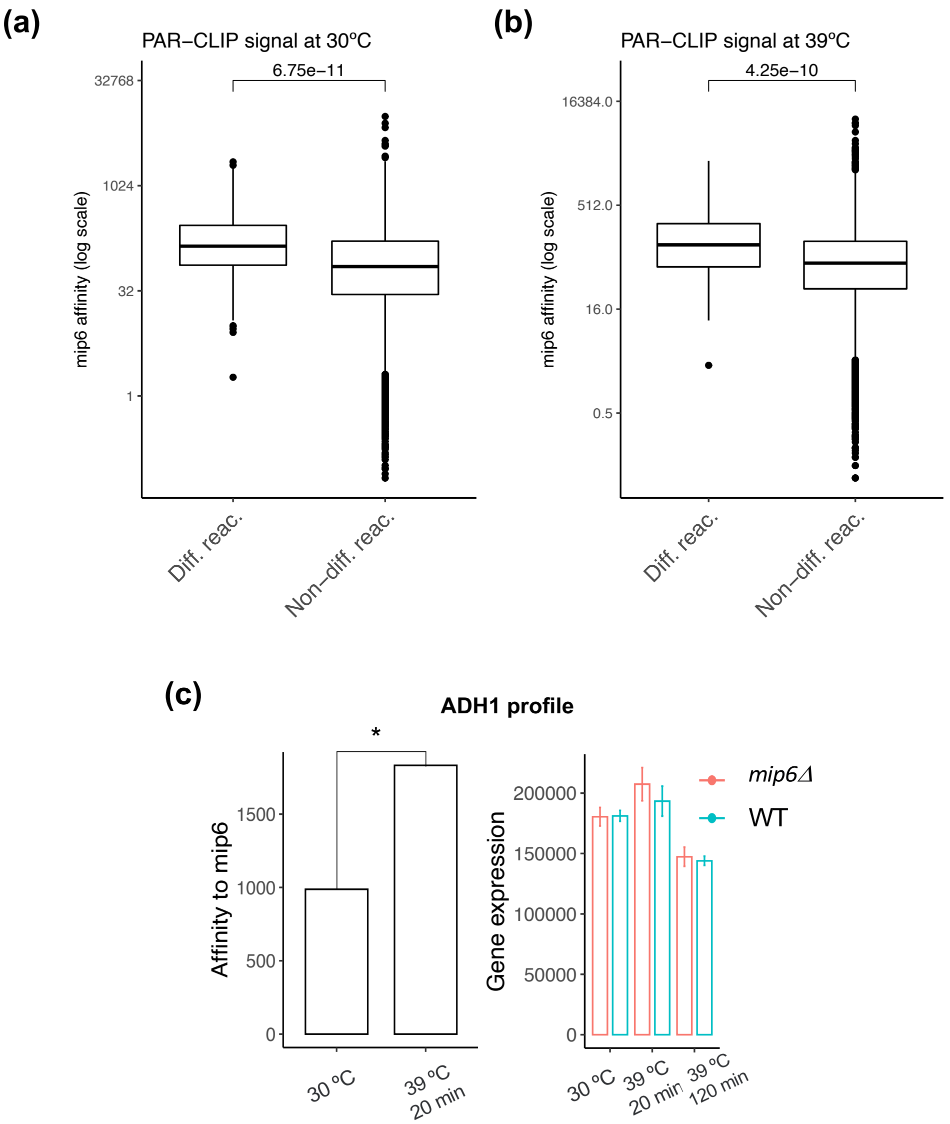


Figure 5.7: Evaluation of mip6 PAR-CLIP data from Martin-Exposito et al., 2019 [106]. a) Signal at 30°C (normal growth condition). b) Signal at 39°C (heat stress). Mip6 affinity is compared between genes involved in differential reactions identified by MAMBA and the rest of genes. Wilcoxon-test p-value is shown [186, 187]. c) ADH1 profile. [left panel] PAR-CLIP data reflecting ADH1 affinity to mip6, which increases during heat-shock. [right panel] ADH1 expression with no significant differences between strains.

5.4.5 Metabolic control by ChIP-seq signal

In order to understand the contribution of histone modifications to metabolic control, MAMBA was run using ChIP-seq data instead of gene expression to predict reaction fluxes. Firstly, we compared reaction status between RNA-seq and ChIP-seq driven MAMBA models and found a higher consistency for the WT as the status of 59.8% of reactions were equally derived from both input data types, while both outputs for $mip6\Delta$ mutant showed a consistency of 40.2% (Table 5.4). MAMBA model using ChIP-seq data also showed a huge difference between WT and $mip6\Delta$ regarding down-regulated reactions after heat-shock, i.e. the mutant presented a lower number of reactions that became inactive after heat-shock according to ChIP-seq data.

Table 5.4: Number of reactions that change over time in both strains according to RNA-seq and ChIP-seq data.

Reaction profile		RNA-seq model		ChIP-seq model		Overlap	
First transition	Second transition	WT	$mip6\Delta$	WT	$mip6\Delta$	WT	$mip6\Delta$
Constant	Decrease	96	57	75	23	64	14
Constant	Increase	90	68	77	59	53	32
Increase	Constant	68	118	57	101	42	80
Increase	Decrease	69	99	12	48	8	37
Decrease	Constant	93	54	86	19	75	5
Decrease	Increase	141	22	103	0	91	0
Total		557	418	410	250	333	168

We also evaluated ChIP-seq data of those genes contained in the list of relevant pathways (Figure 5.6.e), and we found statistically significant differences between strains regarding Peroxisomal FA metabolism (Figure 5.8.a). WT ChIP-seq signal of this set of genes indicated a decrease in histone acetylation signal, that was consistent with the down-regulation of gene expression and the corresponding inactivation of these metabolic reactions and of the flux through the pathway. On the contrary, ChIP-seq data in $mip6\Delta$ mutant did not change. Figure 5.8.a and Table 5.4 show that $mip6\Delta$ mutant has a lower number of de-acetylation changes compared to the WT. This difference between strains was not due to gene expression differences of yeast HDACs (Histone Deacetylases) as they were similarly expressed in both strains (Figure 5.8.b).

5.5 Discussion

In this chapter we have introduced MAMBA (Metabolic Adjustment via Multiomic Block Aggregation), a constraint-based genome-scale metabolic reconstruction algorithm that allows the integration of gene-associated omic data and semi-quantitative metabolomics. Compared to previous approaches, MAMBA has a better metabolite prediction accuracy, which means a more accurate metabolic network characterization. Additionally, MAMBA simultaneously models multiple conditions and can therefore be applied to the analysis of time-course data, providing a modelling framework for dynamic processes. In this work, we applied MAMBA to study metabolic regulation in two yeast strains (WT and the $mip6\Delta$ mutant) after a heat-shock treatment.

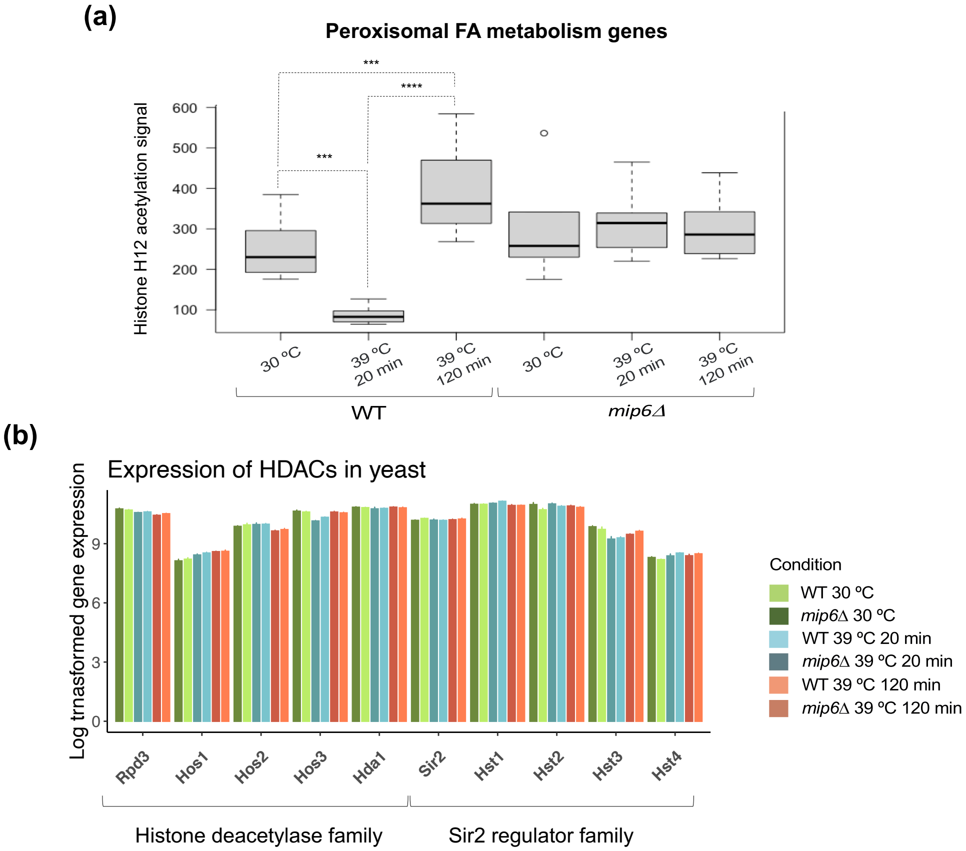


Figure 5.8: Peroxisomal Fatty acid metabolism control by ChIP-seq signal. a) ChIP-seq signal of genes involved in peroxisomal FA metabolism. WT underwent de-acetylation of those genes while no changes were present in the mutant. Statistical significance determined by Wilcoxon-Mann Whitney test [186, 187] (***: p-value $<$ .001; ****: p-value $<$ .0001) . b) Expression of the HDACs in *S. cerevisiae*. No statistically significant differences between strains were found.

The output of MAMBA recapitulates the known yeast behavior under heat stress condition. Specifically, trehalose production is one of the most important heat stress response mechanisms and behaves as a heat protector in yeast [184]. Trehalose production increases under heat stress condition [188] and has been shown to be a powerful stabilizer of proteins and membranes [189]. Only considering transcriptomics data, our data showed that trehalose production reaches its maximum at 20 minutes after the heat-shock and then it starts decreasing to show at 120 minutes similar values to the initial ones. However, trehalose is accumulated in cells according to metabolomics data and MAMBA was able to reveal this behavior. Therefore, MAMBA is able to leverage metabolomics data to improve overall metabolite prediction accuracy beyond gene expression changes. Importantly, MAMBA is also useful for driving novel biological findings. We focused on comparing both strains in terms of heat stress adaptation. Using MAMBA, we have identified the underlying mechanism that explains the differential dynamics between strains both for transcriptomic and metabolomic changes. Overall, our results suggest that the mutated strain has a lower capacity to adapt to heat stress compared to the WT. We found that Carbon metabolism (Glycolysis, Pyruvate metabolism and TCA cycle among others) was differentially regulated between strains, and in consequence, both strains adapt differently to the heat-shock treatment. Results indicate a disconnection between TCA cycle and Glycolysis in WT while the mutant requires more flux through these pathways that could be related to energetic needs. This example also demonstrates the advantages of model-driven approaches over traditional gene expression analysis. Following a standard differential expression and functional enrichment analysis, TCA cycle was not identified as being differentially regulated between the two strains (Table 4.2), possibly because only 3 out of 31 genes annotated in TCA cycle were differentially expressed, which are too few to support a significant enrichment result. However, reactions codified by those three genes are critical in the TCA cycle: Pyruvate Dehydrogenase (pyruvate to ac-CoA), Citrate Synthase (Oxoglutarate to Citrate) and Aconitase (Citrate to Isocitrate) and this critical contribution of these reactions to the pathway activity was captured by the metabolic modelling.

Moreover, the analysis at reaction level returned important differences between strains regarding heat stress adaptation. MAMBA indicated that $mip6\Delta$ mutants fail to shut the peroxisomal fatty acid metabolism down which causes an increased oxidative and heat stress. One subproduct of the peroxisomal FA degradation is Hydrogen Peroxide (H_2O_2) which generates heat when it is metabolized in the cell. This may explain why the $mip6\Delta$ continues generating reducing power after the heat-shock (Table 5.2) as well as why the mutant also produces more Trehalose (heat protective effect) during heat-shock. Additionally, the MAMBA model based on H4K12ac data indicated that the inactivation of peroxisomal FA metabolism in the WT was controlled epigenetically, which is consistent with previous studies showing that the response to heat stress is associated to changes in chromatin acetylation [113]. Globally, the ChIP-seq MAMBA model indicated that the mutant is less efficient in chromatin de-acetylation not only for peroxisomal FA metabolism genes but in general for the whole genome-scale model. Importantly, this observation in the mutant is not caused by a lower expression of HDACs as they are not under-expressed in the mutant. This opens up the question of the existence of a potential histone de-acetylation regulatory mechanism beyond gene expression of effector proteins, that might be linked to mip6.

Ethanol production during heat-shock was also different between strains. Ethanol production increased in WT at time 20min after heat-shock while it decreased in the mutant. Interestingly, $mip6\Delta$ increases lactate production while WT does not [113]. In the metabolic model used, lactate can only be produced by a reaction that also produces glutathione, which is a well known source of cellular reducing power. Additionally, ethanol production in yeast consumes reducing power in the form of NADH. Therefore, the difference in the ethanol/lactate production between strains could also be linked to peroxisomal FA metabolism. Interestingly, PAR-CLIP data on the same WT strain revealed a direct interaction between ADH1 and mip6 that could be related to the known role of mip6 as an mRNA export factor and mRNA stabilizer in the cytoplasm and an alternative explanation of ethanol profile in the mutant compared to the WT.

Proline is also able to confer heat protection to yeast cells [190], however its production decreases during heat-shock in wild type yeast [188]. Proline production increased in the $mip6\Delta$ under heat stress strengthening the hypothesis of a higher-stressed mutants compared to WT cells [113].

All together, the MAMBA model of the $mip6\Delta$ mutant heat-response suggests that the metabolism of the mutant is less flexible to environmental challenges which may be related to the roles of mip6 in the cell: i) interaction with HDACs and ii) mRNA export and stabilizer. As a consequence, mip6 knockdown results into a hyper-stressed state compared to WT and therefore the key yeast thermal protectors, trehalose and proline, are overproduced in the mutant.

In conclusion, we have demonstrated MAMBA is a powerful methodology for constructing robust metabolic networks. MAMBA outperforms other GEMs in terms of metabolite prediction accuracy and is useful to reveal patterns of metabolic control from the combination of matching transcriptomic and metabolomic data. Moreover, MAMBA allows the dynamic modeling of the system and therefore is specially powerful to analyze time-series data and compare over-time profiles among different conditions.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The main goal of this thesis was to develop new tools to advance the system biology field for the integration of multiomic data to create multi-layered systems biology models. Moreover, multiomic data generated as part of the PROMETEO project was used to support this goal and the processing and analysis of different omic data types were also an important part of this work. Additionally, we found a relevant unsolved limitation in data meta-analysis which was the lack of batch effect correction methods in multiomic experimental designs. In this section, the conclusions of this thesis are summarized according to the objectives defined in Chapter 2.

6.1.1 *Objective 1: To develop the different specific pre-processing pipelines for each omic data type.*

- We processed different omic data types that were generated within PROMETEO project: RNA-seq, Histone H4K12 acetylation ChIP-seq and Metabolomics data. Data QC and normalization were applied and batch effect was corrected as samples were collected at two different time points.
- We evaluated the quality of our multiomic dataset by assessing their reproducibility in terms of: i) technical replicability (correlation between replicates) and ii) biological consistency (capturing known yeast biology at heat stress).
- Omic datasets were submitted to public repositories [GEO (RNA-seq and ChIP-seq data; GSE135568) and MetaboLights

(Metabolomic data; MTBLS1320)] that ensures data standards and data availability for the scientific community.

6.1.2 Objective 2: To develop a batch effect correction algorithm for multiomic integration strategies.

- We developed a novel batch effect correction method, MultiBaC, for multiomic designs when all the omic data types may not have been measured for all the batches but there is a common omic measured in all of them.
- We tested MultiBaC performance on simulated multiomic data and checked that it works well for different batch effect magnitudes and requires the validation of PLS models used as part of the omic data prediction step. MultiBaC was compared to other two alternatives [**T**rimmed **S**cores **R**egression (TSR) [143, 144] and Joint-Y PLS (JY-PLS) [145]] and it showed the best performance.
- MultiBaC was compared to other batch-effect correction algorithms (BECAs) [ARSyN [54], ComBat [116, 141] and limma [53]] using real omic data where all omic data types included were measured across all the batches and therefore existing BECAs could be applied. MultiBaC showed similar performance to other methods even on non-common omic data types, for which MultiBaC does not use the batch information, unlike the other approaches.
- We also applied MultiBaC to a real multiomic dataset and demonstrated that batch correction preserves the biological signal of omic data.

- We created an R package, MultiBaC, that includes our method and it was submitted to Bioconductor which is the reference software resource for the bioinformatics community.

6.1.3 Objective 3: To develop novel multiomic integration approaches.

- We presented MAMBA, a multiomic integration method that relies on COBRA framework. MAMBA is the first CBM method that allows the use of semi-quantitative metabolomic data which is the most extended type of metabolomic data.
- MAMBA was compared to MADE [98], other method that only incorporates gene expression data and checked that the inclusion of metabolomic data into CBM boosts network characterization and metabolite prediction accuracy.
- We applied MAMBA to our yeast multiomic dataset and validated its performance by assessing how MAMBA captures known biology of heat stress in yeast.
- MAMBA was also used to extract novel biological insights of a non-characterized $mip6\Delta$ mutant yeast strain. By comparing the metabolic networks predicted by MAMBA between the mutant and WT strains, we found the main metabolic processes that are affected in the mutant.
- MAMBA was coded as a Matlab toolbox and it is freely available at github including comprehensive documentation.

6.2 Research relevance

The relevance of this thesis is highlighted in the following points:

- This thesis has been developed within the framework of competitive research project from the Conselleria dEducació, Cultura i Esport (Generalitat Valenciana), coordinated among different research groups. Thus, the tools developed have been used to analyze the data generated during this project and therefore they contributed to the dissemination of the results.
- Two open source software tools were presented in this thesis: MultiBaC R package and MAMBA for matlab. They contain the equally named methods developed in this thesis which can be used by the scientific community to different purposes: MultiBaC allows the batch effect removal in multiomic datasets and MAMBA performs a multiomic integrative analysis. The fact that our software tools are available at two well known software repositories (Bioconductor and GitHub), makes them easily accessible for potential users from the scientific community. In addition, they are properly documented allowing anyone to learn how to use them.
- Both MAMBA and MultiBaC are tools that allow the use of data in a way that was not possible until now. MultiBaC facilitates the integration of data from different batches which leads to take advantage of public data already generated in order to complete other multiomic datasets. MAMBA, on the other hand, allows the use of targeted and untargeted semi-quantitative metabolomic

data, which existing methods cannot do. This opens a way for researchers to use metabolomic data that they might have not used before. Additionally, both of them are contributions to the growing System Biology field which makes them more relevant as the number of potential users will increase in coming years.

6.3 Future research lines

This PhD dissertation opens some future lines:

Improvement of MultiBaC.
MultiBaC is a very versatile method that is built based on PLS regression models. We saw that MultiBaC's performance is highly dependent on the PLS model validity to catch data variability. However, sometimes bad PLS models can be easily modified to improve their performance via the inner relation. The inner relation in a PLS model is the type of relation assumed for X scores and Y scores. It is usually linear (as in the case of MultiBaC) but it can be transformed into any kind of relationship. In this sense, MultiBaC could be improved by allowing alternative non-linear inner relations in the PLS model.

Improvement of MAMBA.
As we have explained in Chapter 4, MAMBA constraints are based on two omic data: metabolomics plus a gene-centric omic layer. However, MAMBA's optimization function is modular and can be adapted to new layers of omic data. Yet, it would be neccessary to evaluate in deep the

impact of each layer on the final output and then formulate an omic weighting solution that could allow the method adaptation to different characteristics of different omic data types. This way of improvement adds new parameters to the model that should also be evaluated via sensitivity-robustness approach as explained in Chapter 4.

In addition, MAMBA can be coded as a toolbox for matlab or either as an R package. The latter is still not possible as COBRA toolbox is not yet available for R language (opencobra.github.io).

Experimental validations.

Biological insights obtained from MAMBA application have not been fully validated yet. Although we demonstrated that MAMBA output matches with the current knowledge of metabolic changes in yeast after heat-shock, the new findings and hypothesis need to be validated before publication to assess the reliability of MAMBA's results.

Appendix 1: Material

Hardware

The computations in this work have been carried out with a MacBook Air Intel Core i7, CPU 2,2 GHz, 8GB of RAM and the cluster system of Príncipe Felipe Reseach Center for heavier computational tasks.

Software

1. Operating systems:
   - Mac OS Mojave 10.14.6
   - UNIX cluster with Portable Batch System (PBS).
2. Programming environments:
   - MATLAB 2014a (COBRA toolbox and gurobi solver).
   - RStudio Version 1.1.463 (R version 3.6.1 and 4.0.0).
   - Python version 3.6.1.

Appendix 2: Data preprocessing (Chapter 3)

Real problem dataset

```
###############################################
#### PRE−PROCESSING REAL PROBLEM DATASETS ####
###############################################

# Raw data are available at GEO, GSE11521, GSE1002,
    GSE56622, GSE43747.

# RUN IN R USING A .R SCRIPT #
###############################
library(NOISeq)
library(limma)

##### LAB A ----------------------------------------
# Data downloaded from GEO GSE11521, GSE1002.

# Each sample is downloaded as a probe with
    an associated genome measure
# to normalize between arrays.

probe <− read.csv('probe.txt', sep = '\t')
genome <− read.csv('genome.txt', sep = '\t')

### Normalize
```

```
Ri <- apply(genom, 2, sum)
Gj <- apply(probe, 2, sum)
k <- (Ri / Gj)

probe_norm <- t(apply(probe, 1, function (x) x*k))

# Probe is either a gro or a rna sample
# This is repeated for each probe (sample)

labA_RNA <- cbind(<all probe_norm>)
     (if probes are RNA samples)
labA_GRO <- cbind(<all probe_norm>)
     (if probes are GRO samples)

##### LAB B ----------------------------------------
# Processed data is downloaded, counts in RPKM
labB <- read.csv('GSE56622_ZidProcessedDataAll.txt',
                                   quote = '',
                                   sep = '\t', header = T)
ribo_ini <- ribo_values[,c(18,19,28,29)]
rna_ini <- ribo_values[,c(26,27,30,31)]

# Quality control with NOISeq
mydata <- readData(data =<ribo_ini or rna_ini>,
              factors = data.frame(factor(c(0,1,0,1))))
### RNA composition
mycd<- dat(mydata, type = "cd", norm = TRUE,
              refColumn = 1)
explo.plot(mycd, samples = 1:4)

# voom transformation
labB_rna <- voom(rna_ini)$E
labB_ribo <- voom(ribo_ini)$E

##### LAB C -------------------------------------------------
# Processed data is downloaded, RNA-seq counts in RPKM

rna_ini <-
read.csv('GSE43747_Transcript_abundance_mRNAseq.txt',
                  quote = '', sep = '\t',
                  header = T)[,4:7]

# Quality control with NOISeq
mydata <- readData(data = rna_ini,
                  factors = data.frame(factor(c(0,1,0,1)
### RNA composition
mycd<- dat(mydata, type = "cd", norm = TRUE,
refColumn = 1)
```

```
explo.plot(mycd, samples = 1:4)

# voom transformation
labC_rna <- voom(rna_ini)$E

# gPAR-CLIP data is in coverage per region form
    and we need to get gene_related data

# Step 1: transform original data matrix in .bed format
par_ini <-
    read.csv('GSE43747_Binding_site_coverage_gPARCLIP.txt'
    header = T, sep = '\t')

# this file contains the chr, strand, start and site
    position of the peaks
# and the quantification of them for all the samples.
    We create a bed file
# per sample taking the respective columns.

write.table(<sample>, file = '<sample.bed>',
        sep = '\t', row.names = F, col.names = F,
        quote = F)

# Step 2: Map regions to Genes
# RUN IN HPC CLUSTER WITH A .SH SCRIPT #
############################################

# We use RGmatch to map the regions into
    nearest genes.
python rgmatch.py -t 0 -g  saccer3.gtf -b
    <sample.bed> -o <sample.txt>
# A region is associated to a certain gene if
    the region overlaps the TSS,
# the TTS or the gene_body.

######### VERSION OF THE GENOME USED:
Reference Saccharomyces cerevisiae:
  version: sacCer3
  source: UCSC "https://genome.ucsc.edu"
Genes: sacCer3.gtf

# RUN IN R USING A .R SCRIPT #
#################################
# Combining datasets ----------------------------
par_ini <- read.csv (<sample.txt>)
    (merging the quantification for all samples)
labC_PAR <- voom(par_ini)$E
```

```
intersection <- intersect(rownames(labA_RNA),
                          intersect(rownames(labB_rna),
                                    rownames(labC_rna)))
real_dataset <- cbind(labA_RNA[intersection,],
                      labB_RNA[intersection,],
                      labC_RNA[intersection,],
                      labA_GRO[intersection,],
                      labB_RIBO[intersection,],
                      labC_PAR[intersection,])

# Trimmed mean of M (tmm) normalization from NOISeq
is applied to remove
# systematic biases between samples.
```

Proof of concept data

```
###########################################
## PRE-PROCESSING PROOF OF CONCEPT DATA ##
###########################################

# Raw data are available at GEO, GSE33136,
    GSE24488

# RUN IN HPC CLUSTER WITH A .SH SCRIPT #
```

```
######################################################

## RNA-seq and GRO-seq data (GSE33136)---------------

## Step 1: Mapping files to reference genome

########### VERSION OF THE GENOME USED:
Reference Saccharomyces cerevisiae:
  version: sacCer3
  source: UCSC "https://genome.ucsc.edu"
Genes: sacCer3.gtf

########### VERSIONS OF THE SOFTWARE
samtools version 0.1.18
TopHat  version 2.1.0

## Step 1: Mapping reads to genome
tophat -o <outputdir> <sacCer3> <sample.fastq.gz>

## Step 2: Quantification of reads: using htseq
htseq-count -a 20 -m union <sample>.sam
    <sacCer3.gtf > <sample_counts.txt>
```

```
# RUN IN R USING A .R SCRIPT #
################################

## LIBRARIES TO USE
library(NOISeq)
library(limma)

# Get gene lengths ---------------------------------------
gff <- read.csv("saccer3_M.gtf", skip = 5,
header = F, sep = "\t")
genes <- sapply(gff$V9, function(x) {
  strsplit(strsplit(toString(x), ";")
    [[1]][1], "␣")[[1]][2]
})
gff <- data.frame(gff, genes)
gene_len <- sapply(unique(gene_len[,1]),
function(x) {
  aux <- gene_len[which(gene_len[,1]==x),,
  drop = FALSE]
  c(as.character(x), max(aux[,2]))
})
gene_len <- t(gene_len)
vgene_len <- as.numeric(gene_len[,2])
names(vgene_len) <- gene_len[,1]
```

```
vgene_len <- vgene_len[rownames(seqdata)]

# Reading file with region raw counts
raw_data = read.delim("sample_counts.txt",
header = TRUE, as.is = TRUE, sep = "\t")

# Analysis of biases using NOISeq
# (rna-seq and gro-seq data separately)

mydata <- readData(data = raw_data,
factors = data.frame(factor(c(1,
1,1,0,0,0))),
                           length = vgene_len)
### Saturation plot
mysaturation <- dat(mydata, k = 0, ndepth = 7,
type = "saturation")
explo.plot(mysaturation, toplot = 1,
samples = 1:24)

### RNA composition
mycd<- dat(mydata, type = "cd",
norm = F, refColumn = 1)
explo.plot(mycd, samples = 1:6)
```

```
### Length bias
mylenbias <- dat(mydata,
type = "lengthbias")
explo.plot(mylenbias, samples = NULL,
toplot="global")

#### CPM
mycounts <- dat(mydata, factor = NULL,
type = "countsbio")
explo.plot(mycounts, toplot=1,
samples = NULL, plottype = "barplot")

#### TMM normalization
seqdata <- tmm(assayData(mydata)$exprs,
long = vgene_len)
# (merge rna-seq and gro-seq data matrices)

# Voom transformation using limma
lab1_RNA <- voom(seqdata)$E
     (if seqdata = RNA-seq samples)
lab1_GRO <- voom(seqdata)$E
     (if seqdata = GRO-seq samples)

#### RNA and GRO (GSE24488) -----------------
```

```
# Each sample is downloaded as a probe with an
    associated genome measure
# to normalize between arrays.

probe <- read.csv('probe.txt', sep = '\t')
genome <- read.csv('genome.txt', sep = '\t')

### Normalize
Ri <- apply(genom, 2, sum)
Gj <- apply(probe, 2, sum)
k <- (Ri / Gj)

probe_norm <- t(apply(probe, 1, function (x) x*k))

# Probe is either a gro or a rna sample
# This is repeated for each probe (sample)
lab2_RNA <- cbind(<all probe_norm>)
    (if probes are RNA samples)
lab2_GRO <- cbind(<all probe_norm>)
    (if probes are GRO samples)

#### Combine datasets ----------------------------
# matrices structure = features x sample
```

```
intersection <- intersect (rownames(lab1_RNA),
                                   rownames(lab2_RNA))
proof_matrix <- cbind(lab1_RNA[intersection ,],
                             lab1_GRO[intersection ,],
                             lab2_RNA[intersection ,],
                             lab2_RNA[intersection ,])
# Trimmed mean of M (tmm) normalization from
     NOISeq is applied to remove
# systematic biases between samples.
```

Appendix 3: MultiBaC R package vignette (Chapter 3)

MultiBaC user’s guide

***Manuel Ugidos*[1]*, Sonia Tarazona*[2] *and Maria J. Nueda*[3]**

[1]Gene Expression and RNA Metabolism Laboratory, Instituto de Biomedicina de Valencia, Consejo Superior de Investigaciones Científicas (CSIC), Valencia, Spain.
[2]Multivariate Statistical Engineering Group, Department of Applied Statistics, Operations Research and Quality, Universitat Politècnica de València, Valencia, Spain
[3]Department of Mathematics, Alicante Universiy, Spain

26 April 2022

Package

MultiBaC 1.6.0

Contents

1 Introduction

Simultaneously generating multiple omic measurements (e.g. transcriptomics, metabolomics, proteomics or epigenomics) of the same molecular system for one particular study is not always possible. As a consequence, researchers sometimes combine compatible data generated in different labs or in different batches. In such cases, data will usually be affected by an unwanted effect associated to the experimentation event (lab, batch, technology, etc.) that, especially for high throughput molecular assays, may result in important levels of noise contaminating the biological signal. This unwanted source of variation is commonly known as ``batch effect'' and is very frequently seen as the first source of variability in the omic dataset, standing out over the experimental conditions under study.

Removing batch effects becomes then necessary in order to obtain meaningful results from statistical analyses. Provided that the omic experiment has been designed in such a way that batch effects are not confounded with the effects of interest (treatment, disease, cell type, etc.), the so-called Batch Effect Correction Algorithms (BECAs) can be used to remove, or at least mitigate, systematic biases.Therefore these methods are extremely useful to combine data from different laboratories or measured at different times. One of these BECAs is the ARSyN method [1], which relies on the ANOVA-Simultaneous Components Analysis (ASCA) framework to decompose the omic signal into experimental effects and other unwanted effects. ARSyN applies Principal Component Analysis (PCA) to estimate the systematic variation due to batch effect and then removes it from the original data.

BECAs have been traditionally applied to remove batch effects from omic data of the same type, as for example gene expression. However, while removing batch effects from a single omic data type with an appropriate experimental design is relatively straightforward, it can become unapproachable when dealing with multiomic datasets. In the multiomic scenario, each omic modality may have been measured by a different lab or at a different moment in time, and so it is obtained within a different batch. When this is the case, the batch effect will be confounded with the ``omic type effect'' and will be impossible to remove from the data. However, in some scenarios, the multiomic batch effect can be corrected. MultiBaC is the first BECA dealing with batch effect correction in multiomic datasets. MultiBaC can remove batch effects across different omics generated within separate batches provided that at least one common omic data type is included in all the batches.

The **MultiBaC** package includes two BECAs: the ARSyN method for correcting batch effect from a single omic data type and the MultiBac method, which deals with the batch effect problem on multi-omic assays.

2 Batch effect correction on a single omic

2.1 About ARSyN

2.1.1 ARSyN method overview

ARSyN (ASCA Removal of Systematic Noise) is a method that combines Analysis of Variance (ANOVA) and Principal Component Analysis (PCA) for the identification of structured variation from the estimated ANOVA models for experimental and unwanted effects on an omic data matrix. ARSyN can remove undesired signals to obtain noise-filtered data for further analysis [1]. In **MultiBaC** package, ARSyN has been adapted for filtering the noise associated to identified or unidentified batch effects. This adaptation has been called ARSyNbac.

In the ARSyN method, the ANOVA model separates the signal identified with each one of the factors involved in the experimental design from the residuals. The algorithm can be applied on multi-factorial experimental designs. One of the factors in the model can be the batch each sample belongs to, if this information is known. In such case, the ANOVA model is applied to separate the batch effect from the remaining effects and residuals. The PCA analysis will hence detect the possible existence of a structured variation due to the batch effect, that is identified with the principal components explaining a given proportion of the total variation in the data, which can be set by the user.

However, ARSyN can also be applied when the batch factor is not known since the PCA on the residual matrix can detect correlated structure associated to a source of variation not included in the experimental design. We alert of this signal in the residuals when the first eigenvalues of the PCA are noticeably higher than the rest, because if there is not any structure the eigenvalues will be approximately equal. In this case the selection of components is controlled by the beta argument. Components that represent more than beta times the average variability are identified as systematic noise and removed from the original data.

2.1.2 How to cite ARSyN

Nueda MJ, Ferrer A, Conesa A.(2012). ARSyN: A method for the identification and removal of systematic noise in multifactorial time course microarray experiments. *Biostatistics* 13:553-66.

2.2 Example: Yeast expression data

The yeast expression example data sets were collected from the Gene Expression Omnibus (GEO) database and from three different laboratories (batches). In all of them, the effect of glucose starvation in yeast was analyzed. Lab A is the Department of Biochemistry and Molecular Biology from Universitat de Valencia (accession number GSE11521) [2]; Lab B is the Department of Molecular and Cellular Biology from Harvard University (accession number GSE56622) [5]; and Lab C is the Department of Biology from Johns Hopkins University (accession number GSE43747) [6].

After a proper data pre-processing for each case, a voom transformation (with limma R package) was applied when necessary. Finally TMM normalization was performed on the whole set of samples from all labs. A reduced dataset was obtained by selecting 200 omic variables from each data matrix and just 3 samples from lab A. This yeast multiomic reduced dataset is included in *MutiBaC* package to illustrate the usage of the package. The gene expression matrices can be loaded by using the *data("multiyeast")* instruction.

The three studies used equivalent yeast strains and experimental conditions but, as shown in Figure 1 , the main effect on expression is due to data belonging to different labs, which are the batches in this case.

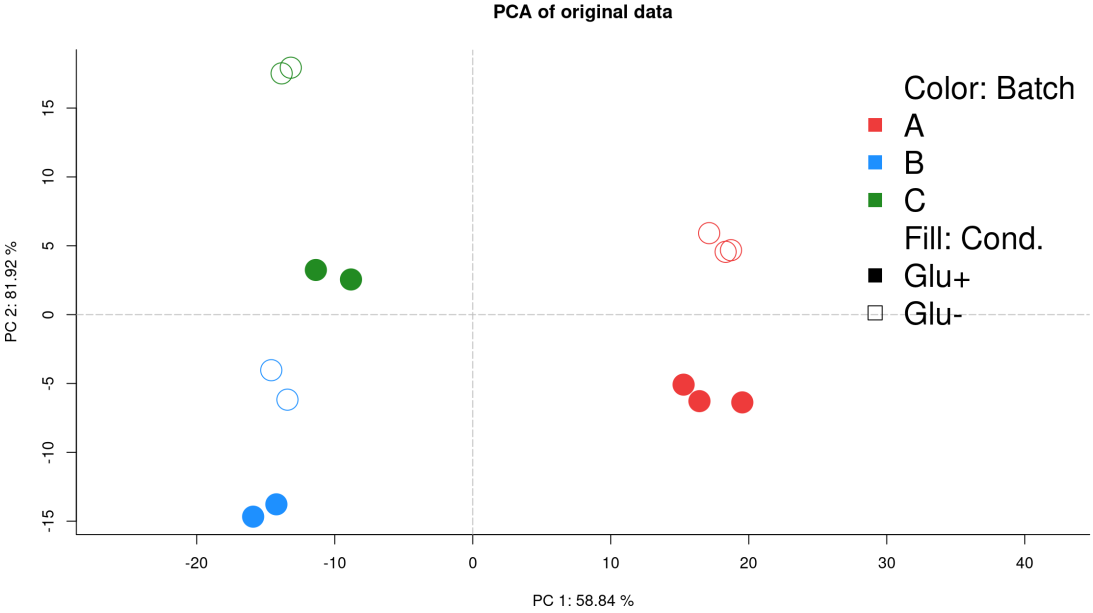


Figure 1: **PCA plot of original gene expression data (before correction)**
Batches are completely separated from each other. Plot generated with MultiBaC package (see Visualization of results Section).

2.3 ARSyNbac input data

The **MultiBaC** package uses *MultiAssayExperiment* objects, a type of Bioconductor container for multiomic studies, that can be created from a list of matrices or *data.frame* objects. These matrices must have features in rows and samples in columns as shown next for one of the data matrices from the yeast example (A.rna). It is important that all data matrices share the variable space. If the number of omic variables and order are not the same, the *createMbac* function will select the common variables. Hence, it is mandatory that rows are named with the same type of identifiers.

```
data("multiyeast")
head(A.rna)
```

```
##         A.rna_Glu+_1 A.rna_Glu+_2 A.rna_Glu+_3 A.rna_Glu-_1 A.rna_Glu-_2
## YOR324C     7.174264     6.976815     7.482661     8.020596     7.736636
## YGL104C     4.239493     4.284775     3.100898     4.957403     3.673252
## YOR142W     8.819824     8.496966     9.026971    10.374525    10.294006
## YOR052C     6.721211     7.011932     7.557519     8.504503     8.586738
## YGR038W     5.878483     5.894121     6.468361     7.856822     7.806318
## YER087W     5.131089     5.295029     5.750657     2.920250     5.496136
##         A.rna_Glu-_3
## YOR324C     7.922425
## YGL104C     1.587184
## YOR142W    10.979224
## YOR052C     8.221168
## YGR038W     8.108436
## YER087W     2.879254
```

A *MultiAssayExperiment* object needs to be created for each batch (lab in this example). The *mbac* new data structure is a list of *MultiAssayExperiment* objects and can be easily generated with the *createMbac* function in the package. The resulting *mbac* object will be the *ARSyNbac* input.

```
data_RNA<- createMbac (inputOmics = list(A.rna, B.rna, C.rna),
                    batchFactor = c("A", "B", "C"),
                    experimentalDesign = list("A" =  c("Glu+", "Glu+",
                                                "Glu+", "Glu-",
                                               "Glu-", "Glu-"),
                                       "B" = c("Glu+", "Glu+",
                                               "Glu-", "Glu-"),
                                       "C" = c("Glu+", "Glu+",
                                               "Glu-", "Glu-")),
                    omicNames = "RNA")
```

These are the arguments for the *createMbac* function:

- **inputOmics** A list containing all the matrices or data.frame objects to be analysed. MultiAssayExperiment objects can alternatively be provided.
- **batchFactor** Either a vector or a factor indicating the batch were each input matrix belongs to (i.e. study, lab, time point, etc.). If NULL (default) no batch is considered and just ARSyNbac noise reduction mode could be applied.
- **experimentalDesign** A list with as many elements as batches. Each element can be a factor, a character vector or a data.frame indicating the experimental conditions for each sample in that batch. When being a data.frame with more than one column (multi-factorial experimental designs), the different columns will be combined into a single one to be used by MultiBaC or ARSyNbac. In any case, the experimental setting must be the same for all batches. In addition, the names of the elements in this list must be the same as declared in *batches* argument. If not (or if NULL), names are forced to be the same in as in *batches* argument and in the same order.
- **omicNames** Vector of names for each input matrix. The common omic is required to have the same name across batches.
- **commonOmic** Name of the common omic between the batches. It must be one of the names in omicNames argument. If NULL (default), the omic name which is common to all batches is selected as commonOmic.

The *mbac* R structure generated by the *createMbac* function is an S3 object that initially contains just one slot, the *ListOfBatches* object. This *mbac* structure will incorporate more elements as they are created when running *ARSyNbac* or *MultiBaC* functions. These new slots are *CorrectedData*, *PLSmodels*, *ARSyNmodels* or *InnerRelation* and are described next:

- **ListOfBatches**: A list of MultiAssayExperiment objects (one per batch).
- **CorrectedData**: Same structure than ListOfBatches but with the corrected data matrices instead of the original ones.
- **PLSmodels**: PLS models created by MultiBaC method (one model per non-common omic data type). Only available for MultiBaC method.
- **ARSyNmodels**: ARSyN models created either by ARSyNbac or MultiBaC functions.
- **InnerRelation**: Table of class *data.frame* containing the inner correlation (i.e. correlation between the scores of X (t) and Y (u) matrices) for each PLS model across all components, for model validation purposes. Only available for MultiBaC method.
- **commonOmic**: Name of the common omic between the batches.

In addition to plot, other method is supplied for visualizing mbac objects: summary, which show the structure of the object.

```
summary(data_RNA)
```

```
## [1] "Object of class mbac: It contains 3 different bacthes and 1 omic type(s)."
```

	RNA
A	TRUE
B	TRUE
C	TRUE

2.4 ARSyNbac correction

The function to remove batch effects or unwanted noise from a single omic data matrix in the **MultiBaC** package is the *ARSyNbac* function, which allows for the following arguments:

```
ARSyNbac (mbac, batchEstimation = TRUE, filterNoise = TRUE,
          Interaction=FALSE, Variability = 0.90, beta = 2,
          modelName = "Model 1",
          showplot = TRUE)
```

- **mbac**: mbac object generated by *createMbac*.
- **batchEstimation**: Logical. If TRUE (default) the batch effect is estimated and used to correct the data. Use TRUE when the source of the batch effect is known.
- **filterNoise^**: Logical. If TRUE (default) structured noise is removed form residuals. Use this option when there is an unknown source of batch effect in data.
- **Interaction**: Logical. Whether to model the interaction between factors or not (FALSE by default).
- **Variability**: From 0 to 1. Minimum percent of data variability that must be explained by each model. Used in batch correction mode. By default, 0.90.
- **beta**: Numeric. Components that represent more than beta times the average variability are identified as systematic noise in residuals. Used in noise reduction mode. By default, 2.
- **modelName**: Name of the model created. This name will be showed if you use the explaine_var plot function. By default, “Model 1”.
- **showplot**: Logical. If TRUE (default), the explained_var plot is showed. This plot represents the number of components selected for the ARSyN model.

Therefore, the *ARSyNbac* function offers three types of analysis: **ARSyNbac batch effect correction**, when the batch information is provided, **ARSyNbac noise reduction**, if the batch information is unknown, and the combination of both modes when there is a known source of batch effect and another possible unknown source of unwanted variability. In the following sections we explain how to proceed with each one of them.

2.4.1 ARSyNbac batch effect correction

When the batch is identified in the *batchFactor* argument of the *mbac* input object, its effect can be estimated and removed by choosing *batchEstimation = TRUE* (considering one source of batch effect only, *filterNoise = FALSE*). Moreover, a possible interaction between the experimental factors and the batch factor can be studied by setting *Interaction=TRUE*.

```
par(mfrow = c(1,2))
arsyn_1 <- ARSyNbac(data_RNA, modelName = "RNA", Variability = 0.95,
                 batchEstimation = TRUE, filterNoise = FALSE, Interaction = FALSE)
plot(arsyn_1, typeP="pca.cor", bty = "L",
     pch = custom_pch, cex = 3, col.per.group = custom_col,
     legend.text = c("Color: Batch", names(data_RNA$ListOfBatches),
                     "Fill: Cond.", unique(colData(data_RNA$ListOfBatches$A)$tfactor)),
     args.legend = list("x" = "topright",
                        "pch" = c(NA, 15, 15, 15,
                                  NA, 15, 0),
                        "col" = c(NA, "brown2", "dodgerblue", "forestgreen",
                                  NA, 1, 1),
                        "bty" = "n",
                        "cex" = 1.2))
```

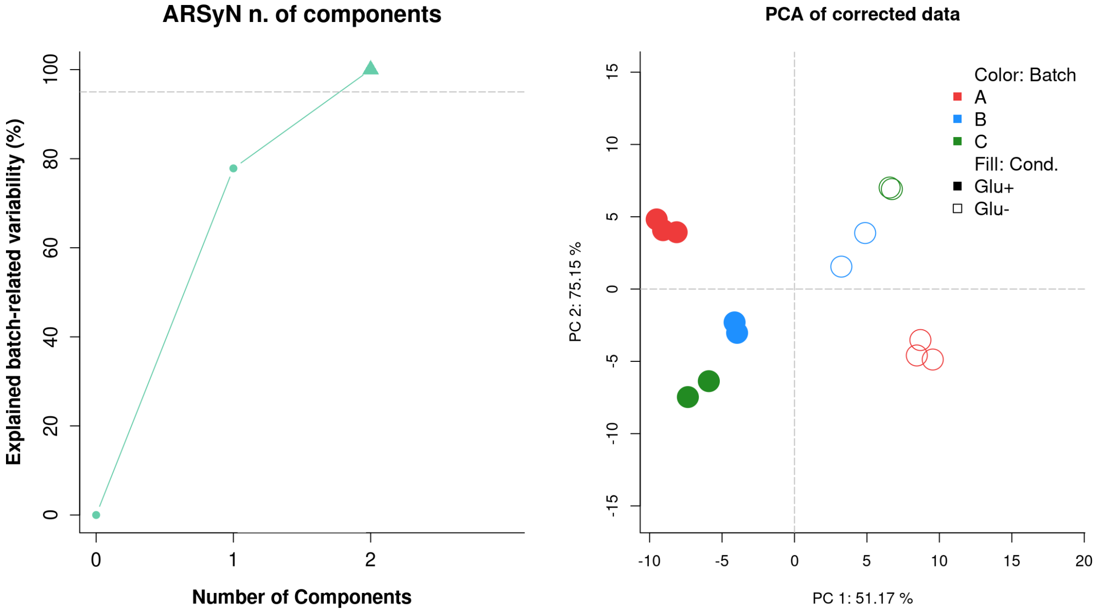


Figure 2: **Batch correction when Interaction=FALSE**
Left: Explained variance plot. Default plot when showplot = TRUE. It represents the number of components (x axis) needed to explain a certain variability (y axis) of the effect of interest (batch effect). The number of components selected in the model is indicated with a triangle symbol. Gray dashed line indicates the threshold given by the Variability argument (in percentage). Right: PCA plot of corrected gene expression data when not considering the interaction batch-condition.

According to the left plot in Figure 2, two principal components (PCs) have been selected to explain at least 95% of the total variability of batch effect. PCA of corrected data with this analysis is shown in the right panel. Now the main source of variability in the data (PC1) is given by the experimental condition, while samples are not clustered by batches anymore.

```
par(mfrow = c(1,2))
arsyn_2 <- ARSyNbac(data_RNA, modelName = "RNA", Variability = 0.95,
                  batchEstimation = TRUE, filterNoise = FALSE, Interaction = TRUE)
plot(arsyn_2, typeP="pca.cor", bty = "L",
     pch = custom_pch, cex = 3, col.per.group = custom_col,
     legend.text = c("Color: Batch", names(data_RNA$ListOfBatches),
                     "Fill: Cond.", unique(colData(data_RNA$ListOfBatches$A)$tfactor)),
     args.legend = list("x" = "topright",
                        "pch" = c(NA, 15, 15, 15,
                                  NA, 15, 0),
                        "col" = c(NA, "brown2", "dodgerblue", "forestgreen",
                                  NA, 1, 1),
                        "bty" = "n",
                        "cex" = 1.2))
```

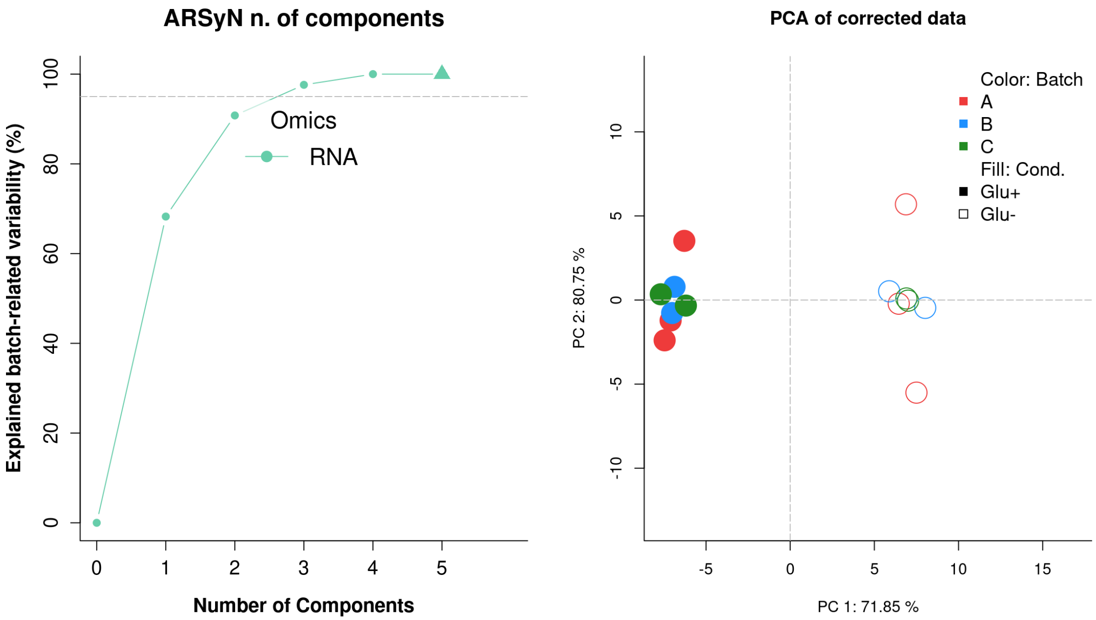


Figure 3: **Batch correction when Interaction = TRUE**
Left: Explained variance plot. Default plot when showplot = TRUE. It represents the number of components (x axis) needed to explain a certain variability (y axis) of the effect of interest (batch effect). The number of components selected in the model is indicated with a triangle symbol. Gray dashed line indicates the threshold given by the Variability argument (in percentage). Right: PCA plot of corrected gene expression data considering the interaction batch-condition.

In Figure 3 (right panel) all the points with negative PC1 correspond to Glu- samples, and the positive PC1 to Glu+ samples, as happened when not including the interaction in the model (Figure 2, right panel). However, in this second model, PC1 explains a higher percentage of the variability in the data, indicating a better batch effect correction. In general, we recomend the use of the default argument (Interaction=FALSE), as including part of the signal as batch effect could lead to a dilution of the effect of the signal of interest. However, the interaction between batch and experimental condition is sometimes strong and we should consider to include it in the model in order to get a better correction of the data.

The PCA plots shown in Figures 1, 2 and 3 have been created by the customized *plot* function in **MultiBaC** package. More details about this function can be found at the “Visualization of ARSyN and MultiBaC results” section, where a complete description of the arguments in *plot* is given.

2.4.2 ARSyNbac noise reduction

When batch is not identified, **ARSyNbac** analyses the existence of systematic noise in the residuals by setting *batchEstimation = FALSE* and *filterNoise = TRUE*.

```
par(mfrow = c(1,2))
arsyn_3 <- ARSyNbac(data_RNA, modelName = "RNA", beta = 0.5,
                    batchEstimation = FALSE, filterNoise = TRUE)
plot(arsyn_3, typeP="pca.cor", bty = "L",
     pch = custom_pch, cex = 3, col.per.group = custom_col,
     legend.text = c("Color: Batch", names(data_RNA$ListOfBatches),
                     "Fill: Cond.", unique(colData(data_RNA$ListOfBatches$A)$tfactor)),
     args.legend = list("x" = "topright",
                        "pch" = c(NA, 15, 15, 15,
                                  NA, 15, 0),
                        "col" = c(NA, "brown2", "dodgerblue", "forestgreen",
                                  NA, 1, 1),
                        "bty" = "n",
                        "cex" = 1.2))
```

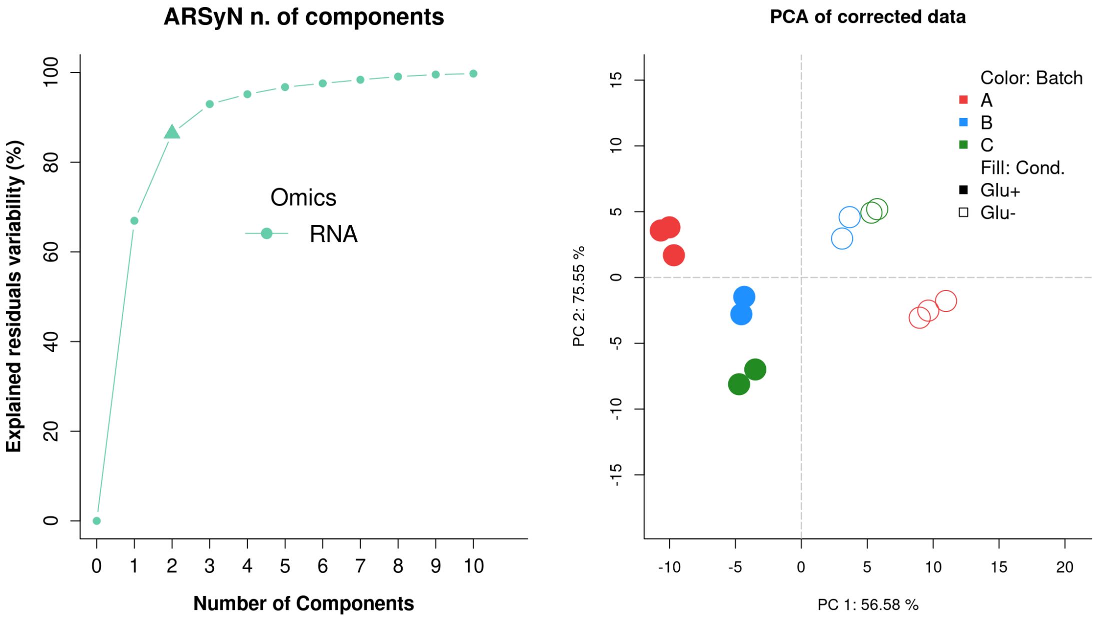


Figure 4: **Noise reduction mode**
Left: Explained variance plot. Default plot when showplot = TRUE. It represents the percentage of variability in the residuals (y axis) explained by a model with a given number of principal components (x axis). The number of selected components in the final model is indicated with a triangle symbol, and computed to explain beta times the average variability of the residuals. Right: PCA plot of corrected gene expression data.

In Figure 4 we can see that even though the batch is considered unidentified (batchEstimation=FALSE), ARSyN has removed the noise from the data by estimating the main source of unwanted variation. In this noise reduction mode, we can modulate the magnitude of the residual noise removal with the beta parameter. Basically, components that represent more than beta times the average variability are identified as systematic noise in residuals (3 components were selected in this model). Thus a greater beta value leads to the selection of a lower number of components in the residuals.

2.4.3 ARSyNbac both modalities

When the source of the batch effect is known but there might be an extra unknown source of unwanted variability, **ARSyNbac** can perform both previous ways by setting *batchEstimation = TRUE* and *filterNoise = TRUE*. Note that this mode could also be useful if the known batch effect does not represent the main source of noise in our data.

```
par(mfrow = c(1,2))
arsyn_4 <- ARSyNbac(data_RNA, modelName = "RNA", beta = 0.5,
                 batchEstimation = TRUE, filterNoise = TRUE,
                 Interaction = TRUE,
                 Variability = 0.95)
plot(arsyn_4, typeP="pca.cor", bty = "L",
     pch = custom_pch, cex = 3, col.per.group = custom_col,
     legend.text = c("Color: Batch", names(data_RNA$ListOfBatches),
                     "Fill: Cond.", unique(colData(data_RNA$ListOfBatches$A)$tfactor)),
     args.legend = list("x" = "topright",
                        "pch" = c(NA, 15, 15, 15,
                                  NA, 15, 0),
                        "col" = c(NA, "brown2", "dodgerblue", "forestgreen",
                                  NA, 1, 1),
                        "bty" = "n",
                        "cex" = 1.2))
```

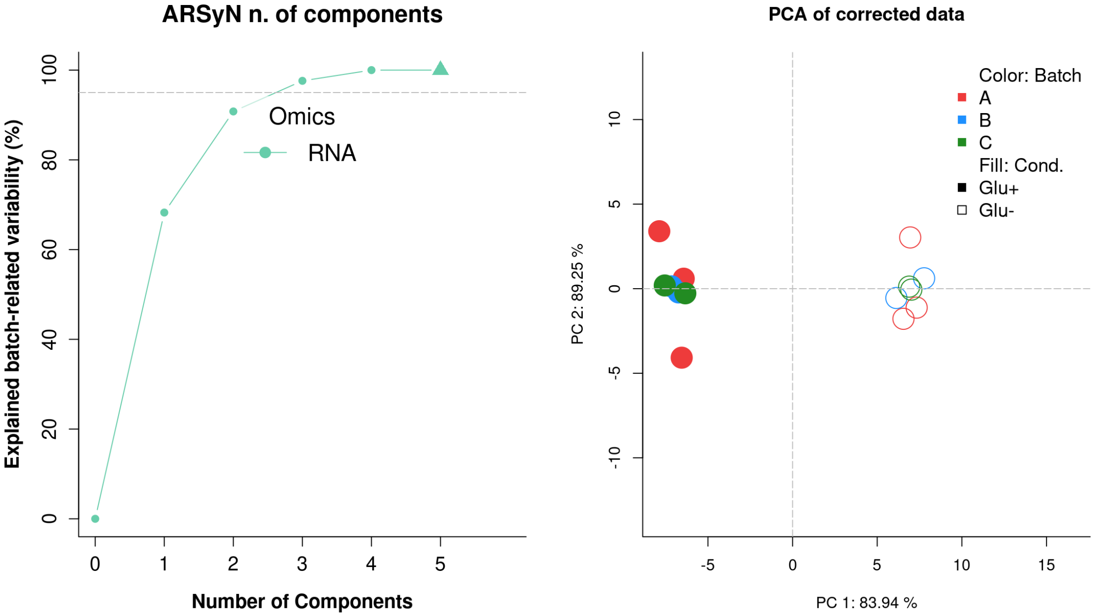


Figure 5: **Both modes together**
Left: Explained variance plot. Default plot when showplot = TRUE. It represents the percentage of variability in the residuals (y axis) explained by a model with a given number of principal components (x axis). The number of selected components in the final model is indicated with a triangle symbol, and computed to explain beta times the average variability of the residuals. Right: PCA plot of corrected gene expression data.

In Figure 5 we can see that performing both modalities together, ARSyNbac reaches its maximum PC1 variance explained (specially compared to 3). In this third mode, we can modulate the magnitude of the residual noise removal with the beta parameter and the batch effect associated explained variance. In this example, as shown in 4, known batch effect represents the main (and almost unique) source of unwanted variatio. Thus, in other scenarios with more than one batch source, the result with this third mode might be very different form the other two previous ways of operation.

3 Batch effect correction on a multiomic dataset

3.1 About MultiBaC

3.1.1 MultiBaC method overview

Multiomic data integration has become a popular approach to understand how biological systems work. However, generating this kind of datasets is still costly and time consuming. Consequently, it is quite common that not all the samples or omic data types are produced at the same time or in the same lab, but in different batches. In addition, when research groups cannot produce their own multiomic datasets, they usually collect them from different public repositories and, therefore, from different studies or laboratories (again, from different batches). Thus, in both situations, batch effects need to be previously removed from such datasets for successful data integration. Methods to correct batch effects on a single data type cannot be applied to correct batch effects across omics and, hence, we developed the MultiBaC strategy, which corrects batch effects from multiomic datasets distributed across different labs or data acquisition events.

However, there are some requirements for the multi-omic data set in order to remove across-omics batch effects with MultiBaC:

- There must be, at least, one common omic data type in all the batches. We may have, for instance, gene expression data measurements in all the batches, and then other different omic data types in each batch. It is not necessary that the commom omic (e.g. gene expression) is measured with the same technology. We could have microarray expression data in one batch and RNA-seq data in another batch, for example.
- The omic feature identifiers must be the same for all the common data matrices. We cannot use, for instance, Ensembl identifiers in one batch and RefSeq in another batch. It is not necessary to have the same number of omic features, e.g. genes, in all the batches. MultiBac will extract the common identifiers from all the common data type matrices to perform the analysis.

- Within the same batch, the experimental design must be the same for all the omic in that batch, that is, the same experimental groups, with the same number of replicates obtained from the same individuals. All these samples must be in the same order in all the omic data matrices.

3.1.2 How to cite MultiBaC

Ugidos M, Tarazona S, Prats-Montalb'an JM, Ferrer A, Conesa A.(2020). MultiBaC: A strategy to remove batch effects between different omic data types. *Statistical Methods in Medical Research* .

3.2 Example: Yeast multiomic dataset

The yeast multiomic dataset comes from the same studies that the yeast expression data described in ARSyNbac section. While the three labs produced gene expression data (RNA), each of them generated a different additional omic data type. Lab A collected transcription rates (GRO, with accession number GSE1002) [2]. Lab B obtained translation rates (RIBO, with accession number GSE56622) [5]. Finally, Lab C measured global PAR-CLIP data (PAR-CLIP, with accession number GSE43747) [6]. Therefore, labs have one shared (RNA) and one distinct (GRO, RIBO and PAR-CLIP, respectively) data types. This distributed multiomic scenario represents the type of correction problem MultiBaC addresses. A scheme of the data structure is shown in Figure 6.

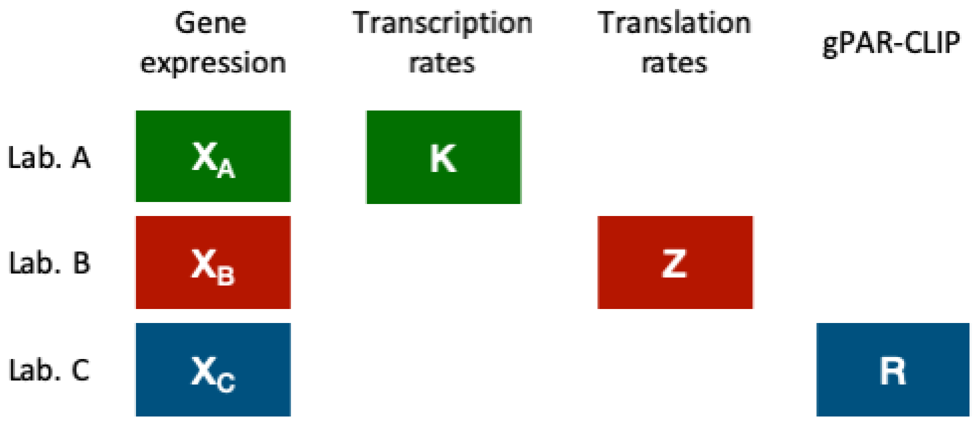


Figure 6: **Scheme of the yeast example data structure**

This yeast multiomic dataset is included in the **MutiBaC** package to illustrate the usage of the package. The six matrices can be loaded by using the *data("multiyeast")* instruction.

3.3 MultiBaC input data

As commented before, the **MultiBaC** package uses *MultiAssayExperiment* objects, a type of Bioconductor container for multiomic studies, that can be created from a list of matrices or *data.frame* objects. These matrices must have features in rows and samples in columns (see example in ARSyNbac section). Since *MultiBaC* computes regression models between omics from the same batch, it is important that matrices from the same batch have the same experimental design: the same number of samples and in the same order. *MultiBaC* relates the commonOmic information from the different batches as well. Thus, it is also important that *commonOmic* matrices share the variable space. In this case, if the number of omic variables and order are not the same, *MultiBaC* will take the common variables for the model. Hence, it is mandatory that rows are named with the same type of identifiers.

The *mbac* object that will be the *MultiBaC* function input can be easily generated with the *createMbac* function in the package:

```
my_mbac <- createMbac (inputOmics = list(A.rna, A.gro,
                                          B.rna, B.ribo,
                                          C.rna, C.par),
                    batchFactor = c("A", "A",
                                    "B", "B",
                                    "C", "C"),
                    experimentalDesign = list("A" =  c("Glu+", "Glu+",
                                                        "Glu+", "Glu-",
                                                       "Glu-", "Glu-"),
                                               "B" = c("Glu+", "Glu+",
                                                       "Glu-", "Glu-"),
                                               "C" = c("Glu+", "Glu+",
                                                       "Glu-", "Glu-")),
                    omicNames = c("RNA", "GRO",
                                  "RNA", "RIBO",
                                  "RNA", "PAR"))
```

More details about the *createMbac* function can be found in ARSyNbac input data Section. Note that we do not need to indicate which is the common omic (in commonOmic argument) since there is only one omic in common (RNA) for all the batches (labs) and the function detects it as the common omic.

3.4 MultiBaC correction

Once the *mbac* object has been created with the *createMbac* function, it is used as the input data for *MultiBaC* function (*mbac* argument), which is the wrapper function for correction of multiomic batch effects.

```
MultiBaC (mbac,
          test.comp = NULL, scale = FALSE,
          center = TRUE, crossval = NULL,
          Interaction = FALSE,
          Variability = 0.90,
          showplot = TRUE,
          showinfo = TRUE)
```

The arguments of the *MultiBaC* function correspond to the different steps of the MultiBaC method:

- **mbac**: mbac object generated by *createMbac*.
- **test.comp**: Maximum number of components allowed for PLS models. If NULL (default), the minimal effective rank of the matrices is used as the maximum number of components.
- **scale**: Logical. Whether X and Y matrices must be scaled. By default, FALSE.
- **center**: Logical. Whether X and Y matrices must be centered. By default, TRUE.
- **crossval**: Integer: number of cross-validation segments. The number of samples (rows of ‘x’) must be at least >= crossvall. If NULL (default), a leave-one-out crossvalidation is performed.
- **Interaction**: Logical. Whether to model the interaction between experimental factors and bacth factor in ARSyN models. By default, FALSE.
- **Variability**: From 0 to 1. Minimum percent of data variability that must be explained for each ARSyN model. By default, 0.90.
- **showplot**: Logical. If TRUE (default), the Q2 and the explained variance plots are shown.
- **showinfo**: Logical. If TRUE (default), the information about the function progress is shown.

The usage of MultiBaC function on the yeast example data is shown bellow:

```
my_final_mbac <- MultiBaC (my_mbac,
                           test.comp = NULL, scale = FALSE,
                           center = TRUE, crossval = NULL,
                           Interaction = TRUE,
                           Variability = 0.9,
                           showplot = TRUE,
                           showinfo = TRUE)
```

```
## Step 1: Create PLS models
```

```
##   - Model for batch A
```

```
##   - Model for batch B
```

```
##   - Model for batch C
```

```
## Step 2: Generating missing omics
```

```
## Step 3: Batch effect correction using ARSyNbac
```

```
## [1] "Table: Inner correlation between scores for each PLS model computed."
##
##
##                      A: GRO      B: RIBO       C: PAR
## -------------   ----------   ----------   ----------
## Component: 1     0.9471967    0.9987790    0.9396986
## Component: 2     0.9277123    0.9991239    0.9998556
## Component: 3     0.9992129    1.0000000    1.0000000
## Component: 4     0.9999867           NA           NA
## Component: 5     1.0000000           NA           NA
```

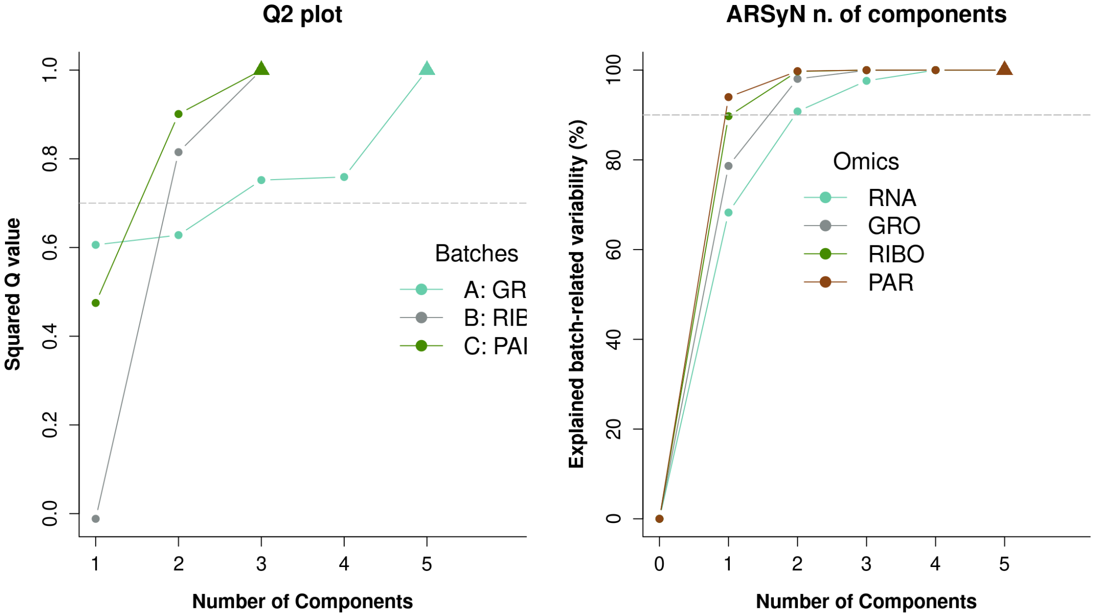


Figure 7: **Q2 and explained variance plots**
Q2 plot (left) shows the number ob components (x) needed to reach a certain Q2 value (y). The number of components selected for each model is indicated with a triangle symbol. Gray dashed line indicates the 0.7 Q2 threshold. Explained variance plot (right) represents the number of components (x) needed to explain a certain varibility (y) of the effect of interest (batch effect). The number of components selected for each model is indicated with a triangle symbol. Gray dashed line indicates the Variability argument in percentage.

By default (*showinfo = TRUE*), the table containing the inner correlations of PLS models is displayed in propmt. Moreover, the default plots (*showplot = TRUE*) are “Q2 plot” and “Explained variance plot” (see Figure 7), which contain important information about *MultiBaC* performance. The “Q2 plot” represents the PLS model prediction capability given by the \(Q^2\) score. The x axis indicates the number of components extracted for the PLS models and the y axis the \(Q^2\) value. The performance of the MultiBaC method will be better for higher \(Q^2\) values, since a high \(Q^2\) indicates a good PLS prediction of the missing omics and hence will result in a better estimation of the batch effect. Note that, depending on the rank of the matrices, each PLS model could have a different maximum number of components. The

“Explained variance plot” provides the batch effect related variability explained using the ASCA decomposition that ARSyN method provides. The x axis indicates the number of components extracted for the ASCA model and y axis reflects the percentage of explained variance. In this case, a higher explained variance leads to a better batch effect estimation. In both plots, the number of components selected for each model is indicated with a triangule symbol. In the “Q2 plot”, the selected number of components is the one that maximize the Q2 value while in the “Explained variance plot”, this number is the minimum number of components that reaches a explained variability (y axis) higher than the *Variability* argument of the function (gray dashed line).

3.5 Running MultiBaC step by step

All the different steps performed by the *MultiBaC* wrapper function can be independently computed with specific functions, as described next, from the initial *mbac* object. The MultiBaC strategy can be divided into three main steps that will be described next in detail: 1) PLS model fitting, 2) Prediction of missing omics, and 3) Batch effect correction.

3.5.1 PLS model fitting

The *genModelList* function produces the PLS models between distinct and common omic data types. It computes the optimal number of components via a crossvalidation approach.

```
my_mbac_2 <- genModelList (my_mbac, test.comp = NULL,
                          scale = FALSE, center = TRUE,
                          crossval = NULL,
                          showinfo = TRUE)
```

```
##   - Model for batch A
```

```
##   - Model for batch B
```

```
##   - Model for batch C
```

The arguments of the *genModelList* function are:

- **mbac**: mbac type object.
- **test.comp** Maximum number of components allowed in PLS models. If NULL (default), the minimal effective rank of the matrices is used as the maximum number of components.
- **scale**: Logical. Whether X and Y matrices must be scaled. By default, FALSE.
- **center**: Logical. Whether X and Y matrices must be centered. By default, TRUE.
- **crossval** Integer indicating the number of cross-validation segments. The number of samples (rows of ‘x’) must be at least >= crossvalI. If NULL (default), a leave-one-out crossvalidation is conducted.
- **showinfo**: A logical value indicating whether to show the information about the function progress. By default, TRUE.

The output of *genModelList* is a *mbac* object with a new slot, *PLSmodels*, a list of the PLS models obtained with the *ropls* package. Each slot of the output corresponds to a batch in *ListOfBatches*. If one batch contains more than one non-common omic, the “batch” element in *genModelList* contains in turn as many elements as non-common omics in that batch, i.e. one PLS model per non-common omic.

3.5.2 Prediction of missing omics

The prediction of the initially missing omics is performed with the *genMissingOmics* function from the output of the *genModelList* function.

```
multiBatchDesign <- genMissingOmics(my_mbac_2)
```

The result after running *genMissingOmics* is a list of *MultiAssayExperiment* structures. In this case, each batch contains all the omics introduced in MultiBaC. For instance, if two batches are being studying, “A” and “B”, given that “A” contains “RNA-seq” and “GRO-seq” data and “B” contains “RNA-seq” and “Metabolomics” data, after applying *genMissingOmics* function, batch “A” will contain “RNA-seq”, “GRO-seq” and also the predicted “Metabolomics” data.

3.5.3 Batch effect correction

```
my_finalwise_mbac <- batchCorrection(my_mbac_2,
                                     multiBatchDesign = multiBatchDesign,
                                     Interaction = TRUE,
                                     Variability = 0.90)
```

As described before, ARSyN applies an ANOVA-like decomposition to the data matrix in order to estimate the batch effect and, next, a PCA is applied on each submatrix. The number of principal components for each PCA is adjusted by the *Variability* argument. The output of this function consists of two different objects: *CorrectedData* and *ARSyNmodels*. The first one has the same structure than *ListOfBatches* slot. However, in this case, all batches contain all the omics introduced in MultiBaC after correcting the batch effect on each omic data type separately. Note that we discard the predicted omic matrices and only use the corrected original matrices for further statistical analyses. The *ARSyNmodels* slot contains the ASCA decomposition models for each omic data type.

4 Visualization of ARSyN and MultiBaC results

As mentioned before, the *ARSyNbac* and *MultiBaC* outputs are *mbac* type objects. Since the *mbac* class incorporates a plotting method, the *plot function* can by applied on *mbac* objects to graphically display additional information about the performance of the methods and the data characteristics. The *plot* function for *mbac* objects accepts several additional arguments:

```
plot (x, typeP = "def",
      col.by.batch = TRUE,
      col.per.group = NULL,
      comp2plot = c(1,2),
      legend.text = NULL,
      args.legend = NULL, ...)
```

Description of the arguments:

- **x**: Object of class “mbac” returned by *MultiBaC* method.
- **typeP**: The type of plot to be displayed. Options are: “def” (default option, “Q2 plot” and “Explained variance plot” in case of MultiBaC and “Explained variance plot” in case of ARSyNbac outputs), “inner” (inner correlation plots for each PLS model acroos the components for MultiBaC output), “pca.org” (PCA plot of original data for MultiBaC or ARSyNbac outputs), “pca.cor” (PCA plot of corrected data for MultiBaC or ARSyNbac outputs), “pca.both” (PCA plots for both original and corrected data for MultiBaC or ARSyNbac outputs), and “batch” (“Batch effect estimation” plot for all the outputs). Remember that PCA plots can only be generated when all the omics share the same variable space (e.g. gene identifiers are provided as names of variables for all data matrices).
- **col.by.batch**: Argument for PCA plots. TRUE or FALSE. If TRUE (default), samples are colored according to the batch factor. If FALSE, samples are colored according to the experimental conditions.
- **col.per.group**: Argument for PCA plots. Color for each group (given by batches or experimental conditions). If NULL (default), the colors are taken from a predefined pallete.
- **comp2plot**: Argument for PCA or InnerRel plot. It indicates which components are to be plotted. The default is c(1,2), which means that, in PCA plots, component 1 is plotted in “x” axis and component 2 in “y” axis, and for InnerRel plots, the inner relation plots of components 1 and 2 are to be shown. If more components are indicated, the function will return as many plots as needed to show all the components.
- **legend.text**: A vector of text used to construct a legend for the plot. Argument for PCA plot. If NULL (default) batch or conditions names included in the mbac object are used.
- **args.legend**: list of additional arguments to pass to legend(); names of the list are used as argument names. Only used if legend.text is supplied.
- **...**: Other graphical arguments.

While the *plot* function can generate all the plot types described above, each plot can also be independently generated by its corresponding function: *Q2_plot (mbac)*, *explained_varPlot (mbac)*, *plot_pca (mbac, typeP = c(“pca.org”, “pca.cor”, “pca.both”), col.by.batch, col.per.group, comp2plot, legend.text, args.legend)*, *batchEstPlot (mbac)*, or *inner_relPlot (mbac, comp2plot = c(1,2))*.

All these plots are useful to validate and understand *MultiBaC* or *ARSyNbac* performance. All of them can be used with a *MultiBaC* output, while those that show information related to the PLS models are not available for an *ARSyNbac* output (see *typeP* argument).

In addition to “Q2 plot” and “Explained variance plot” (Figure 7), which are the default plots, and were explained in previous sections, next sections are devoted to describe the rest of plots in **MultiBaC** package.

4.1 Inner correlation in PLS models

An important aspect to be validated in MultiBaC is the inner correlation between X and Y scores in PLS models. As we indicated before, *MultiBaC* function displays by default this information as numerical output but a visual representation can also be invoked.

```
plot (my_final_mbac, typeP = "inner", comp2plot = c(1,2))
```

```
## Hit <Return> to see next plot:
```

```
## Inner correlation of scores. Batch: A; Model for omic: GRO
```

```
## Warning in par(initpar): graphical parameter "cin" cannot be set
```

```
## Warning in par(initpar): graphical parameter "cra" cannot be set
```

```
## Warning in par(initpar): graphical parameter "csi" cannot be set
```

```
## Warning in par(initpar): graphical parameter "cxy" cannot be set
```

```
## Warning in par(initpar): graphical parameter "din" cannot be set
```

```
## Warning in par(initpar): graphical parameter "page" cannot be set
```

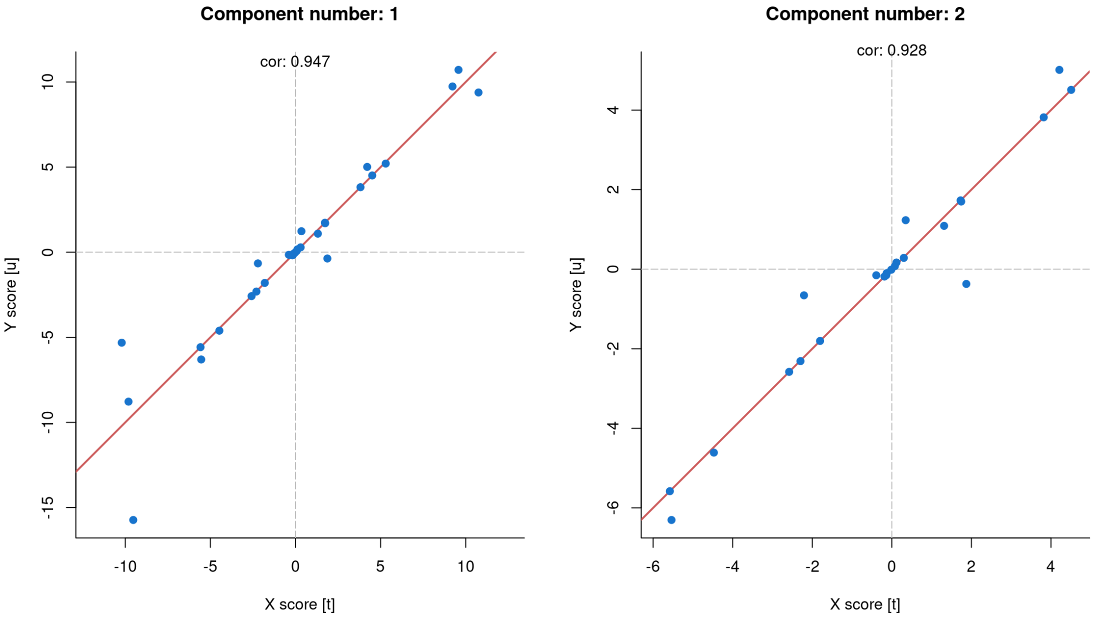


Figure 8: **Plot of inner relations of PLS components**
Only results for batch ‘A’ are shown as example. Each panel represents the inner correlation of one component of the PCA model. Red line indicates the diagonal when the correlation is maximal (1:1).

The inner correlation between scores of the PLS model that relates both omic data types in batch “A” is shown in Figure 8. While we have only shown the plot for batch “A”, running *plot* (typeP = “inner”)}, the inner correlation plots for all the PLS models generated during MultiBaC performance are displayed using the tag *“Hit to see next plot:”*, thus requiring user’s interaction to show the complete set of plots. The information about the model (batches and omics included) is shown in the R prompt too. The “inner correlation plot” is a pivotal result, since it represents the validation of the PLS model. The correlation between x score (t) and y score (u) (in every component) is suppossed to be linear, as shown in Figure 8. If a non-linear correlation is observed, a transformation of data would be desirable.

4.2 Batch effect estimation plot

This plot illustrates the magnitude of the estimated batch effects. Tipically, this plot is used before *MultiBaC* or *ARSyNbac* performance since it just requires a basic *mbac* returned by *createMbac* function.

```
plot (my_final_mbac, typeP = "batch")
```

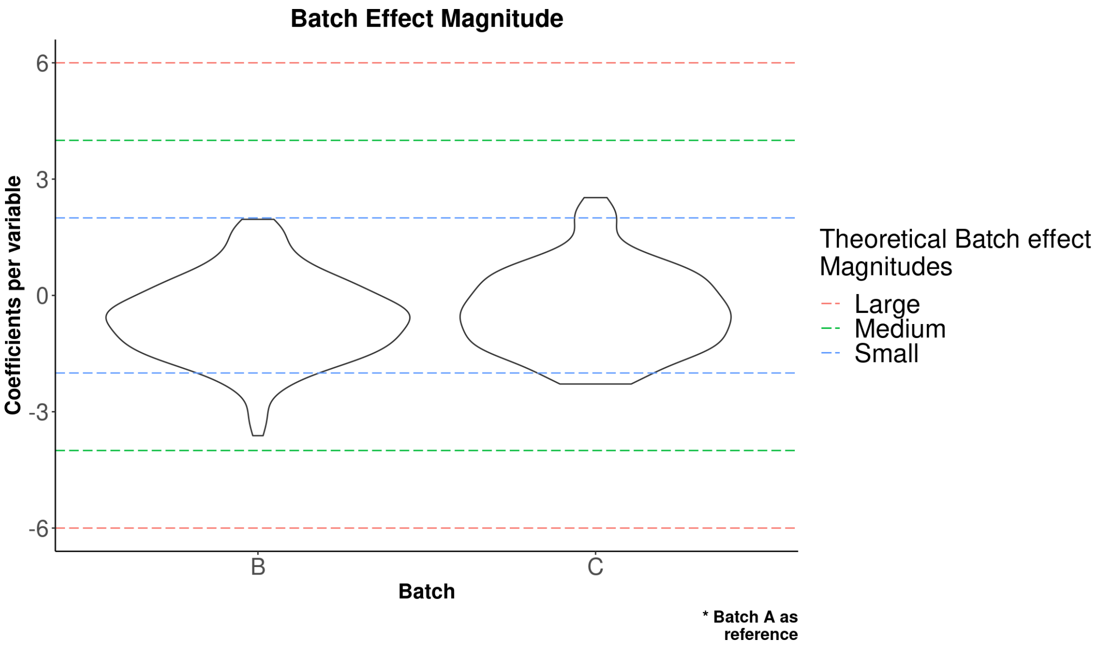


Figure 9: **Batch effect estimation plot**
Dashed lines represent theoretical batch magnitudes. Violin plots represent the distribution of batch effect coefficents observed in data.

Theoretical batch effect magnitudes for the yeast example are displayed in Figure 9. The violin plot shows the distribution of batch effect coefficients along the variable space (genes in case of RNA-seq data). Coefficients with higher values are the one that contribute the most to the existence of a batch effect, thus when the distribution is closer to zero, the batch effect is lower. MultiBaC correction performance has been tested and validated with small and medium batch effect magnitudes while it decreases at high magnitudes. ARSyN is not so much affected by the batch effect magnitude.

4.3 PCA plots

The goodness of *ARSyNbac* or *MultiBaC* correction can be assessed with the PCA plots before and after the correction. In the case of *MultiBaC*, this plot can only be generated when all the omic data matrices share the same variable space. In our yeast example, every omic data type was obtained as gene-related information, thus matrices can be merged by variables (genes) and the PCA is feasible.

An example of the usage of these PCA plots for the *ARSyNbac* output can be found in the ARSyN section. Here we illustrate how to generate and interpret them for MultiBaC correction. The PCA on the original data (Figure 10) and on the corrected data (Figure 11) were obtained with the *plot* function by using either *“typeP = pca.org”* or *“typeP = pca.cor”*, respectively.

```
plot (my_final_mbac, typeP = "pca.org",
      cex.axis = 1, cex.lab = 1, cex = 3, bty = "L",
      cex.main = 1.2, pch = 19)
```

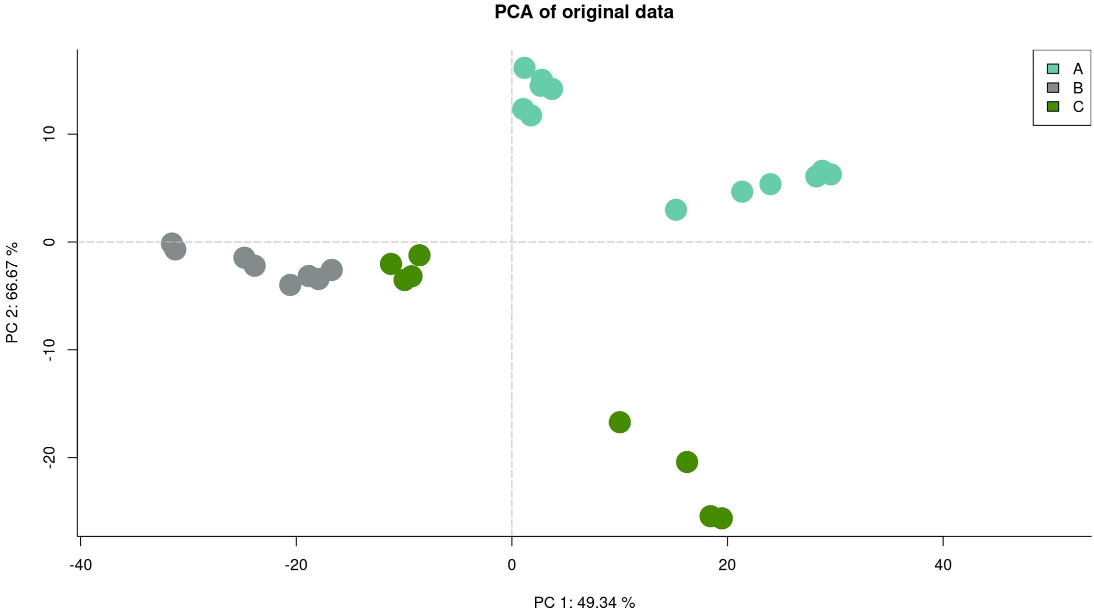


Figure 10: **Default PCA plot on the original data**

```
plot (my_final_mbac, typeP = "pca.cor",
      cex.axis = 1, cex.lab = 1, cex = 3, bty = "L",
      cex.main = 1.2, pch = 19)
```

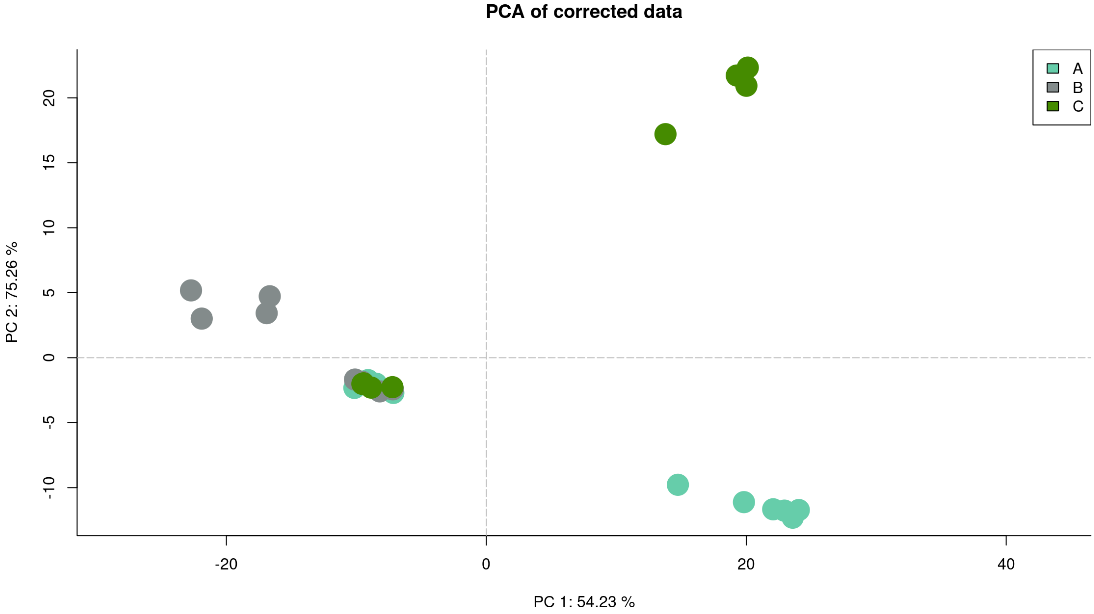


Figure 11: **Default PCA plot on the corrected data**

By default, this function takes random colors to represent each group (batches by default). However, it would be useful to display the experimental factors information too. For that, we recommend the use of custom *col.per.group* and *pch* arguments. An example is shown in Figure 12, using typeP = pca.both to show both PCA plots together. We could also plot more than two components indicating the desired number with *comp2plot* argument. The user can also include a custom legend by using two arguments: *legend.text* and *args.legend*. With *legend.text* we indicate the text labels of the legend and the rest of the legend arguments are collected in *args.legend* (x, y, pch, fill, col, bty, etc). If *legend.text* is not provided to the function, *args.legend* is not considered.

```
custom_col <- c("brown2", "dodgerblue", "forestgreen")
custom_pch <- c(19,19,19,1,1,1,15,15,15,0,0,0, # batch A
                19,19,1,1,17,17,2,2,  # batch B
                19,19,1,1,18,18,5,5)  # batch C

plot(my_final_mbac, typeP = "pca.both", col.by.batch = TRUE,
     col.per.group = custom_col, comp2plot = 1:3,
     cex.axis = 1.3, cex.lab = 1.2, cex = 3, bty = "L",
     cex.main = 1.7, pch = custom_pch,
     legend.text = c("Color", names(my_final_mbac$ListOfBatches),
                     "Shape", c("RNA", "GRO", "RIBO", "PAR"),
                     "Fill", levels(colData(my_final_mbac$ListOfBatches$A)$tfactor)),
     args.legend = list("x" = "topright",
                        "pch" = c(NA, 15, 15, 15,
                                  NA, 19, 15, 17, 18,
                                  NA, 19, 1),
                        "col" = c(NA, "brown2", "dodgerblue", "forestgreen",
                                  NA, rep(1, 4),
                                  NA, 1, 1),
                        "bty" = "n",
                        "cex" = 2))
```

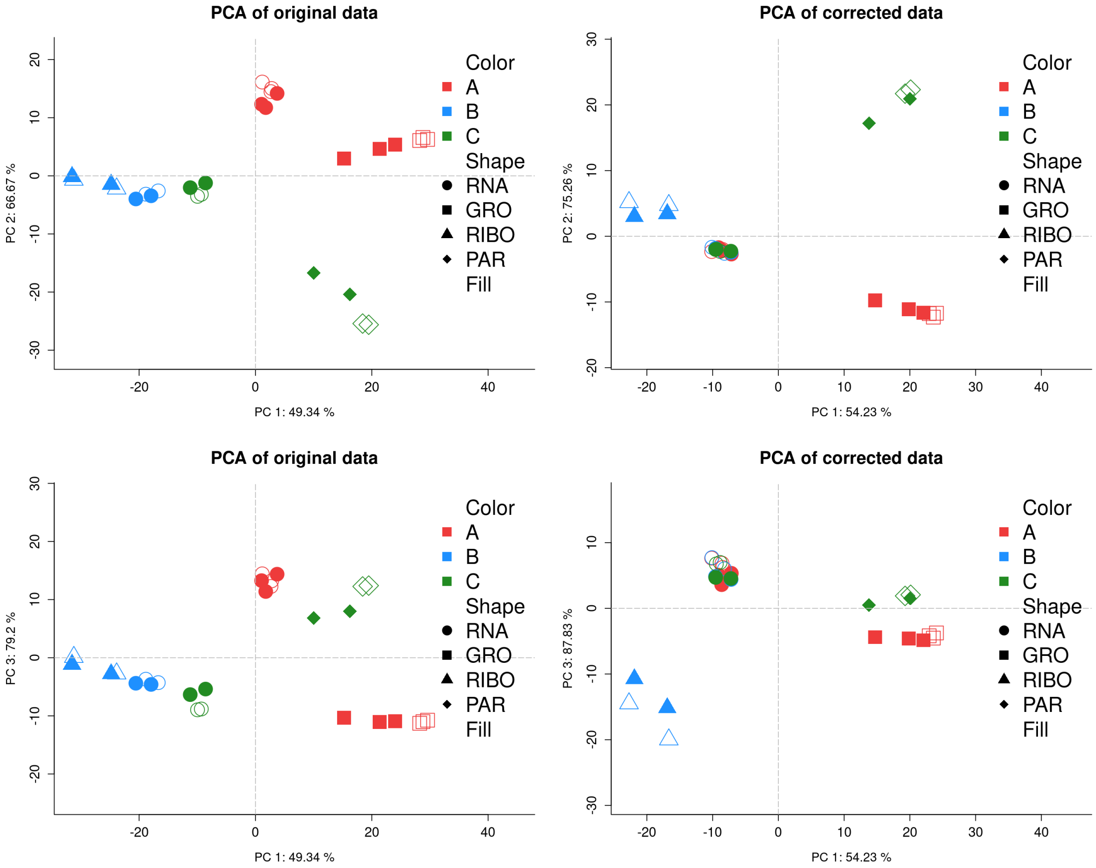


Figure 12: **Customized PCA plots**
Original (left panels) and Corrected (right panels) data. Upper panels show the second principal component (PC) against the first one while panels at the bottom show the third PC against the first one.

In this case, batch effect correction is observable in common data (RNA-seq, dots). Batch effect has been removed as common data is placed all together and after the correction, the components (especially the second and the third component) separate the common data based on the experimental condition instead of separating batches. As shown in the legend, point shape indicates the omic data type, however batch effect correction is only visible in common data.

5 Session info

Here is the output of *sessionInfo()* on the system on which this document was compiled:

```
sessionInfo()
```

```
## R version 4.2.0 RC (2022-04-19 r82224)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /home/biocbuild/bbs-3.15-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.15-bioc/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB              LC_COLLATE=C
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ropls_1.28.0                ggplot2_3.3.5
##  [3] MultiAssayExperiment_1.22.0 SummarizedExperiment_1.26.0
##  [5] Biobase_2.56.0              GenomicRanges_1.48.0
##  [7] GenomeInfoDb_1.32.0         IRanges_2.30.0
##  [9] S4Vectors_0.34.0            BiocGenerics_0.42.0
## [11] MatrixGenerics_1.8.0        matrixStats_0.62.0
## [13] MultiBaC_1.6.0              BiocStyle_2.24.0
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-7           fs_1.5.2               usethis_2.1.5
##  [4] devtools_2.4.3         rprojroot_2.0.3        tools_4.2.0
##  [7] bslib_0.3.1            utf8_1.2.2             R6_2.5.1
## [10] DBI_1.1.2              colorspace_2.0-3       withr_2.5.0
## [13] tidyselect_1.1.2       prettyunits_1.1.1      processx_3.5.3
## [16] compiler_4.2.0         cli_3.3.0              desc_1.4.1
## [19] DelayedArray_0.22.0    labeling_0.4.2         bookdown_0.26
## [22] sass_0.4.1             scales_1.2.0           callr_3.7.0
## [25] stringr_1.4.0          digest_0.6.29          rmarkdown_2.14
## [28] XVector_0.36.0         pkgconfig_2.0.3        htmltools_0.5.2
## [31] sessioninfo_1.2.2      plotrix_3.8-2          fastmap_1.1.0
## [34] highr_0.9              rlang_1.0.2            rstudioapi_0.13
## [37] jquerylib_0.1.4        generics_0.1.2         farver_2.1.0
## [40] jsonlite_1.8.0         dplyr_1.0.8            RCurl_1.98-1.6
## [43] magrittr_2.0.3         GenomeInfoDbData_1.2.8 Matrix_1.4-1
## [46] Rcpp_1.0.8.3           munsell_0.5.0          fansi_1.0.3
## [49] lifecycle_1.0.1        stringi_1.7.6          yaml_2.3.5
## [52] zlibbioc_1.42.0        brio_1.1.3             pkgbuild_1.3.1
## [55] grid_4.2.0             crayon_1.5.1           lattice_0.20-45
## [58] magick_2.7.3           knitr_1.38             ps_1.7.0
## [61] pillar_1.7.0           pkgload_1.2.4          glue_1.6.2
## [64] evaluate_0.15          pcaMethods_1.88.0      remotes_2.4.2
## [67] BiocManager_1.30.17    vctrs_0.4.1            testthat_3.1.4
## [70] gtable_0.3.0           purrr_0.3.4            assertthat_0.2.1
## [73] cachem_1.0.6           xfun_0.30              tibble_3.1.6
## [76] memoise_2.0.1          ellipsis_0.3.2
```

References

1. Nueda MJ, Ferrer A, Conesa A. ARSyN: A method for the identification and removal of systematic noise in multifactorial time course microarray experiments. Biostatistics. 2012;13:553–66.

2. Pelechano V, Pérez-Ortín JE. There is a steady-state transcriptome in exponentially growing yeast cells. Yeast. 2010;27:413–22. doi:10.1002/yea.1768.

3. García-Martínez J, Aranda A, Pérez-Ortín J. Genomic run-on evaluates transcription rates for all yeast genes and identifies gene regulatory mechanisms. Molecular Cell. 15:303–13. doi:10.1016/j.molcel.2004.06.004.

4. Pelechano V, Chávez S, Pérez-Ortín JE. A complete set of nascent transcription rates for yeast genes. PloS one. 5:e15442; e15442–2. doi:10.1371/journal.pone.0015442.

5. Zid BM, O'Shea EK. Promoter sequences direct cytoplasmic localization and translation of mRNAs during starvation in yeast. Nature. 514:117–21. doi:10.1038/nature13578.

6. Freeberg MA, Han T, Moresco JJ, Kong A, Yang Y-C, Lu ZJ, et al. Pervasive and dynamic protein binding sites of the mRNA transcriptome in saccharomyces cerevisiae. Genome biology. 14:R13–3. doi:10.1186/gb-2013-14-2-r13.

Appendix 4: List of differential reactions and their activation state (Chapter 5)

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
2MBALDt	M-2mbald-c	M-2mbald-e	NA	1	-1	-1	-1	1	1
2MBTOHtm	M-2mbtoh-c	M-2mbtoh-m	NA	-1	1	-1	1	1	1
AASAD1	M-L2aadp-c M-atp-c M-h-c M-nadph-c	M-L2aadp6sa-c M-TM-atp-c M-amp-c M-nadp-c M-ppi-c	sce00300-Lysine-biosynthesis sce00770-Pantothenate-and-CoA-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01230-Biosynthesis-of-amino-acids	-1	1	-1	1	0	1
AASAD2	M-L2aadp-c M-atp-c M-h-c M-nadh-c	M-L2aadp6sa-c M-TM-atp-c M-TM-nad-c M-amp-c M-nad-c M-ppi-c	sce00300-Lysine-biosynthesis sce00770-Pantothenate-and-CoA-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01230-Biosynthesis-of-amino-acids	-1	-1	1	-1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ACACT1m	M-accoa-m	M-aacoa-m M-coa-m	sce00071-Fatty-acid-degradation sce00072-Synthesis-and-degradation-of-ketone-bodies sce00280-Valine,-leucine-and-isoleucine-degradation sce00310-Lysine-degradation sce00380-Tryptophan-metabolism sce00620-Pyruvate-metabolism sce00630-Glyoxylate-and-dicarboxylate-metabolism sce00640-Propanoate-metabolism sce00650-Butanoate-metabolism sce00900-Terpenoid-backbone-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce01212-Fatty-acid-metabolism	1	1	-1	1	0	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ACHLE1	M-h2o-c M-iamac-c	M-TM-ac-c M-ac-c M-h-c M-iamoh-c	NA	1	1	-1	1	0	1
ACHLE2	M-h2o-c M-ibutac-c	M-TM-ac-c M-ac-c M-h-c M-ibutoh-c	NA	-1	1	-1	1	1	1
ACHLE3	M-aces-c M-h2o-c	M-TM-ac-c M-TM-etoh-c M-ac-c M-etoh-c M-h-c	NA	-1	1	-1	1	1	1
ADA	M-adn-c M-h2o-c M-h-c	M-ins-c M-nh4-c	sce00230-Purine-metabolism sce01100-Metabolic-pathways	-1	1	-1	-1	1	1
ADK1m	M-amp-m M-atp-m	M-adp-m	sce00230-Purine-metabolism sce00730-Thiamine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	1	-1	1	1	1
ADK4m	M-amp-m M-itp-m	M-adp-m M-idp-m	sce00230-Purine-metabolism sce00730-Thiamine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ADNCYC	M-atp-c	M-TM-atp-c M-camp-c M-ppi-c	sce00230-Purine-metabolism sce01100-Metabolic-pathways sce04113-Meiosis sce04213-Longevity-regulating-pathway	1	-1	-1	1	1	1
ADNK1	M-adn-c M-atp-c	M-TM-atp-c M-adp-c M-amp-c M-h-c	sce00230-Purine-metabolism sce01100-Metabolic-pathways	-1	1	1	1	1	1
ADNUC	M-adn-c M-h2o-c	M-ade-c M-rib–D-c	sce00240-Pyrimidine-metabolism sce00760-Nicotinate-and-nicotinamide-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
AGTi	M-ala–L-c M-glx-c	M-TM-ala–L-c M-TM-gly-c M-gly-c M-pyr-c	sce00250-Alanine,-aspartate-and-glutamate-metabolism sce00260-Glycine,-serine-and-threonine-metabolism sce00630-Glyoxylate-and-dicarboxylate-metabolism sce00680-Methane-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce04146-Peroxisome	-1	1	1	1	1	1
ALCD2x-copy1	M-etoh-c M-nad-c	M-TM-etoh-c M-TM-nad-c M-acald-c M-h-c M-nadh-c	sce00010-Glycolysis-/-Gluconeogenesis sce00071-Fatty-acid-degradation sce00350-Tyrosine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ALDD2ym	M-acald-m M-h2o-m M-nadp-m	M-ac-m M-h-m M-nadph-m	sce00010-Glycolysis-/-Gluconeogenesis sce00071-Fatty-acid-degradation sce00280-Valine,-leucine-and-isoleucine-degradation sce00310-Lysine-degradation sce00330-Arginine-and-proline-metabolism sce00340-Histidine-metabolism sce00380-Tryptophan-metabolism sce00410-beta-Alanine-metabolism sce00561-Glycerolipid-metabolism sce00620-Pyruvate-metabolism sce00770-Pantothenate-and-CoA-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	1	-1	1	1	1
AMPN	M-amp-c M-h2o-c	M-ade-c M-r5p-c	sce00230-Purine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	-1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ASNN	M-asn–L-c M-h2o-c	M-TM-asn–L-c M-TM-asp–L-c M-asp–L-c M-nh4-c	sce00250-Alanine,-aspartate-and-glutamate-metabolism sce00460-Cyanoamino-acid-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	-1	-1	1	1	1
ASPt2n	M-asp–L-c M-h-c	M-TM-asp–L-c M-asp–L-n M-h-n	NA	-1	1	-1	1	1	1
ASPt2r	M-asp–L-e M-h-e	M-TM-asp–L-c M-asp–L-c M-h-c	NA	1	-1	1	1	1	1
ATPH1	M-atp-c M-h2o-c	M-TM-atp-c M-amp-c M-h-c M-pi-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	1	-1	1	1	1	1
ATPM	M-atp-c M-h2o-c	M-TM-atp-c M-adp-c M-h-c M-pi-c	NA	1	-1	-1	-1	-1	1
ATPS3m	M-adp-m M-h-c M-pi-m	M-atp-m M-h2o-m M-h-m	sce00190-Oxidative-phosphorylation sce01100-Metabolic-pathways	-1	0	1	1	1	1
BTDD-RR	M-btd-RR-c M-nad-c	M-TM-nad-c M-actn–R-c M-h-c M-nadh-c	sce00650-Butanoate-metabolism	-1	-1	-1	-1	0	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
CERASE224er	M-cer2-24-r M-coa-r M-h-r	M-psphings-r M-ttccoa-r	sce00600-Sphingolipid-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1
CERS224er	M-psphings-r M-ttccoa-r	M-cer2-24-r M-coa-r M-h-r	sce00600-Sphingolipid-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1
CHLPCTD	M-cholp-c M-ctp-c M-h-c	M-cdpchol-c M-ppi-c	sce00440-Phosphonate-and-phosphinate-metabolism sce00564-Glycerophospholipid-metabolism sce01100-Metabolic-pathways	-1	-1	1	1	1	1
CO2tm	M-co2-c	M-co2-m	NA	-1	1	1	1	1	1
COAtim	M-coa-c	M-coa-m	NA	1	-1	1	1	1	1
CTPS2	M-atp-c M-gln–L-c M-h2o-c M-utp-c	M-TM-atp-c M-TM-gln–L-c M-TM-glu–L-c M-adp-c M-ctp-c M-glu–L-c M-h-c M-pi-c	sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	-1	1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
CYSS	M-acser-c M-h2s-c	M-TM-ac-c M-ac-c M-cys–L-c M-h-c	sce00270-Cysteine-and-methionine-metabolism sce00920-Sulfur-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce01230-Biosynthesis-of-amino-acids	1	-1	1	1	1	1
CYTD	M-cytd-c M-h2o-c M-h-c	M-nh4-c M-uri-c	sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1
CYTK1	M-atp-c M-cmp-c	M-TM-atp-c M-TM-cmp-c M-adp-c M-cdp-c	NA	-1	-1	-1	1	-1	1
DHAK	M-atp-c M-dha-c	M-TM-atp-c M-adp-c M-dhap-c M-h-c	sce00051-Fructose-and-mannose-metabolism sce00561-Glycerolipid-metabolism sce00680-Methane-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
DURIPP	M-duri-c M-pi-c	M-2dr1p-c M-ura-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce00760-Nicotinate-and-nicotinamide-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	1	-1	1	1	1
DUTPDP	M-dutp-c M-h2o-c	M-dump-c M-h-c M-ppi-c	sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	-1	1	-1	1	0	1
D-LACDm	M-ficytc-m M-lac–D-m	M-focytc-m M-pyr-m	sce00620-Pyruvate-metabolism sce01100-Metabolic-pathways	-1	-1	1	1	1	1
ECOAH11p	M-h2o-x M-hxc2coa-x	M-3hxccoa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ECOAH4p	M-3hdcoa-x	M-dc2coa-x M-h2o-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1
ECOAH6p	M-3htdcoa-x	M-h2o-x M-td2coa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1
ECOAH7p	M-3hhdcoa-x	M-h2o-x M-hdd2coa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1
ECOAH8p	M-3hodcoa-x	M-h2o-x M-od2coa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1
EX-2mbald-e	M-2mbald-e	NA	NA	1	1	-1	1	1	1
EX-acald-e	M-acald-e	NA	NA	-1	1	1	1	1	1
EX-asp–L-e	M-asp–L-e	NA	NA	1	1	-1	-1	-1	1
EX-for-e	M-for-e	NA	NA	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
EX-fum-e	M-fum-e	NA	NA	-1	1	1	1	1	1
EX-glu–L-e	M-glu–L-e	NA	NA	1	-1	1	1	1	1
EX-glyc-e	M-glyc-e	NA	NA	1	-1	1	-1	1	-1
EX-lac–D-e	M-lac–D-e	NA	NA	-1	1	1	1	0	1
EX-nh4-e	M-nh4-e	NA	NA	-1	1	-1	1	1	1
EX-oaa-e	M-oaa-e	NA	NA	-1	1	-1	1	1	1
EX-orn-e	M-orn-e	NA	NA	-1	-1	-1	1	1	1
EX-pyr-e	M-pyr-e	NA	NA	1	-1	-1	1	1	1
EX-succ-e	M-succ-e	NA	NA	-1	0	1	1	0	-1
FACOAL161	M-atp-c M-coa-c M-hdcea-c	M-TM-atp-c M-amp-c M-hdcoa-c M-ppi-c	sce00061-Fatty-acid-biosynthesis sce00071-Fatty-acid-degradation sce01100-Metabolic-pathways sce01212-Fatty-acid-metabolism sce04146-Peroxisome	1	1	-1	1	1	1
FACOAL80p	M-atp-x M-coa-x M-octa-x	M-amp-x M-occoa-x M-ppi-x	sce00061-Fatty-acid-biosynthesis sce00071-Fatty-acid-degradation sce01100-Metabolic-pathways sce01212-Fatty-acid-metabolism sce04146-Peroxisome	1	1	-1	1	0	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
FALDH	M-fald-c M-gthrd-c M-nad-c	M-Sfglutth-c M-TM-gthrd-c M-TM-nad-c M-h-c M-nadh-c	sce00010-Glycolysis-/-Gluconeogenesis sce00071-Fatty-acid-degradation sce00350-Tyrosine-metabolism sce00680-Methane-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	1	1	1	1	1
FBP	M-fdp-c M-h2o-c	M-TM-fdp-c M-f6p-c M-pi-c	sce00010-Glycolysis-/-Gluconeogenesis sce00030-Pentose-phosphate-pathway sce00051-Fructose-and-mannose-metabolism sce00680-Methane-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	-1	-1	1	1	1
FECOSTt	M-fecost-e	M-fecost-c	sce02010-ABC-transporters	-1	1	1	1	0	1
FRDm	M-fadh2-m M-fum-m	M-fad-m M-succ-m	NA	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
FTHFLi	M-atp-c M-for-c M-thf-c	M-10fthf-c M-TM-atp-c M-adp-c M-pi-c	sce00670-One-carbon-pool-by-folate sce01100-Metabolic-pathways	-1	-1	1	1	1	1
G3PCt	M-g3pc-e	M-TM-g3pc-c M-g3pc-c	NA	-1	1	1	1	1	1
G3PD1ir	M-dhap-c M-h-c M-nadh-c	M-TM-nad-c M-glyc3p-c M-nad-c	sce00564-Glycerophospholipid-metabolism sce01110-Biosynthesis-of-secondary-metabolites sce04011-MAPK-signaling-pathway	1	-1	1	1	1	1
G3PIt	M-g3pi-e	M-g3pi-c	NA	1	1	-1	-1	1	-1
G3PT	M-glyc3p-c M-h2o-c	M-TM-glyc-c M-glyc-c M-pi-c	sce00561-Glycerolipid-metabolism sce01100-Metabolic-pathways	-1	1	1	1	1	1
G5SADrm	M-glu5sa-m	M-1pyr5c-m M-h2o-m M-h-m	NA	1	-1	1	1	1	1
G5SD2	M-glu5p-c M-h-c M-nadh-c	M-TM-nad-c M-glu5sa-c M-nad-c M-pi-c	sce00330-Arginine-and-proline-metabolism sce00332-Carbapenem-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01230-Biosynthesis-of-amino-acids	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
G6PDH2r	M-g6p-c M-nadp-c	M-6pgl-c M-h-c M-nadph-c	sce00030-Pentose-phosphate-pathway sce00480-Glutathione-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	0	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
GCC2bim	M-alpam-m M-thf-m	M-dhlam-m M-mlthf-m M-nh4-m	sce00010-Glycolysis-/-Gluconeogenesis sce00020-Citrate-cycle-(TCA-cycle) sce00260-Glycine,-serine-and-threonine-metabolism sce00280-Valine,-leucine-and-isoleucine-degradation sce00310-Lysine-degradation sce00380-Tryptophan-metabolism sce00620-Pyruvate-metabolism sce00630-Glyoxylate-and-dicarboxylate-metabolism sce00640-Propanoate-metabolism sce00670-One-carbon-pool-by-folate sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
GLCS2	M-udpg-c	M-TM-udpg-c M-glycogen-c M-h-c M-udp-c	sce00500-Starch-and-sucrose-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	-1	1	1	1	1
GLNt2r	M-gln–L-e M-h-e	M-TM-gln–L-c M-gln–L-c M-h-c	NA	-1	-1	1	1	1	1
GLUSx	M-akg-c M-gln–L-c M-h-c M-nadh-c	M-TM-gln–L-c M-TM-glu–L-c M-TM-nad-c M-glu–L-c M-nad-c	sce00250-Alanine,-aspartate-and-glutamate-metabolism sce00910-Nitrogen-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01230-Biosynthesis-of-amino-acids	-1	-1	1	1	1	1
GLUt7m	M-glu–L-c	M-TM-glu–L-c M-glu–L-m	NA	-1	-1	-1	1	1	1
GLYCDy	M-glyc-c M-nadp-c	M-TM-glyc-c M-dha-c M-h-c M-nadph-c	sce00561-Glycerolipid-metabolism sce01100-Metabolic-pathways	1	-1	1	1	1	1
GLYK	M-atp-c M-glyc-c	M-TM-atp-c M-TM-glyc-c M-adp-c M-glyc3p-c M-h-c	sce00561-Glycerolipid-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
GLYOX	M-h2o-c M-lgt–S-c	M-TM-gthrd-c M-gthrd-c M-h-c M-lac–D-c	sce00620-Pyruvate-metabolism sce01100-Metabolic-pathways	1	-1	-1	1	0	1
GNNUC	M-gsn-c M-h2o-c	M-gua-c M-rib–D-c	sce00240-Pyrimidine-metabolism sce00760-Nicotinate-and-nicotinamide-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1
H2Otp	M-h2o-c	M-h2o-x	NA	-1	0	1	1	1	1
HACD10p	M-3hxccoa-x M-nad-x	M-3ohxccoa-x M-h-x M-nadh-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	-1	-1	-1	1	1	-1
HACD7p	M-3ohdcoa-x M-h-x M-nadh-x	M-3hhdcoa-x M-nad-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	-1	-1	-1	1	1	-1
HCO3E	M-co2-c M-h2o-c	M-h-c M-hco3-c	NA	1	1	-1	-1	0	-1
HCO3tn	M-hco3-c	M-hco3-n	NA	-1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
HMGCOAS	M-coa-c M-h-c M-hmgcoa-c	M-aacoa-c M-accoa-c M-h2o-c	sce00072-Synthesis-and-degradation-of-ketone-bodies sce00280-Valine,-leucine-and-isoleucine-degradation sce00650-Butanoate-metabolism sce00900-Terpenoid-backbone-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	-1	1	1	1	1
HSDxi	M-aspsa-c M-h-c M-nadh-c	M-TM-nad-c M-hom–L-c M-nad-c	sce00260-Glycine,-serine-and-threonine-metabolism sce00270-Cysteine-and-methionine-metabolism sce00300-Lysine-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01230-Biosynthesis-of-amino-acids	1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ICL	M-icit-c	M-TM-succ-c M-glx-c M-succ-c	sce00630-Glyoxylate-and-dicarboxylate-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	1	1	1	1	1
ILEt2r	M-h-e M-ile–L-e	M-TM-ile–L-c M-h-c M-ile–L-c	NA	-1	1	-1	1	1	1
ILEtmi	M-ile–L-m	M-TM-ile–L-c M-ile–L-c	NA	-1	1	-1	1	1	1
IPPMIb	M-2ippm-c M-h2o-c	M-3c3hmp-c	sce00290-Valine,-leucine-and-isoleucine-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01210-2-Oxocarboxylic-acid-metabolism sce01230-Biosynthesis-of-amino-acids	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
IPPSm	M-3mob-m M-accoa-m M-h2o-m	M-3c3hmp-m M-coa-m M-h-m	sce00290-Valine,-leucine-and-isoleucine-biosynthesis sce00620-Pyruvate-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01210-2-Oxocarboxylic-acid-metabolism sce01230-Biosynthesis-of-amino-acids	-1	-1	1	1	1	1
ITCOALm	M-atp-m M-coa-m M-itacon-m	M-adp-m M-itaccoa-m M-pi-m	sce00020-Citrate-cycle-(TCA-cycle) sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
LALDO3	M-h-c M-mthgxl-c M-nadph-c	M-lald–L-c M-nadp-c	sce00040-Pentose-and-glucuronate-interconversions sce00620-Pyruvate-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce04011-MAPK-signaling-pathway	-1	1	1	1	1	1
LEUt2r	M-h-e M-leu–L-e	M-TM-leu–L-c M-h-c M-leu–L-c	NA	1	0	-1	1	1	1
LGTHL	M-gthrd-c M-mthgxl-c	M-TM-gthrd-c M-lgt–S-c	sce00620-Pyruvate-metabolism sce01100-Metabolic-pathways	1	-1	-1	1	0	1
LNS14DM	M-h-c M-lanost-c M-nadph-c M-o2-c	M-44mctr-c M-for-c M-h2o-c M-nadp-c	sce00100-Steroid-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	-1	1	1	1	1
LPP-SC	M-dagpy-SC-c M-h2o-c	M-h-c M-pa-SC-c M-pi-c	sce00561-Glycerolipid-metabolism sce00564-Glycerophospholipid-metabolism sce01110-Biosynthesis-of-secondary-metabolites	1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
LSERDHr	M-nadp-c M-ser–L-c	M-2amsa-c M-TM-ser–L-c M-h-c M-nadph-c	sce00240-Pyrimidine-metabolism sce00260-Glycine,-serine-and-threonine-metabolism sce01100-Metabolic-pathways	1	-1	-1	1	1	-1
MALt2r	M-h-e M-mal–L-e	M-h-c M-mal–L-c	NA	-1	1	-1	-1	-1	-1
MDHm	M-mal–L-m M-nad-m	M-h-m M-nadh-m M-oaa-m	sce00020-Citrate-cycle-(TCA-cycle) sce00270-Cysteine-and-methionine-metabolism sce00620-Pyruvate-metabolism sce00630-Glyoxylate-and-dicarboxylate-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	-1	1	1	1	1
MEVK1	M-atp-c M-mev–R-c	M-5pmev-c M-TM-atp-c M-adp-c M-h-c	sce00900-Terpenoid-backbone-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce04146-Peroxisome	-1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
MEVK3	M-gtp-c M-mev–R-c	M-5pmev-c M-gdp-c M-h-c	sce00900-Terpenoid-backbone-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce04146-Peroxisome	1	-1	-1	1	1	1
MEVK4	M-mev–R-c M-utp-c	M-5pmev-c M-h-c M-udp-c	sce00900-Terpenoid-backbone-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce04146-Peroxisome	1	-1	1	1	1	1
NADPPPS	M-h2o-c M-nadp-c	M-TM-nad-c M-nad-c M-pi-c	NA	1	-1	-1	-1	1	-1
NDP3	M-gdp-c M-h2o-c	M-TM-gmp-c M-gmp-c M-h-c M-pi-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1
NDP7	M-h2o-c M-udp-c	M-h-c M-pi-c M-ump-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	-1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
NDPK3	M-atp-c M-cdp-c	M-TM-atp-c M-adp-c M-ctp-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	-1	1	1	1	1
NDPK8	M-atp-c M-dadp-c	M-TM-atp-c M-adp-c M-datp-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	-1	1	1	1	1
NH4tm	M-nh4-c	M-nh4-m	NA	-1	1	1	1	1	1
NTD2	M-h2o-c M-ump-c	M-pi-c M-uri-c	sce00760-Nicotinate-and-nicotinamide-metabolism	-1	-1	-1	1	1	1
NTP3	M-gtp-c M-h2o-c	M-gdp-c M-h-c M-pi-c	NA	-1	-1	1	-1	-1	-1
OAAt	M-oaa-c	M-oaa-e	NA	-1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
OHPBAT	M-glu–L-c M-ohpb-c	M-TM-glu–L-c M-akg-c M-phthr-c	sce00260-Glycine,-serine-and-threonine-metabolism sce00270-Cysteine-and-methionine-metabolism sce00680-Methane-metabolism sce00750-Vitamin-B6-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce01230-Biosynthesis-of-amino-acids	-1	0	-1	1	1	1
ORNTA	M-akg-c M-orn-c	M-TM-glu–L-c M-glu5sa-c M-glu–L-c	sce00330-Arginine-and-proline-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	0	-1	1	1	1
P5CDm	M-1pyr5c-m M-h2o-m M-nad-m	M-glu–L-m M-h-m M-nadh-m	NA	1	-1	1	-1	1	1
PAK-SC	M-atp-c M-pa-SC-c	M-TM-atp-c M-adp-c M-dagpy-SC-c	NA	1	1	-1	1	1	1
PAPtm	M-pap-c	M-pap-m	NA	1	1	-1	-1	-1	-1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
PDE1	M-camp-c M-h2o-c	M-amp-c M-h-c	sce00230-Purine-metabolism sce01100-Metabolic-pathways	1	-1	-1	1	1	1
GCC2cm-copy2	M-dhlam-m M-nad-m	M-h-m M-lpam-m M-nadh-m	sce00010-Glycolysis-/-Gluconeogenesis sce00020-Citrate-cycle-(TCA-cycle) sce00260-Glycine,-serine-and-threonine-metabolism sce00280-Valine,-leucine-and-isoleucine-degradation sce00310-Lysine-degradation sce00380-Tryptophan-metabolism sce00620-Pyruvate-metabolism sce00630-Glyoxylate-and-dicarboxylate-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	-1	1	1	1	1
PEtm-SC	M-pe-SC-c	M-pe-SC-m	NA	-1	1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
PGI	M-g6p-c	M-f6p-c	sce00010-Glycolysis-/-Gluconeogenesis sce00030-Pentose-phosphate-pathway sce00500-Starch-and-sucrose-metabolism sce00520-Amino-sugar-and-nucleotide-sugar-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	-1	1	1	1	1
PGM	M-2pg-c	M-3pg-c	sce00010-Glycolysis-/-Gluconeogenesis sce00260-Glycine,-serine-and-threonine-metabolism sce00680-Methane-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce01230-Biosynthesis-of-amino-acids	1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
PI35BP5P-SC	M-h2o-c M-ptd135bp-SC-c	M-pi-c M-ptd3ino-SC-c	sce00562-Inositol-phosphate-metabolism sce01100-Metabolic-pathways sce04070-Phosphatidylinositol-signaling-system	-1	1	1	1	1	1
PI3P5K-SC	M-atp-c M-ptd3ino-SC-c	M-TM-atp-c M-adp-c M-h-c M-ptd135bp-SC-c	sce00562-Inositol-phosphate-metabolism sce01100-Metabolic-pathways sce04070-Phosphatidylinositol-signaling-system sce04145-Phagosome	-1	1	1	1	1	1
PLD-SC	M-h2o-c M-pc-SC-c	M-chol-c M-h-c M-pa-SC-c	sce00564-Glycerophospholipid-metabolism sce00565-Ether-lipid-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce04144-Endocytosis	-1	-1	1	1	1	1
PPND	M-nad-c M-pphn-c	M-34hpp-c M-TM-nad-c M-co2-c M-nadh-c	NA	1	1	1	1	-1	1
PROt2r	M-h-e M-pro–L-e	M-TM-pro–L-c M-h-c M-pro–L-c	NA	-1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
PSERSm-SC	M-cdpdag-SC-m M-ser–L-m	M-cmp-m M-h-m M-ps-SC-m	sce00260-Glycine,-serine-and-threonine-metabolism sce00564-Glycerophospholipid-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	1	-1	1	0	1
PTRCt3i	M-h-c M-ptrc-e	M-h-e M-ptrc-c	NA	-1	-1	1	1	1	1
PTRCtex2	M-ptrc-c	M-ptrc-e	NA	-1	1	1	1	1	1
PUNP1	M-adn-c M-pi-c	M-ade-c M-r1p-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce00760-Nicotinate-and-nicotinamide-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	-1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
PUNP6	M-din-c M-pi-c	M-2dr1p-c M-hxan-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce00760-Nicotinate-and-nicotinamide-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	-1	1	1	1	1
PYRt2	M-h-e M-pyr-e	M-h-c M-pyr-c	NA	-1	0	1	1	1	1
RNDR3	M-cdp-c M-trdrd-c	M-dcdp-c M-h2o-c M-trdox-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce00480-Glutathione-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1
SERAT	M-accoa-c M-ser–L-c	M-TM-ser–L-c M-acser-c M-coa-c	NA	1	-1	1	1	1	1
SLFAT	M-adp-c M-h-c M-so4-c	M-aps-c M-pi-c	sce00230-Purine-metabolism sce00920-Sulfur-metabolism sce01100-Metabolic-pathways	1	1	-1	1	1	1
THIORDXi	M-h2o2-c M-trdrd-c	M-h2o-c M-trdox-c	NA	-1	-1	1	1	1	1
THIORDXp	M-h2o2-x M-trdrd-x	M-h2o-x M-trdox-x	sce04122-Sulfur-relay-system	-1	-1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
THMP	M-h2o-c M-thmmp-c	M-TM-thm-c M-pi-c M-thm-c	NA	1	-1	-1	1	1	1
THRt2r	M-h-e M-thr–L-e	M-h-c M-thr–L-c	NA	-1	-1	1	1	1	1
TKT1	M-r5p-c M-xu5p–D-c	M-g3p-c M-s7p-c	sce00030-Pentose-phosphate-pathway sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce01230-Biosynthesis-of-amino-acids	1	1	-1	1	1	1
TMDK1	M-atp-c M-thymd-c	M-TM-atp-c M-adp-c M-dtmp-c M-h-c	NA	-1	1	1	1	-1	1
TMN	M-h2o-c M-thm-c	M-4ahmmp-c M-4mhetz-c M-TM-thm-c M-h-c	NA	1	1	-1	1	1	1
TREH	M-h2o-c M-tre-c	M-TM-glc–D-c M-TM-tre-c M-glc–D-c	sce00500-Starch-and-sucrose-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	-1	-1	1	1	1
TREt2v	M-h-c M-tre-c	M-TM-tre-c M-h-v M-tre-v	NA	1	1	-1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
UPPRT	M-prpp-c M-ura-c	M-ppi-c M-ump-c	sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	1	1	-1	1	0	1
UREASE	M-atp-c M-hco3-c M-urea-c	M-TM-atp-c M-adp-c M-allphn-c M-h-c M-pi-c	sce00220-Arginine-biosynthesis sce00791-Atrazine-degradation sce01100-Metabolic-pathways	-1	-1	-1	1	1	1
URIK1	M-atp-c M-uri-c	M-TM-atp-c M-adp-c M-h-c M-ump-c	sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	-1	1	-1	1	1	1
VALt2r	M-h-e M-val–L-e	M-TM-val–L-c M-h-c M-val–L-c	NA	1	-1	-1	1	0	1
2MBALDt-reverse	M-2mbald-e	M-2mbald-c	NA	1	-1	1	1	1	1
3C3HMPtm-reverse	M-3c3hmp-m	M-3c3hmp-c	NA	-1	-1	1	1	1	1
ACALDt-reverse	M-acald-c	M-acald-e	NA	-1	1	1	1	1	1
ACALDtm-reverse	M-acald-c	M-acald-m	NA	-1	1	-1	1	1	1
ACCOAC-reverse	M-adp-c M-h-c M-malcoa-c M-pi-c	M-TM-atp-c M-accoa-c M-atp-c M-hco3-c	sce00061-Fatty-acid-biosynthesis sce00620-Pyruvate-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01212-Fatty-acid-metabolism	-1	1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ADK1-reverse	M-adp-c	M-TM-atp-c M-amp-c M-atp-c	sce00230-Purine-metabolism sce00730-Thiamine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	-1	-1	1	1	1
ADK3m-reverse	M-adp-m M-gdp-m	M-amp-m M-gtp-m	sce00230-Purine-metabolism sce00730-Thiamine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	1	-1	1	1	1
ADK4m-reverse	M-adp-m M-idp-m	M-amp-m M-itp-m	sce00230-Purine-metabolism sce00730-Thiamine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	-1	1	1	1	1
AKGMAL-reverse	M-akg-e M-mal–L-c	M-akg-c M-mal–L-e	NA	-1	-1	1	1	1	1
ASPt2n-reverse	M-asp–L-n M-h-n	M-TM-asp–L-c M-asp–L-c M-h-c	NA	1	-1	1	1	1	1
ASPt2r-reverse	M-asp–L-c M-h-c	M-TM-asp–L-c M-asp–L-e M-h-e	NA	1	1	-1	1	1	1
BTDD-RR-reverse	M-actn–R-c M-h-c M-nadh-c	M-TM-nad-c M-btd-RR-c M-nad-c	sce00650-Butanoate-metabolism	-1	-1	-1	-1	0	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
CYSt2r-reverse	M-cys–L-c M-h-c	M-cys–L-e M-h-e	NA	-1	1	-1	1	1	1
CYTK1-reverse	M-adp-c M-cdp-c	M-TM-atp-c M-TM-cmp-c M-atp-c M-cmp-c	NA	-1	-1	-1	-1	0	1
DASYN-SC-reverse	M-cdpdag-SC-c M-ppi-c	M-ctp-c M-h-c M-pa-SC-c	sce00564-Glycerophospholipid-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce04070-Phosphatidylinositol-signaling-system	1	1	-1	1	1	1
DURIPP-reverse	M-2dr1p-c M-ura-c	M-duri-c M-pi-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce00760-Nicotinate-and-nicotinamide-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	1	1	1	1	1
ECOAH11p-reverse	M-3hxccoa-x	M-h2o-x M-hxc2coa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ECOAH4p-reverse	M-dc2coa-x M-h2o-x	M-3hdcoa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1
ECOAH6p-reverse	M-h2o-x M-td2coa-x	M-3htdcoa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1
ECOAH7p-reverse	M-h2o-x M-hdd2coa-x	M-3hhdcoa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1
ECOAH8p-reverse	M-h2o-x M-od2coa-x	M-3hodcoa-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ENO-reverse	M-h2o-c M-pep-c	M-2pg-c	sce00010-Glycolysis-/-Gluconeogenesis sce00680-Methane-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce01230-Biosynthesis-of-amino-acids sce03018-RNA-degradation	1	1	-1	1	1	1
EX-h-e-reverse	NA	M-h-e	NA	-1	-1	1	1	1	1
FACOAL140-reverse	M-amp-c M-ppi-c M-tdcoa-c	M-TM-atp-c M-atp-c M-coa-c M-ttdca-c	sce00061-Fatty-acid-biosynthesis sce00071-Fatty-acid-degradation sce01100-Metabolic-pathways sce01212-Fatty-acid-metabolism sce04146-Peroxisome	-1	1	1	1	1	1
FACOAL181-reverse	M-amp-c M-odecoa-c M-ppi-c	M-TM-atp-c M-atp-c M-coa-c M-ocdcea-c	sce00061-Fatty-acid-biosynthesis sce00071-Fatty-acid-degradation sce01100-Metabolic-pathways sce01212-Fatty-acid-metabolism sce04146-Peroxisome	1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
FACOAL80p-reverse	M-amp-x M-occoa-x M-ppi-x	M-atp-x M-coa-x M-octa-x	sce00061-Fatty-acid-biosynthesis sce00071-Fatty-acid-degradation sce01100-Metabolic-pathways sce01212-Fatty-acid-metabolism sce04146-Peroxisome	1	1	-1	1	0	1
FALDH-reverse	M-Sfglutth-c M-h-c M-nadh-c	M-TM-gthrd-c M-TM-nad-c M-fald-c M-gthrd-c M-nad-c	sce00010-Glycolysis-/-Gluconeogenesis sce00071-Fatty-acid-degradation sce00350-Tyrosine-metabolism sce00680-Methane-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
FBA-reverse	M-dhap-c M-g3p-c	M-TM-fdp-c M-fdp-c	sce00010-Glycolysis-/-Gluconeogenesis sce00030-Pentose-phosphate-pathway sce00051-Fructose-and-mannose-metabolism sce00680-Methane-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce01230-Biosynthesis-of-amino-acids	1	-1	1	1	1	1
FECOSTt-reverse	M-fecost-c	M-fecost-e	sce02010-ABC-transporters	-1	1	1	1	0	1
FUMm-reverse	M-mal–L-m	M-fum-m M-h2o-m	sce00020-Citrate-cycle-(TCA-cycle) sce00620-Pyruvate-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	-1	-1	1	1	1
FUMt2r-reverse	M-fum-c M-h-c	M-fum-e M-h-e	NA	-1	1	1	1	1	1
G3PCt-reverse	M-g3pc-c	M-TM-g3pc-c M-g3pc-e	NA	-1	1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
G3PIt-reverse	M-g3pi-c	M-g3pi-e	NA	1	1	-1	-1	1	-1
GLUt2r-reverse	M-glu–L-c M-h-c	M-TM-glu–L-c M-glu–L-e M-h-e	NA	1	1	-1	1	1	1
GLYCt-reverse	M-glyc-e	M-TM-glyc-c M-glyc-c	NA	-1	1	-1	1	1	1
H2Otp-reverse	M-h2o-x	M-h2o-c	NA	-1	-1	1	1	1	1
HACD10p-reverse	M-3ohxccoa-x M-h-x M-nadh-x	M-3hxccoa-x M-nad-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	-1	-1	-1	1	1	-1
HACD7p-reverse	M-3hhdcoa-x M-nad-x	M-3ohdcoa-x M-h-x M-nadh-x	sce00410-beta-Alanine-metabolism sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01200-Carbon-metabolism	-1	-1	-1	1	1	-1
HMGCOAtm-reverse	M-hmgcoa-m	M-hmgcoa-c	NA	1	1	-1	1	0	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
ILETAm-reverse	M-3mop-m M-glu–L-m	M-akg-m M-ile–L-m	sce00270-Cysteine-and-methionine-metabolism sce00280-Valine,-leucine-and-isoleucine-degradation sce00290-Valine,-leucine-and-isoleucine-biosynthesis sce00770-Pantothenate-and-CoA-biosynthesis sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01210-2-Oxocarboxylic-acid-metabolism sce01230-Biosynthesis-of-amino-acids	-1	1	-1	1	1	1
ITCOALm-reverse	M-adp-m M-itaccoa-m M-pi-m	M-atp-m M-coa-m M-itacon-m	sce00020-Citrate-cycle-(TCA-cycle) sce00640-Propanoate-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	-1	-1	1	1	1	1

Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
LSERDHr-reverse	M-2amsa-c M-h-c M-nadph-c	M-TM-ser–L-c M-nadp-c M-ser–L-c	sce00240-Pyrimidine-metabolism sce00260-Glycine,-serine-and-threonine-metabolism sce01100-Metabolic-pathways	1	-1	-1	1	1	-1
LYSt2r-reverse	M-h-c M-lys–L-c	M-TM-lys–L-c M-h-e M-lys–L-e	NA	-1	-1	1	1	1	1
L-LACtm-reverse	M-h-m M-lac–L-m	M-TM-lac–L-c M-h-c M-lac–L-c	NA	-1	1	-1	-1	-1	1
OHPBAT-reverse	M-akg-c M-phthr-c	M-TM-glu–L-c M-glu–L-c M-ohpb-c	sce00260-Glycine,-serine-and-threonine-metabolism sce00270-Cysteine-and-methionine-metabolism sce00680-Methane-metabolism sce00750-Vitamin-B6-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism sce01230-Biosynthesis-of-amino-acids	-1	0	-1	1	1	1
PHEt2r-reverse	M-h-c M-phe–L-c	M-TM-phe–L-c M-h-e M-phe–L-e	NA	-1	-1	1	1	1	1

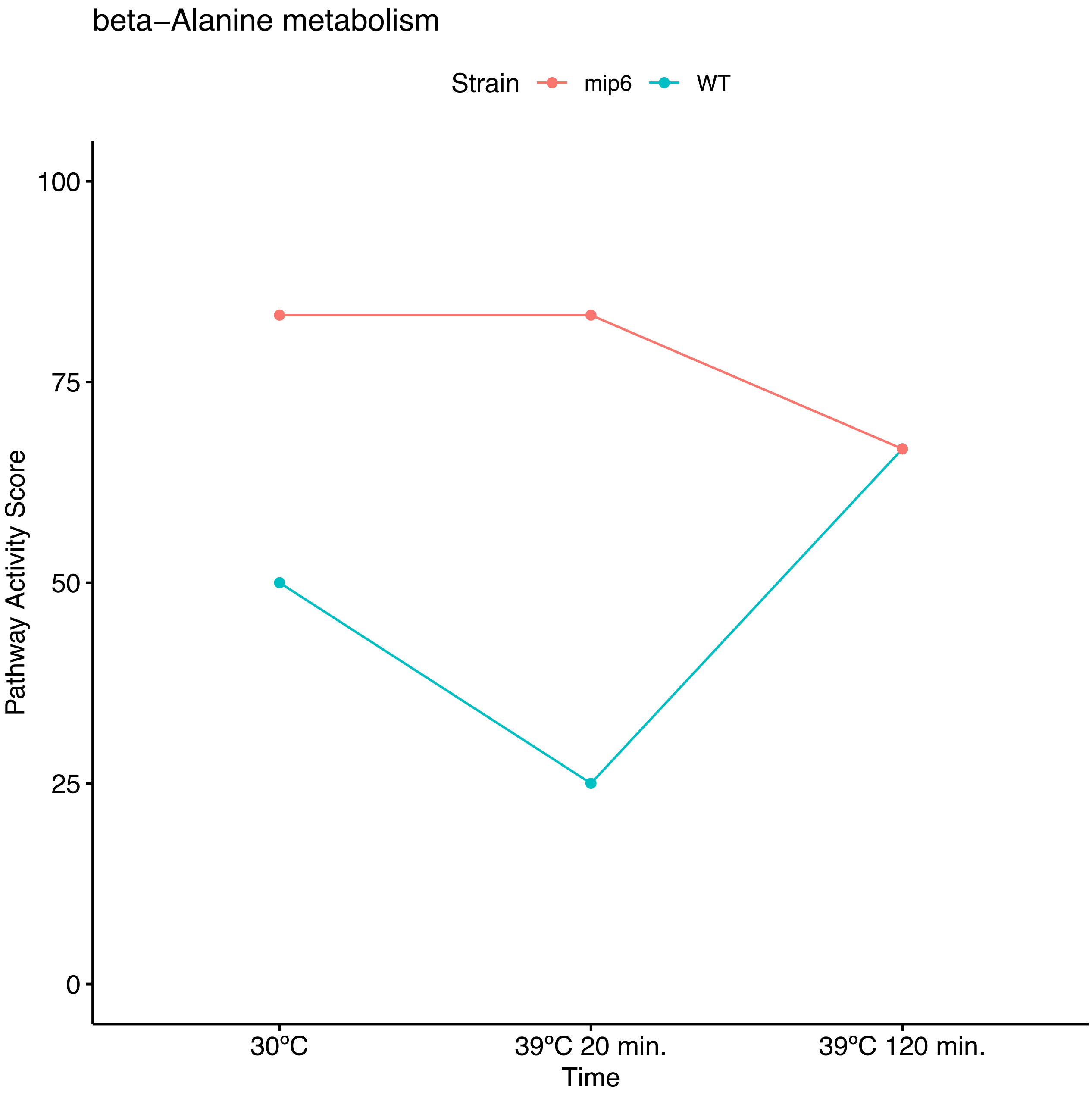
Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
PPM-reverse	M-r5p-c	M-r1p-c	sce00010-Glycolysis-/-Gluconeogenesis sce00030-Pentose-phosphate-pathway sce00052-Galactose-metabolism sce00230-Purine-metabolism sce00500-Starch-and-sucrose-metabolism sce00520-Amino-sugar-and-nucleotide-sugar-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	-1	-1	1	1	1
PRASCSi-reverse	M-25aics-c M-adp-c M-h-c M-pi-c	M-5aizc-c M-TM-asp–L-c M-TM-atp-c M-asp–L-c M-atp-c	sce00230-Purine-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	1	1	-1	1	1	1
PROtm-reverse	M-pro–L-m	M-TM-pro–L-c M-pro–L-c	NA	1	-1	1	-1	-1	1

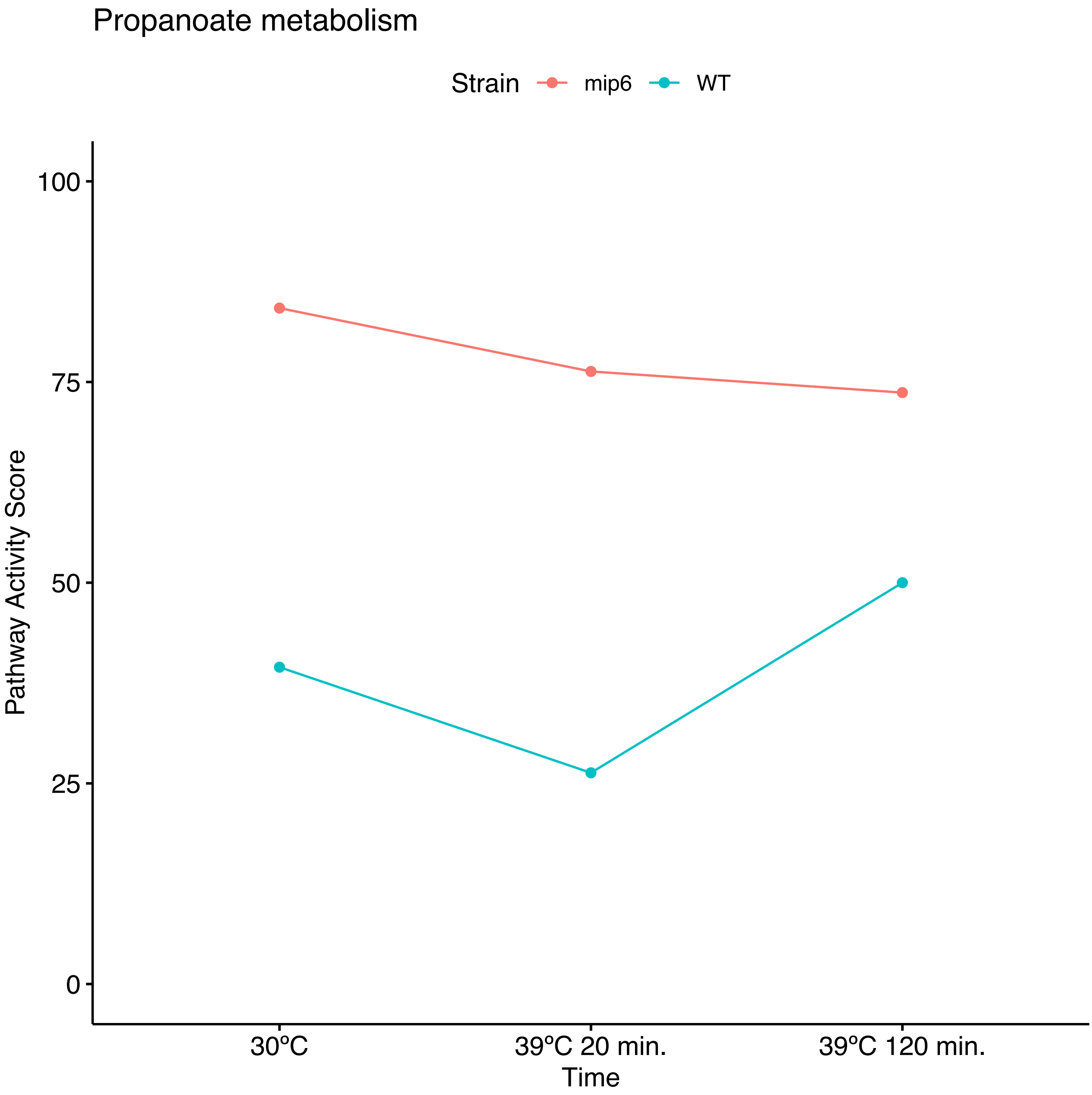
Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
PSERS-SC-reverse	M-cmp-c M-h-c M-ps-SC-c	M-TM-cmp-c M-TM-ser–L-c M-cdpdag-SC-c M-ser–L-c	sce00260-Glycine,-serine-and-threonine-metabolism sce00564-Glycerophospholipid-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	1	1	1	1	1
PSERSm-SC-reverse	M-cmp-m M-h-m M-ps-SC-m	M-cdpdag-SC-m M-ser–L-m	sce00260-Glycine,-serine-and-threonine-metabolism sce00564-Glycerophospholipid-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	1	-1	1	0	1
PTD1INOtn-SC-reverse	M-ptd1ino-SC-n	M-ptd1ino-SC-c	NA	-1	1	1	1	1	1
PUNP6-reverse	M-2dr1p-c M-hxan-c	M-din-c M-pi-c	sce00230-Purine-metabolism sce00240-Pyrimidine-metabolism sce00760-Nicotinate-and-nicotinamide-metabolism sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites	-1	1	1	1	1	1

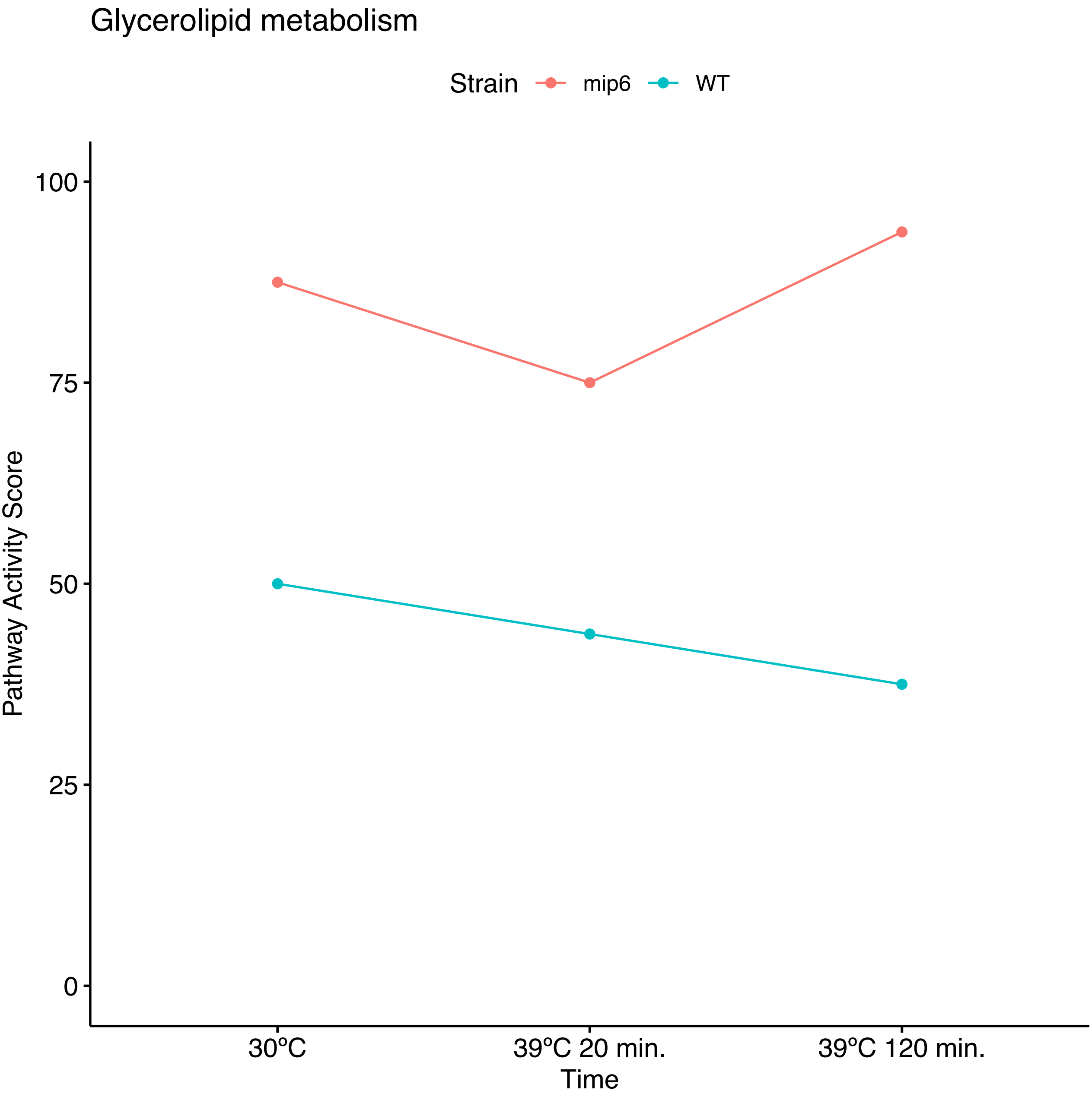
Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
SACCD2-reverse	M-akg-c M-h-c M-lys–L-c M-nadh-c	M-TM-lys–L-c M-TM-nad-c M-h2o-c M-nad-c M-saccrp–L-c	sce00300-Lysine-biosynthesis sce00310-Lysine-degradation sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01230-Biosynthesis-of-amino-acids	-1	-1	-1	1	1	1
SERt2r-reverse	M-h-c M-ser–L-c	M-TM-ser–L-c M-h-e M-ser–L-e	NA	1	-1	-1	1	1	1
SUCD2-u6m-reverse	M-fum-m M-q6h2-m	M-q6-m M-succ-m	sce00020-Citrate-cycle-(TCA-cycle) sce00190-Oxidative-phosphorylation sce01100-Metabolic-pathways sce01110-Biosynthesis-of-secondary-metabolites sce01200-Carbon-metabolism	1	-1	1	1	1	1
THIORDXp-reverse	M-h2o-x M-trdox-x	M-h2o2-x M-trdrd-x	sce04122-Sulfur-relay-system	-1	-1	-1	1	1	1
THRt2r-reverse	M-h-c M-thr–L-c	M-h-e M-thr–L-e	NA	-1	-1	-1	1	1	1
TRDOXtp-reverse	M-trdox-x	M-trdox-c	NA	1	1	-1	1	0	1
TRDRDtp-reverse	M-trdrd-x	M-trdrd-c	NA	1	1	-1	1	0	1
TREt2v-reverse	M-h-v M-tre-v	M-TM-tre-c M-h-c M-tre-c	NA	-1	1	-1	1	1	1

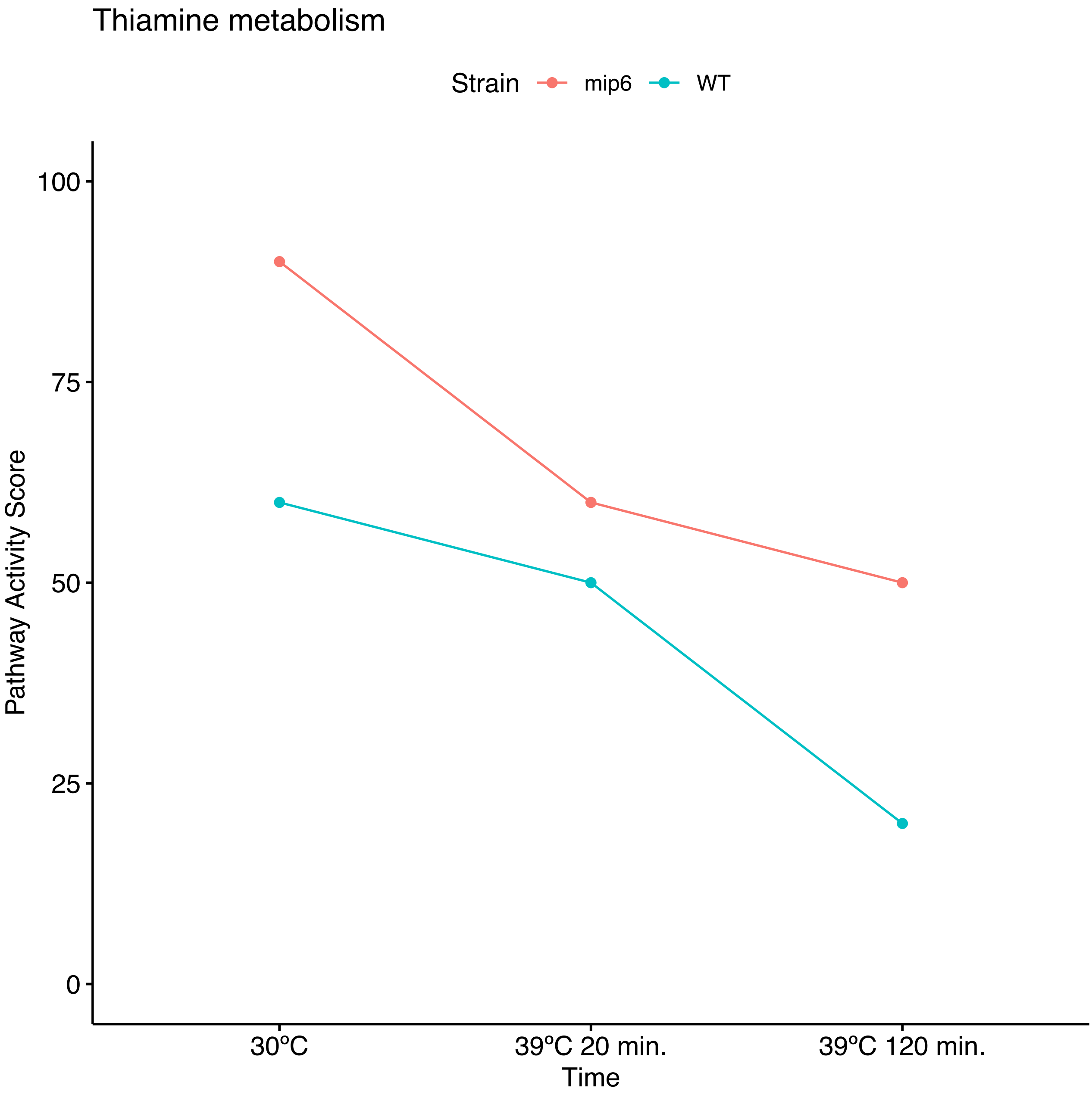
Reaction data				Wild Type			$mip6\Delta$		
Rxn	Reactants	Products	Pathways	t0	t20	t120	t0	t20	t120
URIDK2r-reverse	M-adp-c M-dudp-c	M-TM-atp-c M-atp-c M-dump-c	sce00240-Pyrimidine-metabolism sce01100-Metabolic-pathways	-1	-1	1	1	1	1

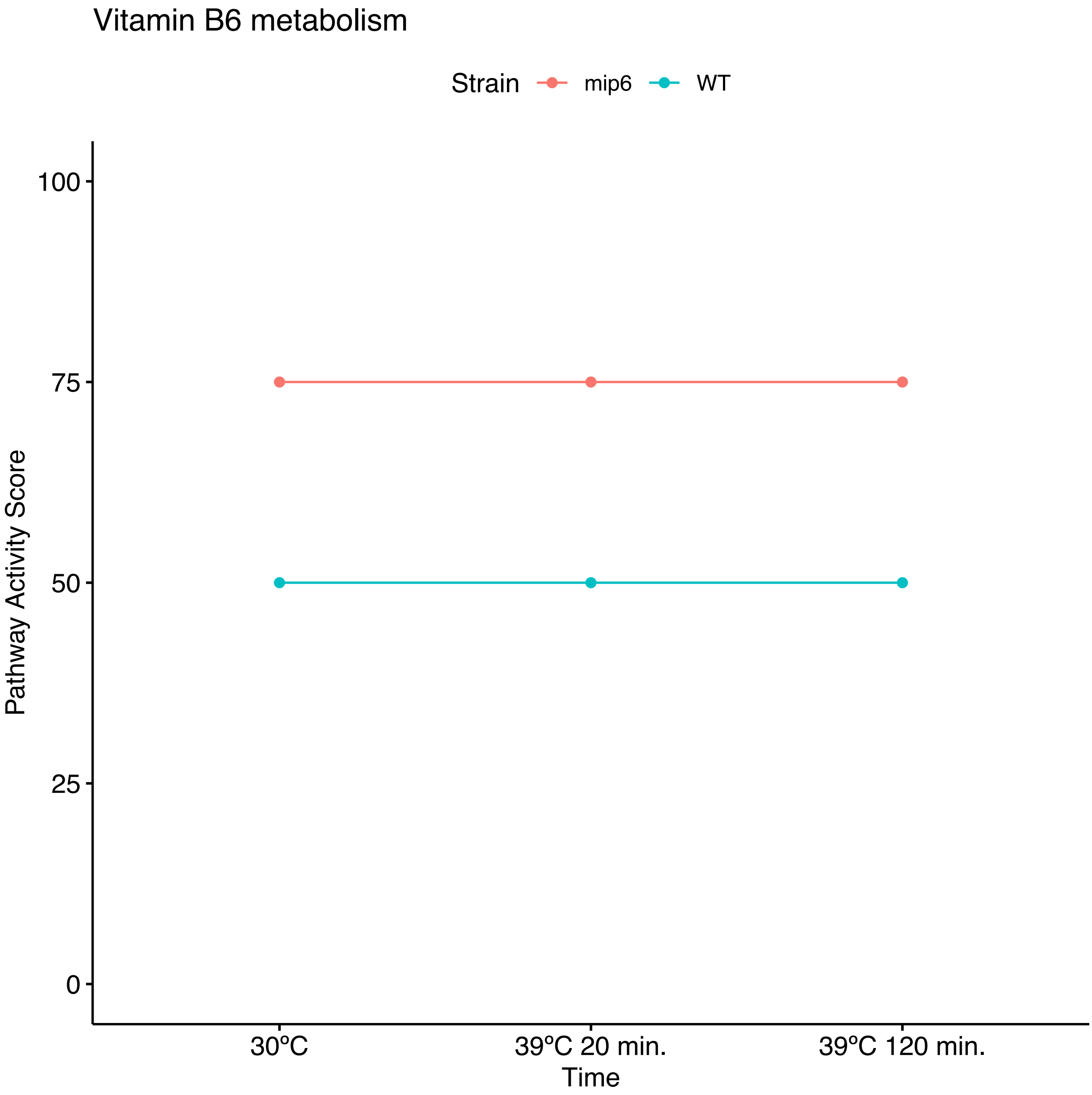
Appendix 5: Pathway Activation Score (PAS) for the list of relevant pathways (Chapter 5)

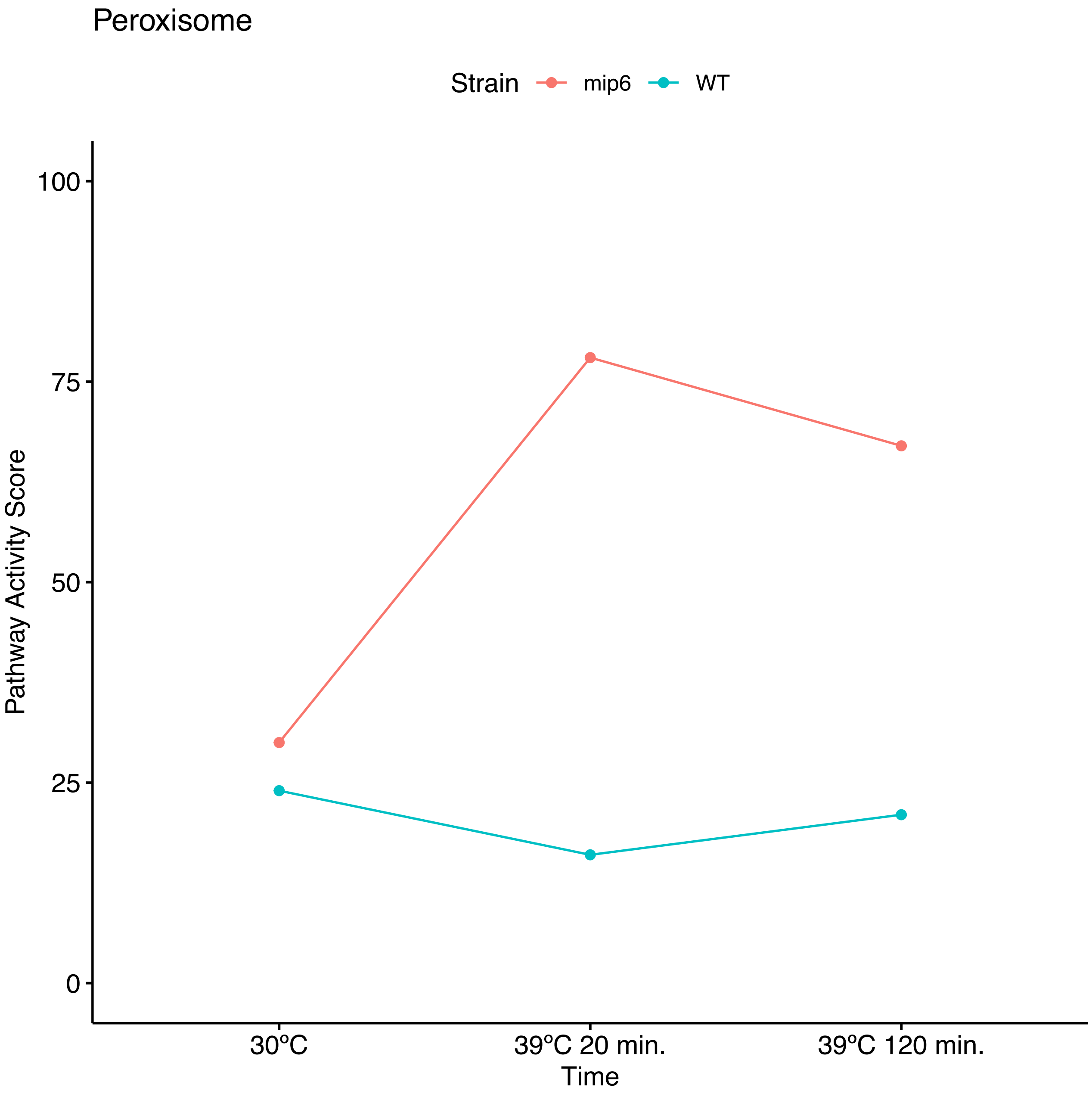

beta−Alanine metabolism
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time

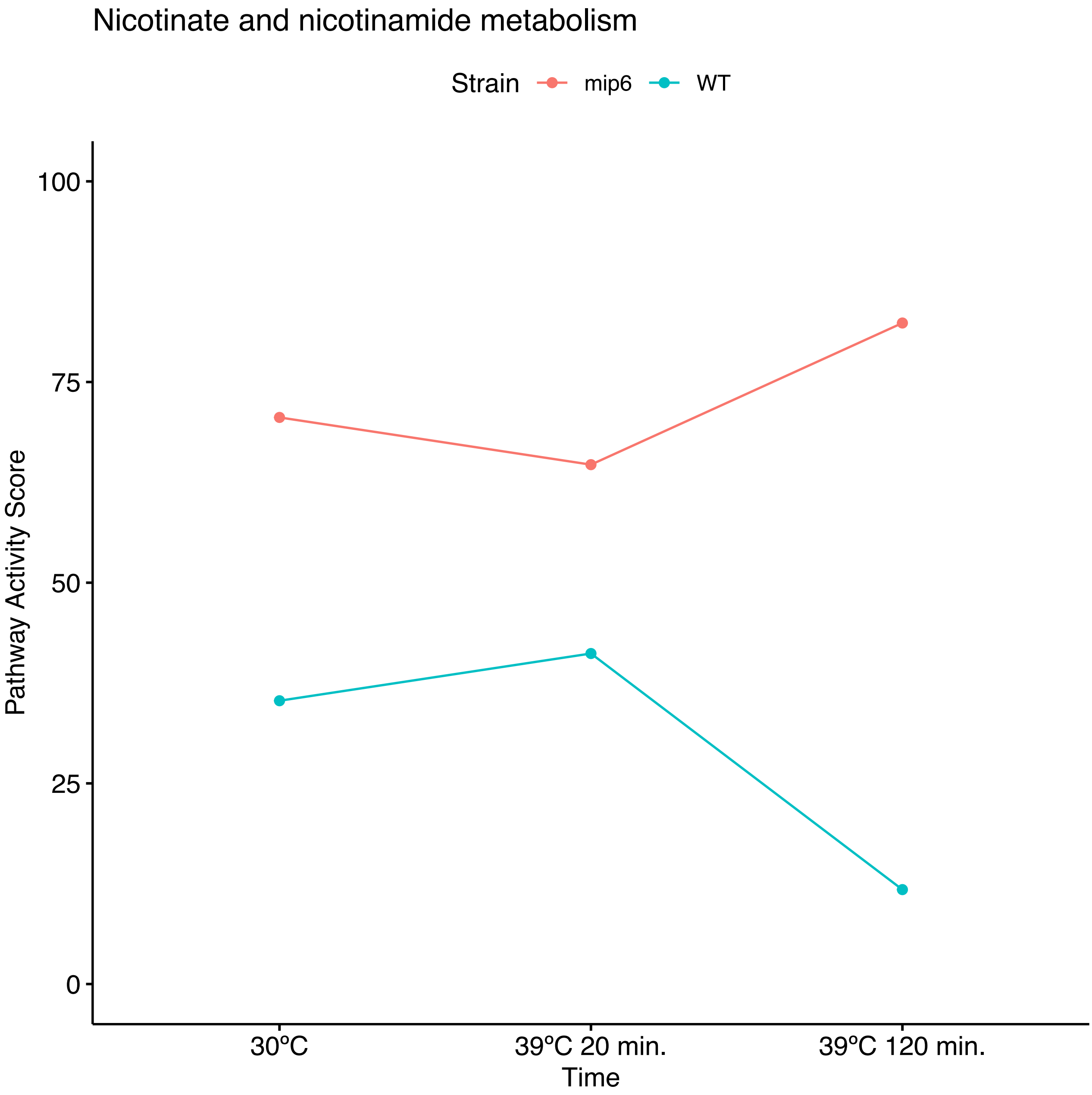

Propanoate metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

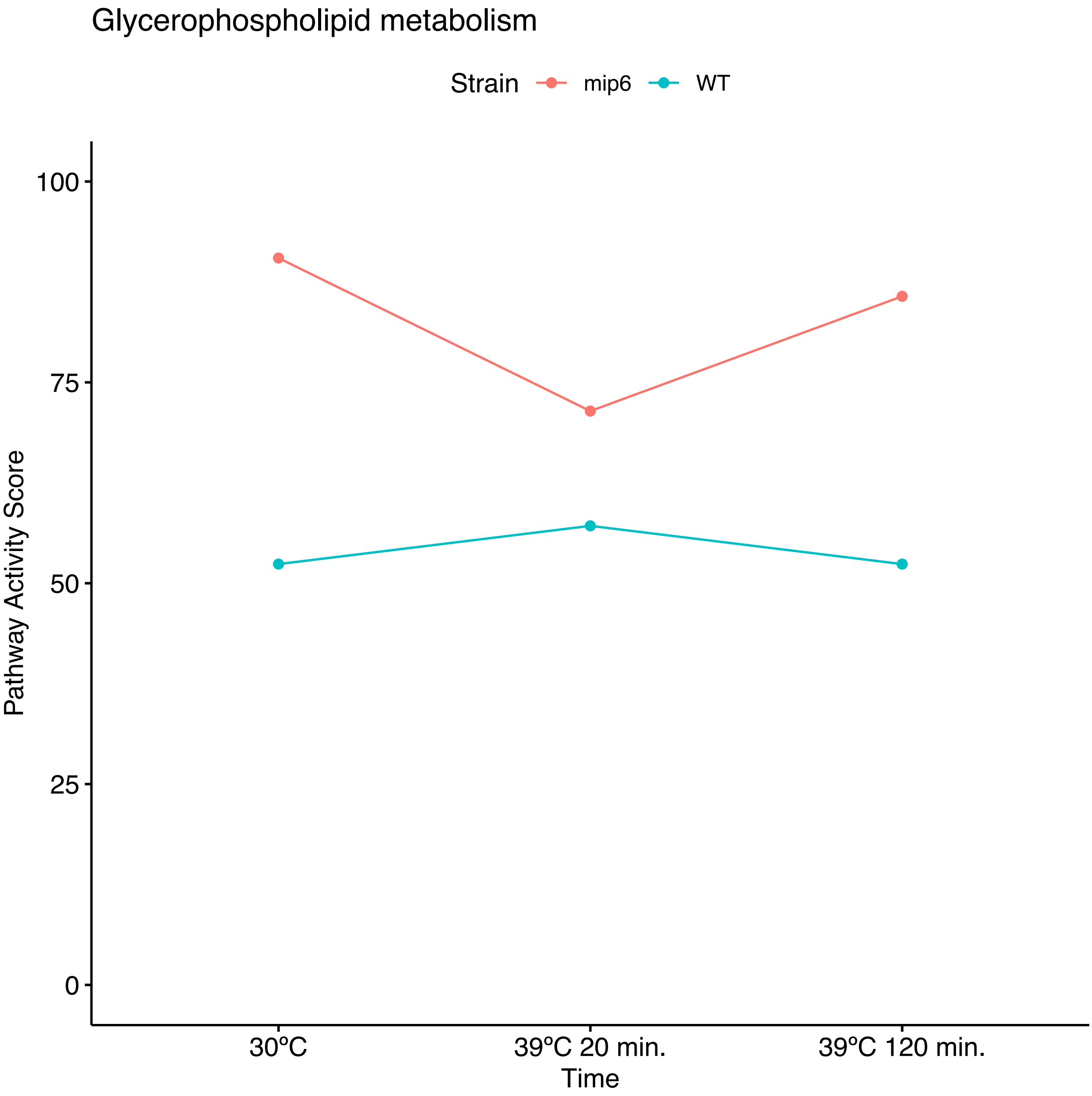

Glycerolipid metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

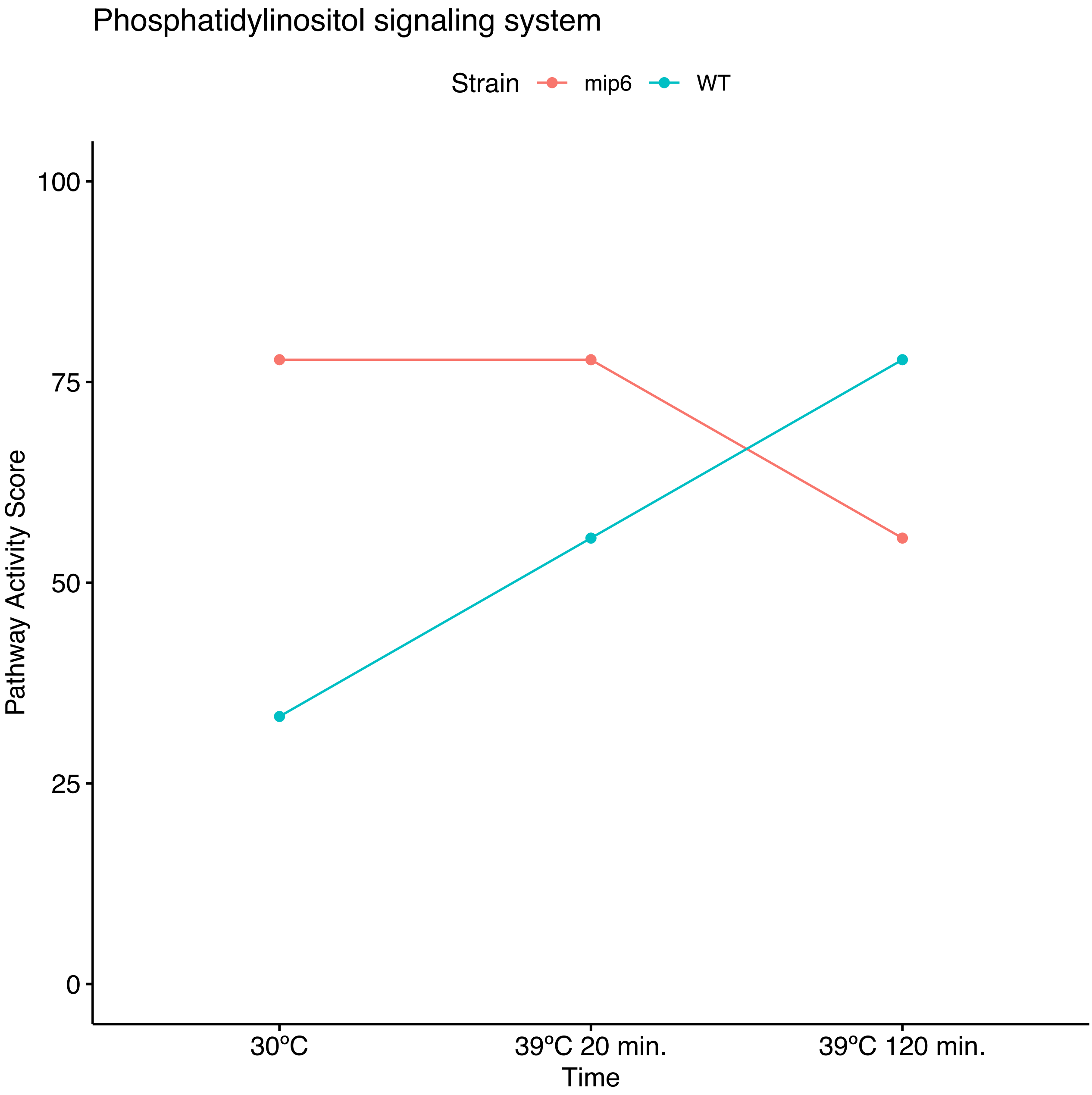

Thiamine metabolism
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time

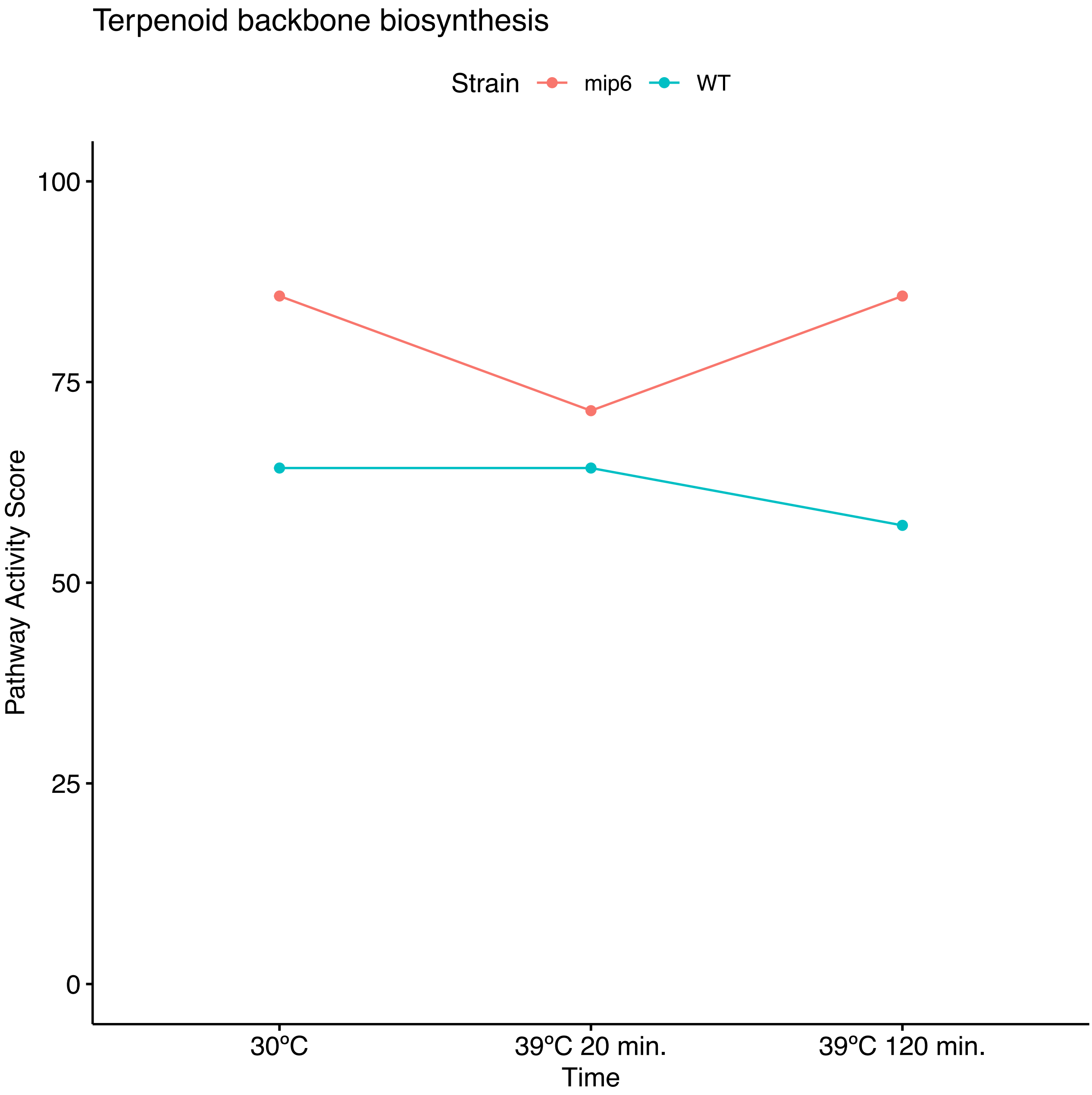

Vitamin B6 metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

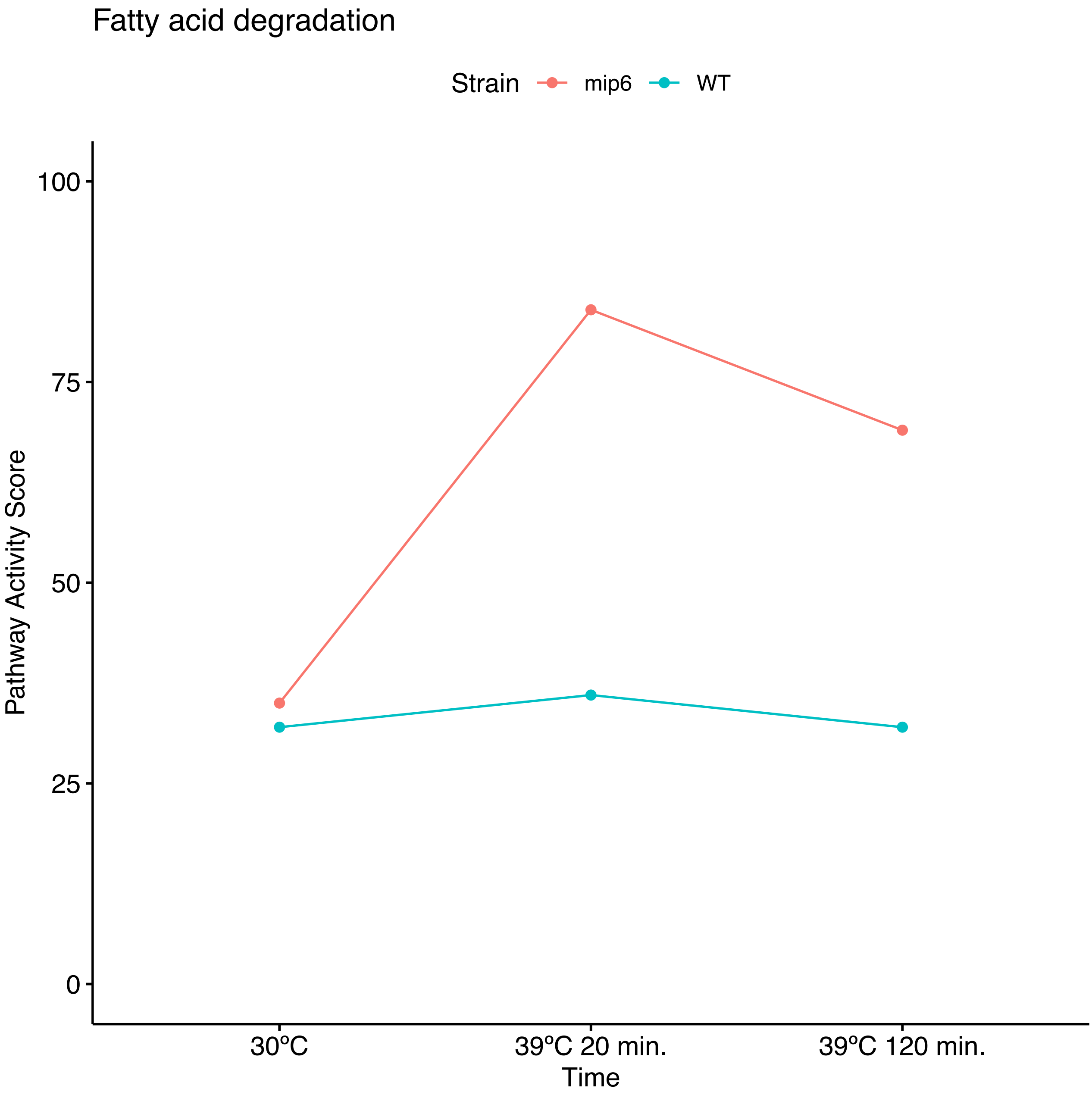

Peroxisome
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time

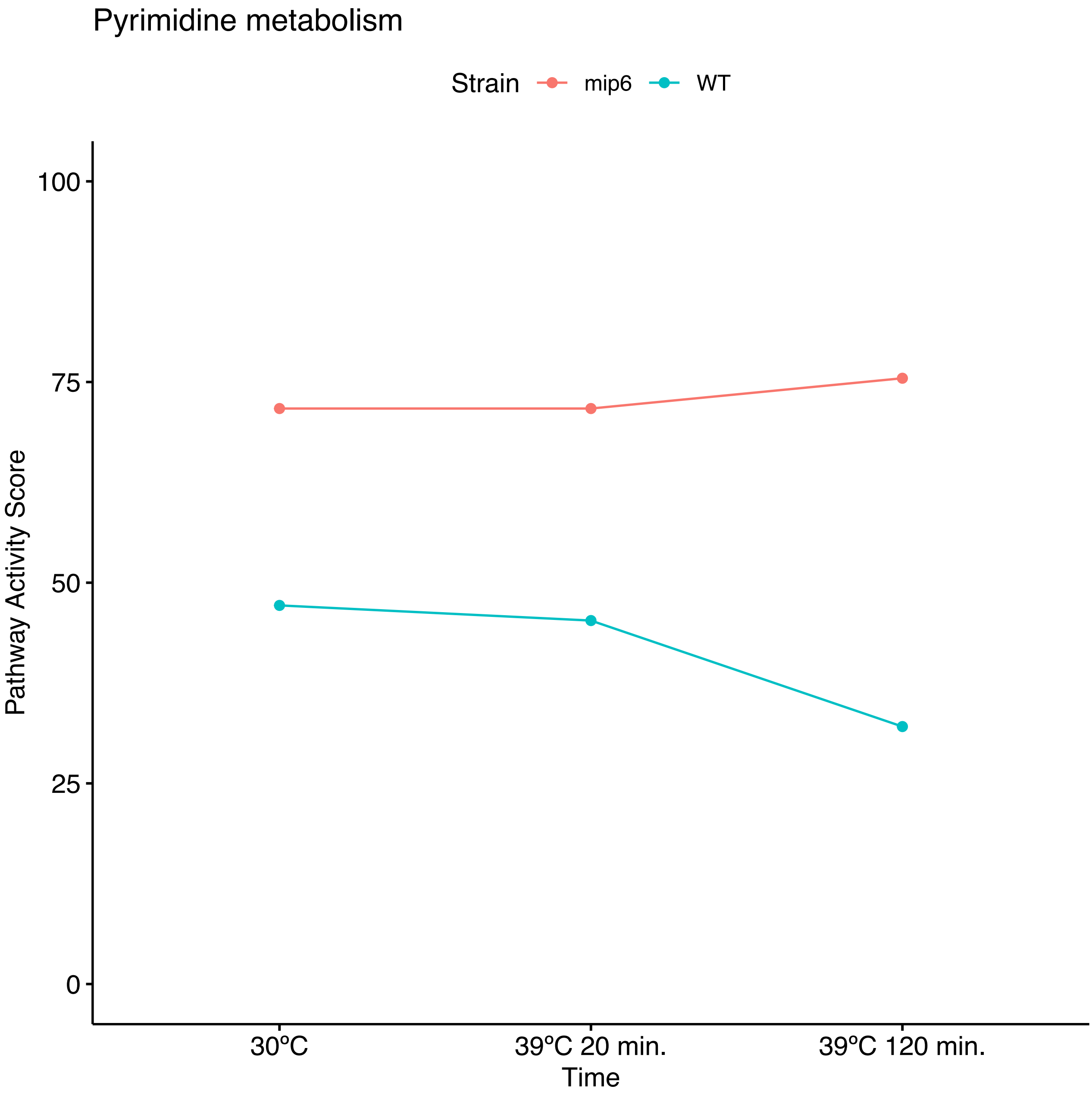

Nicotinate and nicotinamide metabolism
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time

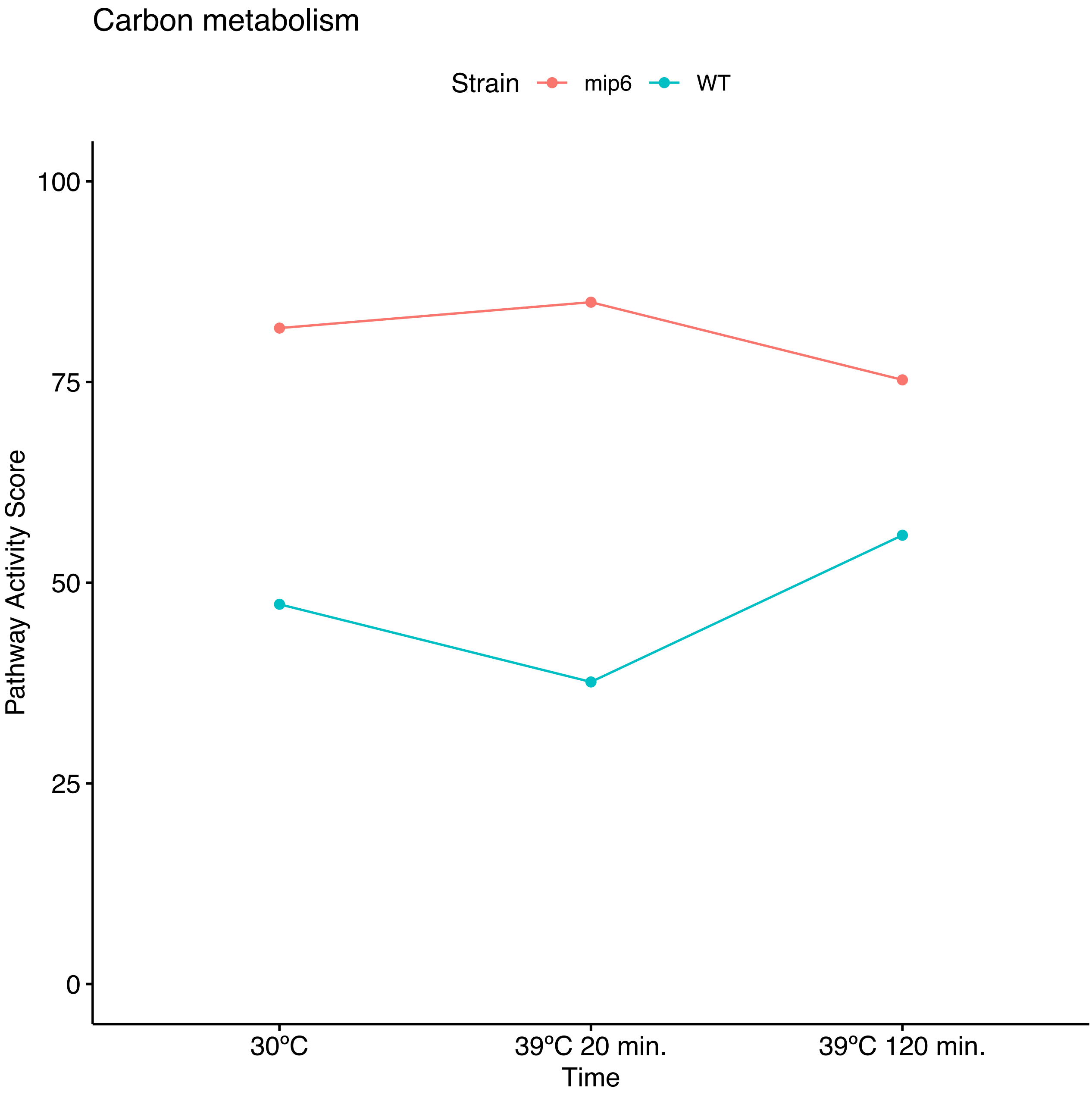

Glycerophospholipid metabolism
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time

Phosphatidylinositol signaling system
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time

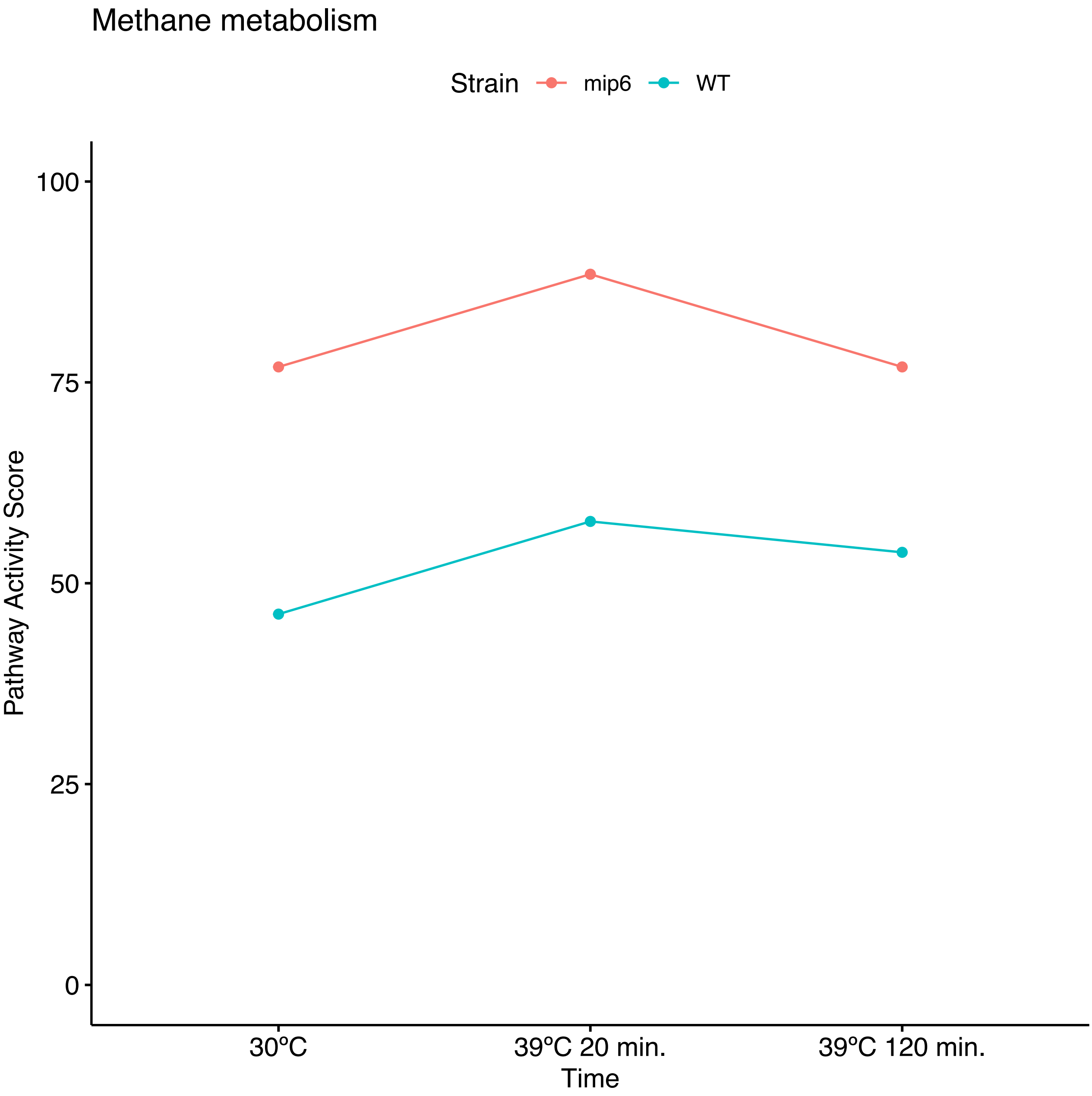

Terpenoid backbone biosynthesis
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

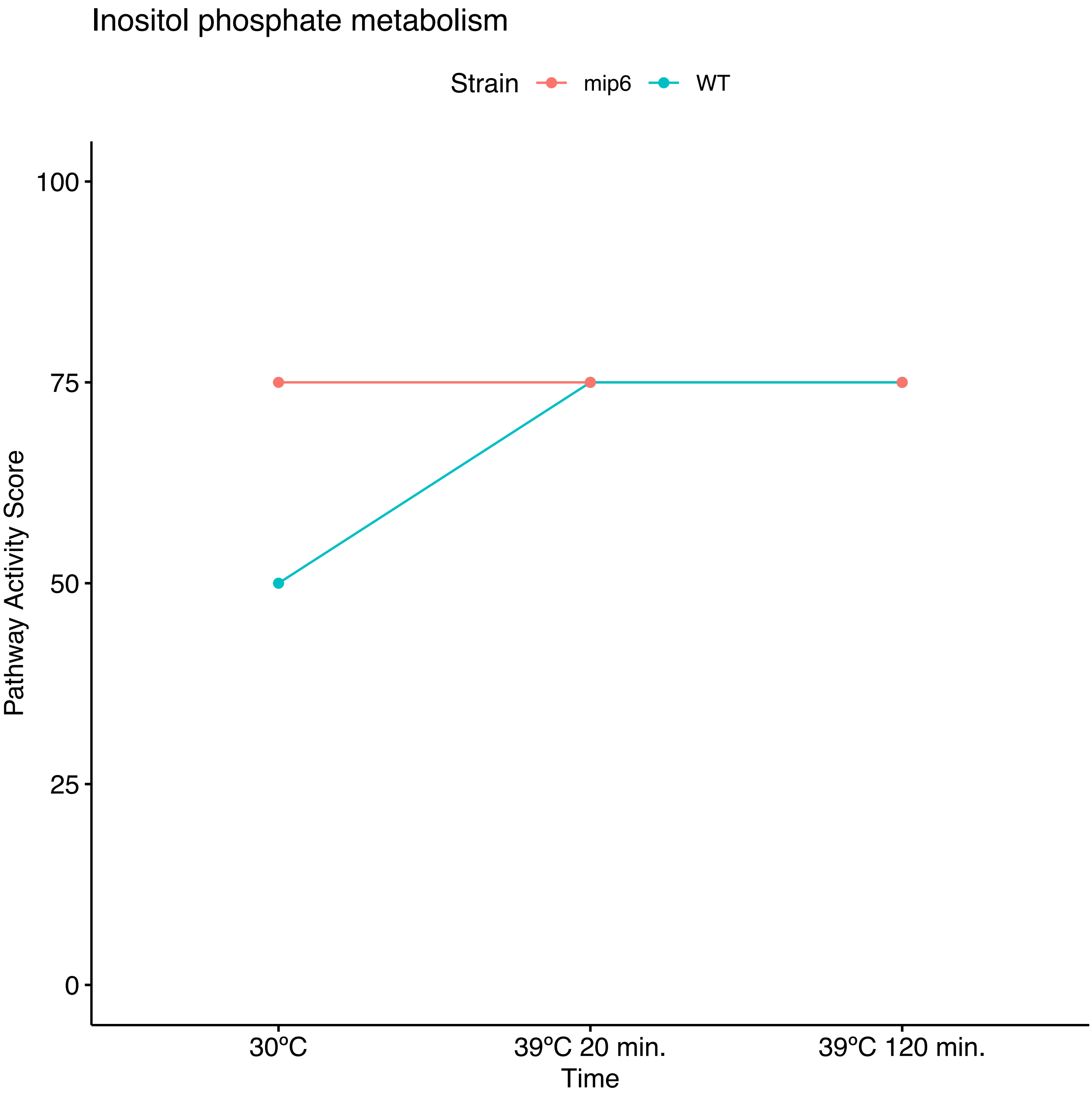

Fatty acid degradation
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time

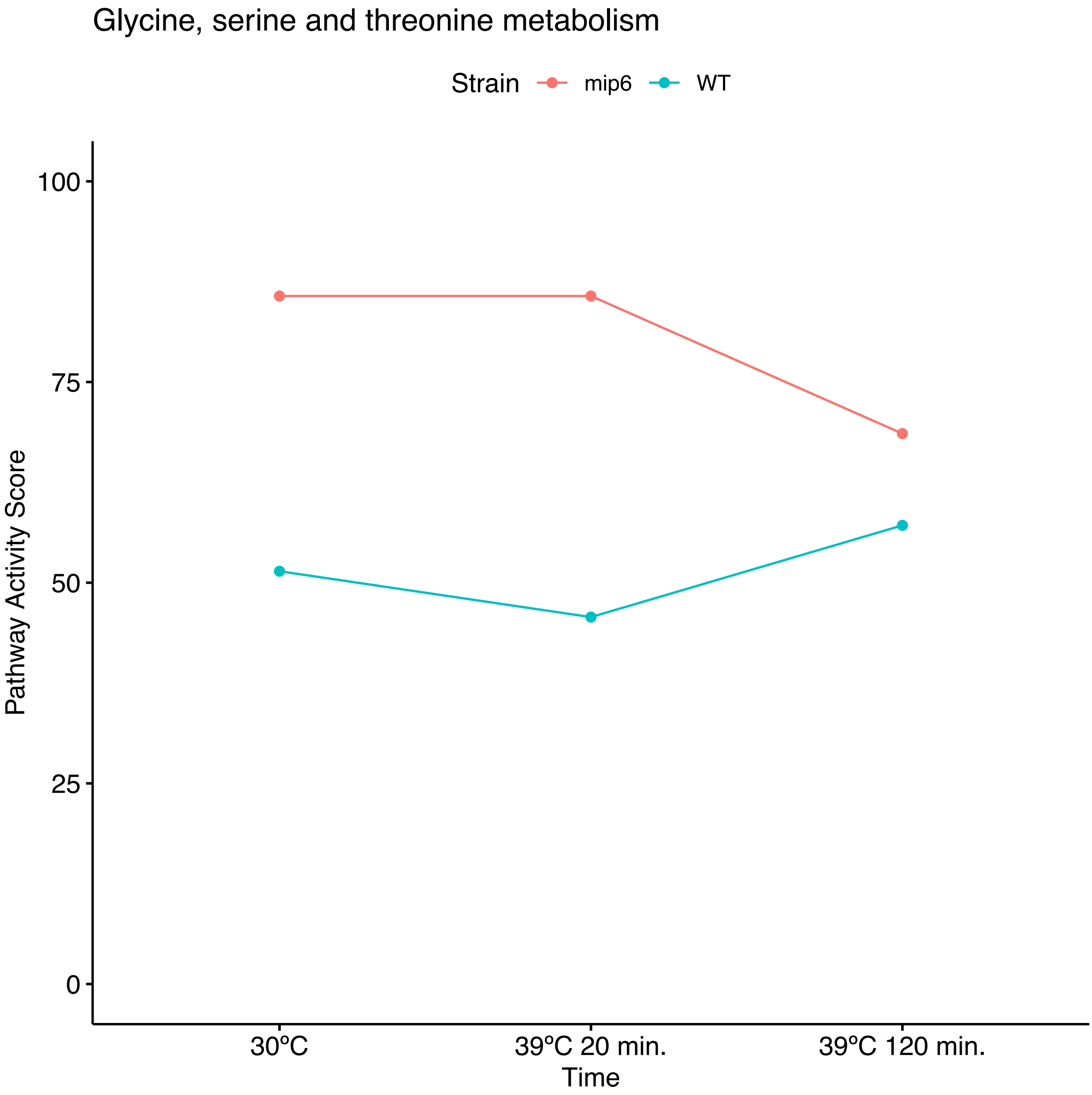

Pyrimidine metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

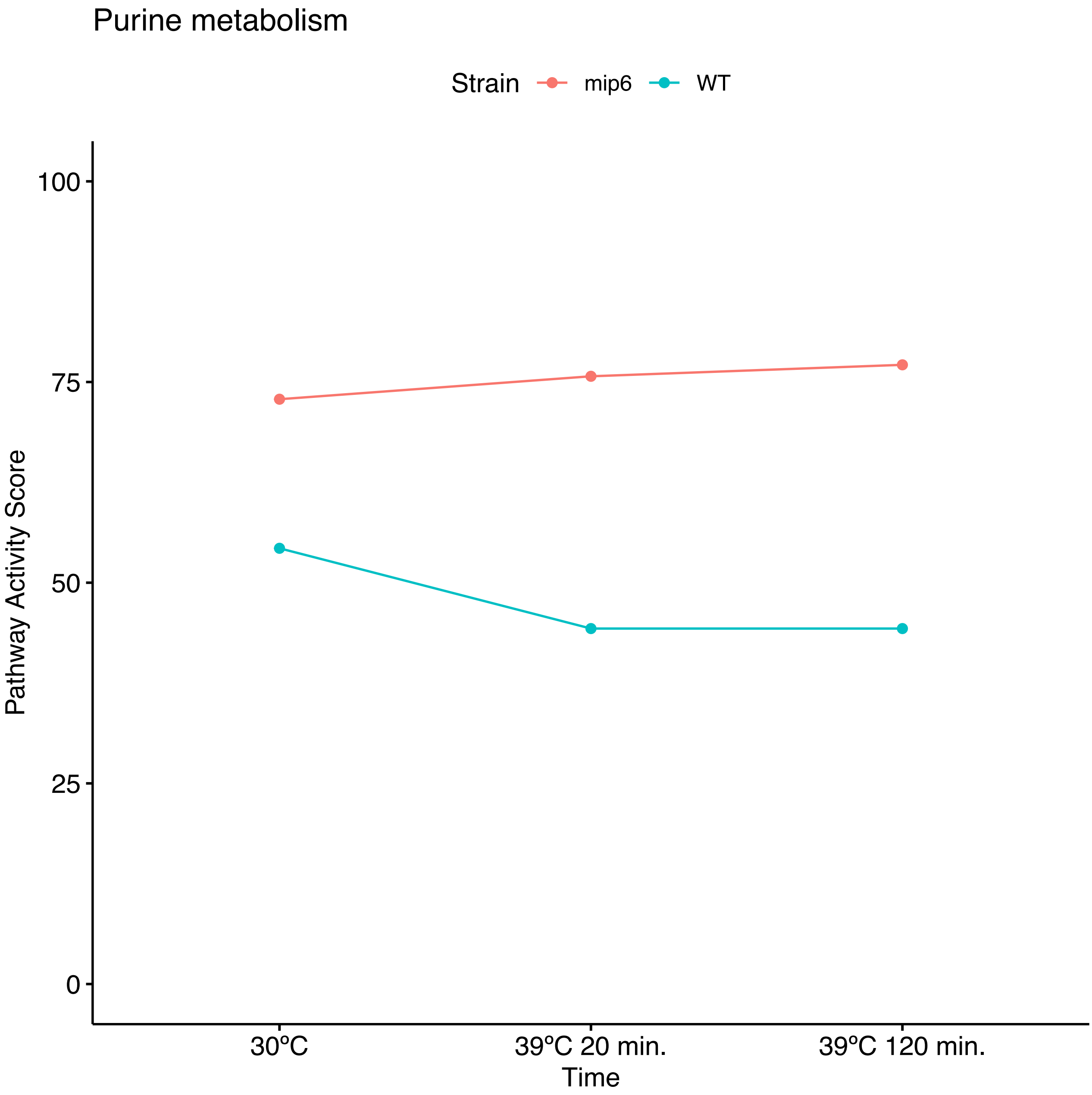


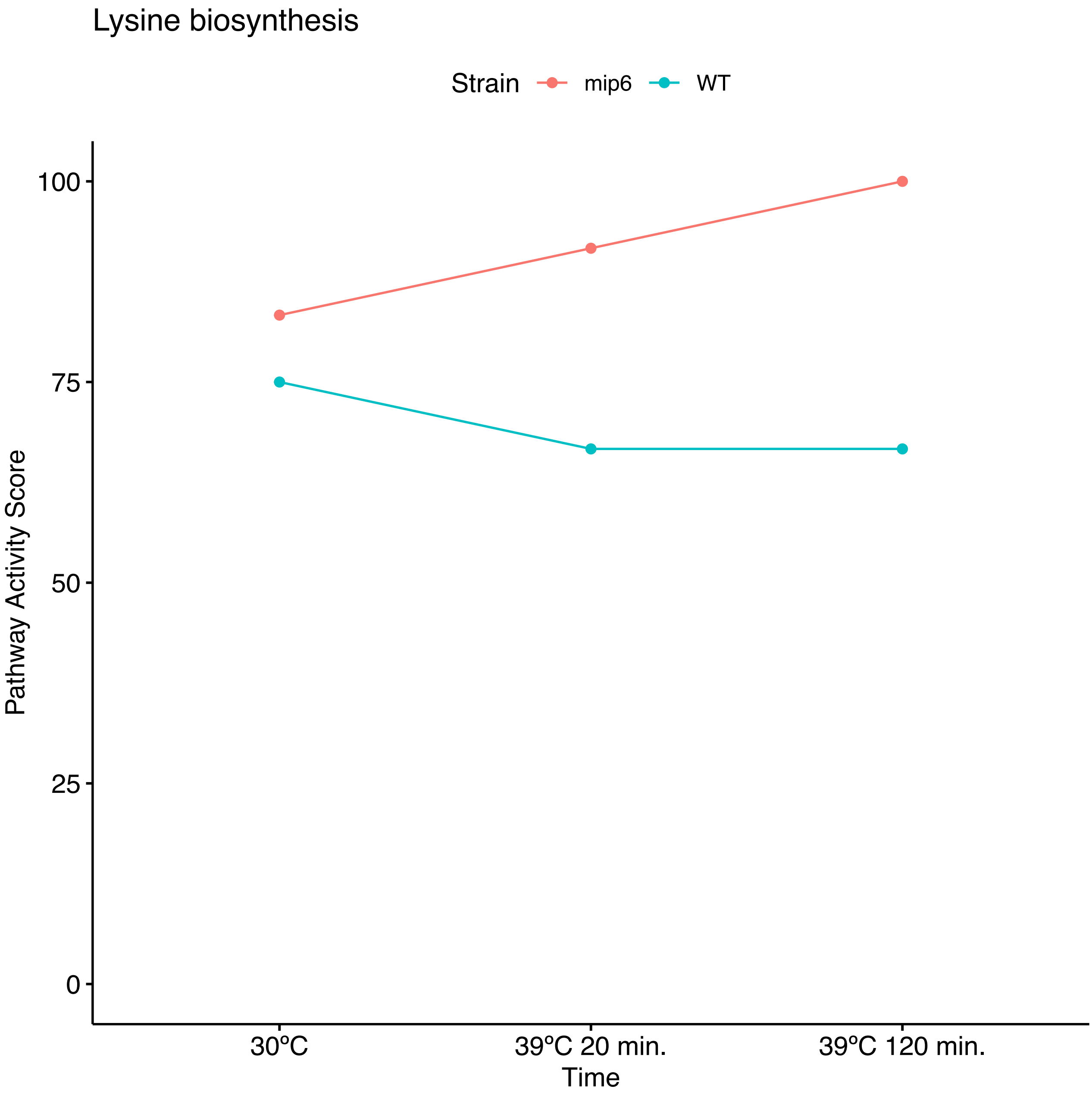
Carbon metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

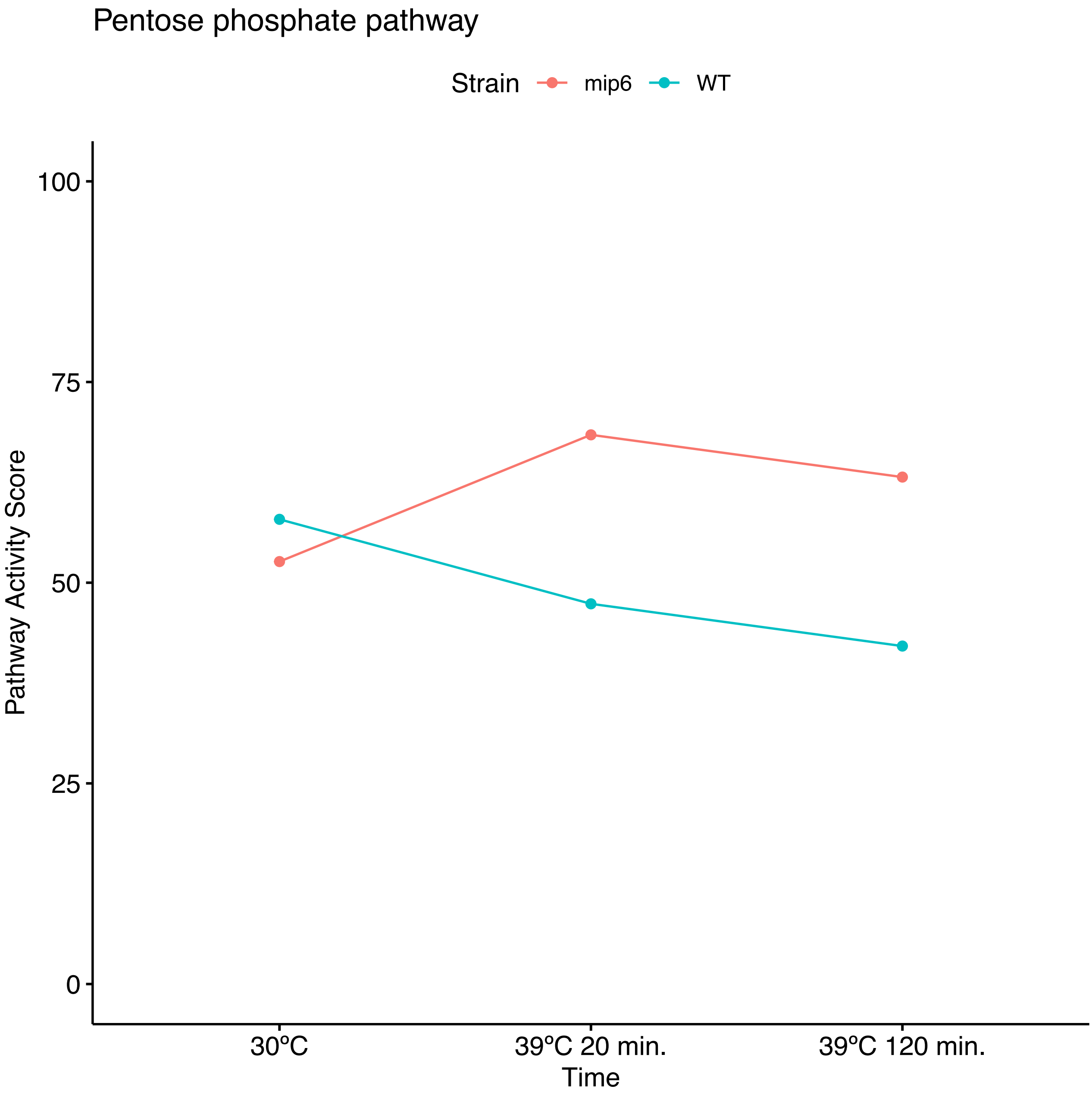

Methane metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

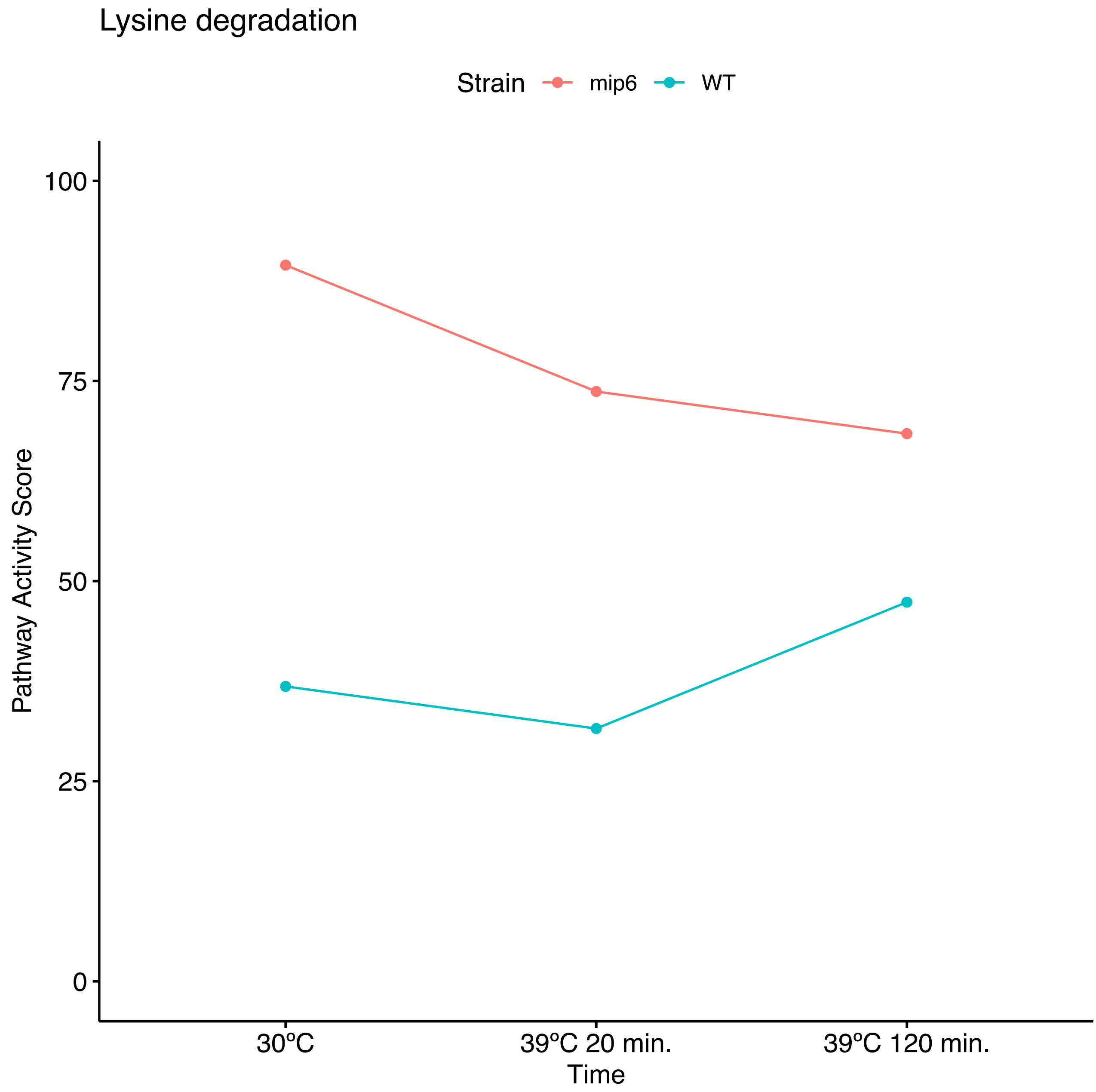

Inositol phosphate metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

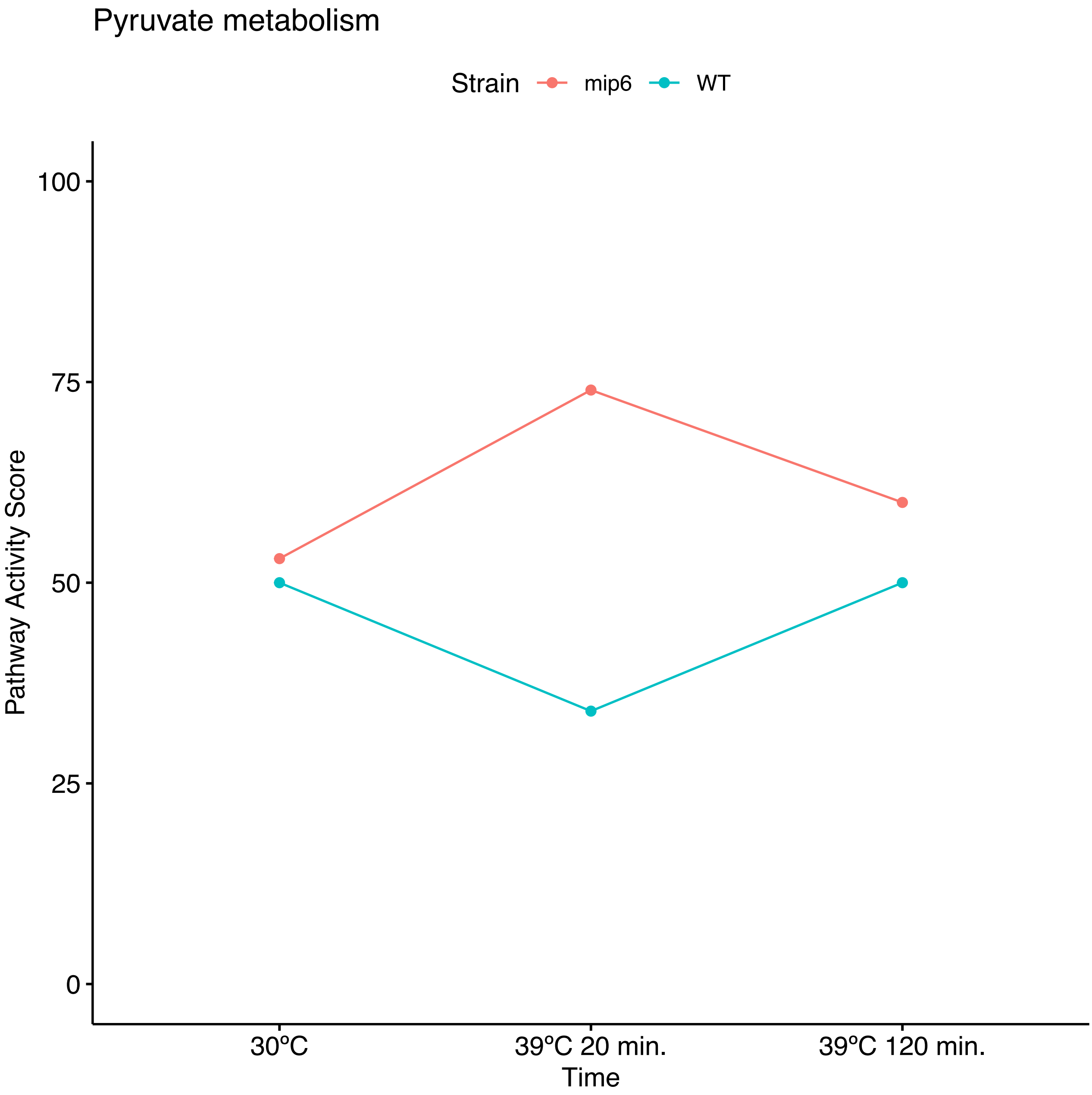

Glycine, serine and threonine metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

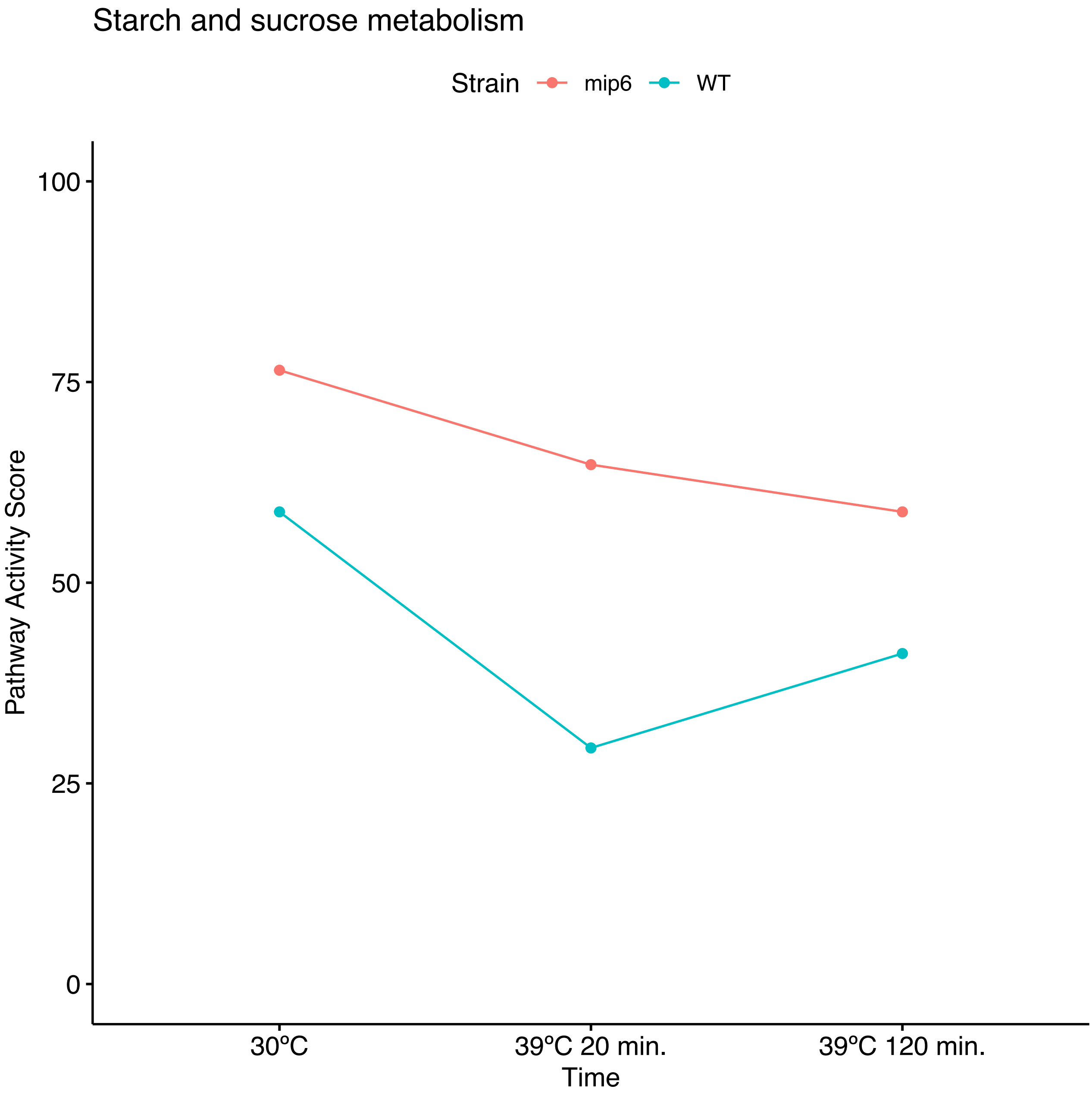

Purine metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time

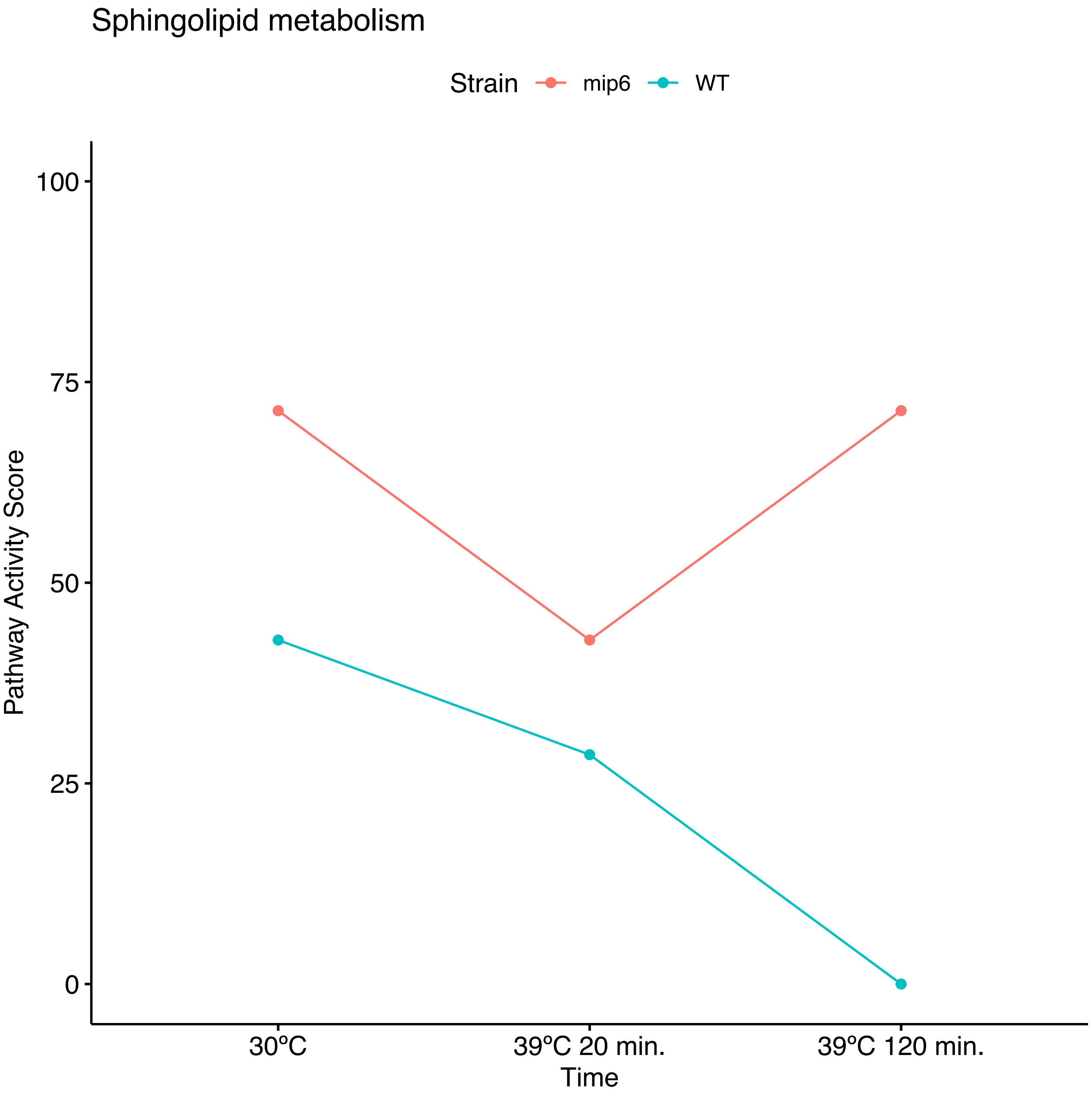

Lysine biosynthesis
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time


Pentose phosphate pathway
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time


Lysine degradation
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time


Pyruvate metabolism
Strain
mip6
WT
Pathway Activity Score
100
75
50
25
0
30ºC
39ºC 20 min.
39ºC 120 min.
Time


Starch and sucrose metabolism
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time


Sphingolipid metabolism
Strain
mip6
WT
100
75
50
25
0
Pathway Activity Score
30ºC
39ºC 20 min.
39ºC 120 min.
Time

References

[1] Ludwig von Bertalanffy. *General System Theory: Foundations, Development, Applications*. 1968 (cit. on p. 3).

[2] Alexander Powell et al. “Disciplinary baptisms: a comparison of the naming stories of genetics, molecular biology, genomics, and systems biology.” In: *History and Philosophy of the Life Sciences* 29.1 (2007), pp. 5–32. ISSN: 0391-9714 (cit. on p. 3).

[3] Rainer Breitling. “What is systems biology?” In: *Frontiers in Physiology* 1 (2010), p. 9. ISSN: 1664-042X (cit. on p. 3).

[4] Maureen A O’Malley. “Evolutionary systems biology: historical and philosophical perspectives on an emerging synthesis.” In: *Advances in Experimental Medicine and Biology* 751 (2012), pp. 1–28. ISSN: 0065-2598 (cit. on p. 3).

[5] Hans V Westerhoff and Bernhard O Palsson. “The evolution of molecular biology into systems biology”. In: *Nature Biotech-*

nology 22.10 (2004), pp. 1249–1252. ISSN: 1546-1696 (cit. on p. 4).

[6] Srdjan Kesić. "Systems biology, emergence and antireductionism". In: *Saudi Journal of Biological Sciences* 23.5 (2016), pp. 584–591. ISSN: 1319-562X (cit. on p. 4).

[7] Katryna Cisek et al. "The application of multi-omics and systems biology to identify therapeutic targets in chronic kidney disease." In: *Nephrology, Dialysis, Transplantation: official publication of the European Dialysis and Transplant Association - European Renal Association* 31.12 (2016), pp. 2003–2011. ISSN: 1460-2385 (cit. on p. 4).

[8] Markus J Herrgård et al. "A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology." In: *Nature Biotechnology* 26.10 (2008), pp. 1155–1160. ISSN: 1546-1696 (cit. on p. 4).

[9] Ines Thiele et al. "A community-driven global reconstruction of human metabolism." In: *Nature Biotechnology* 31.5 (2013), pp. 419–425. ISSN: 1546-1696 (cit. on pp. 4, 149).

[10] Bernhard Palsson and Karsten Zengler. "The challenges of integrating multi-omic data sets." In: *Nature Chemical Biology* 6.11 (2010), pp. 787–789. ISSN: 1552-4469 (cit. on pp. 5, 15).

[11] Sonia Tarazona, Angeles Arzalluz-Luque, and Ana Conesa. "Undisclosed, unmet and neglected challenges in multi-omics

studies". In: *Nature Computational Science* 1.6 (2021), pp. 395–402. ISSN: 2662-8457 (cit. on pp. 5, 15).

[12] Elaine R Mardis. "Next-generation sequencing platforms." In: *Annual review of analytical chemistry (Palo Alto, Calif.)* 6 (2013), pp. 287–303. ISSN: 1936-1335 (cit. on p. 5).

[13] Hyeonju Woo et al. "Modulation of gene expression dynamics by co-transcriptional histone methylations". In: *Experimental & Molecular Medicine* 49.4 (2017), e326–e326. ISSN: 2092-6413 (cit. on pp. 5, 113).

[14] Peter J Park. "ChIPseq: advantages and challenges of a maturing technology". In: *Nature Reviews Genetics* 10.10 (2009), pp. 669–680. ISSN: 1471-0064 (cit. on p. 6).

[15] Jason D Buenrostro et al. "Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position". In: *Nature Methods* 10.12 (2013), pp. 1213–1218. ISSN: 1548-7105 (cit. on p. 6).

[16] Leighton J Core, Joshua J Waterfall, and John T Lis. "Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters." In: *Science (New York, N.Y.)* 322.5909 (2008), pp. 1845–1848. ISSN: 1095-9203 (cit. on p. 6).

[17] L Stirling Churchman and Jonathan S Weissman. "Native Elongating Transcript Sequencing (NET-seq)". In: *Current Protocols in Molecular Biology* 98.1 (2012), pp. 1–17. ISSN: 1934-3639 (cit. on p. 6).

[18] John C Zinder and Christopher D Lima. “Targeting RNA for processing or destruction by the eukaryotic RNA exosome and its cofactors”. In: *Genes & Development* 31.2 (2017), pp. 88–100. ISSN: 1549-5477 (cit. on pp. 6, 113).

[19] Aleksandra Helwak et al. “Mapping the Human miRNA Interactome by CLASH Reveals Frequent Noncanonical Binding”. In: *Cell* 153.3 (2013), pp. 654–665. ISSN: 0092-8674 (cit. on pp. 6, 113).

[20] Gesa Zander et al. “mRNA quality control is bypassed for immediate export of stress-responsive transcripts”. In: *Nature* 540.7634 (2016), pp. 593–596. ISSN: 1476-4687 (cit. on p. 6).

[21] Je-Hyun Yoon et al. “PAR-CLIP analysis uncovers AUF1 impact on target RNA fate and genome integrity”. In: *Nature Communications* 5.1 (2014), p. 5248. ISSN: 2041-1723 (cit. on p. 6).

[22] Florian Erhard et al. “Improved Ribo-seq enables identification of cryptic translation events.” In: *Nature Methods* 15.5 (2018), pp. 363–366. ISSN: 1548-7105 (cit. on p. 6).

[23] K. Chandrasekhar et al. “A Short Review on Proteomics and its Applications”. In: *International Letters of Natural Sciences* 17.September 2015 (2014), pp. 77–84 (cit. on pp. 6, 8).

[24] Sonia Tarazona et al. “Harmonization of quality metrics and power calculation in multi-omic studies”. In: *Nature Communications* 11.1 (2020), p. 3092. ISSN: 2041-1723 (cit. on p. 8).

[25] Janine Egert et al. “DIMA: Data-Driven Selection of an Imputation Algorithm”. In: *Journal of Proteome Research* 20.7 (2021), pp. 3489–3496. ISSN: 1535-3893 (cit. on p. 8).

[26] Igor Bartolomé Marín de Mas. *Development and application of novel model-driven and data-driven approaches to study metabolism in the framework of systems medicine*. 2015 (cit. on p. 9).

[27] Abdul-Hamid Emwas et al. “NMR Spectroscopy for Metabolomics Research”. In: *Metabolites* 9.7 (2019), p. 123. ISSN: 2218-1989 (cit. on pp. 9, 12, 13).

[28] Uwe Sauer. “Metabolic networks in motion: 13C-based flux analysis.” In: *Molecular Systems Biology* 2 (2006), p. 62. ISSN: 1744-4292 (cit. on pp. 10, 39, 151).

[29] Uros Rajcevic et al. “Colorectal cancer derived organotypic spheroids maintain essential tissue characteristics but adapt their metabolism in culture”. In: *Proteome Science* 12.1 (2014), p. 39. ISSN: 1477-5956 (cit. on p. 10).

[30] Hyun Uk Kim et al. “Integrative genome-scale metabolic analysis of Vibrio vulnificus for drug targeting and discovery.” In: *Molecular Systems Biology* 7 (2011), p. 460. ISSN: 1744-4292 (cit. on p. 10).

[31] Fumio Matsuda et al. “Engineering strategy of yeast metabolism for higher alcohol production”. In: *Microbial Cell Factories* 10.1 (2011), p. 70. ISSN: 1475-2859 (cit. on p. 10).

[32] Prahlad T Ram, John Mendelsohn, and Gordon B Mills. "Bioinformatics and systems biology." In: *Molecular Oncology* 6.2 (2012), pp. 147–154. ISSN: 1878-0261 (cit. on p. 11).

[33] James H Bullard et al. "Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments". In: *Bioinformatics* 11.1 (2010), p. 94. ISSN: 1471-2105 (cit. on pp. 12, 130).

[34] Simon Anders, Paul Theodor Pyl, and Wolfgang Huber. "HT-Seqa Python framework to work with high-throughput sequencing data". In: *Bioinformatics* 31.2 (2015), pp. 166–169. ISSN: 1367-4803 (cit. on pp. 12, 61, 129, 133).

[35] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data". In: *Bioinformatics* 26.1 (2010), pp. 139–140. ISSN: 1367-4803 (cit. on pp. 12, 20, 60, 115, 126).

[36] Martina Palomino-Schätzlein et al. "Optimised protocols for the metabolic profiling of S. cerevisiae by 1H-NMR and HRMAS spectroscopy". In: *Analytical and Bioanalytical Chemistry* 405.26 (2013), pp. 8431–8441. ISSN: 1618-2650 (cit. on pp. 13, 16, 115, 130).

[37] David S Wishart et al. "HMDB: the Human Metabolome Database". In: *Nucleic Acids Research* 35.suppl$_1$ (2007), pp. D521–D526. ISSN: 0305-1048 (cit. on p. 13).

[38] Hanna E Röhnisch et al. "AQuA: An Automated Quantification Algorithm for High-Throughput NMR-Based Metabolomics and Its Application in Human Plasma". In: *Analytical Chemistry* 90.3 (2018), pp. 2095–2102. ISSN: 0003-2700 (cit. on p. 13).

[39] Jie Hao et al. "BATMAN–an R package for the automated quantification of metabolites from nuclear magnetic resonance spectra using a Bayesian model." In: *Bioinformatics* 28.15 (2012), pp. 2088–2090. ISSN: 1367-4811 (cit. on p. 13).

[40] Miroslava Cuperlovic-Culf et al. "H-NMR Metabolomics Analysis of Glioblastoma Subtypes: CORRELATION BETWEEN METABOLOMICS AND GENE EXPRESSION CHARACTERISTICS". In: *Journal of Biological Chemistry* 287.24 (2012), pp. 20164–20175. ISSN: 0021-9258 (cit. on p. 13).

[41] Albert-László Barabási and Zoltán N Oltvai. "Network biology: understanding the cell's functional organization". In: *Nature Reviews Genetics* 5.2 (2004), pp. 101–113. ISSN: 1471-0064 (cit. on p. 15).

[42] Charles Auffray, Zhu Chen, and Leroy Hood. "Systems medicine: the future of medical genomics and healthcare". In: *Genome Medicine* 1.1 (2009), p. 2. ISSN: 1756-994X (cit. on p. 15).

[43] Jean-Marie Aerts et al. "From data patterns to mechanistic models in acute critical illness." In: *Journal of Critical Care* 29.4 (2014), pp. 604–610. ISSN: 1557-8615 (cit. on p. 15).

[44] Ciaran Evans, Johanna Hardin, and Daniel M Stoebel. "Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions". In: *Briefings in Bioinformatics* 19.5 (2018), pp. 776–792. ISSN: 1477-4054 (cit. on p. 16).

[45] Bo Wang, Aaron M Goodpaster, and Michael A Kennedy. "Coefficient of Variation, Signal-to-Noise Ratio, and Effects of Normalization in Validation of Biomarkers from NMR-based Metabonomics Studies". In: *Chemometrics and intelligent laboratory systems : an international journal sponsored by the Chemometrics Society* 128 (2013), pp. 9–16. ISSN: 0169-7439 (cit. on p. 16).

[46] Mark D Robinson and Alicia Oshlack. "A scaling normalization method for differential expression analysis of RNA-seq data". In: *Genome Biology* 11.3 (2010), R25. ISSN: 1474-760X (cit. on pp. 17, 119).

[47] Joseph K Pickrell et al. "Understanding mechanisms underlying human gene expression variation with RNA sequencing". In: *Nature* 464.7289 (2010), pp. 768–772. ISSN: 1476-4687 (cit. on p. 17).

[48] Davide Risso et al. "GC-Content Normalization for RNA-Seq Data". In: *Bioinformatics* 12.1 (2011), p. 480. ISSN: 1471-2105 (cit. on p. 17).

[49] Ali Mortazavi et al. “Mapping and quantifying mammalian transcriptomes by RNA-Seq”. In: *Nature Methods* 5.7 (2008), pp. 621–628. ISSN: 1548-7105 (cit. on pp. 18, 115).

[50] Alicia Oshlack and Matthew J Wakefield. “Transcript length bias in RNA-seq data confounds systems biology”. In: *Biology Direct* 4.1 (2009), p. 14. ISSN: 1745-6150 (cit. on p. 18).

[51] Sonia Tarazona et al. “Data quality aware analysis of differential expression in RNA-seq with NOISeq R/Bioc package”. In: *Nucleic Acids Research* 43.21 (2015), e140–e140. ISSN: 1362-4962 (cit. on pp. 18, 21, 59, 61, 118, 129, 130, 133).

[52] Peter Kupfer et al. “Batch correction of microarray data substantially improves the identification of genes differentially expressed in Rheumatoid Arthritis and Osteoarthritis”. In: *Medical Genomics* 5 (2012). ISSN: 17558794 (cit. on pp. 19, 56).

[53] Matthew E Ritchie et al. “{limma} powers differential expression analyses for {RNA}-sequencing and microarray studies”. In: *Nucleic Acids Research* 43.7 (2015), e47 (cit. on pp. 19, 21, 56, 60, 61, 66, 67, 80, 85, 122, 135, 154, 196).

[54] Maria J. Nueda, Alberto Ferrer, and Ana Conesa. “ARSyN: A method for the identification and removal of systematic noise in multifactorial time course microarray experiments”. In: *Biostatistics* 13.3 (2012), pp. 553–566. ISSN: 14654644 (cit. on pp. 19, 57, 66, 68, 95, 121, 130, 196).

[55] Age K Smilde et al. “ANOVA-simultaneous component analysis (ASCA): a new tool for analyzing designed metabolomics data.” In: *Bioinformatics* 21.13 (2005), pp. 3043–3048. ISSN: 1367-4803 (cit. on p. 19).

[56] Michael I Love, Wolfgang Huber, and Simon Anders. “Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2”. In: *Genome Biology* 15.12 (2014), p. 550. ISSN: 1474-760X (cit. on p. 20).

[57] Mark D Robinson and Gordon K Smyth. “Moderated statistical tests for assessing differences in tag abundance.” In: *Bioinformatics* 23.21 (2007), pp. 2881–2887. ISSN: 1367-4811 (cit. on p. 20).

[58] Charity W Law et al. “voom: precision weights unlock linear model analysis tools for RNA-seq read counts”. In: *Genome Biology* 15.2 (2014), R29. ISSN: 1474-760X (cit. on pp. 21, 124).

[59] Ana Conesa and Maria Jose Nueda. *maSigPro: Significant Gene Expression Profile Differences in Time Course Gene Expression Data*. R package version 1.60.0. 2020 (cit. on pp. 21, 124, 137).

[60] Charlotte Soneson and Mauro Delorenzi. “A comparison of methods for differential expression analysis of RNA-seq data”. In: *Bioinformatics* 14.1 (2013), p. 91. ISSN: 1471-2105 (cit. on pp. 21, 122).

[61] Ronald A. Fisher. *Statistical Methods for Research Workers*. Edinburgh: Oliver and Boyd, 1925 (cit. on pp. 22, 128, 181).

[62] Aravind Subramanian et al. "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences* 102.43 (2005), 15545 LP –15550 (cit. on pp. 23, 24).

[63] Sonja Hänzelmann, Robert Castelo, and Justin Guinney. "GSVA: gene set variation analysis for microarray and RNA-Seq data". In: *Bioinformatics* 14.1 (2013), p. 7. ISSN: 1471-2105 (cit. on pp. 24, 178).

[64] Rachel Cavill et al. "Transcriptomic and metabolomic data integration". In: *Briefings in Bioinformatics* 17.5 (2016), pp. 891–901. ISSN: 1477-4054 (cit. on p. 25).

[65] Leandro Balzano-Nogueira et al. "Integrative analyses of TEDDY Omics data reveal lipid metabolism abnormalities, increased intracellular ROS and heightened inflammation prior to autoimmunity for type 1 diabetes". In: *Genome Biology* 22.1 (2021), p. 39. ISSN: 1474-760X (cit. on pp. 25, 43).

[66] Brian J. Schmidt et al. "GIM3E: Condition-specific models of cellular metabolism developed from metabolomics and expression data". In: *Bioinformatics* 29.22 (2013), pp. 2900–2908. ISSN: 13674803 (cit. on pp. 25, 37, 39, 40, 151, 162).

[67] Amit Rai, Kazuki Saito, and Mami Yamazaki. "Integrated omics analysis of specialized metabolism in medicinal plants". In: *The Plant Journal* 90.4 (2017), pp. 764–787. ISSN: 0960-7412 (cit. on p. 26).

[68] Ili Nadhirah Jamil et al. “Systematic Multi-Omics Integration (MOI) Approach in Plant Systems Biology”. In: *Frontiers in Plant Science* 11 (2020). ISSN: 1664-462X (cit. on pp. 26, 33).

[69] Joseph Lee Rodgers and W Alan Nicewander. “Thirteen Ways to Look at the Correlation Coefficient”. In: *The American Statistician* 42.1 (1988), pp. 59–66. ISSN: 0003-1305 (cit. on p. 26).

[70] Sonia Tarazona, Leandro Balzano-Nogueira, and Ana Conesa. “Chapter Eighteen - Multiomics Data Integration in Time Series Experiments”. In: *Data Analysis for Omic Sciences: Methods and Applications*. Vol. 82. Elsevier, 2018, pp. 505–532. ISBN: 0166-526X (cit. on p. 26).

[71] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320. ISSN: 1369-7412 (cit. on p. 27).

[72] Robert Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 00359246 (cit. on p. 27).

[73] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. “The group lasso for logistic regression”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.1 (2008), pp. 53–71. ISSN: 1369-7412 (cit. on p. 27).

[74] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal component analysis”. In: *Chemometrics and Intelligent Laboratory Systems* 2.1 (1987), pp. 37–52. ISSN: 0169-7439 (cit. on p. 28).

[75] Svante Wold, Michael Sjöström, and Lennart Eriksson. “PLS-regression: a basic tool of chemometrics”. In: *Chemometrics and Intelligent Laboratory Systems* 58.2 (2001). PLS Methods, pp. 109–130. ISSN: 0169-7439 (cit. on pp. 29, 43, 73).

[76] Johan A. Westerhuis, Theodora Kourti, and John F. Macgregor. “Analysis of multiblock and hierarchical PCA and PLS models”. In: *Journal of Chemometrics* 12.5 (1998), pp. 301–321. ISSN: 08869383 (cit. on p. 32).

[77] Yun Xu and Royston Goodacre. “Multiblock principal component analysis: an efficient tool for analyzing metabolomics data which contain two influential factors”. In: *Metabolomics* 8.1 (2012), pp. 37–51. ISSN: 1573-3890 (cit. on p. 32).

[78] Hui Zou, Trevor Hastie, and Robert Tibshirani. “Sparse principal component analysis”. In: *Journal of Computational and Graphical Statistics* 15.2 (2006), pp. 265–286. ISSN: 10618600 (cit. on p. 32).

[79] Kim-anh Lê Cao et al. “Sparse PLS : Variable Selection when Integrating Omics data”. In: *Statistical Applications in Genetics and Molecular Biology* 7.1 (2008), p. 1390. ISSN: 1544-6115 (cit. on p. 32).

[80] Matthew Barker and William Rayens. “Partial least squares for discrimination”. In: *Journal of Chemometrics* 17.3 (2003), pp. 166–173. ISSN: 0886-9383 (cit. on p. 32).

[81] Kim-Anh Le Cao et al. *mixOmics: Omics Data Integration Project*. 2017 (cit. on pp. 32, 128).

[82] Johan Trygg and Svante Wold. “O2-PLS, a two-block (XY) latent variable regression (LVR) method with an integral OSC filter”. In: *Journal of Chemometrics* 17.1 (2003), pp. 53–64. ISSN: 0886-9383 (cit. on p. 32).

[83] Martijn Schouteden et al. “Performing DISCO-SCA to search for distinctive and common information in linked data”. In: *Behavior Research Methods* 46.2 (2014), pp. 576–587. ISSN: 1554-3528 (cit. on p. 32).

[84] Eric F Lock et al. “JOINT AND INDIVIDUAL VARIATION EXPLAINED (JIVE) FOR INTEGRATED ANALYSIS OF MULTIPLE DATA TYPES.” In: *The annals of applied statistics* 7.1 (2013), pp. 523–542. ISSN: 1932-6157 (cit. on p. 32).

[85] Bro Rasmus. “Multiway calibration. Multilinear PLS”. In: *Journal of Chemometrics* 10.1 (1996), pp. 47–61 (cit. on p. 32).

[86] Ledyard R Tucker. “Some mathematical notes on three-mode factor analysis”. In: *Psychometrika* 31.3 (1966), pp. 279–311. ISSN: 1860-0980 (cit. on p. 32).

[87] Ana Conesa et al. "A multiway approach to data integration in systems biology based on Tucker3 and N-PLS". In: *Chemometrics and Intelligent Laboratory Systems* 104.1 (2010), pp. 101–111. ISSN: 01697439 (cit. on p. 32).

[88] HERMAN WOLD. "Chapter eleven - Path Models with Latent Variables: The NIPALS Approach. NIPALS = Nonlinear Iterative PArtial Least Squares." In: *Quantitative Sociology*. Ed. by H.M. Blalock et al. International Perspectives on Mathematical and Statistical Modeling. Academic Press, 1975, pp. 307–357. ISBN: 978-0-12-103950-9 (cit. on p. 33).

[89] Farhana R Pinu et al. *Systems Biology and Multi-Omics Integration: Viewpoints from the Metabolomics Research Community*. 2019 (cit. on p. 33).

[90] Rafael Hernández-de Diego et al. "PaintOmics 3: a web resource for the pathway analysis and visualization of multi-omics data." In: *Nucleic Acids Research* 46.W1 (2018), W503–W509. ISSN: 1362-4962 (cit. on pp. 33, 34).

[91] Rafael Hernández de Diego. "Development of bioinformatics resources for the integrative analysis of Next Generation omics data". Universitat Politècnica de València, 2017 (cit. on p. 34).

[92] Igor Bartolomé Marín de Mas. "Development and application of novel model-driven and data-driven approaches to study metabolism in the framework of systems medicine". In: *TDX (Tesis Doctorals en Xarxa)* (2015) (cit. on pp. 34, 35).

[93] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. "What is flux balance analysis?" In: *Nature Biotechnology* 28.3 (2010), pp. 245–248. ISSN: 1546-1696 (cit. on pp. 35, 150).

[94] Nathan D Price, Jan Schellenberger, and Bernhard O Palsson. "Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies." In: *Biophysical journal* 87.4 (2004), pp. 2172–2186. ISSN: 0006-3495 (cit. on p. 35).

[95] Scott A Becker and Bernhard O Palsson. "Context-specific metabolic networks are consistent with experiments." In: *PloS Computational Biology* 4.5 (2008), e1000082. ISSN: 1553-7358 (cit. on pp. 37, 44, 150).

[96] Hadas Zur, Eytan Ruppin, and Tomer Shlomi. "iMAT: an integrative metabolic analysis tool." In: *Bioinformatics* 26.24 (2010), pp. 3140–3142. ISSN: 1367-4811 (cit. on pp. 37, 44, 150).

[97] Emanuel Gonçalves et al. "Optimization approaches for the in silico discovery of optimal targets for gene over/underexpression." In: *Journal of Computational Biology : a journal of computational molecular cell biology* 19.2 (2012), pp. 102–114. ISSN: 1557-8666 (cit. on p. 38).

[98] Paul A Jensen and Jason A Papin. "Functional integration of a metabolic network model and expression data without arbitrary thresholding." In: *Bioinformatics* 27.4 (2011), pp. 541–

547. ISSN: 1367-4811 (cit. on pp. 38, 44, 150, 159, 169, 173, 197).

[99] R M T Fleming, I Thiele, and H P Nasheuer. "Quantitative assignment of reaction directionality in constraint-based models of metabolism: application to Escherichia coli." In: *Biophysical chemistry* 145.2-3 (2009), pp. 47–56. ISSN: 1873-4200 (cit. on pp. 39, 151).

[100] Keren Yizhak et al. "Integrating quantitative proteomics and metabolomics with a genome-scale metabolic network model." In: *Bioinformatics* 26.12 (2010), pp. i255–60. ISSN: 1367-4811 (cit. on pp. 39, 151).

[101] Michael A Reid, Ziwei Dai, and Jason W Locasale. "The impact of cellular metabolism on chromatin dynamics and epigenetics." In: *Nature Cell Biology* 19.11 (2017), pp. 1298–1306. ISSN: 1476-4679 (cit. on pp. 40, 151).

[102] Tengda Huang et al. "Integrated Transcriptomic and Translatomic Inquiry of the Role of Betaine on Lipid Metabolic Dysregulation Induced by a High-Fat Diet". In: *Frontiers in Nutrition* 8 (2021), p. 763. ISSN: 2296-861X (cit. on p. 40).

[103] Víctor Sánchez-Gaya et al. "Elucidating the Role of Chromatin State and Transcription Factors on the Regulation of the Yeast Metabolic Cycle: A Multi-Omic Integrative Approach". In: *Frontiers in Genetics* 9 (2018), p. 578. ISSN: 1664-8021 (cit. on pp. 40, 49, 151).

[104] Nora Yucel et al. "Glucose Metabolism Drives Histone Acetylation Landscape Transitions that Dictate Muscle Stem Cell Function". In: *Cell Reports* 27.13 (2019), 3939–3955.e6. ISSN: 2211-1247 (cit. on p. 40).

[105] Tamaki Suganuma and Jerry L Workman. "Chromatin and Metabolism". In: *Annual Review of Biochemistry* 87.1 (2018), pp. 27–49. ISSN: 0066-4154 (cit. on pp. 40, 151).

[106] Manuel Martín-Expósito et al. "Mip6 binds directly to the Mex67 UBA domain to maintain low levels of Msn2/4 stress-dependent mRNAs". In: *EMBO reports* 20.12 (2019), e47964. ISSN: 1469-221X (cit. on pp. 40, 114, 153, 182–185).

[107] Chen Meng et al. "Dimension reduction techniques for the integrative analysis of multi-omics data". In: *Briefings in Bioinformatics* 17.4 (2016), pp. 628–641. ISSN: 1467-5463 (cit. on p. 43).

[108] Florian Rohart et al. "mixOmics: An R package for omics feature selection and multiple data integration". In: *PloS Computational Biology* 13.11 (2017), e1005752 (cit. on pp. 43, 128).

[109] Anne-Laure Boulesteix et al. "IPF-LASSO: Integrative-Penalized Regression with Penalty Factors for Prediction Based on Multi-Omics Data". In: *Computational and Mathematical Methods in Medicine* 2017 (2017), p. 7691937. ISSN: 1748-670X (cit. on p. 43).

[110] Teresa Rubio et al. “Multi-omic analysis unveils biological pathways in peripheral immune system associated to minimal hepatic encephalopathy appearance in cirrhotic patients”. In: *Scientific Reports* 11.1 (2021), p. 1907. ISSN: 2045-2322 (cit. on p. 43).

[111] Manuel Ugidos et al. “MultiBaC: A strategy to remove batch effects between different omic data types.” In: *Statistical Methods in Medical Research* 29.10 (2020), pp. 2851–2864. ISSN: 1477-0334 (cit. on p. 48).

[112] Manuel Ugidos et al. “MAMBA: a model-driven, constraint-based multiomic integration method”. In: *bioRxiv* (Jan. 2022), p. 2022.10.09.511458 (cit. on pp. 48, 62).

[113] Carme Nuño-Cabanes et al. “A multi-omics dataset of heat-shock response in the yeast RNA binding protein Mip6”. In: *Scientific Data* 7.1 (2020), p. 69. ISSN: 2052-4463 (cit. on pp. 48, 129, 132, 153, 190, 191).

[114] Josep Gregori et al. “Batch effects correction improves the sensitivity of significance tests in spectral counting-based comparative discovery proteomics”. In: *Journal of Proteomics* 75.13 (2012), pp. 3938–3951. ISSN: 1874-3919 (cit. on p. 56).

[115] Yuqing Zhang et al. *Alternative empirical Bayes models for adjusting for batch effects in genomic studies*. 2018 (cit. on p. 56).

[116] Jeffrey T Leek et al. *sva: Surrogate Variable Analysis*. 2016 (cit. on pp. 56, 66, 68, 196).

[117] Johann A Gagnon-Bartsch and Terence P Speed. "Using control genes to correct for unwanted variation in microarray data". In: *Biostatistics (Oxford, England)* 13.3 (2012), pp. 539–552. ISSN: 1468-4357 (cit. on p. 56).

[118] Jeroen J Jansen et al. "ASCA: analysis of multivariate data obtained from an experimental design". In: *Journal of Chemometrics* 19.9 (2005), pp. 469–481. ISSN: 0886-9383 (cit. on p. 57).

[119] María José Nueda et al. "Discovering gene expression patterns in time course microarray experiments by ANOVA-SCA." In: *Bioinformatics* 23.14 (2007), pp. 1792–1800. ISSN: 1367-4811 (cit. on p. 57).

[120] Marco Giordan. "A Two-Stage Procedure for the Removal of Batch Effects in Microarray Studies". In: *Statistics in Biosciences* 6.1 (2014), pp. 73–84. ISSN: 1867-1772 (cit. on p. 57).

[121] Gift Nyamundanda et al. "A Novel Statistical Method to Diagnose, Quantify and Correct Batch Effects in Genomic Studies". In: *Scientific Reports* 7.1 (2017), p. 10849. ISSN: 2045-2322 (cit. on p. 57).

[122] Sarah E Reese et al. "A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis". In: *Bioinformatics* 29.22 (2013), pp. 2877–2883. ISSN: 1367-4803 (cit. on p. 57).

[123] Anna Papiez et al. "BatchI: Batch effect Identification in high-throughput screening data using a dynamic programming algo-

rithm". In: *Bioinformatics* 35.11 (2019), pp. 1885–1892. ISSN: 1367-4803 (cit. on p. 57).

[124] Brittney N. Keel et al. "RNA-Seq Meta-analysis identifies genes in skeletal muscle associated with gain and intake across a multi-season study of crossbred beef steers". In: *Genomics* 19.1 (2018), p. 430 (cit. on p. 57).

[125] Matthew D. Li et al. "Integrated multi-cohort transcriptional meta-analysis of neurodegenerative diseases". In: *Acta Neuropathologica Communications* 2.1 (2014), p. 93 (cit. on p. 57).

[126] Marta Andres-Terre et al. "Integrated, Multi-cohort Analysis Identifies Conserved Transcriptional Signatures across Multiple Respiratory Viruses". In: *Immunity* 43.6 (2015), pp. 1199–1211 (cit. on p. 57).

[127] Vandana Sandhu et al. "Meta-Analysis of 1,200 Transcriptomic Profiles Identifies a Prognostic Model for Pancreatic Ductal Adenocarcinoma". In: *JCO Clinical Cancer Informatics* 3 (2019). PMID: 31070984, pp. 1–16 (cit. on p. 57).

[128] Haiyan Huang, Chun-Chi Liu, and Xianghong Jasmine Zhou. "Bayesian approach to transforming public gene expression repositories into disease diagnosis databases". In: *Proceedings of the National Academy of Sciences* 107.15 (2010), pp. 6823–6828. ISSN: 0027-8424 (cit. on p. 57).

[129] Vicent Pelechano and José E. Pérez-Ortín. “There is a steady-state transcriptome in exponentially growing yeast cells”. In: *Yeast* 27.7 (2010), pp. 413–422 (cit. on pp. 58, 59).

[130] José Garca-Martnez, Agustn Aranda, and JoséE Pérez-Ortn. “Genomic Run-On Evaluates Transcription Rates for All Yeast Genes and Identifies Gene Regulatory Mechanisms”. In: *Molecular Cell* 15.2 (2004), pp. 303–313 (cit. on pp. 58, 59).

[131] Vicent Pelechano, Sebastián Chávez, and JoséE Pérez-Ortín. “A complete set of nascent transcription rates for yeast genes”. In: *PloS one* 5.11 (2010), e15442; e15442–e15442 (cit. on pp. 58, 59).

[132] Brian M Zid and Erin K O’Shea. “Promoter sequences direct cytoplasmic localization and translation of mRNAs during starvation in yeast”. In: *Nature* 514.7520 (2014), pp. 117–121 (cit. on pp. 58, 59, 95).

[133] Mallory A Freeberg et al. “Pervasive and dynamic protein binding sites of the mRNA transcriptome in Saccharomyces cerevisiae”. In: *Genome Biology* 14.2 (2013), R13–R13 (cit. on pp. 58, 59).

[134] Pedro Furió-Tarí, Ana Conesa, and Sonia Tarazona. “RGmatch: matching genomic regions to proximal genes in omics data integration”. In: *Bioinformatics* 17.15 (2016), p. 427. ISSN: 1471-2105 (cit. on pp. 60, 133).

[135] Anastasia McKinlay, Carlos L Araya, and Stanley Fields. "Genome-Wide Analysis of Nascent Transcription in Saccharomyces cerevisiae". In: *G3 (Bethesda, Md.)* 1.7 (2011), pp. 549–558 (cit. on p. 60).

[136] Laia Castells-Roca et al. "Heat shock response in yeast involves changes in both transcription rates and mRNA stabilities". In: *PloS one* 6.2 (2011), e17272–e17272 (cit. on p. 60).

[137] Daehwan Kim et al. "TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions". In: *Genome Biology* 14.4 (2013), R36. ISSN: 1474-760X (cit. on pp. 61, 129).

[138] Carlos Martínez-Mira, Ana Conesa, and Sonia Tarazona. "MOSim: Multi-Omics Simulation in R". In: *bioRxiv* (2018), p. 421834 (cit. on p. 62).

[139] S S Shapiro and M B Wilk. "An Analysis of Variance Test for Normality (Complete Samples)". In: *Biometrika* 52.3/4 (1965), pp. 591–611. ISSN: 00063444 (cit. on p. 64).

[140] J P Royston. "An Extension of Shapiro and Wilk's W Test for Normality to Large Samples". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 31.2 (1982), pp. 115–124. ISSN: 0035-9254 (cit. on p. 64).

[141] W Evan Johnson, Cheng Li, and Ariel Rabinovic. "Adjusting batch effects in microarray expression data using empirical Bayes

methods". In: *Biostatistics* 8.1 (2007), pp. 118–127. ISSN: 1465-4644 (cit. on pp. 66, 68, 196).

[142] Michel Tenenhaus. *La régression PLS, théorie et practique*. Paris: Editions Technip, 1998 (cit. on p. 74).

[143] Abel Folch-Fortuny, Francisco Arteaga, and Alberto Ferrer. "PLS model building with missing data: New algorithms and a comparative study". In: *Journal of Chemometrics* 31.7 (2017). ISSN: 1099128X (cit. on pp. 77, 196).

[144] Abel Folch-Fortuny et al. "Calibration transfer between NIR spectrometers: New proposals and a comparative study". In: *Journal of Chemometrics* 31.3 (2017), e2874 (cit. on pp. 77, 196).

[145] Salvador García Muñoz, John F. MacGregor, and Theodora Kourti. "Product transfer between sites using Joint-Y PLS". In: *Chemometrics and Intelligent Laboratory Systems* 79.1-2 (2005), pp. 101–114. ISSN: 01697439 (cit. on pp. 77, 196).

[146] Jose Manuel Andrade et al. "Procrustes rotation in analytical chemistry, a tutorial". English. In: *Chemometrics and Intelligent Laboratory Systems* 72.2 (2004), pp. 123–132 (cit. on p. 84).

[147] John R. Hurley and Raymond B. Cattell. "The procrustes program: Producing direct rotation to test a hypothesized factor structure". In: *Behavioral Science* 7.2 (1962), pp. 258–262 (cit. on p. 84).

[148] J. A. Hartigan. *Clustering Algorithms*. 99th. New York, NY, USA: John Wiley & Sons, Inc., 1975. ISBN: 047135645X (cit. on p. 91).

[149] J.A. Hartigan and M.A. Wong. “A K-Means Clustering Algorithm”. In: *Applied Statistics* 28.1 (1979), pp. 100–108 (cit. on p. 91).

[150] Marcel Ramos et al. “Software for the Integration of Multiomics Experiments in Bioconductor.” In: *Cancer Research* 77.21 (2017), e39–e42. ISSN: 1538-7445 (cit. on p. 96).

[151] Sijmen De Jong and Cajo J F Ter Braak. “Comments on the PLS kernel algorithm”. In: *Journal of Chemometrics* 8.2 (1994), pp. 169–174. ISSN: 0886-9383 (cit. on p. 109).

[152] Bhupinder S. Dayal and Jhon F. MacGregor. “Improved PLS algorithms”. In: *Journal of Chemometrics* 11.1 (1997), pp. 73–85 (cit. on p. 109).

[153] Susana Rodríguez-Navarro and Ed Hurt. “Linking gene regulation to mRNA production and export”. In: *Current Opinion in Cell Biology* 23.3 (2011), pp. 302–309. ISSN: 0955-0674 (cit. on p. 113).

[154] Encar García-Oliver, Varinia García-Molinero, and Susana Rodríguez-Navarro. “mRNA export and gene expression: The SAGATREX-2 connection”. In: *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms* 1819.6 (2012), pp. 555–565. ISSN: 1874-9399 (cit. on p. 113).

[155] Tony Kouzarides. "Chromatin Modifications and Their Function". In: *Cell* 128.4 (2007), pp. 693–705. ISSN: 0092-8674 (cit. on p. 113).

[156] Tianyi Zhang, Sarah Cooper, and Neil Brockdorff. "The interplay of histone modifications writers that read". In: *EMBO reports* 16.11 (2015), pp. 1467–1481 (cit. on p. 113).

[157] Minjia Tan et al. "Identification of 67 histone marks and histone lysine crotonylation as a new type of histone modification." In: *Cell* 146.6 (2011), pp. 1016–1028. ISSN: 1097-4172 (cit. on p. 113).

[158] David Gomez-Cabrero et al. "STATegra, a comprehensive multi-omics dataset of B-cell differentiation in mouse". In: *Scientific Data* 6.1 (2019), p. 256. ISSN: 2052-4463 (cit. on p. 114).

[159] Yong Zhang et al. "Model-based Analysis of ChIP-Seq (MACS)". In: *Genome Biology* 9.9 (2008), R137. ISSN: 1474-760X (cit. on p. 114).

[160] Ana Conesa et al. "A survey of best practices for RNA-seq data analysis." In: *Genome Biology* 17 (2016), p. 13. ISSN: 1474-760X (cit. on p. 115).

[161] Lauren M McIntyre et al. "RNA-seq: technical variability and sampling". In: *Genomics* 12.1 (2011), p. 293. ISSN: 1471-2164 (cit. on p. 118).

[162] Tamara Steijger et al. “Assessment of transcript reconstruction methods for RNA-seq”. eng. In: *Nature Methods* 10.12 (2013). ID: unige:42242, pp. 1177–84 (cit. on p. 118).

[163] Gordon K Smyth. “Linear models and empirical bayes methods for assessing differential expression in microarray experiments.” In: *Statistical Applications in Genetics and Molecular Biology* 3 (2004), Article3. ISSN: 1544-6115 (cit. on pp. 122, 124).

[164] Ana Conesa et al. “maSigPro: a method to identify significantly differential expression profiles in time-course microarray experiments”. In: *Bioinformatics* 22.9 (2006), pp. 1096–1102. ISSN: 1367-4803 (cit. on p. 124).

[165] María José Nueda, Sonia Tarazona, and Ana Conesa. “Next maSigPro: updating maSigPro bioconductor package for RNA-seq time series”. In: *Bioinformatics* 30.18 (2014), pp. 2598–2602. ISSN: 1367-4803 (cit. on p. 124).

[166] Yoav Benjamini and Yosef Hochberg. “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1 (1995), pp. 289–300. ISSN: 00359246 (cit. on p. 126).

[167] Fernando García-Alcalde et al. “Paintomics: a web based tool for the joint visualization of transcriptomics and metabolomics data”. In: *Bioinformatics* 27.1 (2011), pp. 137–139. ISSN: 1367-4803 (cit. on p. 127).

[168] Minoru Kanehisa et al. “KEGG as a reference resource for gene and protein annotation.” In: *Nucleic Acids Research* 44.D1 (2016), pp. D457–62. ISSN: 1362-4962 (cit. on p. 127).

[169] Jasmin Fisher and Thomas A Henzinger. “Executable cell biology.” In: *Nature Biotechnology* 25.11 (2007), pp. 1239–1249. ISSN: 1087-0156 (cit. on p. 127).

[170] Marcel Martin. “Cutadapt removes adapter sequences from high-throughput sequencing reads”. In: *EMBnet journal* (2011) (cit. on p. 133).

[171] Ben Langmead and Steven L Salzberg. “Fast gapped-read alignment with Bowtie 2”. In: *Nature Methods* 9.4 (2012), pp. 357–359. ISSN: 1548-7105 (cit. on p. 133).

[172] Yajie Yang et al. “Leveraging biological replicates to improve analysis in ChIP-seq experiments”. In: *Computational and Structural Biotechnology Journal* 9.13 (2014), e201401002. ISSN: 2001-0370 (cit. on p. 133).

[173] Aaron R Quinlan and Ira M Hall. “BEDTools: a flexible suite of utilities for comparing genomic features”. In: *Bioinformatics* 26.6 (2010), pp. 841–842. ISSN: 1367-4803 (cit. on p. 133).

[174] “Applications of genome-scale metabolic reconstructions.” In: *Molecular Systems Biology* 5 (2009), p. 320. ISSN: 1744-4292 (cit. on pp. 149, 151).

[175] Adam M Feist and Bernhard O Palsson. "The biomass objective function." In: *Current Opinion in Microbiology* 13.3 (2010), pp. 344–349. ISSN: 1879-0364 (cit. on p. 149).

[176] Adam M Feist et al. "A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information." In: *Molecular Systems Biology* 3 (2007), p. 121. ISSN: 1744-4292 (cit. on p. 151).

[177] Moxley JF et al. "Linking high-resolution metabolic flux phenotypes and transcriptional regulation in yeast modulated by the global regulator Gcn4p." In: *Proceedings of the National Academy of Sciences of the United States of America* 106.16 (2009), pp. 6477–6482. ISSN: 0027-8424 (cit. on p. 151).

[178] Yinmin Gu et al. "Association Analysis between Body Mass Index and Genomic DNA Methylation across 15 Major Cancer Types." In: *Journal of Cancer* 9.14 (2018), pp. 2532–2542. ISSN: 1837-9664 (cit. on p. 151).

[179] Svetlana Volkova et al. "Metabolic Modelling as a Framework for Metabolomics Data Integration and Analysis". In: *Metabolites* 10.8 (2020), p. 303. ISSN: 2218-1989 (cit. on p. 151).

[180] Ali R Zomorrodi and Costas D Maranas. "Improving the iMM904 S. cerevisiae metabolic model using essentiality and synthetic lethality data". In: *Systems Biology* 4.1 (2010), p. 178. ISSN: 1752-0509 (cit. on p. 153).

[181] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2021 (cit. on pp. 154, 162).

[182] Igor Marín de Mas et al. "Stoichiometric gene-to-reaction associations enhance model-driven analysis performance: Metabolic response to chronic exposure to Aldrin in prostate cancer". In: *Genomics* 20.1 (2019), p. 652. ISSN: 1471-2164 (cit. on p. 157).

[183] J.F. Kenney and E.S. Keeping. *Mathematics of statistics*. New York, USA: New York, Van Nostrand, 1962 (cit. on p. 168).

[184] Kevin A. Morano, Chris M. Grant, and W. Scott Moye-Rowley. "The response to heat shock and oxidative stress in saccharomyces cerevisiae". In: *Genetics* 190.4 (2012), pp. 1157–1195. ISSN: 00166731 (cit. on pp. 173, 189).

[185] Pia Erdbrügger and Florian Fröhlich. "The role of very long chain fatty acids in yeast physiology and human diseases". In: *Biological Chemistry* 402.1 (2021), pp. 25–38 (cit. on p. 181).

[186] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. ISSN: 00994987 (cit. on pp. 185, 188).

[187] H B Mann and D R Whitney. "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other". In: *The Annals of Mathematical Statistics* 18.1 (1947), pp. 50–60 (cit. on pp. 185, 188).

[188] Daqiang Pan, Nils Wiedemann, and Bernd Kammerer. "Heat stress-induced metabolic remodeling in Saccharomyces cerevisiae". In: *Metabolites* 9.11 (2019), pp. 1–12. ISSN: 22181989 (cit. on pp. 189, 191).

[189] M A Singer and S Lindquist. "Multiple effects of trehalose on protein folding in vitro and in vivo." In: *Molecular Cell* 1.5 (1998), pp. 639–648. ISSN: 1097-2765 (cit. on p. 189).

[190] Tomohiro Kaino and Hiroshi Takagi. "Proline as a stress protectant in the yeast Saccharomyces cerevisiae: effects of trehalose and PRO1 gene expression on stress tolerance." In: *Bioscience, biotechnology, and biochemistry* 73.9 (2009), pp. 2131–2135. ISSN: 1347-6947 (cit. on p. 191).