Esta investigação foi financiada pela Fundação para a Ciência e Tecnologia através da concessão de uma bolsa de doutoramento (SFRH/BD/80925/2011), co-financiada pelo POPH – QREN – Tipologia 4.1 – Formação Avançada – e comparticipado pelo Fundo Social Europeu (FSE) e por fundos nacionais do Ministério da Ciência, Tecnologia e Ensino Superior (MCTES).

POPH QUALIFICAR É CRESCER.
QREN QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL PORTUGAL 2007.2013
GOVERNO DA REPÚBLICA PORTUGUESA
UNIÃO EUROPEIA Fundo Social Europeu

A framework for the reconstruction and analysis of tissue specific genome-scale metabolic models

Sara Alexandra Gomes Correia

UMinho | 2016

Universidade do Minho
Escola de Engenharia

Sara Alexandra Gomes Correia

A framework for the reconstruction and analysis of tissue specific genome-scale metabolic models

dezembro de 2016

Universidade do Minho
Escola de Engenharia

Sara Alexandra Gomes Correia

A framework for the reconstruction and analysis of tissue specific genome-scale metabolic models

Tese de Doutoramento
Doutoramento em Informática

Trabalho efetuado sob a orientação do:
Doutor Miguel Francisco de Almeida Pereira da Rocha
e do
Doutor Bruno Filipe Marques Costa

dezembro de 2016

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração da presente tese. Confirmo que em todo o trabalho conducente à sua elaboração não recorri à prática de plágio ou qualquer forma de falsificação de resultados.

Mais declaro que tomei conhecimento integral do Código de Conduta Ética da Universidade do Minho.

Universidade do Minho, dezembro de 2016.

Nome Completo: Sara Alexandra Gomes Correia

Sara Alexandra Gomes Correia

Acknowledgements / Agradecimentos

Finalmente, mais um capítulo que se concretiza na minha vida. Ao longo deste percurso encontrei ajuda e apoio em várias pessoas, que de uma forma ou outra contribuiram para a concretização deste projecto, às quais agradeço do fundo do coração. Este é um espaço dedicado inteiramente a vós.

Em primeiro lugar, agradeço aos meus orientadores, Doutor Miguel Rocha e Doutor Bruno Costa, por todo o apoio, orientação e cooperação ao longo deste trabalho. Ao Doutor Miguel Rocha, um obrigada muito especial pelo desafio lançado há cinco anos atrás. Sem o qual não me encontraria, hoje, a trabalhar nesta área tão fascinante e onde me sinto realizada.

Ao Centro de Engenharia Biológica e ao Centro de Ciência e Tecnologias da Computação da Universidade do Minho por terem aceitado serem, em diferentes fases, a minha instituição de acolhimento. À Fundação para a Ciência e Tecnologia pela atribuição da bolsa de doutoramento que tornou este trabalho possível (SFRH/BD/80925/2011).

A todos os colegas do BIOSYSTEMS, grupo de investigação no qual estive inserida durante estes quatro anos. Sem os momentos de descontração passados convosco, teria sido muito mais difícil suportar esta jornada.

Aos colegas do grupo de investição "Systems Biomedicine - EBI-UK", em especial ao Emanuel, que me acolheram em Cambridge durante 3 meses e tornaram a experiência de viver fora do país tão especial.

À Carla, à Cristiana e ao Paulo Vilaça pela vossa presença, ajuda, paciência, conselhos e repreensões, que vieram sempre nos momentos certos. Recordo, com especial carinho os “puxões de orelhas” da Carlinha. Mesmo quando a distância foi muita, nunca vos mantivestes realmente longe (excepto quando a videoconferência acaba porque alguém se esquece do carregador). Obrigada por tudo.

Um obrigada muito especial à Sónia e Daniel Machado pelas discussões interessantes e pela partilha de conhecimento, além das gargalhadas e da boa disposição.

À Ana Alão, pelas “águas gazificadas com sabores” no acolhedor Pão de Forma, pelas conversas que entraram pela madrugada dentro e por nunca me deixares “Alone”. Ao Rafael Pereira, o meu paciente vizinho da frente na sala de trabalho, obrigada pela boa disposição e simpatia. Aos companheiros do café, com boas conversas a acompanhar, Daniel Gomes e João Marcos, obrigada também a vós.

Agradeço aos meus amigos, em especial à Tânia e à Marisa pelo apoio e motivação. À minha “irmã” Sandrina, que sempre acreditou em mim, mesmo não sabendo do que eu andei por aqui a fazer.

Um obrigada muito especial a toda a minha família, que sempre me apoiou em todos os momemtos.

Finalmente, serei eternamente grata aos três homens da minha vida. O meu pai, que desde sempre acreditou em mim. O meu melhor amigo, companheiro e marido, Filipe, por ser o meu pilar. E por último, o Gabriel, meu filho, por ser o meu azimute.

Abstract

In recent years, the development of novel techniques for genome sequencing and other high-throughput methods has enabled the identification and quantification of individual cell components. Genome-scale metabolic models (GSMMs) have been developed for several organisms, including humans. Under the framework of constraint-based modeling, these have provided phenotype prediction methods, useful in fields as metabolic engineering and biomedical research, spanning tasks as drug discovery, biomarker identification and host-pathogen interactions, and targeting diseases such as cancer, Alzheimer, or diabetes.

However, these methods have been limited, since the human body has a diversity of cell types and tissues making the development of specific models an imperative. Methods to provide phenotype simulation with the integration of omics data and to automatically generate tissue-specific models, based on generic human metabolic models and a plethora of omics data, have been proposed. However, their results have not been adequately and critically evaluated and compared. Moreover, their usage is restricted to users with computer science skills, since they are not available in user-friendly software platforms.

In this work, an open-source software framework for the integration of GSMMs with omics data has been provided. It contains methods for the processing and integration of data with models, for the reconstruction of tissue-specific GSMMs and for phenotype simulation using omics data. A user-friendly graphical interface is provided for non-programming users to be able to run these methods, while an open programming interface allows the community to contribute.

The methods have also been validated and compared in representative case studies, being studied the effects of data sources and algorithms in the final results. In particular, glioblastoma has been selected as a more comprehensive case study, where specific models were generated for a representative cell line using different approaches. These have been compared and integrated into a consensus model, which has been further used for analysis and to support phenotype simulation. The results allow insights into cancer metabolism and possible routes towards drug discovery.

Resumo

Nos últimos anos, o desenvolvimento de novas técnicas de sequenciação genómica e outros métodos experimentais de alto débito têm permitido a identificação e quantificação de componentes celulares. Um conjunto de Modelos Metabolicos à Escala Genomica (MMEG) têm sido desenvolvidos para múltiplos organismos, incluindo os seres humanos. Recorrendo à modelação com base em restrições, estes têm fornecido métodos de predição do fenótipo, que têm sido úteis na área da engenharia metabolica e investigação biomédica, abordando tarefas como a descoberta de farmacos, a identificação de biomarcadores e a interação entre agentes patogénicos e hospedeiros, e doenças como o cancro, Alzheimer ou diabetes.

Contudo, estes métodos têm a sua aplicação limitada, dado que o corpo humano é constituído por diversos tecidos e tipos de células, tornando essencial o desenvolvimento de modelos especificos. Neste contexto, têm surgido métodos que permitem a simulação do fenótipo com integração de dados omicos, assim como a reconstrução de modelos específicos baseados num modelo genérico e em conjuntos de dados omicos. Todavia, os seus resultados não foram ainda comparados e avaliados sistematicamente. Além disso, a sua utilização está restrita a utilizadores com competências computacionais, uma vez que não existe nenhuma plataforma de software de fácil utilização.

Neste trabalho, foi desenvolvida uma plataforma de software de acesso livre, que permite a integração de MMEGs com dados omicos. Esta plataforma contém métodos para o precessamento e integração dos dados com os modelos, reconstrução de MMEG para tecidos específicos e simulação do fénotipo utilizando dados omicos. Foi desenvolvida uma interface gráfica que permite a utilização destes métodos por não programadores. A comunidade pode ainda contribuir para a sua extensão através da interface disponibilizada.

Os métodos foram validados e comparados com outros estudos, sendo analisados os efeitos que as fontes de dados e os algoritmos têm nos resultados finais. Em particular, foi selecionado como caso de estudo mais abrangente a reconstrução do modelos metabolicos, usando diferentes abordagens, para uma linha celular do glioblastoma. Posteriormente, estes modelos foram comparados e integrados num modelo consenso, que foi utilizado para análise e simulação de fenótipos. Os resultados obtidos permitem aprofundar o conhecimento do metabolismo do cancro e apontam possíveis caminhos para a descoberta de novos fármacos.

Acronyms

API	Application Programming Interface
ATP	Adenosine Triphosphate
BRENDA	Braunschweig Enzyme Database
CBM	Constraint-Based Modelling
ChEBI	Chemical Entities of Biological Interest
DNA	Deoxyribonucleic acid
EHMM	Edinburgh Human Metabolic Model
FBA	Flux Balance Analysis
FVA	Flux Variance Analysis
GC-MS	Gas Chromatography-Mass Spectrometry
GEB	Gene Expression Barcode
GEO	Gene Expression Omnibus
GIMME	Gene Inactivity Moderated by Metabolism and Expression
GPR	Gene-Protein-Reaction
GSMM	Genome-Scale Metabolic Model
GUI	Graphical User Interface
HCC	hepatocellular carcinoma
HGNC	HUGO Gene Nomenclature Committee
HMDB	Human Metabolome Database
HMDB	Human Metabolome Database
HMR	Human Metabolic Reaction

HPA	Human Protein Atlas
HPRD	Human Protein Reference Database
iMAT	Integrative Metabolic Analysis Tool
INIT	Integrative Network Inference for Tissues
IS	Inconsistency Score
KEGG	Kyoto Encyclopedia of Genes and Genomes
LP	Linear Programming
MBA	Model-Building Algorithm
mCADRE	Metabolic Context specificity Assessed by Deterministic Reaction Evaluation
MEW	Metabolic Engineering Workbench
MEW	Metabolic Engineering Workbenc
MILP	Mixed-Integer Linear Programming
MiMBl	Minimization of Metabolites Balance
MOMA	Minimization of Metabolic Adjustment
MVC	Model-View-Controller
NAD	Nicotinamide Adenine Dinucleotide
NMR	Nuclear Magnetic Resonance
pFBA	parsimonious enzyme usage FBA
PRIME	Personalized Reconstruction of Metabolic models
RMF	Required Metabolic Functionalities
RNA	Ribonucleic acid
ROOM	Regulatory on/off minimization of metabolic flux changes
SB	Systems Biology
SBML	Systems Biology Markup Language
TCGA	The Cancer Genome Atlas
tINIT	Task-driven Integrative Network Inference for Tissues

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In this brief introduction, the contextualization of this work and the main objectives to pursue during this thesis are presented. Also, a general overview of the next chapters is provided.

1.1 Context and motivation

The mathematical modelling of cells has been traditionally achieved through the use of dynamic models. However, since these require kinetic information typically not available, their applicability is limited to small-scale systems [1]. As an alternative, recent efforts allowed the development of genome-scale metabolic models for several organisms (including humans). These have been used to predict cellular metabolism under some simplifying assumptions, namely considering the cell to be in steady-state, i.e. the concentrations of all intracellular compounds are assumed to remain constant throughout time. Together with the known stoichiometry and reversibility of the reactions, this is used, in a constraint-based framework, to determine the possible values for the reaction fluxes. Therefore, cellular behaviour can be predicted using methods such as Flux Balance Analysis (FBA) [2]. Stoichiometric models and simulation methods have been thoroughly used in Metabolic Engineering [3], but also in other applications related to biological discovery and data analysis [4, 5, 6].

Recently, this effort has been extended with the development of four independent human metabolic models [7, 8, 9, 10]. These models aim to represent the metabolism of the most complex multicellular organisms, including a set of biochemical reactions that may occur in distinct tissues or cell types. Over the last years, they have already shown to be useful in biomedical applications, such as in selecting drug targets for hypercholesterolemia [7], in predicting metabolic markers for inborn errors of metabolism [11], and in the study of the Warburg effect in cancer cells [12].

Despite the recent advances in the understanding of human metabolism provided by these models, it is undeniable that the usefulness of these models depends on the capability to address the phenotype simulation of different cell types. This challenge was firstly addressed in [13], where the generic model from [7] is integrated with gene expression and proteomics data to predict the metabolic behaviour of human tissues, such as the liver or the kidney. In the following years, several approaches [14, 15] have been proposed for phenotype simulation with the integration of omics data to improve the prediction quality. However, these methods only allow to characterize the normal physiological behaviour of a cell type and can not be used to simulate the effects of genetic or environmental perturbations, a feature essential for biomedical research.

Therefore, there is a need for reconstructing tissue-specific metabolic models that can be used to simulate the phenotype of distinct cell types in several conditions. In 2010, a model-building algorithm [16] was proposed to tackle this task, taking as its basis a generic model and heuristically pruning it to derive a sub-model that is as consistent as possible with available experimental data. This algorithm was used to construct a model for liver cell, as a validation case study. A different approach, relying more on manual curation, has been followed by Gille and co-workers [17] with the same final result, a liver cell model, but showing more accurate predictions.

The previous approach has been applied to the reconstruction of the models of distinct types of human neurons [18], creating models of brain energy metabolism relevant to the study of Alzheimer's disease, and also in studying the metabolic changes and the host-pathogen interactions occurring during tuberculosis [19].

In spite of these results, the heuristic nature and limited accuracy of the method from [16], together with the results from [17], show that there is the need for more consistent methods for the (semi)-automatic reconstruction of tissue-specific metabolic models, a task that will be targeted in this work.

Additional approaches have been proposed in the following years [9, 20, 21, 22, 23]. However, the comparison of the results is not trivial since each approach uses specific data types as source data. Furthermore, each method is evaluated with specific case studies and data sets in their own publications.

In this work, we propose the development of an integrated framework for the reconstruction of tissue-specific metabolic models and phenotype simulation integrating omics data. Moreover, a graphical interface will be provided for the non-programmers users to be able to run these methods. In the end, we will use this framework to reconstruct a genome-scale metabolic model for one of the most aggressive brain cancers - the glioblastoma.

1.2 Research aims

In this context, the aim of this work will be to develop and systematically evaluate methods and computational tools that allow the reconstruction of genome-scale metabolic models for specific cell types/ tissues and their application in biomedical research. Moreover, phenotype simulation with omics integration methods will also be implemented. We will develop an integrated computational platform that can be used by researchers to build and validate models using distinct data and use those in different case studies. As a case study, a glioblastoma metabolic model will be reconstructed using the methods presented in the developed framework.

This work will, therefore, encompass the following scientific/ technological objectives:

- To devise a computational framework, including tools to load and transform omics data, as well as to integrate them with metabolic models. The data can be specific knowledge on metabolic systems collected from literature (manually) or experimental data from relevant phenotypes – gene expression, proteomics, metabolomics, fluxomics. Several file formats must be supported, such as the Human Protein Atlas files, XML files from the Human Metabolome Database and generic text files (using comma/tab separators).

- To develop computational tools that will allow the reconstruction of genome-scale metabolic models for specific cell types/ tissues and the phenotype prediction using omics data to improve the results. These algorithms will be supported by the infrastructure from the previous step.

- To implement the methods from the previous steps within the context of OptFlux [24], a metabolic engineering reference platform developed within the group.

- To systematically evaluate and compare the previous methods using different omics data as input, with the purpose of finding the best combination of method and omics data to be used in other case studies.

- To reconstruct genome-scale metabolic models for cells with the glioblastoma phenotype, providing their comparison and analysis aiming to uncover insights regarding their metabolism and possible drug discovery efforts.

1.3 Thesis outline

This manuscript has been structured in seven chapters addressing all of the previously stated aims.

The thesis begins in the current chapter (Chapter 1) with a general introduction, together with the statement of the proposed aims and an outline of the manuscript's structure.

Chapter 2 presents a thorough report of the state of the art of the set of subjects involved in this project, namely: metabolic model reconstruction, constraint based modelling of metabolic systems, phenotype simulation methods, context-specific model reconstruction approaches and applications of such methods.

In Chapter 3, the software tools developed during this thesis are explained in detail. These tools are made available in a powerful, yet accessible framework, for the community to use and extend.

Chapter 4 presents the evaluation of the phenotype simulation methods using omics data to improve the predictions.

Chapter 5 presents a critical evaluation of methods for the reconstruction of tissue-specific metabolic models and the consistency between several omics data sources.

The reconstruction of the glioblastoma metabolic models is presented in chapter 6. Here, we detail the reconstruction process, validate and compare the models with other published ones, and use it to gain insight on cancer cell metabolism.

Finally, Chapter 7 presents the general conclusions derived from this work and perspectives for future work.

Chapter 2

State of the Art

This chapter presents the concepts related with systems biology, constraint-based modeling and omics data. The reconstruction process of genome-scale metabolic models is explained. Besides, a summary of the most important methods for the tissue-specific reconstruction models and phenotype simulation integrating omics data are presented.

2.1 Systems biology

Nature is composed of several different species that crossed biological evolution along the years. Each individual is composed of elemental building blocks of life - the cells [25]. In the last decades, deep knowledge about individual cellular components and their functions provided by biological research have clearly shown that most biological processes occur in complex interactions between cellular constituents, such as proteins, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) molecules [26].

In this context, Systems Biology (SB) arises as an interdisciplinary field of study that tries to explain the complex interactions within biological systems [27]. The evolution of SB has been supported by the development of novel techniques for genome sequencing and other high-throughput methods, that have generated the so called ”omics” data, such as genomics [28], transcriptomics [29], proteomics [30], metabolomics [31], and fluxomics [32].

The combination of these data and the knowledge of cellular functions allowed the construction of biological networks or of models capable to simulate the cell behaviour [33].

Biological networks/models can be broadly categorized into three types [34] :

1. *metabolic*: contains all the biochemical reactions that occur in the cell. These networks describe the consumption/synthesis of metabolites that are essential for the growth and cell survival;

2. *regulatory*: aims to represent the regulatory interactions between regulatory elements (e.g. transcription factors, promoters) and their target genes, for instance $A \rightarrow B$ means that gene A controls the expression of gene B;

3. *signalling*: represents the reactions or *”signalling events”* (such as phosphorylation or ubiquitinations) in a network that regulate how a cell responds to its environment, through cascades of information flow.

None of these networks are independent and the combination of the different levels allows for a deeper understanding of cellular processes. However, the integration of all these information into models increases their complexity, while the current modeling capabilities of all these networks prevent their inference in a genome level [6].

Therefore, in the following the focus will rely on metabolic models that are the most developed and the ones addressed in this work.

2.2 Genome-scale metabolic models

Genome-Scale Metabolic Models (GSMMs) are composed by metabolites and reactions that allow the representation of all biochemical processes of the cell. The development of GSMMs starts with genome sequencing [35, 36]. Based on the genome sequence, a functional annotation is performed through the information present in databases such as GenBank [37], Entrez Gene [38] and BioCyc [39]. Next, the set of reactions and the gene-protein-reaction (GPR) associations are collected using information presented in databases such as KEGG [40], BRENDA [41], UniProt [42, 43], MetaCyc [44] and also literature.

GPR associations are composed by logical rules, which represent the relationship between genes, proteins and reactions. This allows to include information about the transcriptional/ translational level, through the reference to the enzymes that catalyse the reactions and the genes encoding those enzymes, into the metabolic models.

Normally, in GSMMs, a GPR association contains only the relationship between genes and a reaction using the logical operators AND or OR to represent the dependency of genes of each reaction. For instance, if reaction r_1 has the GPR $g_1 OR (g_2 AND g_3)$, this means that the reaction r_1 occurs only when gene g_1 or both genes g_2 and g_3 are expressed. The inclusion of GPRs within GSMMs is essential to allow the phenotype prediction of the cell under different genetic conditions, such as gene knockouts and over/underexpression.

Once the draft metabolic model is generated, a set of simulations are required to validate the model. Based on the results, the model may be improved or optimized by the addition/ removal of reactions (Figure 2.1).

Over the last decade, the advances in DNA sequencing techniques and the sequencing costs have decreased allowing to increase the number of organisms having their genome sequenced [45, 46] and, therefore, the number of GSMMs being reconstructed [47].

Nowadays, several tools such as Model SEED [48] or Merlin [49] are avail-

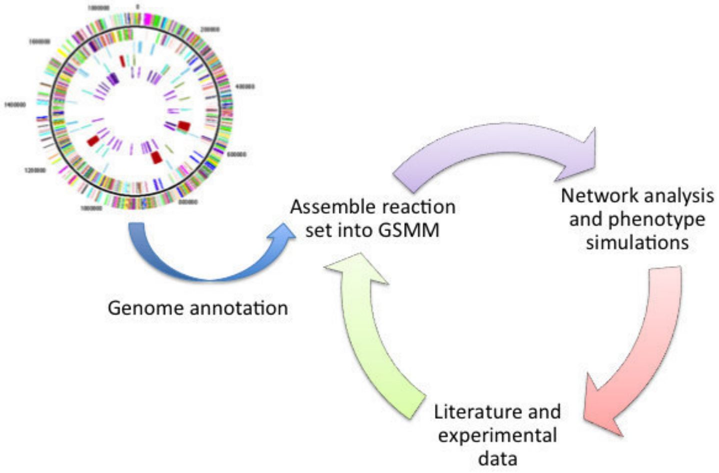

Genome annotation
Assemble reaction set into GSMM
Network analysis and phenotype simulations
Literature and experimental data

Figure 2.1: Model reconstruction cycle.

able to support a faster genome-scale metabolic models reconstruction. The Metabolic Models Reconstruction Using Genome-Scale Information (Merlin) [49], developed in our group, is a freeware tool that supports the reconstruction process, including the functional genomic annotation of the genome and subsequent construction of the portfolio of reactions.

Metabolic models have been used to simulate the cell phenotype under different environmental conditions and genetic changes [50]. These models, together with strain optimization tools, allow the identification of genetic targets for increasing yields productivities and robustness in industrial biotechnology processes [51, 52].

Additionally, over the last years, metabolic models have been used to understand some phenotypes associated with diseases [22], to find drug targets [20] and to study the relationship between different organisms [53] and cell types [18].

2.2.1 Human metabolic models

The human species is one of the most complex organisms since the number of genes, types and diversity of cells are huge. After the human genome sequencing and its annotation [54, 55], efforts have been made in the last decade to reconstruct human genome-scale metabolic models. Until now, four human GSMMs has been proposed [7, 8, 9, 10] and have been used to study human physiology and pathology.

The reconstruction of the first human GSMM was published in 2007 [7], under the name of *Recon 1*. This metabolic model accounts for the functions of 1.905 genes, 2.766 metabolites, and 3.742 metabolic and transport reactions and was reconstructed based on an extensive collection and evaluation of genomic and bibliomic data.

The model was validated through the simulation of 288 known metabolic functions present in different cells and tissue types. All related information is available in the BIGG database [56] (`http://bigg.ucsd.edu/`).

A few months later, a new metabolic model was published by Ma et al. [57], called the *Edinburgh Human Metabolic Model* (EHMM). This network was manually reconstructed by integrating genome annotation from different databases and metabolic reactions information from literature. In the first step of the reconstruction, the authors mainly collected all information from the databases KEGG [58], UniProt [43] and HGNC (*HUGO Gene Nomenclature Committee*) [59]. The second step of the reconstruction integrated information from the *Enzymes and Metabolic Pathways* database [60]. In 2010, the compartmentalization of the EHMM was completed [8]. The compartmentalization required the association of metabolic reactions to different cellular organelles and transport reactions that were added to allow the exchange of metabolites between such organelles.

In 2012, a new metabolic model of human cells, the *iHuman1512* [9], was developed based on the *Human Metabolic Reaction* (HMR) database. This database has been constructed from the two previous models, also incorporating information from KEGG and HumanCyc [61]. During the construction of this database, metabolites with lacking identifiers to external databases were left out along with their corresponding reactions.

This database has been expanded through the incorporation of the lipid metabolism, which accounts for 59 fatty acids rather than relying on generic fatty acid metabolites. The inclusion of fatty acids allowed the integration with lipidomics data and helped in understanding the contribution of lipids to the development of diseases [62]. The resulting HMR database version 2.0 contains 3,765 genes, 6,007 metabolites (3,160 unique metabolites) and 8,181 reactions, with 74% of the reactions associated to one or more genes.

In 2013, a new model has been proposed by Thiele et al. [10]- Recon 2. The Recon 2 is a community-driven expansion of the previous human metabolic model Recon 1, with several additions from other sources, such as the previous model EHMN [8], Hepatonet1 [17], a manually curated and functional model of hepatocyte metabolism, the acylcarnitine–fatty acid oxidation module [63], and the small intestinal enterocyte reconstruction [64]. Recon 2 accounts for 1,789 enzyme-encoding genes, 7,440 reactions and 2,626 unique metabolites distributed over eight cellular compartments. Recently, a new Recon 2 model version was published during 2015 with significant changes on GPR associations (`https://vmh.uni.lu`).

Based on the information available, a summary of the different human metabolic models is presented in following table (Table 2.1).

Table 2.1: Number of reactions, metabolites, genes and compartments present in the available human metabolic models. Species representing the same metabolite in different compartments are here considered as different metabolites.

	Recon 1	**EHMM**	**HMR 2.0**	**Recon 2.04**
Reactions	$3,742$	$6,216$	$8,181$	$7,440$
Metabolites	$2,766$	$6,522$	$6,007$	$5,063$
Genes	$1,905$	$2,693$	$3,765$	$2,140$
Compartments	8	9	8	8

Besides these models, human metabolic information is also available in Reactome [65, 66] and HumanCyc [61] databases. However, this information is not organized as a model and, therefore, can not be used to support phenotypes simulations.

Human GSMMs have been widely used in studies involving the discovery of biomarkers [11], generating context-specific metabolic models [14] and elucidating one of the most important and puzzling hallmarks of cancer, the Warburg effect [12].

2.3 Constraint-based modeling

Biological networks/models can be analysed using different modeling formalisms depending on the question to be answered, the biochemical knowledge and the availability of experimental data [67]. Mathematically, a metabolic model can be represented as a matrix ($S_{m \times n}$) of m metabolites and n reactions,

$$S_{m \times n} = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,n} \end{bmatrix}$$

where each element $s_{i,j}$ is the stoichiometric coefficient of the i^{th} metabolite on the j^{th} reaction.

A detailed quantitative description of the biological processes can be reached by ordinary differential equations [68]. However, kinetic parameters are rarely available for large-scale networks.

The analysis of the metabolic models can be simplified considering the steady-state assumption, i.e., the metabolites concentration remains constant over time [69]. Considering this assumption, it is possible to obtain flux distributions for the vector v, solving the system equations

$$S.v = 0 \tag{2.1}$$

where S is the matrix of stoichiometric coefficients, for a set of m metabolites and a set of n reactions, and v is the vector of n reaction rates (fluxes). Additionally, the maximum and the minimum flux values can be imposed for each reaction to define the thermodynamic feasibility (directionality) and flux capacity [70], as follows:

$$0 \leq v_i \leq vmax_i \quad , \forall i \in N_{irreversible} \tag{2.2}$$

$$vmin_i \leq v_i \leq vmax_i \quad , \forall i \in N_{reversible} \tag{2.3}$$

where v_i is the flux carried over reaction i, $N_{irreversible}$ and $N_{reversible}$ are the sets composed of all reversible and irreversible reactions, respectively, and $vmin_i$ and $vmax_i$ are the lower and upper bounds for the flux over reaction i.

Some flux distributions are impossible to occur *in vivo*. Therefore, additional constraints should be added to represent the cells environmental conditions, for instance, the nutrient uptake rates. Constraint-Based Modelling (CBM) [71] determines possible flux distributions which are optimal to a specific criteria that satisfies the previously defined constraints (Figure 2.2). For instance, to find the flux distributions that correspond to the maximum growth rate of an organism.

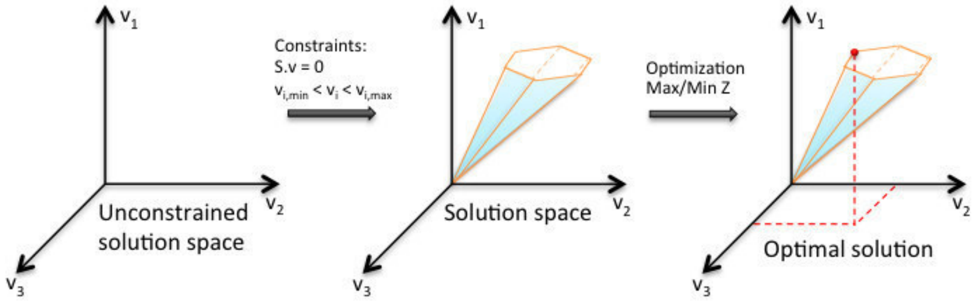


Figure 2.2: Constraint-based modeling.

One of the most widely used CBM approaches is Flux Balance Analysis (FBA) [2, 72], which can be represented as:

$$\begin{aligned} & max/min\ Z \\ & \quad s.t.: S.v = 0 \\ & \qquad v_{i,min} < v_i < v_{i,max} \end{aligned} \tag{2.4}$$

where $Z = c^T v$ is the objective function (a linear combination of fluxes, where

c is a vector of weights indicating how much each reaction contributes to the objective function). Usually the $Z = v_{biomass}$ when the objective function is the maximization of growth rate.

The assumption of maximal growth is acceptable under wild-type conditions. However, sometimes the organism is subjected to genetic perturbations, such as gene deletions. To deal with mutant strains, Segrè and coworkers introduced the Minimization Of Metabolic Adjustment (MOMA) method [73]. This method minimizes the sum of the squared differences between the wild type (typically calculated with FBA or given as a reference flux distribution) and the mutant flux distributions, thus defining a quadratic objective function, which translates into a quadratic programming (QP) problem.

With a similar approach, the Regulatory On/Off Minimization (ROOM) [74] algorithm tries to minimize the number of significantly changed fluxes, relative to the original flux distribution. This approach requires the introduction of binary variables in the objective function, thus converting the LP problem into a MILP one.

Both methods try to minimize the flux distribution difference between the wild type and the mutant cell based on the assumption that the organism will try to adjust its behaviour with the minimum possible effort.

2.4 Omics information

The abundance of biological information generated by high-throughput studies has enabled the identification and quantification of the individual components (genes, proteins and metabolites) of biological systems. These data are globally known as 'omics' data, and include genomics, transcriptomics, proteomics and metabolomics, just to name the most popular. The challenge of using omics data lies on the difficulty to integrate all these data. Nevertheless, when this is possible, such data allows a better understanding of the cell as a whole.

An overview of some techniques and main sources of information in each 'omic' field, is summarized in Table 2.2.

Table 2.2: Techniques and main sources of information for each omic platform.

Field	Techniques	Databases
Genomics	Microarray RNA-sequencing	GEO ArrayExpress GeneNote TCGA
Proteomics	Mass Spectrometry Gel-based protein separation	HPA HPRD
Metabolomics	Gas Chromatography–Mass Spectrometry Nuclear Magnetic Resonance	HMDB

Overall, different types of omics data allow a better understanding of many complex biological processes occurring in the cells and can be used in the reconstruction process of metabolic models.

2.4.1 Transcriptomics

Transcriptomics are, certainly, the most widely available type of omics data. Using DNA microarrays or other techniques, such as RNA-sequencing, allows the quantification of the expression levels of genes inside cells in different conditions [75, 76].

One of the most well-know databases for gene expression data is the Gene Expression Omnibus (GEO) [77]. This is a public repository that archives and distributes microarray, next-generation sequencing and other forms of high-throughput functional genomic data. In addition, a set of computer web-applications are available to query and download the gene expression patterns stored in GEO (`http://www.ncbi.nlm.nih.gov/geo/`).

The ArrayExpress Archive [78] is another repository which contains functional genomics experiments, including gene expression, where it is possible to query and download data in standard formats.

Using tools such as R/Bioconductor [79, 80] (e.g. the `limma` package) on data from the previously mentioned databases, it is possible to understand which genes are differentially expressed between different cell types or phenotypes, for instance normal vs cancer cells. However, this relative expression is unable to provide answers to the questions: " Which are the genes that are expressed in different phenotypes and what are their absolute levels of expression?".

The Gene Expression Barcode (GEB) [81] provides absolute measurements for most annotated genes, organized by tissue types, including diseased tissues. Considering standardized values obtained from GEO and ArrayExpress repositories, it is possible to convert these expression values to silenced and expressed genes applying a threshold. Moreover, it is possible to convert data from a single microarray into expressed/unexpressed values for each gene.

2.4.2 Proteomics

mRNA molecules are not always translated into proteins [82], and therefore amount of protein produced depends on the gene expression and the current state of the cell. Thus, the knowledge about the amounts of proteins in the cell, provided by proteomics data [83], is of foremost relevance. These data can confirm the presence of proteins and quantify the amount of proteins within a cell.

The Human Protein Atlas (HPA) [84] is a database with millions of high-resolution images showing the spatial distribution of protein expression profiles in normal tissues, cancer and cell lines for human cells.

Also, the Human Protein Reference Database (HPRD) [85] database represents and integrates information for each protein in the human proteome. All published data available in this database has been manually extracted from the literature, interpreted and analyzed. Nowadays, this database accounts for more than 30.000 protein entries.

2.4.3 Metabolomics

Another source of information is provided by metabolomics data which allows the identification and quantification of the small molecules present in the cells, tissues, organs and biological fluids using techniques such as Nuclear Magnetic Resonance (NMR) spectroscopy and Gas Chromatography-Mass Spectrometry (GC-MS) [86]. Those metabolites contribute for the design of metabolic pathways and the understanding of the interaction of proteins with environmental cell factors (e.g. drug exposure).

The Human Metabolome Database (HMDB) [87] contains spectroscopic, quantitative, analytic and molecular-scale information about human metabolites, their associated enzymes or transporters, their abundance and disease-related properties.

2.4.4 Others sources

There are other repositories of information, specialized in specific diseases such as cancer ("The Cancer Genome Atlas") and diabetes ("Diabetes Genome Anatomy Project"). The Cancer Genome Atlas (TCGA) [88] is a collaborative effort between many organizations to map the genomic changes that occur in major types and subtypes of cancer. Besides, the data that have been generated by TCGA's network are available in the TCGA Data Portal [89], that provides a platform for searching, downloading, and analyzing data that contains clinical information, genomic characterization data and high-throughput sequencing analysis of the tumor genomes.

2.5 Simulation methods

Several diseases such as cancer, diabetes, hypertension and heart diseases can be related with the abnormal metabolism of cells [90, 12]. Furthermore, human metabolism is complex and involves a large number of reactions that are highly interconnected by common metabolites [91]. Since the function of each tissue is so different, it is also expected that the metabolism of each cell type will also be distinct. However, the lack of information on tissue-specific metabolite exchanges is still a limitation to employ CBM methods [13].

Over the last decade, some approaches have been developed to integrate omics data to improve the phenotype prediction. In this section, the most relevant simulation methods it will be presented that give the flux distribution that better represent the data used as input.

2.5.1 iMAT

The reconstruction of tissue-specific metabolic models and the usage of omics data to improve the phenotype predictions is not new. Indeed, in 2008, Shlomi et al [13] developed the first approach called Integrative Metabolic Analysis Tool (iMAT), to predict the metabolic activity in ten human tissues: brain, heart, kidney, liver, lung, pancreas, prostate, spleen, skeletal muscle and thymus.

This approach integrates information of gene and protein expression with the existing human metabolic network model - Recon 1. The gene expression information was retrieved from the GeneNote (*Gene Normal Tissue Expression*) [92] database, while the Human Protein Reference Database (HPRD) was used as a source for proteomics data [13].

The post-transcriptional regulation is not reflected in the gene and protein expression data, so this method treats the expression levels of enzymes as cues for the probability that their associated reactions have metabolic flux. The highly, lowly and moderately expressed genes values are converted to 1, -1 and 0, respectively, through the gene-protein rules, replacing the logical 'and' and 'or' operators with 'max' and 'min' expressions. This transformation resulted in two subsets of reactions in the model: R_H (highly expressed) and R_L (lowly expressed).

The reconstruction of a context-specific metabolic network is done by solving an optimization problem, using a Mixed Integer Linear Programming (MILP) formulation, to find a steady-state metabolic flux distribution that satisfies the stoichiometric and thermodynamic constraints embedded in the model, the number of flux-carrying reactions associated with highly expressed enzymes is maximized and the number of flux-carrying reactions associated with lowly expressed genes is minimized.

The complete formulation is presented below:

$$
\begin{aligned}
max\ & \left(\sum\nolimits_{i \in R_H} (y_i^+ + y_i^-) + \sum\nolimits_{i \in R_L} y_i^+\right) \\
s.t\ & S.v = 0 \\
& v_{min} \le v_i \le v_{max} \\
& v_i + y_i^+(v_{min\ i} - \epsilon) \ge v_{min\ i} \qquad i \in R_H \\
& v_i + y_i^-(v_{max\ i} + \epsilon) \le v_{max\ i} \qquad i \in R_H \\
& v_{min\ i}(1 - y_i^+) \le v_i \le v_{max\ i}(1 - y_i^+) \qquad i \in R_L \\
& y_i^+, y_i^- \in [0, 1]
\end{aligned}
\tag{2.5}
$$

where v is the flux vector and S is a stoichiometric matrix, v_{min} and v_{max} are lower and upper bounds of the fluxes, respectively, the boolean variables y^+ and y^- represent whether the reaction is active (in either direction) and the ϵ value represent the minimum value that flux must have to for the reaction is considered active.

This method relies on enzyme-expression data to infer tissue-specific metabolic fluxes, thus it is not necessary to define an objective function (biomass equation) and metabolites exchanged by the tissue with biofluids, which indeed are unavailable for human tissues.

In [13], the validation of the predicted tissue-specific metabolic behaviour uses biological information from Human Metabolome Database (HMDB) and Braunschweig Enzyme Database (BRENDA). The metabolite exchanges that depend on membrane transporters were validated based on data on tissue specificity of transporters, obtained from the Human Membrane Transporter

Database and from the Transport Classification Database.

The predicted tissue-specific metabolic behaviour was compared to various data sources of genes, reactions and metabolites of each tissue type. For the ten tissues, the predicted results were significantly correlated with data sets, with the precision and recall varying between 0.36-0.7 and 0.37-0.55 [13], respectively. The accuracy reflects the overlap between the predicted tissue-associations of genes, reactions and metabolites and known tissue-associations derived from various data sources.

2.5.2 GIMME

Also in 2008, another research group developed a new algorithm to generate context-specific metabolic models, the *Gene Inactivity Moderated by Metabolism and Expression* (GIMME) [14].

This method uses gene expression combined with objective functions to create functional metabolic models. However, as expression data is known to be noisy, the results may vary depending on the methods used to convert the fluorescence intensity to semi-quantitative readings of mRNA molecule counts [93].

The GIMME algorithm takes three inputs: i) a set of gene expression data; ii) the template genome-scale metabolic model; iii) one or more Required Metabolic Functionalities (RMF) that the cell is known to perform.

Through the gene-protein-reaction rules, the algorithm uses the gene expression data to determine which reactions are inactive or active in the tissue. Reactions that correspond to expression levels below a specified threshold, chosen by the user, are tentatively declared inactive unless they are required for a desired functionality, according to a predefined objective function. During this process, an inconsistency score (IS) is calculated and represents the disagreement between the gene expression data and the flux distribution for an objective function. The optimization problem tries to minimize the IS to produce a flux distribution with the minimal differences to the expression data.

Therefore, the algorithm produces the flux distribution through a two-step procedure:

1. Run a FBA for each RMF to find the maximum possible flux. The RMFs represent metabolic tasks essential to the cell and the final result must satisfy, such as the growth or the production of a target metabolite.

2. Solve the following linear programming problem:

$$\begin{aligned} &min \sum c_i.|v_i| \\ &S.v = 0 \\ &a_i < v_i < b_i \\ \text{where } c_i = &\begin{cases} cutoff - x_i, & cutoff > x_i \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2.6}$$

   where x_i is the normalized gene expression data mapped to each reaction through the genes association present in the model; $cutoff$ is the value chosen by the user; S is the stoichiometric matrix; v is the flux vector; a_i and b_i are the lower and upper bounds for each reaction. If a reaction is one of the RMFs, the upper bound is set to the value found in step 1 (maximal value) and the lower bound to a fraction of its maximal value; otherwise, the reaction is constrained with the bounds present in the metabolic model.

The algorithm was used to describe the functional genome-scale metabolic models for skeletal muscle cells in different conditions. The results obtained for the human models were less interesting than expected, due the lack of available data for a substantial number of human metabolic reactions [14].

2.5.3 E-flux

The E-flux [15] method predicts the metabolic capacity based on expression data. This method extends FBA [72] by incorporating gene expression data into the metabolic flux constraints present in the formulation.

This approach starts by changing the reaction bounds present in the metabolic model to integrate information from expression data. In short, if the expression for a particular enzyme-coding gene is low, the upper bounds of the reactions associated with this gene, will be replaced by a small value. On the other hand, if expression is high, the new bounds will be similar to the original ones. Once the constraint transformations are defined, FBA is applied to determine a corresponding metabolic state or optimal metabolic capacity.

E-Flux involves solving the following optimization problem:

$$\begin{aligned} & max\ c^T v \\ & \quad S.v = 0 \\ & \quad a_i \leq v_i \leq b_i \end{aligned} \tag{2.7}$$

where v is a flux vector representing a reaction, S is the stoichiometric matrix, c is a vector of weights indicating how much each reaction (v) contributes to the objective function, and a_i and b_i are the lower and upper bounds of reaction i.

In the original publication [15], this method was used to predict the impact of drugs and environmental conditions on mycolic acid biosynthesis capacity in *Mycobacterium tuberculosis.*

2.6 Tissue-specific reconstruction methods

Recent studies have demonstrated that the metabolic profiles of tumor cells most likely depend on genotype and the tissue of origin, and this has implications regarding the design of therapies targeting tumor metabolism [94].

Understanding the human metabolism of different cell types and the interactions between them may lead us to determine efficient diagnosis and treatment of these diseases. Thus, it becomes essential to develop metabolic networks for distinct cell types/tissues.

During the last decade, some approaches have been developed to allow the understanding of cell types metabolism. Shortly, these methods use a generic human metabolic model as template, such *Recon 1*, and integrate omics data from a tissue or context specific. As a result, some of them return a new tissue-specific metabolic model, while others give also the flux distribution that better represent the data used as input. In this section, the most relevant methods will be presented.

2.6.1 MBA

A first approach has been proposed in 2010, named Model-Building Algorithm (MBA) [16]. This algorithm reconstructs a tissue-specific metabolic model from a generic model by integrating a variety of tissue-specific molecular data sources (literature-based knowledge, transcriptomic, proteomic, metabolomic and phenotypic data).

The first step of this algorithm is to infer, from the tissue-specific data, two sets of reactions denoted as the core reactions (C_H) and reactions that have a moderate probability to be carried out in the specific tissue (C_M). This division is made according to the accuracy level of the input data. In general, the C_H set includes human-curated tissue-specific pathways and the C_M set includes reactions certified by molecular data.

The aim of this method is to find the most parsimonious tissue-specific consistent model, which includes all the tissue-specific high-probability reactions (C_H), a maximal number of moderate probability reactions (C_M) and a set of additional reactions from the generic model that are required for gap filling, using a greedy heuristic search that is based on iteratively pruning reactions from the generic model (Figure 2.3). The full set of steps in this method is shown in Algorithm 1.

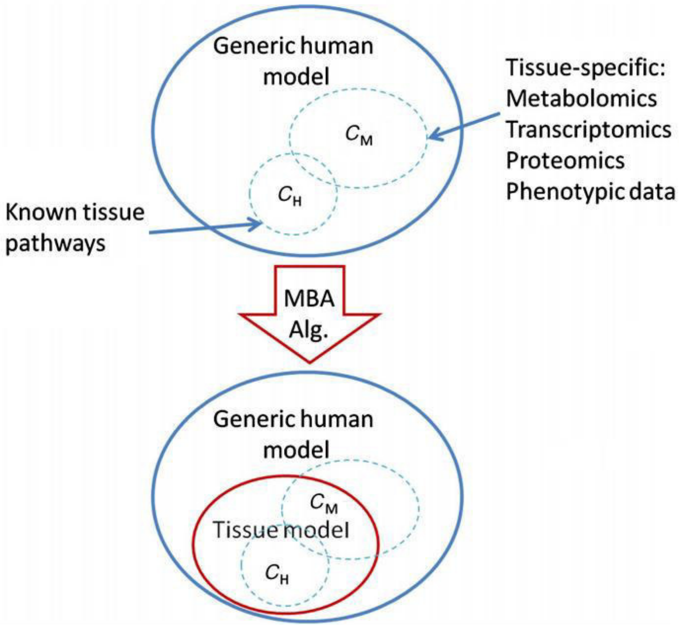


Figure 2.3: The diagram illustrates the function of the model-building algorithm (MBA). The algorithm is given tissue-specific reactions sets (C_H and C_M) as input and reconstructs a tissue model containing all of the C_H reactions, as many as possible C_M reactions, and a minimal set of other generic model reactions that are required for obtaining overall model consistency.

To validate this approach a new metabolic model of liver was built from the generic model Recon 1 [7]. The essential core, C_H, was extracted from literature-based curation, consisting in 37 intact metabolic pathways involving 779 reactions and 873 metabolites.The C_M consisted of a set of 304 reactions, and 484 metabolites and it was assembled from tissue-specific data sources, including metabolomics, transcriptomics, proteomics, and phenotypic data of

Algorithm 1 MBA algorithm pseudo code

```
function GENERATEMODEL($R_G$, $C_H$, $C_M$)
    $R_P \leftarrow R_G$
    $R_S \leftarrow R_P \backslash (C_H \cup C_M)$
    $P \leftarrow randomPermutation(R_S)$
    for ($r \in P$) do
        $inactiveR \leftarrow CheckModel(R_P, r)$
        $e_H \leftarrow inactiveR \cap C_H$
        $e_M \leftarrow inactiveR \cap C_M$
        $e_X \leftarrow inactiveR \backslash (C_H \cup C_H)$
        if ($|e_H| == 0 \; AND \; |e_M| < \delta * |e_X|$) then
            $R_P \leftarrow R_P \backslash (e_M \cup e_X)$
        end if
    end for
    return $R_P$
end function
```

the liver. As a result of applying the algorithm, the liver metabolic model consists of 1,827 reactions and 1,360 metabolites.

2.6.2 INIT and tINIT

The *Integrative Network Inference for Tissues* (INIT) algorithm was proposed in 2012 by Jens Nielsen's team [9]. The algorithm uses cell type specific information from HPA as the main source of evidence for assessing the presence or absence of metabolic enzymes in each of the human cell types. Moreover, other data sources as tissue specific gene expression and metabolomics data from HMDB are also used.

This algorithm requires a connected template human metabolic model as input, so the first step was to provide a reliable and up to date genome-scale model template. So, the Human Metabolic Reaction (HMR) database was built with the elements of previously generic genome-scale human metabolic models (Recon1, EHMM, HumanCyc), as well as with information from KEGG database.

The protein evidence levels retrieved from HPA or gene expression levels from GEO datasets are converted to reaction scores through the GPR associations present in the template model. The algorithm was formulated as a MILP and tries to maximize the sum of scores for reactions that can carry flux. According to the HMDB, the production of metabolites, known to be present in the cell type, will be imposed by the formulation to ensure its synthesis in the final model. Another detail in this formulation is the fact that the steady-state conditions are not imposed allowing a small accumulation of internal metabolites. This avoids the removal of reactions with dead end metabolites.

The INIT formulation and can be specified as:

$$
\begin{aligned}
& max \left(\sum_{i \in R} w_i y_i + \sum_{j \in M} x_j \right) \\
& S.v = b \\
& |v_i| \leq 1000 y_i \\
& |v_i| + 1000(1 - y_i) \geq \varepsilon \\
& v_i \geq 0 \qquad i \in irreversible \\
& b_j \leq 1000 x_i \\
& b_j + 1000(1 - x_i) \geq \varepsilon \\
& b_j \geq 0 \\
& x_j = 1 \qquad j \in present \\
& y_i, x_j \in 0, 1
\end{aligned}
\tag{2.8}
$$

where S is the stoichiometric matrix, v the vector of reaction rates, b a vector of net accumulation or consumption rates for each internal metabolite, R represents the reactions and M the metabolites. The parameter ε is an arbitrarily small positive number and y_i, x_j correspond the active or inactive state of a reaction and a metabolite respectively. The value of w_i can be 20, 15, 10 or -8 to represent the high, medium, low and absent evidence levels for proteins in the HPA. If the evidence comes from gene expression levels,

w_i is calculated as follows:

$$w_i = 5\ log\left(\frac{Signal_{i,j}}{Average_i}\right) \tag{2.9}$$

The signal of gene i in tissue j is divided by the average signal across all the tissues.

A couple of years later, a new version of INIT algorithm was proposed. The *Task-driven Integrative Network Inference for Tissues* (tINIT) [20], which reconstructs tissue-specific metabolic models based on protein evidence from HPA and a set of metabolic tasks that the final context-specific model must perform. These tasks are used to test the production or uptake of external metabolites, but also the activation of pathways that occur in a specific tissue. During the tasks validation in the template model, a set of required reactions will be found and constraints to ensure the flux through these reactions are added to the formulation. Another two improvements from the previous version are the addition of constraints to guarantee that irreversible reactions operate in one direction only and the possibility of choice whether net production of all metabolites should be allowed.

2.6.3 mCADRE

Also in 2012, a new method was developed named *Metabolic Context specificity Assessed by Deterministic Reaction Evaluation* (mCADRE) [21]. This method is able to infer a tissue-specific network based on gene expression data, network topology and reaction confidence levels.

Based on the expression score, the reactions of the global model, used as template, are ranked and separated in two sets: core and non-core. All reactions with expression-based scores higher than a threshold value are included in the core set, while the remaining reactions make the non-core set.

In this method, the expression scores do not represent the expression levels, but rather the frequency of expressed states over several transcript profiles. Hence, it is necessary to initially binarize the expression data. Thus, it is possible to use data retrieved from the Gene Expression Barcode (GEB) project that already contains binary information on which genes are present or not in a specific tissue/ cell type.

Reactions from the non-core set are ranked according to the expression scores, connectivity-based scores and confidence level-based scores. Then, sequentially, each reaction is removed and the consistency of the model is tested. The elimination only occurs if the reaction does not prevent the production of a key-metabolite, i.e. metabolites that have evidence to be produced in the context-specific model reconstruction, and the core consistency is preserved. The algorithm is provided below as Algorithm 2.

Algorithm 2 mCADRE algorithm pseudo code

```
function GENERATEMODEL($(R_G, treshold)$)
    $R_P \leftarrow R_G$
    $R_C \leftarrow score(R_P) > treshold$
    $coreActiveG \leftarrow flux(r)! = 0, r \in R_C$
    $R_{NC} \leftarrow R_P \backslash R_C$
    for $(r \in order(R_{NC}))$ do
        $inactiveR \leftarrow CheckModel(R_P, r)$
        $s1 = |inactiveR \cap R_C|$
        $s2 = |inactiveR \cap R_{NC}|$
        if $(r \notin withExpressionValues\ AND$
                $s1 \backslash s2 <= RACIO\ AND$
                $checkModelFunction(R_p \backslash inactiveR))$ then
            $R_P \leftarrow R_P \backslash inactiveR$
        else
            if $(|s1| == 0\ AND$
                $checkModelFunction(R_p \backslash inactiveR))$ then
                    $R_P \leftarrow R_P \backslash inactiveR$
            end if
        end if
    end for
    return $R_P$
end function
```

Comparing with the MBA algorithm, mCADRE presents some improvements: allows the definition of key metabolites; some reactions of core set can be removed from the final model; and, it is only necessary to run the algorithm once, since the order of pruning the reactions is not random.

2.6.4 FASTCORE and FASTCOMICS

Also similar to MBA, the FASTCORE [22], proposed in 2014, is a generic algorithm for context-specific metabolic models reconstruction that takes as input a core set of reactions and a generic metabolic model.

Firstly, it converts the initial model to a consistency model, i.e. only reactions that can carry flux in at least one feasible flux distribution are preserved. This can be done by using existing approaches such as Flux Variability Analysis (FVA) or a new one proposed in the work of Vlassis and co-workers [22] for fast consistency check (FASTCC) of a network. Next, it searches for a subnetwork from the generic model that contains all reactions present in the core set and a minimal set of additional reactions, necessary to guarantee the consistency of the final model.

Some advantages of this algorithm are that it can be applied to integrate different kinds of "omics" data through the core set compilation by the user, and there is no need to define parameters except the flux threshold ϵ, which is used to guarantee the required minimum flux.

Although the MBA and FASTCORE objectives are the same, that is , to find a minimal consistent model with all core reactions, the strategy is significantly different. While MBA starts with all reactions and iteratively prunes reactions from the generic model, FASTCORE iteratively expands the active set A, starting with $A = \emptyset$.

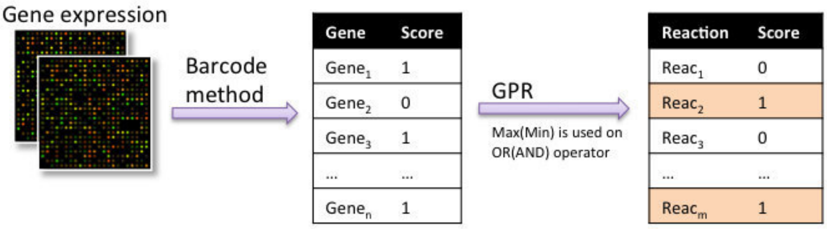
The algorithm maintains a set, $J \subseteq C$, that is initialized with the irreversible reactions in C, and a "penalty" set $P = (N \backslash C) \backslash A$ that contains all non-core reactions that have not been added to the set A. While not all reactions from the core set are in the final model, the algorithm appends the result of a function called *findSparseMode* to the set A. This function returns the set of reactions from the non-core set that maximizes the number of reactions active from the set J. Formally, the algorithm can be described as shown in Algorithm 3.

Algorithm 3 FASTCORE algorithm pseudo code

```
Let $N$ the set of all reaction in the model, $C$ the core set reaction and $I$
the set of irreversible reactions;
function FASTCORE($N, C$)
    $J \leftarrow C \cap I$
    $flipped \leftarrow False, singleton \leftarrow False$
    $A \leftarrow findSparseMode(J, P, singleton)$
    $J \leftarrow C \backslash A$
    while $J \neq \emptyset$ do
        $P \leftarrow P \backslash A$
        $A \leftarrow A \cup findSparseMode(J, P, singleton)$
        if $J \cap A \neq \emptyset$ then
            $J \leftarrow J \backslash A, flipped \leftarrow False$
        else
            if $flipped$ then $flipped \leftarrow False, singleton \leftarrow True$
            else
                $flipped \leftarrow True$
                if $singleton$ then $\widetilde{J} \leftarrow firstElement(J)$
                else
                    $\widetilde{J} \leftarrow J$
                end if
                for $r \in \widetilde{J} \backslash I$ do
                    flip the sign in stoichiometric matrix
                    and swap the bounds of reaction $r$
                end for
            end if
        end if
    end while
end function
```

Based on FASTCORE, a new method has been proposed, also in 2014, termed FASTCOMICS [95]. This method uses microarray expression data to infer the core reactions used in the original method. Microarrays are the most popular of the 'omics' data sources, however the association with the gene expression levels and active reactions is not trivial [82].

FASTCOMICS is performed in two steps: generate the core set of reactions based in transcriptomic data and reconstruct the context-specific metabolic model using the FASTCORE algorithm. The first step of the FASTCOMICS workflow is the discretization of microarray expression levels to build the core set of reactions. The continuous expression values are converted to estimated values of expressed (ones) and no-expressed(zeros) using the GEB algortithm [96]. GEB uses the knowledge of abundantly publicly available microarray data sets and the intensity distribution of each probe set, to classify the genes as expressed or non-expressed (see Figure 2.4).



Gene	Score
Gene$_1$	1
Gene$_2$	0
Gene$_3$	1
...	...
Gene$_n$	1

Reaction	Score
Reac$_1$	0
Reac$_2$	1
Reac$_3$	0
...	...
Reac$_m$	1

Figure 2.4: Conversion of gene expression levels to reaction scores. In the first step, the gene expression values are converted to expressed / not expressed status through the gene expression barcode method. Next, using the GPR associations present in the model the score of each reaction is calculated.

The second step of the workflow, is the reconstruction of the context-specific model through the FASTCORE algorithm and can be depicted in Figure 2.5.

This workflow allows the definition of media constraints and forces the biomass reaction to carry flux to find the required set of reactions that allow the production of biomass.

This new set of reactions is then appended to the core set from the previous step. Finally, a new run of FASTCORE, where all reactions from the core set are forced to carry flux, is performed to find the context-specific model.

When comparing these two methods with other competing algorithms for building of context-specific models like mCADRE [21], tINIT [9] or the MBA [16], FASTCORE and FASTCOMICS reveal a higher performance. Depending

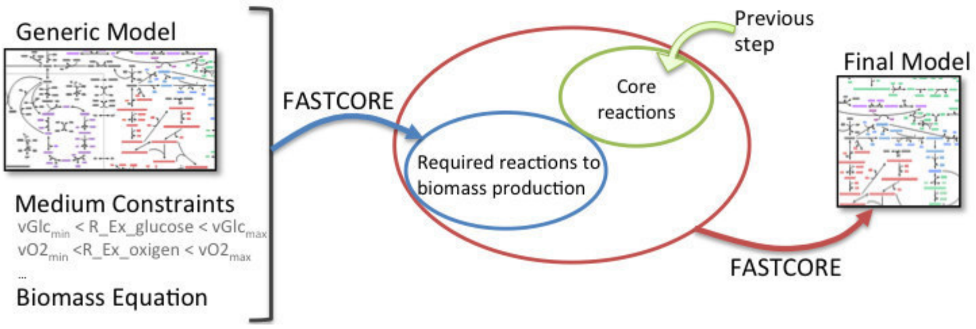

Generic Model
Medium Constraints
vGlc$_{min}$ < R_Ex_glucose < vGlc$_{max}$
vO2$_{min}$ <R_Ex_oxigen < vO2$_{max}$
...
Biomass Equation
FASTCORE
Required reactions to biomass production
Core reactions
Previous step
Final Model
FASTCORE

Figure 2.5: FASTCOMICS workflow: first, the method runs to find the required reactions to biomass production; next, an additional set of reactions are joined and a second run is performed to build the final model.

on the generic model size, FASTCORE can generate the reconstruction of context-specific GSMMs in a few minutes, whereas other algorithms would take hours or days [22].

2.6.5 PRIME

Recently, the *Personalized Reconstruction of Metabolic models* (PRIME) [23] method has been published, which utilizes both molecular and phenotypic data to reconstruct context-specific GSMMs.

Similar to E-flux, the PRIME method tries to adjust the reaction bounds according to the genes expression levels received as input. Nevertheless, some differences have been introduced in this method, namely:

- the bounds of relevant reactions related with the genes that affect the central cellular phenotype are changed;
- additional phenotypic data (growth rate) are used to establish the relation between the gene expression levels and the flux rates and to modify the bounds accordingly;
- modifies the flux bounds within a pre-defined range to avoid the differences between simulation and experimental growth rate.

Similarly to other methods, PRIME takes as input a generic metabolic model, 'omics' and experimental data used to obtain the final model. In this case, gene expression levels (transcriptomic data) and measurements of growth rates are used.

The method workflow can be described in two steps:

1. find the set of genes that significantly correlate with the phenotype (growth rate);
2. the upper bounds of reactions identified in (1) are modified according to the expression levels.

However, PRIME has some limitations since it is based on the assumption that all cells try to maximize their proliferation and depends on measurements of a specific phenotype that in most cases are not available [23].

In this study [23], the authors have built more than 280 models for normal and cancer cell-lines, utilizing them to predict drug targets that inhibit the proliferation of cancer cells, but not the normal cells.

2.7 Conclusion

Several methods have been proposed to improve prediction of the phenotype using omics data and to reconstruct metabolic models for a specific tissue or context. The development of these methods has become possible thanks to the increasing amount of high-throughput data available in the last decades.

Here, the main algorithms were presented, however, most of them are not publicly implemented or their use is difficult for non-programmers. Thus, it is crucial to develop an integrated framework to make these methods available to all researchers.

Chapter 3

Development of Software Tools

This chapter describes the implementation options during the development of the framework for the methods detailed in the previous chapter. The development was made over existing software that will be described here. The new developments regarding this work will be presented next, being provided both a description of the implemented functionality and the implementation technical details.

3.1 Introduction

In the last years, the increasing amount of high-throughput data available allowed the surge of phenotype prediction methods, resorting to the integration of transcriptomic and proteomic data [97], which can improve the accuracy of metabolic model predictions. Generically, these methods can be divided into two categories: the first encompassing methods where reaction fluxes are considered on/off based on a cutoff expression level (including iMAT and GIMME), and the second where the regulation of fluxes is based on relative gene or protein expression (E-flux method). These methods have been detailed in section 2.5.

Furthermore, several tissue-specific metabolic model reconstruction methods have been proposed to deepen knowledge on specific contexts. These methods, already detailed in section 2.6, also use trancriptomics and proteomics as the main sources of input data.

However, until now, the usage of these methods has been limited to developers or experienced bioinformaticians, since a platform that provides a user friendly interface to perform such tasks is not available. Thus, in the course of this work, a framework with the most relevant methods was developed and integrated with an user-friendly open source software, OptFlux [24], a reference tool that provides numerous tools for constraint-based modeling tasks and metabolic engineering applications.

The developed framework is composed by an application programming interface (API) for developers, who can use the provided library to extend the available methods, and a graphical user interface (GUI), integrated into OptFlux in the form of novel plug-ins, which encapsulates the developed tools for non programming users.

The API layer provides three main features: loading and integrating omics data with the metabolic model; simulating the metabolic phenotype and reconstructing tissue-specific metabolic models methods using omics as the main input (Figure 3.1). Some of the implemented methods also use metabolic tasks to evaluate the reaction deletion effect over the reconstruction process. Thus, it was also required to develop methods to import and validate metabolic tasks.

This open-source API framework is available in SourceForge repository (`https://sourceforge.net/p/optflux/`) within a project called `mewomics-integration`. Users with computational skills are able to use the provided library or contribute to its extension with new methods.

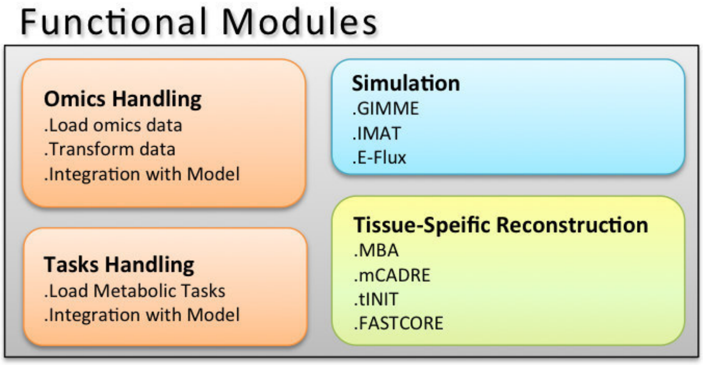


Figure 3.1: Functional Modules developed in the Omics framework.

3.2 Metabolic Engineering Workbench

In this section, we describe the existing core libraries, which provided the basis for the development of the libraries performed in this work.

The *Metabolic Engineering Workbench* (MEW) is a software framework that supports *in silico* metabolic engineering tasks. This framework includes nine libraries: `mewcore`, `regulatorycore`, `biocomponents`, `biologicalnetscore`, `guituilities`, `biovisualizercore`, `availablemodelsapi`, `solvers` and `utilities`, being the most relevant for this work discussed in detail in the present section.

The framework is fully implemented in Java, an object-oriented programming, platform independent and portable language. The execution of all Linear Programming (LP) and Mixed-Integer Linear Programming (MILP) optimization procedures uses GNU Linear Programming Kit (GLPK) [98]. Moreover, LibSBML [99] is used to handle files in the Systems Biology Markup Language (SBML) [100] format. The main libraries and their classes are detailed over the next subsections.

The main capabilities of the MEW framework can be grouped into four distinct functional areas, as shown in Figure 3.2.

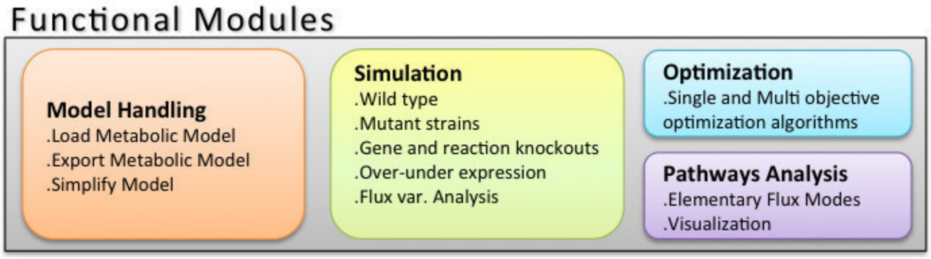


Figure 3.2: Functional Modules present in the Metabolic Engineering Workbench framework.

3.2.1 The BioComponents library

The *BioComponents* library provides classes for reading and writing metabolic models in several formats, such as SBML, Metatool [101], BioOpt/BioMet [102], flat-files and a generic table format (coma/tab separated values).

Each of these file types can be read through the correspondent class from package `container.io.readers`. Those classes implement the `IContainerBuilder` interface, which guarantees the implementation of methods to retrieve all necessary information to build an instance of the class `Container`. The diagram of classes, including the main classes involved in the reading process is depicted in Figure 3.3.

The `Container` class implements a constructor that takes an instance of an implementation of `IContainerBuilder` as argument, which is used to populate the instance object. The main class, `Container`, holds all information related with the metabolic models: reactions, metabolites, genes, pathways and additional information that can be used to integrate the entities from the models with external databases, such as the Kyoto Encyclopedia of Genes and Genomes database (KEGG) or the Chemical Entities of Biological Interest database(ChEBI).

The information present in the class `Container` involves several other classes used to store all the information, as:

- `CompartmentCI`: contains information about a cellular compartment,

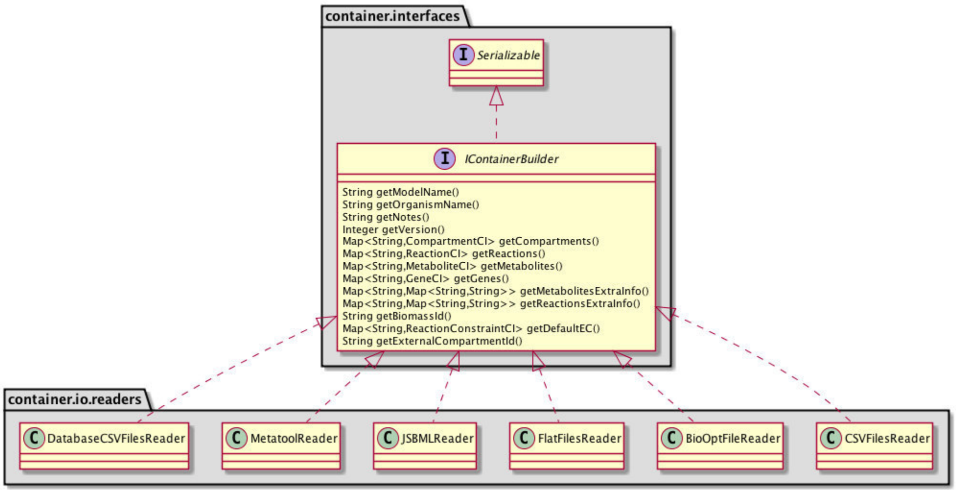

container.interfaces
I Serializable
I IContainerBuilder
String getModelName()
String getOrganismName()
String getNotes()
Integer getVersion()
Map<String,CompartmentCI> getCompartments()
Map<String,ReactionCI> getReactions()
Map<String,MetaboliteCI> getMetabolites()
Map<String,GeneCI> getGenes()
Map<String,Map<String,String>> getMetabolitesExtraInfo()
Map<String,Map<String,String>> getReactionsExtraInfo()
String getBiomassId()
Map<String,ReactionConstraintCI> getDefaultEC()
String getExternalCompartmentId()
container.io.readers
C DatabaseCSVFilesReader
C MetatoolReader
C JSBMLReader
C FlatFilesReader
C BioOptFileReader
C CSVFilesReader

Figure 3.3: Main classes involved in the metabolic models reading process. Each reader class implements the interface `IContainerBuilder`.

as the name, identifier and the list of metabolites.

- `ReactionCI`: contains information on a metabolic reaction, besides the basic information (identifier, name, type, etc.), gene rules and proteins rules are stored based on the GPR associations present in the model. The products and reactants are stored in a map where the metabolite identifier is the key and the stoichiometric coefficient is stored on an instance of `StoichiometryValueCI`.

- `MetaboliteCI`: contains information about a metabolite, such as identifier, name, formula, etc. and a list of reaction identifiers where the metabolite is a reactant or product.

- `GeneCI`: contains information about a gene, as the identifier, name and the list of reactions that contain this gene in their GPR associations.

- `ReactionConstraintsCI`: class to store the lower and upper bounds for a reaction flux.

The external information related with metabolites and reactions are saved as a map of maps, with the structure `Map <String, Map <String, String`

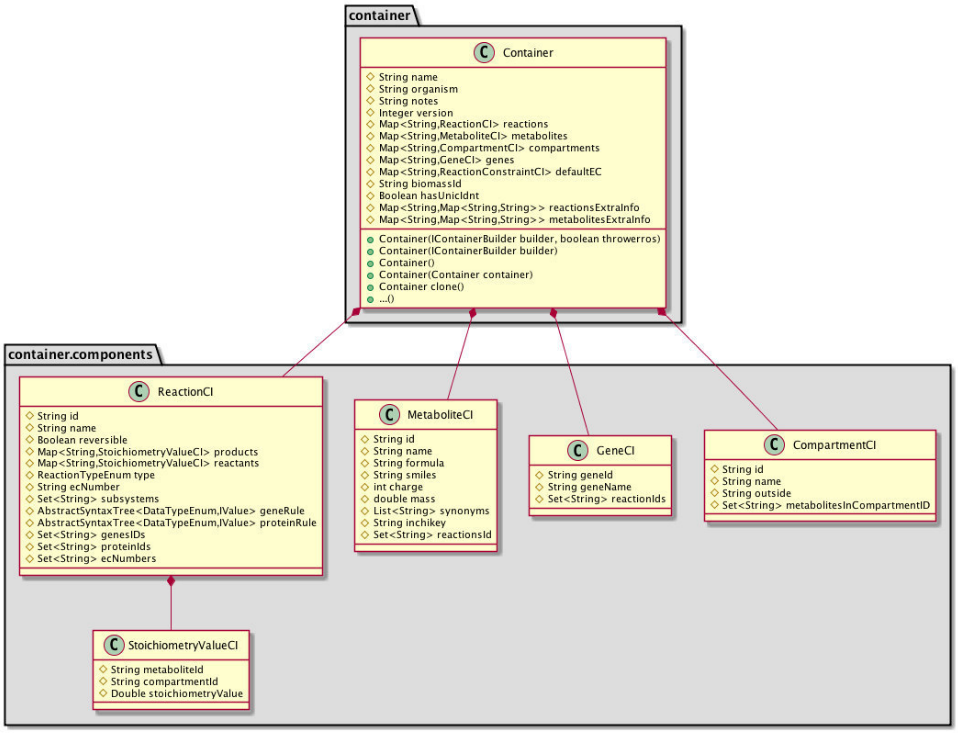


Figure 3.4: The `Container` class and its components used to store the metabolic model information.

>>.The *key* of the external map is the information type and the *key* of the internal map is composed of reaction or metabolite identifiers. For instance, for saving information related with KEGG identifiers associated with metabolites, the external info has the following structure:

`["KEGG"-->` [$meta_{H_2O}$ −>C00001 ,
$meta_{ATP}$ −>C00002,
… ,
$meta_{ala}$−>C19779]
]

The `Container` is the main class for the entire framework, since the metabolic model it represents is the common base in all operations and methods. This library also implements writing methods to save the content of a `Container` in several formats, such as SBML, Metatool or CSV.

3.2.2 The MEWCore library

MEWCore, as the name suggests, is the core library in the MEW framework. It is responsible for the formulation of phenotype prediction methods, the strain optimization procedures, the model simplification methods, the conversion of model formats and for identifying critical genes/reactions. This library is composed by several packages, which are detailed next.

The Model package

This package contains data structures to support all the information regarding stoichiometric metabolic models. A stoichiometric model is composed by metabolite and reaction sets and a matrix with the relation among these entities through the stoichiometric coefficients present in each reaction. The information related to pathways and GPR associations, when available, is also integrated in the model data structure.

The `SteadyStateModel` is the main class used to store the stoichiometric model information. This class aggregates information provided by other classes, namely :

- `IStoichiometricMatrix:` this interface contains the abstract methods to manipulate the stoichiometric matrix .The class `ColtSparseStoichiometricMatrix` is usually used as an implementation of this interface. This class contains the stoichiometric matrix, where each element $a_{i,j}$ represents the coefficient of i^{th} metabolite on the j^{th} reaction;
- `Reaction:` contains information about a reaction, including its name, identifier, reversibility and flux bounds;
- `Metabolite:` contains information about a metabolite, including its identifier, name, compartment as the main fields;
- `Compartment:` contains information related with the compartment, such as name, identifier and the set of metabolite identifiers present in that compartment;

- `Pathway`: contains the metabolites and reactions sets present in a specific pathway.

This class also allows the definition of basic model properties, such as the name, model version and the biomass reaction (a particular reaction used to represent cellular growth).

The `SteadyStateGeneReactionModel` class is an extension of `SteadyStateModel`, where the information about genes, proteins and the GPR associations is stored. The information present in both `SteadyStateGeneReactionModel` and `Container` classes is the same, but organized in a different structure. The overall class diagram is depicted in Figure 3.5

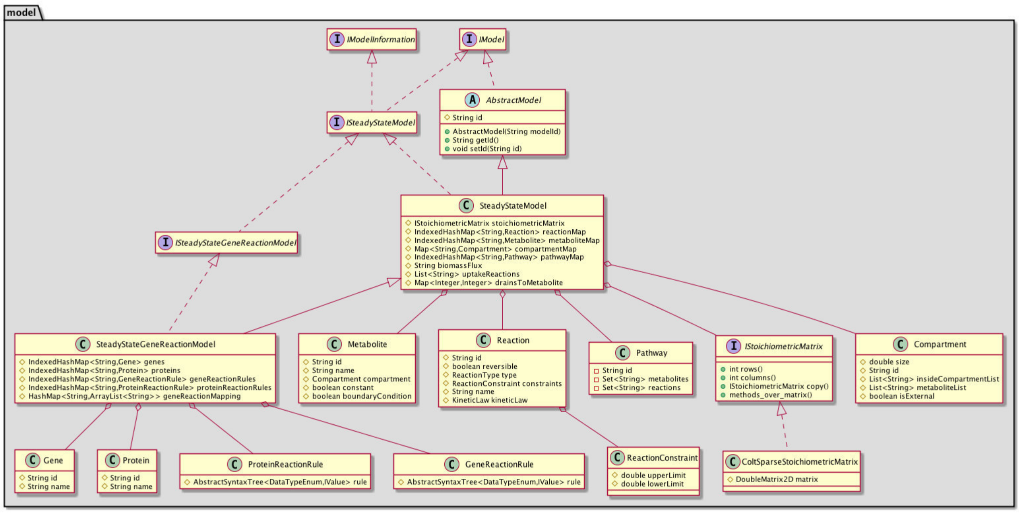

model
I IModelInformation
I IModel
I ISteadyStateModel
A AbstractModel
◇ String id
● AbstractModel(String modelId)
● String getId()
● void setId(String id)
C SteadyStateModel
◇ IStoichiometricMatrix stoichiometricMatrix
◇ IndexedHashMap<String,Reaction> reactionMap
◇ IndexedHashMap<String,Metabolite> metaboliteMap
◇ Map<String,Compartment> compartmentMap
◇ IndexedHashMap<String,Pathway> pathwayMap
◇ String biomassFlux
◇ List<String> uptakeReactions
◇ Map<Integer,Integer> drainsToMetabolite
I ISteadyStateGeneReactionModel
C SteadyStateGeneReactionModel
◇ IndexedHashMap<String,Gene> genes
◇ IndexedHashMap<String,Protein> proteins
◇ IndexedHashMap<String,GeneReactionRule> geneReactionRules
◇ IndexedHashMap<String,ProteinReactionRule> proteinReactionRules
◇ HashMap<String,ArrayList<String>> geneReactionMapping
C Metabolite
◇ String id
◇ String name
◇ Compartment compartment
◇ boolean constant
◇ boolean boundaryCondition
C Reaction
◇ String id
◇ boolean reversible
◇ ReactionType type
◇ ReactionConstraint constraints
◇ String name
◇ KineticLaw kineticLaw
C Pathway
□ String id
□ Set<String> metabolites
□ Set<String> reactions
I IStoichiometricMatrix
● int rows()
● int columns()
● IStoichiometricMatrix copy()
● methods_over_matrix()
C Compartment
◇ double size
◇ String id
◇ List<String> insideCompartmentList
◇ List<String> metaboliteList
◇ boolean isExternal
C Gene
◇ String id
◇ String name
C Protein
◇ String id
◇ String name
C ProteinReactionRule
◇ AbstractSyntaxTree<DataTypeEnum,IValue> rule
C GeneReactionRule
◇ AbstractSyntaxTree<DataTypeEnum,IValue> rule
C ReactionConstraint
◇ double upperLimit
◇ double lowerLimit
C ColtSparseStoichiometricMatrix
◇ DoubleMatrix2D matrix

Figure 3.5: Class diagram representing the structure of the classes used to store the information of steady state metabolic models.

The data structure for the definition of environmental conditions, `EnvironmentalConditions` class, is also contained in this package. It contains the information about metabolite uptakes and reaction constraints used in the phenotype prediction methods and in the strain optimization tasks. The `EnvironmentalConditions` class contains an identifier that characterizes its specific instance and a mapping data structure where the *key*, a reaction identifier, is mapped to a `ReactionConstraint` object, which contains the lower and upper bounds of the reaction.

The Simulation package

The *Simulation* package contains the formulations for the phenotype simulation methods. These methods allow the simulation of wild-type and mutant strains using environmental conditions or gene/reaction knockouts.

In the current version, formulations such as the Flux Variance Analysis (FVA)[103], Flux Balance Analysis (FBA)[2, 72], parsimonious enzyme usage FBA (pFBA), Minimization of Metabolic Adjustment (MOMA)[73], Regulatory on/off minimization of metabolic flux changes (ROOM)[74] and Minimization of Metabolites Balance (MiMBl)[104] are implemented in this package. Each one of these methods interacts with the *Solver* package (described in Section 3.2.3) to solve the underlying linear or integer programming problem.

One of the most important classes, the `SteadyStateSimulationControlCenter`, is responsible for controlling and aggregating all features mentioned above. The Figure 3.6 shows the main classes that interact with the `SteadyStateSimulationControlCenter`.

The `SteadyStateSimulationControlCenter` receives the configuration mapping object as an input. The configuration class contains properties, such as the environmental conditions, the objective function used in the simulation methods, the solver specification, the metabolic model, the simulation method, among others. Moreover, the `SteadyStateSimulationControlCenter` has also a static variable called `factory`, which holds the association between the method name and the formulation class.

The formulation problem is instantiated in execution time, through the `SimulationMethodsFactory` class. Using the factory method pattern in the creation of the formulation problems avoids the replication of code and simplifies the process of adding new simulation methods to the framework.

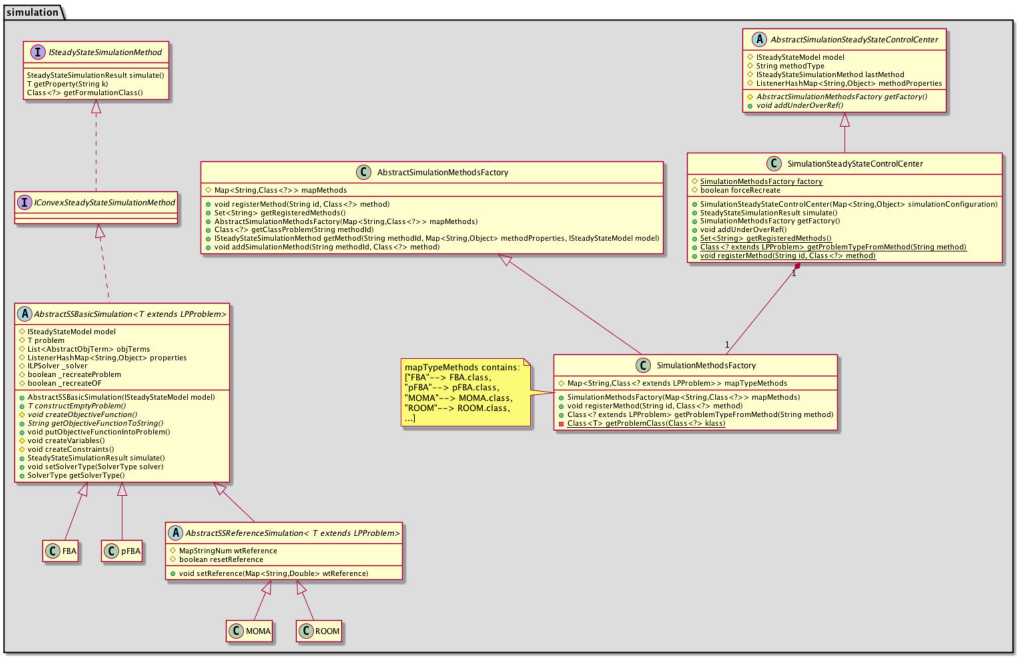

simulation
ISteadyStateSimulationMethod
SteadyStateSimulationResult simulate()
T getProperty(String k)
Class<?> getFormulationClass()
IConvexSteadyStateSimulationMethod
AbstractSSBasicSimulation<T extends LPProblem>
ISteadyStateModel model
T problem
List<AbstractObjTerm> objTerms
ListenerHashMap<String,Object> properties
ILPSolver _solver
boolean _recreateProblem
boolean _recreateOF
AbstractSSBasicSimulation(ISteadyStateModel model)
T constructEmptyProblem()
void createObjectiveFunction()
String getObjectiveFunctionToString()
void putObjectiveFunctionIntoProblem()
void createVariables()
void createConstraints()
SteadyStateSimulationResult simulate()
void setSolverType(SolverType solver)
SolverType getSolverType()
FBA
pFBA
AbstractSSReferenceSimulation< T extends LPProblem>
MapStringNum wtReference
boolean resetReference
void setReference(Map<String,Double> wtReference)
MOMA
ROOM
AbstractSimulationMethodsFactory
Map<String,Class<?>> mapMethods
void registerMethod(String id, Class<?> method)
Set<String> getRegisteredMethods()
AbstractSimulationMethodsFactory(Map<String,Class<?>> mapMethods)
Class<?> getClassProblem(String methodId)
ISteadyStateSimulationMethod getMethod(String methodId, Map<String,Object> methodProperties, ISteadyStateModel model)
void addSimulationMethod(String methodId, Class<?> method)
AbstractSimulationSteadyStateControlCenter
ISteadyStateModel model
String methodType
ISteadyStateSimulationMethod lastMethod
ListenerHashMap<String,Object> methodProperties
AbstractSimulationMethodsFactory getFactory()
void addUnderOverRef()
SimulationSteadyStateControlCenter
SimulationMethodsFactory factory
boolean forceRecreate
SimulationSteadyStateControlCenter(Map<String,Object> simulationConfiguration)
SteadyStateSimulationResult simulate()
SimulationMethodsFactory getFactory()
void addUnderOverRef()
Set<String> getRegisteredMethods()
Class<? extends LPProblem> getProblemTypeFromMethod(String method)
void registerMethod(String id, Class<?> method)
1
1
mapTypeMethods contains:
["FBA"--> FBA.class,
"pFBA"--> pFBA.class,
"MOMA"--> MOMA.class,
"ROOM"--> ROOM.class,
...]
SimulationMethodsFactory
Map<String,Class<? extends LPProblem>> mapTypeMethods
SimulationMethodsFactory(Map<String,Class<?>> mapMethods)
void registerMethod(String id, Class<?> method)
Class<? extends LPProblem> getProblemTypeFromMethod(String method)
Class<T> getProblemClass(Class<?> klass)

Figure 3.6: Class diagram of the main classes involved in the phenotype simulation on the MEW framework.

The interface `ISteadyStateSimulationMethod` defines the functions that must be implemented by any formulation method. The interface `IConvexSteadyStateSimulationMethod`, an extension of the previous one, contains the additional functions to support the persistent mode on the *Solvers* library.

The formulation classes are an extension of the abstract class `AbstractSSBasicSimulation`, which implements the interfaces referenced above, containing the generic methods to interact with the Solvers library. When the formulation problem contains a reference flux distribution, the `AbstractSSReferenceSimulation` should be used as the abstract class. Moreover, all the classes must implement a method called *simulation* responsible to run the formulation problem

At the end, the result of the *simulate* method present in `SteadyStateSimulationControlCenter`, returns an instance of `SimulationSteadyStateResult`, which contains the flux distribution that represents the final phenotype of the solution.

The Figure 3.7 depicts the main steps in phenotype simulation using the MEW classes.

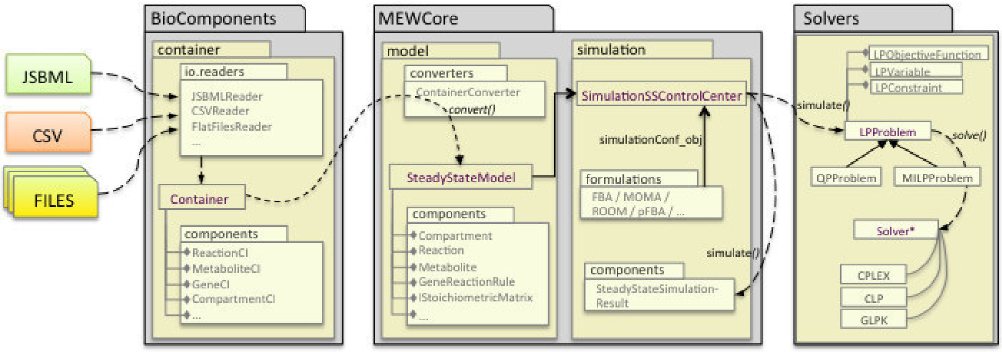


Figure 3.7: The API layer highlighting the *MewCore, BioComponents* and *Solvers* packages of the MEW framework for phenotype simulation.

The Simplification package

The *Simplification* package contains methods for model reduction and solution simplification. The model reduction is an essential step used in the optimization and simulation procedures to reduce search space by removing reactions which cannot carry flux. This could be crucial to save memory and improve performance when GSMMs are used for different tasks.

3.2.3 The Solvers library

The *Solvers* library provides several generic components (variables, constraints and objectives) that can be combined to formulate any of the methods previously mentioned in the *Simulation* package from the *MEWCore* library. Furthermore, the connection to open source and commercial solvers/optimizers, including GLPK[1], CLP[2] and CPLEX[3], is also provided through the implemented classes such as `GPLPKSolver`, `CLPLPSolver` and `CPLEXSolver`.

[1] http://www.gnu.org/software/gplpk/gplpk.html

[2] https://projects.coin-or.org/Clp

[3] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

Recently, this library has been extended to support a persistent mode which allows keeping the problem formulation over several simulations, changing only specific variables, constraints or objective functions as needed.

3.3 OptFlux framework

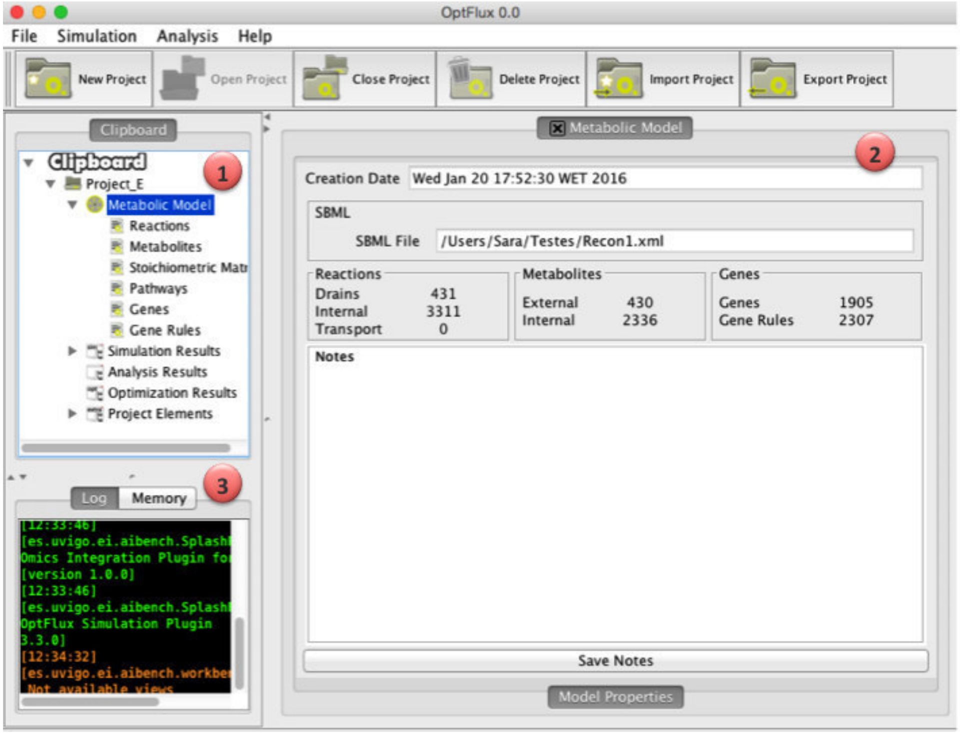
OptFlux [24] is a software framework to support *in silico* constraint-based modeling approaches, mainly for metabolic engineering tasks, which aims to be the reference platform for this community. The methods, algorithms and features implemented on the MEW framework can be used by users without any computer science skills through this user-friendly software tool.

OptFlux is a modular user friendly software based on a plug-in architecture built on top of AIBench [105], which facilitates the addition of new features by software developers. The main window of the application, shown in Figure 3.8, can be divided in three areas: the clipboard, the data viewing panels and a logging area.

All objects created inside the application are associated to a global entity named *Project*, which is always connected to a metabolic model. The user can have multiple projects in the clipboard, however the input data to any operation always belongs to the same project, as well as the corresponding results.

The OptFlux framework is composed by several plug-ins, being the most important:

- **Core:** responsible for creating the project, loading the metabolic models and creating the data types and views to manipulate the model information, such as reactions, genes, metabolites, etc..
- **Model Repository:** adds a new model reader, which adds a repository of validated models by OptFlux's team.



Creation Date Wed Jan 20 17:52:30 WET 2016

SBML

SBML File /Users/Sara/Testes/Recon1.xml

Reactions		Metabolites		Genes	
Drains	431	External	430	Genes	1905
Internal	3311	Internal	2336	Gene Rules	2307
Transport	0				

Notes

```
[12:33:46]
[es.uvigo.ei.aibench.Splash
Omics Integration Plugin fo
[version 1.0.0]
[12:33:46]
[es.uvigo.ei.aibench.Splash
OptFlux Simulation Plugin
3.3.0]
[12:34:32]
[es.uvigo.ei.aibench.workbe
 Not available views
```

Figure 3.8: The OptFlux main window is segmented in three areas: 1 - the clipboard, where the operations and results are showed; 2- the view area, where objecta are visualized; 3- log window or memory monitor, depending on the selected tab.

- **Simulation:** allows the user to perform phenotype simulation of ”wild-type” and mutant strains, predicting the effects of knockout reactions or genes or of over/under expression of genes or reactions.

- **Optimization:** includes single objective and multi-objective optimization methods based on Evolutionary Algorithms and other optimization approaches.

- **Visualization:** allows the user to visualize the model (or select pathways) in a graphical manner, also enabling users to import and export layouts. Moreover, this plug-in allows the overlap of simulation results with the model graphs [106].

3.3.1 The AIBench framework

AIBench [105] is a software development framework, based on Model-View-Controller (MVC) design pattern, that provides a powerful programming model allowing the fast development of applications. The AIBench eases the connection, execution and integration of operations with well defined inputs/outputs. It facilitates the development of a wide range of applications based on generic input-process-output cycles, where the framework acts as the glue between each task.

The MVC design pattern divides a given software application into three interconnected parts: model, controller and views. The idea is to make a clear division between the objects that represent the problem (model) which are controlled by operations, and the visualization of objects (views) that are the GUI elements. Figure 3.9 depicts the MVC components and the interactions between them.

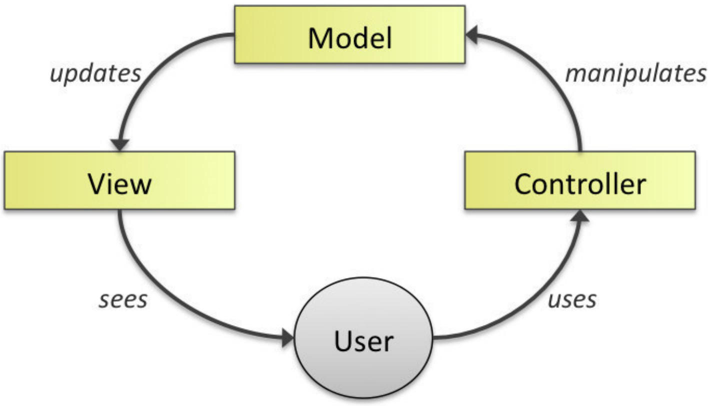


Figure 3.9: A typical collaboration of the MVC components.

In the AIBench framework, these three types of well defined objects are called: *operations, data types* and *views.*

3.3.2 Data types

In Optflux, the *data types* are objects that hold relevant data to the application, such as the models, simulation and optimization results, and are usually an extension of the `AbstractOptFluxDataType`. Each plug-in has its own *datatypes* normally present in the <<`plug-in package`>>`.datatypes` package.

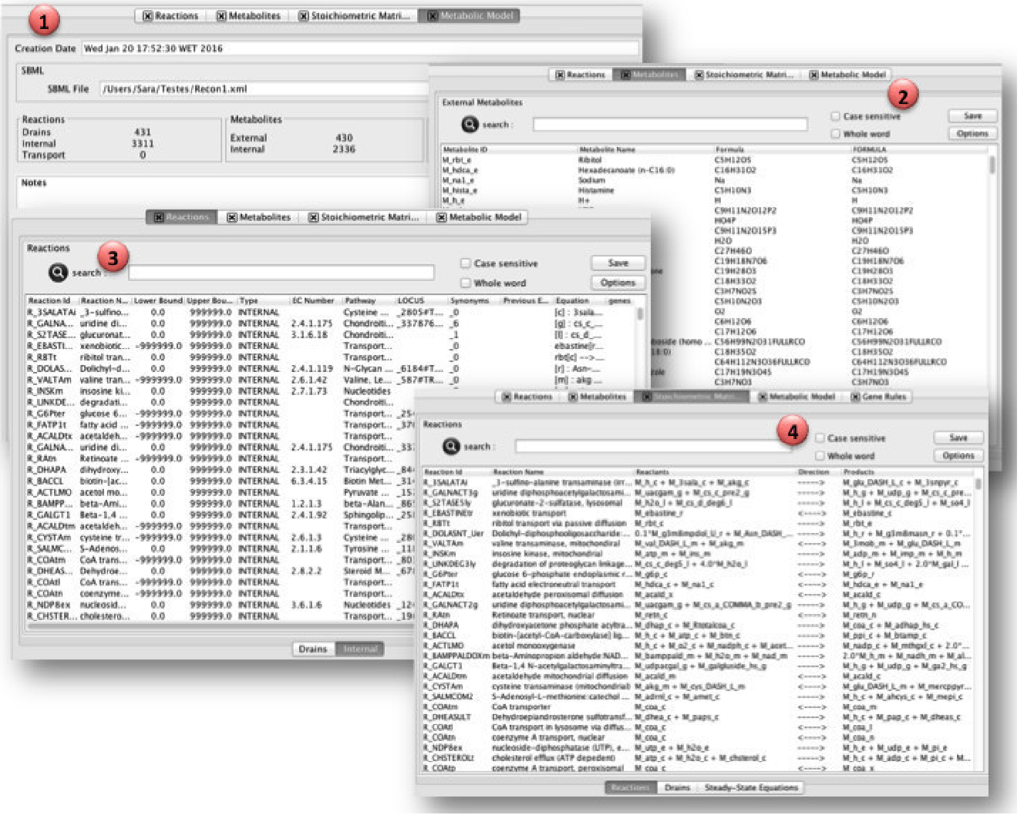
The *datatypes* classes within OptFlux, besides the object information, also have a reference to the project to which the data belongs. Furthermore, these classes are used to encapsulate the data structures present in the MEW framework.

Considering that the metabolic model information in the MEW framework is represented by an implementation of `IModel`, such as `SteadySteateModel`, and the genes, reactions and metabolites by the classes `Gene`, `Reaction` and `Metabolite` respectively, there are data types to encapsulate these data structures. So, the classes `SteadyStateModelBox`, `GeneBox`, `ReactionsBox` and `MetaboliteBox` are data types to encapsulates the MEW classes `SteadySteateModel`, `Gene`, `Reaction` and `Metabolite`.

3.3.3 Views

The *views* enable the output representation of information, being the way to present the *data types* on appropriate GUIs (the same *data types* can have more than one view to visualize their content). The association between data types and the view used to show the information is made on the *plugin.xml* file, available for each plug-in.

The Figure 3.10 presents several *views* for the model information *datatypes*.



Creation Date Wed Jan 20 17:52:30 WET 2016

SBML File /Users/Sara/Testes/Recon1.xml

Reactions	
Drains	431
Internal	3311
Transport	0

Metabolites	
External	430
Internal	2336

Figure 3.10: Views of the metabolic model information in the OptFlux GUI: 1- `MetabolicModelView`, 2 - `MetabolitesExternalView`, 3- texttttReactionsInternalView and 4 -`MatrixView`, to visualize the content of datatypes `SteadyStateModelBox`, `MetabolitesBox`, `ReactionsBox` and `StoichiometricMatrixBox`, respectively.

3.3.4 Operations

The *operations* accept inputs in the form of objects from specific data types, execute commands and generate new objects or update existing ones, from well defined *data types*. The *operations* classes present in each plug-in are, in fact, wrappers to execute the algorithms and procedures from the MEW library. As an example, the operation `NewProjectWizardOperation` is responsible to create a new project, with the correspondent metabolic model data types. Based on the input files, the correct reader present in the *MEWCore* package is called and an instance of the `Container` class is obtained. Next, the

Container is converted to the ModelBox data type and inserted into the clipboard. Moreover, some procedures, such as the biomass reaction definition or the removal of external metabolites present in the model, can be done in this operation.

Considering the "wild-type" phenotype simulation as another operation example, the Figure 3.11 presents the main MVC components and their interactions.

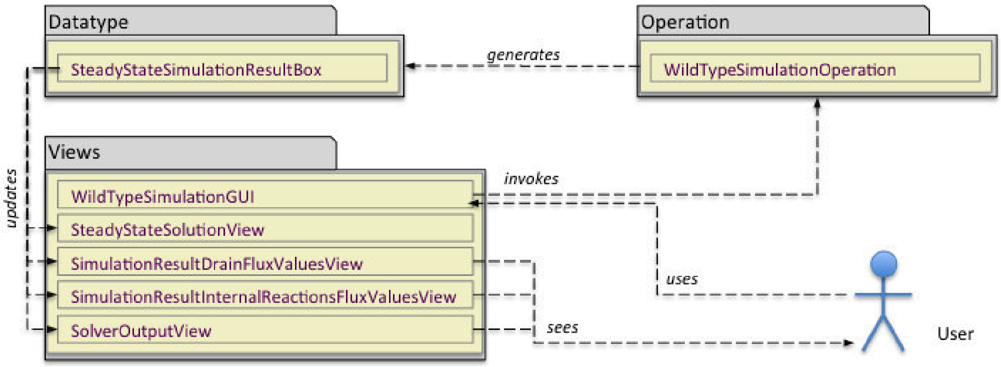

Datatype
SteadyStateSimulationResultBox
generates
Operation
WildTypeSimulationOperation
Views
WildTypeSimulationGUI
SteadyStateSolutionView
SimulationResultDrainFluxValuesView
SimulationResultInternalReactionsFluxValuesView
SolverOutputView
updates
invokes
uses
sees
User

Figure 3.11: Scenario for an user interaction with the *Simulation* plug-in. The three main MVC components are depicted: the user interacts with the WildTypeSimulationGUI (a view) which invokes the WildTypeSimulationOperation (controller). The controller generates the SteadyStateSimulationResultBox (model) which in turn updates the views.

3.4 Omics data integration

The methods for phenotype simulation and for reconstruction of tissue-specific metabolic models require data as input that can be obtained from omics data, such as transcriptomics, metabolomics, proteomics and/or fluxomics. Thus, as a first step it was required the development of methods to import and transform omics data, as well as to integrate them with the metabolic model.

The information loaded from omics data is stored on OmicsContainer instances. This class contains the data characterization, saved as a Condition object (a map of $< property, value >$), a map with score values associated with entities (genes, reaction or metabolites) and external information that can be used to integrate the omics data with the metabolic model.

The reading and integration processes of omics is depicted in Figure 3.12 and encompass four main steps:

1. **Loading:** import the data from the original files to an `OmicsContainer` object.

2. **Data Filtering:** allows the data filtering and identifiers format conversion. This step is optional. The data filtering allows the data selection by using regular expressions over fields of omics data. This can be used to reduce the total amount of imported data and the time consumed. The identifiers conversion is required when the nomenclature used in the omics data and the model is not the same. In this case, it is necessary to have an auxiliary external map ($< id_{model}, id_{omics} >$) with the conversion between nomenclatures.

3. **Data Integration**: the integration of omics data and the metabolic model is done using associations between omics data fields and model fields/associations. Usually, the association is done by the entities identifiers. However, other properties from external information present in the `OmicsContainer` or the entity name can be used. In the end, an instance of the `IOmicsDataMap` class is obtained, where numerical values associated with entities present in the model are stored (score values).

4. **Data Transformation**: this step allows the application of functions over the score values, such as logarithm transformation, conversion from gene to reaction scores through the GPR associations present in the metabolic model and scale the values to a specific range.

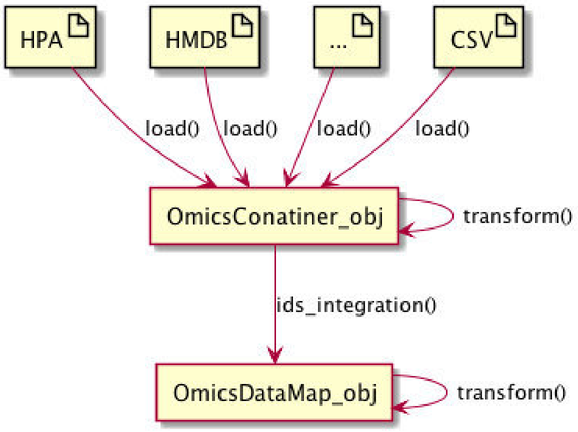


Figure 3.12: Illustration of the data loading and integration processes.The four main steps are: 1- to load original data to an `OmicsContainer` object; 2 - to apply transformation methods to reduce the total amount of data and/or to convert the format of omics identifiers; 3 - to integrate the omics data with the metabolic model; 4 - methods to transform the score values present in the `OmicsDataMap` object.

3.4.1 Data loading

Different omics data sources can be loaded and used in the *Omics* framework. At the moment, readers to specific file formats, such as metabolites data files from HMDB and protein expression levels from HPA are supported. Additionally, data present in CSV files can also be loaded through a generic and flexible reader, named `CSVOmicsReader`. Using the generic CSV reader, the user must specify the column indexes of the identifier and of the numerical values. The other fields are imported as external information to the `OmicsContainer` object.

The implemented Java classes from the Omics framework to load data files to an `OmicsContainer` object are in `omicsintegration.io` package and implement the interface `IOmicsReader` (Figure 3.13).

In the data reading process it is expected that the values associated with each entity (metabolite, gene our reaction) are numerical. Otherwise, the user must provide an additional mapping structure $Map < String, Double >$ to convert each discrete level to a numerical value. At the end, an instance of

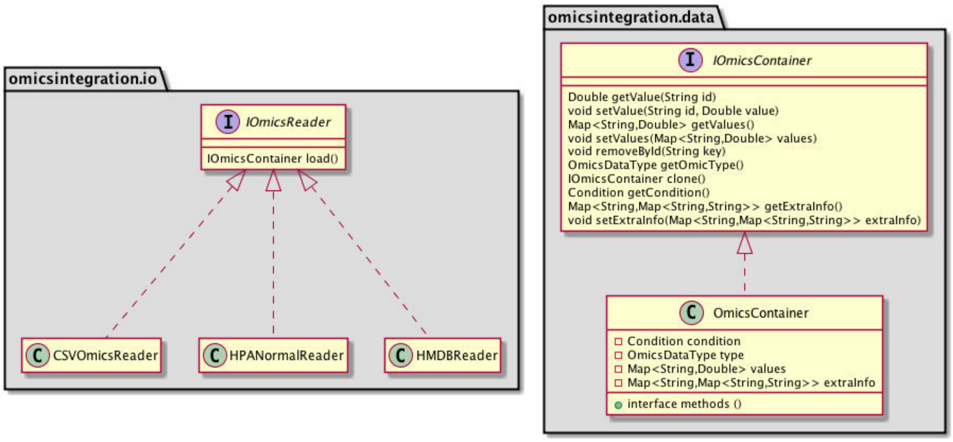

omicsintegration.io
I IOmicsReader
IOmicsContainer load()
C CSVOmicsReader
C HPANormalReader
C HMDBReader
omicsintegration.data
I IOmicsContainer
Double getValue(String id)
void setValue(String id, Double value)
Map<String,Double> getValues()
void setValues(Map<String,Double> values)
void removeById(String key)
OmicsDataType getOmicType()
IOmicsContainer clone()
Condition getCondition()
Map<String,Map<String,String>> getExtraInfo()
void setExtraInfo(Map<String,Map<String,String>> extraInfo)
C OmicsContainer
□ Condition condition
□ OmicsDataType type
□ Map<String,Double> values
□ Map<String,Map<String,String>> extraInfo
● interface methods ()

Figure 3.13: Classes involved in reading omics data. Only the main methods and variables are shown in the diagram. All readers implement the `IOmicsReader` interface and its main method, `load`. As a result, is created an instance of the `OmicsConatiner` class is created with the omics data information.

the class `OmicsContainer` is created.

The `OmicsContainer` class contains the essential information from the omics data, mainly:

- **condition:** contains the information related to the description of the condition/sample, such as the tissue name, the cell type, stage of disease or any other properties used to characterize the data;
- **type:** identifies the omic data type: transcriptomics, fluxomics, metabolomics or proteomics;
- **values:** a map with the structure $Map < String, Double >$, where the *key* is the entity identifier and the value represents the gene/protein expression level, the concentration of a metabolite, the reaction flux or the presence/absent of the entity;
- **extrainfo:** a map with external information which can be used in the integration with the metabolic model. For instance, when the identifiers from the model and omics data do not follow the same nomenclature,

it is necessary to have other fields, such as KEGG or ChEBI identifiers, for metabolomics data, to allow the connection between the model and omics data entities.

3.4.2 Integration with the metabolic model

The integration of omics data with the metabolic model has the objective of setting the identifiers from `OmicsContainer` with the nomenclature used in the metabolic model for the same entities (gene, reaction or metabolite).

This process takes as input an `OmicsContainer` object and converts it into one of the following classes: `GeneDataMap`, `ReactionDataMap`, `MetaboliteDataMap`. These classes implement the interface `IOmicsDataMap` and represent transcriptomics, fluxomics and metabolomics data, respectively. Furthermore, after the integration, only the identifiers present in the metabolic model are retained in the omics data structure. Figure 3.14 shows the class diagram with the main classes employed in the integration process.

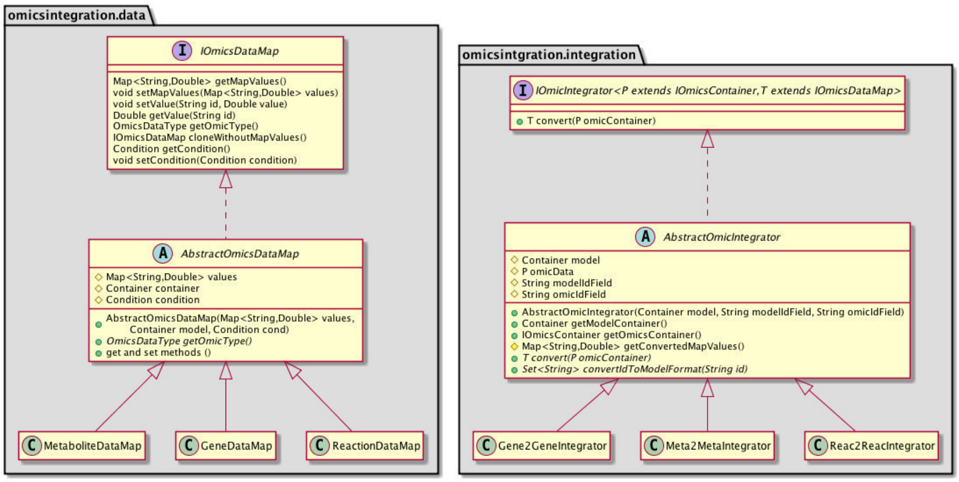

omicsintegration.data
I IOmicsDataMap
Map<String,Double> getMapValues()
void setMapValues(Map<String,Double> values)
void setValue(String id, Double value)
Double getValue(String id)
OmicsDataType getOmicType()
IOmicsDataMap cloneWithoutMapValues()
Condition getCondition()
void setCondition(Condition condition)
A AbstractOmicsDataMap
◇ Map<String,Double> values
◇ Container container
◇ Condition condition
● AbstractOmicsDataMap(Map<String,Double> values, Container model, Condition cond)
● OmicsDataType getOmicType()
● get and set methods ()
C MetaboliteDataMap
C GeneDataMap
C ReactionDataMap
omicsintgration.integration
I IOmicIntegrator<P extends IOmicsContainer,T extends IOmicsDataMap>
● T convert(P omicContainer)
A AbstractOmicIntegrator
◇ Container model
◇ P omicData
◇ String modelIdField
◇ String omicIdField
● AbstractOmicIntegrator(Container model, String modelIdField, String omicIdField)
● Container getModelContainer()
● IOmicsContainer getOmicsContainer()
◇ Map<String,Double> getConvertedMapValues()
● T convert(P omicContainer)
● Set<String> convertIdToModelFormat(String id)
C Gene2GeneIntegrator
C Meta2MetaIntegrator
C Reac2ReacIntegrator

Figure 3.14: Class diagram of main classes used in the integration of omics data with the metabolic model.

Each omics data type has it own integrator class which is responsible for implementing the `convert` method from the interface `IOmicsIntegrator`.

The integration can be made through different information fields, such as the identifier, the name or from external information data structures. Thus, if the model integration field, *modelIdField*, contains the value ”ID” or ”NAME”, the integration is made using the identifier or entity name from the metabolic model, otherwise it is assumed that the field is present in the extra information structure from the `Container` object. Similarly, the omics integration field, `omicIdField`, specifies the used field from the `OmicsContainer`.

During the integration, it is possible to reach more than one value from omics data for the same entity on the metabolic model. This happens, for instance, if omics data identifiers come from transcript sequences instead of genes, because several transcripts can be associated with the same gene. In this case, the maximum value from the transcripts is assumed to be the score value associated to the corresponding gene present in the metabolic model. Another special case occurs when the same omics data entry ($< id, value >$) is associated with more than one identifier in the model. Here, the omics value is replicated for all matching entities from the metabolic model.

3.4.3 Transformation methods

There are two categories of omics transformation methods in the developed framework. The first contains methods over `OmicsContainer` objects to allow filtering data by using regular expressions over omics data fields and to convert the identifiers to a new format based on a given map with entries in the form: $[old_{id} - > new_{id}]$. These two methods are implemented in the `omicsintegration.transform` package under the class names: `TransformOmicsFilter` and `TransformOmicsKeys`. All the transformation classes applied to `OmicsContainer` objects must implement the interface `ITransformOmics`.

The other contains transformations over the `OmicsDataMap` objects to allow the values scaling or the conversion of gene to reaction scores under the GPR associations. In the last method, the operators AND/OR present in the GPR associations are, usually, replaced by the functions Minimum/Maximum.

Thus, if a reaction is regulated by the gene rule "$gene_1\ and\ gene_2$" the reaction score value will be the lower score among these genes. However, it is possible to specify different functions to be applied in the transformation process.

The class `FactoryTransformDataMap` is responsible, in run time, to create a transformation class instance based on the transformation type selected by the user.

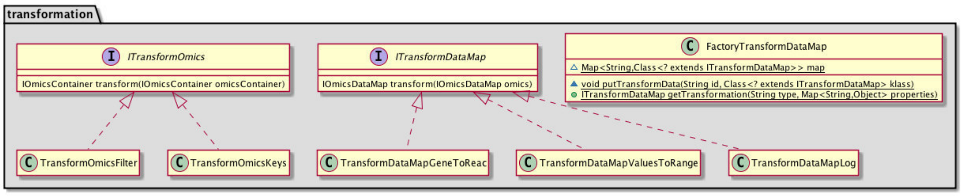

transformation
ITransformOmics
IOmicsContainer transform(IOmicsContainer omicsContainer)
ITransformDataMap
IOmicsDataMap transform(IOmicsDataMap omics)
FactoryTransformDataMap
Map<String,Class<? extends ITransformDataMap>> map
void putTransformData(String id, Class<? extends ITransformDataMap> klass)
ITransformDataMap getTransformation(String type, Map<String,Object> properties)
TransformOmicsFilter
TransformOmicsKeys
TransformDataMapGeneToReac
TransformDataMapValuesToRange
TransformDataMapLog

Figure 3.15: Diagram of classes implementing the transformation methods available in the framework.

Following this class structure, it is easy to develop new transformation methods, only being required the implementation of an interface method in the `ITransformDataMap` class and the registration of the new transformation class in the factory `FactoryTransformDataMap`.

3.5 Simulation methods

The simulation methods allow the phenotype prediction (flux values for the reactions), using omics data to improve the results over traditional constraint-based methods. The objective is to find a flux distribution, where an objective function is defined considering the omics data as a guide for the distribution. The three methods discussed in chapter 2, mainly the E-Flux, IMAT and GIMME , are available in the `omicsintegration.omicssimulation` package from the *Omics* framework.

The class diagram of the implemented algorithms and the connection to the MEW classes are shown on Figure 3.16.

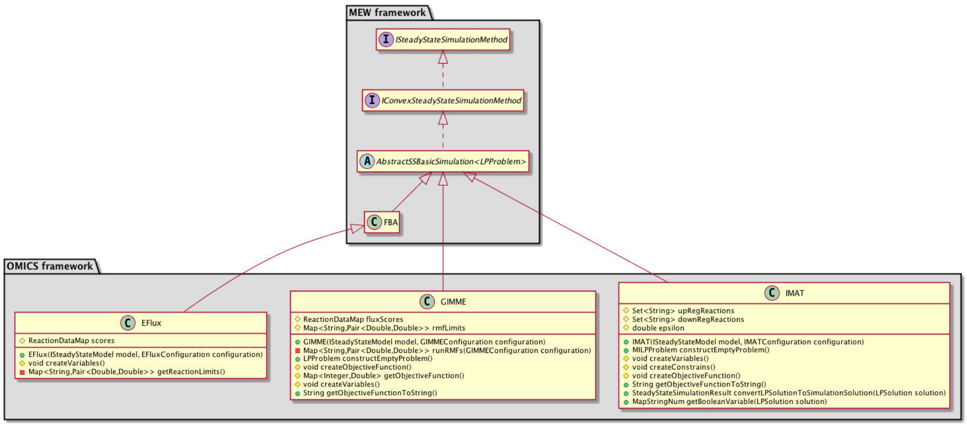

MEW framework
I ISteadyStateSimulationMethod
I IConvexSteadyStateSimulationMethod
A AbstractSSBasicSimulation<LPProblem>
C FBA
OMICS framework
C EFlux
◇ ReactionDataMap scores
● EFlux(ISteadyStateModel model, EFluxConfiguration configuration)
◆ void createVariables()
■ Map<String,Pair<Double,Double>> getReactionLimits()
C GIMME
◇ ReactionDataMap fluxScores
◇ Map<String,Pair<Double,Double>> rmfLimits
● GIMME(ISteadyStateModel model, GIMMEConfiguration configuration)
■ Map<String,Pair<Double,Double>> runRMFs(GIMMEConfiguration configuration)
● LPProblem constructEmptyProblem()
◆ void createObjectiveFunction()
◆ Map<Integer,Double> getObjectiveFunction()
◆ void createVariables()
● String getObjectiveFunctionToString()
C IMAT
◇ Set<String> upRegReactions
◇ Set<String> downRegReactions
◇ double epsilon
● IMAT(ISteadyStateModel model, IMATConfiguration configuration)
● MILPProblem constructEmptyProblem()
◆ void createVariables()
◆ void createConstrains()
◆ void createObjectiveFunction()
● String getObjectiveFunctionToString()
● SteadyStateSimulationResult convertLPSolutionToSimulationSolution(LPSolution solution)
● MapStringNum getBooleanVariable(LPSolution solution)

Figure 3.16: Class diagram of simulation methods with omics integration.

All these methods are made available through the Omics plug-in discussed in the following section 3.7. To simplify the instantiation of the methods, a factory class was implemented, named `FactoryOmicsSimulationMethods`. This class is responsible for returning the instance of the simulation class according to the method chosen by the user in run time. This is used to avoid the replication of code and to simplify the addition of new methods to the framework.

In the implementation of these methods, we separate the omics data processing from the algorithm. This means that all the algorithms accept as input a `ReactionDataMap` as omics data source. The transformation from gene to reaction scores is done using the methods presented in the previous section. This separation in two layers allows to use several omics data sources for each algorithm.

Each class implementing a simulation method is an extension of the class `AbstractSSBasicSimulation<T extends LPProblem>`, which is responsible for creating the simulation problem that will run in the solver.

The configuration to run each method is stored in a specific class: `EFluxConfiguration`, `IMATConfiguration` and `GIMMEConfiguration`. All these classes are extensions of the class `GenericOmicsConfiguration`, which contains the basic information required to run phenotype simulation with omics data integration. In detail, this class has the metabolic model, the omics data and the generic configuration properties, such as the solver type and the environmental conditions to use in the phenotype prediction. Additionally, the configuration class of each method has specific properties used by the algorithm. The hierarchical structure of these classes is shown in Figure 3.17.

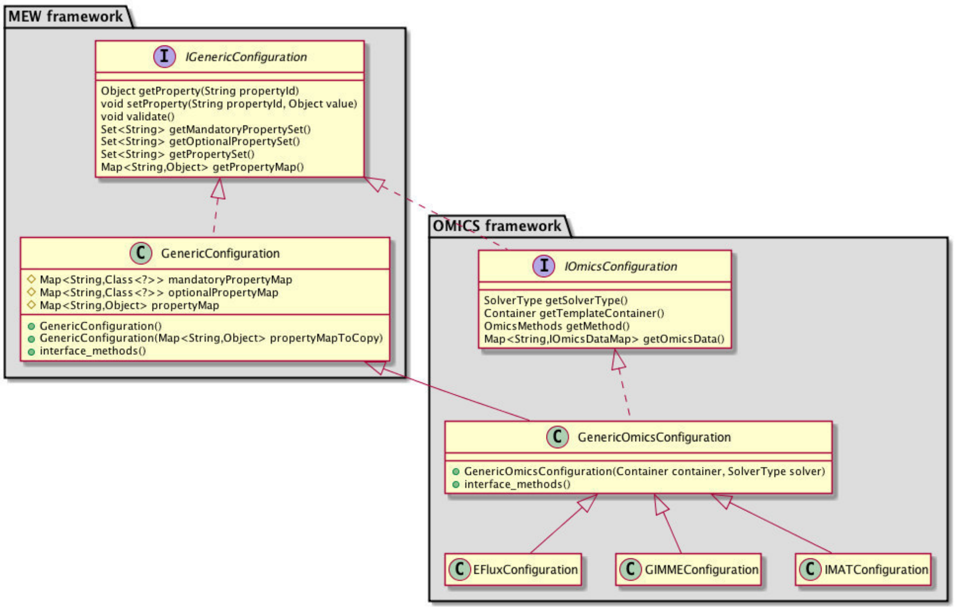

MEW framework
I IGenericConfiguration
Object getProperty(String propertyId)
void setProperty(String propertyId, Object value)
void validate()
Set<String> getMandatoryPropertySet()
Set<String> getOptionalPropertySet()
Set<String> getPropertySet()
Map<String,Object> getPropertyMap()
C GenericConfiguration
◇ Map<String,Class<?>> mandatoryPropertyMap
◇ Map<String,Class<?>> optionalPropertyMap
◇ Map<String,Object> propertyMap
● GenericConfiguration()
● GenericConfiguration(Map<String,Object> propertyMapToCopy)
● interface_methods()
OMICS framework
I IOmicsConfiguration
SolverType getSolverType()
Container getTemplateContainer()
OmicsMethods getMethod()
Map<String,IOmicsDataMap> getOmicsData()
C GenericOmicsConfiguration
● GenericOmicsConfiguration(Container container, SolverType solver)
● interface_methods()
C EFluxConfiguration
C GIMMEConfiguration
C IMATConfiguration

Figure 3.17: Class diagram for the configuration classes used in the phenotype simulation methods of the Omics framework.

The constructor of each simulation method takes as input an instance of the corresponding configuration class, since all the required information is there.

3.5.1 iMAT

The `IMATConfiguration` class holds the configuration to run a phenotype simulation using the iMAT algorithm, mainly up and down regulated reactions sets used and the parameter ϵ value (with 1.0 by default).

The iMAT algorithm is implemented in the class `IMAT`, where the extension of the basic problem created by `AbstractSSBasicSimulation` is done. Here, the methods `createVariables, createConstrains` and `createObjectiveFunction`, are overridden to create the additional variables and constraints for the iMAT formulation and to define the objective function.

3.5.2 GIMME

The `GIMMEConfiguration` class supports two different ways to get the limits of reactions associated with the Required Metabolic Functionalities (RMFs) in the GIMME algorithm:

1. by using a set of RMFs to constrain the reaction of each RMF to a percentage of the maximum possible flux calculated by FBA. In this case, the reaction limits for each RMF are calculated inside the GIMME algorithm through the `runRMFs` function. The lower and upper bounds are populated by a percentage of the maximum flux value obtained by FBA and the flux value itself, respectively.
2. using a `ReactionDataMap`, which contains the maximum flux reaction associated with the RMFs. The reaction lower bound is calculated as a product of the maximum flux and the configuration property *RMF_Percentage*.

The GIMME algorithm class, `GIMME`, overrides the method `createVariables` where the constraints of RMFs associated reactions are changed to the lower and upper bounds obtained by one of the previous described ways.

3.5.3 E-Flux

The E-Flux method is an extension of the FBA formulation being only required the update of the reaction constraints. These reactions constraints, given as argument to the method, are obtained by transforming the transcpriptomic data (scores associated to genes) to reaction scores through the GPR associations.

In the class `EFlux`, the reaction scores received as input are normalized by dividing each reaction score by the maximum of all scores. This normalization converts all scores to values between 0 and 1. Next, the reaction constraints are updated to set the lower and upper bounds to the normalized score. Reactions without associated score in the input data will be constrained with the upper bound of 1 and lower bound of -1, or 0 if the reaction is irreversible.

The external exchange reactions have the lower and uppers bounds as -1 and 1, respectively. However, when a reaction is only for uptake/secretion the upper bound/lower is changed to 0.

3.6 Tissue-specific reconstruction methods

Besides phenotype prediction methods with omics data integration, the framework also has methods to reconstruct tissue-specific metabolic models. In summary, these methods use a generic model as template and evidences provided by omics data and literature to reconstruct a context-specific model, and were explored in detail in chapter 2, section 2.6.

In the framework, four methods were implemented: MBA, mCADRE, tINIT and FASTCORE. Similarly to the simulation method's classes, the constructor receives a configuration object which contains all parameters and data used by the algorithm. These four configuration classes were implemented as an extension of `GenericOmicsConfiguration`.

Again, the layer of omics data processing is independent of the method itself. Actually, all the methods expect a `ReactionDataMap` instance as main input in the configuration object. Therefore, the processing of omics data and conversion to a `ReactionDataMap` object is done using the methods presented in section 3.4. This layer division allows us to use different omics data types with each implemented method.

Each one of the methods was implemented in a class, named from the method's name. These classes are an extension of the `AbstractReconstructionAlgorithm` which implements the interface `ISpecificModelReconstruction`. The hierarchical diagram of classes is depicted in Figure 3.18.

The result of the reconstruction process is an instance of the class `SpecificModelResult` returned by the `generateSpecificModel` method present in each class that implements a specific algorithm.

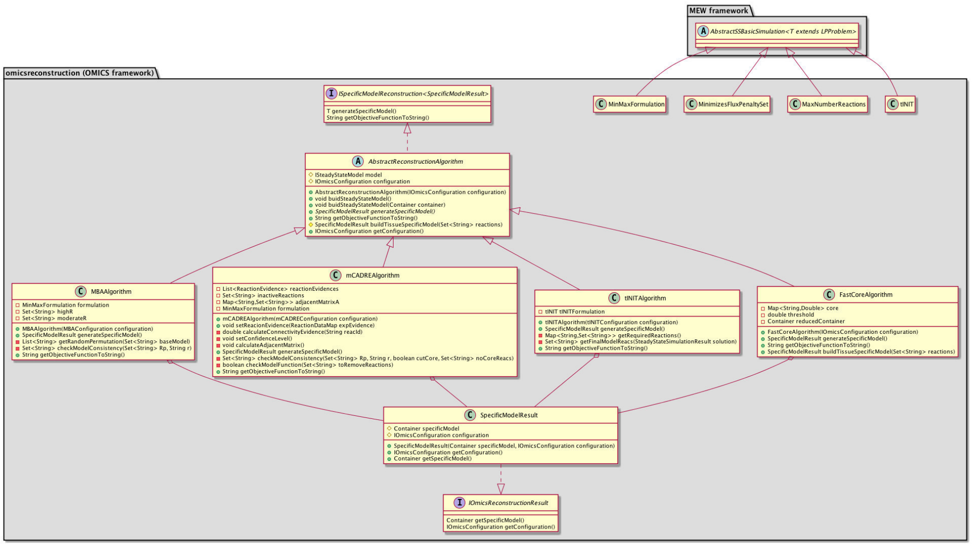

MEW framework
AbstractSSBasicSimulation<T extends LPProblem>
omicsreconstruction (OMICS framework)
MinMaxFormulation
MinimizesFluxPenaltySet
MaxNumberReactions
tINIT
ISpecificModelReconstruction<SpecificModelResult>
T generateSpecificModel()
String getObjectiveFunctionToString()
AbstractReconstructionAlgorithm
ISteadyStateModel model
IOmicsConfiguration configuration
AbstractReconstructionAlgorithm(IOmicsConfiguration configuration)
void buidSteadyStateModel()
void buidSteadyStateModel(Container container)
SpecificModelResult generateSpecificModel()
String getObjectiveFunctionToString()
SpecificModelResult buildTissueSpecificModel(Set<String> reactions)
IOmicsConfiguration getConfiguration()
MBAAlgorithm
MinMaxFormulation formulation
Set<String> highR
Set<String> moderateR
MBAAlgorithm(MBAConfiguration configuration)
SpecificModelResult generateSpecificModel()
List<String> getRandomPermutation(Set<String> baseModel)
Set<String> checkModelConsistency(Set<String> Rp, String r)
String getObjectiveFunctionToString()
mCADREAlgorithm
List<ReactionEvidence> reactionEvidences
Set<String> inactiveReactions
Map<String,Set<String>> adjacentMatrixA
MinMaxFormulation formulation
mCADREAlgorithm(mCADREConfiguration configuration)
void setReacionEvidence(ReactionDataMap expEvidence)
double calculateConnectivityEvidence(String reacId)
void setConfidenceLevel()
void calculateAdjacentMatrix()
SpecificModelResult generateSpecificModel()
Set<String> checkModelConsistency(Set<String> Rp, String r, boolean cutCore, Set<String> noCoreReacs)
boolean checkModelFunction(Set<String> toRemoveReactions)
String getObjectiveFunctionToString()
tINITAlgorithm
tINIT tINITFormulation
tINITAlgorithm(tINITConfiguration configuration)
SpecificModelResult generateSpecificModel()
Map<String,Set<String>> getRequiredReactions()
Set<String> getFinalModelReacs(SteadyStateSimulationResult solution)
String getObjectiveFunctionToString()
FastCoreAlgorithm
Map<String,Double> core
double threshold
Container reducedContainer
FastCoreAlgorithm(IOmicsConfiguration configuration)
SpecificModelResult generateSpecificModel()
String getObjectiveFunctionToString()
SpecificModelResult buildTissueSpecificModel(Set<String> reactions)
SpecificModelResult
Container specificModel
IOmicsConfiguration configuration
SpecificModelResult(Container specificModel, IOmicsConfiguration configuration)
IOmicsConfiguration getConfiguration()
Container getSpecificModel()
IOmicsReconstructionResult
Container getSpecificModel()
IOmicsConfiguration getConfiguration()

Figure 3.18: Class diagram of reconstruction of tissue-specific methods.

The assumptions taken during the implementation of reconstruction methods on this framework will be detailed in the next subsections.

3.6.1 tINIT

The tINIT method has two main steps: first, find out the set of essential reactions to perform the metabolic tasks passed as input; second, simulate the formulation problem described in section 2.6, which is implemented in the `tINIT` class, where the reactions found in previous steps are constrained to have flux.

To perform the first step, additional methods were developed to load metabolic tasks and find the reactions that can not be removed from the model to be able to perform such tasks. The metabolic tasks can be loaded by the `TasksReader` class from a CSV file with the following structure:

```
COM;ID;DESCRIPTION;SHOULD_FAIL;IN;IN_LB;IN_UB;OUT;OUT_LB;OUT_UB;EQU;EQU_LB;EQU_UB;OBJ_REAC
;id1; description;;M_A_e,M_B_c,M_C_e;-1000,-1000,-1000;0,0,0;M_Z_e;0;1000;;;;M_X_e
```

In the task above, the metabolic task id_1, must produce the metabolite M_X, when the drains are open for the excretion of M_Z and allowing uptake of M_A and M_C. Moreover, it is assumed that metabolite M_B can be produced by the cell in the cytosol compartment.

This file can be constructed based on metabolite or reaction entities. The entity references present in the file must be one of the following: identifiers, names or any other field present in the external information from the template model object, `Container`. However, the match between model entities and tasks must be perfect, otherwise an exception will be thrown during the reading process, telling that the metabolite/reaction does not exist in the metabolic model.

The simulation of each metabolic model is done by the `CheckTasks` class. First, the model is reduced to contain only the reactions that can carry flux. Next, the drains present in the metabolic task are constrained to the values present there, while the others are closed by setting the lower and upper bounds to 0. Furthermore, for the internal metabolites, which are assumed to have production, new artificial drains will be inserted in the model to uptake such metabolites. In the previous example, a new drain to uptake M_B_c will be added to the model.

Finally, an FBA with the maximization of the target metabolite, M_X_e, is performed considering the changed model and the constraints imposed to the drains reaction. Based on the result, we simulate the knockout of each reaction that can carry, using FBA, to verify if the knockout reaction is essential or not to satisfy the task. In the end, the set of essential reactions to satisfy the metabolic task is returned.

This set of reactions is afterwards used by the tINIT algorithm by adding new constraints to the reconstruction problem, which guarantees that these reactions must have a positive flux in the final result.

3.6.2 MBA

The MBA algorithm receives two reaction sets as input (C_H and CM). Once again, this information is given to the algorithm under a `MBAConfiguration` instance. Ideally, the final tissue-specific model is built from a significant number of models obtained by running several times the MBA algorithm. Thus, after the construction of each model through the `generateSpecificModel` function present in the `MBAAlgorithm` class, the final model is created through the static function `getFinalModel`. This function takes as input the template metabolic model, the core reactions set and the path of all files with the reaction identifiers of the tissue-specific models. At the end, a set with the reaction identifiers present in the final consensus model are returned.

3.6.3 mCADRE

The `mCADREConfiguration` class contains the reaction scores, the confidence levels and a set of metabolic tasks. These tasks contain the set of metabolites that should be produced during the pruning process. This production is checked on function `checkModelFunction` on the `mCADRE` class.

When the algorithm tries to remove a set of reactions from the model, the production of essential metabolites is tested by adding a fake drain to the model and a simulation is performed to check if the drain has flux excretion. This simulation is done through the `MinMax` formulation problem, where the objective function is the maximization of a constant value and the new drain is constrained to carry a flux larger than 10^{-4}. If the problem has a feasible solution, then it is possible to have metabolite production in the model after the removal of a set of reactions. Alternatively, this step could be done using the FBA with the maximization of the metabolite excretion, however this would be time consuming.

3.6.4 FASTCORE

The FASTCORE algorithm was implemented using 3 main classes:

- `FastCoreAlgorithm`: this is the main class of the method and implements the algorithm presented in the Section 2.6.4;
- `MaxNumberReactions`: implements the LP formulation to find the larger set of reactions from a given set with a positive flux rate;
- `MinimizesFluxPenaltySet`: implements the LP formulation to find the smaller set of reactions with flux from the penalty set, when a given set of reactions must have flux.

These two last classes implement the formulation problems, named LP7 and LP10, respectively in the original paper [22]. Moreover, these classes are an extension of the `AbstractSSBasicSimulation<T extends LPProblem>` class from the MEW framework.

3.6.5 Running tissue-specific model reconstruction

In order to easily use the framework by non-programmers, a class to launch the tissue-specific metabolic model reconstruction methods was implemented, named `GenerateModels`. This class allows to run each method for different omics data such as : HPA, GEB and two sets of reactions (core and moderate). The configuration of all inputs required to build the tissue-specific models is done through a text file where the following fields must be populated:

- **ModelSBMLFile:** path to the SBML template metabolic model;

- **BiomassReaction:** biomass reaction identifier (null if no biomass reaction exists on the model);

- **HPAFile:** path to a CSV data file with HPA protein expression levels. This file must have two columns, the first with the gene identifiers and the other with the expression levels;

- **CHFile and CMFile:** path to files with the core and moderate reaction sets;

- **BarcodeFile:** file with the gene identifiers and the probability to be active or not in the tissue. These data can be obtained from the GEB website and must be converted to a map with the entities in the form $[Gene_{Id} - > score]$ using, for instance, the Bioconductor annotation package;

- **ConfLevelScores, TaskFile:** information used in the tINIT method. The task file must have the structure presented above (section 3.6.1), where the header must have the same order and name fields;

- **CellLine:** identification of tissue , cell or context;

- **convertGeneIdsFile:** file to convert gene identifiers from omics data to model format;

- **ResultsPath:** path where the final metabolic models will be stored;

- **Method:** one of the methods available in the framework. The field must have the value “tINIT” , “mCADRE”, “MBA” or “FASTCORE”;

- **OmicData:** one of these omics data types: “HPA”, “Barcod”, “Sets”;

- **CutOff1 and CutOff2:** values to build the core and moderate sets. The core set will be composed with reactions with a score higher than CutOff1 and the moderate set encompasses the reactions with score values between the two cut offs.

- **tINITCuttOff:** values to use in the tINIT algorithm to create the five expression levels.

Using this class to reconstruct the tissue-specific model there are some rules that are assumed by the `GenerateModel` class, such as:

- The HPA expression levels, "High", "Medium", "Low" and "Not detected" are always converted to the integer values $20, 15, 10$ and -8, respectively;
- The key metabolites used on mCADRE method are the same as published in the original paper [21] as metabolites that should be produced in all tissues cells.

3.7 Omics plug-in

The need to develop the software tools for regular users led us to design and implement new plug-ins, to support the developments present in the previous sections, within OptFlux, thus making the most of pre-existing tools. Thus, two new plug-ins were created, the *Omics* and the *OmicsSimulation* plug-ins.

3.7.1 Implementation

The Omics plug-in was developed to support all the methods presented in section 3.4 which include the reading, integration and the transformation of omics data processes. Following the MVC design pattern, several new components were created, from which the most relevant will be detailed.

The `ImportOmicsWizard` class and its related classes from the `importomicswizard` package, provide a set of dialogs for the `ImportOmicsWizardOperation`, which allows the user to load the omics data from data files and perform some transformations over the original data, such as the conversion of entity identifiers and discrete levels to numerical values and the integration with the metabolic model.

The addition of new readers is easily implemented. The new reader must be an extension of the `AbstractOmicsReader` and implement the following two methods:

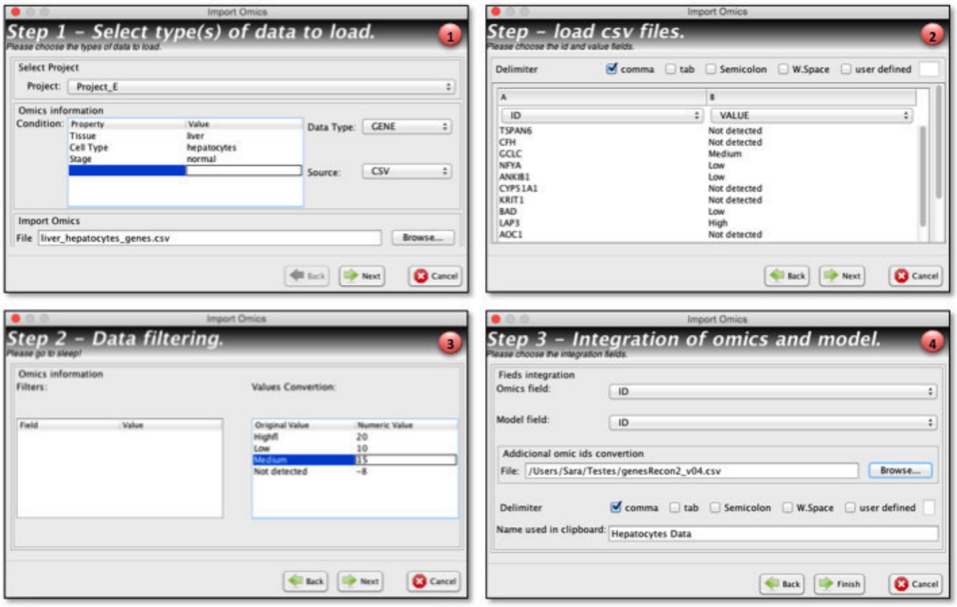
- `needsConfiguration:` this function must return true if the reader requires a specific configuration panel;
- `getConfigurationPanel:` returns the class which extends `AbstractWizardConfigurationPanel` and contains the specific configuration panel.

After the reading process completes, the results are placed in the OptFlux clipboard under the corresponding omics type folder (Gene, Reaction or Metabolite). Moreover, the result can be an instance of one of the following data types: `GeneBox`, `MetaboliteBox` or `ReactionBox`. All these classes are an extension of the `OmicsBox` datatype, and its content can be visualized through the `OmicsView`.

The omics data objects available in the clipboard can be used to apply a transformation over the score values or convert the gene scores to reaction scores through the menu option *Transform omics*. Next, the GUI `TransformOmicsDataGUI` is presented and the user can select the configuration parameters required for the transformation. After the transformation operation, `TransformDataMapOperation`, a new data object will be added to the clipboard.

3.7.2 Functionalities

The steps to load a CSV file are shown in Figure 3.19. The step 2 only appears when the data is imported from a generic CSV file.



Import Omics

Step 1 – Select type(s) of data to load.

Please choose the types of data to load.

Select Project

Project: Project_E

Omics information

Condition:

Property	Value
Tissue	liver
Cell Type	hepatocytes
Stage	normal

Data Type: GENE

Source: CSV

Import Omics

File liver_hepatocytes_genes.csv

Import Omics

Step – load csv files.

Please choose the id and value fields.

Delimiter ☑ comma □ tab □ Semicolon □ W.Space □ user defined

A	B
ID	VALUE
TSPAN6	Not detected
CFH	Not detected
GCLC	Medium
NFYA	Low
ANKIB1	Low
CYP51A1	Not detected
KRIT1	Not detected
BAD	Low
LAP3	High
AOC1	Not detected

Import Omics

Step 2 – Data filtering.

Please go to sleep!

Omics information

Filters:

Field	Value

Values Convertion:

Original Value	Numeric Value
Highfl	20
Low	10
Medium	15
Not detected	-8

Import Omics

Step 3 – Integration of omics and model.

Please choose the integration fields.

Fieds integration

Omics field: ID

Model field: ID

Addicional omic ids convertion

File: /Users/Sara/Testes/genesRecon2_v04.csv

Delimiter ☑ comma □ tab □ Semicolon □ W.Space □ user defined

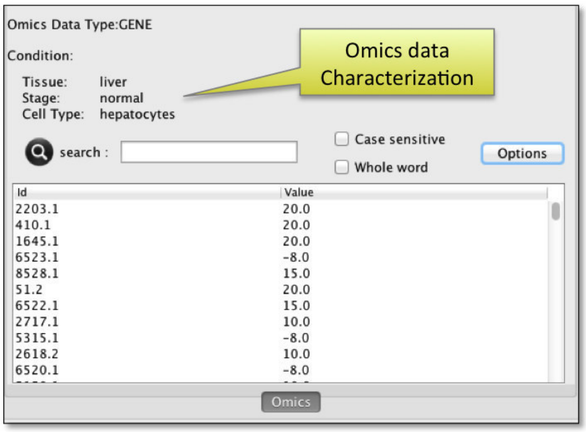
Name used in clipboard: Hepatocytes Data

Figure 3.19: Dialogs to load and integrate omics data with the metabolic model. 1- select the data source and set experiment properties such as tissue and cell type. 2 - choose the identifier and values columns, other fields can be imported as external information. 3 - convert the expression level to numeric values; 4 - choose the fields that will be used to do the integration between omics data and metabolic model. An additional conversion of identifiers can also be set as an external file.

The view depicted in Figure 3.20 presents the experimental conditions to help in data characterization and the score values associated to each entity identifier, in this case the gene identifier.

3.8 Omics simulation plug-in

The simulation plug-in was developed to support the phenotype simulation methods presented in Section 3.5.



Omics Data Type:GENE

Condition:

Tissue: liver
Stage: normal
Cell Type: hepatocytes

Id	Value
2203.1	20.0
410.1	20.0
1645.1	20.0
6523.1	-8.0
8528.1	15.0
51.2	20.0
6522.1	15.0
2717.1	10.0
5315.1	-8.0
2618.2	10.0
6520.1	-8.0

Figure 3.20: Omics data View.

3.8.1 Implementation

The configuration panel contains the parameters used in the configuration of each algorithm. In iMAT, for instance, the user must set the ReactionOmicsBox that will be used as input data, the lower and upper bounds used to build the up and down regulated reaction sets. The reactions with score values lower than the lower bound specified by user in the configuration panel, will be considered down regulated by the algorithm. On the other hand, reaction with scores higher than the upper bound will be considered upregulated.

The E-Flux does not require any additional information. Thus, it is not necessary any configuration panel for this method.

The GIMME configuration panel, GIMMEConfigurationPanel, accepts two ways to limit the RMF's flux reactions: a list of reactions scores, a ReactionOmicsBox datatype, where each score represents the maximum flux for a reaction, or a list of metabolic tasks used to calculate the maximum flux for the objective function, a reaction, of each metabolic task.

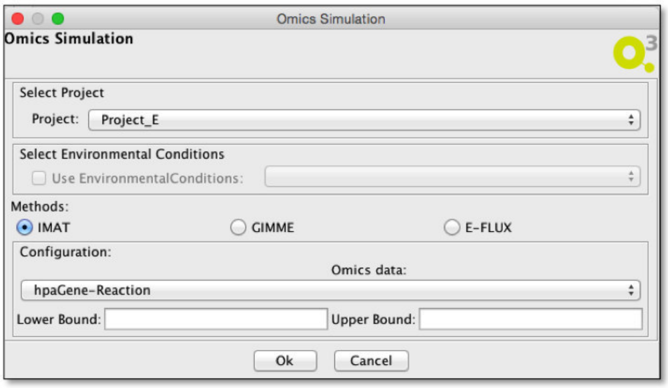
Additional fields are required to run this method, such as:

- **Percentage:** this value is used to constrain the RMF's objective in the formulation. These reactions which represent the RMFs must have a flux higher than a percentage of the maximum possible flux.

- **Cutoff:** is a threshold value set by the user above which a reaction is considered to be present.

3.8.2 Functionalities

The `OmicsSimulationGUI` provides a dialog for the `OmicsSimulationOperation`, which allows the user to configure and launch a strain simulation with omics integration procedure. This dialog is depicted in Figure 3.21.


Omics Simulation

Select Project

Project: Project_E

Select Environmental Conditions

□ Use EnvironmentalConditions:

Methods: ● IMAT ○ GIMME ○ E-FLUX

Configuration:

Omics data: hpaGene-Reaction

Lower Bound:

Upper Bound:

Figure 3.21: Screenshot of the simulation methods configuration dialog.

This GUI allows setting up and configuring several optimization parameters:

- **Select Project:** select the metabolic model associated to the project to perform the simulation;
- **Select Environmental Conditions:** the list of available environmental conditions for this project.
- **Select Omics Data:** select the instance of `ReactionOmisBox` datatype used as input. Only this data type is available, because all the available methods use ReactionDataMap object as input;

- **Select Simulation Method** the method used to perform the simulation;
- **Configuration Panel:** this panel depends of the selected method. Each method has it own panel with the required configuration fields (`IMATConfigurationPanel`, `IGIMMEConfigurationPanel`, `EFluxConfigurationPanel`). All these classes are an extension of the abstract class `AbstractOmicsSimulationConfigurationPanel`.

3.9 Conclusion

This work proposes an integrated framework to use omics information in phenotype predictions and to reconstruct tissue-specific metabolic models. The development is segmented in two layers allowing both users with computational skills and regular users to use the methods implemented during this thesis. Moreover, the addition of new features and new methods can be done easily by programmers using the provided API.

The addition of two new plug-ins to integrate omics data with models and phenotype prediction in the open-source OptFlux platform makes it an attractive resource to an ever increasing community.

The described software was developed in the Java programming language, and is available as an open source packages (`mewomicsintegration` and `optflux-omicsintegration`) in `sourceforge.net/p/optflux/`. Moreover, a docker container is available in the repository `https://hub.docker.com/r/saracorreia/tsmm_U251` which allows the reconstruction of tissue-specific metabolic models for the U-251 cells line (further described in Chapter 6).

Future work contemplates the development of a new plug-in to support a graphical user interface for the tissue-specific reconstruction methods.

Chapter 4

Evaluating Phenotype Simulation Methods

In this chapter, a validation of the phenotype simulation methods implemented in this work was performed, using omics datasets from a previous study. The three implemented methods: E-Flux, GIMME and iMAT were used to perform phenotype prediction and their flux distributions were compared with the experimental data provided in this study.

4.1 Introduction

The nicotinamide adenine dinucleotide (NADH) and the adenosine triphosphate (ATP) cofactors play an important role in metabolism. NAD is involved in redox reactions, carrying electrons from one reaction to another and the ATP is the source of energy to several biological processes that occur in the cell [107]. These cofactors, NADH and ATP, are highly connected in the metabolic networks of most microorganisms [108]. Thus, it is expectable that small changes in their concentration causes significant modifications in several parts of the metabolism.

In 2010, Holm et al. [109] studied the impact of these two cofactors in the *Escherichia coli* metabolism regulation. More specifically, the authors wanted to understand aspects of metabolism that are controlled by the levels of NADH or ATP present in the cell. In the study [109], they compared the phenotype of the wild-type strain and two mutants: *NOX* (overexpression of NADH oxidase) and *ATPase* (overexpression of soluble F1-ATPase). The analysis was done based on the quantification of the metabolic fluxes in central carbon metabolism and the genome-wide transcription for the three *E. coli* strains.

4.2 Methods

As a case study, this dataset was used to evaluate the implemented phenotype simulation methods: E-flux, GIMME and iMAT. The dataset is composed by transcriptomic and ^{13}C-flux data for the three *E .coli* strains. The transcriptomic data can be obtained from the NCBI Gene Expression Omnibus using the accession number GSE20374 and metabolic flux measurements are available in the supplemental material of the publication [109].

The implemented simulation methods were applied using the genome-scale metabolic model iAF1260 [110] to predict the phenotype and considering the gene expression data as input . In all simulations, the glucose uptake constraint present in the original model was overridden with the experimental value for each strain present in the dataset. Thus, for the simulation of each *E .coli* strain, the limit of the glucose uptake present in Table 4.1 was considered:

Table 4.1: The glucose uptake rate constraint for each *E. coli* strain.

	Wild type	**NOX**	**ATPase**
Glucose	9.2	11.7	15.6

4.2.1 Pre-processing the fluxomics data

Following a suggestion from [111], the experimental data values were adjusted to the feasible flux distributions obtained with the metabolic model, with the smallest Euclidean distance to the original values. This modification is desirable because some of the experimental values do not lie within the solution space, being the error propagated to the methods evaluation. Table 4.2 contains the original flux values measured by ^{13}C-labeling and the value after the adjustment, used in this case study for methods evaluation purposes.

Table 4.2: Original measured values and the adjusted values obtained by the closer feasible flux distribution using the metabolic model iAF1260.

	Original			**Adjusted**		
Reaction	**WT**	**NOX**	**ATPase**	**WT**	**NOX**	**ATPase**
Ec_biomass	0.67	0.63	0.58	0.47	0.49	0.39
FUM	1.6	4.8	4.3	1.67	4.84	4.36
G6PDH2r	4.4	4.9	5.1	4.40	4.90	5.10
GAPD	15.3	20.4	28.3	15.31	20.41	28.31
GLCptspp	9.2	11.7	15.6	9.20	11.70	15.60
GND	4.4	4.9	5.1	4.40	4.90	5.10
ICDHyr	2.5	5.6	5.1	2.43	5.56	5.04
ME1	0	0.2	0	0	0.16	0
PGK	−15.3	−20.4	−28.3	−15.31	−20.41	−28.31
PGL	4.4	4.9	5.1	4.40	4.90	5.10
PPC	4	4	6.5	4.07	4.04	6.56
PPCK	2	1.8	4.8	1.93	1.76	4.74
RPE	2.4	2.8	3	2.39	2.80	2.99
RPI	−2	−2.1	−2.1	−1.99	−2.08	−2.09
TALA	1.3	1.5	1.6	1.30	1.51	1.61
TKT1	1.3	1.5	1.6	1.30	1.51	1.61
TKT2	1.1	1.3	1.4	1.09	1.29	1.39
TPI	7.1	9.6	13.5	7.06	9.52	13.43

4.2.2 E-Flux

The implementation of E-flux, described in section 2.5.3, is basically an extension of the FBA problem, where the fluxes are constrained based on trascriptomic data. The score value of each reaction is calculated based on the GPR association, where the OR / AND operators are converted to Max / Plus functions as described in the original publication [15]. These scores are then normalized causing each reaction, r_j, to be constrained with an upper bound b_j, between 0 to 1, and a lower bound equal to $-b_i$ or 0, when the reaction is irreversible. Furthermore, all uptake reactions are constrained to a lower bound of -1. The resulting flux distribution is adimensional. Thus, to compare it with the original flux distribution, the values were scaled by the experimental measured glucose uptake rates.

4.2.3 GIMME

The GIMME method, detailed in section 2.5.2, besides the transcriptomic data, receives three parameters as argument:

1. the gene expression cutoff, which was set to the 25th percentile of the gene expression values;

2. the metabolic function that represent the Required Metabolic Functionality - in this case, the maximization of biomass production;

3. the required fraction of the objective value, which was set to 90% of the maximum growth rate.

For each strain, according to the description in section 2.5.2, a FBA was performed to find the maximum possible flux rate of the biomass equation. Next, the algorithm was run with a constraint over this reaction flux which must be at least 90% of the maximum flux.

4.2.4 iMAT

The iMAT (section 2.5.1) implementation also takes three parameters as input: the low and high expression thresholds used to calculate the up and down regulated reactions, and a threshold which was set to 1, as in the original publication [13]. The two expression thresholds were set to the 25th and 75th percentiles of the experimental flux data. The reactions with score lower than the 25th percentile were considered downregulated, while scores higher than the 75th percentile belong to upregulated reactions.

The Table 4.3 presents the threshold values used in each strain phenotype prediction by GIMME and iMAT methods.

Table 4.3: The 25th and 75th percentiles of the experimental flux data from each strain used as thresholds by GIMME and iMAT methods.

Strain	**25th percentile**	**75th percentile**
Wild type	8.12	11.41
NOX	8.95	11.27
ATPase	8.58	11.25

4.3 Results

The phenotype prediction, using the implemented methods, was done for the three *E. coli* strains using the transcriptomic data available from [109]. The GIMME and iMAT methods were previously available in the COBRA Toolbox. Moreover, this dataset was already used to compare and evaluate the results of several methods in Machado et al. [111].

Here, the prediction capability of each implemented method in our framework is compared with the experimental data [109] and the previous work [111] with the same datasets.

Figure 4.1 shows the secretion flux rates for the two available external experimental measurements in the dataset.

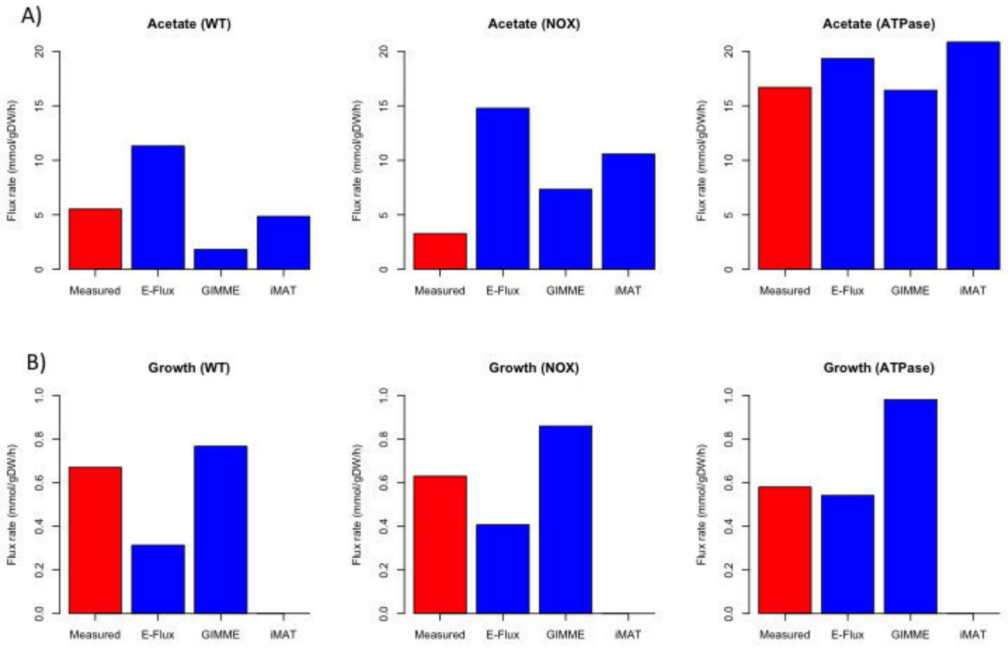


Figure 4.1: Predicted and measured flux rates for acetate secretion and growth for the three *E. coli* strains using the methods E-Flux, GIMME and iMAT.

The secretion of acetate is reached in all phenotype predictions. However, none of the algorithms are able to show the decreasing flux rate between the wild type and the NOX strain, as shown in the Figure 4.1 A). Regarding the growth rate prediction, E-flux and GIMME are capable to predict cellular growth. However, once again, the small decreasing flux rates in the ATPase strain are not shown by the predicted values.

These results are significantly different from those presented in [111], where the phenotype prediction for the two mutants (NOX and ATPase) were analyzed. In their study, none of the methods presented secretion rates for acetate and only the E-Flux is able to predict growth.

4.3.1 Prediction error

In order to evaluate the prediction capability of the methods, the flux distributions obtained using transcriptomic data from three *E. coli* strains are compared against the adjusted experimental measured fluxes.

Figure 4.2 shows the distribution of the normalized prediction error for each method across the three strains (wild-type, NOX and ATPase). The flux distribution errors were compared with the error obtained using the pFBA (parsimonious version of FBA) method for the phenotype predictions.

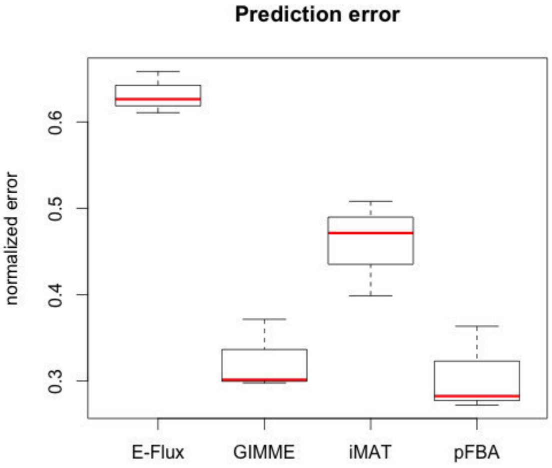


Figure 4.2: Prediction error for the simulation methods. Distribution of normalized prediction error for the methods E-Flux, GIMME, IMAT, pFBA.

The normalized prediction error was calculated for each simulation, comparing its results with the adjusted experimental values. The estimation error is given by the equation:

$$error = \frac{\|v^{exp} - v^{sim}\|}{\|v^{exp}\|} \tag{4.1}$$

where v^{exp} is the vector of adjusted flux values and v^{sim} is the vector of predicted values.

Similarly as concluded in [111], the three methods have a lower predictive capability, when compared with the pFBA. However, in our results the iMAT present a significant improvement, comparing with the results obtained in [111] are considered.

Next, to better understand how the distribution error varies accross the measured flux reactions, a heatmap of the differences between the predicted and measured fluxes is presented in Figure 4.3.

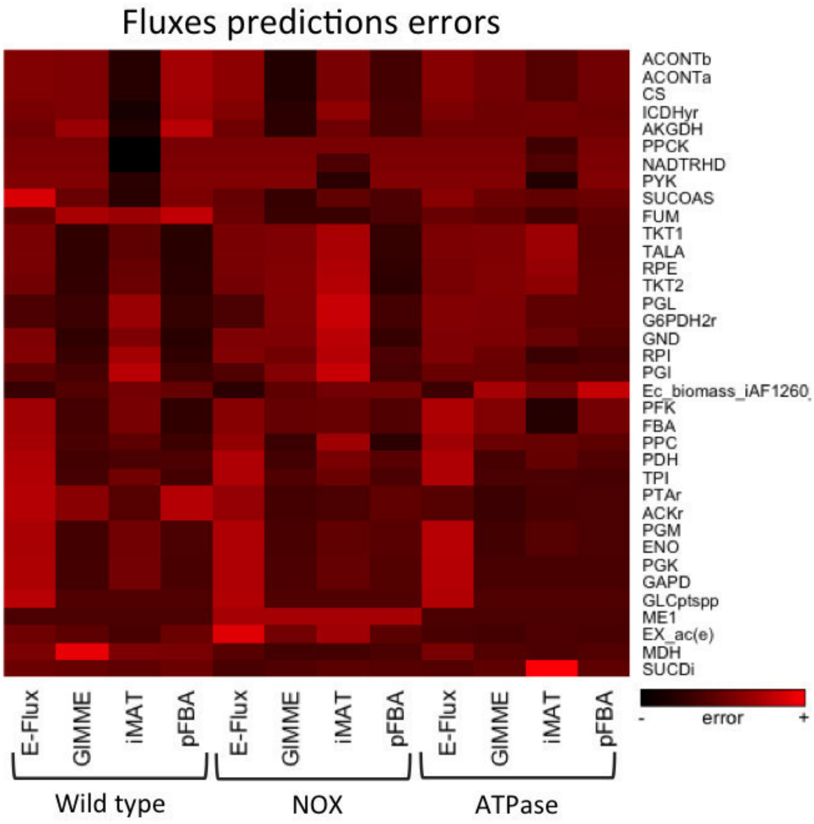


Figure 4.3: Difference between predicted and measured fluxes for all evaluated methods across all *E. coli* strains.

Looking to the prediction errors for a specific strain, it is visible that the GIMME and pFBA methods have similar error distributions across the presented reactions. Moreover, considering the wild type strain, the E-Flux method has a better prediction for reactions where other methods fail, such as: *ACONTa*, *CS*, and *AKGDH*.

4.3.2 Central carbon metabolism

The experimental measured fluxes belong to the central carbon metabolism of *E. coli*. In order to analyse the flux distribution over these reactions and compare it to the predicted flux distributions, Figure 4.4 presents the main reactions involved and the flux distributions for the wild type strain.

Analysing these flux distributions, it is visible that some reactions carry flux in the opposite direction when compared with the measured fluxes, such as the *R_PGI* reaction for the iMAT flux distribution and the *R_TKT1* and the *R_TKT2* reactions in the E-flux.

The conversion of *PEP* to *Pyruvate* is essentially done through the *R_GLCptspp*, instead of the emphR_PYK reaction. Moreover, the E-Flux distribution has an alternative pathway, not seen in this image, to convert *Pyruvate* into *acetyl-CoA*, since the reaction *R_CS* from the TCA cycle has flux.

Finally, the flux rates associated with the reaction *R_ACKr* confirm the acetate production in the predictions.

4.4 Conclusion

In this chapter, a validation of the implemented simulation methods was performed, comparing our results with previous published work [111]. The two studies use the same *E. coli* metabolic model and transcriptomics data to perform phenotype prediction, using E-Flux, GIMME and iMAT as simulation methods. Next, the ^{13}C flux measurements from [109] were used to evaluate the capability prediction of such methods .

In general, our results are similar with the published results. In both cases, the phenotype predictions using pFBA have a lower prediction error associated. This can be explained by the low number of measured reactions, only 36 of the 2382 present in the metabolic model iAF1260.

The main difference between the two studies is found in the secretion of acetate. In [111], none of the predicted flux distributions have secretion of acetate, but in our results the acetate production occurs. This difference might be explained since the methods allow different flux distributions achieving optimal values for the objective function.

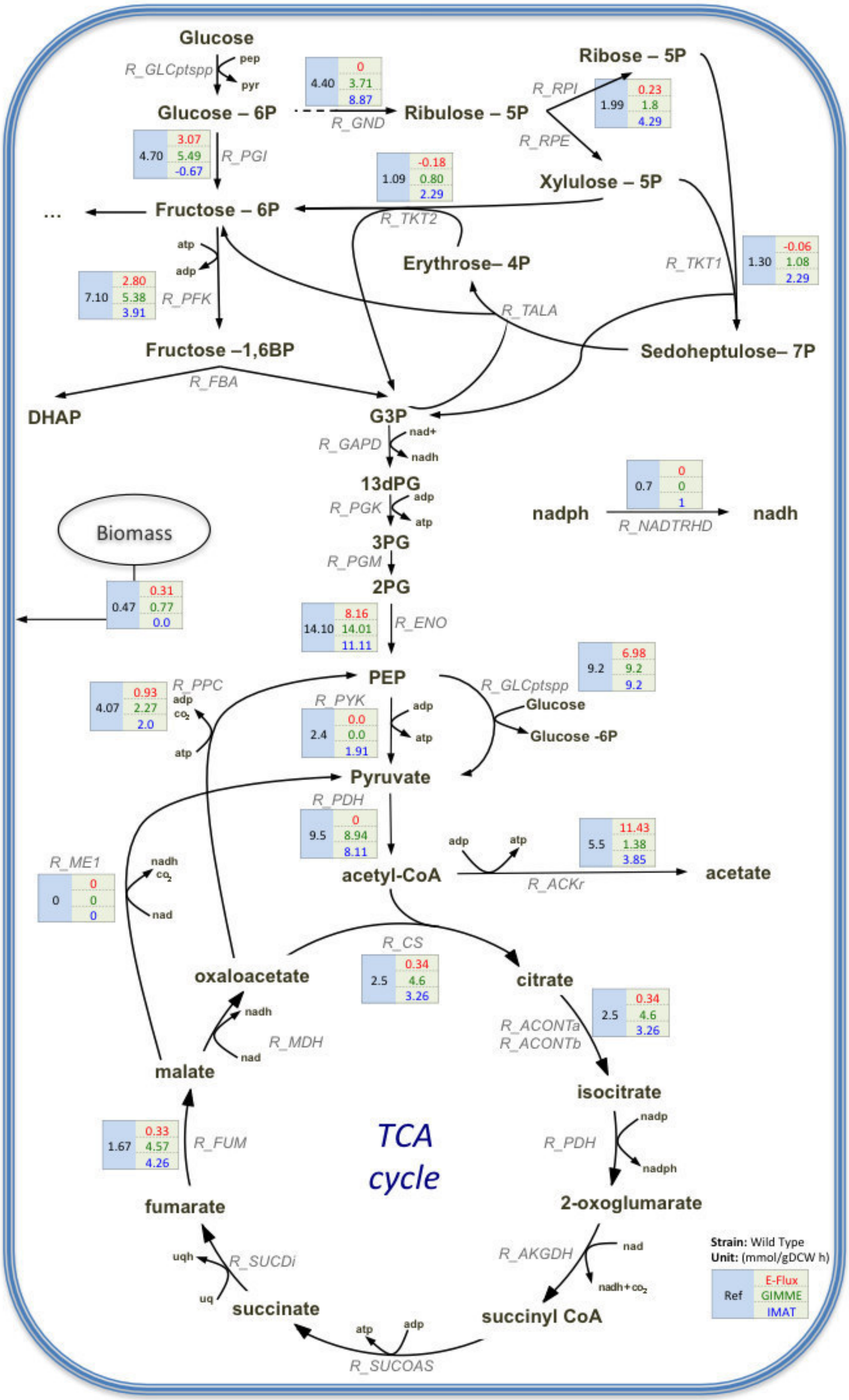


Figure 4.4: Metabolic flux distribution in the central carbon metabolism of *E. coli* wild type strain, of experimental data (black), and using the simulation methods: E-Flux (red), GIMME (green) and iMAT(blue).

Chapter 5

Evaluating Tissue-Specific Model Reconstruction Algorithms and Data Sources

This chapter presents the comparison and analysis of the consistency between several omics data sources. Moreover, four published approaches were used to reconstruct hepatocytes metabolic models using different omics data. These models were compared and validated through a list of metabolic tasks, which hepatocytes cell must perform. Finally, based on the results, a method to build a consensus final model is proposed to generate our final hepatocytes metabolic model.

5.1 Introduction

In chapter 2, we presented methods for integrating genome-scale metabolic models with omics data, which can be separated in two categories. The first one encompasses all the methods that use these data to improve the prediction of metabolic flux distributions, such as iMAT, GIMME and E-Flux. These methods have been already critically evaluated and compared in published work by Machado et al [111] and in the previous chapter. The other category covers the context-specific reconstruction methods which use generic metabolic models and omics data as input. Here, we include methods such as MBA [16], mCADRE [21], tINIT [20] and FASTCORE [22].

Recently, two categories of approaches have been proposed to test model building algorithms [112]. The first encompasses tests for assessing the algorithms robustness against noise, while the second covers the comparison of a set of functionalities that models are able to perform, using published data as reference. However, these tests were done using a single omics data type as input, which does not allow the analyses of the effects of input data in final models behaviour. Moreover, important methods such as tINIT, mCADRE and MBA were not considered in validation process.

Thus, the impact of using different omics datasets on the final results of those algorithms is a question that remains to be answered.

5.2 Methods

The human liver is one of the most important organs in the regulation of the human metabolism, being responsible for numerous functions, as the production of bile, removal of toxic substances, decomposition of red cells and chemical regulation of the plasma [113]. The liver consists in different types of cells: parenchymal cells (hepatocytes and bile duct cells) and nonparenchymal cells. Disorders in the metabolism of distinct cell types cause a number of diseases, like hepatitis, nonalcoholic fatty liver disease or hepatocellular carcinoma (HCC) [114]. HCC strikes about half a million humans in the world and it is the most usual form of primary cancer [115].

The analysis of the differences at a molecular level of healthy and disease states, made possible by the enhanced high throughput technologies and decreasing costs of obtaining different *omics* data, can help to clarify the functional mechanisms of liver cells and related diseases [116].

About 78% of the liver tissue is formed by hepatocyte cells that are the principal site of the metabolic conversions underlying the diverse physiological functions of the liver [117]. To have a better understanding of how hepatocyte cells work, different algorithms have been applied to reconstruct tissue specific metabolic models for this cell type [16, 21]. Also, Gille et al. built a manually curated GSMM for hepatocytes, the HepatoNet1 [17].

Here, hepatocytes metabolic models were reconstructed using different omics data sources and different algorithms, to evaluate the effects that each of those variables have in the resulting tissue-specific metabolic models and their behavior.

5.2.1 Generic human metabolic models

At the time of this work, three generic genome-scale human metabolic models and a reaction database used to reconstruct tissue-specific models were available [7, 8, 10, 62].

An analysis of the most used generic human metabolic models [7, 10, 62] in the reconstruction of tissue-specific models was performed to highlight the main differences between them. This was done using an integration system developed in our research group by Liu et al. (unpublished). The reaction and metabolites are unified in a Neo4j graph database [118], where information present in the models such as KEGG and ChEBI identifiers, chemical formulas and names are used to integrate metabolites into clusters. The integration of reactions was done using these clusters and assuming that reactions from different models are the same if they have the same metabolites (i.e. metabolites joined in the same clusters) as reagents and products.

During the integration process, the presence of protons in the reactions was ignored. So, if two reactions from a model differ only in the presence/absence of protons they will be considered the same reaction. This assumption is only made for the overlap analysis of generic metabolic models.

In the present analysis, the drain reactions were not considered. Therefore, the metabolic models Recon 1, Recon 2 and HMR 2.0 have 3.207, 6.462 and 6.896 unique reactions, respectively.

The overlap between models is shown in Figure 5.1. As expected, the Recon 2 has almost all reactions present in Recon 1, its previous version.

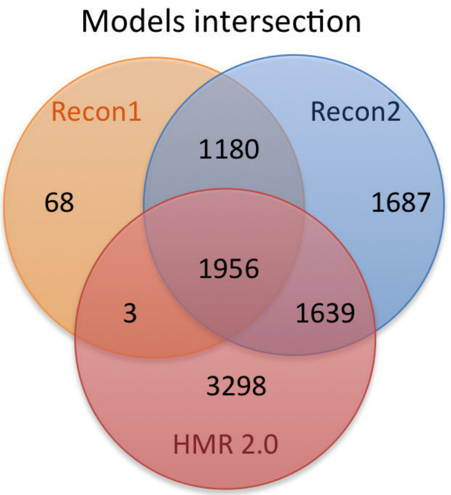


Figure 5.1: Number of integrated reactions across the three main generic human metabolic models - Recon 1, Recon 2 and HMR 2.0.

Being the HMR2.0 constructed by integrating the elements of stoichiometric networks of human metabolism, namely Recon1, the Edinburgh Human Metabolic Model and the KEGG database [62], it was expected a better consensus between the two analysed models (Recon 1 and HMR2.0).

Next, the analysis of which pathways are associated with the non integrated reactions was done using the subsystem information present in the models (loaded from their SBML files). The differences between Recon1 and Recon2 are essentially related with fatty acid synthesis and oxidation pathways (248 reactions) and transport reactions between compartments (544 reactions).

The comparison between Recon 2 and HMR2.0 is more difficult to make, since the pathway identifiers are different and in some cases a single pathway in one model is split in more than one pathway in the other. Moreover, 967 non integrated reactions from HMR2.0 do not have information about their subsystem. Nevertheless, the reactions not integrated between the two models belong essentially to the pathways: Glycerolipid metabolism, Formation and hydrolysis of cholesterol esters, Glycerophospholipid metabolism, Carnitine shuttle, Sphingolipid metabolism, Leukotriene metabolism, Phenylalanine, tyrosine and tryptophan biosynthesis.

The difficulty of model integration and the poor overlap between them was already discussed in 2011 by Stobbe et al. [119]. The standardization of metabolite names and identifiers and the manual curation are still required to improve and develop an unified and biologically accurate metabolic model.

5.2.2 Reference metabolic model

In 2010, a genome-scale metabolic network of human hepatocytes was presented by Gille et al, the HepatoNet1 [17]. This network enables the application of constraint-based modeling techniques to discriminate allowable metabolic states in hepatocytes in different environmental conditions. The initial list of reactions to consider to establish a stoichiometric model of human hepatocyte metabolism was obtained from the two existing global reconstructions of the human metabolic network - Recon 1 and EHMN, and from the KEGG database. Moreover, databases like BRENDA, Reactome and UniProtKB were used for validation proposes [17].

The resulting metabolic network satisfies 442 different metabolic objectives, related to known metabolic liver functions, and guarantees that impossible tasks are not achievable within the network. Furthermore, the final metabolic model comprises 777 metabolites in six intracellular and two extracellular compartments and 2.539 reactions, including 1.466 transport reactions.

5.2.3 Input data sources

All the context-specific reconstruction methods used take as input a generic metabolic model and information from omics data. The omics data used for the reconstruction of the hepatocytes metabolic models were obtained from proteomics and transcriptomics. Manually curated sets of reactions used in [16], to reconstruct the hepatocytes metabolic model, were considered.

Proteomics data were retrieved from the Human Protein Atlas (HPA) [84], which contains the profiles of human proteins in all major human healthy and cancer cells. The information was collected for the liver tissue (hepatocytes) from HPA version 12 and Ensembl [120] version 73.37. After a conversion from Ensembl gene identifiers to gene symbols, duplicated genes with different evidence levels were removed. Table 5.1 presents the list of removed genes during this process.

Table 5.1: Gene symbols with different evidence levels in Human Protein Atlas.

Ensembl ID	Expression Level	Gene Symbol
ENSG00000169894	Medium	MUC3A
ENSG00000228273	High	MUC3A
ENSG00000115540	Low	MOB4
ENSG00000270757	Medium	MOB4
ENSG00000123444	Not detected	KBTBD4
ENSG00000231880	High	KBTBD4
ENSG00000080200	Not detected	CRYBG3
ENSG00000233280	High	CRYBG3
ENSG00000243649	Low	CFB
ENSG00000244255	Medium	CFB
ENSG00000181464	Medium	CDRT1
ENSG00000241322	High	CDRT1
ENSG00000169894	Medium	MUC3A

Transcriptomics data were collected from the Gene Expression Barcode (GEB) [96] (HGU133plus2 (Human) cells v3). The conversion to gene expression levels was done considering the average level of probe sets for each gene. The mapping between probe sets and gene symbols was performed using the library "hgu133plus2.db" [121] from Bioconductor [122]. Bioconductor provides tools for the analysis of high-throughput genomic data, using the R statistical programming language. In the version 3.2, Bioconductor contains 1104 software packages, 257 experiment data packages, and 917 annotation packages.

The context-specific reconstruction methods, chosen for this work, use different formats of input data. Specifically, MBA uses two sets of reactions, where each reaction has *High* or *Moderate* probability to be in the final model, while mCADRE and FASTCORE expect only one set of reactions as input. In the tINIT method, each reaction from the generic metabolic model must have a score value of 20, 15, 10, -8 representing the *High*, *Moderate*, *Low* or *Not detected* evidence of protein expression levels respectively. A default value of -2 is used for reactions without information in input data. The Table 5.2 summarizes the input data type supported by each algorithm.

Table 5.2: Required and optional input data for each algorithm.

	Required	**Optional**
MBA	Two reaction sets (high and moderate probability)	
tINIT	Reaction scores	Set of required reactions calculated based on a set of metabolic tasks. Set of metabolites that final model must produce
mCADRE	One reaction set (core)	Set of metabolites that must be produced in the final model
FASTCORE	One reaction set (core)	

These input data diversity leads to the requirement of data transformation, in order to allow its use in different methods. The continuous data from GEB was classified as *High*, *Moderate* and *Low*, if the gene expression evidence on that tissue is greater than 0.9, between 0.5 and 0.9, and between 0.1 and 0.5, respectively. The genes with expression evidence below 0.1 were considered not expressed in hepatocytes. The core reaction sets used in mCADRE and FASTCORE methods were built considering the union of "High" and "Moderate" gene evidences from HPA, or gene expression evidence greater than or equal to 0.5 from GEB, through the GPR association present in the template model.

The Table 5.3 summarizes the assumptions used to create the input data sets for each algorithm.

Table 5.3: The table summarizes the assumptions and thresholds used to create the sets used as input by the algorithms.

Algorithm	**input**	$\boldsymbol{C_H}$, $\boldsymbol{C_M}$	**HPA**	**GEB**
MBA	C_H	C_H	*High*	$[0.9, 1.0]$
	C_M	C_M	Moderate	$[0.5, 0.9[$
tINIT	High	C_H	*High*	$[0.9, 1.0]$
	Moderate	C_M	Moderate	$[0.5, 0.9[$
	Low		Low	$[0.1, 0.5[$
	Not detecetd		Not detected	$[0.0, 0.1[$
mCADRE	core	$C_H \cup C_M$	High ∪ Moderate	$[0.5, 1.0]$
FASTCORE	core	$C_H \cup C_M$	High ∪ Moderate	$[0.5, 1.0]$

Applying these transformation rules is possible to adapt different input data sources, such as HPA, GEB and C_H and C_M sets, for all methods.

5.2.4 Reconstruction workflow

A framework with the four tissue-specific metabolic models reconstruction methods was implemented as described in chapter 2. All the algorithms in this framework were adjusted to receive a reaction scores map as main input. Nevertheless, some methods such as mCADRE and tINIT, still allow the use of a set of metabolites which must be produced in the final model.

Therefore, the algorithms are made independent from the omics data source, and the separation of these two layers allows to use different data sources combined with each algorithm for the generation of tissue-specific metabolic models.

Generically, the hepatocytes metabolic models reconstruction process had four main steps:

1. First, it was necessary to collect the data from HPA and GEB repositories. The Ensembl gene identifiers present in HPA information were converted to gene symbols to allow the integration with the template metabolic model - Recon 1. A similar transformation was required to convert the GEB information, where the original expression level is associated with probe sets. The reaction sets C_H and C_M from [16] are already at the reaction scores level, so the current and next steps were not required.

2. Based on the data from previous step, it was required to convert the gene scores to reaction scores. This was performed through GPR associations present in the template model by the substitution of AND/OR operators by the Min/Max functions. If one of the gene scores is unknown, its value was ignored in the GPR association.

3. Next, the final core reaction sets were built based on the assumptions described in Table 5.3. In this step, some additional configurations were required depending on the selected algorithm. tINIT, for instance, receives a set of metabolic tasks as input to obtain the required reactions to perform those tasks. The metabolic tasks, that should occur in all cell types, were retrieved from [20]. A set of metabolites can be defined in mCADRE and tINIT algorithms, ensuring the production of those by the resulting models. These configurations and algorithm parameters were set with default values from the original publications [20, 21].

4. Finally, the algorithm was run to reconstruct a hepatocytes metabolic model. The final MBA models were constructed based on 50 intermediate metabolic models. According to [16], a larger number would be

desirable, but the time needed to generate each model prevented larger numbers of replications.

The workflow described above can be depicted in Figure 5.2.

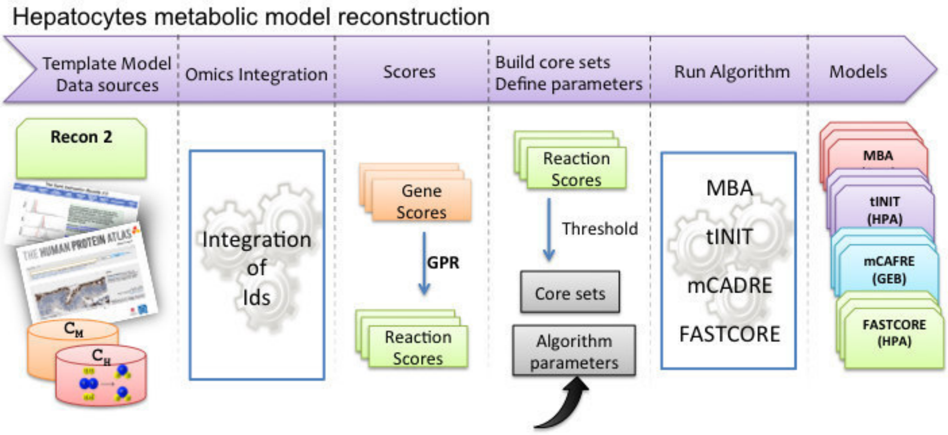


Figure 5.2: The hepatocytes models reconstruction workflow encompasses four main steps: 1- transformation of input data identifiers to model notation; 2 - conversion of gene scores to reaction score through the GPR associations, by substitution of the operators And/Or by the functions Min/Max; 3- configuration of the algorithms properties and data filtering based on thresholds; 4 - run the algorithm.

This pipeline was applied considering Recon 1 human metabolic model as template. At the end, 12 hepatocytes metabolic models were reconstructed based on the combination of four methods (MBA, mCADRE, tINIT, FASTCORE) and three data sources (C_H and C_M reaction sets, HPA and GEB).

5.2.5 Model validation

The quality of the metabolic models was further validated using the metabolic functions that are known to occur in hepatocytes taken from HepatoNet1 [17]. This set includes a total of 433 functional tasks divided in two categories: 310 network tasks and 123 physiological tasks.

Each task is composed by two sets of metabolites, that can be uptaken and excreted by the model and an objective function which represents the target metabolite to be produced. All reactions connected to the extracellular environment, also called drains, not present in the task should be closed, i. e. constrained not to allow any flux.

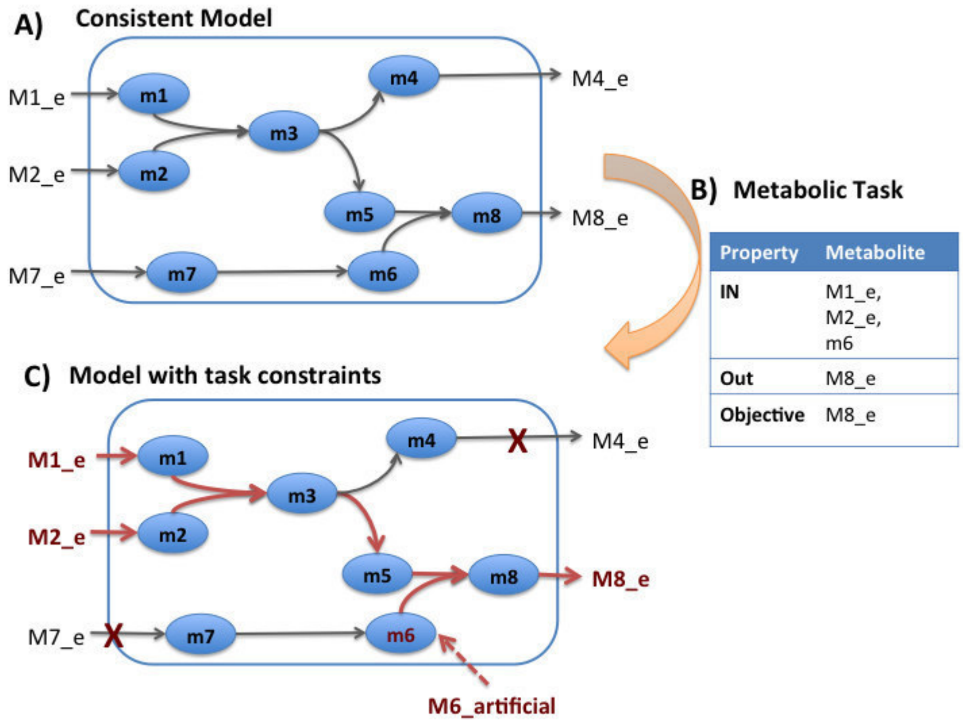
Internal metabolites are accepted in the metabolic task definition. In this case, artificial reactions are added to the model to allow the uptake or excretion of those metabolites. In the task validation process, it is assumed that all internal metabolites involved, when present in the model, can be consumed/produced. This assumption can be done without affecting the final validation result because the model is consistent, i.e, all reactions are able to carry flux, which implies that all present metabolites can be produced/consumed.

A FVA for each reaction is performed to find reactions where the maximum and minimum possible flux is equal to 0. Afterwards, these reactions will be removed from the original model. This simplification of the model is done before the validation process starts. Moreover, tasks with metabolites not present in the model are tested without the uptake/excretion of these metabolites. However, tasks will not be validated if the objective metabolite is not present in the model.

Figure 5.3 shows the main model modifications by the integration of a metabolic task.

5.2.6 Consensus model reconstruction

The final hepatocytes consensus model was reconstructed based on the 12 metabolic models obtained by the process described above. The main idea is to build a model starting with the common reactions present in most of the models and append a set of reactions to the final model so, it will be able to perform all the metabolic tasks. The final reconstruction consisted on three main steps:



Property	Metabolite
IN	M1_e, M2_e, m6
Out	M8_e
Objective	M8_e

Figure 5.3: A) Consistent metabolic model, where all the reactions are able to carry flux. B) Metabolic task to simulate the production of metabolite m_8, allowing the uptake of $M1_e$ and$M2_e$ and assuming that m_6 is produced in the model. C) Modifications to the system: close all drains not present in the metabolic task and insert artificial drains for the internal metabolite. The red lines represent the flux distribution after maximizing the objective function.

1. Build 12 $partial-models$, hereafter designed as $pModels$, where the i^{th} model contains the reactions present in at least i hepatocytes models, where $i \in \{1, 2, ...12\}$. Therefore, $pModel_1$ contains the union of all reactions from the 12 models, while $pModel_{12}$ contains only the reactions present in the intersection of all models. Additionally, the template model Recon 1 is considered as $pModel_0$.

2. Run the validation tasks process for each $pModel$ and choose a value of n, where $pModel_n$ is the smallest model with an acceptable number of valid tasks. .

3. Next, it is necessary to calculate the reactions and valid task sets that

differ between 2 neighbouring models (i.e. models with indexes i and $i+1$). As a result, two lists are created, the lost reactions set (LRS) and the lost metabolic tasks set (LMT). Each lost metabolic tasks set, LMT_i represents the set of tasks satisfied by $pModel_i$, when compared with $pModel_{i+1}$. Similarly, each LRS set, LRS_i, represents the reactions present in $pModel_i$ and not in $pModel_{i+1}$.

4. Finally, run Algorithm 4 to generate the final model. The algorithm starts with $pModel_i$, and taking into consideration the LRS_i and LMT_i, finds the reactions of LRS_i that are not required to perform the tasks in LMT_i. These reactions compose the $toDel_i$ set. At the end of each iteration, the reactions that do not have an influence in the loss of metabolic tasks performance, between two partial models ($toDel_i$), are appended to the LRS_{i-1} in the next iteration. The process ends with the processing of $pModel_0$, in this case the full Recon 1 model.

Figure 5.4 shows the steps in the reconstruction of the consensus final model algorithm.

5.3 Results

The hepatocytes metabolic models were generated using Recon 1 the as template model and the GEB, HPA and the C_H and C_M sets from [16] as input data, by the four methods considered in this study: MBA, tINIT, mCADRE and FASTCORE.

Three main questions were answered: Are omics data consistent across different data sources? What is the overlap of the resulting metabolic models obtained using different methods and different data sources? How do the obtained models behave in functional terms regarding metabolic tasks?

Algorithm 4 Reconstruct the final model based on pseudo-models.

```
function buildFinalModel($pModels$, $n$, $LMT$, $LRS$)
    $toDel = \{\}$
    for $i \in \{n, n-1, .., 0\}$ do
        $finalModel = pModel_i$
        $reacs_i = LRS[i]$
        $tasks_i = LMT[i]$
        for ($r \in reacs_i \cup toDel$) do
            $allValidWithKO = isAllValid(finalModel, tasks_i, r)$
            if ($allValidWithKO$) then
                $finalModel = finalModel \backslash r$
                $toDel = toDel \cup r$
            end if
        end for
    end for
    return $finalModel$
end function

function isAllValid(($finalModel, tasks_i, r$))
    Test if all tasks present in $tasks_i$ are satisfied by the $finalModel$ when the reaction $r$ is removed.
end function
```

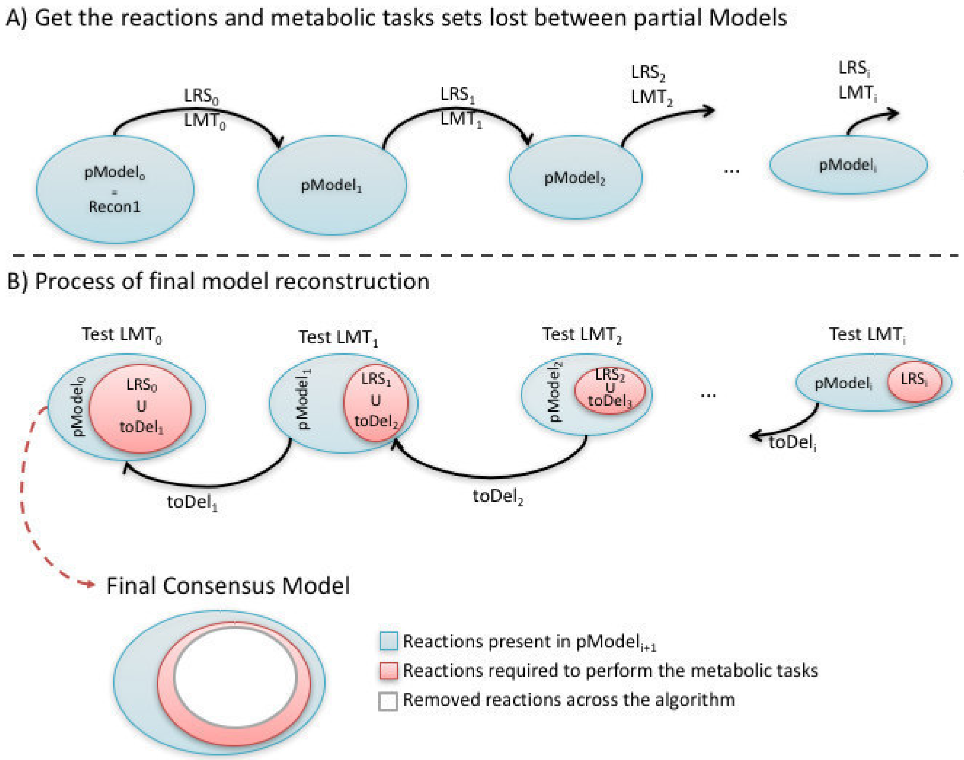

A) Get the reactions and metabolic tasks sets lost between partial Models
LRS0
LMT0
LRS1
LMT1
LRS2
LMT2
LRSi
LMTi
pModel0 = Recon1
pModel1
pModel2
...
pModeli
...
B) Process of final model reconstruction
Test LMT0
Test LMT1
Test LMT2
Test LMTi
pModel0
LRS0 U toDel1
pModel1
LRS1 U toDel2
pModel2
LRS2 U toDel3
...
pModeli
LRSi
toDeli
toDel1
toDel2
Final Consensus Model
Reactions present in pModeli+1
Reactions required to perform the metabolic tasks
Removed reactions across the algorithm

Figure 5.4: A) Build the lost reaction set (LRS) and the lost metabolic tasks set (LMT) between each par of partial models. B) For each $pMode_i$l the process find the reaction from LRS_i that can be removed from $pModel_I$ without affect the metabolic tasks present in (LMT_i) production. The set of reactions that can be removed will be added to LRS set in the next iteration.

5.3.1 Omics data consistency

The HPA (version 12) has evidence information related with 16324 genes in hepatocytes. The reliability of the data is also scored as “supportive” or “uncertain”, depending on similarity in immunostaining patterns and consistency with protein/gene characterization data [84]. On the other hand, the GEB transcriptome (HGU133plus2_cells_v3) has information for 20149 genes, of which 5772 have evidence of being expressed in hepatocytes [96].

Together, these two data sources have information for 21921 genes, but only 14552 are present in both (Figure 5.5A). Moreover, the number of genes with evidence of being expressed in the tissue in both sources is only of

3549, around 24% of all shared genes (Figure 5.5B). These numbers decrease significantly if using only HPA information marked as “supportive”. In this scenario, only 3868 genes are present also in GEB and only 1294 of them have expression evidence.

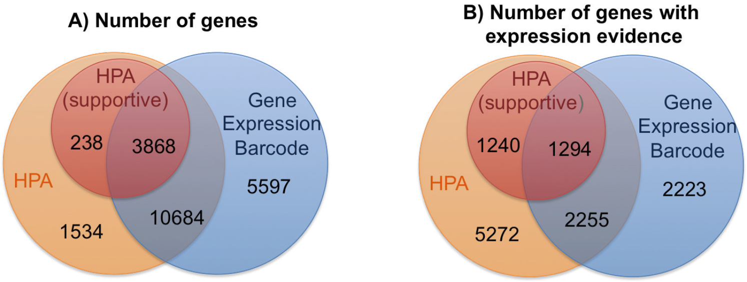


Figure 5.5: A) Number of genes present in Gene Expression Barcode and Human Protein Atlas. In HPA, the number of genes with reliability “supportive” and “uncertain” are shown. B) Number of genes with evidence level “Low”, “Moderate” or “High” in HPA and gene expression evidence higher than 0 in Gene Expression Barcode.

Next, evidence levels frequencies (*High, Moderate, Low*) were calculated across the GEB and HPA, as shown in Figure 5.6 using the thresholds of Table 5.3.

Only a small number of genes have similar evidence levels in both data sources. Furthermore, a significant number of genes have contradictory levels of evidence - genes with expression evidence in one data source and not expressed in the other.

Regarding the HPA data, only the information scored as ”supportive” was considered in this work. Despite the number of genes present in HPA and GEB repositories was higher, only the genes present in the template metabolic model are useful in the hepatocytes models reconstruction. The Figure 5.7 shows the overlap between the data sources and the genes present in Recon 1.

From the genes present in the Recon 1 model with information in GEB and HPA (supportive), there are 15% of genes with “High’” or “Moderate”

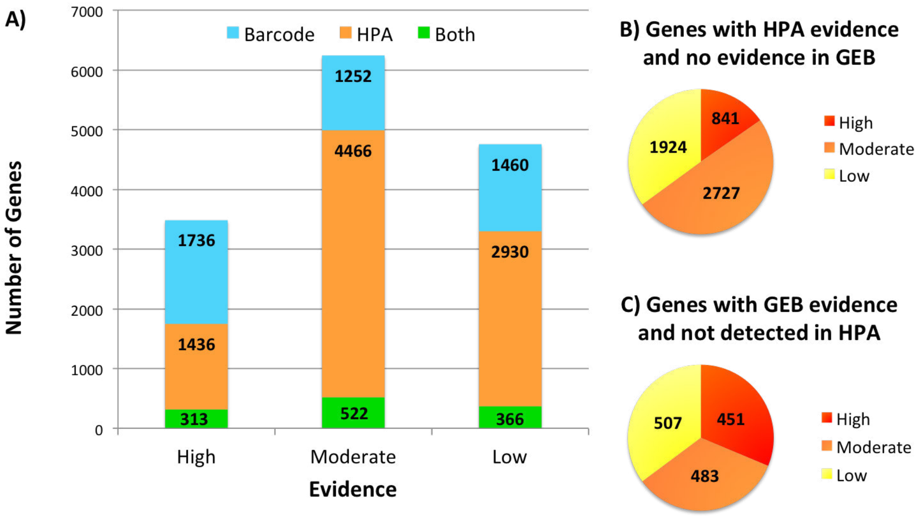


Figure 5.6: A) Distribution of genes from Gene Expression Barcode project and Human Protein Atlas across the evidence levels - “High”, “Moderate” and “Low”. The ranges [0.9, 1], [0.5, 0.9[and [0.1, 0.5[were used to classify the data into “Low”, “Moderate” and “High” levels. B) Genes with no evidence to be present in hepatocytes from GEB, but with evidence in the HPA. C) Genes with no evidence to be present in hepatocytes from HPA, but with evidence in GEB.

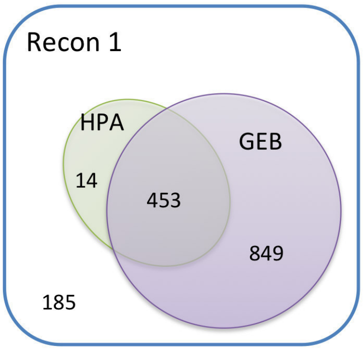


Figure 5.7: Number of metabolic genes present in the human metabolic models Recon 1 with evidence in HPA(suportive), GEB, both and none of the omic data types.

evidence in one of the sources and not expressed in the other. This number increases to 22% if we also consider “Low” evidence level.

As mentioned before, in the developed framework, all methods receive reaction scores calculated based on omics data. Thus, it was necessary to convert the gene expression evidence levels from GEB and HPA to reaction scores through the GPR associations. After this, the transformation impact of omics discrepancies in the values of reaction scores was analysed and those were compared to the manually curated set C_H from Jerby et al. [16].

In Figure 5.8 A, the poor overlap of the reaction scores calculated based on different sources can be observed. Considering all data sources and Recon 1 as generic model, 1903 reactions show some evidences that support their inclusion in the hepatocytes metabolic model, but only 386 are supported by all sources. The numbers are further dramatically reduced if we consider only moderate or high levels of evidence (Figure 5.8 B-C).

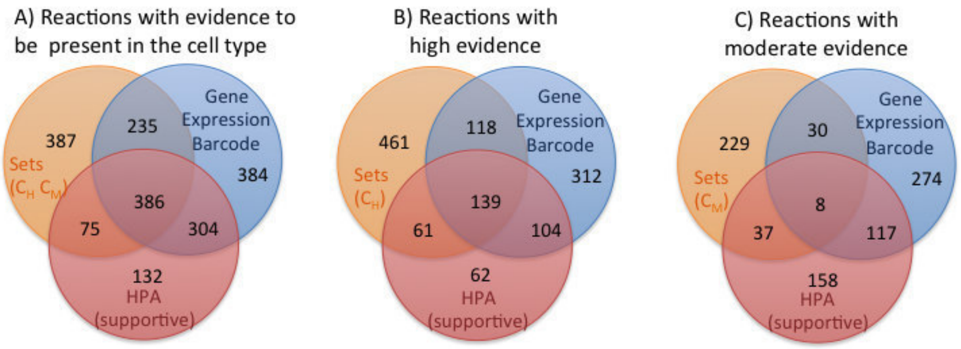


Figure 5.8: Overlap of reaction evidence levels for the three input data sources (C_H and C_M, GEB and HPA) A) Reactions with evidence that support their inclusion in the hepatocytes metabolic model. B) Number of reactions that have a high level of evidence of expression for each data source. C) Number of reactions that have a moderate evidence of expression for each data source.

5.3.2 Hepatocytes metabolic models

The resulting metabolic models have between 1178 and 2139 reactions. Table 5.4 presents the size of each model reconstructed in this study.

Table 5.4: Number of reactions for all hepatocytes metabolic models.

	C_H **and** C_M	**HPA**	**GEB**
MBA	1748	1246	1577
tINIT	1750	11837	2139
mCADRE	1760	1178	1511
FASTCORE	1817	1220	1542

The Figure 5.9 shows the relations between the 12 metabolic models generated through hierarchical clustering considering the Euclidean distance as measure. This was done using the `hclust` function on the R software.

The models obtained using the C_H and C_M sets as input data group together. Regarding the remaining, the mCADRE and MBA resulting models group according to their data (HPA and GEB), while the models created by tINIT cluster together (Figure 5.9). Overall, the data used as input seems to be the most relevant factor in the final result.

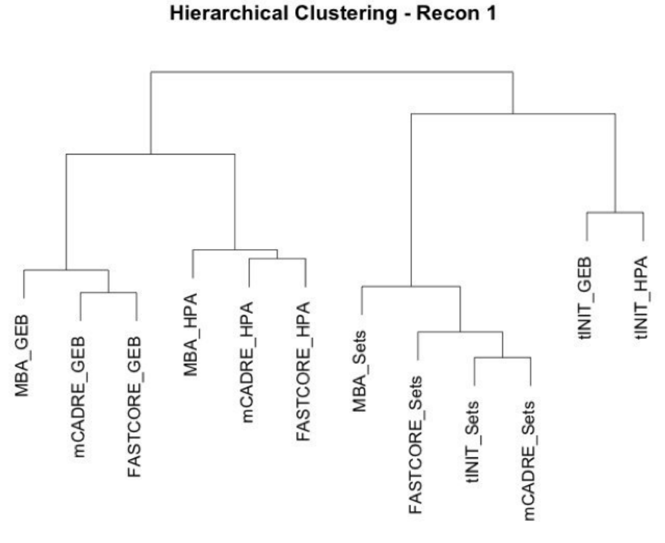


Figure 5.9: Results from hierarchical clustering of the resulting 12 models for each human generic metabolic model.

A more detailed comparison between the models reconstructed using the same algorithm or the same data source is available in Figure 5.10, A and B respectively. Considering the models generated by the same algorithm, it is observed that MBA has a smaller overlap (only 930 reactions) compared to the other methods. This could be explained by the fewer number of metabolic models generated for the reconstruction of the final consensus model.

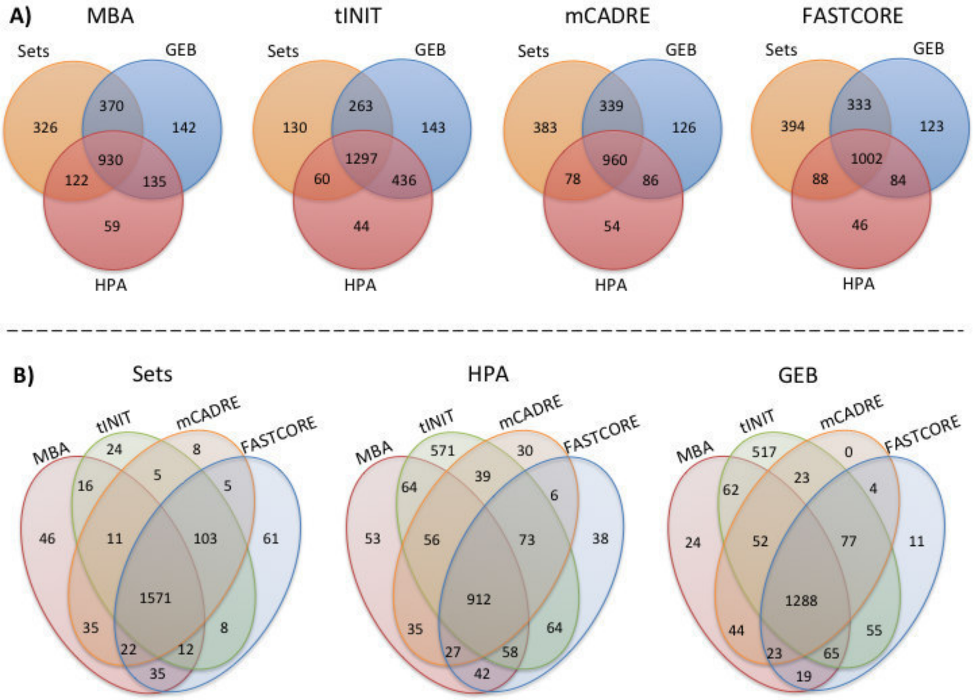


Figure 5.10: Hepatocytes metabolic models reaction intersection considering: (A) the same algorithm; (B) the same omics data source.

A lower number of reactions does not mean that the algorithm or data source have poor overlap. So, the correlation of model size and the number of reactions present in all models is presented in Table 5.5 to simplify the analysis of the models overlap.

The values presented above show that the same input data under different algorithms produces metabolic models with lower variance than using the same algorithm for different omics data type. Furthermore, the mean of reactions that belong to all models of the same algorithm is around 66%, and around 78% when the models are grouped by data source. Again, the

Table 5.5: Percentage of number of reactions of each model that are present in the intersection of models with the same omics data as input or algorithm.

Algorithm	**Input Data**	**$\cap$ Omics**	**$\cap$ Methods**
MBA	*Sets*	90%	53%
	HPA	73%	59%
	GEB	82%	75%
tINIT	*Sets*	90%	74%
	HPA	50%	71%
	GEB	60%	61%
mCADRE	*Sets*	89%	55%
	HPA	77%	81%
	GEB	85%	64%
FASTCORE	*Sets*	86%	55%
	HPA	75%	82%
	GEB	84%	65%

variability of the final results seems to be dominated by the data source factor.

5.3.3 Models validation

A set of metabolic tasks known to occur in hepatocytes cells was previously presented by Gille et al. [17]. Some of these tasks are impossible to satisfy with Recon 1 as template metabolic model, because they use metabolites which are not present in the model. Thus, these tasks and disease related tasks will not be considered in the validation process.

The generic Recon 1 human metabolic model is able to satisfy 281 of the remaining 363 metabolic functions tested. This set of 281 metabolic tasks was validated in each hepatocyte metabolic model to analyse the quality of the generated models. The Table 5.6 presents the model size and the percentage of tasks that remains successful in the tissue-specific model when compared with the generic metabolic model.

Here, it is clear that FASTCORE is able to produce consistent models independent of the input data. tINIT also has a significant percentage of valid tasks when the data source is HPA. However, generically the number of

Table 5.6: Percentage of liver metabolic functions that each metabolic model performs when compared with the template model - Recon 1.

	C_H **and** C_M	**HPA**	**GEB**
MBA	14%	8%	22%
tINIT	5%	70%	29%
mCADRE	3%	24%	24%
FASTCORE	54%	47%	67%

satisfied metabolic tasks is very low compared with the performance of the template metabolic model - Recon 1.

5.3.4 Final consensus model

The reconstruction process based on the combination of all models was done to achieve the final consensus hepatocytes metabolic model. The number reactions and the number of valid tasks satisfied by each of the partial models can be observed in Figure 5.11.

The $pModel_1$ contains all the reactions present in at least one of the 12 metabolic models. Furthermore, this *partial-model* is capable to satisfy all the metabolic tasks as the Recon 1. As can be seen in the Figure 5.11 the number of valid tasks decreasing between $pModel_6$ and $pModel_7$ is significant. So, $pModel_6$ was considered the starting point of the strategy of building the final model based on the models combination.

At the end, a metabolic model with 1.859 reactions was obtained. This model satisfies all the 281 metabolic tasks also satisfied by the template model Recon 1 but keeping only 50% of the reactions.

Finally, Figure 5.12 presents the relation between the models size and the number of satisfied tasks for all reconstructed models, including our hepatocytes final consensus model.

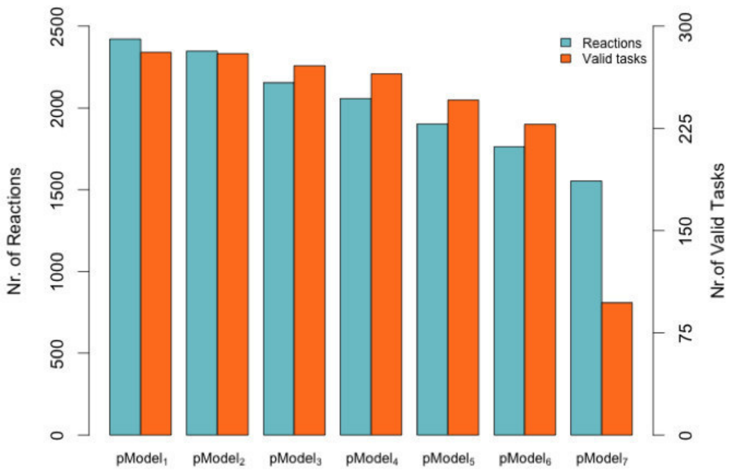


Figure 5.11: Each $pModel_i$ obtained by the 12 hepatocytes models combinations, where the index i represents the minimum number of models required for reactions to be present. Blue bars represent the number of reactions, and orange bars the number of tasks satisfied by the *partial-models*.

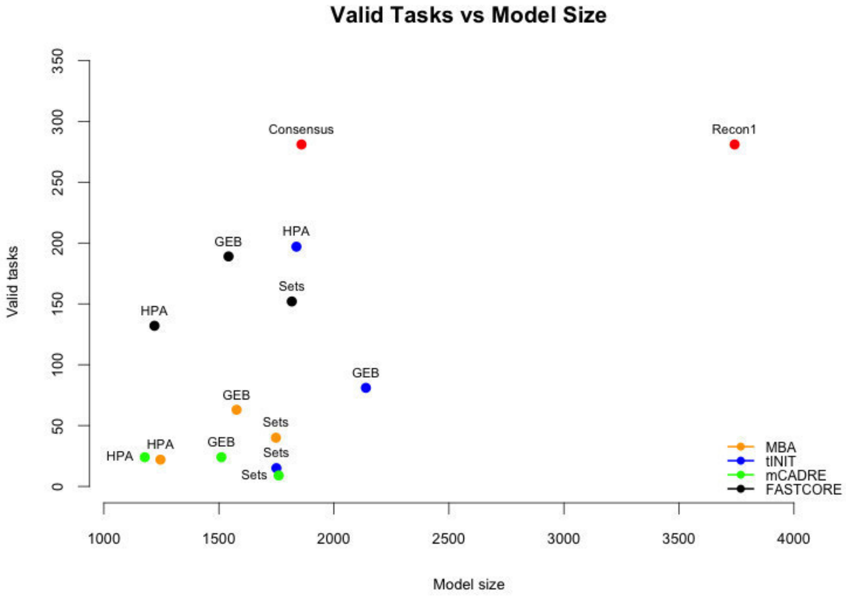


Figure 5.12: Correlation between tasks and models size.

5.4 Conclusion

In this chapter, a critical evaluation of the most important methods for the reconstruction of tissue-specific metabolic models was presented. Moreover, the consistency of information across important omics data sources was analysed and these data were used to verify the impact of such differences in the final metabolic models generated by each method.

The results show that metabolic models obtained depend more on the data sources used as inputs, than on the algorithm used for the reconstruction. To validate the accuracy of the obtained metabolic models, a set of metabolic functions that should be performed in hepatocytes was tested for each metabolic model. Generically, the number of satisfied liver metabolic functions was surprisingly low with exception of the models generated by FASTCORE and tINIT when HPA data was used as input .

This shows that methods for the reconstruction of tissue-specific metabolic models, based on a single omics data source, are not enough to generate high quality metabolic models. Here, it was also presented a strategy to build a final metabolic model using the combination of generated models through different algorithms and data sources. This process shows that with a similar number of reactions, it is possible to achieve a final model capable of satisfying all possible metabolic tasks. However, this strategy depends on metabolic functions knowledge which remains unknown for the most tissues / cell-types.

Methods to combine several omics data sources to rank the reactions for the reconstruction process could be a solution to improve the results of these methods. Indeed, this study emphasizes the need for the development of reliable methods for omics data integration, which seem to be required to support the reconstruction of complex models of human cells, but also reinforce the need to be able to incorporate known phenotypical data available from literature or human experts.

Chapter 6

Glioblastoma Model Reconstruction and Analysis

In this chapter, the reconstruction process of the U-251 cell line (a human cell line derived from a malignant glioblastoma tumor) metabolic model is described. The framework, described in previous chapters, together with the Recon 1 human metabolic model used as template model, and data retrieved from Human Protein Atlas and Gene Expression Barcode, were used to achieve the final model. Moreover, analyses were performed to validate the final model and compare the resulting model with other models already available for this cell line.

6.1 Cancer and glioblastoma

6.1.1 Hallmarks of cancer

Cancer is a collection of diseases characterized by unregulated cell growth and the invasion of other tissues/organs in the body [123]. Cancer cells present a huge number of genetic changes that contribute to the abnormal cell behaviour, specially in how they grow and divide when compared to normal cells. The mutations occurring in the genome can originate two major types of mutated genes that contribute to the development of cancer: oncogenes, allowing cells to grow and survive when they should not, and tumor suppressor genes with recessive loss of function [124]. In 2000, Hanahan and Weinberg [125] defined six hallmarks of cancer which comprise biological capabilities acquired during the development of cancer, described below:

- **Sustaining proliferative signaling:** normal cells control the growth and division cycle through growth-promoting signals, which contribute for the normal tissue architecture and function. In cancer cells, these signals are deregulated leading to unregulated growth.

- **Evading growth suppressors:** normally, cells respond to inhibitory signals to maintain homeostasis. In cancer, the acquired mutations interfere with the response to growth inhibitory signals.

- **Resisting cell death:** normal cells are eliminated by apoptosis (programmed cell death) when they suffer different types of DNA damage. Cancer cells have a variety of strategies to limit or circumvent apoptosis, being the loss of TP53 tumor suppressor function one of the most well known.

- **Enabling replicative immortality:** the number of cell divisions is finite and controlled by the shortening of chromosomal ends, telomeres, that occurs during DNA replication. Cancer cells maintain the length of telomeres, which allows the unlimited replication of the cells.

- **Inducing angiogenesis:** cells depend on blood vessels to supply oxygen and nutrients. In normal cells, the vascular architecture remains mainly constant in adults. However, the formation of new vessels is essential for tumor growth and survival.

- **Activating invasion and metastasis:** mutations in genes involved in the cell-cell and cell-extracelular adhesion allow the movement of cancer cells to other parts of the body. This is a major cause of cancer death, since the disease is not in a specific organ, but spread over the whole body.

Recently, the authors added two emerging hallmarks of cancer to the previous list [126]. The first is the reprogramming of energy metabolism by cancer cells, while the second is the capacity of cancer cells to avoid the attack and elimination by immune cells. Additionally, two new enabling characteristics were also added by the authors: the genome instability and mutations, and the tumour-promoting inflammation. Both characteristics contribute for the acquisition of hallmark capabilities by cells.

Under aerobic conditions, normal human cells process glucose on mitochondrial oxidative phosphorylation to generate the energy required by cellular processes. When oxygen is limited, cells can redirect the pyruvate generated by glycolysis to produce lactate, instead of the oxidative phosphorylation. Cancer cells tend to convert glucose to lactate even when oxygen is present. This anomalous characteristic of cancer cell energy metabolism was observed by Otto Warburg [127], and the phenomenon is known as the Warburg Effect [128] (Figure 6.1).

6.1.2 Glioblastoma

Glioblastoma (GBM), also known as astrocytoma grade IV, is the most common and aggressive type of brain cancer in adults [129]. Based on their clinical and biological characteristics, GBMs can be divided into two categories [130]. Primary GBMs are the most common, being characterized by the amplification and mutations in the *EGFR* gene and the deletion of the *PTEN* and *CDKN2A* genes [131]. The protein encoded by the *EGFR* gene is a receptor for members of the epidermal growth factor family, which leads to cell proliferation. The *PTEN* and the *CDKN2A* genes are known to be important tumor suppressor genes [132, 133]. Secondary GBMs, contrary to the previous category, affect younger patients who had been affected

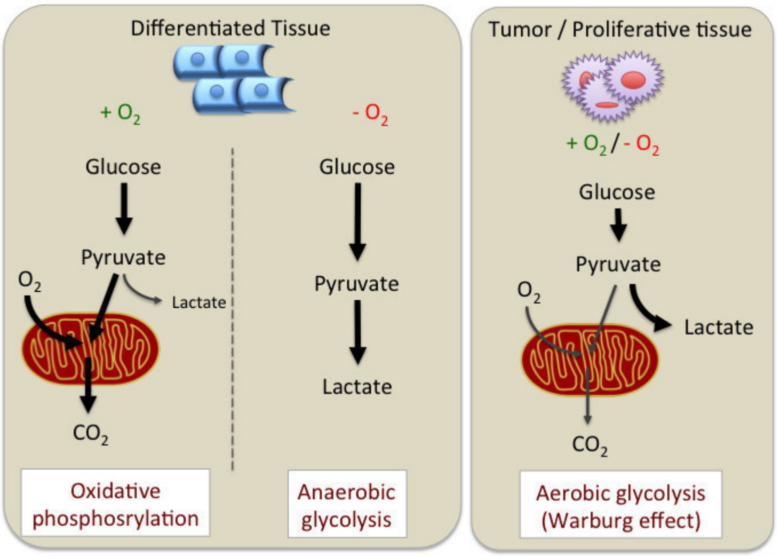


Figure 6.1: Schematic representation of oxidative phosphorylation, anaerobic glycolysis, and aerobic glycolysis, also known as the Warburg effect. In the presence of oxygen, normal cells metabolize glucose via oxidative phosphorylation. When oxygen is limited, cells redirect the pyruvate to lactate production. Cancers cells tend to convert most of glucose to lactate even in the presence of oxygen (aerobic glycolysis). Figure adapted from [128].

by a lower grade astrocytoma before. These GBMs are characterized by mutations in the *TP53* gene and overexpression of PDGFR [131]. Several studies have identified alterations in the *IDH1/2* genes (encode the cytosolic and mitochondrial isoforms of NADP$^+$-dependent isocitrate dehydrogenases), that are also observed in secondary GBMs [129, 134, 135, 136].

These molecular abnormalities are present in both categories, but with different frequencies. As an example, the frequency of *TP53* mutation in secondary GBM is more than 65%, but only 28% in primary GBM [137].

6.2 Phenotype simulation

The methods implemented in the developed framework for phenotype simulation (GIMME, iMAT and E-Flux), described in sections 2.5 and 3.5, were used to perform phenotype prediction of glioblastoma cells, under different conditions, using the Recon 1 as metabolic model and the transcriptomic data retrieved from [138]. These data set were also available in GEO with the accession identifier GSM803632. The biomass equation used in the simulations was taken from the Recon 2 metabolic model.

The biomass flux rate obtained using the FBA simulation method was of 0.084 $mmol/gDW/hr$, using the RPMI-1640 medium as described in [139]. The phenotype simulations given by GIMME and iMAT also took into consideration this medium. The E-Flux algorithm formulation assumes the value -1 as the lower bound for all uptake fluxes, so the medium is not considered in the phenotype simulation.

In chapter 4, it was observed that the phenotype predictions using pFBA have a lower prediction error associated, when compared with the other simulation methods. Similarly to this previous study, we compare the flux exchange rates obtained by different methods with the experimental values taken from [140]. This data set contains the measurements, obtained using mass spectrometry, of consumption and release profiles of 219 metabolites from the medium across the NCI-60 cancer cell lines. From all measured metabolites, only 36 were considered in this analysis, since these have an exchange reaction associated in the Recon 1 metabolic model. All values from the simulation results were normalized by the glucose uptake.

The normalized prediction errors for each method are presented in Table 6.1. Once again, the estimation error was calculated using the equation:

$$error = \frac{\|v^{exp} - v^{sim}\|}{\|v^{exp}\|} \tag{6.1}$$

where v^{exp} is the vector of measured flux values and v^{sim} is the vector of predicted values.

In chapter 4, and also in Machado et al. [111], the pFBA method achieves better results when compared with the other simulation methods. However, this is not observed in this case. Here, the GIMME and iMAT methods have a better prediction capability when compared with pFBA. So, it seems

Table 6.1: The normalized prediction errors, associated to the simulation methods pFBA, GIMME, E-Flux and iMAT, for glioblastoma phenotype prediction using the Recon 1 as a metabolic model.

	pFBA	**E-Flux**	**GIMME**	**iMAT**
Error	0.9152	1.0004	0.7484	0.7598

that transcriptomic data can play an important role in the improvement of phenotype predictions of metabolic models, at least in some cases.

6.3 Tissue-specific model reconstruction

Recon1 was used as the template model to the glioblastoma metabolic model reconstruction. The main reasons for this choice are related to the size of the model, being the time consummed to generate the tissue-specific models much lower than using other metabolic models as Recon 2, and the possibility to easily compare the resulting model with already published glioblastoma metabolic models [21, 23]. The input data used by the reconstruction algorithms present in our framework were retrieved from HPA and GEB databases. We used these two data sources in combination with four reconstruction methods to achieve the final U-251 metabolic model.

6.3.1 Omics data sources

The reconstruction of the U-251 metabolic model starts with the collection of information from omics databases. HPA and GEB have transcriptomics and proteomics evidences for this cell line.

The Recon 1 metabolic model, used as template in the reconstruction of tissue-specific models, has 1905 genes. The HPA and GEB databases have information for 1335 and 1293 genes from the Recon1. The Figure 6.2-A shows that most of the genes are present in both databases. However, if we take into account the expression evidence levels, the number of genes and reactions with the same evidence level is surprisingly low (Figure 6.2-B,C). The gene expression levels present in the omics data sources were converted to reaction evidence levels through the gene-protein-reactions(GPR) rules present in the metabolic model. During the conversion, the operators AND/OR present in the GPRs are substituted by the MIN/MAX functions, respectively.

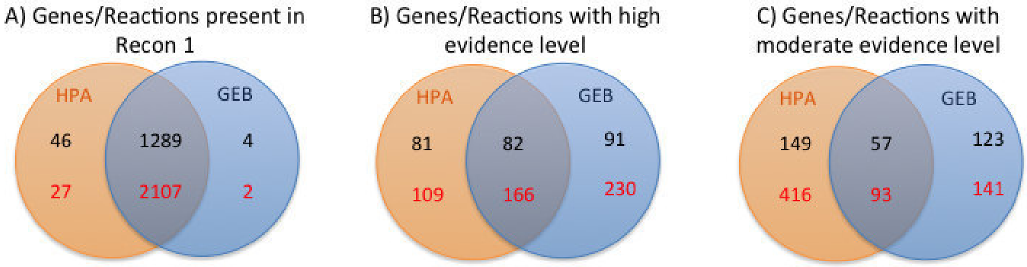


Figure 6.2: Genes with expression evidence for U-251 cell line in Human Protein Atlas (HPA) and Gene Expression Barcode (GEB). The red numbers represent the number of reactions with evidence to be active (using the Gene-Protein-Reaction rules present in the model) and the black numbers the number of genes. A) Intersection of genes present in both data sources and in the Recon 1 metabolic model. B, C) Recon 1 genes and reactions with high (B) and moderate(C) evidence to be expressed in the U-251 cell line.

The lower overlap in the high and moderate sets of reactions considering the cutoffs of "High"/ 0.9 and "Medium"/ 0.5 from data retrieved from HPA/GEB can have a significant impact in the resulting models, independently of the used algorithm.

The reconstruction of the tissue-specific metabolic models was done considering the cutoffs already present in Table 5.3 on chapter 5.

6.3.2 U-251 metabolic models

Following the same approach used on chapter 5, eight U-251 cell line metabolic models were created. Each model was reconstructed using one of the available algorithms in your framework (FASTCORE, MBA, mCADRE and tINIT) and an omics data source (HPA and GEB). The size of each model is presented on Table 6.2.

Table 6.2: Number of reaction of each U-251 metabolic model.

Algorithm	Data Source	Reactions
MBA	HPA	1563
	GEB	1752
mCADRE	HPA	1170
	GEB	1110
tINIT	HPA	2048
	GEB	1146
FASTCORE	HPA	1219
	GEB	1137

Figure 6.3 shows the overlap of the resulting models from the different methods, when each of the omics data source was considered.

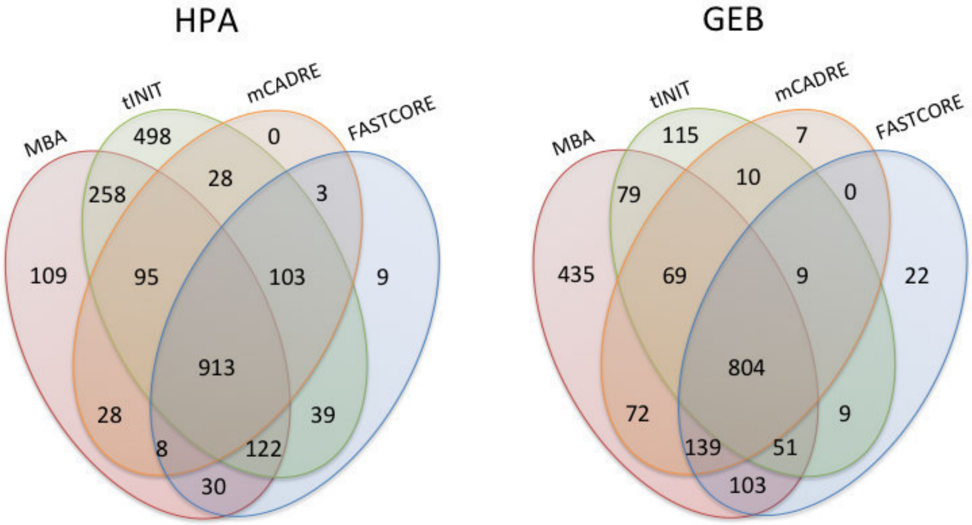


Figure 6.3: Reactions overlap of U-251 metabolic models grouped by data source (HPA and GEB).

Comparing the U-251 models, reconstructed with the same data source, tINIT and MBA algorithms produce models with a higher number of exclusive reactions, i.e., reactions present in a single model. The number of reactions shared by all models for each data source is similar, 913 and 804 for HPA and GEB data sources, respectively. However, the intersection of these two sets is only of 577 reactions.

Figure 6.4 presents the intersection of these same models, but considering the algorithm instead of the data source as categories.

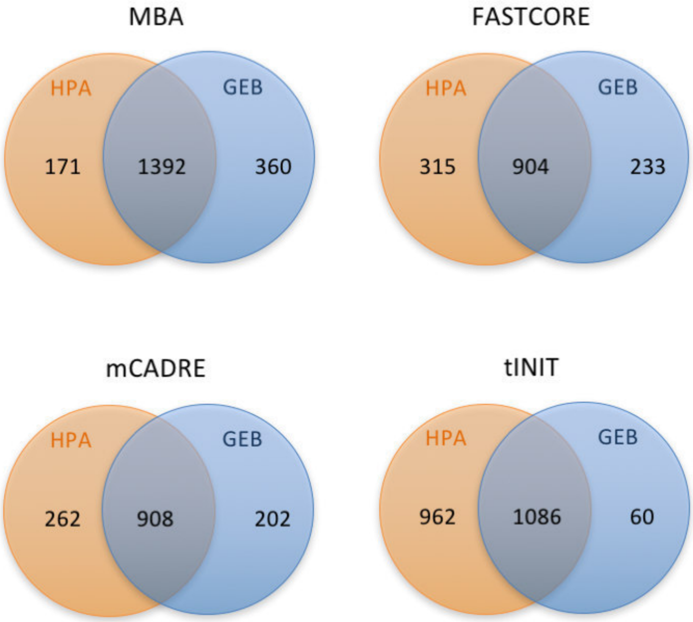


Figure 6.4: Reactions overlap of U-251 metabolic models grouped by algorithms (MBA, FASTCORE, mCADRE and tINIT).

Considering the models generated by the same algorithm, it is observed that MBA has the highest overlap when compared to the other methods. However, with exception of tINIT_HPA metabolic model, the MBA models have a significant increase in the number of reactions, when compared with other models. The mCADRE and FASTCORE models have a similar number of reactions and the models generated by HPA and GEB are also of similar size. In order to explore the similarity of the models, the Figure 6.5 depicts the hierarchical clustering of the U-251 metabolic models.

Figure 6.5 presents the intersection of these same models, but considering the algorithm, instead of the data source as a discriminant factor.

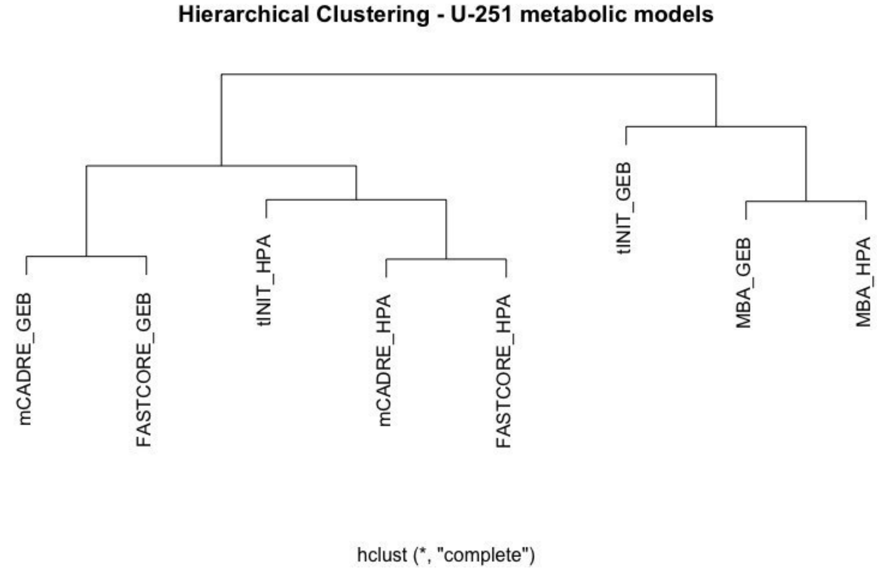


Figure 6.5: Hierarchical clustering of U-251 metabolic models.

The mCADRE and FASTCORE models are grouped first by data source, and then by the algorithm used in the reconstruction process. The MBA models depend more on the algorithm than on the data source used to build the U-251 models. The models reconstructed by tINIT belong to different branches of the tree, showing that the data source used as input can have a huge influence on the result.

One of the hallmarks of cancer is the capability that cancer cells have to proliferate. To address this issue, FBA simulations were done to test the biomass production of each model. The biomass equation was collected from the Recon 2 metabolic model and the RPMI-1640 medium [139] has been considered in all simulations. As a result, none of the models was able to produce biomass. So, we tested how many biomass precursors could be produced in each metabolic model, by adding additional reactions to excrete each biomass precursor and simulating the maximization of these reactions. Table 6.3 presents the number of biomass precursors produced by each of the U-251 metabolic models.

Table 6.3: Number of biomass precursors produced by each of the U-251 metabolic models. The biomass equation was obtained from Recon 2 metabolic model which contains 38 precursors metabolites.

Algorithm	Data Source	Nr. of Precursors	% of Precursors
MBA	HPA	26	68%
	GEB	25	68%
mCADRE	HPA	5	13%
	GEB	12	32%
tINIT	HPA	31	82%
	GEB	17	45%
FASTCORE	HPA	19	50%
	GEB	16	42%

The U-251 model generated by the tINIT algorithm using HPA data has the highest number of biomass precursors satisfied. This is expectable since this model has approximately 500 more reactions than the remaining models.

Given these results, the reconstruction of a single, unified and global U-251 metabolic model is required. The final metabolic model must be able to carry flux on the biomass reaction, to allow to simulate the proliferation of cells, predicting growth rate.

6.3.3 Consensus model

The final U-251 metabolic model was built considering all previously reconstructed models by different methods and data sources. The process, already detailed in section 5.2.6, starts with the reconstruction of the partial models ($pModel_i \quad i \in 1, ..8$). Each partial model ($pModel_i$) contains the reactions present in at least i U-251 metabolic models. Thus, the $pModel_5$ for instance contains all reactions present in five or more models from the set of eight models.

Here, the tasks are defined by the production of biomass precursors. So, we checked how many biomass precursors each partial model was able to produce. Figure 6.6 depicts the number of reactions and the number of biomass precursors produced by each *pModel*.

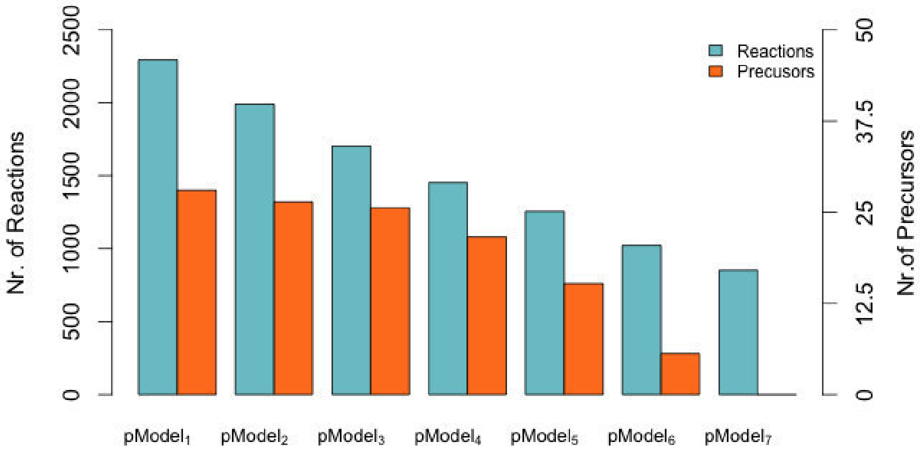


Figure 6.6: Number of reactions (green bars) and number of biomass precursors that can be produced (orange bars) by the $pModels$. The $pModel_8$ was ignored since the previous $pModel$ does not produce any of the biomass precursors.

As can be observed in the picture, the highest decrease on the number of produced biomass precursors occurs between $pModel_4$ and $pModel_5$. Thus, the process of reconstructing the final consensus model starts with $pModel_4$ as the initial model. In each iteration, the algorithm takes as input a partial model ($pModel_i$) and tries to remove the maximum number of reactions that were lost between $pModel_i$ and $pModel_{i+1}$, maintaining the biomass precursors produced by $pModel_i$ and not by $pModel_{i+1}$. In this case, the lost reaction set (LRS) and the lost biomass precursors set (LBS) are composed by the difference between the two partial models $pModel_4$ and $pModel_5$. At the end of each iteration, the reactions that do not have an influence in the loss of biomass precursors between two partial model ($toDel_i$), are appended to the LRS in the next iteration. The process ends with the processing of $pModel_0$, in this case the full Recon 1 model. Figure 6.7 shows the steps in the reconstruction of the consensus final model.

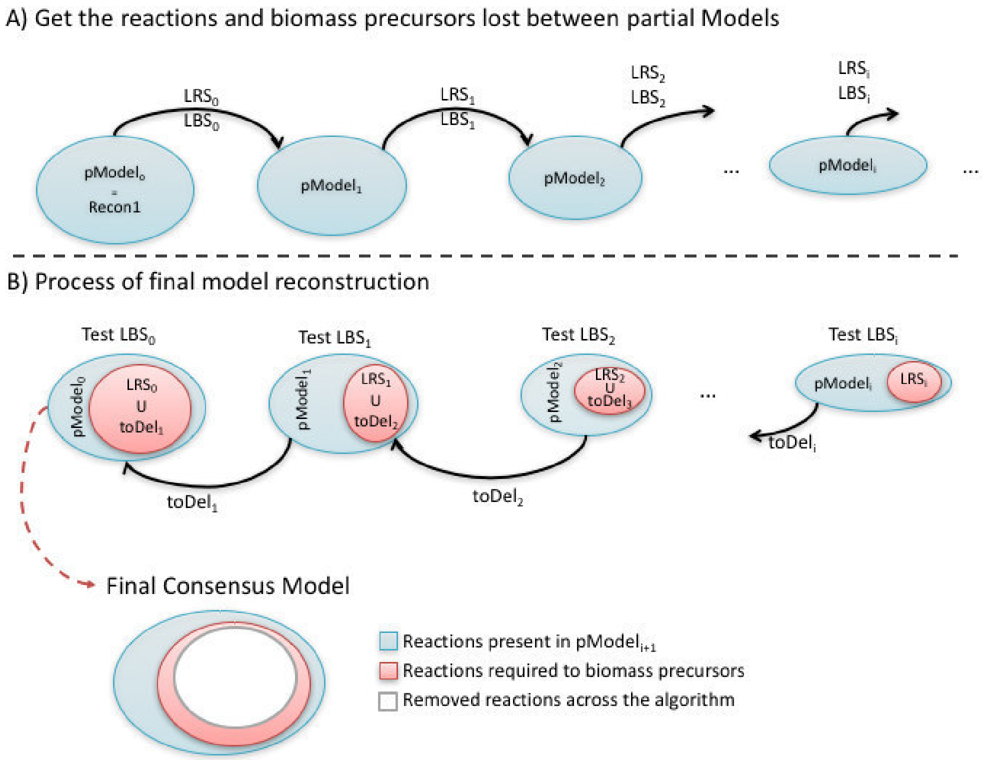

A) Get the reactions and biomass precursors lost between partial Models
LRS₀
LBS₀
LRS₁
LBS₁
LRS₂
LBS₂
LRSᵢ
LBSᵢ
pModel₀
=
Recon1
pModel₁
pModel₂
...
pModelᵢ
...
B) Process of final model reconstruction
Test LBS₀
Test LBS₁
Test LBS₂
Test LBSᵢ
pModel₀
LRS₀
U
toDel₁
pModel₁
LRS₁
U
toDel₂
pModel₂
LRS₂
U
toDel₃
...
pModelᵢ
LRSᵢ
toDelᵢ
toDel₁
toDel₂
Final Consensus Model
🟦 Reactions present in pModelᵢ₊₁
🟥 Reactions required to biomass precursors
□ Removed reactions across the algorithm

Figure 6.7: A) Build the lost reaction set (LRS) and the lost biomass precursors set (LBS) between each par of partial models. B) For each $pMode_i$ the process find the reaction from LRS_i that can be removed from $pModel_I$ without affect the biomass precursors (LBS_i) production. The set of reactions that can be removed will be added to LRS set in the next iteration.

The final consensus model obtained is composed of 922 genes, 1.376 metabolites and 1.457 reactions. This model is able to simulate the biomass production, through FBA, using the RPMI-1640 medium [139]. The flux rate for biomass equation is around 0.0291 $mmol/gDW/hr$. Although the lower biomass flux rate, when compared with the original model Recon 1 (0.084 $mmol/gDW/hr$), this process is able to achieve a final consensus model based in all previous models capable to simulate the biomass production.

6.4 Critical genes

The validation of metabolic models is a hard task when fluxomics data are not available. So, some tests were done to check if the consensus metabolic model has a better phenotype prediction capability than the global model Recon 1.

As a first validation, we calculated the predicted critical genes of both models. We considered critical genes as the genes that inhibit growth when they are removed from the model. We obtained these gene sets through FBA simulations, when each gene present in the model was knocked out, i.e. the reactions associated through GPRs to this gene were constrained to have no flux. At the end, the final consensus model of the U-251 cell line has 89 critical genes, of which 80 are also critical genes in Recon 1. Thus, nine genes are only critical on the U-251 metabolic model - *G6PT2, SLC5A7, NME2, NME1, SLC6A14, PTDSS1, SLC16A10, CDS1, CTPS.* Remarkably, most of these genes have been associated to cancer cell growth in several research studies. The function and the relevance of these genes on glioblastoma cancer cells are detailed next:

- The **G6PT2** gene regulates the Glucose-6P transport from cytoplasm to the lumen of the endoplasmic reticulum. Studies demonstrate that intracellular signalling and invasive phenotype of brain tumor cells could be regulated by this gene [141]. Moreover, silencing the G6PT gene in U-87 brain tumor-derived glioma cells induce necrosis and late apoptosis [142]. Thus, control of the G6PT expression can lead to the development of new strategies to prevent cancer development in glial cells.
- The **SLC5A7** gene encodes a high-affinity choline transporter. Choline is used for the synthesis of essential lipid components of cell membranes [143]. A higher choline concentration in the cells has been related with cell proliferation and malignant progression of cancer [144, 145] being the abnormal choline metabolism considered, by Glunde et al., as a new hallmark of cancer [146]. Kumar et al. [147] demonstrate that using

specific choline kinase inhibitors may be a promising new strategy for treatment of brain tumors.

- The **SLC6A14** gene encodes the protein called *sodium- and chloride-dependent neutral and basic amino acid transporter B(0+)* which can transport all essential amino acids, as well as glutamine, arginine, and asparagine [148]. Cancer cells, to support their rapid cell growth, induce the over-expression of this gene. This phenomenon has been observed in cervical cancer, colorectal cancer and breast cancer cell lines [149, 150, 151]. The SLC6A14 deletion was studied in mouse models of breast cancer by Badu et al. [152]. The study demonstrated that the development and progression of breast cancer were markedly decreased *in vitro* and *in vivo* when SLC6A14 is deleted.

- The **CDS1** is a protein coding gene which regulates the amount of phosphatidylinositol available for signaling by catalyzing the conversion of phosphatidic acid to CDP-diacylglycerol. CDP-diacylglycerol is an important precursor for the synthesis of phosphatidylinositol (PtdIns), phosphatidylglycerol, and cardiolipin [153, 154]. The cardiolipin compound is one of the biomass precursors present in Recon 2 biomass equation. Thus, its production is essential.

- The **PTDSS1** gene encodes phosphatidylserine synthase 1 (PSS1) which is involved in the production of phosphatidylserine. This gene is involved in a patent related to the development of a molecular-based method of cancer diagnosis and prognosis. Together with five others genes, the PTDSS1 has a higher expression in tumor samples when compared with control samples [155].

- The **CTPS** gene encodes an enzyme responsible for the conversion of UTP (uridine triphosphate) to CTP (cytidine triphospate). The development of methods and pharmaceutical compositions to inhibit the lymphocyte proliferation through the CPTS1 inhibitors has been protected by a patent [156].

- The **NME2 / NME1** genes were identified as potential tumor suppressors, which reduce the tumor progression and proliferation [157]. Thus, it was unexpected that these genes were essential for the metabolic model. To understand this result, we did a deep analysis of the reactions where these genes are involved. The two genes regulate the activation of nucleoside-diphosphate kinase reactions in the nucleus. These reactions are responsible to produce essential metabolites present in biomass equation, namely Deoxyguanosine triphosphate (dGTP), Deoxycytidine triphosphate (dCTP), Deoxyadenosine triphosphate (dATP) and Deoxythymidine triphosphate (dTTP). These metabolites are used in cells for DNA synthesis.

6.5 The Warburg effect

The phenomenon known as "Warburg effect" consists in the capability that cancer cells have to generate the energy needed for cellular processes through aerobic glycolysis instead of the oxidative phosphorylation, as normal cells do [158]. The aerobic glycolysis is an inefficient way to achieve ATP production (2 ATP molecules per one glucose molecule), when compared with the oxidative phosphorylation (32 ATP molecules per one molecule of glucose), leading to lactate secretion [159].

In 2011, Shlomi et al. [12] argued that the Warburg effect is a consequence of the metabolic adaptation of cancer cells to increase biomass. They developed a new simulation method based on FBA which accounts for the enzyme solvent capability as a constraint. In the study, it was clear that the three phases (optimal, intermediate and low yield metabolism) observed experimentally during oncogenic progression can be observed in the *in silico* simulations.

Using the same approach, we tested if our model was able to simulate the lactate secretion even in the presence of oxygen. Therefore, we used our U-251 consensus model with the RPMI1640 medium as before, with different amounts of glucose uptake between 0 and the uptake value needed to reach the maximal growth rate (0.14 $mmol/gDW/h$). The molecular weights and turnover numbers used for the Recon 1 metabolic model reactions were obtained from the original publication [12].

Using the FBA with solvent capacity constraints, the biomass yield decreases at high growth rates, as shown in Figure 6.8.

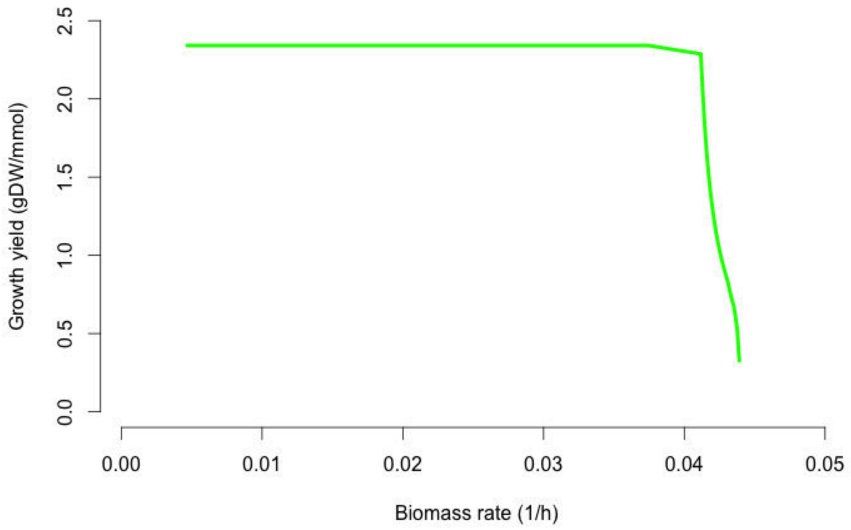


Figure 6.8: Predicted maximal growth yield of U-251 cell line (per unit of glucose uptake) for a range of predicted growth rates obtained by simulation of an extension of FBA which considers enzyme solvent capacity of the cells [12].

Considering the lactate secretion and the oxygen consumption fluxes for the range of growth rates, it is visible that the lactate production occurs even in the presence of oxygen (Figure 6.9).

In the figure, three different phases in the growth yield are clear:

(i) *Optimal yield* - characterized by the absence of lactate production.

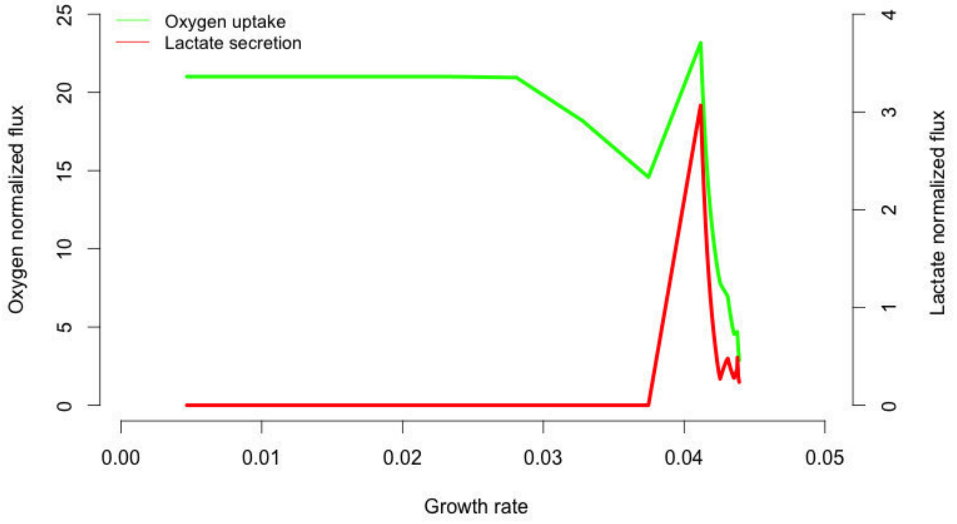


Figure 6.9: Predicted lactate secretion flux (red line) and oxygen uptake flux (green line) for a range of growth rates. Growth rates were obtained by varying the glucose uptake rate limit from 0.0 to 0.14 $mmol/gDWh/h$. The maximal growth rate is obtained when the glucose uptake is around of 0.1347 $mmol/gDWh/h$. Flux values were normalized by the glucose uptake rate.

Even with a small decrease of oxygen uptake the growth yield remains constant.

(ii) *Small decreasing in yield* - in this phase, the growth yield has a small decrease when compared with the previous phase. Moreover, the lactate production has a significant increase and oxygen also increases, reaching higher values when compared with the oxygen consumption values from the previous phase.

(iii) *Low yield* - characterized by a sharp decrease in oxygen consumption, lactate production fluxes and also growth yield. The method with solvent capacity constraints used in the simulation could be the reason for the decreasing of these fluxes (since the objective is the biomass production all reactions that are not essential to biomass production will decrease to minimum levels).

As a conclusion, the present model (consensus U-251 model) with the incorporation of solvent capacity constraints lead to refined predictions of cancer metabolic phenotypes, such as the Warburg effect.

6.6 Other tissue-specific metabolic models

Glioblastoma GSMMs were already reconstructed in previous studies [23, 21]. In this section, the overlap and a functional analysis between our model and the previous models, also generated considering Recon 1 as a template, are presented. The glioblastoma tumor cells and U-251 cell line GSMMs reconstructed by mCADRE and PRIME algorithms respectively, were used to perform the comparison with our consensus model. The overlap between all glioblastoma metabolic models is provided in Figure 6.10.

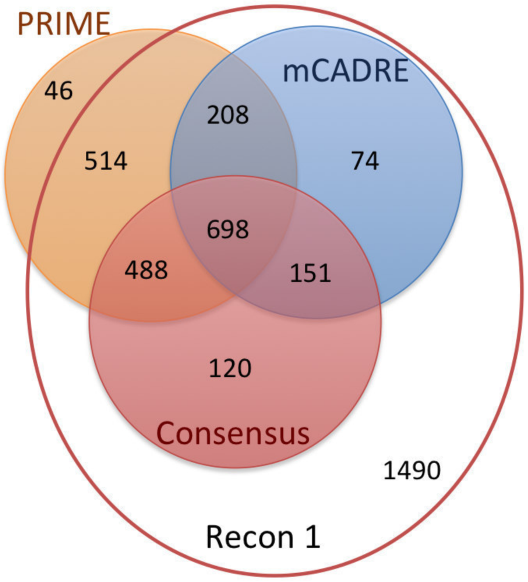


Figure 6.10: Overlap of metabolic models. The PRIME and mCADRE models are available in the methods publication articles. The consensus model is our model, reconstructed during this study.

Analyzing the model obtained by PRIME, we verified that the Recon 1 template model used by the algorithm is not the original model, but an extended version which has 46 extra reactions. These reactions are essentially for excretion of cytosol metabolites which can lead to significant differences in the phenotype simulation results. The models PRIME, mCADRE and Consensus are composed by 1952, 1131 and 1457 reactions respectively.

Next, we performed the phenotype prediction using the simulation methods present in our framework (pFBA, iMAT, GIMME and E-Flux). Transcriptomics data published by Gholami et al. [138] were used as input in the simulation methods (the same data used in section 6.2). Experimental flux values publised by Jain et al. [140] (also used in section 6.2) were used to compare with the flux exchange rates obtained by different methods and the normalized prediction errors were calculated using the equation 6.1.

In this study, the glioblastoma tumor cells metabolic model obtained by the mCADRE reconstruction method was not considered, because this model is not able to grow when simulated using the Recon2 biomass equation, even with the removal of the metabolites present in the biomass equation and not in the model.

The normalized prediction errors are given in Table 6.4.

Table 6.4: The normalized prediction errors, associated with the simulation methods pFBA, GIMME, E-Flux and iMAT, for U-251 model reconstructed by PRIME and consensus U-251 model.

	pFBA	**E-Flux**	**GIMME**	**iMAT**
Consensus	1.2866	0.7427	0.7639	0.7102
PRIME	0.7849	2.9739	0.7481	0.7461

Most of the method and model combinations have a normalized error around 0.7. The best combination, reaching a lower prediction error was obtained with the consensus U-251 metabolic model developed during this study using the iMAT simulation method.

6.7 Conclusion

In this chapter, the reconstruction process of our gliobastoma metabolic model was presented. Several methods and data sources were used to reconstruct metabolic models for the U-251 cell line (derived from a malignant glioblastoma tumor). The final model here presented was constructed based on all these models and taking the biomass production, retrieved from Recon 2 human metabolic model, as a requirement.

Taking this model as reference, we calculated the list of essential genes for the cell growth, and validated their function in published data. Most of the genes have been associated with tumor growth inhibition in the literature.

Our glioblastoma metabolic model has also the capability to predict the Warburg effect when the model is simulated by an extension of FBA, which accounts for the enzyme solvent capability as a constraint. Moreover, this model presents better results than other published model [23] when it is simulated with transcriptomic data, and the predicted flux distribution has a lower error comparing with experimental measurements for external metabolites.

Based on these results, our automatically generated glioblastoma metabolic model could represent a good starting point to achieve a curated metabolic model with good phenotype predictions.

Chapter 7

Conclusions and Future Work

In the final chapter of this thesis, the main conclusions of this work are presented. Some topics for future work are put forward.

7.1 General conclusions

The work developed along this thesis had as main goal the development of a framework for the reconstruction and analysis of tissue-specific metabolic models. Additionally, three of the most used phenotype prediction methods, published in recent years were also implemented, as well as a set of methods that allow loading and integrating omics data with the genome-scale metabolic models.

The clear division between the two layers, omics data processing and simulation/reconstruction methods, in the framework allows to use different omics data sources with the implemented methods. Moreover, the developed plug-ins in the open-source OptFlux platform make it an attractive resource to an ever increasing ME community.

The initial evaluation of the phenotype simulation methods implemented in the framework confirmed the results from [111]. In both case studies, the pFBA method has a lower prediction error than the iMAT, GIMME and E-Flux methods, taking into account 36 experimental measured fluxes.

Next, the critical evaluation of the methods for the reconstruction of tissue-specific models showed that the omics data sources used in the building process have more impact in the final result than the method itself. This emphasizes the need for the development of reliable methods to integrate and compile information from different data sources.

Furthermore, the results reveal that, for a specific case of hepatocytes cells, none of the methods was capable of originating a tissue-specific model which satisfies all the metabolic tasks performed by the template model and related with the liver function. So, a strategy to build a consensus final metabolic model using the combination of generated models through different algorithms and data sources was developed to improve the prediction capability of the final model. However, this strategy depends on metabolic functions knowledge which remains unknown for most tissues / cell-types.

Finally, we reconstruct a metabolic model for U-251 cell line , targeting the understanding of metabolic alterations related glioblastoma, one of the most aggressive tumors in humans. The final model was reconstructed based on metabolic models obtained using different methods and data sources. This model achieves better results when compared with other models for the same phenotype. The automatic reconstruction of a consensus model shows that is a good starting point to achieve a curated metabolic model with good phenotype predictions.

In summary, the developed framework helps in the reconstruction of tissue-specific metabolic models and allows the usage of phenotype prediction methods by common users through plug-ins in OptFlux. Additionally, the development of new methods by programmers can be easily done by extending the current framework.

7.2 Topics for future work

The framework and related plug-ins developed during this thesis provided a valuable contribution for the systems biology community. Nevertheless, some topics can be explored in future work:

- Development of new methods for the reconstruction of tissue-specific models based on meta-heuristics from the field of Evolutionary Computation, since these allow the competition of hypothetical models and the definition of flexible objective functions.
- From a software development perspective, an ongoing objective is the development of a new plug-in to support the methods for the tissue-specific reconstruction methods. At the moment only programmers, or at least users with a good working knowledge of command-line tools, are able to use the framework to reconstruction of tissue-specific metabolic models.

Bibliography

[1] Smallbone, K., Simeonidis, E., Swainston, N., Mendes, P.: Towards a genome-scale kinetic model of cellular metabolism. BMC Systems Biology 4(1), 1 (2010)

[2] Kauffman, K.J., Prakash, P., Edwards, J.S.: Advances in flux balance analysis. Current Opinion in Biotechnology 14(5), 491–496 (Oct 2003)

[3] Rocha, M., Maia, P., Mendes, R., et al.: Natural computation metaheuristics for the in silico optimization of microbial strains. BMC bioinformatics 9(1), 1 (2008)

[4] Feist, A.M., Palsson, B.Ø.: The growing scope of applications of genome-scale metabolic reconstructions using escherichia coli. Nature biotechnology 26(6), 659–667 (2008)

[5] Barghini, P., Di Gioia, D., Fava, F., Ruzzi, M.: Vanillin production using metabolically engineered escherichia coli under non-growing conditions. Microbial cell factories 6(1), 13 (2007)

[6] McCloskey, D., Palsson, B.Ø., Feist, A.M.: Basic and applied uses of genome-scale metabolic network reconstructions of escherichia coli. Molecular systems biology 9(1), 661 (2013)

[7] Duarte, N.C., Becker, S.A., Jamshidi, N., et al.: Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proceedings of the National Academy of Sciences of the United States of America 104(6), 1777–1782 (2007)

[8] Hao, T., Ma, H.W., Zhao, X.M., Goryanin, I.: Compartmentalization of the Edinburgh Human Metabolic Network. BMC bioinformatics 11, 393 (Jan 2010)

[9] Agren, R., Bordel, S., Mardinoglu, A., et al.: Reconstruction of Genome-Scale Active Metabolic Networks for 69 Human Cell Types and 16 Cancer Types Using INIT. PLoS Computational Biology 8(5), e1002518 (May 2012)

[10] Thiele, I., Swainston, N., Fleming, R.M., et al.: A community-driven global reconstruction of human metabolism. Nature biotechnology 31(5), 419–425 (2013)

[11] Shlomi, T., Cabili, M.N., Ruppin, E.: Predicting metabolic biomarkers of human inborn errors of metabolism. Molecular systems biology 5(263), 263 (Jan 2009)

[12] Shlomi, T., Benyamini, T., Gottlieb, E., Sharan, R., Ruppin, E.: Genome-scale metabolic modeling elucidates the role of proliferative adaptation in causing the Warburg effect. PLoS computational biology 7(3), e1002018 (Mar 2011)

[13] Shlomi, T., Cabili, M.N., Herrgå rd, M.J., Palsson, B.O., Ruppin, E.: Network-based prediction of human tissue-specific metabolism. Nature biotechnology 26(9), 1003–10 (Sep 2008)

[14] Becker, S.a., Palsson, B.O.: Context-specific metabolic networks are consistent with experiments. PLoS computational biology 4(5), e1000082 (May 2008)

[15] Colijn, C., Brandes, A., Zucker, J., et al.: Interpreting expression data with metabolic flux models: predicting mycobacterium tuberculosis mycolic acid production. PLoS Comput Biol 5(8), e1000489 (2009)

[16] Jerby, L., Shlomi, T., Ruppin, E.: Computational reconstruction of tissue-specific metabolic models: application to human liver metabolism. Molecular systems biology 6(401), 401 (Sep 2010)

[17] Gille, C., Bölling, C., Hoppe, A., et al.: HepatoNet1: a comprehensive metabolic reconstruction of the human hepatocyte for the analysis of liver physiology. Molecular systems biology 6(411), 411 (Sep 2010)

[18] Lewis, N.E., Schramm, G., Bordbar, A., et al.: Large-scale in silico modeling of metabolic interactions between cell types in the human brain. Nature Biotechnology 28(12), 1279–1285 (Nov 2010)

[19] Bordbar, A., Lewis, N.E., Schellenberger, J., Palsson, B.Ø., Jamshidi, N.: Insight into human alveolar macrophage and M. tuberculosis interactions via metabolic reconstructions. Molecular systems biology 6 (Oct 2010)

[20] Agren, R., Mardinoglu, A., Asplund, A., et al.: Identification of anticancer drugs for hepatocellular carcinoma through personalized genome-scale metabolic modeling. Molecular Systems Biology 10(3), 1–13 (2014)

[21] Wang, Y., Eddy, J.a., Price, N.D.: Reconstruction of genome-scale metabolic models for 126 human tissues using mCADRE. BMC Systems Biology 6(1), 153 (2012)

[22] Vlassis, N., Pacheco, M.P., Sauter, T.: Fast reconstruction of compact context-specific metabolic network models. PLoS computational biology 10(1), e1003424 (2014)

[23] Yizhak, K., Gaude, E., Le Dévédec, S., et al.: Phenotype-based cell-specific metabolic modeling reveals metabolic liabilities of cancer. eLife 3, e03641 (2014)

[24] Rocha, I., Maia, P., Evangelista, P., et al.: Optflux: an open-source software platform for in silico metabolic engineering. BMC systems biology 4(1), 45 (2010)

[25] Stearns, S.C.: The evolution of life histories, vol. 249. Oxford University Press Oxford (1992)

[26] Barabási, A., Oltvai, Z.: Network biology: understanding the cell's functional organization. Nature Reviews Genetics 5(2), 101–113 (2004)

[27] Kitano, H.: Systems biology: a brief overview. Science (New York, N.Y.) 295(5560), 1662–4 (Mar 2002)

[28] Morozova, O., Marra, M.A.: Applications of next-generation sequencing technologies in functional genomics. Genomics 92(5), 255–264 (2008)

[29] Sorek, R., Cossart, P.: Prokaryotic transcriptomics: a new view on regulation, physiology and pathogenicity. Nature Reviews Genetics 11(1), 9–16 (2010)

[30] Martens, L., Hermjakob, H., Jones, P., et al.: Pride: the proteomics identifications database. Proteomics 5(13), 3537–3545 (2005)

[31] Bundy, J.G., Davey, M.P., Viant, M.R.: Environmental metabolomics: a critical review and future perspectives. Metabolomics 5(1), 3–21 (2009)

[32] Winter, G., Krömer, J.O.: Fluxomics–connecting 'omics analysis and phenotypes. Environmental microbiology 15(7), 1901–1916 (2013)

[33] Palsson, B.O.: Systems biology. Cambridge university press (2015)

[34] Acerbi, E., Decraene, J., Gouaillard, A.: Computational reconstruction of biochemical networks. In: Information Fusion (FUSION), 2012 15th International Conference on. pp. 1134–1141. IEEE (2012)

[35] Rocha, I., Förster, J., Nielsen, J.: Design and application of genome-scale reconstructed metabolic models. In: Microbial Gene Essentiality: Protocols and Bioinformatics, pp. 409–431. Springer (2008)

[36] Thiele, I., Palsson, B.Ø.: A protocol for generating a high-quality genome-scale metabolic reconstruction. Nature protocols 5(1), 93–121 (2010)

[37] Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., et al.: Genbank. Nucleic acids research 28(1), 15–18 (2000)

[38] Maglott, D., Ostell, J., Pruitt, K.D., Tatusova, T.: Entrez gene: gene-centered information at ncbi. Nucleic acids research 33(suppl 1), D54–D58 (2005)

[39] Caspi, R., Foerster, H., Fulcher, C.A., et al.: The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. Nucleic acids research 36(suppl 1), D623–D631 (2008)

[40] Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., Tanabe, M.: Kegg as a reference resource for gene and protein annotation. Nucleic acids research p. gkv1070 (2015)

[41] Schomburg, I., Chang, A., Ebeling, C., et al.: Brenda, the enzyme database: updates and major new developments. Nucleic acids research 32(suppl 1), D431–D433 (2004)

[42] Wu, C.H., Apweiler, R., Bairoch, A., et al.: The universal protein resource (uniprot): an expanding universe of protein information. Nucleic acids research 34(suppl 1), D187–D191 (2006)

[43] Bairoch, A., Apweiler, R., Wu, C.H., et al.: The Universal Protein Resource (UniProt). Nucleic acids research 33(Database issue), D154–9 (Jan 2005)

[44] Caspi, R., Foerster, H., Fulcher, C.A., et al.: The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. Nucleic acids research 36(suppl 1), D623–D631 (2008)

[45] Oberhardt, M.A., Palsson, B.Ø., Papin, J.A.: Applications of genome-scale metabolic reconstructions. Molecular systems biology 5(1), 320 (2009)

[46] Soon, W.W., Hariharan, M., Snyder, M.P.: High-throughput sequencing for biology and medicine. Molecular systems biology 9(1), 640 (2013)

[47] Kim, T.Y., Sohn, S.B., Kim, Y.B., Kim, W.J., Lee, S.Y.: Recent advances in reconstruction and applications of genome-scale metabolic models. Current opinion in biotechnology 23(4), 617–623 (2012)

[48] Henry, C.S., DeJongh, M., Best, A.A., et al.: High-throughput generation, optimization and analysis of genome-scale metabolic models. Nature biotechnology 28(9), 977–982 (2010)

[49] Dias, O., Rocha, M., Ferreira, E.C., Rocha, I.: Reconstructing genome-scale metabolic models with merlin. Nucleic acids research p. gkv294 (2015)

[50] Patil, K.R., Akesson, M., Nielsen, J.: Use of genome-scale microbial models for metabolic engineering. Current opinion in biotechnology 15(1), 64–9 (Feb 2004)

[51] Atsumi, S., Cann, A.F., Connor, M.R., et al.: Metabolic engineering of escherichia coli for 1-butanol production. Metabolic engineering 10(6), 305–311 (2008)

[52] Yadav, V.G., De Mey, M., Lim, C.G., Ajikumar, P.K., Stephanopoulos, G.: The future of metabolic engineering and synthetic biology: towards a systematic practice. Metabolic engineering 14(3), 233–241 (2012)

[53] Heinken, A., Sahoo, S., Fleming, R.M., Thiele, I.: Systems-level characterization of a host-microbe metabolic symbiosis in the mammalian gut. Gut microbes 4(1), 28–40 (2013)

[54] Lander, E.S., Linton, L.M., Birren, B., et al.: Initial sequencing and analysis of the human genome. Nature 409(6822), 860–921 (2001)

[55] Stein, L.D.: Human genome: end of the beginning. Nature 431(7011), 915–916 (2004)

[56] Schellenberger, J., Park, J.O., Conrad, T.M., Palsson, B.O.: BiGG : a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions Database (2010)

[57] Ma, H., Sorokin, A., Mazein, A., et al.: The Edinburgh human metabolic network reconstruction and its functional analysis. Molecular systems biology 3, 135 (2007)

[58] Ogata, H., Goto, S., Sato, K., et al.: KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic acids research 27(1), 29–34 (Jan 1999)

[59] Povey, S., Lovering, R., Bruford, E., et al.: The HUGO Gene Nomenclature Committee (HGNC). Human genetics 109(6), 678–80 (Dec 2001)

[60] Selkov, E., Basmanova, S., Gaasterland, T., et al.: The metabolic pathway collection from EMP: the enzymes and metabolic pathways database. Nucleic acids research 24(1), 26–8 (Jan 1996)

[61] Romero, P., Wagg, J., Green, M.L., et al.: Computational prediction of human metabolic pathways from the complete human genome. Genome biology 6(1), R2 (Jan 2005)

[62] Mardinoglu, A., Agren, R., Kampf, C., et al.: Genome-scale metabolic modelling of hepatocytes reveals serine deficiency in patients with non-alcoholic fatty liver disease. Nature communications 5 (2014)

[63] Sahoo, S., Franzson, L., Jonsson, J.J., Thiele, I.: A compendium of inborn errors of metabolism mapped onto the human metabolic network. Molecular bioSystems 8(10), 2545–2558 (2012)

[64] Sahoo, S., Thiele, I.: Predicting the impact of diet and enzymopathies on human small intestinal epithelial cells. Human molecular genetics 22(13), 2705–2722 (2013)

[65] Matthews, L., Gopinath, G., Gillespie, M., et al.: Reactome knowledgebase of human biological pathways and processes. Nucleic acids research 37(Database issue), D619–22 (Jan 2009)

[66] Croft, D., O'Kelly, G., Wu, G., et al.: Reactome: a database of reactions, pathways and biological processes. Nucleic acids research 39(Database issue), D691–7 (Jan 2011)

[67] Machado, D., Costa, R.S., Rocha, M., et al.: Modeling formalisms in systems biology. AMB express 1(1), 1–14 (2011)

[68] Resat, H., Petzold, L., Pettigrew, M.F.: Kinetic modeling of biological systems. In: Computational Systems Biology, pp. 311–335. Springer (2009)

[69] Varma, A., Palsson, B.O.: Metabolic flux balancing: Basic concepts, scientific and practical use. Bio/technology 12 (1994)

[70] Pfau, T., Christian, N., Ebenhöh, O.: Systems approaches to modelling pathways and networks. Briefings in functional genomics p. elr022 (2011)

[71] S, K., J, S.: Stoichiometric and constraint-based modelling. in: System Modeling in Cellular Biology: From Concepts to Nuts and Bolts pp. 73–96 (2006)

[72] Orth, J.D., Thiele, I., Palsson, B.O.: What is flux balance analysis? Nature biotechnology 28(3), 245–8 (Mar 2010)

[73] Segre, D., Vitkup, D., Church, G.M.: Analysis of optimality in natural and perturbed metabolic networks. Proceedings of the National Academy of Sciences 99(23), 15112–15117 (2002)

[74] Shlomi, T., Berkman, O., Ruppin, E.: Regulatory on/off minimization of metabolic flux 102(21) (2005)

[75] Brown, O.: Quantitative monitoring of gene expression patterns with a complementary dna microarray. Science 270, 467–470 (1995)

[76] Wang, Z., Gerstein, M., Snyder, M.: Rna-seq: a revolutionary tool for transcriptomics. Nature Reviews Genetics 10(1), 57–63 (2009)

[77] Barrett, T., Troup, D.B., Wilhite, S.E., et al.: NCBI GEO: archive for functional genomics data sets–10 years on. Nucleic acids research 39(Database issue), D1005–D1010 (Jan 2011)

[78] Parkinson, H., Kapushesky, M., Shojatalab, M., et al.: Arrayexpress-a public database of microarray experiments and gene expression profiles. Nucleic acids research 35(suppl 1), D747–D750 (2007)

[79] Ripley, B.: The r project in statistical computing. MSOR Connections. The newsletter of the LTSN Maths, Stats & OR Network 1(1), 23–25 (2001)

[80] Gentleman, R., Carey, V., Bates, D., et al.: Bioconductor: open software development for computational biology and bioinformatics. Genome biology 5(10), R80 (2004)

[81] McCall, M.N., Uppal, K., Jaffee, H.a., Zilliox, M.J., Irizarry, R.a.: The Gene Expression Barcode: leveraging public data repositories to begin cataloging the human and murine transcriptomes. Nucleic acids research 39(Database issue), D1011–5 (Jan 2011)

[82] Buckingham, S.: The major world of micrornas. Nature (2003)

[83] Wilkins, M., Appel, R., Van Eyk, J., et al.: Guidelines for the next 10 years of proteomics. Proteomics 6(1), 4–8 (2006)

[84] Uhlen, M., Oksvold, P., Fagerberg, L., Lundberg, E., et al.: Towards a knowledge-based Human Protein Atlas. Nat Biotech 28(12), 1248–1250 (Dec 2010)

[85] Prasad, T., Goel, R., Kandasamy, K., et al.: Human protein reference database - 2009 update. Nucleic acids research 37(suppl 1), D767–D772 (2009)

[86] Kaddurah-Daouk, R., Kristal, B., Weinshilboum, R.: Metabolomics: a global biochemical approach to drug response and disease. Annu. Rev. Pharmacol. Toxicol. 48, 653–683 (2008)

[87] Wishart, D.S., Knox, C., Guo, A.C., et al.: HMDB: a knowledgebase for the human metabolome. Nucleic Acids Research 37(suppl 1), D603–D610 (Jan 2009)

[88] The Cancer Genome Atlas (homepage). `http://cancergenome.nih.gov`

[89] TCGA Data Portal (homepage). `https://tcga-data.nci.nih.gov/tcga`

[90] Muoio, D., Newgard, C.: Obesity-related derangements in metabolic regulation. Annu. Rev. Biochem. 75, 367–401 (2006)

[91] Nielsen, J.: Systems biology of lipid metabolism: From yeast to human. FEBS Letters 583(24), 3905 – 3913 (2009)

[92] Shmueli, O., Horn-Saban, S., Chalifa-Caspi, V., et al.: Genenote: whole genome expression profiles in normal human tissues. Comptes rendus biologies 326(10), 1067–1072 (2003)

[93] Irizarry, R., Wu, Z., Jaffee, H.: Comparison of affymetrix genechip expression measures. Bioinformatics 22(7), 789–794 (2006)

[94] Yuneva, M.O., Fan, T.W.M., Allen, T.D., et al.: The Metabolic Profile of Tumors Depends on Both the Responsible Genetic Lesion and Tissue Type. Cell Metab 15(2), 157–170 (Feb 2012)

[95] Pacheco, M.P., Sauter, T.: Fast reconstruction of compact context-specific metabolic networks via integration of microarray data. arXiv preprint arXiv:1407.6534 (2014)

[96] McCall, M.N., Jaffee, H.A., Zelisko, S.J., Sinha, N., et al.: The Gene Expression Barcode 3.0: improved data processing and mining tools. Nucleic Acids Research 42(D1), D938–D943 (Jan 2014)

[97] Saha, R., Chowdhury, A., Maranas, C.D.: Recent advances in the reconstruction of metabolic models and integration of omics data. Current opinion in biotechnology 29, 39–45 (2014)

[98] Glpk - (gnu linear programming kit). `http://www.gnu.org/software/glpk/`

[99] Bornstein, B.J., Keating, S.M., Jouraku, A., Hucka, M.: Libsbml: an api library for sbml. Bioinformatics 24(6), 880–881 (2008)

[100] Hucka, M., Finney, A., Sauro, H.M., et al.: The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4), 524–531 (2003)

[101] Pfeiffer, T., Nu, J., Montero, F., Schuster, S., et al.: Metatool: for studying metabolic networks. Bioinformatics 15(3), 251–257 (1999)

[102] Cvijovic, M., Olivares-Hernández, R., Agren, R., et al.: Biomet toolbox: genome-wide analysis of metabolism. Nucleic acids research 38(suppl 2), W144–W149 (2010)

[103] Mahadevan, R., Schilling, C.: The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metabolic engineering 5(4), 264–276 (2003)

[104] Brochado, A.R., Andrejev, S., Maranas, C.D., Patil, K.R.: Impact of stoichiometry representation on simulation of genotype-phenotype relationships in metabolic networks. PLoS Comput Biol 8(11), e1002758 (2012)

[105] Glez-Peña, D., Reboiro-Jato, M., Maia, P., et al.: Aibench: a rapid application development framework for translational research in biomedicine. Computer methods and programs in biomedicine 98(2), 191–203 (2010)

[106] Noronha, A., Vilaça, P., Rocha, M.: An integrated network visualization framework towards metabolic engineering applications. BMC bioinformatics 15(1), 1 (2014)

[107] Russell, J.B., Cook, G.M.: Energetics of bacterial growth: balance of anabolic and catabolic reactions. Microbiological reviews 59(1), 48–62 (1995)

[108] Stein, L.R., Imai, S.i.: The dynamic regulation of nad metabolism in mitochondria. Trends in Endocrinology & Metabolism 23(9), 420–428 (2012)

[109] Holm, A.K., Blank, L.M., Oldiges, M., et al.: Metabolic and transcriptional response to cofactor perturbations in escherichia coli. Journal of Biological Chemistry 285(23), 17498–17506 (2010)

[110] Feist, A.M., Henry, C.S., Reed, J.L., et al.: A genome-scale metabolic reconstruction for escherichia coli k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. Molecular systems biology 3(1) (2007)

[111] Machado, D., Herrgård, M.: Systematic evaluation of methods for integration of transcriptomic data into constraint-based models of metabolism. PLoS Comput. Biol 10, e1003580 (2014)

[112] Pires Pacheco, M., Pfau, T., Sauter, T.: Benchmarking procedures for high-throughput context specific reconstruction algorithms. Frontiers in Physiology 6, 410 (2015)

[113] Tortora, G.J., Derrickson, B.H.: Principles of anatomy and physiology. Wiley, Hoboken, New Jersey, USA (2012)

[114] Baffy, G., Brunt, E.M., Caldwell, S.H.: Hepatocellular carcinoma in non-alcoholic fatty liver disease: an emerging menace. Journal of hepatology 56(6), 1384–1391 (2012)

[115] Jemal, A., Bray, F., Center, M.M., et al.: Global cancer statistics. CA: A Cancer Journal for Clinicians 61(2), 69–90 (2011)

[116] Kampf, C., Mardinoglu, A., Fagerberg, L., et al.: The human liver-specific proteome defined by transcriptomics and antibody-based profiling. The FASEB Journal 28(7), 2901–2914 (2014)

[117] Ishibashi, H., Nakamura, M., Komori, A., Migita, K., Shimoda, S.: Liver architecture, cell function, and disease. Seminars in Immunopathology 31, 399–409 (2009)

[118] neo4j open source graph database &#187;

[119] Stobbe, M.D., Houten, S.M., Jansen, G.A., van Kampen, A.H., Moerland, P.D.: Critical assessment of human metabolic pathway databases: a stepping stone for future integration. BMC systems biology 5(1), 1 (2011)

[120] Flicek, P., Amode, M.R., Barrell, D., et al.: Ensembl 2014. Nucleic Acids Research 42(D1), D749–D755 (Jan 2014)

[121] Carlson, M.: hgu133plus2.db: Affymetrix Human Genome U133 Plus 2.0 Array annotation data (chip hgu133plus2) (2014)

[122] Huber, W., Carey, V.J., Gentleman, R., et al.: Orchestrating high-throughput genomic analysis with bioconductor. Nature methods 12(2), 115–121 (2015)

[123] Pecorino, L.: Molecular biology of cancer: mechanisms, targets, and therapeutics. Oxford university press (2012)

[124] Weinberg, R.: The biology of cancer. Garland science (2013)

[125] Hanahan, D., Weinberg, R.A.: The hallmarks of cancer. cell 100(1), 57–70 (2000)

[126] Hanahan, D., Weinberg, R.A.: Hallmarks of cancer: the next generation. cell 144(5), 646–674 (2011)

[127] Warburg, O., Wind, F., Negelein, E.: The metabolism of tumors in the body. The Journal of general physiology 8(6), 519–530 (1927)

[128] Vander Heiden, M.G., Cantley, L.C., Thompson, C.B.: Understanding the warburg effect: the metabolic requirements of cell proliferation. science 324(5930), 1029–1033 (2009)

[129] Parsons, D.W., Jones, S., Zhang, X., et al.: An integrated genomic analysis of human glioblastoma multiforme. Science 321(5897), 1807–1812 (2008)

[130] Ohgaki, H., Kleihues, P.: The definition of primary and secondary glioblastoma. Clinical cancer research 19(4), 764–772 (2013)

[131] Louis, D.N., Ohgaki, H., Wiestler, O.D., et al.: The 2007 who classification of tumours of the central nervous system. Acta neuropathologica 114(2), 97–109 (2007)

[132] Smith, J.S., Tachibana, I., Passe, S.M., et al.: Pten mutation, egfr amplification, and outcome in patients with anaplastic astrocytoma and glioblastoma multiforme. Journal of the National Cancer Institute 93(16), 1246–1256 (2001)

[133] Kraus, J.A., Glesmann, N., Beck, M., et al.: Molecular analysis of the pten, tp53 and cdkn2a tumor suppressor genes in long-term survivors of glioblastoma multiforme. Journal of neuro-oncology 48(2), 89–94 (2000)

[134] Pollack, I.F., Hamilton, R.L., Sobol, R.W., et al.: Idh1 mutations are common in malignant gliomas arising in adolescents: a report from the children's oncology group. Child's nervous system 27(1), 87–94 (2011)

[135] Christensen, B.C., Smith, A.A., Zheng, S., et al.: Dna methylation, isocitrate dehydrogenase mutation, and survival in glioma. Journal of the National Cancer Institute 103(2), 143–153 (2011)

[136] Karunakaran, S., Umapathy, N.S., Thangaraju, M., et al.: Interaction of tryptophan derivatives with slc6a14 (atb0,+) reveals the potential of the transporter as a drug target for cancer chemotherapy. Biochemical Journal 414(3), 343–355 (2008)

[137] Ohgaki, H., Kleihues, P.: Genetic pathways to primary and secondary glioblastoma. The American journal of pathology 170(5), 1445–1453 (2007)

[138] Gholami, A.M., Hahne, H., Wu, Z., et al.: Global proteome analysis of the nci-60 cell line panel. Cell reports 4(3), 609–620 (2013)

[139] Folger, O., Jerby, L., Frezza, C., et al.: Predicting selective drug targets in cancer through metabolic networks. Molecular systems biology 7(1), 501 (2011)

[140] Jain, M., Nilsson, R., Sharma, S., et al.: Metabolite profiling identifies a key role for glycine in rapid cancer cell proliferation. Science 336(6084), 1040–1044 (2012)

[141] Belkaid, A., Currie, J.C., Desgagnés, J., Annabi, B.: The chemopreventive properties of chlorogenic acid reveal a potential new role for the microsomal glucose-6-phosphate translocase in brain tumor progression. Cancer Cell International 6(1), 7 (2006)

[142] Belkaid, A., Copland, I.B., Massillon, D., Annabi, B.: Silencing of the human microsomal glucose-6-phosphate translocase induces glioma cell death: Potential new anticancer target for curcumin. FEBS letters 580(15), 3746–3752 (2006)

[143] Zeisel, S.H.: Choline: an essential nutrient for humans. Nutrition 16(7), 669–671 (2000)

[144] Inazu, M.: Choline transporter-like proteins ctls/slc44 family as a novel molecular target for cancer therapy. Biopharmaceutics & drug disposition 35(8), 431–449 (2014)

[145] Awwad, H.M., Geisel, J., Obeid, R.: The role of choline in prostate cancer. Clinical biochemistry 45(18), 1548–1553 (2012)

[146] Glunde, K., Bhujwalla, Z.M., Ronen, S.M.: Choline metabolism in malignant transformation. Nature Reviews Cancer 11(12), 835–848 (2011)

[147] Kumar, M., Arlauckas, S.P., Saksena, S., et al.: Magnetic resonance spectroscopy for detection of choline kinase inhibition in the treatment of brain tumors. Molecular cancer therapeutics 14(4), 899–908 (2015)

[148] Ganapathy, V., Thangaraju, M., Prasad, P.D.: Nutrient transporters in cancer: relevance to warburg hypothesis and beyond. Pharmacology & therapeutics 121(1), 29–40 (2009)

[149] Gupta, N., Miyauchi, S., Martindale, R.G., et al.: Upregulation of the amino acid transporter atb 0,+(slc6a14) in colorectal cancer and metastasis in humans. Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease 1741(1), 215–223 (2005)

[150] Gupta, N., Prasad, P.D., Ghamande, S., et al.: Up-regulation of the amino acid transporter atb 0,+(slc6a14) in carcinoma of the cervix. Gynecologic oncology 100(1), 8–13 (2006)

[151] Karunakaran, S., Umapathy, N.S., Thangaraju, M., et al.: Interaction of tryptophan derivatives with slc6a14 (atb0,+) reveals the potential of the transporter as a drug target for cancer chemotherapy. Biochemical Journal 414(3), 343–355 (2008)

[152] Babu, E., Bhutia, Y.D., Ramachandran, S., et al.: Deletion of the amino acid transporter slc6a14 suppresses tumour growth in spontaneous mouse models of breast cancer. Biochemical Journal 469(1), 17–23 (2015)

[153] Schlame, M., Hostetler, K.Y.: Cardiolipin synthase from mammalian mitochondria. Biochimica et Biophysica Acta (BBA)-Lipids and Lipid Metabolism 1348(1), 207–213 (1997)

[154] D'Souza, K., Kim, Y.J., Balla, T., Epand, R.M.: Distinct properties of the two isoforms of cdp-diacylglycerol synthase. Biochemistry 53(47), 7358–7367 (2014)

[155] Libutti, S.K., He, M.: Molecular-based method of cancer diagnosis and prognosis (Nov 10 2015)

[156] Latour, S., Fischer, A., Martin, E., Arkwright, P.: Methods and pharmaceutical compositions (ctps 1 inhibitors, eg norleucine) for inhibiting t cell proliferation in a subject in need thereof (Apr 17 2014)

[157] Gordon, G.J., Rockwell, G.N., Jensen, R.V., et al.: Identification of novel candidate oncogenes and tumor suppressors in malignant pleural mesothelioma using large-scale transcriptional profiling. The American journal of pathology 166(6), 1827–1840 (2005)

[158] Kim, J.w., Dang, C.V.: Cancer's molecular sweet tooth and the warburg effect. Cancer research 66(18), 8927–8930 (2006)

[159] Schuster, S., Boley, D., Möller, P., Stark, H., Kaleta, C.: Mathematical models for explaining the warburg effect: a review focussed on atp and biomass production. Biochemical Society Transactions 43(6), 1187–1194 (2015)