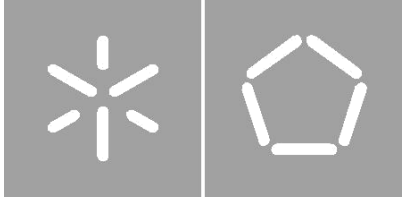
University of Minho
School of Engineering

Susana Raquel da Silva Barbosa

Development of a data integration pipeline for human metabolic models and databases

January, 2016



University of Minho
School of Engineering

Susana Raquel da Silva Barbosa

Development of a data integration pipeline for human metabolic models and databases

Master's Thesis
Bioinformatics Msc

Supervisor

Professor Doutor Miguel Rocha

January, 2016

Declaração

Nome: Susana Raquel da Silva Barbosa

Endereço eletrónico: susanabarbosa.30@gmail.com

Número do cartão de cidadão:14104824

Título da Dissertação: Development of a data integration pipeline for human metabolic models and databases

Orientador: Professor Doutor Miguel Rocha

Ano de conclusão: 2016

Designação do Mestrado: Mestrado em Bioinformática

Universidade do Minho, 31 / 01 / 2016

Assinatura:_Susana Barbosa_____

Acknowledgements

First of all, I would like to thank my parents, that allowed me to study and always had my back despite the adversities. To my brother for the cakes and desserts, that were always a breath of fresh air.

To all my friends that walked with me in this journey, specially my ladies with their motivational talk.

I thank Prof. Miguel Rocha, that allowed me to do this work, for his available time guiding me through it.

I thank Sara that supported me in the decision of being autodidact and learn Java to integrate in this project, teaching me the first steps. To Liu who introduced me to the Neo4j world and was also giving me hints about Java. Both were fundamental in this work and am deeply thankful by all the transmitted knowledge.

Lastly, I would like to let a very special thanking note to my all-time companion that was always by my side. Thankful for all the lost hours with me, the patience, the caring and comprehension. You were there through all the moments, so I would not fall and always gave me strength to proceed with the challenges I proposed myself to.

Abstract

Systems Biology aims to integrate experimental and computational approaches with the purpose of explaining and predicting the organisms' behavior. The development of mathematical models *in silico* gives us a better in-depth knowledge of their biological mechanism. Bioinformatics tools enabled the integration of a large amount of complex biological data into computer models, but also capable to perform computational simulations with these models, that can predict the organisms' phenotypic behavior in different conditions.

Up to date, genome-scale metabolic models (GSMMs) include several metabolic components of an organism. These are related to the metabolic capabilities encoded in the genome. In recent years, multiple GSMMs have been built by several research groups. With the increase in number, of these models, important issues regarding the standardization have arisen, a common problem is the different nomenclatures used by each of the research groups.

In this work, the major focus is to address these problems, specifically for the human GSSMs. Therefore, the two most recent human GSMMs were selected to go through a data integration process.

Integration strategies of these models most important entities (metabolites and reactions), were defined based on an exhaustive analysis of the models. The broad knowledge of their attributes enabled the creation of effective and efficient integration methods, supported by a core database developed in the local research group.

The final result of this work, is a unified repository of the human metabolism. It contains all the metabolites and reactions that were automatically integrated along with some manual curation.

Resumo

A Biologia de Sistemas pretende integrar abordagens experimentais e computacionais com o objetivo de explicar e prever o comportamento dos organismos. O desenvolvimento *in silico* de modelos matemáticos permite atingir um conhecimento mais aprofundado dos seus mecanismos biológicos. Através de ferramentas Bioinformáticas é possível integrar uma grande quantidade de dados complexos nestes modelos computadorizados, assim como, realizar simulações computacionais que permitem prever o comportamento fenotípico dos organismos em diferentes condições ambientais.

Até à data, os Modelos Metabólicos à Escala Genómica (MMEGs) incluem muitos componentes metabólicos de um organismo, relacionando a codificação do seu genoma com as suas capacidades metabólicas. Nos últimos anos, têm sido construídos vários MMEGs, por diferentes grupos de investigação. Com o crescente surgimento destes, tem-se denotado grandes falhas ao nível da padronização, uma vez que são utilizadas diferentes nomenclaturas por cada grupo de investigação.

Neste trabalho, pretende-se colmatar essas falhas especificamente para os MMEGs humanos. Deste modo, foram selecionados os dois MMEGs humanos mais recentes, para passarem por um processo de integração de dados.

As estratégias de integração das entidades mais importantes destes modelos (os metabolitos e as reações) foram definidas com base numa análise exaustiva dos modelos. O conhecimento dos atributos destes permitiu construir métodos eficientes e eficazes, tendo como núcleo uma base de dados desenvolvida no grupo de acolhimento.

O resultado final deste trabalho é um repositório unificado do metabolismo humano. Neste, estão contidos todos os metabolitos e reações que foram integrados automaticamente, com alguma verificação manual.

Contents

List of Figures

List of Tables

List of Acronyms

Ac-FAO	Acylcarnitine and Fatty-Acid Oxidation
ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
BiGG	Biochemical Genetic and Genomic
BKM-react	BRENDA-KEGG-MetaCyc-reactions
BRENDA	BRaunschweig ENzyme DAtabase
ChEBI	Chemical Entities of Biological Interest
CTS	Chemical Translation Service
DB	Database
EC	Enzyme Commission
EHMN	Edinburgh Human Metabolic Network
ETL	Extract, Transform and Load
FBA	Flux Balance Analysis
GSMM	Genome-Scale Metabolic Model
HepatoNet1	Hepatocyte Network 1
HMR 2.0	Human Metabolic Reaction 2.0
HTTP	HyperText Transfer Protocol
InChI	International Chemical Identifier
JVM	Java Virtual Machine
IUBMB	International Union of Biochemistry and Molecular Biology
KEGG	Kyoto Encyclopedia of Genes and Genomes
PGDB	Pathway/Genome Database
REST	REpresentational State Transfer

SB	Systems Biology
SBML	Systems Biology Markup Language
SMILES	Simplified Molecular Input Line Entry System
SQL	Structured Query Language
XML	eXtensible Markup Language

Chapter 1

Introduction

1.1 Context

In the last decades, different research areas, mainly Biology and Computer Sciences, were combined to obtain a better understanding of cellular behaviour. The combination of these fields with the advances in high-throughput techniques allowed the study and a better understanding of how cells work and react to environmental perturbations, the regulation of cellular processes, and even anticipate the behaviour of cells. The metabolism represents all chemical transformations (reactions) that occur within the cell, making this network one of the best characterized in Biology [1].

Systems Biology (SB) resorts to computational tools to study cellular organisms in a system scope by developing models that allow *in silico* phenotype predictions [2]. The computational simulation of mathematical models allows a deeper comprehension of the internal nature and dynamics of cells' processes. Thus, SB is intertwined with Bioinformatics, an information management discipline dedicated to biological data storage, processing, analysis, forecasting and modelling, relying on the use of computer sciences [3].

In the last years, several metabolic models have been proposed to explain and study cell behaviour. These models, under constraint-based modelling approaches, allow the representation of biological knowledge in a mathematical format and the computation of physiological states [4-6], addressing a variety of scientific questions [7, 8]. There are small-scale models, which are composed by a few pathways, and genome-scale models that include all known components of an organism.

Genome-scale metabolic models (GSMMs) [9] integrate physiological information and biochemical metabolic pathways with genome sequences. These models allow the interaction between subsystems, which increases the capability of phenotype prediction in different scenarios. Computational analysis of genome-scale metabolic models, has been used for biochemical [10, 11], biomedical [12-14], bioengineering [15, 16] and Metabolic Engineering [7] purposes. In Metabolic Engineering, the manipulation of metabolic models has the purpose of optimizing the genetic processes to increase the production of an interesting compound. One of the widely used organisms for industrial requirements is *Escherichia coli.* Several *E. coli* strains have been systematically designed to overproduce target metabolites, such as lactate, ethanol, succinate and aminoacids [17] through *in silico* analysis.

Another application of GSMMs is in the biomedical research area, where one of the major tasks is to understand the relationship between metabolism and human diseases. Multifactorial diseases involve alterations in hundreds of genes, developmental factors and environmental conditions [18]. In this context, the knowledge of metabolism is fundamental to understand the phenotypic behaviour of cells. Thus, it is crucial to identify perturbed molecular mechanisms that cause such diseases to discover possible drug targets. The knowledge of metabolism together with individual's genetic background, environmental factors and their predisposition to genetic diseases, may lead to personalised medicine in the near future. For all these reasons, nowadays, one of the biggest challenges is to be able to completely represent *in silico* the human metabolism. To this end, human metabolic networks [19-22] have been constructed during the last years.

1.2 Motivation

One of the steps in the reconstruction of a GSMM is the annotation of its components [23]. This step aims to unambiguously identify those components and provide an efficient mapping between the identifiers and information essential for the reconstruction and analysis processes. However, errors may occur in this step. The incomplete or incorrect annotations present in databases and available models can be propagated to all new reconstructions.

In the development of human GSMMs, other problems have been detected, such as the non-uniqueness of metabolite identifiers across models, unbalanced atomic species arising from an incorrect stoichiometry or formula for one or more reactions, incorrect or missing cofactors, among others [24].

The lack of usage of standards for the annotation of metabolites and reactions makes the comparison of different models extremely difficult. Moreover, it is natural that distinct databases share common records, but the lack of cross-references between them makes the task of achieving a consensus quite hard. In addition, each database focus different types of content which makes the overlap low [18].

In the existing protocols for genome-scale metabolic reconstruction [23], it is recommended to annotate metabolites with a primary identifier from at least one of the metabolic databases (e.g., KEGG [25]), but not all of the existing models follow this recommendation.

One way to minimize models annotation problems, and reach a more complete set of information is to integrate several data sources, including databases and models. Usually, models integrate several databases through the mapping of identifiers. The majority of data integration applications are exclusive to chemical compounds, lacking the capability to integrate chemical reactions. Furthermore, the integration of GSMMs exposes additional complexity, since these systems incorporate additional mechanisms, for instance the need of gathering the information of multiple instances that represent the same metabolite (e.g. in multiple compartments). Integration systems are prone to a variety of errors such as, for example, a metabolite being associated to identifiers of

another metabolite. According to previous comparative evaluation of computational integration tools [26], it is important to establish data quality control and confidence scores for incoming information. In the end, a manual analysis of the data is always necessary.

This work focuses in human genome-scale metabolic models. Hence, a pipeline will be developed that will integrate human models with the existing databases seeking to overcome the limitations of the available platforms. To attain an integrated GSMM of the human species, the laborious task of manual curation is essential, which in turn may provide valuable insights to improve the integration pipeline.

The final result of this work will be a human genome-scale repository. It may be used for the creation of a new human GSMM, composed by curated data, which might be safely used for the most diverse studies and applications, especially within the biomedical field. In addition, this will be an important step towards the standardization and evolution of the human GSMMs, since the scientific community will be able to use this human repository as a standard to improve the models continually.

1.3 Objectives

The main goal of this work is to efficiently integrate existing human genome-scale metabolic models with a set of bioinformatics databases that include information on reactions and metabolites, enriching an existing data integration platform developed in the host group, to build an integrated, unified and global repository of human metabolism.

This will encompass the following scientific/ technological objectives:

- Review the main tools and standards within the fields of human metabolic modelling and data integration in this context;
- Implement a pipeline to integrate data from human genome-scale metabolic models with data from metabolic-oriented databases (e.g. KEGG, MetaCyc [27]);
- Analysis and manual curation of the previous integration result;

- Improve the pipeline according to the insights provided by manual curation;
- Build a final integrated human genome-scale repository.

1.4 Structure of the Thesis

The thesis is organized into six chapters. This chapter includes a short description of the state of the art, as well as the motivation and objectives for this work.

The remaining chapters are organized as follows:

Chapter 2 - Computational representations of the human metabolism

Description of the constitution and main concepts of the metabolic models, with focus on the main Human Genome-Scale Metabolic Models. Likewise, existing metabolic database are characterized. Finally, an analysis is made to existing applications of the metabolic data integration.

Chapter 3 - Graph Databases for Metabolic Data Integration

Explanation of the structure of a graph database, in particular Neo4j and the advantages of its use. Then, the details of a particular database are exposed, which are used in the process of integrating metabolic data.

Chapter 4 - Building a Human Metabolic Integrated Repository

In this chapter are defined which strategies to use in the integration process and, consequently, the construction of a repository.

Chapter 5 - Results

Here, the results of the study, of the selected models for the integration process, are presented, as well as the obtained results with that integration.

Chapter 6 – Conclusions

Final considerations about the obtained results in this work, limitations and future work.

Chapter 2

Computational representations of the human metabolism

The aim of Systems Biology is to understand biological systems, studying their structure, dynamics, control and design methods [28], by integrating computational and theoretical approaches with experimental data. New knowledge in this field is being gathered through tools, such as automated genome annotation, genome-scale metabolic reconstructions and regulatory network reconstruction using microarray data [29].

Bioinformatics is an interdisciplinary field that combines computer science, statistics, mathematics and engineering, to develop methods and software tools for understanding biological data [3]. Bioinformatics and Systems Biology are rapidly growing fields that focus on the complex interactions in biological systems and how these interactions give rise to the function and behaviour of these systems.

The huge technological advances resulted in high performance experimental techniques, which caused a high rate of biological data generated recently. The management of these data and their integration is being realized through Bioinformatics tools. As a result, this provides new insights to understand biological processes that generate new hypotheses to be tested, making it an iterative cycle.

2.1 Metabolic Models

Metabolic models are reconstructed with the aim of understanding the relationship between the genome and the physiology of an organism/cell [9]. These comprise the chemical reactions of metabolism and their metabolic pathways. The main components of a model are the metabolic reactions, the compartments where these reactions occur, the metabolites that participate in those reactions, the enzymes that catalyse these reactions, and the genes that encode those enzymes.

Metabolic models are complex and are based on a highly interconnected network of reactions and metabolites. Computational approaches are required to elucidate and understand metabolic genotype–phenotype relationships. Therefore, their reconstruction has become an indispensable tool for studying the systems biology of metabolism [17, 30-34]. The number of metabolic reconstructions for organisms increased due to the available whole genome sequences.

The genome-scale metabolic model reconstruction process should be careful, since models are used to simulate the behaviour of organisms. The reconstruction process involves several steps such as: genome annotation; identification of the reactions from the annotated genome sequence and available literature; determination of the reaction stoichiometry; definition of subcellular compartments; assignment of localization; determination of the biomass composition; estimation of energy requirements, and definition of model constraints [9, 17, 23, 35].

After assembling the genome scale metabolic network, there is a conversion of the reconstruction to a genome-scale metabolic model. The conversion is a process wherein the reconstruction is converted into a mathematical format, where the systems boundaries are defined, becoming a condition-specific model [23]. Finally, the model is validated and the cycle is repeated until a consensus is reached [36], as shown in Figure 2.1. The final model could be used to perform phenotype simulations through a constraint based modelling approach such as Flux Balance Analysis (FBA).

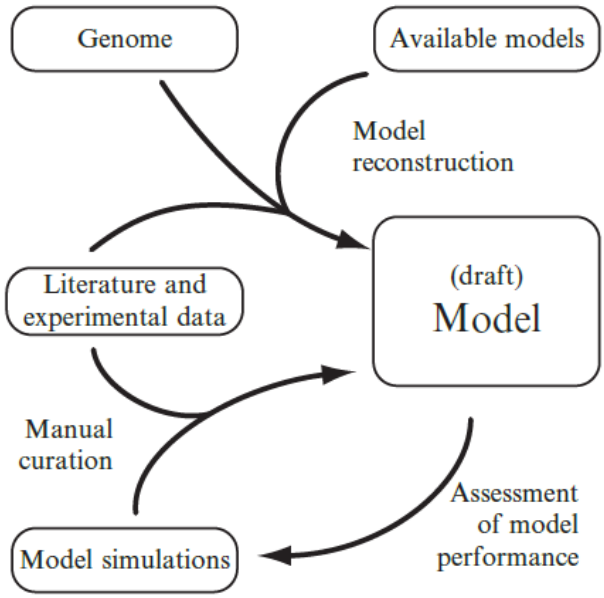


Figure 2.1 - Overview of the process of a genome-scale metabolic model reconstruction and iterative refinement cycle. [36]

GSMMs are commonly represented using the Systems Biology Markup Language (SBML) [37]. This is a software-independent language, XML-based, available in an open and free format. Usually, GSMMs described using SBML have the following elements:

- Compartments: each compartment is a container where reactions take place.
- Species: each species is a chemical substance or entity that takes part in a reaction. This entity mainly represents the association of a metabolite with a compartment, where the reaction occurs, of which the metabolite is part. Thus, the same metabolite can be found in different compartments, therefore existing multiple species for the same metabolite. The genes and other substances involved in reactions can also be stored as a species component.
- Reactions: An entity describing some transformation, transport or binding process that can change one or more species.

GSMMs have been applied in several studies, such as: contextualization of high-throughput data; guidance of metabolic engineering; directing hypothesis-driven discovery; interrogation of multi-species relationships; and network property discovery [7].

Despite the growing amount of GSMMs, the quality of annotation of those does not follow the same trend. One of the main reasons is the lack of an operational standard procedure that is followed by all research groups. This leads to many problems concerning the comparison of models of the same organism. A model needs to be reconstructed multiple times until a final version is reached being that, the more complex the organism, the longer the process will be. Accordingly, every reconstructed model (by different research groups) for the same organism is different and uses different biological data.

The comparison between models is important to obtain a more complete model with all the available information. At the moment, this has been a laborious task, especially for the human organism. Since the amounts of data are huge and there is no unique standard for the annotation of metabolites and reactions in different models, making the identifiers of these entities completely different, the direct comparison is impossible. Thus, it is only possible to do this task in a semi-automatic procedure, using the unique properties of the entities, which are the markers belonging to metabolic databases (i.e. KEGG). Although the databases share common records, most of the times they do not have cross-references between them, making the task of reaching a consensus even harder.

The molecular formula of the metabolites can also help in this comparison, although they cannot be used as single identifiers since several metabolites can have the same formula, but be chemically different due to the structural organization of the atoms.

2.1.1 Human Genome-Scale Metabolic Models

In the past years, several groups have dedicated their efforts to the reconstruction of human metabolic networks. From these studies, several genome-scale metabolic models were developed, viz. Recon 1 [19], Edinburgh [20], Recon 2 [21] and HMR 2.0 [22] (see comparison between these models in Table 2.1) which are used as reference models.

These models have been used to explain metabolic behaviours, such as Warburg effect on cancer [38], to discover new targets for drugs and new biomarkers [39].

The *Homo sapiens* Recon 1 [19] was the first human genome-scale metabolic model fully compartmentalized. This reconstruction was based on genomic and bibliomic data. The manual literature-based reconstruction ensured that the network components and their interactions were based on direct physical evidence and reflected the knowledge of human metabolism. The validation of the basic functionality of this model was made through the *in silico* simulation of 288 known metabolic functions. The final model accounts for the functions of 1496 unique genes, 2004 proteins, 2766 metabolites, and 3311 metabolic and transport reactions.

The Edinburgh human metabolic network (EHMN) [20] was manually reconstructed by integrating genome annotation data from several databases and metabolic reaction information from literature. This model contains 2322 genes, 2671 metabolites, 2823 metabolic reactions and more than 800 Enzyme Commission (EC) numbers. However, since 1189 transport reactions and 457 exchange reactions were not considered, due to the fact that subcellular location information was still not included, researchers felt that there was a need to perform a compartmentalization of this model [40].

A compartmentalization enables a better understanding of the complexity of the human metabolism, because the micro-environments in different organelles may lead to distinct functions of the same protein and the use of different enzymes for the same metabolic reaction. Therefore, the previous model was extended by integrating the sub-cellular location information for the reactions, adding transport reactions and refining the protein-reaction relationships based on the location information. The validation of this "new" model was made by analysing pathways to examine the capability of the network to synthesize and/or degrade some key metabolites.

The Recon 2 [21] is a consensus "metabolic re-construction", being the most comprehensive representation of human metabolism that is applicable to computational modelling. This model combines information from EHMN, HepatoNet1 (comprehensive reconstruction of human hepatocytes) [41], Ac-FAO [13] module (with acylcarnitine (AC) and fatty acid oxidation (FAO) metabolism) and the human small

intestinal enterocyte reconstruction with the content of Recon1. More than 370 transport and exchange reactions were added based on a review of literature. After this, unambiguous third-party identifiers for cellular compartments, metabolites, enzymes and reactions were applied.

The Human Metabolic Reaction 2.0 (HMR 2.0) [22] database is the largest biochemical reaction database for human metabolism in terms of number of reactions/genes/metabolites (including all of the genes, metabolites and reactions in the recently published models), as well as in terms of covering most parts of metabolism. The HMR 2.0 database was constructed using previously published genome-scale models and pathway databases, including KEGG, HumanCyc [42], Reactome [43] and LIPIDMAPS [44] Lipidomics Gateway. This was the result of the expansion of the previous HMR database [45] by including the lipid metabolism. Thus, it contains 59 fatty acids, which enable mapping and integration of lipidomics data. The resulting HMR2.0 database contains 3765 genes, 6007 metabolites (3160 unique metabolites) and 8181 reactions, and 74 percent of the reactions associated to one or more genes. Integration of extensive lipid metabolism may allow not only for the understanding of the contribution of lipids to the development of diseases, but also for the study of the relationship between lipid metabolism and cellular molecular mechanisms [46].

Table 2.1 - Comparison of features of Recon1, Edinburgh (with compartmentalization), Recon 2 and HMR 2.0.

Number of	Recon 1	Edinburgh	Recon 2	HMR 2.0
Compartments	8	9	8	9
Metabolites (Species)	2766	3347	5063	6007
Unique Metabolites	1509	-	2626	3160
Genes	1501	2322	2158	3765
Reactions	3744	6216	7440	8181

2.2 Metabolic Databases

Metabolic databases are a valuable tool for the reconstruction and interpretation of metabolic models. These databases have detailed information about chemical entities and reactions, which are the primary components of these models. In the reconstruction of GSMMs, chemical entities can be annotated by several databases, being the most popular KEGG [25] and the ChEBI [47].

KEGG (Kyoto Encyclopedia of Genes and Genomes) is a knowledge base for systematic analysis of gene functions, linking genomic information with higher order functional information [25]. It can be seen as a computer representation of the biological system, where biological objects and their relationships at the molecular, cellular and organism levels are represented as separate database entries [48]. This integrates genomic, chemical and systemic functional information. Currently, it consists of 4 categories (Systems, Genomic, Chemical and Health information) and 17 main databases. The Chemical category is the most important for the comprehension of the metabolic models. Here, there is information related with the metabolites through the Compound database (which is a gathering of metabolites and other small molecules) and the Glycan database (that contains carbohydrates). Biochemical reactions are represented in the Reaction database, where the metabolites are involved as products and/or reagents. The reactions are connected to the enzyme databases, allowing the integration of genomic analysis. Thereby, through a simple search of a compound, the basic biochemical information, reactions and metabolic pathways in which it participates and even biomedical information, can be accessed.

ChEBI (Chemical Entities of Biological Interest) is a freely available dictionary of molecular entities focused on “small” chemical compounds. It systematically combines information from various databases, which is manually annotated and curated. So, it focuses on high quality manual annotation, non-redundancy, and provision of a chemical ontology rather than full coverage of the vast chemical space. In addition to molecular entities, ChEBI contains groups (parts of molecular entities) and classes of entities. The ChEBI database also includes a chemical ontology, which allows the

relationships between molecular entities or classes of entities and their parents and/or children to be specified in a structured way [47]. Thus, ChEBI provides detailed chemical data about compounds, which is important for the study of chemical interactions that occur within a cell.

Both databases attempt to supply cross-references between themselves and also for other databases. Despite the importance of these databases, for the study and integration of the existing GSMMs, there are also another databases that could be used to extend and improve the annotation of models entities.

BRENDA (BRaunschweig ENzyme DAtabase) is a collection of enzyme functional and molecular data [49]. It represents a comprehensive database containing all enzymes classified according to the EC classification system of the IUBMB [50] (International Union of Biochemistry and Molecular Biology). BRENDA contains information on all classified enzymes of organisms of all taxonomic groups.

The information is extracted from primary literature and manually curated. Each entry is linked to a literature reference and to the source organism. Additionally, BRENDA can provide large amounts of information as the existence of enzymes in organisms, their cellular localization, their involvement in human diseases, their active centers and interfaces and additional kinetic data. In summary, BRENDA stands out for not being limited to a specific aspect of the enzyme or to a specific organism.

BioCyc collection of Pathway/Genome Databases (PGDBs) provides electronic reference sources on the pathways and genomes of many organisms [51]. PGDBs provide genomic information with an extensive dimension that allows researchers to analyse the relationship between organism's genome and metabolic network. BioCyc is primarily microbial, but contains databases for humans and for important model organisms. Consequently, it allows comparative analysis, since multiple PGDBs are available within one place. BioCyc organizes its databases in tiers, according to the amount of manual curation and update they received.

Within BioCyc, the most relevant databases for the reconstruction and integration of GSMMs are HumanCyc and MetaCyc that are in tier one, which means that these have been created through intensive manual efforts and get continuously updated.

MetaCyc is a non-redundant reference database, of small-molecule metabolism that contains experimentally verified metabolic pathways and enzyme information [52]. MetaCyc contains pathways involved in the primary and secondary metabolisms, as well as associated metabolites, reactions, enzymes and genes. The MetaCyc goal is to catalog the metabolism universe, storing a representative sample of each of the pathways experimentally elucidated. Once that MetaCyc is based in experimentally elucidated and verified data only, it becomes a valuable asset for the study of the models metabolic pathways.

HumanCyc is a database of enzymatic reactions and human metabolic pathways [42]. The aforementioned is a PGDB for *Homo sapiens*, from the annotated human genome. It provides an expandable human metabolic map diagram, and it has been used to generate a steady-state quantitative model of human metabolism. HumanCyc positions human genes in a pathway context thereby facilitating analysis of gene expression, proteomics, and metabolomics datasets. Although HumanCyc is considered a curated database, it is not yet completely verified. Thus, despite having a helpful use, specifically in the human models, it must be cautiously used.

As reported, there are several databases with different purposes. The diversity and amount of information in databases allows to structure the knowledge to become more usable. However, since there are many available data in different databases, some problems regarding redundancy and inconsistency between them, can occur. The majority of the databases comprise cross-references between their data, but nonetheless it is possible to find incoherent information in considerable amounts.

2.3 Metabolic Data Integration: studies and applications

In Stobbe et al. [18], the comparison of five databases (EHMN, Recon 1, HumanCyc, Reactome and the metabolic subsets of KEGG) and their analysis showed that only a small core of metabolic network coexists in all five databases. In particular, in the case of reactions, the overlap is quite low, since only 199 reactions can be found in all five databases. This occurrence is partially due to conceptual differences in the databases, like, for instance, in EHMN, where 23 percent of unique reactions are transport reactions and reactions in lipid metabolism. Knowing that one of the main reasons for the lack of overlap is the difference of content, not only the syntax, each one of the five pathway databases provides a valuable piece of the puzzle. This inability to directly make use of metabolite/reaction information from biological databases or other models, due to incompatibilities of representation, duplications and errors, is an obstacle for the reconstruction of new high quality metabolic models.

One of the purposes of information integration is the reconciliation of heterogeneous data sources to obtain a non-redundant, non-ambiguous and complete information system. The integration of heterogeneous data sources significantly enhances the quality of a reconstruction that eventually improves the quality standards. Thus, the reconciliation of metabolites and reactions is an essential step in the development of comprehensive metabolic models.

In order to fill the existing gaps in the databases and specially the lack of standards for the identifiers, applications that integrate multiple available resources have been created. This can ease, for instance, the process of automatically updating the annotations of GSMMs. Here, three open-source applications will be described that do the mapping between available metabolite identifiers.

MetMask [53] (the metabolite masking tool) is an application that integrates reference chemical libraries and external public databases, building from these a local database that contains groups of identifiers. The primary identifiers and their connections with other public databases are extracted from the external databases. The collected data is

used to merge groups that are considered compatible by the heuristics implemented in the application. These heuristics include the use of rules that restrict the inclusion of identifiers in the group, depending for instance, if this is coming from the same source. This application allows to import other data, which can increase the accuracy of the merge. The result of the search by identifier can be seen in a graph format.

The Chemical Translation Service [54] (CTS) is a web-based tool to map identical chemical identifiers. It aggregates multiple external sources in a single database where the main identifier is the International Chemical Identifier (InChI) code Hash Key. The InChI is a textual identifier to define a chemical substances and it can encode molecular information, including structural data. Depending on the chemical molecule, the InChI can reach thousands of characters; this eventually is a problem for database searching. Additionally, an InChI contains non-alphanumeric characters, which may also be another problem for certain searching engines. To overcome this problem, the InChIKey is a twenty-seven-character code generated from the InChI hash algorithm that contains only regular characters; this identifier eases the search of information within databases.

This tool finds all standard InChIKeys that are linked to an input identifier and returns all identifiers of the requested output type(s) that are linked to the same standard InChIKeys. The CTS consists in three major services: the Discovery Service (detects chemicals); the Convert Service (interconverts any chemical identifier into other chemical identifiers); the Batch Convert Service (converts multiple identifiers of the same type into multiple identifiers). This application is very useful, for instance, to detect doublets.

UniChem [55] is a web application available for free, for compound identifier mapping. It was designed to optimize the efficiency with which structure-based hyperlinks may be built and maintained between chemistry-based resources. It has some characteristics in common with the previous application (CTS) since it is also based on standard InChIs and InChIKeys to make the match of the identifiers. Furthermore, multiple external databases are also aggregated in a single database, where the queries are sent. In this case, before being integrated, the external databases go through quality tests, where, for instance, it is verified if an InChI pattern of an entry can be converted into an InChIKey

pattern. Moreover, UniChem preserves which database identifiers are associated with a certain InChI identifier [26].

The main purpose of these applications is to provide a tool to support the mapping of metabolite identifiers between distinct sources, which is a valuable utility in metabolic network reconstruction projects. Most of these tools cover a large set of the identifiers, although there is a poor match among databases.

The InChI based applications were considered the best approach for the mapping [26]. In particular, CTS showed a better capability to map identifiers. Despite this, the task to annotate these identifiers still remains laborious, since the automatic updating of the metabolite identifiers in metabolic network reconstructions is not yet possible. Although these applications help supplement information relating to metabolites, it is not sufficient for a complete and correct unification of all of the information present in a GSMM. A more sophisticated pipeline for the integration of metabolites and biochemical reactions still remains a challenge.

Parallel to these tools there are also a few databases that integrate several metabolic sources. The BKM-react (BRENDA-KEGG-MetaCyc-reactions) [56] is a comprehensive non-redundant biochemical reaction database containing both enzyme-catalysed and spontaneous reactions. Biochemical reactions collected from BRENDA, KEGG and MetaCyc were matched and integrated by aligning substrates and products. The reaction comparisons follow two steps: the comparison of reactant structures using InChIs and the compound name comparison (including synonyms). The result of a query is the display of all aligned reactions for all databases in comparison. BKM-react can significantly facilitate and accelerate the construction of accurate biochemical models.

MetaNetX is a user-friendly and self-explanatory website for accessing, analysing and manipulating GSMMs as well as biochemical pathways [57]. It consistently integrates data from various public resources (models, metabolic pathways, etc.) in a single repository. All repository or user uploaded models are automatically reconciled in a common namespace defined by MNXref [58]. MNXref uses information on cellular compartments, reactions, and metabolites that is sourced from a number of external

resources. Consequently, for each entity mentioned, it indicates which external resource provided the information used in MNXref.

MetaNetX, which allows an exhaustive analysis of any model, is a user-friendly tool that can be useful when establishing a comparison between models. In addition, two or more GSMMs or pathways can be compared to determine shared parts. It is also possible to create new models based on the analysis and modification of existing models. For instance, MetaNetX can be used to increase/update a model. Information concerning the models, as well as results of analyses performed on them, is provided in the form of tables.

MetRxn [59] is a knowledge base that includes standardized metabolite and reaction descriptions by integrating information from databases and genome-scale metabolic models into a single unified data set. This approach is identical to the aforementioned MetaNetX, however the methods used are different, which leads to different results. For instance, while MNXref provides a single reconciliation based on heuristic merging of stereoisomers, MetRxn provides two reconciliations in which stereoisomers are considered separately or as a single entity [58]. This affects, for instance, the final number of unique reactions. Furthermore, the MetRxn is much more limited in terms of analysis and comparison of the models.

Among these, the MetaNetX reveals itself, at the moment, as the best tool for the study and comparison of models.

Chapter 3

Graph Databases for Metabolic Data Integration

This work emerged from the urgent need to create tools with the capability to compare metabolic models, with the focus on human GSMMs. The integration of these models implies a detailed examination of the components found in these models (i.e., species, reactions, etc.). Therefore, it is necessary to integrate existing databases of biochemical information with the human GSMMs. This will provide a more detailed information for the entities, to clarify cases of ambiguity and to standardize the annotation. In order to assist this integration, an approach based in graph databases is used. This chapter briefly describes graph databases and their use for metabolic data integration, in a tool previously developed in the host research group.

3.1 Graph Databases

Graph databases are based on graph structures. A graph is defined as a set of vertices (or nodes) and a set of edges (or relationships) connecting the vertices. The edges can be directed or undirected depending of the type of graphs. To summarise, a graph system allows to show how certain entities are related.

Connected entities (or nodes) are contained within a graph database, each of which is constituted by a set of outgoing edges and / or incoming edges and a set of properties (key-value-pairs), where one of them is a unique identifier. These nodes can be labelled with one or more labels that identify its role in context. In turn, a relationship (or edge) has two nodes (a start node and an end node), a direction, a type, and can also have a set of properties.

The use of a graph database may be advantageous versus a relational database, since it is not necessary to make an extensive entity-relationship study of the domain to define a database schema. Because, in graph databases, anything can connect to anything, future changes can be easily applied. This flexibility also means fewer migrations. Furthermore, in certain types of operations (e.g., extensive and recursive join operations) they are much faster [60], allowing to scale with large data sets more naturally, given that only part of the graph is traversed to obtain the result of a query. Thus, when querying highly related data, a graph database can be many folds faster than a relational database.

Graph databases engines also operate with Create, Read, Update, and Delete operations (CRUD). Some databases are able to operate under non-native engine (i.e., third party storage engines). However, with the use of the non-native graph storage, the graph database becomes a query mediator that translates graph query languages to the non-native storage engine. In these cases, the query engines have to make more computational effort. This happens because non-native graph storage typically depends on a mature non-graph backend (such as MySQL), while native graph storage is optimized and designed for storing and managing graphs.

3.2 Neo4j

Neo4j [61] is an open-source NoSQL graph database implemented in Java and Scala. It is a high performance graph store with all the features expected from a mature and robust database, such as, with a proper query language and ACID (Atomicity, Consistency, Isolation, and Durability) transactions. Through atomicity, it is guaranteed that, if a

transaction fails, the database remains unchanged. Changes made in the database have to respect its integrity, thus maintaining consistency. The isolation of the parallel transactions prevents transactions from interfering with each other. The results of a committed transaction stay available permanently, thus making graph databases durable and is quite reliable.

Neo4j provides native graph storage that enables its engine to perform native graph processing. Therefore, defining the relationships at runtime time have minimal impact in the performance of the queries.

Cypher is a declarative graph query language, specific to Neo4j. It is focused in the clarity of their queries, being its language based on iconography and English prose. Although it is a simple language, it is nonetheless powerful, allowing both simple and complex queries to be easily expressed. Knowing that "( )" represent the nodes and "[ ]" the relationships, it is easy, at a first sight, to write and interpret a relatively easy query. This proximity with graphical representation makes it extremely intuitive.

Neo4j is implemented on top of the JVM (Java Virtual Machine), and can be easily accessed by either a RESTful[1] HTTP[2] API[3] or Java API. It has support for several other programming languages (e.g., python, ruby, etc.) and platforms (e.g., node.js), not being required for the user to have an extensive knowledge of the server and the resources it hosts in advance.

[1] REpresentational State Transfer
[2] Hypertext Transfer Protocol
[3] Application Programming Interface

Furthermore, it is also possible to query and visualize the data (example in Figure 3.1), through a Web application (the Neo4j Browser). Here, the graph can be visually explored, providing the possibility to see properties of the entities by a simple click on a node or a relationship. It is also possible to expand connected nodes with the interface. This graph visualization tool provided by the database allows a more intuitive perception and analysis of the data. All these features enable the development and analysis of big network structures.

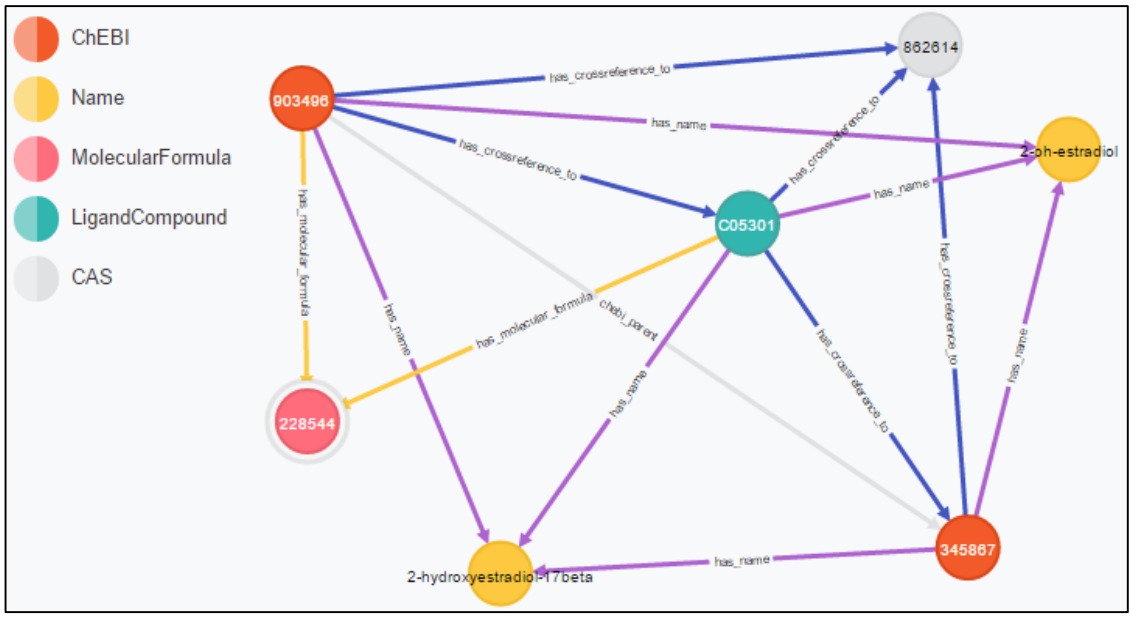


Figure 3.1 - Example of a query to unified database using the Neo4j Web application.

In the Neo4j graph database, as expected, each graph is constituted by many nodes that can be related unidirectionally. It is mandatory that each node has a label that says what it represents. Moreover, Neo4j allows the addition of more than one label to a node, since this can fulfil several different roles. Using labels it is possible to bundle nodes, which facilitates the queries. The relationships can also have labels that tell which way a certain node can relate with another. Both nodes and relationships may be established by properties that are, just as it sounds, the properties of the node, meaning that they keep the relevant information about every node.

3.3 A graph database for metabolic data integration

As a basis for this work, an existing metabolic data integration system was used to aid the integration tasks (http://darwin.di.uminho.pt/biodb/). This system loaded information from several databases of compounds and reactions in a single unified Neo4j graph database (Table 3.1). This database is the central storage to catalogue biochemical entities for analysis and integration purposes. Parallel to this database, it also provides an integrated database that integrates the instances from the central storage based on several rules and axioms.

Table 3.1 - Number of metabolites (first part) and reactions (second part) in the unified database.

Database	Records	Version
KEGG (Compound)	17909	Release 68.0 (October 1, 2013)
KEGG (Glycan)	10988	Release 68.0 (October 1, 2013)
KEGG (Drug)	10126	Release 68.0 (October 1, 2013)
MetaCyc	15983	Release 17.5
BiGG	2835	Schellenberger et al. (2010)
SEED	16996	Aziz et al. (2013)
ChEBI	81231	Release 110 (December 2, 2013)
KEGG	9886	Release 68.0 (October 1, 2013)
MetaCyc	12264	Release 17.5
BiGG	7135	Schellenberger et al. (2010)
SEED	13246	Aziz et al. (2013)

The system has two levels: a first level loads information from each data source in a unified resource. The second level, the integrated database fuses instances from the unified database by creating clusters of metabolic instances. These instances are potential compounds or reactions that are considered as equivalent (i.e., the same molecule or the same reaction).

Since it is a Neo4j database, it is possible to query and visualize the data from the Web platform. This feature is very useful because it allows an intuitive analysis of the references of the properties connected among species. When a species is missing a reference, it is necessary to find at least one reference to identify it. In this graph database, each instance of a compound of a biochemical database (e.g., KEGG, MetaCyc, etc.) is a distinct node. These nodes are connected by their related properties, as well as other identifiers from other databases. Therefore, it is possible to traverse the graph through the properties (of a specie) to identify the references that are related to each other.

3.3.1 Database Structure

The unified schema is similar to an ontology, with entities, classes and relationships. Ontologies allow for organizing and giving structure to information, enabling computers to reason about the data. The objects in an ontology are characterized by a class hierarchy and are related to each other by relationships. This relationship usually follows the subject-predicate-object triple (e.g., CoA `has_formula` C21H36N7O16P3S).

The Metabolite and the Reaction are the top classes of the database, since they are the core entities of this domain (Figure 3.3). Then, they are extended by several sub-classes, that are specific to other biochemical database instances (e.g., <*Metabolite*, *ChEBI*>, <*Metabolite*, *LigandCompoud*>, etc.). Both Reaction and Compound are considered an entity class.

The primary attributes (Table 3.2) are represented as property classes (e.g., Molecular Formula, Name, InChI, etc.). However, due to the diversity of properties between instances, lesser popular attributes are considered secondary properties, and these are stored as attributes of the entities.

Table 3.2 - Primary properties of the metabolites. KEGG instances with Mol structures have computed (*) InChI and SMILES.

	Synonyms	Formula	Charge	Mol	InChI	SMILES
KEGG (Compound)	Yes	Yes	No	Yes	No *	No *
KEGG (Glycan)	Yes	Yes	No	No	No	No
KEGG (Drug)	Yes	Yes	No	Yes	No *	No *
MetaCyc	Yes	Yes	Yes	No	Yes	Yes
BiGG	No	Yes	Yes	No	No	No
SEED	Yes	Yes	Yes	No	No	No
ChEBI	Yes	Yes	Yes	Yes	Yes	Yes

Finally, a set of relationships is declared to specify how classes interact between each other (Figure 3.2). Among the variety of possible relationships, the most relevant is the property attribution with the relationship `has_<property name>` that connects an entity object with a property object. Related entities may also be connected with each other with the `has_crossreference` relationship. This relationship usually indicates that they are strong candidates for entity fusion. Apart from the identity relationship, the hierarchy between compounds is established with `instance_of/super` relationship (e.g., META:ETOH `instanse_of` META:Primary-Alcohols). Finally, reaction entities are related to metabolites with the `left/right` relationship to describe stoichiometry.

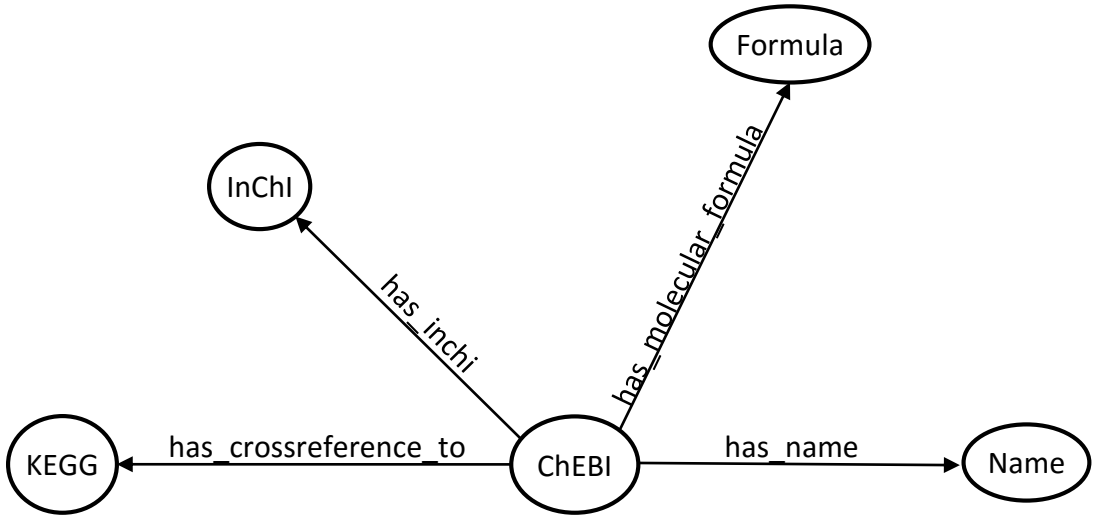


Figure 3.2 - Representative scheme of a part of the graph visualization, in the Neo4j platform.

In the Neo4j graph environment, the classes are nodes, while relationships are edges between nodes, this implies that nodes are both properties (i.e., names, formulas, etc.) and entities (i.e., a specific KEGG compound, a reaction).

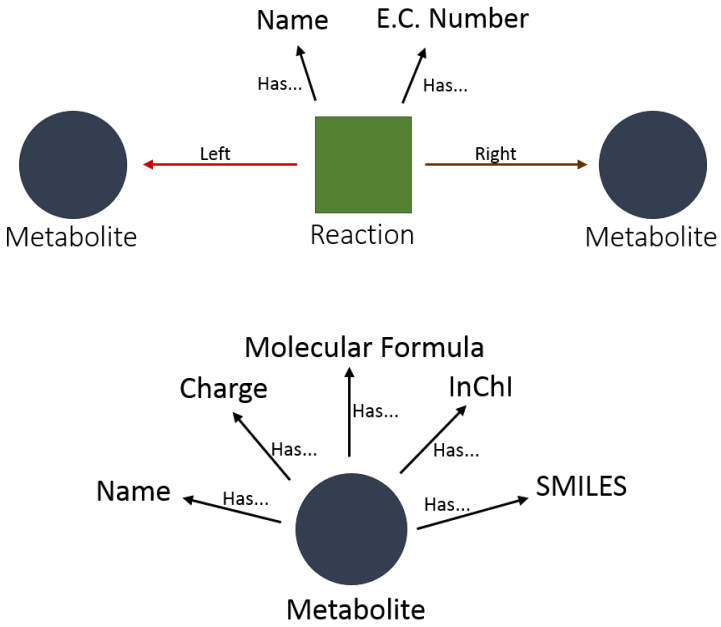


Figure 3.3 - Organization of unified graph database. a) Reaction schema. Each reaction is connected to its reactants and products (left or right). It has Name and E.C. Number. b) Each metabolite is composed of Name, Charge, Molecular Formula, InChI, SMILES (Simplified Molecular Input Line Entry System).

3.3.2 Populating the Database: ETL

The unified graph database is populated by a pipeline similar to other Extract, Transform and Load (ETL) systems. An ETL tool is based on three core tasks:

- Extract, where in a set of heterogeneous data sources (i.e. KEGG, ChEBI, BiGG [62], etc.) are loaded. The main role of this task is to read the data from the source and convert into a standard format. This is highly dependent on the type of data source.
- Transform, where the transformation of those data, for the structure that unifies them (unification schema), occurs. The transformation task converts to the graph data format. This involves identification of the classes and relationships required by the instance to be loaded into the unified database.
- Load, where the data is loaded into the unified database.

Besides these three stages, to secure a robust ETL pipeline, it is necessary to complement it with several additional subsystems. These subsystems are related to a variety of tasks that are relevant to an integration system, such as logging, backup, quality check and many more. Within these roles, a few were chosen to be implemented based on their relevance to solve the inherent problem. A quality screening handler is based on systematically applied tests to checking of issues regarding to the quality of data. These can be invalid descriptors of attributes such as invalid chemical formulas, unidentified references. Within this data context, molecular formulas must be subject to a normalization process (since equal molecular formulas can be written in different ways, e.g. CHO, C1H1O1, and OCH).

The addition of a new data source involves an analysis of the data type, made by the data profiling system. This analysis assists the data conformer in the purpose of "translating" data to a unified version.

In addition to these subsystems, an ETL system requires a data storage system in which it serves as an intermediate store to support the process. Depending of the heterogeneity of the sources, the design process of this component is usually time

consuming and prone to future changes. Graph database entities are represented by generic vertices annotated by properties to provide some identity to these. Then, relationships are created by connecting these vertices with edges. Therefore, it is possible to represent a flexible schema since everything can relate to anything.

3.3.3 Integration Algorithms

The metabolites of the global database were integrated based on their properties and attributes. The selection of these was based on the necessity and the scope. Take, for instance, the InChI property which is commonly referred in the literature as the best descriptor for biochemical molecules, due to its unique and unambiguous representation. However, integration with InChI will only apply to a few molecules that have an InChI descriptor. Thus, other properties less viable are needed to increase the scope.

The InChI Key is used to discard the protonation state of the molecules. The InChI Key is a 25 letter hash key (XXXXXXXXXXXXXX-YYYYYYYYFV-P), having the first block of 14 letters to encode the structure of the molecule, the second block to encode the stereo structure and last three characters to encode meta information, the standard InChI flag, the version, and lastly the protonation. By discarding the last character that encodes protonation, it is possible to capture molecules independent of their protonation state. A limitation to this strategy is that unlike InChI the InChI Key is not unique, however this happens only in rare occasions were hash collisions occur.

Important properties are the ChEBI parent relationships. Some ChEBI entities are related to each other by a chebi_parent relationship. This, allows to fuse internal replicas among ChEBI instances. The KEGG remark attribute is also used. In many cases the remark attribute of KEGG instances has a “Same as:” notification (e.g., Same as: C06217). The integration exploits this attribute to recover a few entities that are replicas between the KEGG Compound, Glycan and Drug databases.

The cross-references were also used since many compounds lack proper descriptors to determine the identity of the metabolites. Also, the cross-referencing is in fact the largest contributor to resolve metabolites.

Each of the methods generates an independent set of metabolite clusters, and later after evaluation, they are merged together to create the master integration set. To summarize, five methods were combined together: the InChI, the InChIKey, ChEBI parent references, KEGG remark attribute, and cross-references. A total of 22.001 metabolite clusters were created using those strategies.

While the reconciliation of metabolites relies essentially on the attributes (e.g., chemical structures, names, molecular formula), the reaction instances offers a very limited set attributes.

Some databases have a name for certain reactions, usually a popular name, but perhaps the most relevant attribute related to reactions is the Enzyme Commission Number (EC). A problem of this classification method is the ambiguity of a few numbers, as example the EC 1.1.1.1 which represents the role of an alcohol dehydrogenase is valid for every reaction that acts on a primary alcohol to produce an aldehyde, a simple query with this EC number in KEGG would result in several reactions. Therefore, a single EC number may span to multiple stoichiometry.

The integration of reactions relies on a different method. The comparison of the stoichiometry between reactions allows precisely to identify replicas among reaction instances.

As an example, the following reactions from KEGG and MetaCyc:

$$r_1 : C00631 \leftrightarrow C00074 + C00001 \quad (3.1)$$

$$r_2 : 2 - PG \leftrightarrow PHOSPHO - ENOL - PYRUVATE + WATER \quad (3.2)$$

The identity of the species plays an important role in the identity of the reactions. If species in the first reaction are equal to the ones in the second, then the two reactions can be safely assumed as identical.

However, there are some aspects about the stoichiometry of the reaction that must be taken into account. The protonation state of the species in the stoichiometry changes the number of hydrogen atoms in their molecular formula. This implies to balance the stoichiometry with the addition of hydrogen molecules. But since on the metabolite integration method it was assumed that the protonation state is not relevant for the metabolite identity, in the case of reactions this should also be true.

$$r_3 : C00002 + C00022 \leftrightarrow C00008 + C00074 \quad (3.3)$$

$$r_4 : ATP + PYRUVATE \leftrightarrow ADP + PHOSPHO - ENOL - PYRUVATE + PROTON \quad (3.4)$$

This can be done by ignoring the hydrogen molecule in the stoichiometry, which is a common practice in many integration platforms. The direction of the reaction also plays a role in the stoichiometry of the reaction. As aforementioned in the domain relationships, a reaction has left_component and right_component these are the left and right side of the stoichiometric equation. However, the left and right it is not related to any orientation. Given for example, the MetaCyc has examples of reactions with direction marked as RIGHT-TO-LEFT. It would be convenient to normalize the direction by inverting the stoichiometry of these cases. However, since the direction of reactions is not easy to define as they are dependent on several variables (e.g., enzyme regulation, concentration of species, etc.), most databases do not specify any direction for their reactions, leaving the judgement to the user.

The reaction integration algorithm (Algorithm 1) takes the stoichiometry of the reactions and performs mapping substitutions to the integrated species. As an example, in reactions (3.1) and (3.2) assuming every species is correctly integrated, substitution of the original species would result in identical stoichiometry, thus the returning stoichiometry dictionary would be $[\,[c_a \leftrightarrow c_b + c_c] \rightarrow \langle r_1, r_2 \rangle]$ (previous species were replaced by c_a, c_b, c_c).

The P set controls species to ignore, since the species found in P are ignored from the stoichiometry. This allows to remove the proton species from reactions (Algorithm 1, line 11).

Algorithm 1: Reaction Stoichiometry Unification

Input: U a unification map, R a set of reactions, P a set of metabolites to ignore
Output: S a dictionary with reactions grouped by stoichiometry

1: **procedure** ReactionIntegration(U, R, P)
2: $S[s] = \{\}$ ⇒ Initialize empty dictionary
3: **for** $r \in R$ **do**
4: $s = \{\}$ ⇒ Initialize empty stoichiometry
5: **for** $\langle m, i \rangle \in r$ **do**
6: **if** $U[m]$ **then**
7: $c \leftarrow U[m]$
8: **else**
9: $c \leftarrow m$
10: **end if**
11: **if** $m \notin P$ or $c \notin P$ **then** ⇒ Test if to ignore species
12: $s[c] \leftarrow \begin{cases} 1 & \text{if } i \geq 0 \\ -1 & \text{if } i < 0 \end{cases}$
13: **end if**
14: **end for**
15: Add r to $S[s]$
16: **end for**
17: **return** S
18: **end procedure**

3.3.4 Querying the Database

The integration platform provides a REST API to interact with the system. This API allows both queries and modifications in the integration. Together with the REST API, a Java interface is also provided.

Currently, REST APIs are common within online services due to their simplicity. These APIs rely on HTTP verbs (i.e., GET, POST, PUT, DELETE) to perform operations (Table 3.3), that mimic the CRUD operations of database engines.

The database is also available for download, allowing to run the database embedded with Java applications eliminating the delay for network communication.

Table 3.3 – Description of the functions most commonly used to query the unified database.

	Functions	Method	Path Variable	Parameter
1.	`/metabolic/model/mmd/{model}/spi/{id}`	GET	model id	-
2.	`/metabolic/model/mmd/{model}/rxn`	GET	model	-
3.	`/metabolic/model/mmd/{model}/spi/ref/upload`	POST	model	file
4.	`/integration/explore/cpd/findByReferenceId`	GET	-	referenceid

Below, there is a description of the functions from Table 3.3:

1. Get the entity ModelSpecies, through the parameters model's name and speciesEntry.
2. Get a list of entity ModelReaction, through the parameter model's name.
3. Upload the SpeciesModel with references, through the parameters model's name and file.
4. Get the integration map, through the parameter ID of reference.

Chapter 4

Building a Human Metabolic Integrated Repository

The human GSSMs are models that are complex and difficult to integrate, thus, it is necessary to detain a broad knowledge about these, so that more efficient strategies can be defined, to execute that task and also be able to construct a Human Metabolic Integrated quality repository. In this chapter, the methods created for the metabolites and reactions' integration process, are described.

4.1 Implementation language

The framework developed in this work was implemented in Java language programming [64], making use of its main features, such as, the object-oriented patterns, the portability and a library (known as Application Programming Interface, API) with a wide range of packages [65]. Furthermore, this language allows an easy interaction with SQL server database and Neo4j server.

4.2 Issues in the Integration of Human Metabolic Models

Even though the SBML files of the human GSMMs follow mostly a standard format, most of the existing content does not follow that premise. This way, it is mandatory to follow strategies that lead to a consistent comparison between them. In order to design these strategies, there is the need to analyse the content of each used GSMM.

When intending to make a comparison between files it is crucial that the amount of data presented is similar in each one. As for the human GSMMs, the Compartment, Species and Reaction entities in particular must have an identical number of specimens. This fact is very important, since if there is a large differential at this level, there will be more variability between the models' data and, as a consequence, fewer parts in common. Taking it into account (as it was verified in the aforementioned Table 2.1) and also the fact that they are the most recent, the Recon 2 and HMR2.0 models were the ones selected for the execution of this work.

The main strategy for enabling a comparison of the models is to use the properties present in the non-redundant and inconsistent entities. Specifically, for the human GSMMs, the data that is normally present and that follows that principle are the references to the available external databases. These references are present in the annotations of the Metabolites (Species) entities. Therefore, if two metabolites of different models are annotated with the same reference, one can say, with a great amount of certainty that they are the same, but it cannot be completely certain upfront, since there can be cases where the metabolites were wrongly annotated, misleading these direct comparisons.

Although the existence of references is common in current models, there is a major gap at this level, making the comparison task difficult. Besides the existing gap regarding the coverage of metabolites essential to this information, there is also a lack of agreement on the types of used references, so it is frequent to have two metabolites in different models, which are the same, but that are not directly identified as such. An example of

that, is when a metabolite is annotated with a reference to the KEGG database and with another for ChEBI.

Some of the aforementioned problems can be solved by the adequate usage of the species' properties. However, it is necessary that the properties used within the search are trustworthy. Ideally, only the unique characteristics such as InChI should be used. However, the presence of such properties in human GSMMs is not very wide, so it is necessary to use other resources.

The reactions contained in human GSMMs do not have unique properties that can identify them easily. Therefore, the only way to integrate reactions is through the comparison of their species (metabolites-associated compartments) that bring up different kinds of problems, associated with metabolites' data that are not yet fully integrated. In addition, this task stumbles upon other problems such as the reversibility, the different balancing equations and the different cofactors of the reactions.

Other than that, each reaction belongs to a subsystem, with whom it is annotated. Sometimes equivalent reactions are located in different subsystems. This brings into question the integration, being mandatory to understand in which manner are the subsystems organized. A subsystem is a metabolic pathway that represents a series of chemical reactions occurring within a cell. The initial metabolite is typically modified by a sequence of chemical reactions until arriving to a final product. Bearing all that in mind, the study of subsystems contained in the Recon 2 and HMR 2.0 GSMMs allows for a better understanding of the real differences of content among these.

4.3 Local database

In order to organize the content and to ease its analysis, a local database was created. This database received only the data from Recon 2 and HMR 2.0. Therefore, an intensive study of the data can be accomplished, leading to a clarification of potential strategies capable of being used for the integration of GSMMs. In addition, possible errors were pursued and, whenever possible, rectified.

The local database is a related database written in the SQL language. This allows an easy interaction between Java and the MySQL server. Besides, the tables' visualization in MySQL enables a fast and effective manual error detection.

4.3.1 Structure

The local database has as entities the *Metabolite*, *MetaboliteXref*, *Compartment*, *Species*, *SpecieXref*, *Reaction*, *ReactionRight* and *ReactionLeft*. Each of these entities has many attributes:

- ID: the unique and primary key of all entities. It is a number generated automatically;
- Name;
- Entry: the unique key present in each model. In the *Metabolite* entity this property is altered. Each element takes on the entry of its species, without the suffix refering to the compartment (Figure 4.1);

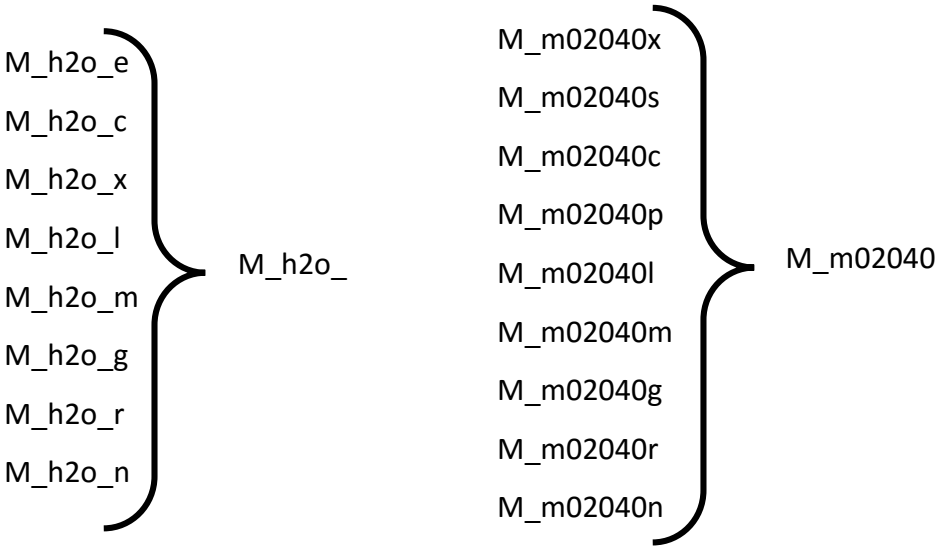


Figure 4.1 - A group of species entries is transformed into a metabolite entry. Recon 2 on the left side and HMR2.0 on the right side.

- Source: identify the origin of the element (e.g. Recon 2 or HMR 2.0).

Metabolite-only attributes:

- MCLASS: abbreviation of metabolite class. In this case, all of them are compounds.

Species-only attributes:

- InChI: textual identifier to define chemical substances;
- Formula: the expression that describes the composition of the compound;
- Charge: the net charge is the arithmetic sum of positive and negative charges.

MetaboliteXref and *SpeciesXref* attributes:

- Ref_type: abbreviation of reference type. The references can be from several sources, as citation, E.C number, database, etc. In these entities, all references are the database type;
- Tag: identify the origin of the reference (e.g. KEGG, ChEBI, HMDB, etc.);
- Value: unique identifier of the reference.

Reaction-only attributes:

- Orientation: defines the reaction's reversibility and direction. It can be from left to right or reversible;
- Subsystem: a pathway, where the reaction is included;
- Type: Identifies the type of reaction. It may be an internal, translocation, drain or biomass reaction.

ReactionLeft/*ReactionRight*-only attributes:

- Value: stoichiometric coefficient.

The distribution of these attributes by entity is illustrated in Figures 4.2 and 4.3, where the Entity-Relationship model is depicted. The tables (entities) are related with one another. Each metabolite has several species (since they represent the same compound), so these have a one-to-many relationship, acquiring each species the primary key (ID) of the corresponding metabolite. The remaining relationships are all

like that since a metabolite can have multiple references (metaboliteXref) and compartments just like each species can have multiple references.

Using the local database, it was possible to make several qualitative and quantitative data analyses. In addition, this database is an asset, since in this way there is not always the need to load the models' files, decreasing the time of data accesses.

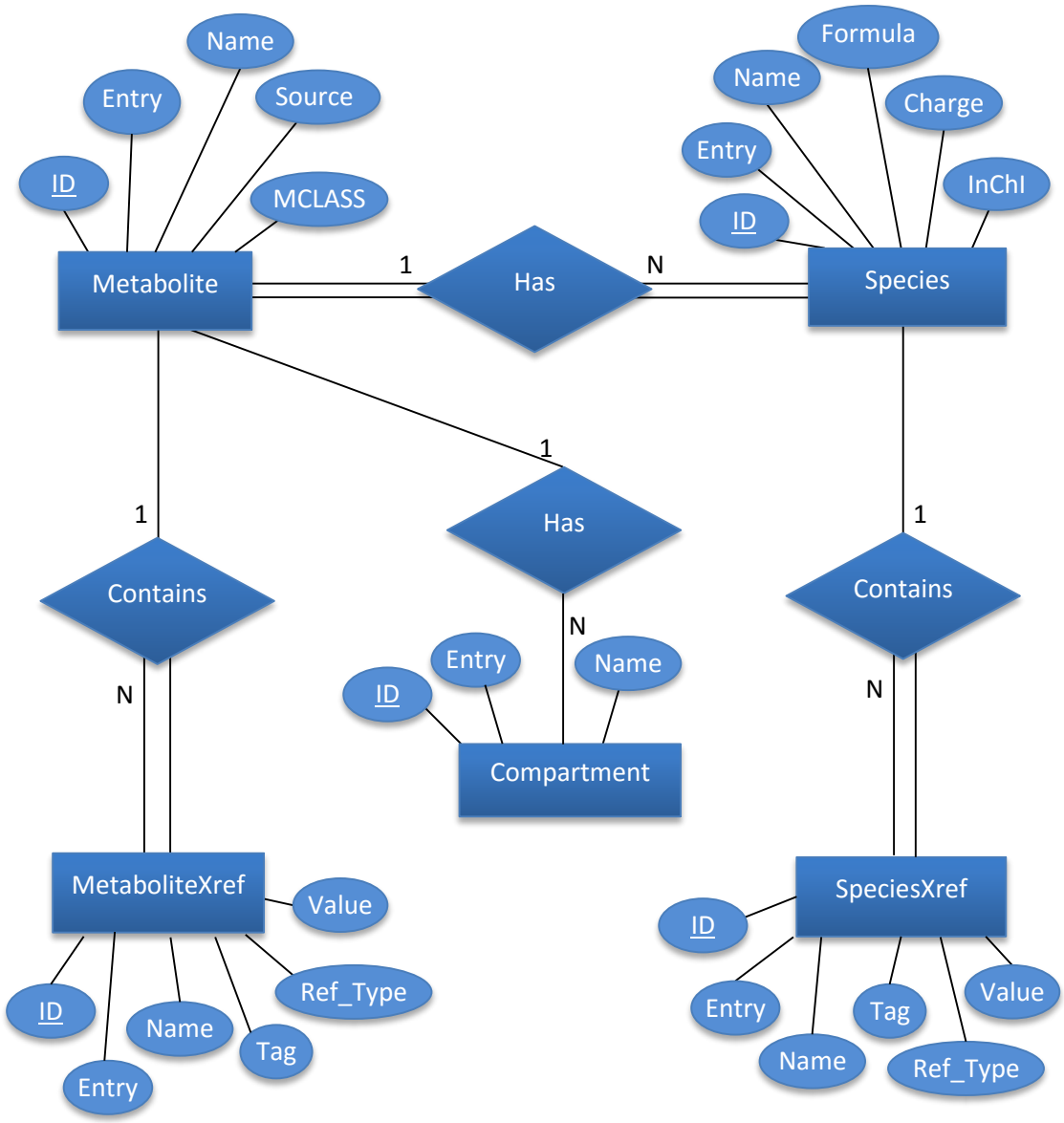


Figure 4.2 - Metabolite's Entity-Relationship Model.

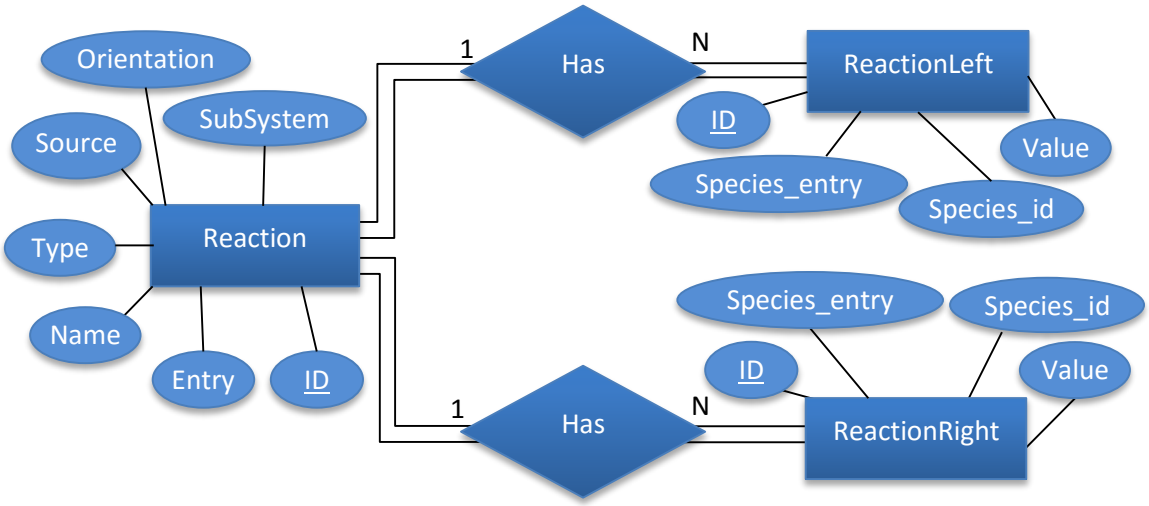


Figure 4.3 - Reaction's Entity-Relationship Model.

4.3.2 Data Loading from SBML

Both Recon 2 and HMR 2.0 are available in the SBML format. The core libraries of the OptFlux metabolic engineering platform [63] were used to import the models. This can be done by using the *biocomponents* and *utilities* package, which provide a *JSBMLReader* that uses the JSBML Java library to read and build a *Container* object with the components found in SBML file. However, the SBML format can be quite complex due to XML namespace extensions, such as by adding RDF terms to annotate entities. The Optflux reader ignores the RDF extension since it is irrelevant for mathematical modelling purposes. Therefore, it was necessary to apply some modifications to the original *JSBMLReader* since it did not meet the requirements for this project. These modifications included the adaptation to the present Gene-Protein-Rules in the human models.

Initially, the data of both models were processed aiming to adapt to the new common data warehouse correcting simultaneously some typing errors. This adaptation includes the selection of the useful and relevant data (such as the InChI property or the HMDB reference) for this work.

The models' data were imported, processed and exported to the database described in the previous section. This way it was possible to make a deeper and specific models'

analysis also allowing for an easier error detection. The errors found both within models in general or specific to an entity were rectified during the processing stage.

This processing is done in the *JSBMLReader* class, where the data from the file are filtered and rectified so that the extracted information is already prepared to be integrated in other platforms. In the case of InChI, for instance, there were several problems, such as lacking of the prefix "InChI =" before the code, the fact that the code is not within predefined standards by IUPAC, among others. Hence, it is guaranteed that only the correctly formatted InChIs are exported, the same happens with other properties. After extracting the file data, this information was placed in their respective entities (described in the previous section): *Metabolite*, *Compartment*, *MetaboliteXref*, *Species*, *SpecieXref*, *Reaction*, *ReactionLeft* and *ReactionRight*.

It was necessary the development of the *MetaboliteImpl* and *ReactionImpl* classes that get the present information from in the local database. In the *MetaboliteImpl* class, the relevant information is extracted from the *Container* and allocated in the first five entities mentioned above. Here, for each species a new element of each of these entities is created, except when one of these elements already exists. A new metabolite can be present in several compartments, and so it can be represented by several species. In this way, only one metabolite is created for several species.

The *ReactionImpl* class does the same as the previous, although adapted to the reactions. For each reaction, a new member of the entity is generated with the same name and also of the *ReactionLeft* and *ReactionRight* entities. The species usually named reactants are represented by *ReactionLeft* and the products by *ReactionRight*.

All these entities are tables that were implemented with the javax.persistence package. Posteriorly, with an open session (configured with the properties of the local database) the entities and its elements were written (in SQL language) in a beforehand created schema.

The data in the local database can be accessed by using the SQL language in MySQL Workbench or combining SQL with Java. The visualization of data in MySQL Workbench facilitates the manual detection of typing errors that were not expected. On the other

hand, SQL combined with Java allows all kind of data analysis and also its use for other tasks. Making use of the *HbmMetaboliteDaoImpl* class it is possible to do database queries and extract the desired information. In each method, a SQLQuery is created and executed through an open session for the database.

4.4 Integrating Human Models in the Graph Database

4.4.1 Integrating Metabolites

As stated previously, it is predictable that Recon 2 and HMR2.0 have multiple common entities. The most reliable method to compare metabolites is to see if they share the same references. This way, strategies were tested taking that assumption into account, one using the local database and the other, the unified resource described in the previous chapter.

The initial strategy, through the local database, verifies for each model's metabolite if it possesses external references (to databases). If this occurs, the references are used to search in the local database for metabolites with those references. Therefore, for each metabolite's reference a query is made particularly to the *MetaboliteXref* entity, where tag (reference's type) and value (ID of the reference) have to match simultaneously. In case a metabolite from another model with at least an equal reference, is found (tag and value) this is considered a pair. However, first it is verified if the previous method did not find more than one metabolite (from the other model), since sometimes there can be different metabolites with the same reference. Besides, a metabolite with two types of references (e.g. KEGG and ChEBI) can have two compatible metabolites in the other model, since there can be a metabolite with the reference of the KEGG type and the other with the ChEBI type.

In another strategy, the Recon 2 and HMR2.0 models were loaded into the unified database, making available the species, reactions and compartments of each model. Each reference of a new species (in the unified database) is integrated and grouped to a cluster, if there is at least another equivalent reference. Since the references are associated to their species, if a reference is grouped to a cluster, the species that have that reference will stay associated to the respective Cluster (Example in Figure 4.4). This way all the equal species are associated to the same cluster, so each cluster represents an integrated metabolite.

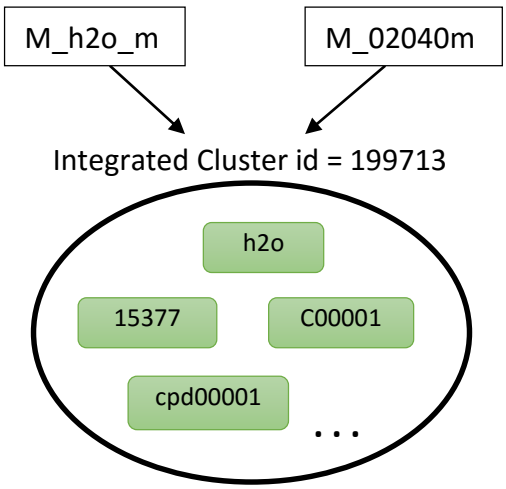


Figure 4.4 - Example of an integrated cluster. In the square label, are represented two species of water, through the entry used in the HMR2.0 and Recon 2 models, respectively. The references in the circle (cluster) are from the BiGG, ChEBI, KEGG and MetaCyc databases.

The fact that the integrated database contains the data of the most important databases, and combines them to form clusters, adds credibility and confidence in the integration of the metabolites. In order to benefit from this tool, it was necessary that the references would be added to each species. The *GenerateSpeciesMapping* receives as input a list of species entries of a model, and for each entry it gets its *Species* entity, through a query to the local database. Through the species, the references associated with it are retrieved. These results are put in a map, in which the specie's entry is the key and the value is another map, wherein the key is the reference's tag and the value is the unique identifier of the reference (example in Figure 4.5).

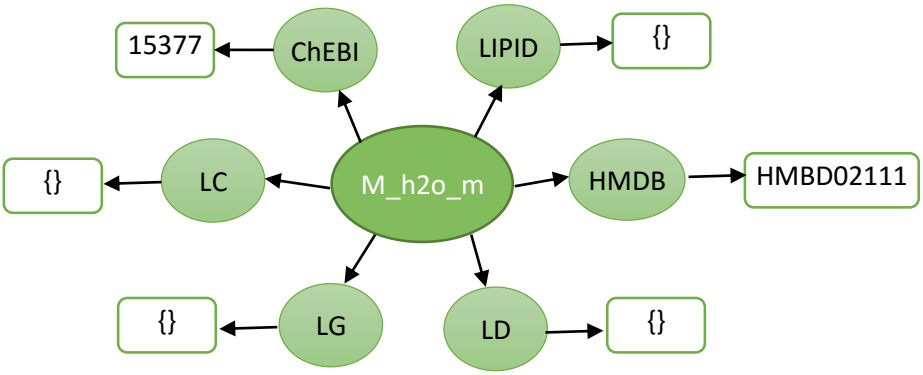


Figure 4.5 - Representative schema of an element in *SpeciesMapping*. LIPID-LIPIDMAPS; LC-LigandCompound; LG-LigandGlycan; LD-LigandDrug.

The references' types that are not present in a given species, are equally placed on the map, with an empty string as value. This is important to follow the formation rules of the unified database. This *SpeciesMapping* is converted into a text file, where the MetaboliteSpecies label followed by the labels of the references constitute the header and each line has the information in the map corresponding to the header. Thus, this file follows the constitution and nomenclature required, so that the upload in the unified database can be successful. The upload is done in the *Upload* class that uses the retrofit package. In order to upload the file, a connection to the unified database is prepared with `RestAdapter.Builder()`. The method created uses this connection for uploading, having as parameters the file and the model name, and is annotated with the POST HTTP verb. All these methods make it possible to populate the species (in the unified database) with its references present in the model.

In the *ClusterResults* class various methods were built, where the ultimate goal is the discovery of the clusterID's (Integrated Cluster ID for each distinct cluster in the unified database) in common (between the two models) and consequently the common metabolites. In order to fulfil this objective, several tasks are performed individually, for each model, to get a list of the clusterID's present in each. This is shown as Algorithm 2.

Algorithm 2: SpeciesClusterMapping

Input: S a set of species entities.
Output: $M -$ SpeciesClusterMapping

```
1:  Procedure ClusterMapping(S)
2:      for s ∈ S do
3:          M[m] ← {}                          ⇒ Initialize empty dictionary
4:          R ← s.getIdsCrossReferences()
5:          if R ≠ ∅ then
6:              P[p] = {}                      ⇒ Initialize empty dictionary
7:              for r ∈ R do
8:                  I ← findByReferenceId(r)
9:                  C ← getIdCluster(I)
10:                 if C ≠ ∅ then
11:                     p ← C
12:                     Add I to P[p]
13:                 end if
14:             end for
15:         end if
16:         Add P to M[s]
17:     end for
18:     return M
    end procedure
```

In the algorithm *SpecieClusterMapping*, starting from a set of species entities of a model, you gets for each species, the cluster's id to which it is associated, and the references responsible for that combination.

The resulting algorithm map is useful to understand if all references of a species are grouped in the same cluster. Sometimes, for some reason (such as a bad annotation of the species) the references of the same species are grouped into different clusters. In addition, there may be references that do not belong to any cluster.

If all references (of a species) are integrated within the same cluster, the id of this cluster is associated to the speciesEntry. This conclusion is stored in the map named *SpecieEntry_ClusterID*, in which the key is the species's entry and the value is the cluster's id (`Map<speciesEntry, clusterID>`). These data are also stored in a text file, where the first and second columns are the map's key and value, respectively, as mentioned above.

Upon completion of these tasks for each model, the list of unique clusterID's is taken from the *SpeciesEntry_ClusterID* map through of each speciesEntry. The shared clusterID's between the lists represent the common metabolites in the models. For every shared clusterID the speciesEntry with which it is associated (on the *SpecieEntry_ClusterID* map) is used, to get the metabolite of this species through the local database. Thus, it is possible to find the metabolites integrated between models.

The species without references will not have a clusterID associated. Therefore, it would be useful to populate these species with the respective existing references in the metabolic databases. This strategy can also amplify the odds of the species with a reference being associated to a cluster, since sometimes that same reference is not present in any existing cluster.

As already mentioned above, each species is composed by a set of data that besides describing it, helps in the identification process. Thus, some of these data were selected to be used in the pursuit of the lacking references. This search for more references through species' properties is made using the unified graph database in the Neo4j platform. In the *FindXrefs* class, the connection to the unified Neo4j graph database is done using the packages and methods made available online by the Neo4j platform. In this class, a specific node of a property is found by label, property's label and property's value.

This search is made by the execution of a Neo4j query. The query has the following example format:

```
MATCH (n: InChI) WHERE n.key= 'InChI=1S/Se/q-2' RETURN n
```

In this example, "InChI" is the label, "key" is the property's label, and "InChI=1S/Se/q-2" is the property's value. These variables differ depending on the property of the species that is being used. Each property has its specifications both in the unified database and in the model. Once the wanted node is found, its relationships, which have the desired label, are obtained. The relationship's label also varies depending on the property. Following the previous example, in this case the label is: "has_inchi", because this is the relationship's name of the references' nodes with the InChI's nodes. Consequently, it can find the nodes that are on the other side of the relationship. These nodes are only

really important if they are references, so they are just saved if the label coincides with one of the labels of the references (for example: LigandCompound, ChEBI, etc.).

Given that many species have no unambiguous properties, it is necessary to combine two properties reducing the likelihood of ambiguous references to be found. In this case, the nodes of interest are those that relate, at the same time, with the nodes of properties, and the ones which possess as label one of the ones in the references. The query has the following format:

```
MATCH (m:MolecularFormula)-[:has_molecular_formula]-(l:%s)-[has_name]-(n:Name)
WHERE m.key= {MolecularFormula} AND n.key={Name} RETURN l
```

In this example, "%s" can be any label reference. The "`l`" is already the reference node.

The information in selected nodes, i.e., the references, is placed on a map, with the same format of the *SpeciesMapping*, mentioned above, and it is also converted into a text file. Once again, the created file is uploaded in the unified database, updating the species' references.

All validated pairs of metabolites, were stored in an entity (the repository) created for that purpose.

4.4.2 Integrating Reactions

The reactions are more complex entities, since they are constituted by species and their stoichiometric values, thus the comparison between these becomes harder and requires more steps. These steps were implemented in several methods in the *MatchingReactions* class. Each reaction of a model is compared with all reactions of the other model, to find their correspondent. This comparison is actually a comparison between species, present in a reaction, since those are the ones that will dictate whether this reaction is compatible with another, from the other model, or not. Hereupon, the resulting map from the metabolite integration is crucial for developing a map that consists in a double key and value. The double key represents a clusterID-stoichiometric value pair that is exclusive for each species. Therefore, for each

species, a map is created, if there is a clusterID. The clusterID is simply taken from the *SpecieEntry_ClusterID* map.

The stoichiometric value demands a bigger complexity, since it can differ from one reaction to another. In order to get to know the stoichiometric values of each species and also the reactions in which it participates, the local database is queried. If the species has many stoichiometric values, the map is filled with multiple double keys of the same species (example: `215155=-1`, `215155=-3`, `215155=1)`. Each double key is associated to a list with the reactions that contain the pair represented in it (Figure 4.6). This process is shown as Algorithm 3.

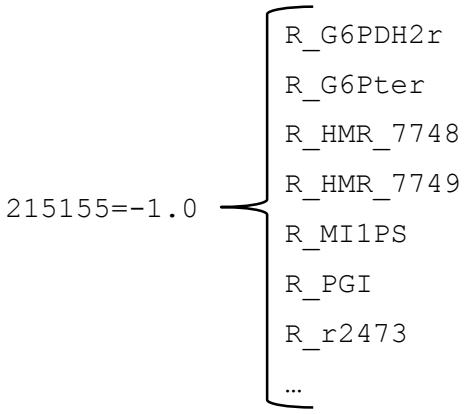


Figure 4.6 – Example of an element of the *SpeciesInReactions* map. In the left side is double key (clusterID and stoichiometric value) and in right side are the values (reactionsEntry).

In case a species is a reactant (left side of the reaction), its stoichiometric value is multiplied by -1. Consequently, all the reactant species will have a negative stoichiometric value enabling its distinction from the products (right side of reaction). If a species is a reactant and product in different reactions at the same time, it is represented with different double keys, with the same clusterID's, but different stoichiometric values (negative or positive). The information concerning the side of the reaction in which a species is will also be retrieved from the local database through the *ReactionLeft* and *ReactionRight* entities.

Algorithm 3: SpeciesInReactions

Input: $S - SpeciesEntry\_clusterId$ map
Output: $R - SpeciesInReactions$ map
Procedure $SpeciesInReactions(S)$
  for $s \in S$ **do**
    $R[r] \leftarrow \{\}$ ⇒ Initialize empty dictionary
    $C \leftarrow S[s]$
    if $C \neq \emptyset$ **then**
      for $side \in sides$ **do**
        $E \leftarrow getReactionEntries(s, side)$
        for $e \in E$ **do**
          $V \leftarrow getStoi(e, side, s)$
          $r[C] \leftarrow V$
          Add e to $R[r]$
        end for
      end for
    end if
  end for
  return R
end procedure

For each reactionEntry (of each model) present in the local database, it can be checked to which double key it is associated to in the *SpeciesInReactions* map. This enables the construction of another map, for each model, with the composition of each reaction. In other words, to each reactionEntry, is associated a list of clusterID-Stoichiometric Value pairs, creating a map *ReactionComposition* (`Map<reactionEntry, List<Map<clusterID=stoichiometricValue>>`). It is possible that not all species, within a certain reaction, have an associated clusterID, so it is checked if the list of the reaction's composition has the same size of the same reaction in the local database.

At the end of this method, the results are two *ReactionComposition* maps, one for Recon 2 and one for HMR2.0 (example in Figure 4.7). These two maps enable an efficient and effective comparison between reactions.

R_PGI→	{215155=-1.0}	{218178=1.0}
R_HMR_7749→	{215155=-1.0}	{218178=1.0}
R_NH4tn→	{213462=-1.0}	{213462=1.0}
R_HMR_4873→	{213462=-1.0}	{213462=1.0}

Figure 4.7 - Four examples of the reactions' composition in *reactionComposition* map. The first two (of different models) are the same reaction, the same happens with the rest.

During the comparison process, the various matching possibilities between the reactions are tested.

First, it is checked if the reactions are a perfect match. The perfect match is achieved if the reactants species of a reaction are equivalent to those of the other reaction, and the same happens for the products, being mandatory that the reactions have the same number of reactants and products. Taking into account that in the *ReactionComposition* map, the reactants and products are identified, respectively, with a negative and positive stoichiometric value, it suffices to verify if two composition lists, of different reactions, are the same to guarantee the premise mentioned above.

All reactions, for which the perfect match was not found, go through to the second test. This test is the reversible perfect match that consists in the previous strategy with some differences at the level of the stoichiometric value. Here, being the reactions reversible, they might be in different compliances. As an example, the reactants of the reaction 2 and the products of the reaction 3 can have the same elements, as well as the products of reaction 2 and reactants of reaction 3 (as shown in Figure 4.8).

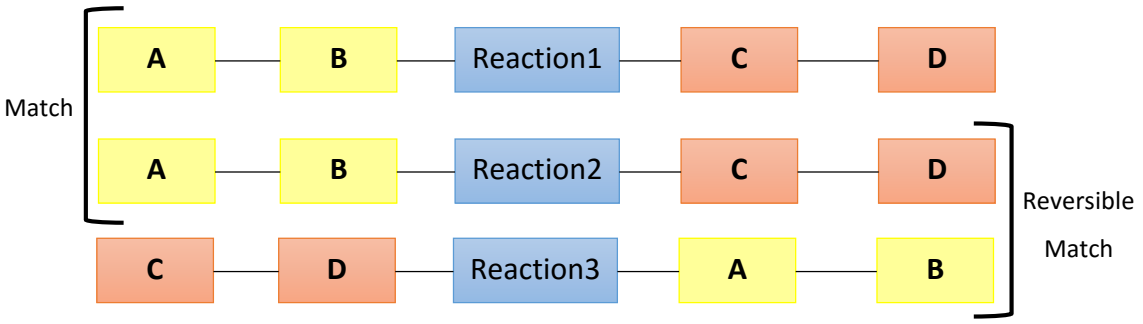


Figure 4.8 - Explanation of the compatibility of reactions. The A, B, C and D letters represent the species. The reactants and the products are, respectively, the left side and right side of each reaction.

Aiming to detect the occurrence of these cases, one of the *ReactionComposition* maps of the models to be compared is inverted, meaning that all the stoichiometric values are multiplied by -1. The intention behind this step is to invert the composition of the reaction, so the reactants (with negative stoichiometric values) became products (with positive stoichiometric values) and vice-versa. This way, it is possible to make the comparison of the reactions through the same method used in the perfect match (equals method), but using an original *ReactionComposition* map of a model and an inverted one of another model (example in Figure 4.9).

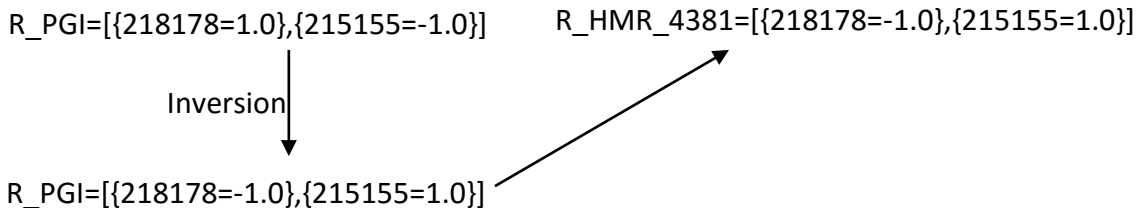


Figure 4.9 - Representative scheme of how the inversion of the composition of a reaction occurs and how it becomes equivalent to another from some other model. The reactionEntry R_PGI and the R_HMR_4381 belong, respectively, to the *ReactionComposition* map of Recon 2 and the HMR2.0.

The remaining reactions, which until this step do not have a pair, might actually, not be identical with any of the other model or may have slight differences, which makes difficult the formation of a pair. It is quite common to find identical reactions with different balancing equations, concretely in human GSMMs. A thorough study of these allows for the realization that there are identical reactions in different models, wherein

one is represented by their basic formula and the other has a proton added to the same formula. In order to fill these gaps, these reactions are subjected to a last test, the partial match.

In this test, it is accepted that there is one difference between the reactions. First, it is verified the case where the difference between reactions is a species that represents a proton ($^{1}H^{+}$). This compound is represented in the Recon 2 and HMR2.0, respectively, as `M_h_%` and `M_m02039%` (the % symbol represents any letter that represents the compartment). These species are associated to a clusterID with the number "218651". This way, it is easy to identify it, so it can be removed from the composition list of each reaction (that did not make it in the previous tests). After carrying out this task, these reactions are subjected again to the perfect match and reversible perfect match tests. Thereby, partially identical reactions are obtained where the difference is a proton.

The reactions that continue without a correspondence are exposed to one last test, in which the original composition lists are used again (without the proton being removed). In this case, the elements of each composition list of a reaction are compared one by one with the ones from another reaction. If just one element differs between the reactions, they are partially identical.

All the reactions that match in these tests are stored in distinct files, whose title identifies the match type, the header the models' name, and in each line there is a pair. The remaining are still unmatched.

Some reactions in the unified database have references for the KEGG database through the integration of their species. Thus, it is also possible to say that two reactions are equivalent, if these have the same reference in KEGG.

All validated pairs of reactions, were stored in an entity (the repository) created for that purpose.

Chapter 5

Results

In this section, the results of a detailed models' study are revealed. Aiming to proceed with the Recon 2 and HMR2.0 models' integration, it was fundamental to know their constitution. This enabled the perception of which were the, similarities and differences between models, once that, to find equalities, it is also needed a knowledge of the differences.

5.1 Characterizing the human models

5.1.1 Global characterization of the models

Primarily, a quantitative analysis of both models' data was done to enable a real awareness of the useful information that could be used to intersect both models and find similarities. The references to the external databases, as already mentioned throughout this work, are the most valuable and useful data that can be used for the integration. Hence, the priority was to know the extension of this property in each model. Both models have slightly more than half of the annotated metabolites with references (Figure 5.1). That is a value that falls short of the expectable and needed for a good integration.

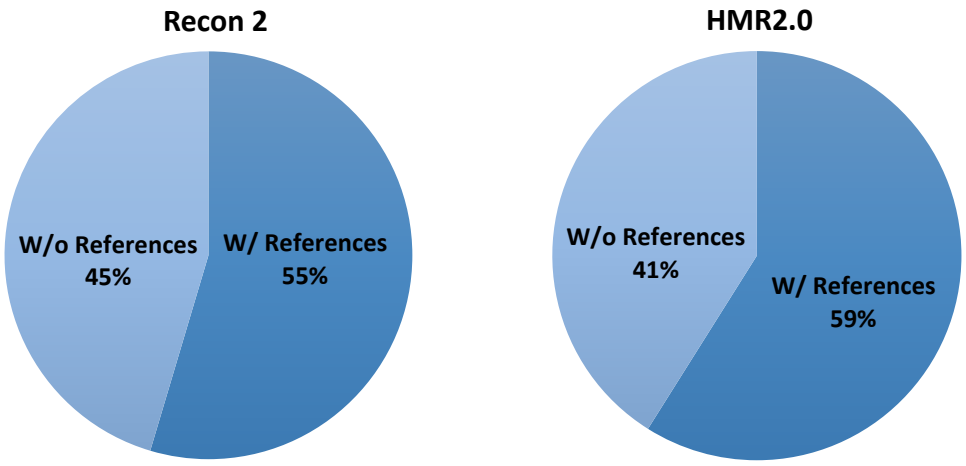


Figure 5.1 - Evaluation of the amount of metabolites that have references per model.

Each model has references for several types of databases, so it is pivotal to know which are the ones present in each model and which are the ones in common.

In the Recon 2 model, 61% of the species possess references of which 80%, 79% and 22% have, respectively, references of the ChEBI, HMDB, and KEGG (Compound) types (Figure 5.2). Once the sum of these percentages is not 100%, this shows that each species has more than one type of reference. Consequently, this situation was verified and analysed, being displayed in Figure 5.3 that for more than half of the species that own references, those are from the ChEBI and HMDB types.

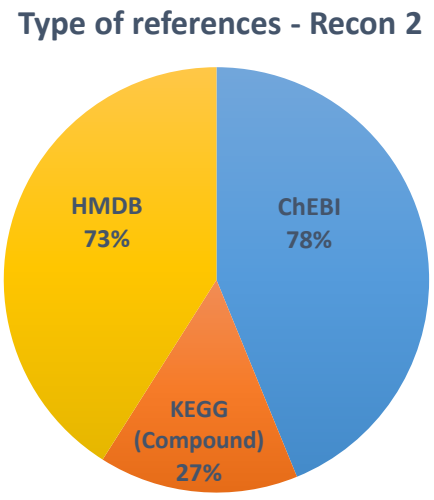


Figure 5.2 - In the Recon 2 model, 55% of the metabolites have references, and these are splitted in three types.

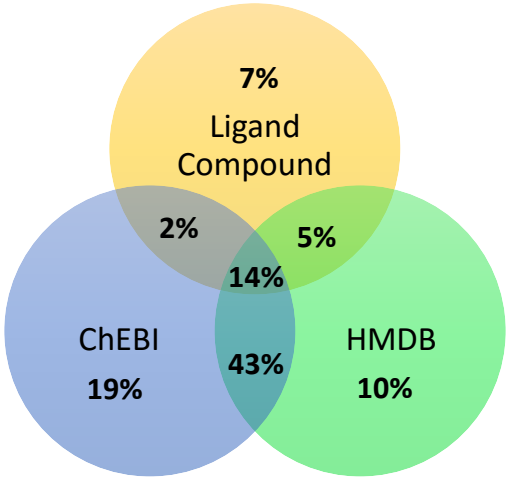


Figure 5.3 - In Recon 2, each species can have more than one type of reference. In this Venn diagram it can be seen the amount of metabolites that are dependent of just one type of reference and also the ones that possess multiple references.

Knowing that Recon 2 has more than one type of reference for each metabolite, this is very advantageous, since in addition to raising the odds of finding identical metabolites, it also increases the trust rating, if a metabolite with more than one type of reference of equal value, is found.

On the other hand, in the HMR2.0 model, it is possible to assume that each metabolite only owns one type of reference (Figure 5.4). Also, in this model, the type of ChEBI references is prevalent (74% of the metabolites with references).

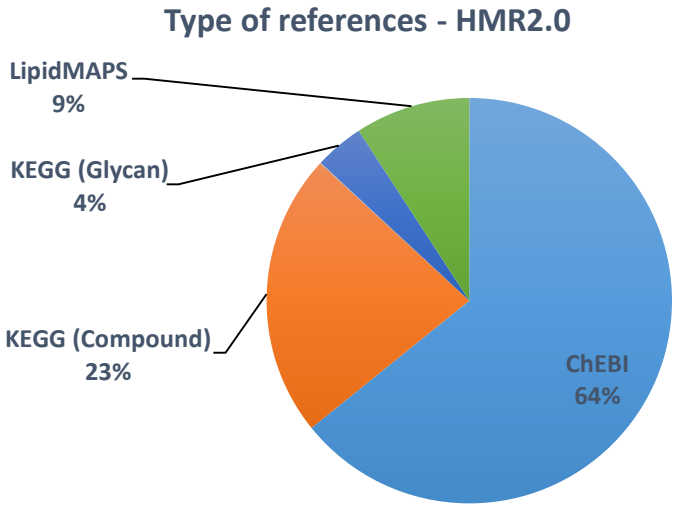


Figure 5.4 - In the HMR2.0 model, 59% of the metabolites have references, and these are splitted in four types.

Despite the fact that more than half the metabolites of both models have references, not all of them are useful to make a direct metabolites comparison. Only the ChEBI and LigandCompound are (common) to both models. Focusing in the metabolites with references, in Recon 2 85% of these have ChEBI and/or LigandCompound references, and in HMR2.0 87% have ChEBI or LigandCompound (since in this model each metabolite only has one reference, maximum) meaning that 46% and 51% of the metabolites respectively, can be compared using these references only.

As mentioned in the integration method of metabolites using the unified database, it is important to populate all the species with references, so that these can be associated to clusters. Taking into account that only 55% and 59% of the metabolites of the Recon 2 and HMR2.0 models respectively, have references, it is important to know the content present in metabolites (and consequently in species) (Table 5.1), in each model, so that it can be used in the search for more and new references.

Table 5.1 - Global analysis of the model's metabolites. The bottom part of the table bases itself only in the metabolites that do not have references (45% and 41%, in Recon 2 and HMR2.0, respectively).

Number of metabolites w/		Recon 2		HMR 2.0	
Formula		2518	96%	3067	97%
InChI		1330	51%	0	0%
References		1434	55%	1863	59%
W/o references	Formula	1132	95%	1257	97%
	InChI	184	15%	0	0%
	Nothing	60	5%	41	3%

The selection of properties (to be used in integrating metabolites using the unified database) was made based on quality and quantity of data meaning that the most unambiguous and present in most metabolites data, were ideal to identify the references of each species accurately and effectively. The selection was also different according to the model, since although both have the same basic constitution, they differ in some details.

As it can be seen in Table 5.1, almost all the metabolites of both models have a formula, while in InChI, besides only being found in Recon 2, it is only present in 51% of the metabolites. Although the InChI property is not so present in the Recon 2 as the formula, it is a more valuable data, since it is the most unambiguous and non-redundant. This is the ideal characteristic to find new references in a reliable manner. Another property that can also be used (that is always present), though it is one of the most ambiguous, is the name. In the HMR2.0 case, where the only properties available to be used are very ambiguous (Name and Formula), these should be used in simultaneously.

Apart from this type of properties, references to the least common databases can be used to find references to databases such as KEGG and ChEBI. The HMDB reference is really important since 10% of the metabolites have it alone and more than half have it along with other references. The BiGG and SEED databases can be withdrawn from the species' entry, removing the prefix "M_" and the suffix "_%", wherein "%", may be any letter that identifies the compartment. This becomes very useful since all the Recon 1 species have references for these databases. Being the Recon 2 its evolution one can only wait that lots of its species are within these databases.

In the HMR2.0, there is only a single characteristic that can be used alone besides the main references (ChEBI and KEGG). The reference LIPIDMAPS is exclusive in 9% of the metabolites.

In both models, there are four types of reactions: the Internal, Translocation, Drain and Biomass. These have about the same distribution in both models (Figure 5.5).

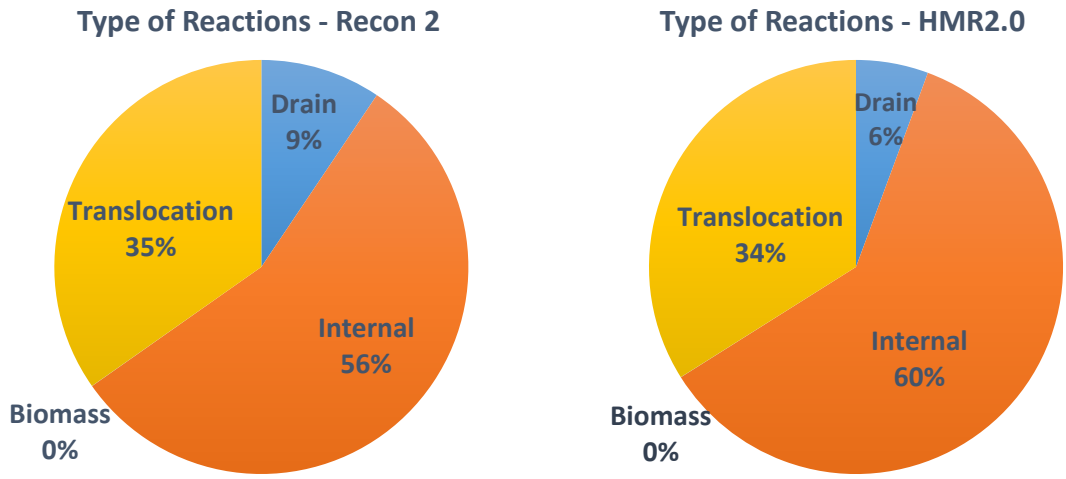


Figure 5.5 - Type of reactions in Recon 2 and HMR2.0 models.

The Internal type contains reactions in which the reactants and the products are in the same compartment. The translocation type represents the reactions that transport a compound of a compartment to the other. The drain reactions are metabolite uptake or excretion fluxes reactions. In Recon 2, these are characterized by having only a compound in its constitution. In turn, the HMR2.0 model uses the boundary compartment, to complete its reactions, so there is no reaction with one of the sides empty. These reactions belong to the Exchange subsystem, in both models. The biomass reactions are generic, where a big quantity of reactants results in one or a small number of products.

5.1.2 Examples of inconsistencies found

In the Recon 2 GSMM, 103 metabolites were found with more than a name associated, meaning that each of these metabolites had its origin in species that have the same compound, but have different names. The average rate of different names per metabolite is two. Despite not being erroneous since there are always synonyms, the difference between names is an inconsistency that can lead to integration errors. Since the use of different names for the same compound is one of the major obstacles to the

comparison between multiple models, it would be important to avoid situations like these within the actual model.

As shown in the Tables 5.2 and 5.3, cases were detected where for the same metabolite there were species with different references. In those examples, it can be seen that there is also the problem of different names for the same metabolite. Also, not all species are as complete as they could be, since some of its peers contain more references.

Table 5.2 - Representative table of an inconsistency situation. Specifically in the Name, ChEBI and Ligand Compound (LC) entities. Met_id-Metabolite_id.

ID	Entry	Name	Charge	Formula	ChEBI	HMDB	LC	Met_id
2079	M_adp_g	ADP(3-)	-3	C10H12N5O10P2	456216	1341	NULL	1090
2080	M_adp_e	ADP(3-)	-3	C10H12N5O10P2	456216	1341	NULL	1090
2081	M_adp_n	ADP(3-)	-3	C10H12N5O10P2	456216	1341	NULL	1090
2082	M_adp_l	ADP(3-)	-3	C10H12N5O10P2	456216	1341	NULL	1090
2083	M_adp_m	ADP	-3	C10H12N5O10P2	456216	1341	C00008	1090
2084	M_adp_r	ADP(3-)	-3	C10H12N5O10P2	456216	1341	NULL	1090
2085	M_adp_x	ADP(3-)	-3	C10H12N5O10P2	456216	1341	NULL	1090
2086	M_adp_c	ADP	-3	C10H12N5O10P2	16761	1341	C00008	1090

Table 5.3 - Representative table of an inconsistency situation. Specifically in the Name, ChEBI and Ligand Compound (LC) entities. Met_id-Metabolite_id.

ID	Entry	Name	Charge	Formula	ChEBI	HMDB	LC	Met_id
847	M_atp_x	ATP(4-)	-4	C10H12N5O13P3	57299	538	C00002	444
848	M_atp_c	ATP(4-)	-4	C10H12N5O13P3	30616	538	C00002	444
849	M_atp_m	ATP(4-)	-4	C10H12N5O13P3	57299	538	C00002	444
850	M_atp_l	ATP(3-)	-4	C10H12N5O13P3	57299	538	NULL	444
851	M_atp_n	ATP(3-)	-4	C10H12N5O13P3	57299	538	NULL	444
852	M_atp_r	ATP(4-)	-4	C10H12N5O13P3	57299	538	C00002	444
853	M_atp_e	ATP(4-)	-4	C10H12N5O13P3	57299	538	C00002	444
854	M_atp_g	ATP(3-)	-4	C10H12N5O13P3	57299	538	NULL	444

In Table 5.4, it can be seen that a metabolite from the GSMM HMR2.0 has species with different formulas. This difference is not substantial, but like in the previous cases, it can cause integration problems. Other than that, a search for the available reference (Ligand Compound) has revealed that none of the formulas are correct, since the formula

associated to this compound is C12H17O10.Na (Sodium 2-O-L-rhamnopyranosyl-4-deoxy-alpha-L-threo-hex-4-eno-pyranosiduronate).

Table 5.4 - Representative table of an inconsistency situation. Specifically in the Formula entity. LC-LigandCompound

ID	Entry	Formula	LC	Metabolite_id
10388	M_m02357x	C12H17O10	C08241	5427
10389	M_m02357c	C12H17O10.	C08241	5427
10390	M_m02357s	C12H17O10.	C08241	5427

5.2 Integration results

5.2.1 Metabolites

Through the metabolites integration method that uses the local database (described in section 4.4.1), a list of the metabolite pairs was created. This list was based in the use of these references, where it was demanded that there was at least one shared reference among the compared metabolites. It was not possible to demand that all the references of a Recon metabolite were similar to those of another HMR2.0 metabolite, since, as already mentioned, each HMR2.0 metabolite only has one type of reference. At the end of this method, 550 pairs of metabolites were obtained, in which each metabolite is unique.

In the initial list of metabolite pairs, 32 cases were found in which every distinct Recon 2 metabolite was compatible with more than a metabolite from the HMR2.0, the opposite also happened in 19 cases (see example in Figure 5.6).

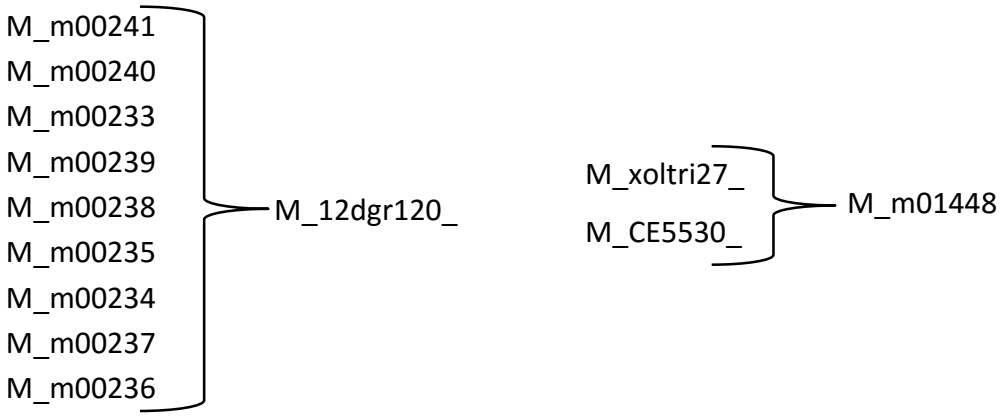


Figure 5.6 - Examples of different metabolites that are compatible with just a metabolite of the other model. The first example refers to this occurrence of the HMR2.0 to the Recon 2, and the second is the reverse.

After detecting this situation, a specific study of each model was made, to understand how many cases of different metabolites with the same references exist, and what motivates this. In order to proceed with this task, the same method (metabolites' integration) was used, but with the peculiarity of the input (entry of model 1, entry of model 2) being solely from a model, meaning that the model will run "against" itself. Besides that, in this case, it is accepted that, for each metabolite, there can be a list of compatible metabolites. This way, the different metabolites that possess the same references within each model, are found.

After the conclusion of this task, 62 cases were found in the Recon 2, of two different metabolites that share the same references and 1 case with three, making a total of 124 metabolites. In the HMR2.0, there are 14 cases that are very variable in terms of metabolites' number, which consist in: 1 case with three, nine and eleven; 3 cases of eight and 8 cases of two metabolites, totalling 63 metabolites. A manual verification of these cases, has allowed to realize that, in both models, these metabolites contain references that point to generic compounds or, when they point to a specific compound, its similar differs in just a small part of the structure. Additionally, only the name of these metabolites, apparently similar, is different, amongst the properties. The name's constitution of these metabolites, indicates the differences at the isomer structure's

level (e.g.: 1,2-diacylglycerol-LD-SM pool; 1,2-diacylglycerol-bile-PC pool). Deep down, these metabolites are the same compound but with different isomer structures.

This raises one more problem in terms of integration, because, even though these metabolites seem apparently similar, they participate in different reactions. If it were assumed that all the similar ones are the same, there would be a problem in the reactions, because lots of distinct reactions within the same model would become the same. Thereby, it would be useful if there were different references for each different isomer structure of a compound.

Using the metabolites' integration method that aims for the use of the unified database, after its species were populated with the references within the models, 809 clusterID in common between models were obtained, meaning 809 shared metabolites.

In order to increase these results, new references were sought through the Neo4j platform, as described in section 4.4.1. The general results of this search were very good since several species were populated with references (Table 5.5). In the Recon 2 case, the InChI property was the one where there were more populated species (44% of the species). Regarding the HMR2.0, as it was expected, once there are properties that are present in all the species, the Name & Formula properties were the ones that resulted in more populated species (25%). However, it was a far smaller percentage, when compared with the Recon 2's best result. Additionally, in the Recon 2's case, it was possible to use more properties (to obtain references) and none of them originated a result lower than 26%.

Table 5.5 - Number of species, which through a certain property, have obtained references, by search method for references in Neo4j platform of the unified database.

Recon 2			HMR2.0		
BiGG	1931	38%	LIPIDMAPS	145	2,4%
SEED	1422	28%	Name & Formula	1551	26%
InChI	2204	44%			
HMDB	1321	26%			

Even though all these data indicate that the Recon 2's species were populated with more references, only a detailed analysis can confirm if it was really an advantage. In Figure 5.7, that analysis is made, where, for each property, the number of populated species with references reached is divided into three categories. The first category (brown column), exposes the number of species that did not have any references and earned some, after being populated. The second category (green column) represents the number of species that already had references in the model, but were populated with new references. At last, the third category (blue column) exhibits the number of species that already had references and were not populated, since the ones earned were repeated.

As the species are populated, it is expected that the number of species associated with a cluster increases, since more species have references that may be contained in a cluster. Consequently, the number of clusterIDs (representing metabolites) also increases, so this was also a parameter analysed for each model. The yellow, green and blue lines represent, respectively, the Recon 2 and HMR2.0 models, and the clusterIDs in common (i.e. the metabolites). The criterion used to define the order in which the species in the unified database were populated was the number of species for which each property obtained references. Thus, the species (in the unified database) have been populated with references from each property in the following order: InChI, BiGG, SEED, HMDB, Name & Formula and LIPIDMAPS.

Thereafter, the unified graph database was updated with the new references. This way there were more species included in clusters, raising the number of generated metabolites and consequently the number of metabolites' pairs between models.

The number of species that were devoid of references and that earned them, is quite low, as opposed to the number of species that have earned new references, that is quite high. The number of integrated metabolites increased, every time the species in the unified database were populated with the references that came from the respective properties, with exception of the SEED property. It is normal that this happened, since the BiGG and SEED databases share most of their references. The number of shared

metabolites between models also increased, as the species were being populated, reaching a peak of 1105.

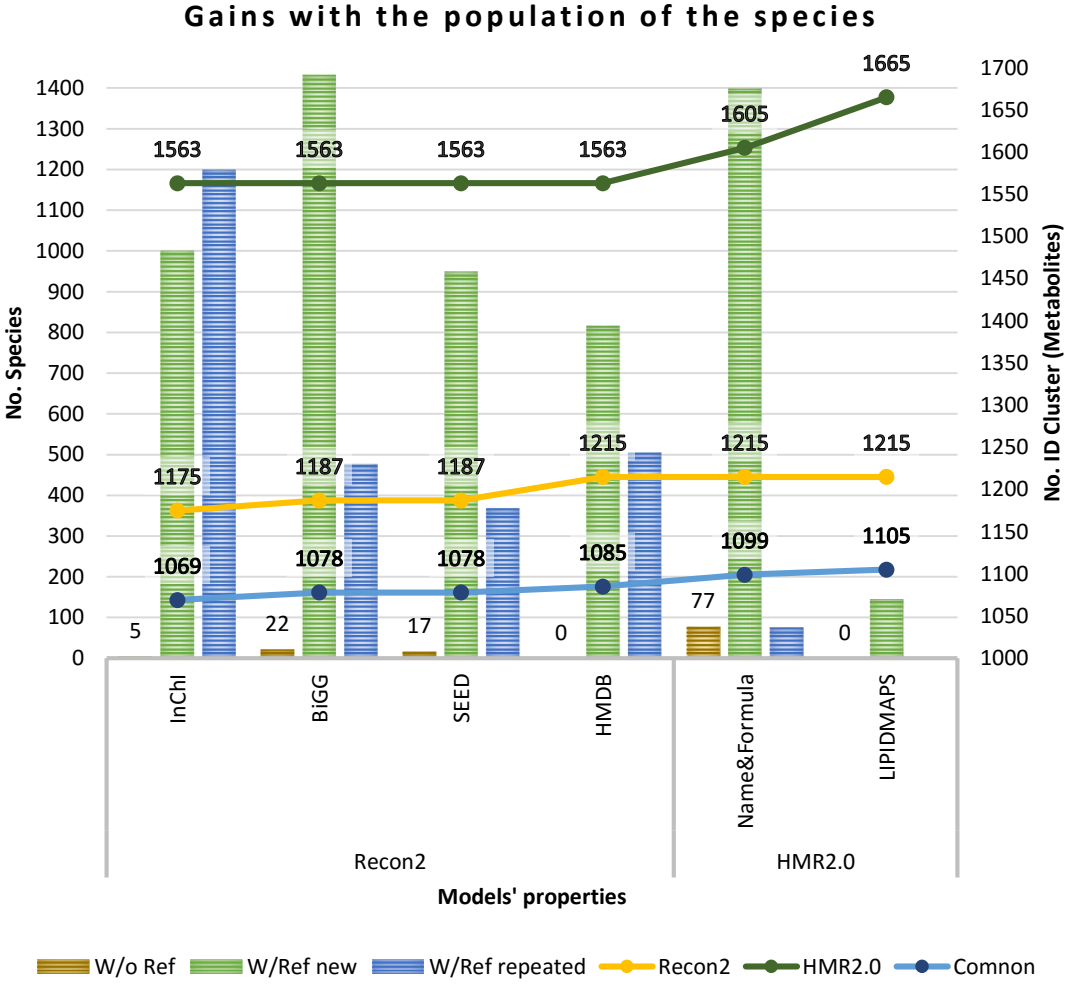


Figure 5.7 - Gains with the population of the species in unified database. The columns represent the number of species that were populated with references through a certain property. The lines are the clusterID numbers (metabolites) that were obtained (through the unified database), as the species were being populated.

From the list of 1105 metabolite pairs, 526 are in common with the list mentioned above (with 550 metabolites' pairs). This particular situation led to an increase of common metabolites that rose from 550 to 1129 (Figure 5.8).

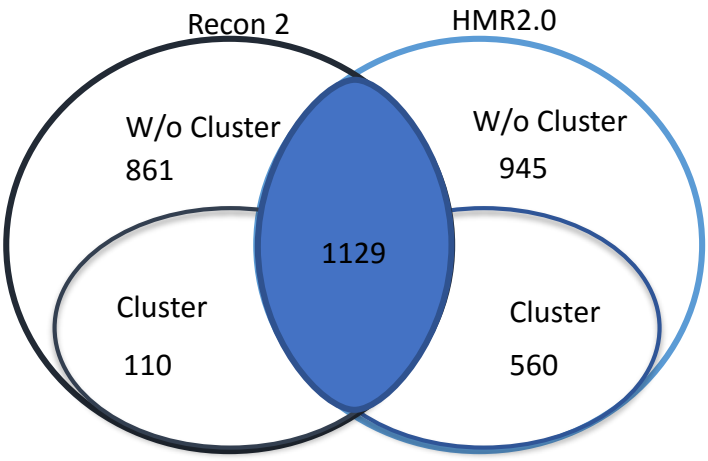


Figure 5.8 - Representation of numeric values of metabolites found through the Clusters of the unified graph database versus the metabolites of the local database (originated from the models). In the middle are the shared metabolites.

In this schema, it is possible to verify that there are metabolites, that despite being associated to a cluster, are still not part of the metabolites shared by the models. These metabolites are highly likely exclusive to each model, since, if they were identical, they would share the same cluster, and that does not happen.

In Recon 2, 33% of the metabolites were not associated to any cluster, nor shared metabolites through their possible references in the model. In the HMR2.0 case, these metabolites represent 30% of the total metabolites. The shared metabolites represent 43% and 36% of the Recon 2 and HMR2.0 metabolites, respectively.

5.2.2 Reactions

After the analysis and integration of the metabolites is done, the next step is the reactions' integration. This integration was done using the methods described in section 4.3.2. The general results are pretty satisfying (Table 5.6). In Recon 2, 44% of reactions are integrated. It is also important to note that these integrated reactions represent 71% of the reactions' number in the Recon 2's *ReactionComposition* map (62% of the total reactions). Concerning the HMR2.0, 43% of the reactions are integrated. Of the reactions present in the HMR2.0's *ReactionComposition* map (63% of the total reactions), 68% are integrated.

Table 5.6 - Number of unique reactions, by method, for which there is correspondence in the other model. The grey columns represent the results using the reversibility.

	Perfect		Proton		Partial		Total
Recon 2	2230	268	636	58	53	3	3248
HMR2.0	2479	263	596	66	67	3	3474

In this integration, using the reactions in general, it was detected that some reactions of different types formed a pair. This happens because the same reaction (constitution wise), can act in different ways in the human organism. In other words, a reaction can either be internal or translocation, since their compounds are the same, changing only, the compartments in which they are present. An example of that are the Recon 2 reactions with the R_GLYC3PFADm(5.1) and R_G3PD2m(5.2) entries.

$$\mathrm{M\_glyc3p\_m} + \mathrm{M\_fad\_m} \rightarrow \mathrm{M\_fad\_m} + \mathrm{M\_dhap\_m} \quad (5.1)$$

$$\mathrm{M\_glyc3p\_c} + \mathrm{M\_fad\_m} \rightarrow \mathrm{M\_fad\_m} + \mathrm{M\_dhap\_c} \quad (5.2)$$

Both reactions are identical to HMR2.0 reaction with the entry R_HMR_0449 (5.3).

$$\mathrm{M\_m01802m} + \mathrm{M\_m02914m} \rightarrow \mathrm{M\_m01803m} + \mathrm{M\_m01690m} \quad (5.3)$$

Despite these similarities, it is important to compare reactions of the same type, so that the integration's quality can be improved. Furthermore, it is possible to have a better perception of the differences between models. This way, the integration method was again executed using the reactions (of both models) of the same type. The total results obtained (Table 5.7) for each model, are different from the first, as there has been an increase in the number of unique reactions integrated (about +6.5%) for the Recon 2 model, and a decrease (about -1.1%) for the HMR2.0 model.

The justification for this change of values sticks with the fact that there are no longer reactions that stay paired with a reaction of a different type. When this is allowed, a reaction that is paired is forbidden of pairing with a reaction of its type. This event causes some noise that is mostly provoked by the drain-type reactions of the HMR2.0 model. These pair unduly, frequently, with Recon 2's translocation reactions.

It can be verified that the total number of reactions of each model, is not the same. These occurrences are due to the existence of equal reactions that occur in different compartments, but that are distinguished by the model. For the most part of the reactions, a pair that acts in the same compartment is found, but sometimes that does not happen. In these cases, it is accepted that two reactions with equal compositions but different compartments, form a pair. If these data were discarded, integration data that could be valuable for the formation of a unified model, would be lost. For instance, the Recon 2's reaction with the "R_TREH" entry is present in the Cytosol compartment, but it is not in that same HMR2.0 compartment (it is in Extracellular). In the future, it will be important to know that the "R_TREH" reaction has a compatible reaction (in the other model), even though it is in a different compartment.

It is also important to note that, both results (Tables 5.6 and 5.7) have shown that all the integration strategies (Perfect, Partial, etc.) used were useful, being the perfect match (as expected), responsible for majority of the results.

Table 5.7 - Number of unique reactions, by type and method, for which there is correspondence in the other model. The Recon2 is the grey column and the HMR2.0 is the white. * line with results using the reversibility.

	Internal		Transport		Drain	
Perfect	725	734	1484	1493	0	0
Perfect *	263	263	0	0	0	0
Proton	570	538	65	58	0	0
Proton *	58	66	0	0	0	0
Partial	0	0	0	0	294	281
Partial *	1	1	0	0	0	0
Subtotals	1617	1602	1549	1551	294	281

The results obtained from the integration by reaction type allowed the clarification and increase of the method's quality. The reactions of the drain type only corresponded through the partial integration method, due to the fact that these, in Recon 2, are only constituted by one compound, being that in HMR2.0 they are constituted by two, as explained above. This way, even though the integration was partial, the reactions (of

both models) that constituted a pair can be considered equal. This means that, in reality, there is only a pair of reactions (internal type) that differs in a compound.

Through this method, 3460 and 3434 reactions of the Recon 2 and HMR2.0, respectively, were integrated. This represents 75% and 67% of the reactions in the respective *ReactionComposition* maps. Taking into account, that the majority had a perfect match and, in the others, the difference was in a proton only, one can conclude that the rest of the reactions in the maps (25% and 33%) really are different. Having as a base that 62% and 63% of the Recon 2 and HMR2.0's total reactions, respectively, are in the respective *ReactionComposition* maps, meaning that 47% and 42% of the total reactions are integrated. Thus, it is likely that 16% of the Recon 2 and 21% of the HMR2.0 reactions are exclusive.

From the integrated reactions, 3253 (1529 Internal, 1444 Translocation and 280 Drain) are unique pairs, hence, the reactions of different compartments, are not contemplated here.

5.3 Analysis of a specific subsystem: the Glycolysis Pathway

The study of models by parts, as subsystems, allows for a real perception of the differences between the models, seeing that the same subsystem in different models can have small differences due to the way the pathway was completed. Besides that, there are also different subsystems among models. Seeing that this property also has different nomenclatures depending on the models, it is difficult to understand which are the common subsystems in both models. The created list of reactions' pairs through the references present in the unified database, allows for a primary perception of which subsystems will be common. This list is constituted by the common references (28), the respective entries of the reactions' models and subsystems associated to each reaction (Table A.1). This list has a total of 86 lines, being that there were various cases where the same reaction had multiple different entries due to the discrimination of the

compartment in which they act. Individually, in the unified database, the Recon 2 has 51 reactions with references (of which 33 are unique) and HMR2.0 has 1932 reactions with references (of which 1496 are unique).

Through Table A.1, it is possible to say with great certainty, that some subsystems are the same, like the Recon 2's Tetrahydrobiopterin metabolism and the HMR2.0's Biopterin metabolism. In other cases, the result is inconclusive, and it might be due to the fact that there are few examples or due to being ambiguous, e.g. the Recon 2's Sphingolipid metabolism subsystem has reactions that, in the HMR2.0, belong to the Sphingolipid metabolism subsystem just as to the Glycosphingolipid metabolism. Seeing that, a study of the subsystems in general, would be very laborious and long-lasting, a reference pathway was chosen for the effect.

The Glycolysis is one of the most important and described pathways. This is present in nearly all living organisms, since it is a great source of energy, that does not depend on oxygen [66]. Succinctly, in this pathway, the glucose is converted into pyruvate, through several steps. Considering these factors, there is a high probability of the Glycolysis pathway to be similar in both models, and also of being fully integrated.

So, its existence in the models was confirmed, and the nomenclature used was verified. In the Recon 2, this pathway is named "Glycolysis/gluconeogenesis" and in HMR2.0 is "Glycolysis / Gluconeogenesis". The Gluconeogenesis pathway, that is associated to Glycolysis, is basically its inverse, i.e. the Gluconeogenesis is production of glucose from pyruvate.

Although the names used are the same, there are small details that prevent using the same nomenclature for both models. In other words, for the reactions that are part of this pathway in each model, it is necessary to use two different strings. Therefore, the list of reactions' names and list of metabolites' names are obtained through of the local database by the correspondent subsystem (pathway).

As expected, the number of reactions, species and metabolites is very similar (Table 5.8). In particular, the number of reactions is the same even though that does not mean they are exactly the same.

Table 5.8 - Numeric results of subsystem Glycolysis.

Number of	Reactions	Species	Metabolites
Recon 2	40	84	51
HMR2.0	40	71	46

All the species of this subsystem are associated to a cluster, thus it is possible to make a complete comparison. Resorting to the methods described in section 4.4.1 (Integrating Metabolites) it was possible to find 37 clusters (in the unified database) which are equivalent to 37 metabolites shared by models. Individually, the number of clusters for the Recon 2 is the same as the metabolites' number (51) in the local database. In the case of the HMR2.0, the number of clusters is 44, so there are least two metabolites, compared with data from the local database. The analysis in this particular situation has led to the conclusion that what originated this situation are metabolites with a small difference in structure (in specific, beta carbon).

Although both models have 40 reactions, the shared reactions list has a total of 28 reactions, having each model 12 exclusive reactions, in this subsystem. These conclusions were drawn using the methods described in the section 4.4.2 (Integrating Reactions), where only the entries of these subsystem' reactions were given as input.

The manner in which the models represent the subsystem is illustrated in the schema of the Figure 5.9. The construction of this schema was possible due to the results obtained through the reactions' integration as well as the manual verification. The manual data analysis allowed the detection of reactions that transformed the same compounds but resorting to different co-factors, as it can be verified in the reactions number 2, 2.1 and 2.2 of the Table 5.9. These reactions cannot be considered equal, because their constitution is different, even though the substratum and the product are the same.

Moreover, cases were solved in which two different reactions of a model were the equal to just one from the other model. Again, this is explained by the reactions that occur in different compartments. For instance, the "R_ALDD2x" and "R_ALDD2xm" reactions (Recon 2). The first occurs in the cytosol compartment and the second in the mitochondrion. Both were compatible with the "R_HMR_8357" reaction (HMR2.0) that occurs in the mitochondrion compartment. This way, the "R_ALDD2xm" and

“R_HMR_8357” reactions were considered identical and the “R_ALDD2x”, exclusive to Recon 2 in this subsystem, not least because, it is identical to the reaction “R_HMR_1568”, which in the HMR2.0 case, participates in the Pyruvate subsystem.

Over the construction of the Glycolysis pathway schema, comparing with the literature [67], reactions were detected (identified in the models as part of the Glycolysis subsystem) that belong to another subsystems, but that produce or consume the Glycolysis substrate. These reactions are the numbers: 14, 14.1, 15, 24, 30 and 31. The subsystems to which they should belong are indicated in the schema. In the case of the reactions number 30, 31 of the Pyruvate metabolism, these are two reactions that from Acetate produce Acetyl-CoA, having as an intermediate the Acetyl adenylate compound. In the Glycolysis the Acetate consumption and the Acetyl-CoA production is the responsibility of only one reaction (present on the KEGG database with the ID number R00235).

The fundamental part of this pathway is shared by both models, existing parts that are represented exclusively by one of the models. The HMR2.0 model is the most complete since it as almost all the reactions that participate in this pathway [67]. Recon 2 focuses mostly in the essential reactions of this pathway, having several reactions for each step of it.

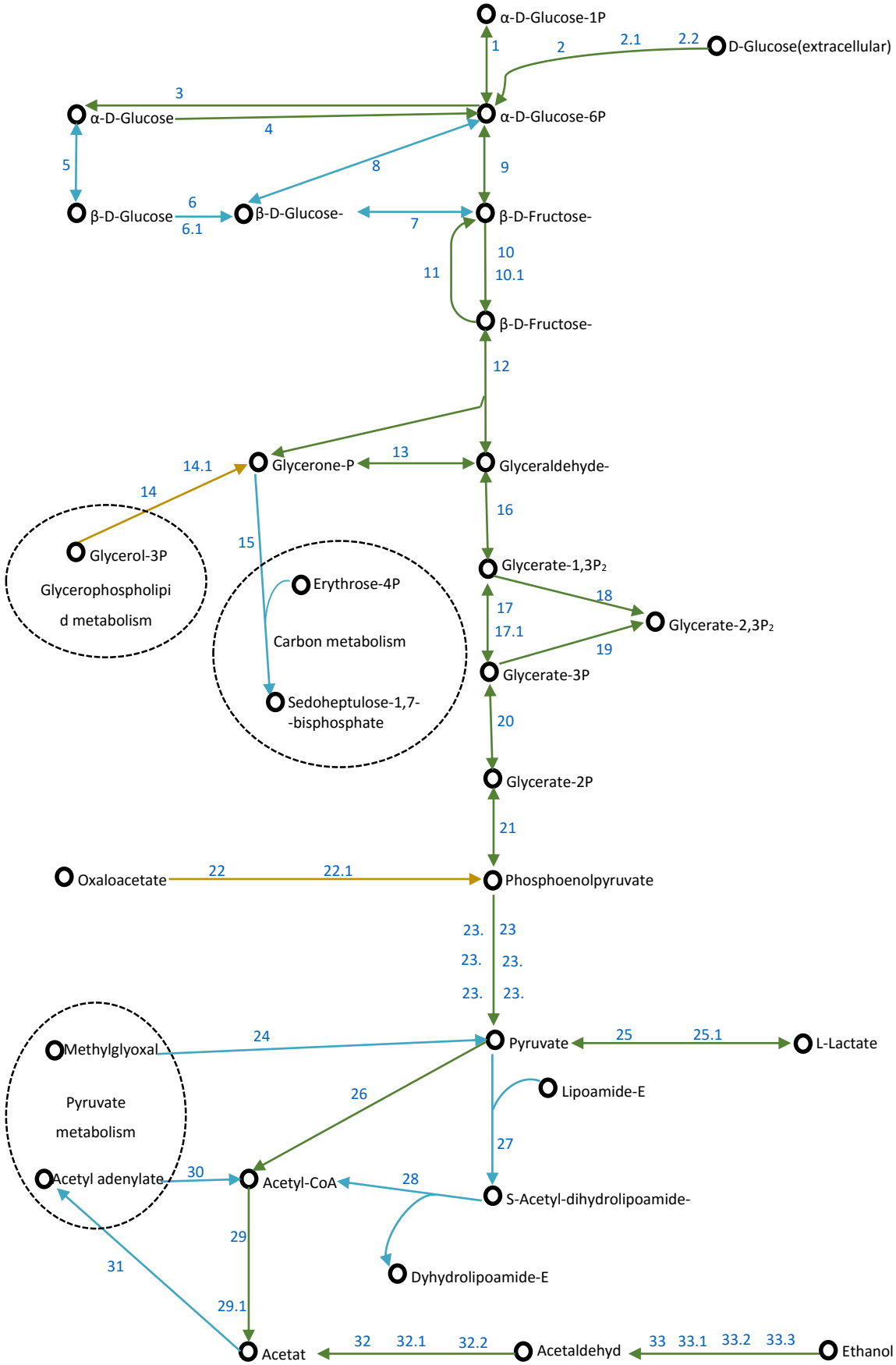


Figure 5.9 - Representation of glycolysis subsystem. The colours dark yellow, blue and green, represent, respectively, the unique reactions of Recon 2, HMR2.0, and the reactions shared by both.

Table 5.9 - Reactions' data (identified by a number) represented in the schema above. The colours dark yellow, blue and green, represent, respectively, the unique reactions of Recon 2, HMR2.0, and the reactions shared by both. The cofactors that are represented by the same order of its reactions, being "-" the discerning element of the left and right sides. The entry of the reactions are abbreviated, missing the prefix ("R_" and "R_HMR_"). Subtitle of the compartments: C-Cytosol, R-Endoplasmic reticulum, M-Mitochondria, X(Recon 2) -Peroxisome, P(HMR2.0)-Peroxisome.

Number	Recon 2	Cofactors	Comp.	HMR2.0	Cofactors	Comp.
1	PGMT		C	4396		C
2	r0354	ITP-IDP	C			
2.1	r0355	dATP-dADP	C			
2.2	CBPPer	Cbp+H^+-NH_4^++CO_2	R	8652	Cbp-NH_3+CO_2	R
3	G6PPer	H_2O-P_i	R	4521	H_2O-P_i	R
4	HEX1	ATP-ADP	C	4394	ATP-ADP	C
5				7745		C
6				7746	ATP-ADP	C
6.1				7747	ADP-AMP	C
7				7749		C
8				7748		C
9	PGI		C	4381		C
10	PFK	ATP-ADP	C	4379	ATP-ADP	C
10.1				4301	UTP-UDP	C
11	FBP		C	4377		C
12	FBA		C	4375		C
13	TPI		C	4391		C
14	G3PD2m	FAD-$FADH_2$	C & M			
14.1	r0202	NAD^+-NADH+H^+	X			
15				4355		C
16	GAPD	P_i+NAD^+-NADH+H^+	C	4373	P_i+NAD^+-NADH+H^+	C
17	PGK	ADP-ATP	C	4368	ADP-ATP	C
17.1	ACYP		C	4370		C
18	DPGM		C	4371		C
19	DPGase	H_2O- P_i	C	4372	H_2O- P_i	C
20	PGM		C	4365		C
21	ENO	-H_2O	C	4363	-H_2O	C
22	PEPCK	GTP-GDP+CO_2	C			
22.1	PEPCKm	GTP-GDP+CO_2	M			
23	PYK	ADP-ATP	C	4358	ADP-ATP	C
23.1	r0165	UDP-UTP	C			
23.2	r0280	dADP-dATP	C			
23.3	r0153	CDP-CTP	C			
23.4	r0413	dGDP-dGTP	C			
23.5	r0122	GDP-GTP	C			
24				4360	H_2O + $NADP^+$ - NADPH + H^+	C

25	r0173	NAD^+-NADH+H^+	X	4281	NADH+H^+-NAD^+	P
25.1	LDH_L	NAD^+-NADH+H^+	C	4388	NADH+H^+-NAD^+	C
26	PDHm	CoA +NAD^+-NADH + H^+ + CO2	M	4137	CoA +NAD^+-NADH + H^+ + CO2	M
27				6410	-CO_2	
28				6412	CoA-	M
29	ACS	CoA+ ATP - AMP+PP_i	C	4097	CoA+ ATP - AMP+PP_i	C
29.1	ACSm	CoA+ ATP - AMP+PP_i	M	4099	CoA+ ATP - AMP+PP_i	M
30				4108	CoA-AMP	C
31				4133	ATP- PP_i	C
32	ALDD2y	H_2O+$NADP^+$-NADPH+H	C	4283	H_2O+$NADP^+$-NADPH+H	C
32.1	ALDD2xm	H_2O+NAD^+-NADH+H	M	8357	H_2O+NAD^+-NADH+H	M
32.2	ALDD2x	H_2O+NAD^+-NADH+H	C			
33	ALCD2yf	$NADP^+$-NADPH+H	C	3907	$NADP^+$-NADPH+H	C
33.1	ALCD2if	NAD^+-NADH+H	C	3905	NAD^+-NADH+H	C
33.2	ETOHMO	O_2 + NADPH+H-$NADP^+$ + H_2O	C	8757	O_2 + NADPH+H-$NADP^+$ + H_2O	C
33.3	CAT2p	H_2O_2-H_2O	X	8360	H_2O_2-H_2O	P

Chapter 6

Conclusions

The main objective of this work was to build an integrated, unified and global repository of the human metabolism. This presupposed the design of a pipeline, to integrate the most important entities of the models, which are the metabolites and the reactions. The base models used for this integration were the Recon 2 and HMR2.0 GSMMs, since they are the most up to date.

The initial study made to the models revealed the necessity to extend the annotation of the species, with additional references from external databases. The usage of a unified database revealed to be very advantageous, since it increased the detection rate by more than 50%, of the common metabolites between the models. The effective integration of the metabolites is essential for the integration of the reactions.

As result of the integration, a repository was built, that given an entry, it can easily tell if there is a corresponding metabolite or reaction in the other model. This is very important so that, in the future, a complete and unified model can be created.

The manual analysis of the Glycolysis subsystem, gave us the conclusion that the designed methods of this work, to automatically integrate the metabolites and the reactions of the Recon 2 and the HMR2.0 models, gave us a very satisfying initial results. To consider the integration perfect is relative, since the essential part is to know, which are the metabolites and reactions that the models share. So, to consider two reactions identical, where the difference is in the compartment where they act within their model,

is not wrong, since they are actually the same. Beyond that, this information can be helpful in the future to create a unified model. In the construction of a unified model, it is important to know that there is a shared reaction between models, even if it is present in different compartments in each model. It is clear that, in the study of the Glycolysis subsystem, it was important to clarify these little differences, so that a detailed analysis could be performed, that would highlight the difference between models.

A limitation of this work is the lack of unique identifiers for all the metabolites. This limits severely the automatic integration, since it opens space for ambiguity, and it is not possible to tell with any certainty, if the non-integrated metabolites and reactions, are actually different. Therefore, as a future work, it would be important to manually verify, for the remaining metabolites and reactions, their identity and if they are exclusive to each model. Strategies to identify the common subsystems could aid in this task.

Appendix A

Results

Table A.1 - Integration's result of the model's reactions from Recon 2 and HMR2.0, employing the direct comparison method through of the KEGG (Reaction) references (common element).

Recon 2			HMR2.0	
Entry	Subsystem	KEGG	Entry	Subsystem
R_r0013	Aminosugar metabolism	R00022	R_HMR_3988	Amino sugar and nucleotide sugar metabolism
R_RE3519X	Arachidonic acid metabolism	R07036	R_HMR_1058	Arachidonic acid metabolism
R_RE3519X	Arachidonic acid metabolism	R07036	R_HMR_1057	Arachidonic acid metabolism
R_RE3519X	Arachidonic acid metabolism	R07036	R_HMR_1055	Arachidonic acid metabolism
R_RE3519R	Arachidonic acid metabolism	R07036	R_HMR_1058	Arachidonic acid metabolism
R_RE3519R	Arachidonic acid metabolism	R07036	R_HMR_1057	Arachidonic acid metabolism
R_RE3519R	Arachidonic acid metabolism	R07036	R_HMR_1055	Arachidonic acid metabolism
R_RE3519C	Arachidonic acid metabolism	R07036	R_HMR_1058	Arachidonic acid metabolism
R_RE3519C	Arachidonic acid metabolism	R07036	R_HMR_1057	Arachidonic acid metabolism
R_RE3519C	Arachidonic acid metabolism	R07036	R_HMR_1055	Arachidonic acid metabolism
R_RE3520N	Arachidonic acid metabolism	R07039	R_HMR_1043	Arachidonic acid metabolism

Recon 2			HMR2.0	
Entry	Subsystem	KEGG	Entry	Subsystem
R_RE3520N	Arachidonic acid metabolism	R07039	R_HMR_1045	Arachidonic acid metabolism
R_RE3520N	Arachidonic acid metabolism	R07039	R_HMR_1049	Arachidonic acid metabolism
R_RE3520N	Arachidonic acid metabolism	R07039	R_HMR_1048	Arachidonic acid metabolism
R_RE3520M	Arachidonic acid metabolism	R07039	R_HMR_1043	Arachidonic acid metabolism
R_RE3520M	Arachidonic acid metabolism	R07039	R_HMR_1045	Arachidonic acid metabolism
R_RE3520M	Arachidonic acid metabolism	R07039	R_HMR_1049	Arachidonic acid metabolism
R_RE3520M	Arachidonic acid metabolism	R07039	R_HMR_1048	Arachidonic acid metabolism
R_RE3520E	Arachidonic acid metabolism	R07039	R_HMR_1043	Arachidonic acid metabolism
R_RE3520E	Arachidonic acid metabolism	R07039	R_HMR_1045	Arachidonic acid metabolism
R_RE3520E	Arachidonic acid metabolism	R07039	R_HMR_1049	Arachidonic acid metabolism
R_RE3520E	Arachidonic acid metabolism	R07039	R_HMR_1048	Arachidonic acid metabolism
R_RE3520C	Arachidonic acid metabolism	R07039	R_HMR_1043	Arachidonic acid metabolism
R_RE3520C	Arachidonic acid metabolism	R07039	R_HMR_1045	Arachidonic acid metabolism
R_RE3520C	Arachidonic acid metabolism	R07039	R_HMR_1049	Arachidonic acid metabolism
R_RE3520C	Arachidonic acid metabolism	R07039	R_HMR_1048	Arachidonic acid metabolism
R_r0744	Bile acid synthesis	R04813	R_HMR_1644	Steroid metabolism
R_r0744	Bile acid synthesis	R04813	R_HMR_1642	Bile acid biosynthesis
R_TXASr	Eicosanoid metabolism	R02268	R_HMR_1313	prostaglandin biosynthesis
R_RE3556C	Eicosanoid metabolism	R04565	R_HMR_1401	prostaglandin biosynthesis
R_RE3567C	Eicosanoid metabolism	R05057	R_HMR_1323	prostaglandin biosynthesis
R_RE3566C	Eicosanoid metabolism	R05056	R_HMR_1322	prostaglandin biosynthesis
R_r0717	Fatty acid oxidation	R04738	R_HMR_3122	Beta oxidation of even-chain fatty acids (mitochondrial)
R_r0717	Fatty acid oxidation	R04738	R_HMR_3079	Beta oxidation of even-chain fatty acids (peroxisomal)

Recon 2			HMR2.0	
Entry	Subsystem	KEGG	Entry	Subsystem
R_r0729	Fatty acid oxidation	R04744	R_HMR_3143	Beta oxidation of even-chain fatty acids (mitochondrial)
R_r0729	Fatty acid oxidation	R04744	R_HMR_3091	Beta oxidation of even-chain fatty acids (peroxisomal)
R_r0734	Fatty acid oxidation	R04749	R_HMR_3157	Beta oxidation of even-chain fatty acids (mitochondrial)
R_r0734	Fatty acid oxidation	R04749	R_HMR_3099	Beta oxidation of even-chain fatty acids (peroxisomal)
R_r0731	Fatty acid oxidation	R04746	R_HMR_3150	Beta oxidation of even-chain fatty acids (mitochondrial)
R_r0731	Fatty acid oxidation	R04746	R_HMR_3095	Beta oxidation of even-chain fatty acids (peroxisomal)
R_r0720	Fatty acid oxidation	R04740	R_HMR_3129	Beta oxidation of even-chain fatty acids (mitochondrial)
R_r0720	Fatty acid oxidation	R04740	R_HMR_3083	Beta oxidation of even-chain fatty acids (peroxisomal)
R_r0721	Fatty acid oxidation	R04740	R_HMR_3129	Beta oxidation of even-chain fatty acids (mitochondrial)
R_r0721	Fatty acid oxidation	R04740	R_HMR_3083	Beta oxidation of even-chain fatty acids (peroxisomal)
R_r0661	Fatty acid oxidation	R04170	R_HMR_3136	Beta oxidation of even-chain fatty acids (mitochondrial)
R_r0661	Fatty acid oxidation	R04170	R_HMR_3087	Beta oxidation of even-chain fatty acids (peroxisomal)
R_T2M26DCOAHLx	Limonene and pinene degradation	R06412	R_HMR_9566	Isolated
R_T2M26DCOAHLx	Limonene and pinene degradation	R06412	R_HMR_9565	Isolated
R_T2M26DCOAHLm	Limonene and pinene degradation	R06412	R_HMR_9566	Isolated
R_T2M26DCOAHLm	Limonene and pinene degradation	R06412	R_HMR_9565	Isolated
R_C2M26DCOAHLx	Limonene and pinene degradation	R06411	R_HMR_9567	Isolated

Recon 2			HMR2.0	
Entry	Subsystem	KEGG	Entry	Subsystem
R_C2M26DCOAHLx	Limonene and pinene degradation	R06411	R_HMR_9568	Isolated
R_C2M26DCOAHLm	Limonene and pinene degradation	R06411	R_HMR_9567	Isolated
R_C2M26DCOAHLm	Limonene and pinene degradation	R06411	R_HMR_9568	Isolated
R_RE1860C	Miscellaneous	R03538	R_HMR_9471	Isolated
R_RE1860E	Miscellaneous	R03538	R_HMR_9471	Isolated
R_DOLPH_Ler	N-glycan synthesis	R06258	R_HMR_7274	N-glycan metabolism
R_DOLPH_Uer	N-glycan synthesis	R06258	R_HMR_7274	N-glycan metabolism
R_SMS	Sphingolipid metabolism	R08969	R_HMR_0736	Sphingolipid metabolism
R_SMS	Sphingolipid metabolism	R08969	R_HMR_0735	Sphingolipid metabolism
R_SMPD3g	Sphingolipid metabolism	R02541	R_HMR_8242	Sphingolipid metabolism
R_SMPD3g	Sphingolipid metabolism	R02541	R_HMR_0797	Glycosphingolipid metabolism
R_SMPD3g	Sphingolipid metabolism	R02541	R_HMR_0795	Sphingolipid metabolism
R_SMPD3l	Sphingolipid metabolism	R02541	R_HMR_8242	Sphingolipid metabolism
R_SMPD3l	Sphingolipid metabolism	R02541	R_HMR_0797	Glycosphingolipid metabolism
R_SMPD3l	Sphingolipid metabolism	R02541	R_HMR_0795	Sphingolipid metabolism
R_CYSO	Taurine and hypotaurine metabolism	R00893	R_HMR_3908	Bile acid biosynthesis
R_r0708	Tetrahydrobiopterin metabolism	R04639	R_HMR_4817	Biopterin metabolism
R_r0708	Tetrahydrobiopterin metabolism	R04639	R_HMR_4818	Biopterin metabolism
R_r0709	Tetrahydrobiopterin metabolism	R04639	R_HMR_4817	Biopterin metabolism
R_r0709	Tetrahydrobiopterin metabolism	R04639	R_HMR_4818	Biopterin metabolism
R_r0778	Tetrahydrobiopterin metabolism	R05048	R_HMR_4836	Biopterin metabolism
R_r0778	Tetrahydrobiopterin metabolism	R05048	R_HMR_4835	Biopterin metabolism
R_r0777	Tetrahydrobiopterin metabolism	R05048	R_HMR_4836	Biopterin metabolism

Recon 2			HMR2.0	
Entry	Subsystem	KEGG	Entry	Subsystem
R_r0777	Tetrahydrobiopterin metabolism	R05048	R_HMR_4835	Biopterin metabolism
R_r0120	Tetrahydrobiopterin metabolism	R00428	R_HMR_4170	Biopterin metabolism
R_r0120	Tetrahydrobiopterin metabolism	R00428	R_HMR_4169	Biopterin metabolism
R_r0121	Tetrahydrobiopterin metabolism	R00428	R_HMR_4170	Biopterin metabolism
R_r0121	Tetrahydrobiopterin metabolism	R00428	R_HMR_4169	Biopterin metabolism
R_5HTRPDOX	Tryptophan metabolism	R02702	R_HMR_6716	Phenylalanine, tyrosine and tryptophan biosynthesis
R_3HAO	Tryptophan metabolism	R02665	R_HMR_4228	Nicotinate and nicotinamide metabolism
R_r0716	Unassigned	R04738	R_HMR_3122	Beta oxidation of even-chain fatty acids (mitochondrial)
R_r0716	Unassigned	R04738	R_HMR_3079	Beta oxidation of even-chain fatty acids (peroxisomal)
R_r1377	Unassigned	R06982	R_HMR_3935	Tyrosine metabolism
R_MMEm	Valine, leucine, and isoleucine metabolism	R02765	R_HMR_3213	Valine, leucine, and isoleucine metabolism

Bibliography

[1] K. Radrich, Y. Tsuruoka, P. Dobson, A. Gevorgyan, N. Swainston, G. Baart*, et al.*, "Integration of metabolic databases for the reconstruction of genome-scale metabolic networks.," *BMC systems biology,* vol. 4, p. 114, 2010.

[2] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, "Basic Principles," in *Systems Biology in Practice*, ed: Wiley-VCH Verlag GmbH & Co. KGaA, 2005, pp. 1-17.

[3] N. M. Luscombe, D. Greenbaum, and M. Gerstein, "What is bioinformatics? A proposed definition and overview of the field," *Methods of information in medicine,* vol. 40, pp. 346-358, 2001.

[4] B. Ø. Palsson, "Systems Biology: Properties of Reconstructed Networks," vol. 9, 2006.

[5] J. D. Orth, I. Thiele, and B. Ø. Palsson, "What is flux balance analysis?," *Nature biotechnology,* vol. 28, pp. 245-8, 2010.

[6] J. Schellenberger, R. Que, R. M. T. Fleming, I. Thiele, J. D. Orth, A. M. Feist*, et al.*, "Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0.," *Nature protocols,* vol. 6, pp. 1290-307, 2011.

[7] M. A. Oberhardt, B. Ø. Palsson, and J. A. Papin, "Applications of genome-scale metabolic reconstructions.," *Molecular systems biology,* vol. 5, p. 320, 2009.

[8] A. Bordbar and B. O. Palsson, "Using the reconstructed genome-scale human metabolic network to study physiology and pathology.," *Journal of internal medicine,* vol. 271, pp. 131-41, 2012.

[9] G. J. E. Baart and D. E. Martens, "Genome-scale metabolic models: reconstruction and analysis.," *Methods in molecular biology (Clifton, N.J.),* vol. 799, pp. 107-26, 2012.

[10] A. Kümmel, S. Panke, and M. Heinemann, "Putative regulatory sites unraveled by network-embedded thermodynamic analysis of metabolome data.," *Molecular systems biology,* vol. 2, p. 2006.0034, 2006.

[11] Ó. Rolfsson, G. Paglia, M. Magnusdóttir, B. Ø. Palsson, and I. Thiele, "Inferring the metabolism of human orphan metabolites from their metabolic network context affirms human gluconokinase activity.," *The Biochemical journal,* vol. 449, pp. 427-35, 2013.

[12] O. Folger, L. Jerby, C. Frezza, E. Gottlieb, E. Ruppin, and T. Shlomi, "Predicting selective drug targets in cancer through metabolic networks.," *Molecular systems biology,* vol. 7, p. 501, 2011.

[13] S. Sahoo, L. Franzson, J. J. Jonsson, and I. Thiele, "A compendium of inborn errors of metabolism mapped onto the human metabolic network.," *Molecular bioSystems,* vol. 8, pp. 2545-58, 2012.

[14] A. K. Heinken, S. Sahoo, R. M. Fleming, and I. Thiele, "Systems-level characterization of a host-microbe metabolic symbiosis in the mammalian gut.," *Gut microbes,* vol. 4, pp. 28-40, 2013.

[15] H. Yim, R. Haselbeck, W. Niu, C. Pujol-Baxley, A. Burgard, J. Boldt*, et al.*, "Metabolic engineering of Escherichia coli for direct production of 1,4-butanediol.," *Nature chemical biology,* vol. 7, pp. 445-52, 2011.

[16] J. M. Otero, D. Cimini, K. R. Patil, S. G. Poulsen, L. Olsson, and J. Nielsen, "Industrial systems biology of Saccharomyces cerevisiae enables novel succinic acid cell factory.," *PloS one,* vol. 8, p. e54144, 2013.

[17] A. M. Feist and B. O. Palsson, "The growing scope of applications of genome-scale metabolic reconstructions using Escherichia coli," *Nat Biotechnol,* vol. 26, pp. 659-67, Jun 2008.

[18] M. D. Stobbe, S. M. Houten, G. a. Jansen, A. H. C. van Kampen, and P. D. Moerland, "Critical assessment of human metabolic pathway databases: a stepping stone for future integration.," *BMC systems biology,* vol. 5, p. 165, 2011.

[19] N. C. Duarte, S. a. Becker, N. Jamshidi, I. Thiele, M. L. Mo, T. D. Vo*, 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,* vol. 104, pp. 1777-82, 2007.

[20] H. Ma, A. Sorokin, A. Mazein, A. Selkov, E. Selkov, O. Demin*, et al.*, "The Edinburgh human metabolic network reconstruction and its functional analysis.," *Molecular systems biology,* vol. 3, p. 135, 2007.

[21] I. Thiele, N. Swainston, R. M. T. Fleming, A. Hoppe, S. Sahoo, M. K. Aurich*, et al.*, "A community-driven global reconstruction of human metabolism.," *Nature biotechnology,* vol. 31, pp. 419-25, 2013.

[22] A. Mardinoglu, R. Agren, C. Kampf, A. Asplund, M. Uhlen, and J. Nielsen, "Genome-scale metabolic modelling of hepatocytes reveals serine deficiency in patients with non-alcoholic fatty liver disease.," *Nature communications,* vol. 5, p. 3083, 2014.

[23] I. Thiele and B. Ø. Palsson, "A protocol for generating a high-quality genome-scale metabolic reconstruction.," *Nature protocols,* vol. 5, pp. 93-121, 2010.

[24] M. G. Poolman, B. K. Bonde, A. Gevorgyan, H. H. Patel, and D. A. Fell, "Challenges to be faced in the reconstruction of metabolic networks from public databases," *IEE Proceedings - Systems Biology,* vol. 153, p. 379, 2006.

[25] M. Kanehisa, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research,* vol. 28, pp. 27-30, 2000.

[26] H. S. Haraldsdóttir, I. Thiele, and R. M. Fleming, "Comparative evaluation of open source software for mapping between metabolite identifiers in metabolic network reconstructions: application to Recon 2.," *Journal of cheminformatics,* vol. 6, p. 2, 2014.

[27] R. Caspi, H. Foerster, C. a. Fulcher, R. Hopkinson, J. Ingraham, P. Kaipa*, et al.*, "MetaCyc: a multiorganism database of metabolic pathways and enzymes.," *Nucleic acids research,* vol. 34, pp. D511-D516, 2006.

[28] H. Kitano, "Systems biology: a brief overview," *Science,* vol. 295, pp. 1662-1664, 2002.

[29] J. S. Edwards and M. Covert, "Minireview Metabolic modelling of microbes : the flux-balance approach," vol. 4, pp. 133-140, 2002.

[30] E. Almaas, B. Kovacs, T. Vicsek, Z. Oltvai, and A. Barabási, "Global organization of metabolic fluxes in the bacterium Escherichia coli," *Nature,* vol. 270, pp. 839-843, 2004.

[31] I. Thiele, N. D. Price, T. D. Vo, and B. Ø. Palsson, "Candidate metabolic network states in human mitochondria. Impact of diabetes, ischemia, and diet.," *The Journal of biological chemistry,* vol. 280, pp. 11683-95, 2005.

[32] C. Pal, B. Papp, M. J. Lercher, P. Csermely, S. G. Oliver, and L. D. Hurst, "Chance and necessity in the evolution of minimal metabolic networks," *Nature,* vol. 440, pp. 667-70, Mar 30 2006.

[33] M. W. Covert, E. M. Knight, J. L. Reed, M. J. Herrgard, and B. O. Palsson, "Integrating high-throughput and computational data elucidates bacterial networks.," *Nature,* vol. 429, pp. 92-6, 2004.

[34] D. Segre, D. Vitkup, and G. M. Church, "Analysis of optimality in natural and perturbed metabolic networks," *Proc Natl Acad Sci U S A,* vol. 99, pp. 15112-7, Nov 12 2002.

[35] I. Rocha, J. Förster, and J. Nielsen, "Design and application of genome-scale reconstructed metabolic models.," *Methods in molecular biology (Clifton, N.J.),* vol. 416, pp. 409-31, 2008.

[36] F. Santos, J. Boele, and B. Teusink, "A practical guide to genome-scale metabolic models and their analysis.," *Methods in enzymology,* vol. 500, pp. 509-32, 2011.

[37] M. Hucka, a. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano*, et al.*, "The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models," *Bioinformatics,* vol. 19, pp. 524-531, 2003.

[38] T. Shlomi, T. Benyamini, E. Gottlieb, R. Sharan, and E. Ruppin, "Genome-scale metabolic modeling elucidates the role of proliferative adaptation in causing the Warburg effect," *PLoS Comput Biol,* vol. 7, p. e1002018, Mar 2011.

[39] L. Jerby and E. Ruppin, "Predicting drug targets and biomarkers of cancer via genome-scale metabolic modeling," *Clin Cancer Res,* vol. 18, pp. 5572-84, Oct 15 2012.

[40] T. Hao, H.-W. Ma, X.-M. Zhao, and I. Goryanin, "Compartmentalization of the Edinburgh Human Metabolic Network.," *BMC bioinformatics,* vol. 11, p. 393, 2010.

[41] C. Gille, C. Bölling, A. Hoppe, S. Bulik, S. Hoffmann, K. Hübner*, et al.*, "HepatoNet1: a comprehensive metabolic reconstruction of the human hepatocyte for the analysis of liver physiology.," *Molecular systems biology,* vol. 6, p. 411, 2010.

[42] P. Romero, J. Wagg, M. L. Green, D. Kaiser, M. Krummenacker, and P. D. Karp, "Computational prediction of human metabolic pathways from the complete human genome.," *Genome biology,* vol. 6, p. R2, 2005.

[43] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono*, et al.*, "Reactome: a knowledgebase of biological pathways.," *Nucleic acids research,* vol. 33, pp. D428-32, 2005.

[44] E. Fahy, S. Subramaniam, R. C. Murphy, M. Nishijima, C. R. H. Raetz, T. Shimizu*, et al.*, "Update of the LIPID MAPS comprehensive classification system for lipids.," *Journal of lipid research,* vol. 50 Suppl, pp. S9-14, 2009.

[45] R. Agren, S. Bordel, A. Mardinoglu, N. Pornputtapong, I. Nookaew, and J. Nielsen, "Reconstruction of genome-scale active metabolic networks for 69 human cell types and 16 cancer types using INIT," *PLoS Comput Biol,* vol. 8, p. e1002518, 2012.

[46] A. Mardinoglu, R. Agren, C. Kampf, A. Asplund, I. Nookaew, P. Jacobson*, et al.*, "Integration of clinical data with a genome-scale metabolic model of the human adipocyte," *Mol Syst Biol,* vol. 9, p. 649, 2013.

[47] K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught*, et al.*, "ChEBI: a database and ontology for chemical entities of biological interest," *Nucleic Acids Res,* vol. 36, pp. D344-50, Jan 2008.

[48] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh*, et al.*, "KEGG for linking genomes to life and the environment," *Nucleic Acids Res,* vol. 36, pp. D480-4, Jan 2008.

[49] I. Schomburg, "BRENDA, enzyme data and metabolic information," *Nucleic Acids Research,* vol. 30, pp. 47-49, 2002.

[50] E. C. Webb, "Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.," 1992.

[51] R. Caspi, H. Foerster, C. A. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse*, et al.*, "The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases.," *Nucleic acids research,* vol. 36, pp. D623-31, 2008.

[52] C. J. Krieger, C. J. Krieger, P. Zhang, P. Zhang, L. a. Mueller, L. a. Mueller*, et al.*, "MetaCyc: a multiorganism database of metabolic pathways and enzymes," *Nucleic acids research,* vol. 32, pp. 438-442, 2004.

[53] H. Redestig, M. Kusano, A. Fukushima, F. Matsuda, K. Saito, and M. Arita, "Consolidating metabolite identifiers to enable contextual and multi-platform metabolomics data analysis.," *BMC bioinformatics,* vol. 11, p. 214, 2010.

[54] G. Wohlgemuth, P. K. Haldiya, E. Willighagen, T. Kind, and O. Fiehn, "The Chemical Translation Service--a web-based tool to improve standardization of metabolomic reports.," *Bioinformatics (Oxford, England),* vol. 26, pp. 2647-8, 2010.

[55] J. Chambers, M. Davies, A. Gaulton, A. Hersey, S. Velankar, R. Petryszak*, et al.*, "UniChem: a unified chemical structure cross-referencing and identifier tracking system.," *Journal of cheminformatics,* vol. 5, p. 3, 2013.

[56] M. Lang, M. Stelzer, and D. Schomburg, "BKM-react, an integrated biochemical reaction database," *BMC Biochem,* vol. 12, p. 42, 2011.

[57] M. Ganter, T. Bernard, S. Moretti, J. Stelling, and M. Pagni, "MetaNetX.org: a website and repository for accessing, analysing and manipulating metabolic networks," *Bioinformatics,* vol. 29, pp. 815-6, Mar 15 2013.

[58] T. Bernard, A. Bridge, A. Morgat, S. Moretti, I. Xenarios, and M. Pagni, "Reconciliation of metabolites and biochemical reactions for metabolic networks.," *Briefings in bioinformatics,* vol. 15, pp. 123-35, 2014.

[59] A. Kumar, P. F. Suthers, and C. D. Maranas, "MetRxn: a knowledgebase of metabolites and reactions spanning metabolic models and databases.," *BMC bioinformatics,* vol. 13, p. 6, 2012.

[60] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, and D. Wilkins, "A comparison of a graph database and a relational database," *Proceedings of the 48th Annual Southeast Regional Conference on ACM SE 10,* p. 1, 2010.

[61] T. N. Team, "The Neo4j Manual v2.1.7 - Neo Technology - http://neo4j.com/docs/2.1.7/," 2015.

[62] J. Schellenberger, J. O. Park, T. M. Conrad, and B. Ø. Palsson, "BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions.," *BMC bioinformatics,* vol. 11, p. 213, 2010.

[63] I. Rocha, P. Maia, P. Evangelista, P. Vilaca, S. Soares, J. P. Pinto*, et al.*, "OptFlux: an open-source software platform for in silico metabolic engineering," *BMC Syst Biol,* vol. 4, p. 45, 2010.

[64] Oracle, "Java Platform (JDK) 8u65 / 8u66 - http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html#javasejdk."

[65] Oracle, "Java™ Platform, Standard Edition 8 API Specification - http://docs.oracle.com/javase/8/docs/api/."

[66] J. M. Berg, J. L. Tymoczko, and L. Stryer, "Glycolysis Is an Energy-Conversion Pathway in Many Organisms," ed: W H Freeman, 2002.

[67] "KEGG PATHWAY: Glycolysis / Gluconeogenesis - http://www.genome.jp/kegg-bin/show_pathway?map00010."