Universidade do Minho
Escola de Engenharia
Departamento de Informática

João Manuel Capela Araújo Ribeiro

Biochemical Complex data Generation and Integration in Genome-scale Metabolic Models

January 2021

Universidade do Minho
Escola de Engenharia
Departamento de Informática

João Manuel Capela Araújo Ribeiro

Biochemical Complex data Generation and Integration in Genome-scale Metabolic Models

Master dissertation
Master Degree in Bioinformatics

Dissertation supervised by
Oscar Manuel Lima Dias
Filipe Alexandre Wang Liu

January 2021

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Licença concedida aos utilizadores deste trabalho

ACKNOWLEDGMENTS

Em primeiro lugar, gostaria de expressar enorme gratidão aos meus orientadores, Oscar Dias e Filipe Liu, por me terem proposto o tema desta tese que tanto me intrigou e desafiou. Para além disso, queria agrader-lhes por toda a transferência de conhecimento, dicas, propostas, reptos e correções que me providenciaram em abundância ao longo deste ano. Por último, mas não menos importante, obrigado pela amizade, integração e compreensão, num ano que para todos foi muito difícil.

Não posso terminar esta tese sem agradecer ao professor Miguel Rocha, tanto por ter inspirado o meu precurso académico, passando por me ter facilitado a vida como delegado de curso, e por me ter concedido a oportunidade de estar envolvido num grupo de investigação que fervilha de curiosidade científica e sentido crítico.

Às gentes do 16 (por ordem alfabética): Alexandre, Bastos, Davide, Diogo, Emanuel, Fernando, Marta, Miguel e Sequeira. Agradeço-lhes pela disponibilidade em me ajudar, em serem ajudados e, mais importante que tudo isso, serem muito mais que colegas de trabalho, serem grandes amigos. Desde implementações no *merlin*, passando pelo BOIMMG e acabando a jogar gatinhos, JackBox ou SSS, obrigado por estarem sempre lá.

À Carolina deixo um especial agradecimento por ter tornado a minha vida de supervisor demasiado fácil, assim como por ter embelezado algumas figuras que foram usadas nesta tese. Não posso deixar de pedir desculpa pelas diversas ocasiões em que posso ter sido extremamente "reles", não foi por mal. Espero tê-la inspirado para os desafios que tem pela frente.

Ao Nuno, Rui e Tiago deixo um enorme abraço por terem enchido os meus anos em Braga de boas memórias, momentos e alegrias. Sem vocês nunca teria levado para a vida alguns dos "segredos desta cidade".

Ao Varela, um verdadeiro irmão, tão ou mais louco que eu, a pessoa que ajudou a despertar em mim o interesse pela bioinformática, deixo um forte agradecimento por todos os jogos de xadrez, guitarradas, conversas profundas e confidência.

Ao Branco, Rogério, Miguel, Xavier, Joana, Inês, Teresa e Fábio, pessoas que o Porto fez entrar na minha vida. Obrigado pela força e pelos "verdes anos" que me proporcionaram na muy nobre FCUP.

À Sofia tenho de deixar o meu maior agradecimento, pois sem ela nada disto seria possível. Um verdadeiro exemplo de que a perseverança, rigor científico e convicção recompensam. Deixo aqui a gratidão por poder ter conhecido e ter sido tão cúmplice de alguém tão único e espetacular. Agradeço também por me mostrares que apesar das vicissitudes da vida, estarás sempre disponível para me ajudar. Pelo amor, carinho, clarividência, inspiração e apoio tanto pessoal como científico, um obrigado de corpo inteiro.

Aos meus amigos de infância cuja amizade perdurou até aos dias de hoje, Teixeira, Bárbara, Anas, Zé e Ricardo, obrigado por me mostrarem que a amizade pode durar o tempo de uma vida.

Aos meus pais e irmã, um grande obrigado por me ajudarem e apoiarem incondicionalmente, apesar do meu ocasional mau humor e desarrumação. Nunca irei, no tempo de uma vida, conseguir ter a oportunidade de vos retribuir totalmente o carinho, amor e compreensão que me deram e tanto contribuiram para o meu sucesso académico e realização pessoal.

Ao resto da minha família que me viu crescer e que me vêm terminar mais uma etapa importante da minha vida, obrigado por contribuirem para o meu sucesso e felicidade.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

The (re-)construction of Genome-Scale Metabolic (GSM) models is highly dependent on biochemical databases. In fact, the biochemical data within these databases is limited, lacking, most of the times, in structurally defined compounds' representations. In order to circumvent this limitation, compounds are frequently represented by their generic version. Lipids are paradigmatic cases: given that a multitude of lipid species can occur in nature, not only is their storage in databases hampered, but also their integration into GSM models. Accordingly, converting one lipid version, in GSM models, into another can be tricky, as these compounds possess side chains that are likely to be transferred all across their biosynthetic network. Hence, converting a lipid implies that all its precursors have to be converted as well, requiring information on lipid specificity and biosynthetic context.

The present work represents a strategy to tackle this issue. Biochemical cOmplex data Integration in Metabolic Models at Genome scale (BOIMMG)'s pipeline encompasses the integration and processing of biochemical data from different sources, aiming at expanding the current knowledge in lipid biosynthesis, and its integration in GSM models.

Generic reactions retrieved from MetaCyc were handled and transformed into reactions with structurally defined lipid species. More than 30 generic reactions were fully (and 27 partially) characterized, allowing to predict over 30000 new lipid structures and their biosynthetic context.

The integration of BOIMMG's data into GSM models was conducted for electron-transfer quinones, glycerolipids, and phospholipids metabolism. The validation accounted on the comparison of models with different versions of these metabolites. BOIMMG's conversion modules were applied to *Escherichia coli*'s iJR904 model [1], generating 53 more matching lipids and 38 more matching reactions with iJR904 model's iteration iAF1260b [2, 3], in which the conversion was performed and curated manually.

To the best of our knowledge, BOIMMG's database is the only with biosynthetic information regarding structurally defined lipids. Moreover, there is no other state-of-the-art tool capable of automatically generating complex lipid-specific networks.

Keywords: Genome-Scale Metabolic models, Biochemistry, Lipid metabolism, Biosynthesis, Bioinformatics, Chemoinformatics

RESUMO

A reconstrução de modelos metabólicos à escala genómica (GSM na língua inglesa) depende grandemente da informação bioquímica presente em bases de dados. De facto, esta informação é muitas vezes limitada, podendo não conter representações de compostos estruturalmente definidos. Como tentativa de contornar esta limitação, os compostos químicos são frequentemente representados pela sua representação genérica. Os lípidos são casos paradigmáticos, dado que uma multitude de diferentes espécies químicas de lípidos ocorrem na natureza, dificultando o seu armazenamento em bases de dados, assim como a sua integração em modelos GSM. Desta forma, o processo de converter lípidos de uma versão genérica para uma versão estruturalmente definida não é trivial, dado que estes compostos possuem cadeias laterais que são transferidas ao longo das suas vias de biossíntese. Consequentemente, essa conversão implica que todos os precursores desses lípidos também sejam convertidos, requerendo haver informação relativa a lípidos específicos e às suas relações biossintéticas.

O presente trabalho representa uma estratégia para resolver esse problema. A *pipeline* do software desenvolvido no âmbito deste trabalho, *Biochemical cOmplex dataIntegration in Metabolic Models at Genome scale* (BOIMMG), engloba a integração e processamento de dados bioquímicos de diferentes fontes, visando a expansão do conhecimento atual na biossíntese de lípidos, assim como a sua integração em modelos GSM.

Relativamente à segunda fase, reações genéricas extraídas da base de dados MetaCyc foram processadas e transformadas em reações com lípidos estruturalmente definidos. Mais de 30 reações genéricas foram completamente (e 27 parcialmente) caracterizadas, permitindo prever mais de 30000 novas estruturas de lípidos, assim como os seus contextos biossintéticos.

A integração dos dados nos modelos GSM foi conduzido para o metabolismo das quinonas transportadoras de eletrões, glicerolípidos e fosfolípidos. A validação teve em conta a comparação entre modelos com diferentes versões destes metabolitos. Os módulos de conversão do BOIMMG foram aplicados ao modelo iJR904 de *Escherichia coli* [1], gerando mais 53 lípidos e 38 reações que se encontram no modelo iAF1260b [2, 3], uma iteração do modelo iJR904 cuja conversão de lípidos se procedeu manualmente.

A base de dados gerada pelo método BOIMMG é a única que contém informação biossintética relata a lípidos estruturalmente definidos. Adicionalmente, BOIMMG é uma ferramenta única que permite gerar redes complexas de lípidos automaticamente.

Palavras-Chave: Modelos Metabólicos à Escala Genómica, Bioquímica, *Gap-filling*, Ontologias

CONTENTS

1 INTRODUCTION 1

1.1 Context and Motivation 1

1.2 Objectives 2

1.3 Document organization 3

2 STATE OF THE ART 5

2.1 Computational Systems Biology 5

2.1.1 Emergence of Systems Biology 5

2.1.2 Systems Biology 5

2.2 Metabolic Modelling 6

2.2.1 Genome-Scale Metabolic modelling 6

2.2.2 Reconstruction tools 8

2.2.3 Simulation and phenotype prediction methods 8

2.2.4 *COBRApy* 10

2.2.5 Applications of GSM models 11

2.3 Computational Representation of Compounds 12

2.3.1 Background 12

2.3.2 Connection tables 13

2.3.3 Line notations 14

2.3.4 Fragment codes 17

2.3.5 Generic structures 17

2.3.6 *rdkit* 18

2.4 Biochemical databases and online resources 19

2.5 Lipids 20

2.5.1 Glycerolipids 21

2.5.2 Glycerophopholipids 21

2.5.3 Sphingolipids 22

2.5.4 Prenol Lipids 24

2.6 The compound representation problem 26

2.6.1 Compounds representation in GSM models 26

2.6.2 Useful considerations for the biochemical representation problem 29

2.6.3 State-of-the-art tools 29

2.7 Graph databases and Neo4j 32

3 BOIMMG FRAMEWORK 34

3.1 Formal Notation 35

3.1.1 Data structures and integration definitions 37

3.1.2 Biochemical definitions and operations 38

3.2 Databases Integration 41
3.2.1 Data details and extraction 42
3.2.2 Data transformation and loading 48
3.2.3 Data integration 50
3.2.4 Data integration in SLM's hierarchy 51
3.3 Semi-automated knowledge expansion 52
3.3.1 Electron-transfer quinones 53
3.3.2 Other Lipids 53
3.4 Integration in GSM models 56
3.4.1 Software overview 57
3.4.2 Model Mapper 59
3.4.3 Network Modifiers 59
3.4.4 Revisor 61
3.5 BOIMMG's evaluation and validation 64
3.5.1 Knowledge expansion evaluation 64
3.5.2 BOIMMG's data integration in GSM models 66
3.6 Web-service implementation 67
3.6.1 Navigation module implementation 68
3.6.2 Submissions module 69
4 RESULTS AND DISCUSSION 71
4.1 Databases integration assessment 71
4.1.1 Generic compounds integration 71
4.1.2 Structurally defined compounds integration 72
4.1.3 Lipids integration in SLM's hierarchy 74
4.2 Final database topology and general statistics 75
4.2.1 General statistics 76
4.3 Knowledge Expansion 77
4.3.1 Capacity of generating new reactions 78
4.4 BOIMMG's data integration in GSM models 79
4.4.1 Simple Representation Case 79
4.4.2 Redundant Representation Case 82
4.5 Web-service 88
4.5.1 Navigation module 88
4.5.2 Submissions module 89
5 CONCLUSION 93
5.1 Conclusion remarks 93
5.2 Future perspectives 94
5.3 Scientific outcomes 94
Bibliography 95
6 SUPPLEMENTARY MATERIAL 102

6.1 Added lipids and their correspondence to the iAF1260b model 104

ACRONYMS

A

ACP Acyl Carrier Protein.

B

BiGG Biochemical, Genetic, and Genomic.

BOIMMG Biochemical cOmplex data Integration in Metabolic Models at Genome scale.

BRENDA BRaunschweig Enzyme Database.

C

CASE Computer-Assisted Structure Elucidation.

ChEBI Chemical Entities of Biological Interest.

COBRA COnstraints-Based Reconstruction and Analysis.

CSS Cascading Style Sheet.

CT Connection Tables.

D

DNA Deoxyribose Nucleic Acid.

E

EC Enzyme Comission.

F

FAME fatty acid methyl ester.

FBA Flux Balance Analysis.

FVA Flux Variability Analysis.

G

GENRE Genome-Scale Metabolic Reconstruction.

GL Glycerolipids.

GP Glycerophospholipids.

GPR gene-protein-reaction.

GSM Genome-Scale Metabolic.

H

HTML HyperText Markup Language.

HTTP HyperText Transfer Protocol.

I

ID identifier.

InChI International Chemical Identifier.

IUPAC International Union of Pure and Applied Chemistry.

K

KEGG Kyoto Encyclopedia of Genes and Genomes.

L

LMSD LIPID MAPS Structure Database.

LP Linear Programming.

M

MDL Molecular Design Limited.

ME Metabolic Engineering.

MILP Mixed-Integer Linear Programming.

MM Model Mapper.

MOMA Minimization of Metabolic Adjustment.

MS Metabolite Swapper.

N

NGS Next Generation Sequencing.

NH Network Handlers.

NM Network Modifier.

NMR Nuclear Magnetic Resonance.

P

PGDB Pathway/Genome Database.

PL Prenol Lipids.

Q

QP Quadratic Programming.

R

RG Relationships Generator.

ROOM Regulatory On/Off Minimization of metabolic fluxes.

RPS Representation Problem Solver.

RRC Redundant Representation Case.

RS Reactions Swapper.

S

SBML Systems Biology Markup Language.

SLIMEr Split Lipids Into Measurable Entities.

SLM SwissLipids.

SLN Sybyl Line Notation.

SMARTS SMILES arbitrary target specification.

SMILES Simplified Molecular Input Line Entry System.

SP Sphingolipids.

SRC Simple Representation Case.

T

TCDB Transporter Classification Database.

TSV tab-separated values.

W

WLN Wiswesser LineFormula Notation.

LIST OF FIGURES

Figure 1 Reconstruction of GSM models workflow 8
Figure 2 Applications of GSM models. Adapted from [6] 11
Figure 3 Benzene structure arbitrarily labeled 13
Figure 4 A Connection Tables (CT) of benzene used in CTfile format. 14
Figure 5 The International Chemical Identifier (InChI) of β-D-glucose. 16
Figure 6 Example of a fragment encoding of phenylalanine. 17
Figure 7 Example of a generic structures of ubiquinones and phosphatidylglycerols 18
Figure 8 1-O-hexadecyl-2-(9Z-octadecenoyl)-sn-glycerol structure 21
Figure 9 1-heptadecanoyl-2-(9Z-tetradecenoyl)-sn-glycero-3-phospho-(1'-myo-inositol) structure 22
Figure 10 N-(octadecanoyl)-sphing-4-enine-1-phosphocholine structure 23
Figure 11 Isoprene unit. 25
Figure 12 "ISA" reactions defined in the Yeast v6.0 model 28
Figure 13 "Precursor_of" relationships 30
Figure 14 "Is_a" relationships 30
Figure 15 Succinate dehydrogenase reaction. 31
Figure 16 Biochemical cOmplex data Integration in Metabolic Models at Genome scale (BOIMMG) pipeline 34
Figure 17 General workflow of the databases integration 42
Figure 18 Graph transformation 50
Figure 19 Establishment of the electron-transfer quinones relationships 54
Figure 20 Knowledge expansion stage 55
Figure 21 Handling reactions' SMARTS transformations 56
Figure 22 Knowledge expansion algorithm applied to cardiolipin 57
Figure 23 Software architecture 58
Figure 24 Types of swaps 61
Figure 25 BOIMMG's web-service general architecture. 68
Figure 26 Navigation module architecture 69
Figure 27 Submissions module's *modus operandi*. 70
Figure 28 Generic compounds that have been integrated 72
Figure 29 Structurally defined compounds that have been integrated 73
Figure 30 Percentage of structurally defined lipids with overlap in all databases 73
Figure 31 Percentage of structurally defined lipids with overlap in SwissLipids (SLM) and LIPID MAPS Structure Database (LMSD) 74
Figure 32 Database's final topology 75

Figure 33 Venn diagram with the integrated and listed lipids as well as the ones not listed (Unique BOIMMG's lipids) 77

Figure 34 Wrongly generated reactions and unestablished relationships percentage 78

Figure 35 Intersection between *E. coli* and *S.cerevisiae* model's metabolite set 81

Figure 36 The intersection between *E. coli* and *S.cerevisiae* model's reaction set 82

Figure 37 The intersection between iAF1260b and iJR904 model's metabolite set 87

Figure 38 The intersection between iAF1260b and iJR904 model's reaction set 88

Figure 39 Home page from BOIMMG's web-service. 89

Figure 40 Lipids hierarchy in BOIMMG's web-service. 89

Figure 41 Lipid web page 90

Figure 42 Submissions' module menu. 90

Figure 43 SRC mode parameters and model submission page. 91

Figure 44 RRC mode parameters and model submission page. 91

Figure 45 Status page rendering information related to the state of the submission. 92

Figure 46 BioISO flux analysis for the complete cardiolipins. The checkmark associated to each cardiolipin indicates that it is carrying flux. 102

Figure 47 BioISO flux analysis for the complete phosphatidylethanolamines. The checkmark associated to each phosphatidylethanolamines indicates that it is carrying flux. 102

Figure 48 BioISO flux analysis for the complete phosphatidylglycerol. The checkmark associated to each phosphatidylglycerols indicates that it is carrying flux.103

Figure 49 BioISO flux analysis for the complete phosphatidylserine. The checkmark associated to each phosphatidylserine indicates that it is carrying flux. 103

LIST OF TABLES

Table 1 Computational tools that ease the reconstruction of GSM models 9
Table 2 CT of benzene 13
Table 3 The different line notations of benzene ring. 15
Table 4 The different biochemical databases. 19
Table 5 Glycerolipids (GL) computational representation. 22
Table 6 Glycerophospholipids (GP) computational representation 23
Table 7 The different types of sphingolipids, their backbone, and their computational representation. 24
Table 8 Electron-transfer quinones 25
Table 9 Electron-transfer quinones' head group representation 26
Table 10 The eletron-transfer quinones complete representation in several databases 27
Table 11 The notation that will be further used for writing algorithms and definitions 35
Table 12 List of symbols to represent sets 35
Table 13 Information contained in the tab-separated values (TSV) file retrieved from SLM database. 43
Table 14 Model SEED files and where they can be downloaded. 44
Table 15 Information contained in the tab delimited text file retrieved from ModelSEED database 45
Table 16 Information contained in the tab delimited text file retrieved from ModelSEED database 46
Table 17 Information contained in the tab delimited text file retrieved from ModelSEED database 46
Table 18 Information contained in the TSV file retrieved from LIPID MAPS database. 47
Table 19 Results of LMSD and Model SEED compounds integration in SLM's hierarchy 74
Table 20 Integration of LMSD and Model SEED compounds in SLM hierarchy 74
Table 21 Number of integrated subclasses per class of lipid 75
Table 22 Number of relationships per type in BOIMMG's database 77
Table 23 Quinone chemical species before and after BOIMMG's network modification 80
Table 24 Metabolite set match between iMM904 model and the previously modified iML1414 82
Table 25 Quinone chemical species before and after BOIMMG's network modification 83
Table 26 Lipids present in the biomass equation of iJR904's metabolic model 84

Table 27 Continuation. Lipids present in the biomass equation of iJR904's metabolic model 85
Table 28 Added reactions to gap-fill the granulated model. 86
Table 29 Number of new reactions in iJR904 model per type 88
Table 30 Added lipids and their correspondence to the iAF1260b model 104
Table 31 Added lipids and their correspondence to the iAF1260b model - rest of the lipids 105

1

INTRODUCTION

1.1 Context and Motivation

The study of biological systems as a whole by accounting the interactions between all the parts is dubbed as Systems Biology. In fact, those entire systems represent a paraphernalia of data that needs to be summarized and contextualized. Correspondingly, the reconstruction of biomolecular networks as mathematical models is oftentimes employed as means to compile genomic, biochemical and physiological knowledge [4].

The inexorable emergence of high-throughput sequencing techniques allowed the generation of the so-called *omics* data, leading to an increasing number of sequenced genomes [5] and available biochemical data. Therefore, the development of metabolic models at genome-scale has been facilitated, over the years. As a consequence, GSM models have eventually arisen as tools used for a deeper and faster comprehension of living organisms' metabolism. These computational tools have been guiding metabolic engineering towards the improvement of cell factories, attempting to drive more flux into the production of value added compounds [6, 7].

As useful as these models can be, their reconstruction is limited by the biochemical set present in conventionally used databases. Complex macromolecules's metabolism is simplistically represented by their generic version, as biochemical data sources lack in specificity and structurally defined compounds. Accordingly, lipid representation is a paradigmatic case. Considering that each subclass is composed by a multitude of combinations of fatty acids, long chain alcohols or even repeating isoprene units, their representation in GSM models is often generic. In turn, it is expected that these models do not account on the individual lipid distribution for each subclass [7]. Moreover, they will not be able to capture the complex topology of lipid biosynthesis network, making it harder to represent and flexibly manipulate a given model's lipid metabolism.

Therefore, it becomes imperative to create a resource that could automatically revise lipid metabolism in GSM models, accelerating their reconstruction and, at the same time, ensure lipid specificity. Correspondingly, being able to rapidly predict the lipid and fatty acid distribution in genetically and environmentally perturbed biological systems would be relevant.

The development of such tool requires the integration of lipid specific information and generation of valuable biochemical knowledge capable of capturing the biosynthetic relationships between lipids. Although other online resources such as LIPID MAPS [8] and SwissLipids (SLM) [9] represent valuable and useful contributions to the field, they lack in biosynthetic information. Moreover, to the best of our knowledge, there is no automatic computational tool that integrates specific lipid information in GSM models without using experimental data.

1.2 Objectives

The main goal of the present work was to develop a chemo- and bioinformatics tool able to generate relevant biochemical information and to ensure its integration in GSM models. Such tool must be capable of capturing relevant biosynthetic relationships between lipids, by generating annotated functional and structural relationships in a graph database. Then, lipid representation in GSM models is to be revised automatically, using the previously generated information. Accordingly, the representation of electron-transfer quinones, phospholipids and glycerolipids will be revised in *Escherichia coli*, *Saccharomyces cerevisiae* model, and validated against another model of *E. coli* with lipid specificity.

Given this aim, the following objectives were specified

- Generate structural hierarchies and biosynthetic relationships for electron-transfer quinones (semi-automatically);
- Integrate and generate structural hierarchies and biosynthetic relationships for several types of lipids (automatically);
- Develop new bioinformatics tools capable of revising the lipids representation (automatically);
- Test the developed tools in different GSM models, in order to validate the present work;
- Implement a user-friendly web-service both to navigate in the database and revise the compounds representation in GSM models.

Firstly, the structural hierarchies and biosynthetic relationships will be generated and integrated using the chemoinformatics tool *rdkit* (https://www.rdkit.org/), consulting the available information regarding lipid species. Moreover, information present in SLM [9], LMSD [10] and ModelSEED [11] is aimed to be extracted and integrated. Furthermore, this information will be stored and handled in a graph database using *Neo4j* database management system (https://neo4j.com/). Moreover, a bioinformatics tool is aimed to be developed on top of *COBRApy* [12] to revise the lipid metabolism in GSM models. For this task, information retrieved from the previously implemented database will be used.

Finally, the web-service will be developed using *Django* (https://www.djangoproject.com/) and *Flask* (https://flask.palletsprojects.com/) frameworks. This web project and the aforementioned tool are to be compiled in *Docker* (https://www.docker.com/) containers along with their dependencies and requirements.

1.3 Document organization

Chapter 2 - State of the art

This chapter aims at contextualizing the present work by describing and combining concepts in GSM modelling, chemoinformatics, and lipid's computational representation.

- Section 2.1 contextualizes the field of study of the present work.
- Section 2.2 describes fundamental concepts in GSM modelling, main reconstruction and simulation tools, as well as the applications of GSM models.
- Section 2.3 enumerates the main types of chemical representations, ending with the description of the chemoinformatics tool *rdkit*.
- Section 2.4 briefly describes and compares the state-of-the-art databases and online resources regarding relevant biochemical information. Herein, the computational representations of compounds available in each database are enumerated.
- Section 2.5 relates basic information about the main lipid classes with their computational representation.
- Section 2.6. enumerates fundamental considerations about the representation of compounds in GSM models. Furthermore, it explains and analyses the advantages and disadvantages of state-of-the-art tools regarding this issue.
- Section 2.7. justifies the usage of a graph database in the present work, as well as briefly describes *Neo4j* database management system.

Chapter 3 - BOIMMG framework

This chapter's main goal is the thorough description of all the processes, algorithms, and definitions.

- Section 3.1 starts by presenting formal notations and biochemical definitions to be utilized in the following sections.
- Section 3.2 describes processes regarding the data integration.
- Section 3.3 describes methods and algorithms regarding knowledge expansion.
- Section 3.4 describes software architecture and integrated methods regarding BOIMMG's data integration in GSM models.
- Section 3.5 enumerates all the definitions and methods of evaluation and validation of both knowledge expansion, and BOIMMG's data integration in GSM models.

- Section 3.6 describes the web-service implementation.

Chapter 4 - Results and discussion

The results of each stage of BOIMMG's pipeline are shown, described and discussed in chapter 4.

- Section 4.1 delivers results on the extraction and integration of lipid data from several sources.
- Section 4.2, general statistics are enumerated, described, and analysed.
- Section 4.3 shows and describes the knowledge expansion stage results regarding the relationships' aptitude to generate new reactions.
- Section 4.4 thoroughly presents and analyses the results concerning BOIMMG's data integration in GSM models.
- Section 4.5 contains illustrations on how to use the web-service.

Chapter 5 - Conclusion

The last chapter summarizes the present work, enumerates future perspectives and includes this work's scientific outcomes.

2

STATE OF THE ART

2.1 Computational Systems Biology

2.1.1 Emergence of Systems Biology

During the 20^{th} century, molecular biology relied upon reductionist approaches [13], based on the idea that the whole could be explained by the accurate understanding of the separate parts. The prevailing reductionist mindset led to one of the most remarkable scientific breakthroughs of human history: the Watson and Crick's discovery of Deoxyribose Nucleic Acid (DNA) structure [14]. More than 20 years later, DNA sequencing methods appeared such as Maxam–Gilbert's and Sanger's [15, 16], being dubbed as the first generation of sequencing techniques [17]. At the turn of the century, the first genomes were fully sequenced, including human's [18]. Later on, high-throughput sequencing techniques started to emerge. As this novel technologies were quite more fast and automated than its predecessors [17], it did not take long until big amounts of the so-called *omics* data were generated.

These high-throughput sequencing technologies, also known as Next Generation Sequencing (NGS), drove biology into a paradigm shift. Although the prevailing approaches allowed big steps towards the understanding of biological components, it turned out to be insufficient when it came to comprehend systems as a whole. Biological systems are complex and its components are multifuncional, diverse, and their interaction is usually selective and nonlinear. Accordingly, evolution managed to create a whole set of interactions between system's components. From cells to ecological webs, the interpretation of those complex entities should adopt integrative and system-level approaches. [19]

2.1.2 Systems Biology

Systems Biology is an emergent field that aims at the better understanding of biological systems in an integrative manner, rather than only entirely focused on the system's components separately. To achieve such goal, several efforts in different scientific fields have had to converge. Molecular biology, computer science, genetics and so forth are examples of fields whose contributions have helped to converge into a better comprehension of biological systems.

According to Kitano [20], there are four domains of learning when it comes to accomplish this goal.

- **System's Structure**: the understanding of structural relationships between components as well as the interpretation of the quantitative data related to those. This information can range from regulatory relationships between genes to the organism physical structure.
- **System's Dynamics or System's Behaviour**: the comprehension of systems' behaviour over time, given a set of conditions.
- **Control Methods**: the deeper understanding of mechanisms that manage the system's state. For instance, mechanisms underlying the cell cycle.
- **Design Method**: the development of new ways to design and change biological systems aiming at its properties improvement.

A common practice in biology is the adoption of hierarchical thinking for biomolecules. For instance, amino acids are the building blocks for larger molecules like peptides, then, together, they form polipeptide chains, secondary structures and so forth. Correspondingly, genome-scale networks must not deviate from this practice [4]. Reactions must be the irreducible components, which, along with others, combine into modules and form metabolic pathways [4].

The ultimate goal of Systems Biology is the full understanding of biological entities. Such objective would be accomplished with a combination of both computational and experimental approaches [19].

2.2 Metabolic Modelling

2.2.1 Genome-Scale Metabolic modelling

Although the first GSM model was reconstructed in 1999 [21], the dawn of metabolic modelling had occurred years before. Yet, these models were biochemically limited and did not resort to genome-wide information. The advent of several online databases and bioinformatics resources made it easier to access NGS outputs and analyse them, as well. Consequently, along with the increasing number of sequenced genomes, GSM modelling also picked up the pace (http://darwin.di.uminho.pt/models, http://bigg.ucsd.edu/models) for both eukaryotic and prokaryotic organisms.

GSM models compile Biochemical, Genetic, and Genomic (BiGG) [22] knowledge. They have been relevant in the better and faster comprehension of metabolic networks, over the last 20 years. Models such as these must rely on several principles [4]:

- Equations must be formulated for all chemical reactions taking place in the cell.
- Experimental data and annotated genome sequences must be combined.

- Cell's functions are subjected to constraints (physico-chemical, topological, environmental and regulatory);
- Cells have different behaviours in different environments.
- Mass conservation law must be applied. Chemical equations can be set in a stoichiometric matrix (**S**) and fluxes of each reaction can be put in a vector (**v**). So, applying the mass conservation law, the linear equation $S \cdot v = 0$ can define the steady states of a given system.
- Cells are subjected to selective pressure. Objective functions can be defined in order to find optimal states under a given environment and genetic perturbations.

Concerning this principles, a protocol was developed by Thiele and Palsson in order to produce high-quality GSM models [23].

The first two stages focus on the generation of a Genome-Scale Metabolic Reconstruction (GENRE), whereas the last two stages are focused on the conversion of the GENRE into a mathematical model, its evaluation, and refinement.

The first main stage is the creation of the draft reconstruction. Such process includes the search for genes related to different metabolic functions (metabolic genes). Accordingly, it can be operated by identifying biochemical reactions in databases, literature and in the annotated genome (see Figure 1). [23–27]

The following stage is the manual reconstruction refinement [23, 28], where one must check whether there is missing and incorrect information in the reconstructed network [24]. Accordingly, an evaluation is performed and several information is collected. Then, the refinement is conducted as well as the reconstruction assembly, adopting a pathway-by-pathway approach [23]. Such *modus operandi* could be helpful to identify missing gene annotations and additional reactions as well. Useful information is nonetheless gleaned: substrate and co-factor usage, reactions' stoichiometry, directionality, compartmentalization as well as energy and growth requirements [23, 27]. Also, gene-protein-reaction (GPR) associations and the biomass composition are determined [27].

The next step encompasses the conversion from a reconstructed network into a mathematical model [23, 28]. At this stage, a stoichiometric matrix **S** is defined, where the columns correspond to the reactions and the rows correspond to the metabolites. Once the **S** matrix is defined, the systems boundaries have to be determined [23]. Then, an objective function is set and constraints ought to be added to the model by accounting on specific media conditions [28]. The outcome from this stage is a condition-specific and computable model that is, desirably, saved in Systems Biology Markup Language (SBML) format [29].

Finally, the fourth stage is the *in silico* network verification, evaluation and validation [23, 28]. At this stage, gap-filling is performed, the stoichiometrically balanced cycles are identified and, desirably, eliminated [23]. Furthermore, the production of biomass precursors is tested, reactions not carrying flux are identified and other simulations for phenotypic predictions take place. Experimental data is then used to validate the results [25, 28].

The reconstruction of GSM models is an iterative process. Thus, arriving at the fourth stage does not mean that the process is over. Correspondingly, the process continues on and on, until a desirable model is achieved (regarding the expert purpose and scope). [23, 28]

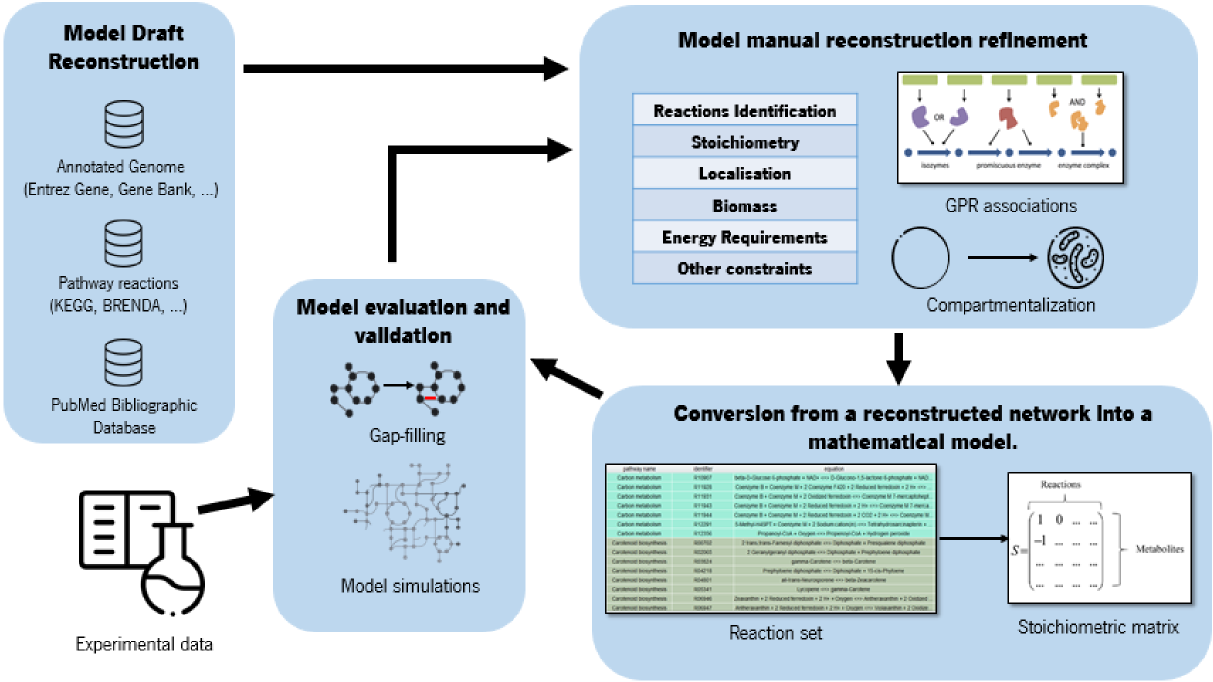


Figure 1: Reconstructed process of GSM models. Adapted from [25]. The GPR associations scheme was taken from [30] and reaction set taken from merlin 4.0

2.2.2 Reconstruction tools

The reconstruction of GSM models involves strenuous efforts and time-consuming processes [28]. Hence, in the last 15 years, several computational tools were developed in order to assist such a laborious task (see Table 1).

The computational tools enumerated in Table 1 assist and automate several steps on the reconstruction of GSM models. All these tools are able to generate a draft reconstruction, however, FAME (2012) is not able to assist the model reconstruction for microorganisms which are not in KEGG [39]. In fact, CarveMe (2018), Model SEED version 2.2 and Pathway Tools version 22.0 have great performance in the draft reconstruction [28]. As for the manual refinement stage, *merlin* version 4.0 provides a suitable interface to perform it [25].

2.2.3 Simulation and phenotype prediction methods

At the end of the third phase of the reconstruction process, one would have a well defined mathematical model. This model is represented by a stoichiometric matrix in which the constraints (substrate availability, uptake rate, secretion rate, etc) of the network are defined.

Table 1: Computational tools that ease the reconstruction of GSM models

Tool	Type	Associated databases	Reactions inferred from	Reference
AuReMe	Command-line	BiGG, MetaCyc	Template models	[31]
CarveMe	Command-line	BiGG	Template model	[32]
AutoKEGGRec	Command-line	KEGG	Database	[33]
CoReCo	Command-line	KEGG	Database	[26]
RAVEN	Command-line	KEGG, MetaCyc	Database and template models	[34]
merlin	Standalone	KEGG MetaCyc, UniProtKB, TCDB	Database	[25]
Pathway Tools	Standalone	MetaCyc and PGDB	Database	[35]
MetaDraft	Standalone	BiGG	Template model	[36]
MEMOSys	Web service	KEGG, Uniprot, ChEBI	Template models	[37]
Model SEED	Web service	Model SEED	Template model	[38]
FAME	Web service	KEGG	Database	[39]
KBase	Web service	KEGG BiGG and MetaCyc	Database	[40]

Moreover, a steady state is considered, so that the amount of a given compound being produced is equal to the amount consumed. Ultimately, the constraints and compound balances impose a specific space of flux distributions. In other words, an interval of consumption and production rates is defined for every metabolite taking place in every reaction of the network. [41]

The space of flux distributions represents itself potential physiological states [42]. These states can be simulated and predicted as well. For that, the following *in silico* approaches can be employed.

- Flux Balance Analysis (FBA) - This approach is underlain by the mass balances and steady-state growth assumption. Additionally, a linear objective function is defined. Generically, FBA computes the way how the fluxes must be balanced towards the achievement of an optimal homeostatic state [42], using a Linear Programming (LP) approach on the optimization process.

- pFBA - This is a parsimonious version of FBA where two LP optimizations are performed. An optimization to maximize (or minimize) the objective function and another to minimize the sum of the fluxes [43].

- Flux Variability Analysis (FVA) - Rather than identifying all the optimal solutions, this approach aims at the estimation of the flux variability within a specific solution. This is an LP-based approach that operates from alternate optimal solutions. Using one of those solutions, each reaction is subsequently minimized and maximized in order to estimate the range of flux variability [44].

- Minimization of Metabolic Adjustment (MOMA) - This approach aims at the determination of the flux distribution of mutants, using Quadratic Programming (QP). Considering gene deletion constraints, MOMA minimizes the distance between the wild-type's point and the mutant's in the flux space. Although suboptimal for the mutant, the solution generated by MOMA is the closest to the wild-type optimal state [45].
- Regulatory On/Off Minimization of metabolic fluxes (ROOM) - By using Mixed-Integer Linear Programming (MILP), this approach aims at predicting the metabolic steady-state after gene knockouts. ROOM minimizes the number of flux changes and tries to maintain flux linearity. Furthermore, as for gene knockouts, ROOM tries to redirect metabolic flux to short alternative pathways in order to soften the gene knockout effect. [46]

Indeed, the afore mentioned approaches reveal to be very useful for the fourth stage of the GSM model reconstruction. The applications of such techniques encompasses knockout simulations, genotype-phenotype predictions in different growth environments, iterative model improvement and discovery of regulatory interactions. Also, gap-filling can be performed by using some of these methods, allowing one to discover "dead-end" metabolites, new metabolic reactions and functions. [42]

2.2.4 *COBRApy*

COnstraints-Based Reconstruction and Analysis (COBRA) for python, also known as *COBRApy* [12] is a python package that includes the great majority of COBRA methods [47]. It relies on the object oriented programming paradigm to represent the main components of complex metabolic networks such as metabolites, reactions, and genes.

It includes methods that allow reading, manipulating and exporting an altered model. The compatible formats for GSM model files are tab-separated values (TSV), SBML, and MatLab format. Model manipulation includes adding, modifying or removing metabolites, reactions, genes and so forth. This is particularly useful for those interested in developing software packages that modify GSM models.

The metabolite object includes relevant properties for model mapping and integration. These can be the name of the metabolite, its formula, charge, or annotations. The metabolite's annotations are substantially useful when it comes to mapping the model and revising the compounds' representation. In turn, this property includes cross-references, and information related to the compound chemical structure.

Finally, *COBRApy* gathers a group of methods to simulate, analyse and even gap-fill the metabolic network.

2.2.5 Applications of GSM models

At first, GSM models were built aiming at the deeper and faster comprehension of an organism metabolism concerning its genetics. However, other applications have arisen, eventually. Their applications encompass the guidance of Metabolic Engineering (ME), the contextualization of *omics* data, network property discovery and multi-species relationships [6].

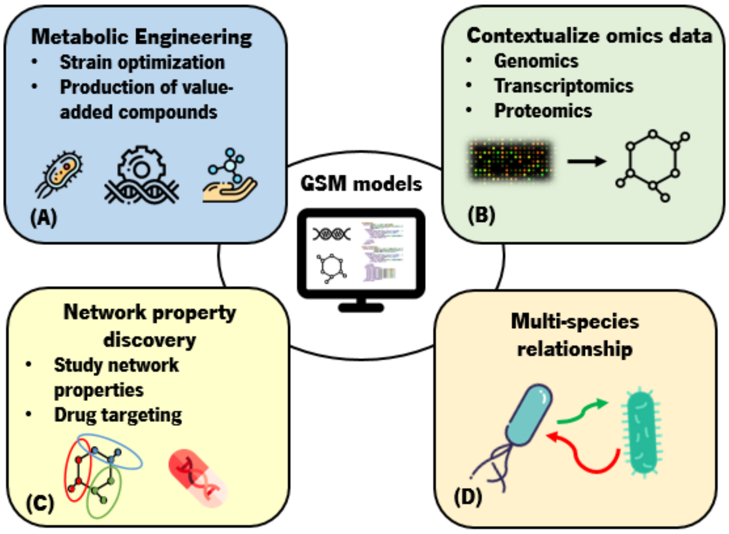


Figure 2: Applications of GSM models. Adapted from [6]

Genome-scale approaches can guide ME towards the production of desirable compounds and strain optimization (Figure 2A). Small scale approaches were traditionally employed by ME to achieve such aims [6]. Yet, metabolic networks are complex and so is the interaction between their components. Metabolic reactions are controlled by several mechanisms and layers of regulation, making the prediction of certain metabolic outcomes difficult to perform. Hence, a systems-wide tool like GSM models can guide ME into the identification of targets for genetic and environmental alterations [48]. Besides, it can save time by minimizing the number of experimental attempts to achieve optimal phenotypes and outputs [48].

High-throughput techniques generate high amounts of *omics* data. GSM models can be useful to organize, and to put these data into the context of the whole system (Figure 2B). Moreover, this data can be converted into constraints, since they represent the organism's physiological states under certain environmental circumstances [6, 48]. Correspondingly, the range of solutions in the flux distribution can be narrowed [48]. In a way, GSM models can be completed by *omics* data. Simultaneously, these data, which are oftentimes noisy, incomplete and complex, can be contextualized by GSM models.

Other important contributions of GSM models encompass both discovery of network properties (Figure 2C) and multi-species's relationships (Figure 2D). From the existence of loops to pathway redundancy, multiple discoveries of network properties were driven by the usage and analysis of GSM models [6]. Also, by analysing the model of a pathogen, one can predict essential genes and metabolites for drug targeting [49].

2.3 Computational Representation of Compounds

2.3.1 Background

Molecules and atoms have been depicted since the early 1800s. It started in 1808 with the first scientific description of the atomic theory by John Dalton. Back then, he filled his publications with the first symbols of atoms and molecules [50]. Later, the so-called chemical formula was proposed by Berzelius as a way of representing the relative numbers of atoms in a compound. This was the first chemical representation suitable to be typed in a text body. The following two centuries were gifted by several breakthroughs that allowed the better understanding of chemical structures and properties.

So, generically, it can be stated that a chemical compound is a substance with two or more atoms of different types linked by chemical bonds and held in a well defined stoichiometry. The atoms are arranged spatially in different manners. This relative arrangement of the constituent atoms, along with the electronic structure, define the chemical structure of the compound. [51]

The representation of chemical compounds relies on the chemical structure, the type of atoms and/or even on their chemical properties. [52]

The advent of methods such as crystallography, Nuclear Magnetic Resonance (NMR) and Computer-Assisted Structure Elucidation (CASE) enabled the better and large-scale description of molecular structures [52]. As a result of those advances, more complex molecular structures had been identified. As a consequence, conventional methods of naming compounds (systematic chemical nomenclature) started to become unsuitable for such complexity since the names generated by those were often long and complex [53].

Chemical graph formalism

From a computational perspective, chemical structures can be represented as graphs, where the atoms are the nodes and the bonds are the edges. These graphs are undirected and labeled, since bonds have no direction and the nodes are labeled with the atom symbol. Moreover, concerning that two atoms can have more than one bond linking them, analogously, two nodes can have more than two edges between them. [53–55]

In fact, the analogy between a given compound structure and a topological graph underlies the development of many algorithms. These algorithms allow one to process and glean information about a given compound structure, such as the constituent atoms and their bonds. However, they are not able to capture their 3D structure. [54]

2.3.2 Connection tables

Connection Tables (CT) are 2D descriptors of chemical compounds [55]. Generically, there are two ways of presenting CTs. The first encompasses two lists: one for the atoms and another for the bonds (see Table 2: the connection table of benzene (depicted in Figure 3)). Another way of presenting these tables stands on redundant CT, which is later converted into a non-redundant one. [54]

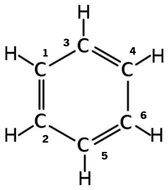


Figure 3: Benzene structure arbitrarily labeled

Table 2: CT of benzene

Atom list

1	C
2	C
3	C
4	C
5	C
6	C

Bond list

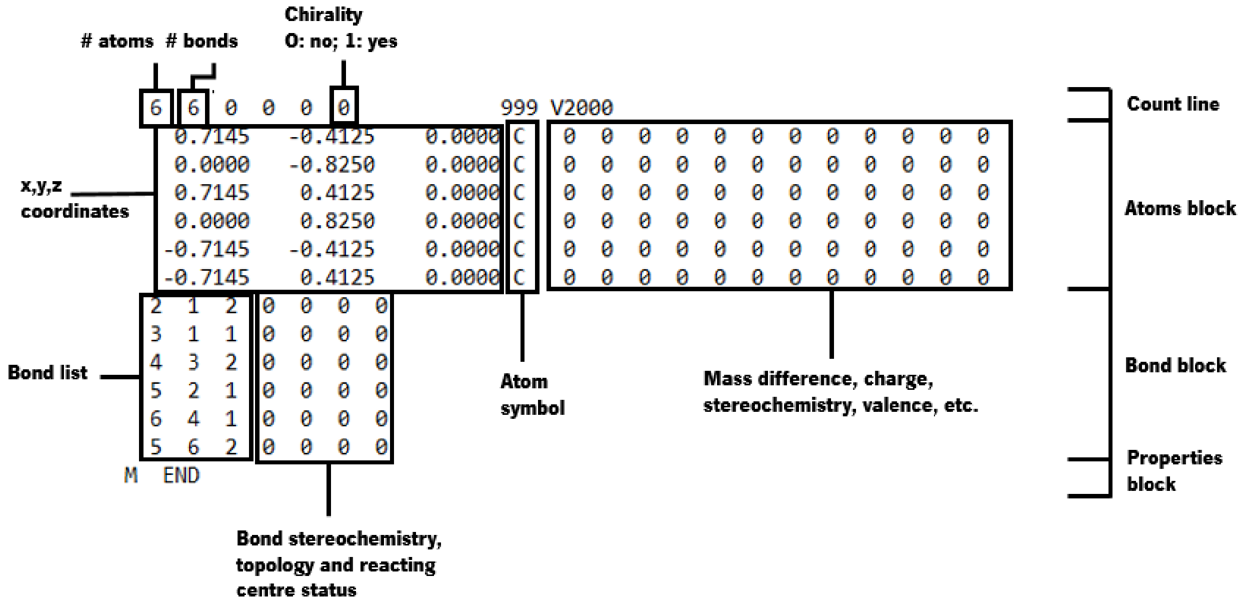
1 atom	2 atom	bond order
2	1	2
3	1	1
3	1	1
4	3	2
5	2	1
6	4	1
5	6	2

Either way, atoms are labeled canonically or arbitrarily, however, canonical labeling ensures an unique molecular representation and the reproducibility of its structure [53]. This canonicalization is underlain by the Morgan algorithm [56]. This algorithm's criteria to identify compounds is the connectivity value, starting on choosing the atom with the higher value and moving on its neighbors in a descendent manner [53]. Since the publication of the original algorithm in 1965, an update was performed in order to handle stereochemistry [57].

There are other ways of representing molecular graphs such as the matrix-based ones (adjacency matrix, distance matrix, bond-electron matrix, etc), however, in these approaches, the number of entries increases with the square of the number of atoms. Whereas in the CT, the number of entries increases linearly with the increase of the number of atoms. [54, 55]

Even so, CTs turn out to have some limitations. A CT only accounts on single valence bound structures, lacking on the representation, for instance, of delocalized bonds [53]. Moreover, CTs do not handle π-systems, coordination, inorganic compounds, reaction intermediates [58] and are not suitable for indexing in databases.

Although CTs have such limitations, they reveal to be the most utilized way of representing compound structures [55], with particular emphasis to Molecular Design Limited (MDL) connection table or CTfile [53, 59] (see Figure 4). Several versions of CTs were also developed: molfile, Rgroup file, Reaction file, Structure-data files, Reaction-data files and so on [53].



```
  6  6  0  0  0  0            999 V2000
   0.7145   -0.4125    0.0000 C  0  0  0  0  0  0  0  0  0  0  0  0
   0.0000   -0.8250    0.0000 C  0  0  0  0  0  0  0  0  0  0  0  0
   0.7145    0.4125    0.0000 C  0  0  0  0  0  0  0  0  0  0  0  0
   0.0000    0.8250    0.0000 C  0  0  0  0  0  0  0  0  0  0  0  0
  -0.7145   -0.4125    0.0000 C  0  0  0  0  0  0  0  0  0  0  0  0
  -0.7145    0.4125    0.0000 C  0  0  0  0  0  0  0  0  0  0  0  0
  2  1  2  0  0  0  0
  3  1  1  0  0  0  0
  4  3  2  0  0  0  0
  5  2  1  0  0  0  0
  6  4  1  0  0  0  0
  5  6  2  0  0  0  0
M  END
```

Figure 4: A CT of benzene used in CTfile format. CT downloaded from https://www.ebi.ac.uk/chebi/searchId.do?chebiId=CHEBI:16716 (ChEBI [60]).

2.3.3 Line notations

Line notations are linear strings of alphanumeric symbols [53] as well as compact, human readable and machine-friendly ways to encode molecules' structures [61]. The first widely used notation was the Wiswesser LineFormula Notation (WLN) [53, 62]. However, it revealed to have some limitations encoding compounds' stereochemistry. Eventually, its usage started to phase out in the 1980s. Later on, the Simplified Molecular Input Line Entry System (SMILES) emerged [63] and prevailed as the most popular until now [61, 64]. Furthermore, other line notations have been developed over the years (see Table 3, referring to the line notation of benzene structure depicted in Figure 3). Apart from SMILES, the IUPAC's InChI is the most widely used notation [61].

Table 3: The different line notations of benzene ring.

Line notation	Representation	Reference
WLN	R	[62]
SMILES	C1=CC=CC=C1	[63]
SLN	C[1]H:CH:CH:CH:CH:CH:@1	[65]
ROSDAL	1-2=3-4=5-6=1;	[66]
InChI	1S/C6H6/c1-2-4-6-5-3-1/h1-6H	[67]
InChIKey	UHOVQNZJYSORNB-UHFFFAOYSA-N	[67]
CAS	71-43-2	[68]

SMILES

SMILES are compact, human-readable and -writable. There are two main types of symbols in this notation: *atoms* and *bonds*. Having considered that, one is able to generate a molecular graph and, thus, apply graph-based algorithms to it. This notation is defined as a string with the chemical element symbol, bonds, indexes of broken cycles and parentheses enclosing branching vertices [63, 69]. The main disadvantage of SMILES is that they are not canonical (a compound can have different representations) [61]. However, several efforts have been made towards the development of a standard method for the generation of canonical SMILES [61, 63].

SMILES have several extensions such as SMILES arbitrary target specification (SMARTS) and SMIRKS.

SMARTS is a description language for molecular patterns that allows querying molecules substructures. Here, the labels of the nodes and edges are extended to include *logical operators*, special atomic and bond symbols. This allows SMARTS to represent atoms and bonds in a general manner. For instance, the symbols **[c,n;H1]** represent a molecule with an aromatic carbon or an aromatic nitrogen, and exactly one hydrogen atom.

SMIRKS results from the hybridization of SMILES and SMARTS language. It aims at the description of generic reactions by accounting on the transformation of reactants into products [70].

InChI

InChI is a way of representing compounds developed by International Union of Pure and Applied Chemistry (IUPAC) in 2005 as a non-proprietary, open source, and freely available resource [71]. InChI generation method relies on a structure-based and hierarchical approach. It ensures a strict uniqueness for each compound. Although it encompasses the entire organic compounds representation, InChI does not cover all the inorganic set [67].

An InChI is generated by using a CT. The method is divided in three steps: normalization, canonicalization and serialization [53, 67, 71]. Those steps are described below:

1. **Normalization** - the compound structure is converted into data structures. The normalization of mobile hydrogens, variable protonation and charge is performed [53].

2. **Canonicalization** - The atoms are labeled with numbers in a canonical manner by accounting on atomic equivalence/inequivalence relations [53, 67].

3. **Serialization** - the structure is serialized taking into consideration the labels assigned to each atom (node) and the output is generated [53].

As a hierarchical approach, this method formulates an output string (example in Table 3) with hierarchical layers. This is the most important characteristic of InChI [67]. Each layer (divided by "/") represents a class of structural information: the main layer (divided into the formula, atomic bonds and H-atoms sub-layers), charge, stereochemistry, isotope, fixed-H, and reconnected layer [53] (see an example in Figure 5). Note that not all layers are represented in the Figure. This occurs because only the main layer is present in all InChIs, the other layers are variable, depending on the compound properties and structure.

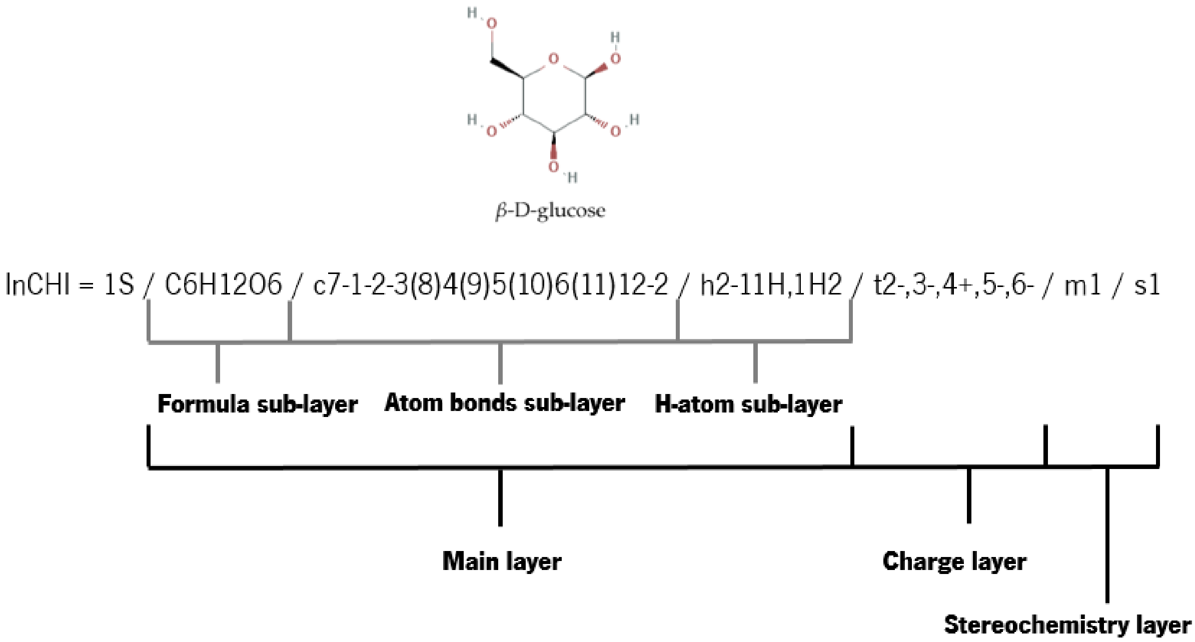


Figure 5: The InChI of β-D-glucose.

One of the main limitations of InChI is that search engines have difficulties in breaking the InChI character strings [53]. Another problem generated by the high detail of InChI is the incredibly high number of characters they can contain. This can be another drawback when it comes to storing this notation [53]. In order to address these limitations, the InChI can be converted into an InChIKey. Regarding the fixed number of characters in the key (check the 24 numbers of benzene InChIKey in Table 3), the storage of the chemical structure as well as its indexation in databases have been facilitated. InChIKey is divided into three blocks: the first block is for connectivity, the second for stereochemistry and the third for protonation.

2.3.4 Fragment codes

By describing a given compound, a set of characteristics are pointed out concerning the compound's functional groups, ring system, and so on. Correspondingly, a set of chemical substructures or characteristics that describe a given compound are the basis of the fragment coding system [53].

As a matter of fact, fragment codes are basically dictionaries that allow the indexation of the chemical "fragments" [54]. This can be useful to represent either the presence or the absence of fragments in a given structure. Assuming that 0s (zeros) represent the absence of a given substructure and 1s the presence of it, one can record a given compound as a bitstring composed only by 1s and 0s (zeros). This representation is often dubbed as "fingerprints" (see Figure 6) [53, 54]. As simple as these can be, they have one major limitation: ambiguity. The same fragment code can index different structures. However, this coding system reveals to be useful when it comes to divide molecules into classes. [54]

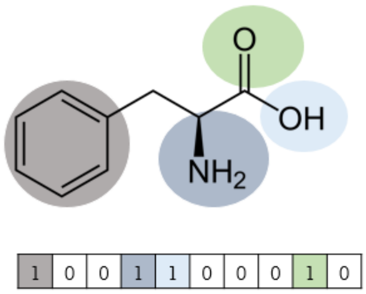


Figure 6: Example of a fragment encoding of phenylalanine. Adapted from [54].

2.3.5 Generic structures

Generic or abstract structures are diagrams that allow one to represent classes or sub-classes of compounds. These structures are composed by a fixed core and variable parts (R-groups). The definition of a given compound can rely on the part attached to the R-group. However, this does not occur in all abstract molecules. As for the representation of compounds such as quinones, repeating units are defined as enclosed structure regions assigned with a variable (number of repeating units). Whereas for other lipids, the R-group is defined by other molecules (fatty acids in this case). Having considered that, the class of compounds "*ubiquinones*" can be represented as depicted in Figure 7. Moreover, a region in R-group is enclosed by brackets with an associated variable (defining how much isoprene units the *ubiquinone* contains). In this case, the number of isoprene units in the R-group will specify the *ubiquinone*. On the other hand, *phosphatidylglycerol* has the fixed core composed by two glycerol bones linked by a phosphate group, whereas the R-group is defined by other molecules (fatty acids in this case), as shown in Figure 7). Herein, the variable structure is defined by the character R.

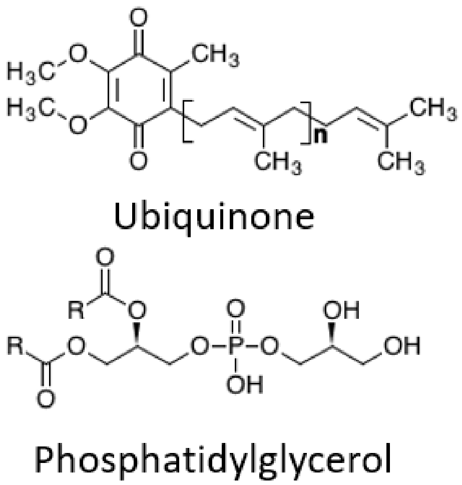


Figure 7: Example of a generic structures of ubiquinones and phosphatidylglycerols (retrieved from KEGG [72]). The R-group in the ubiquinone is enclosed by brackets and defined by an integer (number of isoprene units), whereas the R-group in the phosphatidylglycerol is defined by the character R (fatty acid's hydrocarbon chain).

2.3.6 *rdkit*

rdkit (http://www.rdkit.org) is an open-source chemoinformatics toolkit that allows handling and manipulating chemical structures. It includes Python wrappers that permit one to read molecular line notations, manipulating R groups, generating new molecules with SMARTS transformations, and so forth.

As for similarity and substructure search, SMARTS notations can be easily read and used to find all the molecules that share a specific substructure. In turn, one can decompose a compound and extract that specific substructure. This is particularly relevant for gleaning information of both long chains and molecular fixed cores.

Another interesting feature is the fact that one can filter molecules based on the chemical elements that compose their substructures, the type of established bonds (single, double or triple), and other characteristics.

Finally, *rdkit* allows the conversion of line notations from either manipulated or newly generated structures. Furthermore, it allows the estimation of relevant chemical properties based on the molecule structure.

Conclusively, *rdkit* is a flexible software with multiple features for reading, manipulating and retrieving chemical structures.

2.4 Biochemical databases and online resources

The access to biochemical databases underlies the GSM models' reconstruction. From annotated genomes, and their sequence, to information related to biochemical reactions, all of this information is extracted from databases.

Table 4: The different biochemical databases.

Database	Type of data	Compounds structure format	Website
ChEBI	C	MOL file, InChI, SMILES, InCHIKey	https://www.ebi.ac.uk/chebi/
Metacyc	G, C, P, E and R	MOL file, InChI, SMILES, InChIKey	https://metacyc.org/
ModelSEED	C, R and M	SMILES, InCHIKey	http://modelseed.org/genomes/
KEGG	G, C, P, E and R	MOL file	https://www.genome.jp/kegg/
BRENDA	E	InCHI	https://www.brenda-enzymes.org/
BiGG	M, C and R	-	http://bigg.ucsd.edu/
MetaNetX	M, C and R	SMILES, InCHI and InCHIKey	https://www.metanetx.org/
LMSD	C (lipids)	MOL file, InChI, SMILES, InChIKey	https://www.lipidmaps.org/
SLM	C (lipids) and R	MOL file, InChI, SMILES, InChIKey	https://www.swisslipids.org/
REACTOME	P, R, C	-	https://reactome.org/

G - genes; C - compounds; P - pathways; E - enzymes; M - models; R - reactions;

ChEBI [60] is a database that contains an ontology for compounds of biological interest. It has thousands of annotated metabolites with cross-references to other databases.

Metacyc [73] is a curated database for metabolic pathways. It contains information related to genes, metabolites, reactions and enzymes. Moreover, Metacyc contains several internally and externally developed ontologies. All this information is adapted for all domains of life.

ModelSEED [11] is an online resource that assists on the reconstruction of GSM models. It uses RAST to automatically generate annotations [74]. ModelSEED has an internal database that integrates different GSM models, biochemical reactions, metabolites as well as subsystems.

KEGG [72] is an online resource with eighteen databases. They are divided into four categories: systems information (includes information about pathways, functional hierarquies and modules), genomic information, chemical information (compounds, reactions, enzymes, etc) and health information. Markedly, KEGG represents a useful resource for the understanding and contextualization of large-scale molecular data.

BRENDA [75] is a database with a wide collection of enzyme functional data. All these data is extracted from literature and annotated manually by experts. Accordingly, the topics covered by this collection of enzyme data ranges from enzyme function and structure to genomic and protein sequences. Furthermore, all the information is organized by the respective Enzyme Comission (EC) number.

BiGG Models [22] is a knowledge base that contains high-quality and curated GSM models. It aims at the standardization of reactions and metabolites across all models and their integration. Moreover, it has a platform that enables the quick search and visualization of models.

MetaNetX [76] is a GSM models and biochemical pathways repository. It uses MNXref [77]: an algorithm of ”reconciliation” between the different nomenclatures of compounds. This algorithm is particularly relevant in the context of GSM models since metabolite identifiers found therein do not have references to databases of compounds. Instead, they are specific to the research group which is reconstructing the model [76].

The Lipid Maps Structure Database LMSD [8] is a relational database with annotated lipid structures and their *in silico* representation. LMSD divides lipids into eight categories (fatty acyls, glycerolipids, glycerophospholipids, sphingolipids, sterol lipids, prenol lipids, saccharolipids, polyketides). Moreover, this database is populated by data extracted from other databases, literature, experimental data and structures of lipids generated *in silico*.

SLM [9] is a knowledge resource that aims at exploring and describing lipidomic data. It includes an hierarchical classification of lipids as well as the linkage between them, metabolism and mass spectrometry data. Moreover, it is worth noting that SLM′ strategy is hypothesis-driven, as it incorporates *in silico* feasible predictions of lipids′ structures. To date (November 2020), 779759 lipid structures are comprised in this database.

REACTOME [78] is a freely available relational database that contains manually curated biological information. In this database, the core unit is the reaction. Moreover, biological compounds as well as their interactions are markedly organized into processes and pathways. Indeed, intermediary metabolism, signaling, regulation and other processes are well described and curated. Moreover, REACTOME browser provides a zoomable visualization of general representation of pathways as well as detailed information inside each pathway.

2.5 Lipids

Lipids are a ubiquitous and varied group of compounds. Their biological roles include being cell membranes′ components, energy storage sources, and being involved in signaling pathways. Generically, they are defined as hydrophobic and amphipathic molecules, and their main components are ketoacyl and isoprene groups. All of them derive from the condensation either of one of those sub units or another. Remarkably, there are no accurate estimations of how many different lipidic structures exist, as there happens to occur a panoply of alterations and transformations to their structures [10]. Moreover, due to the high number of different components with respect to the hydrocarbon chains and linking bonds, their combination to form lipidic structures results in a combinatorial explosion of different defined structures.

In 2005′s LIPID MAPS consortium, a set of standards regarding the lipid classification were defined. Correspondingly, the defined main classes were the following: Fatty Acids, Glycerolipids, Glycerophospholipids, Sphingolipids, Sterol Lipids, Prenol Lipids, Saccharolipids and Polyketides [10, 79].

Accordingly, one can split the structure of a lipid into two different parts: the backbone and the side chain. The backbone corresponds to the structural part that is common to a plurality of lipids (e.g. a phosphocholine), whereas the side chain is variable across a given class, and is composed either by ketoacyl or isoprene groups.

In the present work, only the main Glycerolipids (GL), Glycerophospholipids (GP), Sphingolipids (SP), and Prenol Lipids (PL) will be explored.

2.5.1 Glycerolipids

According to the LIPID MAPS classification, GL are a group of compounds that includes acylglycerols as well as alkyl and 1Z-alkenyl variants [10]. Their generic structure holds a glycerol backbone linked to one, two or three hydrophobic chains that have either ester or ether linkage to the backbone. They have a major role in bacteria, plant and mammalian's cell membrane formation [10]. An example of a GL structure is shown in Figure 8.

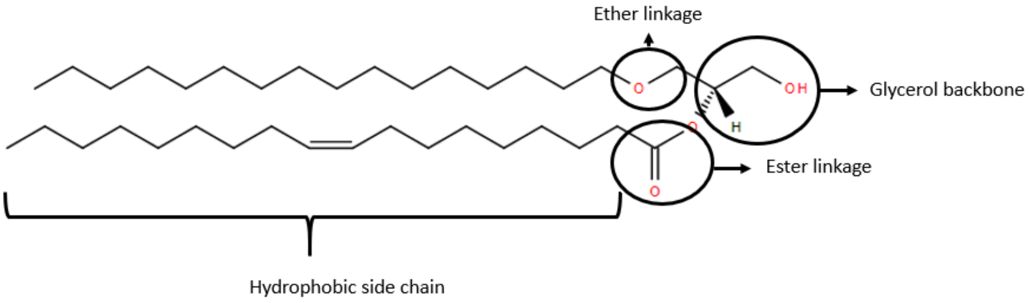


Figure 8: 1-O-hexadecyl-2-(9Z-octadecenoyl)-sn-glycerol structure extracted from LIPID MAPS. Here, the main features of this class of molecules are depicted.

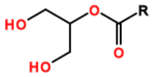
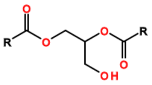
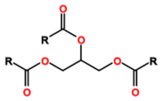
Computational representation

The main structural groups of GL are the following: the glycerol backbone and the side chains. Commonly, in order to represent the abstract structure of each class of GL, the hydrocarbon chains are represented as R groups. The different types of GL are depicted in Table 5. It is worth noting that these representations can vary according to the type of linkage and position of the hydrophobic side chain.

2.5.2 Glycerophopholipids

GP, commonly referred to as phospholipids, are key components of the membrane lipidic bilayer of all types of cells. As for their structure, they are typically composed by at least one glycerol unit in the backbone, as well as at least one phosphate group. Furthermore, the

Table 5: GL computational representation. The chosen compounds are representative, as their representation can vary according to the type of linkage and position of the hydrophobic side chain.

Glycerolipid	Backbone	SMILES (Backbone)
Monoacylglycerols		OCC(CO)OC(R)=O
Diacylglycerols		OCC(COC(R)=O)OC(R)=O
Triacylglycerols		RC(=O)OCC(COC(R)=O)OC(R)=O

sn-1 or/and sn-2 positions of the glycerol backbone are occupied by long chain fatty acids. Nevertheless, what distinguishes them the most is the nature of the polar head group at the sn-3 position. This description can be followed up in Figure 9.

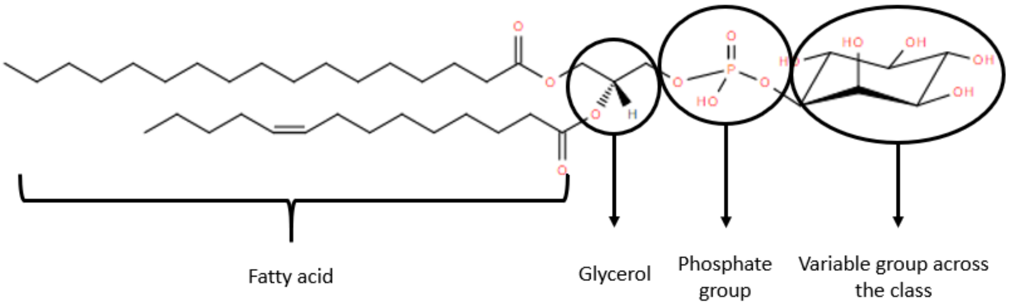


Figure 9: 1-heptadecanoyl-2-(9Z-tetradecenoyl)-sn-glycero-3-phospho-(1'-myo-inositol) structure extracted from LIPID MAPS. Here, the main features of this class of molecules are depicted.

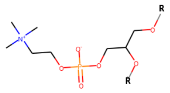
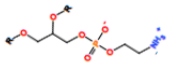
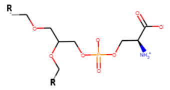
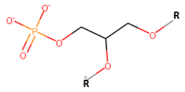
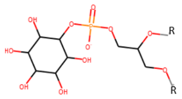
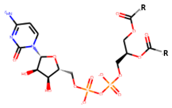
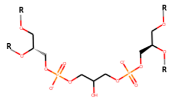
Computational representation

The main groups of GP are the following: the backbone (glycerophosphate), the polar group and the fatty acids. Commonly, in order to represent the abstract structure of each class of GP, the hydrocarbon chains are represented as R groups. The different types of GP is depicted in Table 6.

2.5.3 Sphingolipids

SP are a diverse group of compounds that share a sphingoid base backbone. The difference between chemical species within the class resides on the presence or absence of fatty acids, as well as on the structural groups attached to the sphingoid backbone (phosphocholines, phosphoethanoamines, sugar monomers, and polymers). The previous description is illustrated

Table 6: GP computational representation.

Glycerophospholipids	Backbone	SMILES (Backbone)
Glycerophosphocholines		C[N+](C)(C)CCOP([O-])(=O)OCC(COR)OR
Glycerophosphoethanolamines		[NH3+]CCOP([O-])(=O)OCC(COR)OR
Glycerophosphoserines		[NH3+][C@@H](COP([O-])(=O)OCC(COC(R))OC(R))C([O-])=O
Glycerophosphates		[O-]P([O-])(=O)OCC(CO(R))OR
Glycerophosphoinositols		OC1C(O)C(O)C(OP([O-])(=O)OCC(COR)OR)C(O)C1O
CDP-Glycerols		Nc1ccn([C@@H]2O[C@H](COP([O-])(=O)OP([O-])(=O)OC[C@@H](COC(R)=O)OC(R)=O)[C@@H](O)[C@H]2O)c(=O)n1
Glycerophosphoglycerophospho- glycerols		OC(COP([O-])(=O)OC[C@@H](COR)OR)COP([O-])(=O)OC[C@@H](COR)OR

in Figure 10. This group includes ceramides, phosphosphingolipids, and glycosphingolipids, essentially.

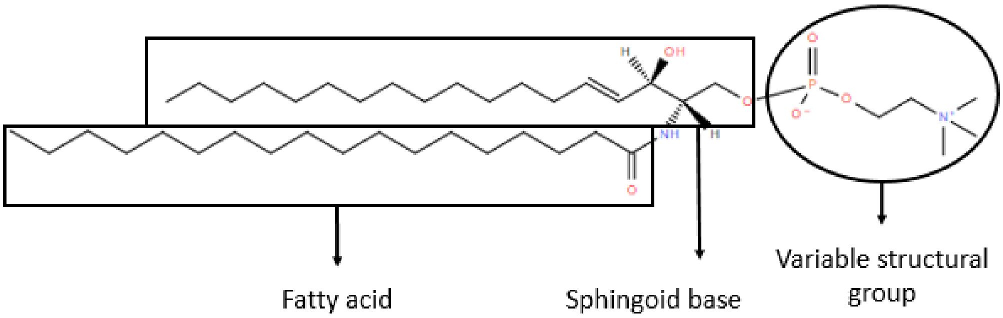
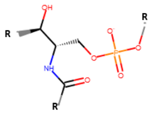
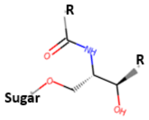
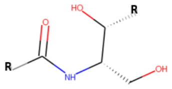
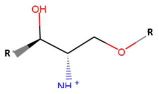


Figure 10: N-(octadecanoyl)-sphing-4-enine-1-phosphocholine taken from LIPID MAPS web-page. In this figure is shown the basic structure of a sphingolipid.

Computational representation

The main groups of SP are the following: the sphingoid backbone, the variable group and the fatty acid. Commonly, in order to represent the abstract structure of each class of SP, the hydrocarbon chains are represented as R groups. The different types of SP are depicted in Table 7.

Table 7: The different types of sphingolipids, their backbone, and their computational representation.

Sphingolipid	Backbone	SMILES (Backbone)
Sphingophospholipid		O[C@H](R)[C@H](COP([O-])(=O)OR)NC(R)=O
Glycosphingolipids		O[C@H](R)[C@H](COR)NC(R)=O
Ceramides		OC[C@H](NC(R)=O)[C@H](O)R
Sphingoid bases and derivatives		[NH3+][C@@H](COR)[C@H](O)R

2.5.4 Prenol Lipids

PL are a class of compounds that contain terpene units. Their ultimate precursors are the isopentenyl diphosphate and dimethylallyl diphosphate. Although this class is extensively wide, the present work will only focus on the electron-transfer quinones, as the polymeric nature of electron-transfer quinones rises problems in their computational representation.

Electron-transfer quinones are polymers that are represented by both generic and complete structures in several databases. Moreover, the biosynthetic pathway of these quinones is relatively small. These features will be explained and described thoroughly in the next paragraphs.

Ubiquinones, plastoquinones, menaquinones, rhodoquinones, and phylloquinones are a group of electron-transfer quinones with isoprene units arranged in a long side chain. The length of this chain varies among species. [80]

Ubiquinones are benzoquinones that occur in the plasma membranes of prokaryotes and in the inner mitochondrial membrane of eukaryotes [81]. They act as electron-transfer species in the oxidative phosphorylation stage of cellular respiration [80].

Plastoquinones are benzoquinones that occur in the chloroplast tylacoids of cyanobacteria and plants [80], having an important role in photosynthetic electron-transfer chain [82].

Table 8: Electron-transfer quinones

Quinone	Type	Occuring organisms	Electron-transfer in
ubiquinone	benzoquinone	prokaryotes and eukaryotes	oxidative phosphorylation
plastoquinone	benzoquinone	cyanobacteria and plants	photosynthesis
menaquinone	naphthoquinone	bacteria and archea	anaerobic respiration and photosynthesis
phylloquinones	naphthoquinone	plants and cyanobacteria	aerobic photosynthesis

Menaquinones are naphthoquinones that are involved in anaerobic ATP-generating redox reactions [83]. These compounds are constituents of membranes in bacteria and archaea and are the most common respiratory quinones found in biological systems [84].

Rhodoquinones are benzoquinones that are involved in anaerobic metabolism of bacteria and eukaryotes that live in hypoxic environments [85].

Phylloquinones are naphthoquinones found in plants' and cyanobacterial chloroplasts. They act as electron-acceptors during oxygenic photosynthesis [80, 86].

Computational representation

Isoprenoid quinones have two main structural groups: a head group and an isoprenoid side chain. The generic structure of these quinones is represented as follows: the head group is the backbone (defining the abstract or generic representation of the different isoprenoid quinones), whereas the variable part is the isoprenoid side chain (the number of repeating units will define the complete structure of these quinones). In Table 9 the head group of each electron-transfer quinone is described, with the exception of phylloquinone. In Figure 11, the isoprene unit along with its SMILES notation is illustrated.

It is worth noting that the computational representation of such compounds differs from the previously described lipids. The side chains of electron-transfer quinones only vary in the number of isoprene units. In contrast with the other lipids, the side chains do not present combinations of other molecules like fatty acids. Rather, the differentiation factor across the same subclass is the length of the side chain.

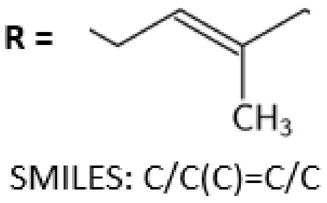
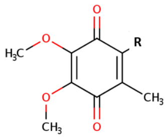
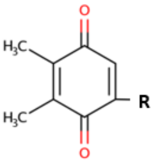
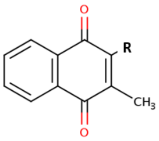


Figure 11: Isoprene unit.

Reference databases are not in total accordance with literature reports [80] regarding the variety of electron-transfer quinones and their precursors. Most of the databases do not list all the possible quinones' representation as enumerated in Table 10.

Table 9: Electron-transfer quinones' head group representation and the complete quinone representation in databases (with represented isoprene units)

Quinone	Head group	SMILES (Head group)	Number of isoprene units (in databases)
Ubiquinone		C1=C(C)C(=O)C(OC)(R)=C(OC)C1=O	1-10
Plastoquinone		RC1=CC(=O)C(C) =C(C)C1=O	1,2,3,8,9
Menaquinone		RC1=C(C) C(=O)c2ccccc2C1=O	1-13

2.6 The compound representation problem

The biochemical information present in GSM models is highly dependent on the data extracted from databases. As pointed out in the previous section, due to the complex structural representation of several compounds and their biosynthetic precursors, biochemical databases often represent them as abstract classes. On the other hand, in metabolic models, the absence and the complexity of such information lead experts to set biosynthetic pseudo reactions. Though this strategy can cope with the complex representation of such chemical species, several information can be lost.

2.6.1 Compounds representation in GSM models

Electron-transfer quinones

Several quinones and quinols are acceptors and donors of electrons, respectively. Even though different molecules of these kind are species-specific, the available genomic and proteomic resources are insufficient to provide valuable information regarding the type of quinone each organism should use. For this reason, it is relevant to flexibly change these species' representation version and make it easier to highlight similarities between different metabolic models.

Markedly, it is expected that different GSM models utilize different chemical species of quinones (with different number of isoprene units). While it is true that this distinction is important for the model accuracy, it is not that it eases comparisons between multiple models.

Table 10: The eletron-transfer quinones complete representation in several databases

Database	Quinone	Complete representation	All possible representations?
ChEBI	Ubiquinone	Ubiquinone-[1-10]	yes
	Plastoquinone	Plastoquinone-9	no
	Menaquinone	Menatetrenone, Menaquinone-[7-9]	no
MetaCyc	Ubiquinone	Ubiquinone-[1-10]	yes
	Plastoquinone	Plastoquinone-9	no
	Menaquinone	Menaquinone-[1-13]	no
Model SEED	Ubiquinone	Ubiquinone-1,-2,-6,-8,-9	no
	Plastoquinone	Plastoquinone-1,-9	no
	Menaquinone	Menaquinone-[2-13]	no
KEGG	Ubiquinone	Ubiquinone-1, -2, -6,-8,-9,-10	no
	Plastoquinone	Plastoquinone-1,-9	no
	Menaquinone	Menaquinone-9	no
BIGG	Ubiquinone	Ubiquinone-6,-8,-9,-10	no
	Plastoquinone	-	no
	Menaquinone	Menaquinone-4,-6,-8,-9,-10,-11	no
LIPID MAPS	Ubiquinone	Ubiquinone-4,-6,-8,-9,-10	no
	Plastoquinone	Plastoquinone-1	no
	Menaquinone	Menaquinone-9	no

As far as this is concerned, quinones with different side chains can be considered as different instances of the same entity.

Furthermore, the biosynthetic intermediates of these chemical species are specific, as they possess the same number of isoprene units as the quinone they are producing. For this reason, when introducing these pathways or swapping the type of quinone, inconsistencies might appear across the network. Thus, when one intends to swap a quinone, one has to do so for the whole set of biosynthetic precursors and conjugated acids.

Other Lipids

Due to lipids' variability and complex structures, their representation in GSM models is often generic. For instance, in biochemical databases, lipids' biosynthetic pathways are represented generically, as they are composed by generic reactions with generic lipids as reactants and/or products. Such reactions can be derived into reactions with structurally defined metabolites, originating complete/granulated reactions.

There are different approaches regarding the representation of lipid metabolism in GSM models. The first approach is the definition of a singular molecular formula as an abstract representation of several molecules. As an example, consider the Equation 1 retrieved from the *Saccharomyces cerevisiae S288C* model (iMM904) [9]. Herein, the 1-Acyl-sn-glycerol 3-phosphate, with a chemical formula of C1920H3622O700P100, is defined as a representative entity of 100 molecules of its kind. Correspondingly, the reactions with this metabolite will always have the stoichiometric coefficient scaled by 1/100. This formulation derives from the stoichiometries of the *Acyl CoA*s assigned as reactants.

$$\begin{aligned}&\mathit{Glycerol3\text{-}phosphate} + 0.08\ \mathit{Dodecanoyl\text{-}CoA} + 0.17\ \mathit{Hexadecenoyl\text{-}CoA} + 0.19\ \mathit{Octadecynoyl\text{-}CoA}\\ &+0.1\ \mathit{Tetradecanoyl\text{-}CoA} + 0.05\ \mathit{Stearoyl\text{-}CoA} + 0.27\ \mathit{Palmitoyl\text{-}CoA} + 0.09\ \mathit{Octadecenoyl\text{-}CoA}\\ &\rightarrow 0.01\ \mathit{1\text{-}Acyl\text{-}sn\text{-}glycerol\ 3\text{-}phosphate} + \mathit{CoA}\end{aligned} \tag{1}$$

This strategy was extensively adopted in several yeast models prior to the Yeast consensus model [7]. Although it reveals to be very concise, its rigideness in stoichiometric coefficients and the demand of having all the molecules listed in the reaction does not capture the adaptability and flexibility of the lipidome [7].

Another approach encompasses the usage of "ISA" reactions. Essentially, "ISA" reactions are additional reactions that permit the encapsulation of specific chemical species into generic ones. However, instead of defining rigid stoichiometries to the chemical species in one single reaction, "ISA" reactions are defined per each chemical species of a kind. All these reactions will produce an abstract entity, which will, afterwards, be used in other reactions (Figure 12).

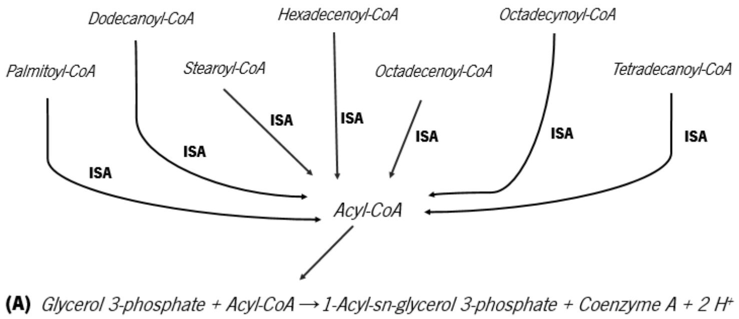


Figure 12: Some of the "ISA" reactions defined in the Yeast v6.0 model. There is one "ISA" reaction per Acyl CoA chemical species. This way, all the species are encapsulated into an abstract entity

The main advantage, comparing to the first approach, is that this one does not require the presence of all chemical species in order to satisfy the generic term requirement. Therefore, using "ISA" reactions, an "OR" rule (the model can use one chemical species OR another) is set and the eventual need for the generic compound is fulfilled. [7]

According to Aung and collaborators [7], the drawbacks of such approach are the following:

- Loss of information on individual reaction specificity;
- Chemical species that should not be used in some reactions will, however, be in computational simulations due to their conversion into the generic compound;
- Not capable of performing an inverse strategy: transforming a generic entity into a well defined compound;

Conversely, aiming for more accuracy in the representation of lipids, another strategy can be employed. This approach relies upon the generation of categorized individual reactions with structurally defined chemical species. This way, the abstract compound could be granulated, utilizing specific and adequate chemical species. As far as this approach is concerned, it can largely improve the accuracy and level of detail of GSM models' lipid metabolism. Moreover, it might provide better insights into the lipidome flexibility and output. However, this strategy would largely increase the number of reactions in the model, which can be a drawback, depending on the user's modelling scope. [7]

2.6.2 Useful considerations for the biochemical representation problem

As afore mentioned, the biochemical representation problem in GSM models highlights several gaps between lipid-specific databases and modelling. Therefore, several useful relationships and considerations will be, hereby, discussed.

Relationships

Biosynthetic relationships between compounds should be considered to ensure the correct and accurate representation of biochemical entities. Since structurally defined chemical species are related to their structurally defined precursors, the need of such relations is raised.

In fact, as shown in Figure 13, the information extracted from these relationships are of paramount importance when trying to granulate generic reactions. This is particularly corroborated by the absence of data related to reactions concerning structurally defined lipids.

Figure 13 depicts a way of capturing these relationships. Herein, functional and structural relationships are being established simultaneously, as the biosynthetic precursor of a specific compound is selected by the structural similarity between the product and reactants' side chains. In turn, this can also be considered a functional relationship, as it contains biosynthetic information.

Furthermore, it would be fundamental to assess whether a structurally defined chemical species is an instance of a generic entity. If so, structural relationships can be established. Examples of useful structural relationships are shown in Figure 14.

As for the electron donors' and acceptors' representation in GSM models, useful relationships can be established, mainly the "conjugated_acid_of" and "conjugated_base_of" ones. This information will be particularly useful when swapping electron-transfer chemical species, as if one has changed, its conjugated acid or base has to be changed, as well.

2.6.3 State-of-the-art tools

This subsection aims at describing both the advantages and disadvantages of the state-of-the-art computational tools for the revision of lipids' representation in GSM models.

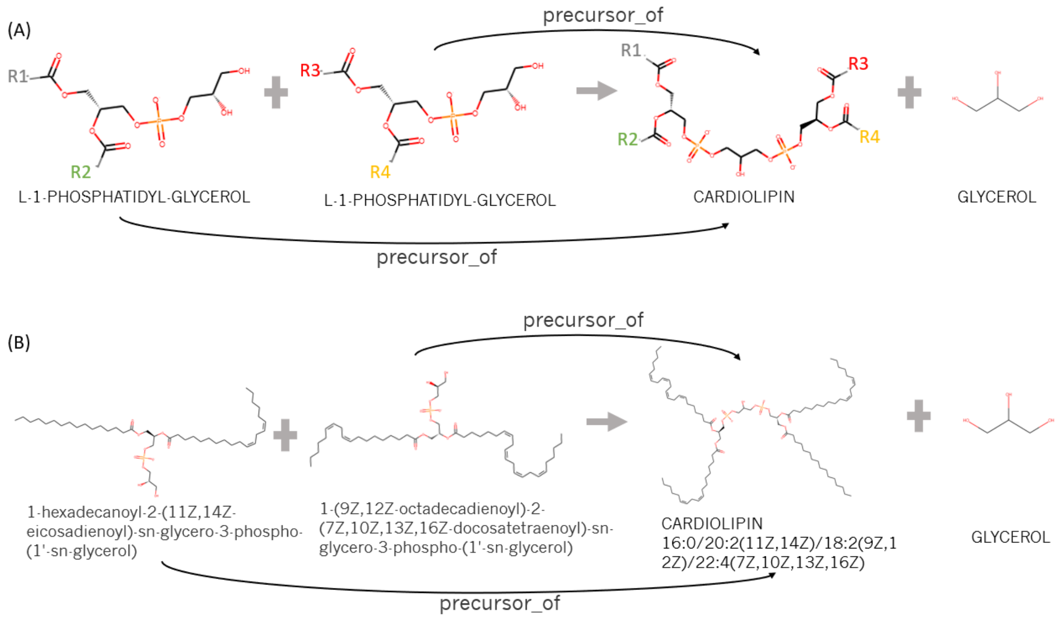

(A)
precursor_of
R1
R2
R3
R4
L-1-PHOSPHATIDYL-GLYCEROL
L-1-PHOSPHATIDYL-GLYCEROL
CARDIOLIPIN
GLYCEROL
precursor_of
(B)
precursor_of
1-hexadecanoyl-2-(11Z,14Z-eicosadienoyl)-sn-glycero-3-phospho-(1'-sn-glycerol)
1-(9Z,12Z-octadecadienoyl)-2-(7Z,10Z,13Z,16Z-docosatetraenoyl)-sn-glycero-3-phospho-(1'-sn-glycerol)
CARDIOLIPIN 16:0/20:2(11Z,14Z)/18:2(9Z,12Z)/22:4(7Z,10Z,13Z,16Z)
GLYCEROL
precursor_of

Figure 13: Reaction of cardiolipin synthesis using two L-1-phosphatidyl glycerol as reactants. The reaction **A** corresponds to the generic reaction constituted by generic compounds, whereas the reaction **B** is the granulation of the generic reaction **A** with the respective individual chemical species of each abstract entity. The "R groups" are replaced by variable parts. In the lipids' case, the "R groups" are replaced by carbon chains. The "precursor_of" relationship is established relying upon the type of side chain attached to the backbone.

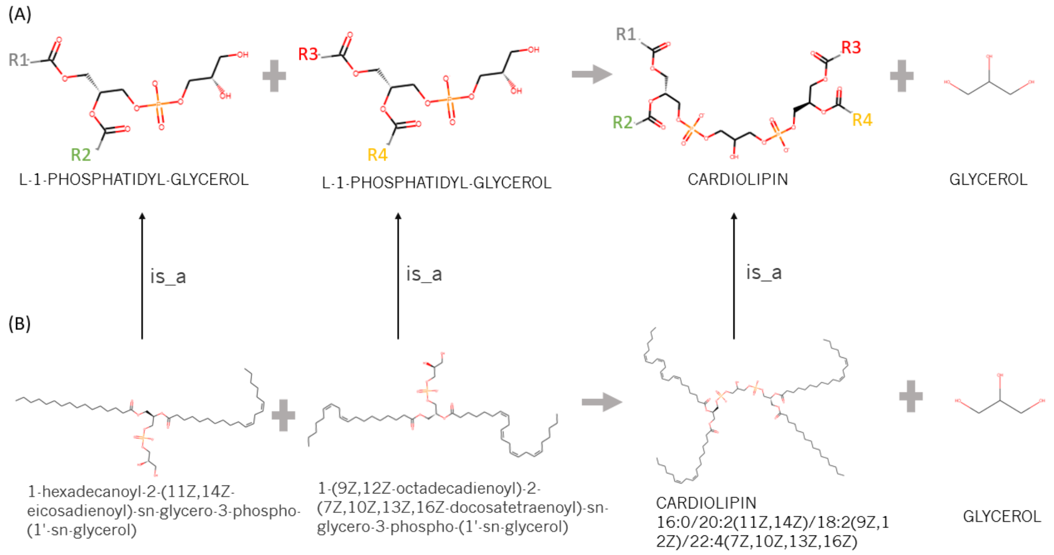

(A)
R1
R2
R3
R4
L-1-PHOSPHATIDYL-GLYCEROL
L-1-PHOSPHATIDYL-GLYCEROL
CARDIOLIPIN
GLYCEROL
is_a
is_a
is_a
(B)
1-hexadecanoyl-2-(11Z,14Z-eicosadienoyl)-sn-glycero-3-phospho-(1'-sn-glycerol)
1-(9Z,12Z-octadecadienoyl)-2-(7Z,10Z,13Z,16Z-docosatetraenoyl)-sn-glycero-3-phospho-(1'-sn-glycerol)
CARDIOLIPIN 16:0/20:2(11Z,14Z)/18:2(9Z,12Z)/22:4(7Z,10Z,13Z,16Z)
GLYCEROL

Figure 14: Reaction of cardiolipin synthesis using two L-1-phosphatidyl glycerol as reactants. The reaction **A** corresponds to the generic reaction constituted by generic compounds, whereas the reaction **B** is the granulation of the generic reaction **A** with the respective structurally defined chemical species of each abstract lipid. The "R groups" are replaced by variable parts. In the lipids' case, the "R groups" are replaced by carbon chains. The "is_a" relationship is established relying upon the type of backbone.

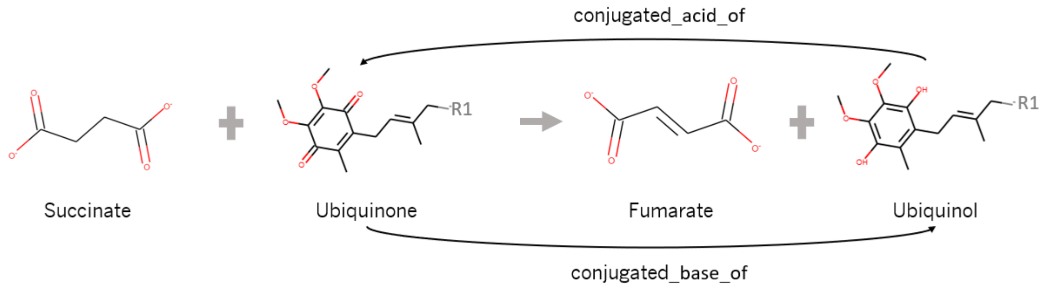


Figure 15: Succinate dehydrogenase reaction. This reaction is a typical acid-base reaction where the conjugated base and acid are generated. Correspondingly, it shows useful relationships between electron acceptors and donors.

SLIMEr

Split Lipids Into Measurable Entities (SLIMEr) [87] is a computational resource that revises lipids' representation as biomass components. SLIMEr combines fatty acid methyl ester (FAME) analysis with lipid profiling data to obtain correct predictions of the lipids' production and decomposition. The algorithm splits the generic lipid molecule into two parts for this matter: the acyl chains and the backbone.

FAME analysis is utilized to formulate the correct relative abundances of the acyl chains. Correspondingly, the acyl chains are assembled into a generic token (representing all the acyl chains) in a pseudo-reaction, whereas lipid profiling data is used for the estimation of relative abundances of each lipid class. Then, they are assembled into a generic backbone (representing all the lipid classes). Finally, a pseudo-reaction is generated to link the two previously assembled parts.

The authors tested their approach in an yeast model, using experimental data measured in 9 different conditions. Unsurprisingly, as experimental data was used to define stoichiometric coefficients for each backbone and acyl chain, the results from model simulations were similar to the experimentally determined. This approach has been described as useful towards the improvement of yeast model's predictive capability under different levels of stress.

The major drawbacks of this computational resource are the high dependence on experimental data, lack of the topological representation of lipid metabolism, and the fact of being only available on MatLab, which is not freely available for the non-academic community.

Poupin et al. 2020

The work of Poupin and collaborators [88] aims at mapping lipids in GSM models using ChEBI ontology. Specifically, this approach tries to reduce the gap between experimentally measured molecules and GSM models, as the differences in the level of annotation between lipidomics data and these models are patent.

The association between generic chemical structures and structurally defined ones is established accounting on the distance in the ChEBI ontology. Accordingly, the distance in the level of annotation of a generic compound in the metabolic network and another molecule species in the lipidomics dataset can be computed.

This method is available in a python package and can be also used in the MetExplore web-service [89].

Zhukova et al. 2014

Zhukova and collaborators' method [90, 91] relies on ChEBI ontology to generalize GSM models and highlight possible missing reactions.

The algorithm starts to compute a model by generalizing compounds and reactions. Herein, reactions that have the same generic compounds are assumed to be generic reactions. In the end, one will obtain a generalized model that allows an easier analysis and curation. [90]

Cyclic processes (with repeating reactions) taking place in the cell, such as the β-oxidation of fatty acids, can be captured as a cycle as well. So, any missing reaction in these processes will be easily highlighted. [91]

This method is available in a python package and is used by Mimoza [92] to aid in the generalized network visualization.

2.7 Graph databases and Neo4j

Graph databases had been used to connect complex biologic data over the years. Modelling the interactions between pathways, reactions, genes as well as metabolites and proteins can be tricky, involving hierarchies, and multiple relationship types between the same or different entities. The retrieval of such interactions ends up, most of the times, being highly recursive. In relational databases, multiple self-referential *JOINS* would be needed to capture much of this information, making it substantially inefficient.

In the case of lipids' hierarchies, implementing a relational database to represent and query this data would have a staggering difference in performance and in the length of the queries. For instance, if one wants to get the whole set of classes associated to a given lipid in a relational database, one has to query multiple *JOINS* in order to go all the way up in the hierarchy.

Furthermore, biological problems are oftentimes dealt with as path-y. Metabolic pathways are undoubtedly paths composed by reactions, that, in turn, are composed by metabolites. The establishment of "precursor_of" relationships will, in all likelihood, generate relevant and highly branched paths easy to retrieve in a graph database. However, if one wanted to model it in a relational rigid schema, one would require more storage space. Accordingly, each of the pathway intermediates of a given lipid would have to be linked directly to it by an intermediate table. In turn, each of the intermediates would, compulsorily, possess most of the intermediates associated to its successor. Thus, highly redundant data would be generated and stored without necessity. On the other hand, using a graph database, one would only have to establish relationships between direct precursors and successors (having reactions that transform one into another). Consequently, one would retrieve the intermediates by simply applying classic graph crossing algorithms.

Neo4j (https://neo4j.com/) is a No Structured Query Language (NoSQL), open-source, and graph-based database management system that applies graph theory for storage purposes. It includes a query language based on *Cypher*, which is very similar to SQL, and allows one to create, modify, transverse and extract useful information from the database. Moreover, *Neo4j* possess drivers that allow one to integrate *Cypher* queries in software developed in GO, Python, C#, JavaScript and Java. Lastly, it includes a highly efficient primary tool designated *Neo4j Admin*. This tool allows one to import high volumes of data in less than one minute, making backups, reports, and even analysing the consistency of the database. These features make it staggeringly easy to integrate and access information from a *Neo4j* database into a custom software.

3

BOIMMG FRAMEWORK

Several online resources and ontologies were designed to account on the structural relationships between chemical species. However, they fail to capture the functional relationships between compounds and their biosynthetic precursors. This is highly relevant in the context of GSM modelling, as it eases the process of converting generic biosynthetic pathways into specific ones and vice-versa. Moreover, to the best of our knowledge, there is no computational approach to integrate such information.

BOIMMG, a novel and modular approach aiming at tackling several issues in the representation of lipids in GSM models, is proposed in this chapter.

BOIMMG is here presented as a framework that follows essentially three steps: the integration of several databases, the generation of complex biochemical knowledge, and its integration in GSM models (Figure 16).

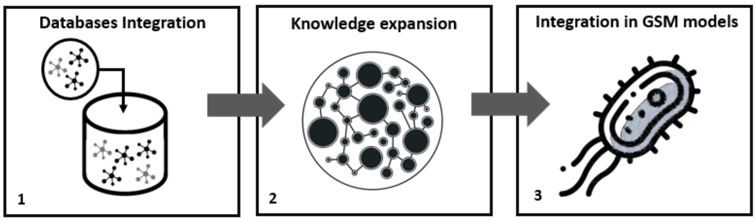


Figure 16: In this figure is depicted the BOIMMG pipeline. 1 - Integration of several databases; 2 - Semi-automated knowledge expansion 3 - integration of this information in GSM models

The second step of the pipeline can be a laborious task as the number of possible relationships between chemical species is high. Herein, a completely automated method for the detection of such relationships was not employed, since the high number of false positives could lead to a time-consuming curation. Thus, a semi-automated strategy was set and will be thoroughly described in the next section.

Lastly, the third part is the integration of the previously gathered and generated data in GSM models. This integration, however specific for each chemical species, was implemented in a flexible way to address similar cases. Consequently, as it is to be described in the next sections, different approaches were employed for the integration of glycerolipids, glycerophospholipids and electron-transfer quinones.

3.1 Formal Notation

A formal notation will be herein defined to describe the developed algorithms. The patterns enumerated in Table 11 will be followed to standardize the algorithms' notation.

Table 11: The notation that will be further used for writing algorithms and definitions

Type	Definition	Example
Cut zero	Empty set	$\emptyset$
Calligraphy typing	Special sets	$\mathcal{C}, \mathcal{R}, \mathcal{O}$
Capital letters	Sets	A, B, C
Lowercase letters	Single instances	a, b, c
Greek letters	Generic Functions	π, τ, Ψ
Verbose word	Biochemical functions	IsStructuralParentOf, GetReactions
Lowercase word	Specific property	*name*, *id*
Sigma	Boolean	σ

The notation can be extended to include derivations from the ones already defined.

Definition 1. Properties of single instances
A subscript symbol can define properties of single instances of a lowercase letter. For example: if a given instance is defined by a, and A is a given set of attributes, then a_A is defined by the attributes of a. Similarly, if a given identifier is defined by id and an instance by a, then a_{id} will be the instance's identifier.

Definition 2. Properties of a set or of special sets
Properties of a set or of a special set can be defined by a subscript symbol of a capital letter, or calligraphy typing letter. For example: if a set is defined by A, and B is a given set of attributes, then A_B is defined by the attributes associated to the set A. Similarly, if a given identifier is defined by id and a set by A, then A_{id} will be the identifier associated to a given set.

The following notation will be considered for the elements within brackets, and braces.

Table 12: List of symbols to represent sets

Type	Definition	Example
Brackets	Ordered set	$\langle\rangle$
Braces	Unordered sets	$\{\}$

Moreover, several basic operations over sets and single instances will be useful later on. They will be defined as follows.

- **Element of a set:** an arbitrary instance x is an element of a given set X when it is written $x \in X$;
- **Subset of a set:** an arbitrary set A is a subset of B if it is written $A \subseteq B$;
- **Union operation:** the union between an arbitrary set A and another defined as B is written as $A \cup B$;

- **Set size:** the total number of elements of the arbitrary set A will be written as $|A|$;
- **Cartesian product:** the Cartesian product of A and B, written $A \times B$ is defined as follows: $\{\langle a, b\rangle : a \in A \; and \; b \in B\}$;
- **Power set:** the power set of a given set A, written $\theta(A)$, is defined as the set of all the subsets of A;
- **Negation:** the negation, written $\neg$ returns $false$ if $\neg true$ and $true$ if $\neg false$;
- **Incrementation:** incrementation applied to an arbitrary $i \in \mathbb{N}$, written $i++$, increments 1 to the integer i;

Function 1. Subset or substring of a given length:
Domain: $\tau : \theta(A) \times (\mathbb{N} \cup \{-1\}) \rightarrow \theta(A)$, such that A is an arbitrary set
For an arbitrary ordered set X, the subset $Y \subseteq X$ with length i will be defined by the following function: $\tau(X,\ i)$. Moreover, for this purpose, the last element of a set will be defined by the integer -1. Hence, the subset Y with all elements of X except the last will be defined by the following: $\tau(X,\ -1)$. Analogously, if a string is defined by the ordered set S, the substring of S with length $|S| - 1$ will be returned by the function: $\tau(S,\ -1)$

Function 2. Get the i^{th} element of a set:
Domain: $\varphi : \theta(A) \times (\mathbb{N} \cup \{-1\}) \rightarrow A$
Let $A = \{a1, a2, ..., an\}$, such that $a1$ is the first element of a, $a2$ is the second element of a, and so forth. In order to obtain the i^{th} element of a, one will write as follows: $\varphi(A,\ i)$. For example: the third element of the ordered set A will be returned by the following function call: $\varphi(A,\ 3)$. Such as in function 1, the integer -1 can be defined as the last element. Hence, returning the last element of A would be achieved with the following function call: $\varphi(A,\ -1)$

Function 3. Add elements to a given set:
Domain: $\pi : B \times \theta(A) \rightarrow \theta(A) \cup \theta(B)$
An arbitrary element b will be added to the end of the ordered set A by the following function: $\pi(b,\ A)$. For example: let $A = \langle a1, a2, ...an\rangle$ and b an arbitrary instance, the function $\pi(b,\ A)$ will mutate A such that A will be, for that time forth, the set $\langle a1, a2, ...an, b\rangle$.

Function 4. Dictionary
Domain: $Dict : \mathcal{K} \rightarrow \mathcal{V}$
A dictionary will be written, from this time forth, as $Dict_{<something>}$ such that $something$ is the variable name associated to this function. If $\mathcal{K}$ is the universal set of keys in a given dictionary $Dict$, and $\mathcal{V}$ the image set associated to $\mathcal{K}$, the function $Dict$ will be defined as follows:

$$Dict(x) = \begin{cases} v \; for \; x = k \\ Dict(x) \; otherwise \end{cases} \tag{2}$$

Thus, assigning the value a to the key b would be defined as follows: $Dict(b) \leftarrow a$. Whereas returning a given value associated to the arbitrary key b will be written as follows: $Dict(b)$. Moreover, in order to define a dictionary in which both $\mathcal{K}$ and $\mathcal{V}$ are $\emptyset$, one will write as follows: $Dict \leftarrow Dict[\emptyset \rightarrow \emptyset]$.

3.1.1 Data structures and integration definitions

Formal definitions regarding data structures and integration will be enumerated in the present section.

Definition 3. Directed graphs
$G = (\mathcal{N},\ \mathcal{E})$ such that G is the directed graph. $\mathcal{N}$ is defined as the set of all nodes of G and $\mathcal{E}$ is defined as the set of all edges of G. Each edge starts in a given node of G and ends up in a different node of G, having a well-defined orientation.

Definition 4. Node Labels
One and only one label l is assigned to each node $n \in \mathcal{N}$. Each node's label n_l has to respect the following condition: $n_l \in \mathcal{N_L}$ such as $\mathcal{N_L} = \{$ "ModelSeedCompound", "SwissLipidsCompound", "LipidMapsCompound", "Compound" $\}$

Definition 5. Edge Labels
One and only one label l is assigned to each edge. Each label e_l has to respect the following condition: $e_l \in \mathcal{E_L}$ such as $\mathcal{E_L} = \{$ "is_a", "component_of"$\}$

Definition 6. Objects
Considering Objects as the universal set $\mathcal{O}$ such that $\mathcal{O} \in \mathcal{N}$ and $o \in \mathcal{O}$, o is defined by $\langle o_i,\ o_l,\ o_A \rangle$, where o_i is the internal identifier of a given node, and o_A is the set of attributes of o.

Definition 7. Relationships
Considering Relationships as the universal set $\mathcal{R}$, such that $\mathcal{R} \in \mathcal{E}$ and $r \in R$, the relationship r can be defined as the association of two Objects such that, $r = \langle o1,\ r_i,\ r_l,\ r_A,\ o2 \rangle$, and $o1,\ o2 \in \mathcal{O}$ where r_i is the internal identifier of an edge, $r_l \in \mathcal{R_L}$, r_A is a set of attributes of r, whereas $o1$ is the object from where the relationship starts and $o2$ is the object where the relationship ends. In this case, r_l will have the chemical and/or biological meaning regarding the two involved Objects ($o1$ and $o2$).

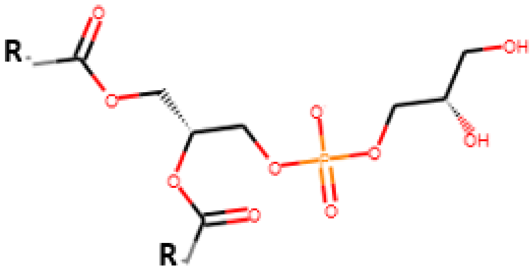
Definition 8. Structurally defined compounds' uniqueness
Considering the compounds defined by the arbitrary Object instances $o1$ and $o2$ such that $o1$,$o2 \in \mathcal{O}$, and $o1_{inchikey}$ is the InChIKey of the compound $o1$ and $o1_{inchikey} \in o1_A$, $o2_{inchikey}$ is the InChIKey of the compound $o2$ and $o2_{inchikey} \in o2_A$, then $o1 = o2$ if but only if $\tau(o1_{inchikey}\ , -1) = \tau(o2_{inchikey}\ , -1)$

Definition 9. Generic compounds' uniqueness
Considering the compounds defined by the arbitrary Object instances $o1$ and $o2$ such that $o1$,$o2 \in \mathcal{O}$, considering that $o1_{canonical\_smiles}$ is the canonical SMILES of the compound $o1$ and $o1_{canonical\_smiles} \in o1_A$, $o2_{canonical\_smiles}$ is the canonical SMILES of the compound $o2$ and $o2_{canonical\_smiles} \in o2_A$, then $o1 = o2$ if but only if $o1_{canonical\_smiles} = o2_{canonical\_smiles}$

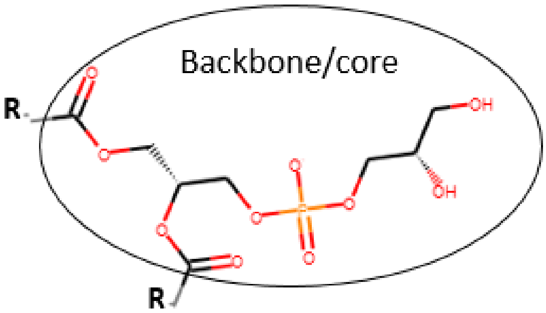
3.1.2 Biochemical definitions and operations

Useful biochemical definitions and operations will be enumerated in this subsection. Accordingly, the understanding of the present work will be considerably facilitated with the following definitions.

Definition 10. R groups
Let a be the structure bellow. The R group of the structure a is defined by the letter **R**. It corresponds to a variable structure that is attached to a well defined and fixed substructure. From this time forth, an R group will be defined by the following variable: R_{group}.

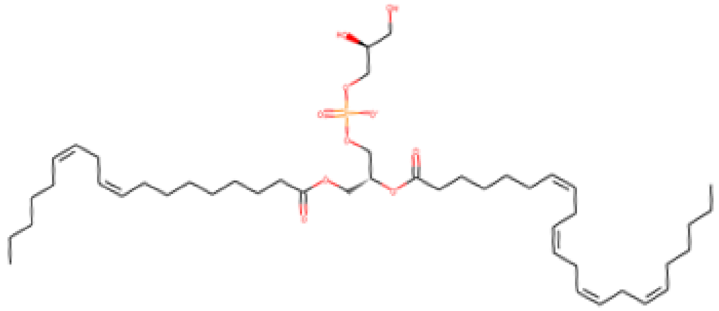


Definition 11. Backbone/core
Let a be the structure bellow. The backbone/core of the structure a is defined by the substructure of a (rounded below). It can be stated that the backbone/core is a well defined and fixed substructure that corresponds to the whole structure of the molecule without the R_{groups}.



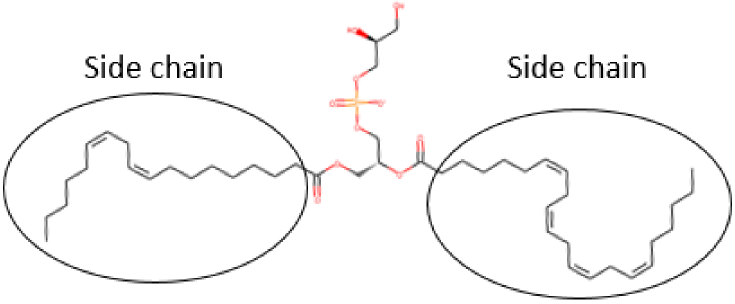
Definition 12. Generic/abstract compound
Let the generic compound be a. a will be represented by a structure with at least one R_{group}. Thus, a generic compound can be defined by an entity whose structure is not completely defined. Furthermore, it is common that these compounds are assigned to a given chemical class, as they can represent a set of chemical variants with the same backbone.

Definition 13. Structurally defined compound
Let c be the structurally defined compound illustrated below. c, by definition, will be one compound whose correspondent structure has no R_{groups} and is completely structurally defined.



Definition 14. Side chain

Let c be the compound illustrated below. The side chains will be, from this time forth, defined by the substructures that substitute the R_{groups}. Furthermore, the side chain attached to the backbone will generate one structurally defined compound.

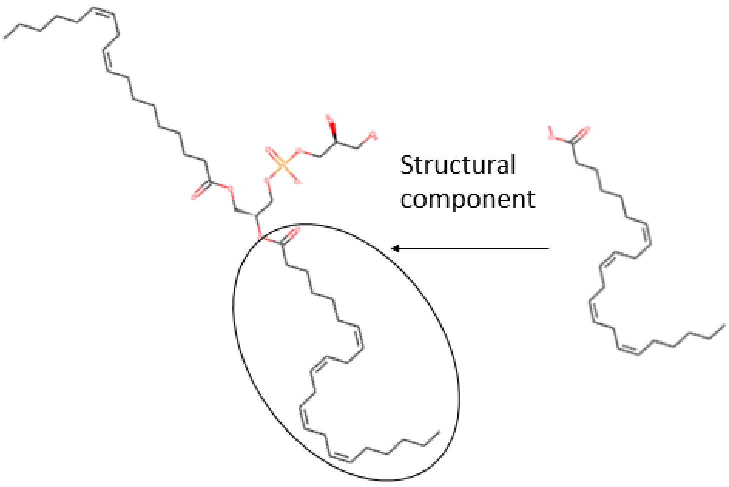


Definition 15. Structural parent and child

Considering the generic compound p, p is the structural parent of another compound c if and only if p has a common backbone with c. c is not necessarily a structurally defined compound. Analogously, if the previous condition is satisfied, then one can also state that c is a structural child of p.

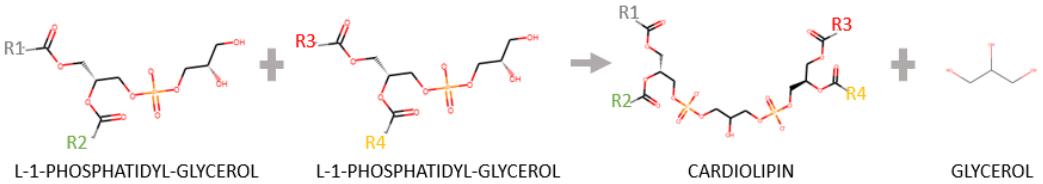
Definition 16. Structural components

Considering the two compounds illustrated below, the one on the right as p and the one on the left as c, c is a structural component of p if and only if the whole structure of c is contained within p.



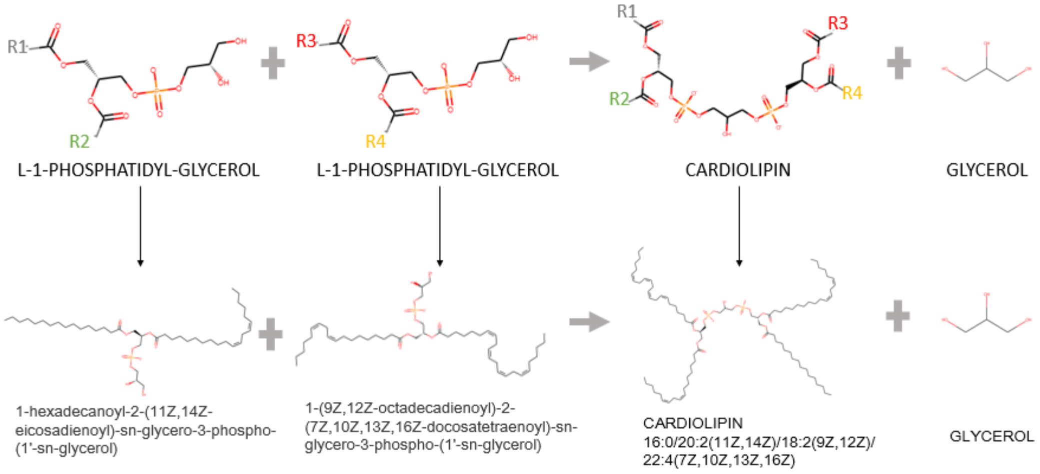
Definition 17. Generic reaction

A generic reaction will be, for this time forth, defined as reactions where at least one of the reactants and products is a generic compound (according to Definition 12). An example of one generic reaction is illustrated below.



Definition 18. Granulated/complete reaction

A granulated reaction will be defined, from this time forth, as reactions that can be assumed as derivations of a generic reaction (according to the Definition 18). The reactants and products of these reactions are all structurally defined compounds (according to the Definition 13). The aforementioned derivation is illustrated below. The generic reaction is on top of the image, whereas the granulated one is present below the arrows.


R1
R2
L-1-PHOSPHATIDYL-GLYCEROL
R3
R4
L-1-PHOSPHATIDYL-GLYCEROL
R1
R3
R2
R4
CARDIOLIPIN
GLYCEROL
1-hexadecanoyl-2-(11Z,14Z-eicosadienoyl)-sn-glycero-3-phospho-(1'-sn-glycerol)
1-(9Z,12Z-octadecadienoyl)-2-(7Z,10Z,13Z,16Z-docosatetraenoyl)-sn-glycero-3-phospho-(1'-sn-glycerol)
CARDIOLIPIN 16:0/20:2(11Z,14Z)/18:2(9Z,12Z)/22:4(7Z,10Z,13Z,16Z)
GLYCEROL

Definition 19. Generic biosynthetic pathway
A given generic biosynthetic pathway is defined by the set of generic reactions (Definition 17) connected towards the production of a generic compound (Definition 12).

For the following functions, let one consider $\mathcal{C}$ the universal set of arbitrary compounds, where $\mathcal{C} = \{c1, c2, c3, ..., cn\}$. Here, c are the instances of the entity compound $\mathcal{C}$

Function 5. IsStructuralParentOf
Domain: IsStructuralParentOf : $\mathcal{C} \times \mathcal{C} \rightarrow \mathbb{B}$
Let $c1$ and $c2$ be arbitrary compounds. If $c1$ is structural parent of $c2$ (according to Definition 15), then the function returns $true$, otherwise it returns $false$. Analogously, if IsStructuralParentOf($c1$, $c2$), then $c2$ is necessarily structural child of $c1$.

Function 6. HaveCommonStructuralParentWith
Domain: HaveCommonStructuralParentWith : $\mathcal{C} \times \mathcal{C} \rightarrow \mathbb{B}$
Let $c1$ and $c2$ be arbitrary compounds. Also, let $c3$ be the structural parent of $c1$ and $c4$ be the structural parent of $c2$ (according to Definition 15), if $c4 = c3$, then this function returns $true$, otherwise it returns $false$.

3.2 Databases Integration

The first step of BOIMMG's workflow is the integration of several databases, which has encompassed the data extraction from different sources, its transformation, loading and further integration.

Existing databases with specific information of lipids must be considered, according to the present work's scope. To the best of our knowledge, LMSD [8] and SLM [9] are the databases with the broader collection of structurally defined lipid species. Moreover, to bridge these lipid-specific databases with metabolic modelling, Model SEED data was also integrated. It is worth noting that Model SEED [11] database not only gathers information from several GSM models but also from BiGG [93], KEGG [72] and MetaCyc [35].

The general workflow is defined in Figure 17. The workflow starts with the data extraction from different sources. Posteriorly, the raw data will be transformed into a standard format towards its conversion into a graph. Lastly, this information will be loaded into a *Neo4j* graph database (version 3.9.). Furthermore, data integration will allow to cope with high volumes of redundant data.

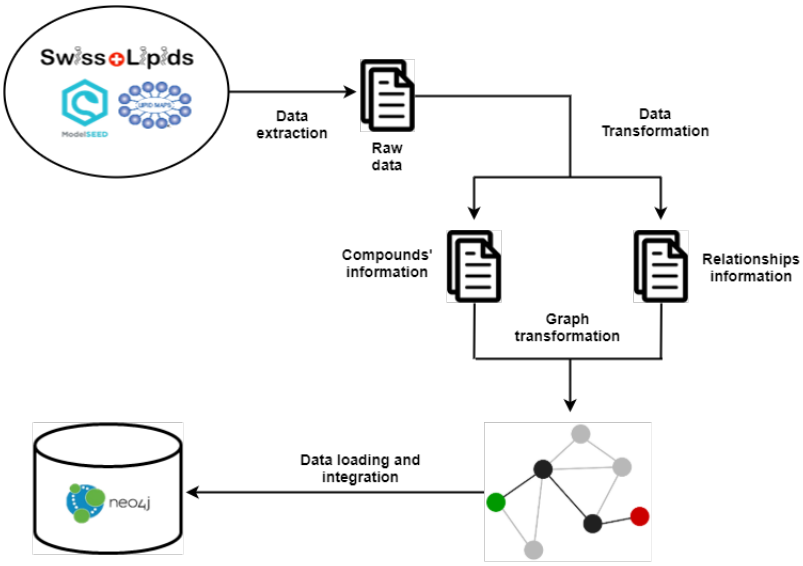


Figure 17: General workflow of the databases integration. It starts by extracting raw data files from SLM, LMSD, and ModelSEED. Posteriorly, the data is transformed in two different standardized files: one for the nodes (compounds) and another for the edges (relationships). This information is converted into graphs and loaded into a *Neo4j* database. In the end, an integration of the redundant data will be performed.

3.2.1 Data details and extraction

This subsection will describe each database's data files characteristics and extraction method employed.

SwissLipids (SLM)

Up to the present date (November 2020), the SLM database accounts for 779759 lipid structures. Moreover, it includes a hierarchy of chemical structures starting from the "Lipid" entity (root) and ending up with the structurally defined lipids (leaves). In addition, SLM contains information about each lipid's structural components. Regarding the data extraction, a TSV format file was downloaded from *https://www.swisslipids.org/#/downloads*. An example of the information contained in this file is shown in Table 13.

Table 13: Information contained in the TSV file retrieved from SLM database. It contains information about compounds: their structural parents, components, name, synonyms, structure representations and so forth. This table contains an example of the information present in each row.

Data	**Example**
SLM identifier	SLM:000000178
Level	Isomeric subspecies
Name	N-(docosanoyl)-15-methylhexadecasphing-4-enine
Abbreviation	Cer(iso-d17:1(4E)/22:0)
Synonyms	N-docosanoyl-15-methylhexadecasphing-4-enine — Ceramide (iso-d17:1(4E)/22:0)
Lipid class	SLM:000000002
Parent	SLM:000392021
Components	SLM:000000827 (n-acyl)
SMILES (pH7.3)	CCCCCCCCCCCCCCCCCCCCCC(=O)N[C@@H](CO)[C@H](O)\C=C\CCCCCCCCCC(C)C
InChI (pH7.3)	InChI=1S/C39H77NO3/c1-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-22-25-28-31-34-39(43)40-37(35-41)38 (42)33-30-27-24-21-19-20-23-26-29-32-36(2)3/h30,33,36-38,41-42H, 4-29,31-32,34-35H2,1-3H3,(H,40,43)/b33-30+/t37-,38+/m0/s1
InChI key (pH7.3)	InChIKey=XMCZTIGIXKXPGG-KNEGYRQWSA-N
Formula (pH7.3)	C39H77NO3
Charge (pH7.3)	0
Mass (pH7.3)	608.0336
Exact Mass (neutral form)	607.590345
...	...
CHEBI	71377
LIPID MAPS	-
HMDB	-
PMID	19372430

ModelSEED

Up to the present date (November 2020), ModelSEED included 33978 compounds. Data was extracted from three different tab-delimited text files: one with the compounds general information (Table 15), a file with their chemical structure representation (Table 16), and another with the cross-references to other databases (Table 17). These files were downloaded from the links presented in Table 14.

Table 14: Model SEED files and where they can be downloaded.

File	**url**
ModelSEED compounds' general information file	*https://github.com/ModelSEED/ModelSEEDDatabase/blob/master/Biochemistry/compounds.tsv*
ModelSEED compounds' structures file	*https://github.com/ModelSEED/ModelSEEDDatabase/blob/master/Biochemistry/Structures/Unique_ModelSEED_Structures.txt*
ModelSEED compounds' cross-references	*https://raw.githubusercontent.com/ModelSEED/ModelSEEDDatabase/master/Biochemistry/Aliases/Unique_ModelSEED_Compound_Aliases.txt*

Table 15: Information contained in the tab delimited text file retrieved from ModelSEED database, which contains general information about chemical compounds. This table contains an example of the information present in each row.

Data	Example
ModelSEED identifier	cpd15421
Abbreviation	cdpdodecg
Name	CDP-1,2-dioctadecanoylglycerol
Formula	C48H87N3O15P2
Mass	1007
Source	Primary Database
InchiKey	PDCWLWQTNQCGRI-IGIWICMZSA-L
Charge	-2
is_core	1
is_obsolete	0
linked_compound	-
is_cofactor	0
deltag	233.19
...	...
Aliases	Name: CDP-1,2-dioctadecanoylglycerol—BiGG: cdpdodecg—EcoCyc: CPD-12814—MetaCyc: CPD-12814—iAF1260: cdpdodecg—iGT196: cdpdodecg
SMILES	CCCCCCCCCCCCCCCCCC(=O)OC[C@H](COP(=O)([O-])OP (=O)([O-])OC[C@H]1O[C@@H](n2ccc(N)nc2=O)[C@H](O)[C@@H]1O)OC(=O) CCCCCCCCCCCCCCCCC
notes	GC—EQ—EQU

Table 16: Information contained in the tab delimited text file retrieved from ModelSEED database, which contains information about chemical compounds' structures. This table contains an example of the information present in each row.

Data	Example
ModelSEED identifier	cpd15421
Type	SMILES
Aliases	CPD-12814
Formula	C48H87N3O15P2
Charge	-2
Structure	CCCCCCCCCCCCCCCCCC(=O)OC[C@H] (COP(=O)([O-])OP(=O) ([O-])OC[C@H]1O[C@@H] (n2ccc(N)nc2=O)[C@H](O)[C@@H]1O) OC(=O)CCCCCCCCCCCCCCCCC

Table 17: Information contained in the tab delimited text file retrieved from ModelSEED database, which contains information about chemical compounds' aliases in several databases. This table contains an example of the information present in each row.

Data	Example
ModelSEED identifier	cpd15421
External ID	cdpdodecg
Source	BiGG

LIPID MAPS

LIPID MAPS compiles five databases: Lipid Structures (LMSD), lipid-related genes and Proteins (LMPD), In-Silico Structure Database (LMISSD), Computationally-generated Bulk Lipids (COMP_DB), and Lipidomic Ion Mobility Database. For the present work, only the LMSD will be considered. Up to the present date (November 2020), this database includes more than 45000 lipid structures. Data was extracted from a TSV format file downloaded in *https://www.lipidmaps.org/rest/compound/lm_id/LM/all/download*. An example of the information contained in this file is described in Table 18.

Table 18: Information contained in the TSV file retrieved from LIPID MAPS database. This table contains an example of the information present in each row.

Data	Example
LIPID MAPS identifier	LMGP13010001
Name	CDP-DG(12:0/12:0)
sys_name	1,2-Didodecanoyl-sn-glycero-3-cytidine-5'-diphosphate
Synonyms	CDP-DG(24:0); CDP-DG(12:0/12:0)
Abbrev	CDP-DG 24:0
abbrev_chains	CDP-DG 12:0/12:0
core	Glycerophospholipids [GP]
main_class	CDP-Glycerols [GP13]
sub_class	CDP-diacylglycerols [GP1301]
class_level4	-
exactmass	841.389098
formula	C36H65N3O15P2
inchi	InChI=1S/C36H65N3O15P2/c1-3-5-7-9-11-13-15-17-19-21-31(40)49-25-28(52-32(41)22-20-18-16-14-12-10-8-6-4-2)26-50-55(45,46)54-56(47,48)51-27-29-33(42)34(43)35(53-29)39-24-23-30(37)38-36(39)44/h23-24,28-29,33-35,42-43H,3-22,25-27H2,1-2H3,(H,45,46)(H,47,48)(H2,37,38,44)/t28-,29-,33+,34?,35-/m1/s1
inchi_key	PTPPKXVNJJIECF-MYNNLVAUSA-N
...	...
SMILES	[C@]([H])(OC(CCCCCCCCCCC)=O)(COP(=O)(O)OP(=O)(O)OC[C@H]1O[C@@H](N2C=CC(N)=NC2=O)C(O)[C@H]1O)COC(CCCCCCCCCCC)=O

3.2.2 Data transformation and loading

Each source's data files have their format and type of information. Hence, data transformations were performed in order to standardize it.

SLM raw data transformation

SLM's raw data was divided into two main parts: compounds' general information and hierarchy.

As for the compounds' general information, the raw data was transformed encompassing the following tasks:

- Generic compounds were filtered only to include the ones with the hierarchy's "Level" of "Class";
- Rows were filtered to include only structural defined compounds, besides the ones filtered in the previous task;
- Relevant compounds' features were maintained: name, SMILES, InChI, InChIKey, formula, charge, and mass (pH 7.3). The rest was not considered.
- Each SMILES was transformed into canonical SMILES with *rdkit*;

It is worth noting that all SMILES were converted into canonical SMILES prior to the data loading, to standardize the abstract compounds, as these cannot be represented by InchiKeys nor InChIs. Nevertheless, only the structurally defined compounds and the main classes were imported, preventing the integration of unnecessary entries into BOIMMG's database.

As far as the relationships are concerned, only two types were considered: "is_a" and "component_of". "Is_a" relationships are defined when there is a structural relationship between compounds. On the other hand, the type "component_of" refers to the presence of a given molecule in another one. This information is present in the SLM data file (Figure 13). A starting point and an ending point were defined for each relationship to transform it into a standard format.

The "component_of" relationships were extracted from SLM's raw data as follows:

- **Starting point**: SLM's identifiers in the "Components" column (e.g. in the example presented in Figure 13, the starting point would be "SLM:000000827");
- **Ending point**: the SLM's identifier of the compound being analysed (e.g. in the example presented in Figure 13, the ending point would be "SLM:000000178")

On the other hand, "is_a" relationships were extracted from SLM's raw data as follows:

- **Starting point**: the SLM's identifier of the compound being analysed (e.g. "SLM:000000178" in Figure 13);

- **Ending point**: SLM's identifiers in the "Parent" column (e.g. in the example presented in Figure 13, the ending point would be "SLM:000392021");

ModelSEED raw data transformation

As for ModelSEED data, three different files were used and merged. Moreover, several data was transformed and filtered as follows:

- Relevant chemical information: name, SMILES, InChI, InChIKey, formula, charge, and mass; the rest was otherwise ignored.
- SMILES were converted into canonical SMILES with *rdkit*;
- Compounds' aliases were filtered to include only the ones related to BiGG, MetaCyc, MetaNetX, and KEGG;

LMSD raw data transformation

The following tasks were performed to obtain a standard format of the LMSD data:

- Relevant compounds' data was included: name, SMILES, InChI, InChIKey, formula, charge, and mass; the rest was otherwise discarded.
- SMILES were transformed into canonical SMILES with *rdkit*;

Graph transformation and Data loading

As for the data loading, the *neo4j-admin* tools were used, as the respective data importer is considerably faster than using *Neo4j Cypher* (see more in https://neo4j.com/docs/operations-manual/current/tutorial/neo4j-admin-import/). Correspondingly, the information retrieved from different sources was converted into two different files with the *neo4j-admin* importer's standard format. The first file corresponded to each compound's general information, whereas the second contained information about relationships established between compounds. It is worth noting that the latter were extracted from SLM's hierarchy, which does not necessarily include LMSD nor Model SEED compounds.

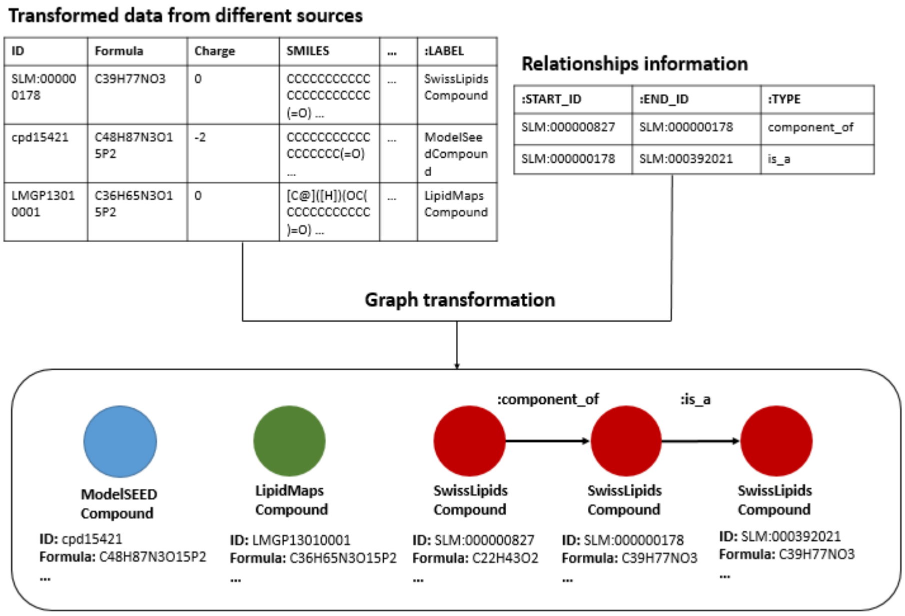
This information was then converted into a directed graph (Definition 3).

Each node of the database was tagged with one and only one label (Definition 4). The same logic was applied to the edges (Definition 5).

The set of labels $\mathcal{E}_{\mathcal{L}}$ in Definition 5 is the one considered for this stage of BOIMMG's pipeline. However, other labels will be considered in further stages of the present work.

Furthermore, instances of each node will be defined as Objects (Definition 6). Accordingly, each Object will contain properties and an internal identifier. As edges connect nodes, Objects will be associated by Relationships (Definition 7). Each Relationship will also have a specific set of properties, two and only two different associated Objects.

The final result of the data loading and graph transformation stage will be a directed graph, in which each node will correspond to either a generic or structurally defined lipid. Moreover, each edge will correspond to a biochemical relationship between two lipids (Figure 18).



Transformed data from different sources

ID	Formula	Charge	SMILES	...	:LABEL
SLM:000000178	C39H77NO3	0	CCCCCCCCCCCCCCCCCCCCCC(=O) ...	...	SwissLipidsCompound
cpd15421	C48H87N3O15P2	-2	CCCCCCCCCCCCCCCCCC(=O) ...	...	ModelSeedCompound
LMGP13010001	C36H65N3O15P2	0	[C@]([H])(OC(CCCCCCCCCCC)=O) ...	...	LipidMaps Compound

Relationships information

:START_ID	:END_ID	:TYPE
SLM:000000827	SLM:000000178	component_of
SLM:000000178	SLM:000392021	is_a

Figure 18: Graph transformation from two files. One with information about compounds retrieved from different sources and the other with information about the relationships between them.

The graph presented in Figure 18 is partially disconnected. Only the compounds retrieved from SLM are connected, as only the SLM's hierarchy and relationships were imported, not including the Model SEED and LMSD's compounds. Hence, the disconnected information will be conducted in the next section.

3.2.3 Data integration

A consensual node for each hierarchy's entity was defined to facilitate the posterior knowledge expansion and data integration. Therefore, all nodes tagged with the label "SwissLipidsCompound" were cloned and tagged with the new label "Compound", which will be, from this time forth, referred to as the "consensual node". All nodes associated with SLM data were cloned first, as they already were included in the hierarchy. Accordingly, all relationships associated to a given cloned compound will be transferred into its respective "consensual node". The "consensual node" will represent the integrated information regarding the different sources.

Thus, the data integration was conducted by matching the chemical structures retrieved from different sources. Correspondingly, the Definition 8 was considered to integrate possible redundant information,. In fact, the InChIKey is a practical line notation suitable for integrating this information, as it contains information about the compound's structure, stereochemistry and protonation in only 27 characters. However, the last character will be ignored as it represents the compound's state of protonation. In other words, a molecule with the same chemical structure but in different states of protonation will be considered the same. In a simplistic sense, the protonation state corresponds to the absence or presence of protons in a given molecule. Hence, in a stoichiometric sense, two structurally and stereochemically equal molecules in different protonation states will only differ in the total number of hydrogen atoms. Hence, there is no point on considering those two compounds as different entities, as the only implication in GSM models will be the potential unbalance of reactions. The balance problem will be solved by adding or removing protons from the correct reactions' side (either in the reactants or products side).

While this is true for structurally defined compounds, it is not for generic ones, as they do not have a well-defined structure, InChIKey nor InChI. Consequently, for these cases, the integration was performed based on the canonical SMILES. As described in the previous section, these were generated by *rdkit*. Thus, one and only one possible SMILES representation for each different generic compound is expected. Correspondingly, a case sensitive match test regarding SMILES strings was performed (Definition 9) to determine which generic compounds have the same structure.

Model SEED and LMSD compounds were then integrated and associated with the already created "consensual nodes". On the other hand, the ones that did not match with any of the SLM compounds were cloned into another "consensual node". As for the latter, these other "consensual nodes" will be disconnected from the previously loaded hierarchy. Correspondingly, the integration of the disconnected "consensual nodes" into the SLM hierarchy was conducted. This integration will be discussed in the following subsection.

3.2.4 Data integration in SLM's hierarchy

As for the relationships integration, the Algorithm 1 was implemented. For the "is_a" relationship type, two main filters were considered: one for the backbone and another for the side chains. A substructure match test was performed against all classes of interest in BOIMMG's database to assess the backbone presence. On the other hand to guarantee the correct side chain requirement, a filter was implemented for checking whether the number of R_{groups} in the abstract class is equal to the number of side chains in the structurally defined compound. When it came to establishing correct "component_of" relationships, the side chains were extracted from the compounds and matched by structural similarity with the database's existing structures. It is worth noting that all these processes were powered by *rdkit*.

Algorithm 1: Establishment of relationships between LIPID MAPS, ModelSEED compounds and BOIMMG compounds

```
Input: $\mathcal{R}$ as the universal set of relationships in the database
       $C$ such as $C \subseteq \mathcal{O}$ is a list of previously chosen abstract compounds
       $coreSMILES$ is a SMILES string representing the $core$ of the components
Output: No output is returned, as the $\mathcal{R}$ set will be mutated
begin
    $Dict_{database} \leftarrow$ DatabaseReader() // It returns the dictionary function with $\mathcal{K}$:
      database identifiers; and $\mathcal{V}$: SMILES strings
    for $c \in C$ do
        $F \leftarrow$ SubstructureMatchTest($c_{smiles}$, $Dict_{database}$) // It checks whether there are
          compounds in the database that possess the same backbone as the compound $c$.
          It returns $F \subseteq \mathcal{O}$
        for $f \in F$ do
            $S \leftarrow$ GetSideChains($f_{smiles}$, $c_{smiles}$) // It removes the core of $f$ and returns
              the set of side chains
            $\sigma_{sidechainsViability} \leftarrow$ SideChainsCheck($S$, $f_{smiles}$, $coreSMILES$) // checking
              whether the number of $R_{groups}$ is equal to the number of $sidechains$
            if $\sigma_{sidechainsViability}$ then
                $\pi(\langle f,\ i,$ "is_a" , $\emptyset$ ,$c\rangle,\mathcal{R})$
                for $s \in S$ do
                    $p \leftarrow$ JoinCoreWithSidechain($s$,$coreSMILES$) // function that joins the
                      identified side chains with the component $core$, it returns the
                      component object $p \in \mathcal{O}$
                    $\pi(\langle p,\ i,$ "component_of", $\emptyset,\ f\rangle,\mathcal{R})$
                end
            end
        end
    end
end
```

3.3 Semi-automated knowledge expansion

The second step of BOIMMG's pipeline is the generation and expansion of biochemical knowledge. The present module can capture functional and structural relationships between complex chemical species and their biosynthetic precursors.

In the previous chapter, relevant relationships aiming at metabolic modelling were described. Structural relationships were mainly set and integrated into BOIMMG's database. Regarding the "component_of" type relations, they were also loaded from the SLM database and inferred for LMSD and ModelSEED lipids. However, the "precursor_of" type of relationships do not exist in any other database.

Regarding the lack of annotated information about electron-transfer quinones, their hierarchy and relationships were constructed semi-automatically resorting to few methods (enumerated in the next subsection). Nevertheless, for other lipids, a novel semi-automated module was developed to extract biosynthetic information from generic pathways and reactions. These methodologies will be thoroughly described in the following subsections.

3.3.1 Electron-transfer quinones

As shown in Chapter 2, the information about the structurally defined chemical species of electron-transfer quinones is scarce in biochemical databases, when compared to literature [80]. Thus, the annotation of such species and the relationships between them are relevant.

A few methods (enumerated below) were used and an extensive manual curation were performed to tackle such a problem. The algorithm represented in Figure 19 describes the computational method used to generate the electron-transfer quinone relationships. The workflow can be defined as follows:

1. Provide the KEGG pathway identifier of quinones biosynthesis;
2. Set a SMILES generic representation for each generic compound;
3. Search for similarity against ModelSEED compounds database;
4. Automatically generate functional and structural relationships;
5. Manual curation;

After the automatic generation of the relationships, extensive data curation was conducted. The curation was performed pathway-by-pathway, starting by correcting errors produced by the automatic method. Then, generic structures were generated for each biosynthetic precursor, conjugated bases or acids without representation in ModelSEED. As afore-mentioned, the quinones and quinols reported in the literature are not completely represented in reference biochemical databases. For this reason, defined structures such as those reported in the literature were generated and the relationships with their respective abstract compounds were set. Furthermore, each conjugated base or acid relationships were manually established for the abstract compounds and computationally inferred for the structurally defined chemical species.

3.3.2 Other Lipids

The other lipids representation problem was solved similarly, however, due to the considerably higher number of structurally defined chemical species, an automatic method was here developed. Correspondingly, the following considerations must be taken into account:

- For each lipid there is a respective generic biosynthetic pathway;
- Each reaction of those pathways can be represented by a SMARTS notation readable by *rdkit* modules;

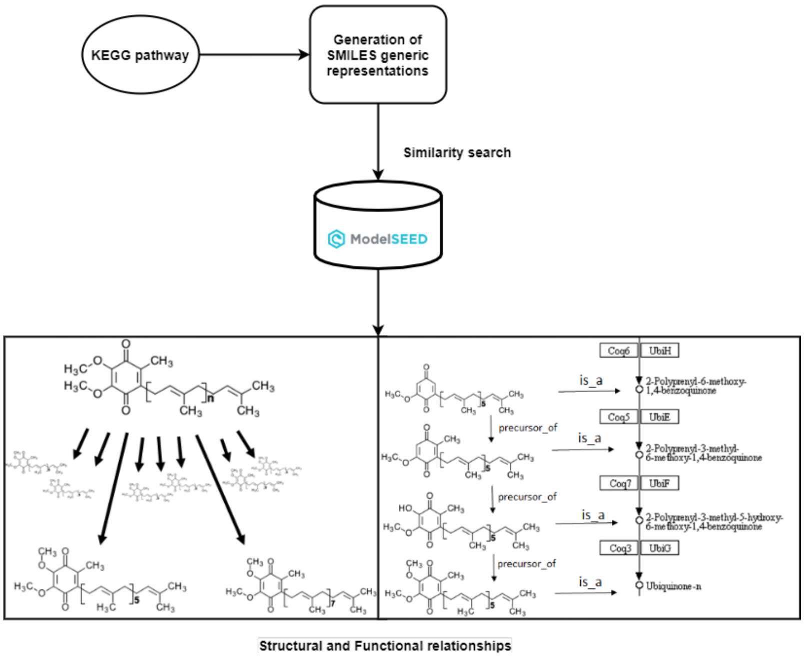

KEGG pathway
Generation of SMILES generic representations
Similarity search
ModelSEED
Coq6
UbiH
is_a
2-Polyprenyl-6-methoxy-1,4-benzoquinone
precursor_of
Coq5
UbiE
2-Polyprenyl-3-methyl-6-methoxy-1,4-benzoquinone
Coq7
UbiF
2-Polyprenyl-3-methyl-5-hydroxy-6-methoxy-1,4-benzoquinone
Coq3
UbiG
Ubiquinone-n
Structural and Functional relationships

Figure 19: This fluxogram depicts the computational method used to aid the establishment of the electron-transfer quinones relationships. This method received the quinones biosynthesis KEGG pathway. Then, a SMILES generic representation was set for each generic compound. Afterward, an iterative similarity search was conducted against the ModelSEED compounds database. This process was performed aiming to assess which compounds were structurally similar to the previously generated structure. Finally, the relationships were set based upon the backbone and sidechain structures, and in the role of each compound in the biosynthetic pathway.

- Each generic entity in the biosynthetic pathway has structural descendants in BOIMMG's hierarchy;

The knowledge expansion and generation workflow is depicted in Figure 20. Essentially, this operation encompasses three computational components: the Network Handlers (NH), the Relationships Generator (RG) and BOIMMG's internal database. Moreover, the SMARTS reaction representations are requested from the user to achieve higher accuracy and to avoid a high number of false positives. Ultimately, the RG algorithm aims to generate missing biosynthetic precursors and establish relevant relationships between the different classes and instances of lipids.

The NH module is composed by operations which cope with the MetaCyc information such as the pathways ontology and templates. Fundamentally, NH converts the pathway ontology into a directed graph where the nodes and edges represent the pathways and their relationships, respectively. In this ontology, the instances (nodes without predecessors) correspond to specific pathways, whereas the classes (nodes with predecessors) are entities that represent various pathways. For instance, the "Phospholipids Biosynthesis" is a class of pathways, representing all the characterized biosynthetic pathways of phospholipids.

Moreover, NH converts the pathway template into a directed graph. The reactions are represented with nodes and the pathway sequence of reactions is represented with directed edges.

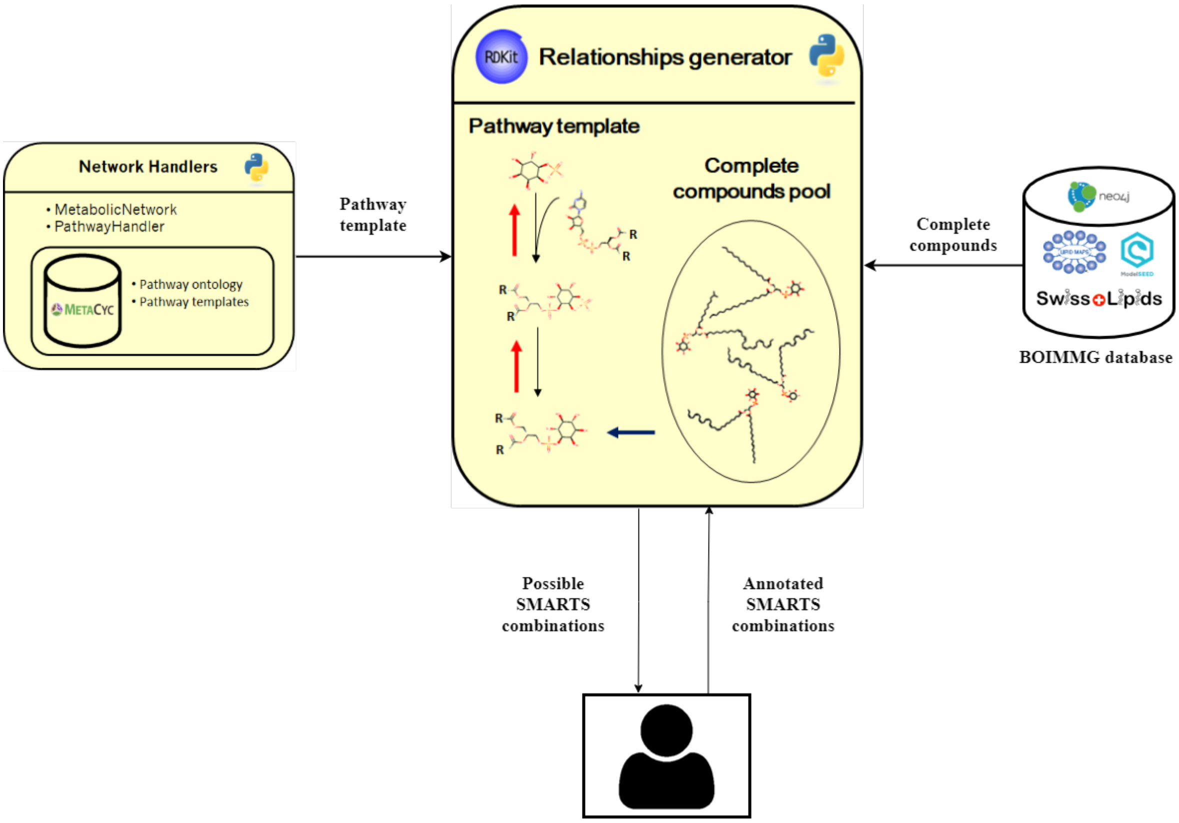


Figure 20: This flowchart depicts the general interaction between the main computational components and the user to capture valuable relationships between compounds. The NH provide the pathway templates to the RG algorithm. Herein, all combinations of the SMARTS notation for all the pathway reactions are provided to the expert. Then, the user selects the correct notations and provide these to the RG. The RG requires all the hierarchy descendants of the biosynthetic target of the selected pathway. BOIMMG's database provide these descendants. The RG algorithm starts to navigate through the biosynthetic pathway generating the chemical structures of each descendant's biosynthetic precursor (red arrows). If these already exist, "precursor_of" relationships will be set. Otherwise, the new precursors chemical structure will be added to the database and relations will be established (blue arrow).

The RG is a module that includes an operation to establish biosynthetic relationships. It receives as input the specific biosynthetic pathway template, the annotated reactions' SMARTS, and the structural children of the compound to be synthesized.

The workflow starts with selecting a specific class of compounds to be synthesized, then the NH seeks the ontology instances of that given class (Pathway ontology in Figure 21). Afterwards, the RG iterates over the chosen pathway's reaction set and generates the possible reactions' SMARTS (SMARTS 1,2,n in Figure 21). At this moment, the user must be able to choose the correct SMARTS. Then, the annotated SMARTS are stored and processed by the RG algorithm (Figure 22).

After gathering all the pathway's and related reactions' information, the RG algorithm starts by reversing the sequence of pathway reactions. Then, it converts the pathway reactions into virtual ones where the products are converted into reactants and vice-versa. Subsequently, having the first generic reactant of the first virtual reaction in the reverse pathway (the biosynthetic target), the next step is to load all the hierarchy's structural descendants (structurally defined chemical species) from BOIMMG's database.

Finally, the algorithm iterates over the descendants set and navigates through the reverse pathway graph. For each node (reaction), the algorithm uses the annotated transformations to predict chemical structures utilized in that reaction. If the predicted structure exists in the database, the ”precursor_of” relationship is established. Otherwise, the predicted chemical structure is stored and the respective relationships are set. Cardiolipin is shown as an example in Figure 22.

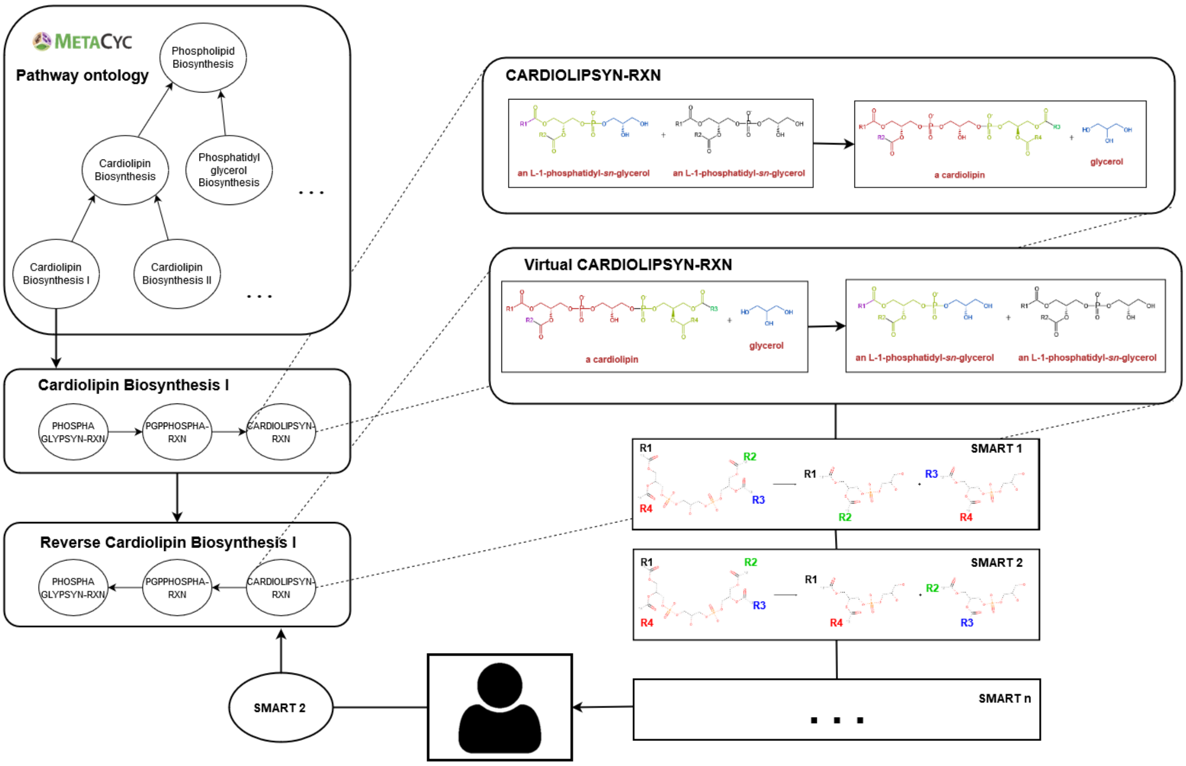

METACYC
Pathway ontology
Phospholipid Biosynthesis
Cardiolipin Biosynthesis
Phosphatidyl glycerol Biosynthesis
...
Cardiolipin Biosynthesis I
Cardiolipin Biosynthesis II
...
Cardiolipin Biosynthesis I
PHOSPHA GLYPSYN-RXN
PGPPHOSPHA-RXN
CARDIOLIPSYN-RXN
Reverse Cardiolipin Biosynthesis I
PHOSPHA GLYPSYN-RXN
PGPPHOSPHA-RXN
CARDIOLIPSYN-RXN
SMART 2
CARDIOLIPSYN-RXN
an L-1-phosphatidyl-sn-glycerol
an L-1-phosphatidyl-sn-glycerol
a cardiolipin
glycerol
Virtual CARDIOLIPSYN-RXN
a cardiolipin
glycerol
an L-1-phosphatidyl-sn-glycerol
an L-1-phosphatidyl-sn-glycerol
SMART 1
R1
R2
R3
R4
SMART 2
SMART n
...

Figure 21: This figure illustrates an example of how different combinations of reaction SMARTS are provided. Firstly, a given MetaCyc pathway ontology instance is selected and converted into its reverse pathway. Each reaction is transformed into virtual reactions where each reactant is converted into product and vice versa. Then, the algorithm generates a set of SMARTS combinations for each pathway reaction. Finally, the user selects and provides to the RG algorithm the correct reaction SMARTS.

3.4 Integration in GSM models

In this section, the integration of the previously generated information in GSM models will be described. A completely automatic method was developed. The main goal was to provide

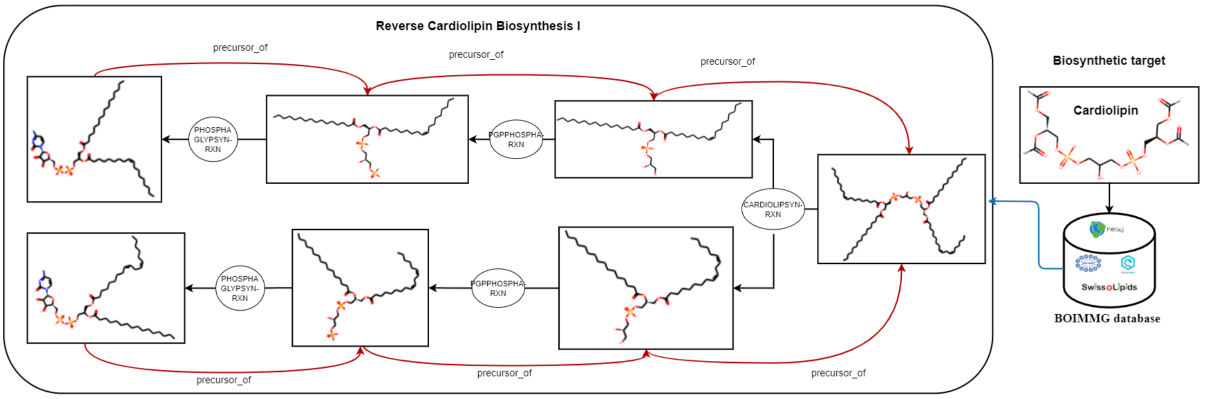


Figure 22: This scheme shows an example of how the RG algorithm works for cardiolipin. It starts by receiving all the biosynthetic target structural descendants from BOIMMG's database (blue arrow). Subsequently, the previously generated virtual reactions, using the SMARTS transformations provided by the user, will generate the biosynthetic precursors' chemical structure. The red arrows represent the established biosynthetic relationships.

a computational tool to generalize, granulate, and integrate complex structurally defined chemical species.

3.4.1 Software overview

Data integration in GSM models respects the workflow presented in Figure 23. The computational method to automate this process was developed on top of *COBRApy* and encompasses five components: the Representation Problem Solver (RPS), the Model Mapper (MM), the Network Modifier (NM), the Revisor, and BOIMMG's database.

The RPS is the module that handles the user's requests and the GSM model. It aims to solve two different problems, which will be referred to, from this time forth, as Simple Representation Case (SRC) and Redundant Representation Case (RRC) problems.

SRC problem will be assigned to those classes of chemical species that one and only one of its kind occurs in living organisms (e.g. *Escherichia coli* uses the ubiquinone with 8 isoprene units long). In order to address the representation problem of those chemical species, it would be necessary to swap one compound into another, including their biosynthetic precursors.

On the other hand, the RRC problem will be assigned to those classes of chemical species in which more than one of its kind occurs in a living organism simultaneously (e.g. phosphatidylcholines with different acyl side chains can occur in the same organism). The strategy employed to tackle this problem encompasses the granulation of generic species into the designated structurally defined compounds. Accordingly, new reactions must be generated towards the biosynthesis of these compounds and their biosynthetic intermediates.

The MM identifies all the metabolites in the model and establishes links with several biochemical databases including BOIMMG's. Other modules will then use the metabolites' map to add or modify reactions and metabolites in the model correctly.

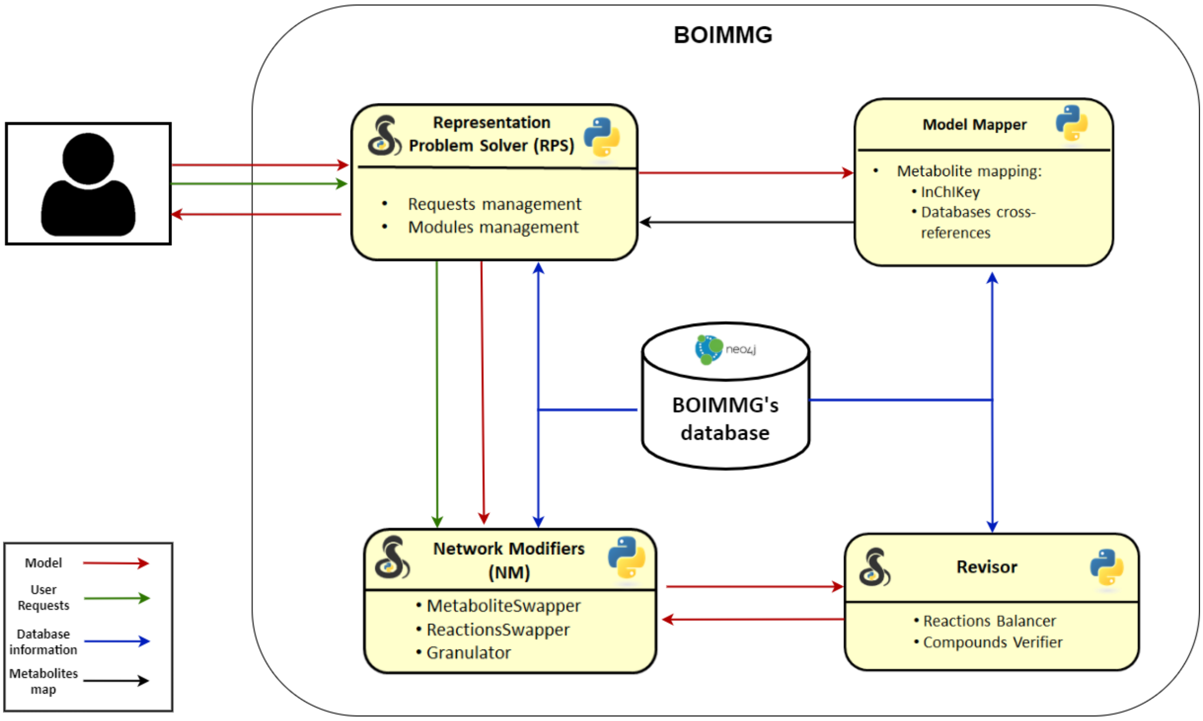


Figure 23: General scheme representing the software architecture of the BOIMMG's framework. The software is composed of five main components: NM, Revisor, RPS, MM and BOIMMG's database. The workflow starts with the user requests and the submission of a GSM model. Then, the model is mapped by the MM regarding their InChIKey and databases' cross-references. Eventually, the NM will swap or granulate the metabolites from their generic version into their complete (structurally defined) version, vice-versa or from the defined into another defined version. Moreover, all the reactions associated to the swapped or granulated metabolites will be changed to their correct format. Afterwards, the Revisor will balance the altered reactions if necessary. The modified model is, then, retrieved to the user.

The NM component is composed by the Metabolite Swapper (MS), the Reactions Swapper (RS) and the Granulator. The primary function of MS is to swap metabolites from one chemical species into another. For instance, if one wants to generalize all the quinones, the MS will swap all the requested metabolites and their biosynthetic precursors to the respective generic entity. The metabolite swapping process includes the compound's annotations, formula, charge, and identifier, depending on the model's database format. Whereas the RS will change all the reactions associated to the swapped metabolites. Depending on the database's format, the RS will search for the new reaction's database identifier (ID) and its respective Gibbs free energy. While these components are used to address the SRC problem, the Granulator is used to solve the RRC. The Granulator's function is to ensure that all molecular species and associated biosynthetic reactions are introduced in the model.

The Revisor provides an analysis of the compounds' representation in the GSM model and the biochemical consistency of each modified or added reaction. It checks whether the reactions are balanced and if the compounds' biosynthetic precursors are correctly annotated and assigned to a specific reaction. This component is important as, after swapping metabolites, there could be reactions that are unbalanced. Also, their products and reactants could not make sense from a representational point of view. Accordingly, this module uses an algebraic method to balance reactions, which will be described in the following subsections.

The general workflow can be followed up in the Figure 23. Depending on the user's scope, BOIMMG can perform different tasks. Nevertheless, BOIMMG requires either three or four inputs, depending on the type of representation problem: the GSM model, the actual target to granulate or swap, the chemical species present in the model, and the target components (e.g. fatty acids). The latter is only required for RRC type of problem.

The user performs the swapping or granulation request, and a GSM model is submitted. The swapping or granulation process is performed considering all the information available in BOIMMG's database. After swapping or granulating all the compounds and reactions, the Revisor module will revise the model. The altered model is then returned to the user.

3.4.2 Model Mapper

The Model Mapper aims at identifying all the compounds in the model, considering their chemical structure and their database links. It iterates all over the model's metabolite set and analyses their assigned annotations.

These annotations must contain databases' identifiers to determine whether the compound is available in BOIMMG's database. Therefore, this module generates two maps: one to determine which compounds are in BOIMMG's database, and another to get the correspondence between the model compounds and biochemical databases (Algorithm 2)

This mapping will be particularly relevant in the swapping and granulation process, as these tasks will highly depend on the chemical structure's representations present in the model. Thus, determining which metabolites are represented in the model is essential.

Furthermore, this module will help to update the created model maps whenever a compound is swapped or generated.

3.4.3 Network Modifiers

The NM are a crucial component of BOIMMG, as they provide the means to change the compounds' and reactions' representation based on the information retrieved from BOIMMG's database. The NM are composed by the MS, the RS, and Granulator classes. The first class instantiates the RS, as only the reactions with swapped metabolites are considered to be swapped.

Three metabolite swapping types are considered for the implementation of the two swapping classes: the 0, 1 and 2 (as depicted in Figure 24). The type 0 represents a swapping from one metabolite into another with a common structural parent. On the other hand, the type 1 and 2 involves the swapping of one metabolite, their biosynthetic precursors and their conjugated bases and acids (Algorithm 3). The type 1 encompasses only the swap between structurally

Algorithm 2: Algorithm to map the metabolites present in the model.

```
Input: $model$ such as $model$ is the GSM model
Output: $Dict_{modelMap}$ and $Dict_{boimmgModelMap}$ as dictionary functions to consult each of
  the metabolites in the model.
begin
    $M \leftarrow model_{metabolites}$
    $Dict_{modelMap} \leftarrow Dict[\emptyset \rightarrow \emptyset]$
    $Dict_{boimmgModelMap} \leftarrow Dict[\emptyset \rightarrow \emptyset]$
    for $m \in M$ do
        $A \leftarrow m_{annotations}$
        for $a \in A$ do
            $Dict_{boimmgModelMap}(a) \leftarrow m$
            $o_i \leftarrow$ GetBOIMMGCompoundByDBLink($a$) // function that returns the
              BOIMMG identifier of the queried compound if it exists, otherwise returns
              null, such that $o \in \mathcal{O}$
            if $o_i \neq null$ then
                $Dict_{boimmgModelMap}(o_i) \leftarrow m$
            end
        end
    end
    return $\langle Dict_{modelMap}, Dict_{boimmgModelMap} \rangle$
end
```

defined metabolites with the same backbone. Lastly, the type 2 represents the swap between a defined metabolite with its structural parent.

Regarding the type 1 and 2 metabolite swapping, all the information from biosynthetic precursors and conjugated base and acid are extracted from BOIMMG's database.

The Reactions Swapper analyses the swapped metabolites in the changed reactions and changes their format. It searches for reactions in KEGG, BiGG and ModelSEED to assign their identifiers and aliases to the new model reaction. If the reaction does not exist in any of the aforementioned databases, an arbitrary identifier is assigned. Herein and for the granulation process, it is assumed that the enzymes catalyzing such reactions are promiscuous for a specific group of chemical species (ubiquinones, menaquinones, and so forth).

As for the Granulator's implementation, the general algorithm is described below (Algorithm 4). Herein, a set of components and a generic compound are received as inputs. Starting from this, the Granulator will extract all the structurally defined molecules belonging to the generic compound class received as input. Furthermore, from these molecules only those containing the requested structural components will be selected. These compounds will be pushed into a stack. Then, the granulated biosynthetic pathway will be built, iteratively. Each extracted lipid will be added to the network in each iteration, starting from granulating all its structural parent's associated reactions.

The granulation is performed straightforwardly. Considering that the reaction's metabolites are generic, their respective structural children will replace each of them. This information is

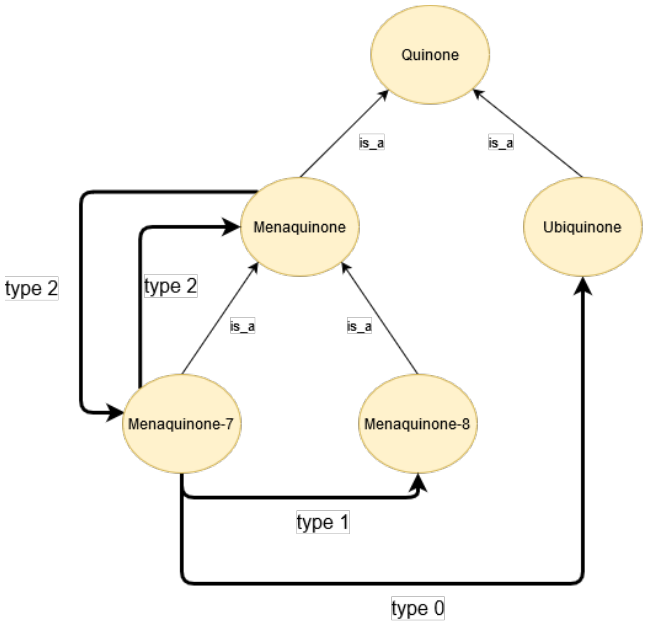


Figure 24: This figure depicts the types of possible swaps. The type 0 occurs from one compound to another with a common parent, or a structural parent to a child. Whereas the type 1 and type 2 aim at swapping all the biosynthetic precursors, conjugated bases and acids. Nevertheless, type 1 involves the swapping between two structurally defined chemical species. On the other hand, the type 2 swap encompasses changing one structurally defined compound to its structural parent, as well as the other way around.

extracted previously from the database (”is_a” and ”precursor_of” relationships). Then, in each iteration, new compounds (the target’s precursors) will be added to the metabolic network. Thus, in order to trace back the biosynthetic pathway of the initially requested compound, their precursors are pushed into the stack.

In the next iteration, a compound is popped from the stack and the process is repeated. This process goes on and on until the stack is empty.

3.4.4 Revisor

The Revisor’s function is to revise the compounds representation in the model and ensure that all the swapped and granulated reactions are balanced.

The correct compounds’ representation is checked using the maps previously generated, as these have information on the model’s metabolites. Furthermore, this component will check whether the reactant’s side chain is transferred within a specific reaction. If not, the reaction is removed from the model. Otherwise, the reaction is kept.

Algorithm 3: Algorithm to swap metabolites. (*) The condition is variable considering the type of swap. Type 1: *HasCommonStructuralParentWith*. Type 2: *IsStructuralChildOf*.

```
Input: $replacer\_id$ is the compound identifier of the metabolite that will replace the other in
  the swapping process.
        $model$ is the GSM model
        $precursorsInModel$ are the biosynthetic precursors of the metabolite to be swapped
        $conjugatedAcidAndBaseInModel$ are the conjugated acid and base of the
  metabolite to be swapped
Output: This method will not return anything, rather, it will mutate $model$
begin
    $P \leftarrow$ GetPrecursorsFromDatabase($replacer\_id$)
    $A \leftarrow$ GetConjugatedAcidAndBaseFromDatabase($replacer\_id$)
    $C \leftarrow \emptyset$
    for $m \in precursorsInModel$ do
        for $p \in biosyntheticPrecursors$ do
            if HasCommonStructuralParentWith($m$,$p$) (*) then
                ChangeMetaboliteFormat($m$,$p$) // this method will mutate $m$, changing
                  the metabolite format to $p$'s format (formula, mass, annotation, etc)
                $\pi(m, C)$
            end
        end
    end
    for $m \in conjugatedAcidAndBaseInModel$ do
        for $c \in conjugatedAcidAndBase$ do
            if HasCommonStructuralParentWith($m$,$c$) (*) then
                ChangeMetaboliteFormat($m$,$c$)
                $\pi(m, C)$
            end
        end
    end
    ChangeReactions($C$) // this method will change all the reactions' format regarding
      the metabolites that were swapped
end
```

Reaction's balance

An algebraic approach was considered to balance the either granulated or swapped reactions. Let one consider the following unbalanced reaction's equation:

$$\mathrm{H_3PO_4} + \mathrm{CaCO_3} \longrightarrow \mathrm{Ca_3(PO_4)_2} + \mathrm{H_2CO_3} \tag{3}$$

Now, using it as a skeletal chemical equation, let one consider X such that X is a column vector and $X = \{x1, x2, ..., xn\}$, where xi is the i^{th} stoichiometric coefficient, such that $xi \geq 1$. This leads to the following equation:

$$x1\,\mathrm{H_3PO_4} + x2\,\mathrm{CaCO_3} \longrightarrow x3\,\mathrm{Ca_3(PO_4)_2} + x4\,\mathrm{H_2CO_3} \tag{4}$$

Algorithm 4: This algorithm aims at generating granulated reactions with granulated compounds from a generic biosynthetic pathway.

```
Input: $components$ such that $components$ is the list of the requested structural components
       $parent$ such that $parent$ is the structural parent that is going to be granulated
Output: This algorithm will return a set of new granulated reactions
begin
    $S \leftarrow$ RequestStructurallyDefinedLipids($components$,$parent$) // it returns all the
      structurally defined compounds that are structural children of $parent$ and whose
      components are present in the set $components$
    $R \leftarrow \emptyset$
    while $S \neq \emptyset$ do
        $l \leftarrow \varphi(S, -1)$
        $p \leftarrow$ GetStructuralParent($l$)
        $R \leftarrow$ GetReactions($p$) // get all the reactions that uses a given metabolite, it
          returns the set of reactions
        $V \leftarrow$ GetPrecursorsFromDatabase($l$) // it returns the set of precursors of the
          compound $l$
        for $r \in R$ do
            $g \leftarrow$ GranulateReaction($r$, $l$, $v$) // It derives the generic reaction $r$ to a
              granulated reaction (in accordance with Definition 18). It returns the
              reaction $g$.
            $N \leftarrow$ GetReactants($g$)
            $\pi(g,\ R)$
            $S \leftarrow S \cup N$
        end
    end
    return $R$
end
```

The Dalton's requirement must be satisfied to each coefficient: mass and atom's conservation law.

Then, the reaction's matrix A can be built using the Definition 20. In this case, the matrix would be formulated as represented right below the definition.

Definition 20. Reaction Matrix
Considering the Equation 4, the $m \times n$ matrix A such that m is the number of rows and n is the number of columns will hereby represent the reaction's matrix. Moreover, by definition, each A's value a_{ij} represents the number of atoms of each chemical element i in each molecule j.

$$\mathbf{A} = \begin{array}{cccc|c} H3PO_4 & CaCO_3 & Ca_3(PO_4)_2 & H_2CO_3 & \\ 0 & -1 & 3 & 0 & Ca \\ 0 & -1 & 0 & 1 & C \\ -4 & -3 & 8 & 3 & O \\ -3 & 0 & 0 & 2 & H \\ -1 & 0 & 2 & 0 & P \end{array}$$

Now, if the Equation 5 is considered, one can derive it into an homogeneous system of five equations such as the one in Equation 6.

$$AX = 0 \ (null\ column\ vector) \tag{5}$$

$$\begin{aligned} Ca : x1 &= 3x3 \\ C : x1 &= x4 \\ O : 3x1 + 4x2 &= 8x3 + 3x4 \\ H : 3x2 &= 2x4 \\ P : x2 &= 2x3 \end{aligned} \tag{6}$$

Lastly, this system of equations is solved by extending and using the *Sympy* linear equations' solver modules [94]. The ultimate result is a set of stoichiometric coefficients, which are assigned to the respective reaction, afterwards.

3.5 BOIMMG's evaluation and validation

3.5.1 Knowledge expansion evaluation

The knowledge expansion stage will be evaluated based on whether a set of established relationships can generate new reactions. Moreover, the number of not established relationships will be further discussed. The assessment methods will be enumerated in this section.

Capacity of generating new reactions

This first metric will evaluate whether each established relationship is capable of generating balanced reactions.

MetaCyc generic reactions were used for this end. Each reaction's reactant and product will be replaced by their respective structural children, respecting the previously established biosynthesis relationships. Moreover, the resultant reaction is to be balanced by the algebraic method described in section 3.4.4. The resultant reaction will then be defined as in Definition 18.

Condition 1. Generating new reactions - Condition 1
Considering the generic reaction $r1$, where $S1$ is the set of generic reactants of $r1$ and $P1$ is the set of generic products of $r1$, $r1'$ will be the new complete reaction derived from the reaction $r1$, such that $S1'$, and $P1'$ are the reactants and products of $r1'$, respectively. This condition is true if and only if $\varphi(S1', i)$ is the structural child of $\varphi(S1, i)$, and $\varphi(P1', j)$ is the structural child of $\varphi(P1, j)$, such that $i \in \{x \in \mathbb{N} : x \leq |S1|\}$ and $j \in \{y \in \mathbb{N} : y \leq |P1|\}$.

Condition 2. Generating new reactions - Condition 2
A new complete reaction will be generated if and only if it is balanced according to the algebraic method described in section 3.4.4.

Herein, the wrongly established "precursor_of" relationships per reaction are defined as those that do not respect Conditions 1 and 2. Accordingly, the bad relationships per reaction can be identified considering the following:

- If, for each evaluated generic reaction, there is, at least, one relationship whose origin or target's structural parent is not any of the generic reaction's reactants or products, respectively;
- The generated complete reaction cannot be balanced by the algebraic method described in section 3.4.4.

Moreover, the number of unestablished relationships was quantified. Unestablished relationships are defined as in Definition 21.

Definition 21. Unestablished relationship per product
Considering P the set of generic products of a given generic reaction r, the number of unestablished relationships will be determined by the number of structural children of each element of P that has no relationship associated to r.

Hence, 70 generic reactions extracted from MetaCyc were evaluated. The percentage of complete reactions wrongly derived from a generic reaction was defined by the following: the number of wrongly generated reactions divided by the total number of possible reactions. Moreover, the number of unestablished relationships was also evaluated. The percentage of unestablished relationships was defined by the following: the number of unestablished relationships divided by the number of potential targets.

3.5.2 BOIMMG's data integration in GSM models

Regarding this topic, two different cases were defined: SRC and RRC. The SRC's case study was based on either generalizing or granulating electron-transfer quinones. While for the RRC, only the granulation was performed and applied to glycerolipids and glycerophospholipids. As mentioned in section 3.3.1, the solution to address SRC type of cases encompassed swapping one version of a given chemical species into another. Whereas to solve RRC a more complex solution was employed (described in section 3.3.3.).

To validate BOIMMG's data integration for the SRC problem, the metabolite and reaction set of two different models (the *E. coli K-12 MG1655*'s iML1515 model [95] and the *S. cerevisiae*'s iMM904 [96]) were compared, before and after being either generalized or granulated. Though from different types (gram-negative bacteria and yeast, respectively), these two models were selected because *E. coli* uses an ubiquinone-8, whereas *S. cerevisiae* uses ubiquinone-6. Consequently, a generalization should be able to leverage the overlap between these two models regarding ubiquinones. Although their biosynthetic pathways are different, few reactions are shared.

On the other hand, for the RRC problem, the *E. coli* iJR904's metabolic network [1] was granulated to include structurally defined lipids. This granulation occurred receiving as input the biosynthesis targets (Cardiolipins and Acyl phosphatidylglycerol) and five components: myristic acid (14 carbons), myristoleic acid (14 carbons and a double bond), palmitate (16 carbons), palmitoleic acid (16 carbons and one double bond), and (Z)-11-octadecenoic acid (18 carbons and one double bond). Moreover, the granulation was configured to build lipids with only one type of fatty acids as components of their side chains. However, it is also possible to perform the granulation of more complex lipids (mixing the side chains).

Afterwards, the granulated model was compared to its iteration iAF1260b [2, 3], which includes structurally defined lipids. Moreover, for each added compound present in the biomass reaction, the capacity of carrying flux was evaluated using Biological networks In Silico Optimization (BioISO)'s web-service [97] at https://bioiso.bio.di.uminho.pt/, after adding reactions that would link the model network to the generated one.

BioISO [97] can determine whether the selected model's reactions are carrying flux when maximized or minimized. Also, it allows for tracking errors in the metabolic network that impair the synthesis of the reactants and products of a given reaction.

Model comparison

For model comparison, Algorithm 5 was used both for the comparison of the metabolites and reaction sets.

It is worth noting that although the reaction set comparison algorithm is equal to the Algorithm 5, it has an additional method of comparison if no annotations are found equal. This additional method performs a comparison between all the metabolites present in the two reactions. Protons are added either to the reactants or products side to circumvent the fact that some reactions could present the same metabolites in different protonation states. If the set of metabolites is equal, the reaction is considered as equal as well.

Algorithm 5: Algorithm for metabolite comparison between two GSM models.

```
Input: $model1$ such that $model1$ is the first model
       $model2$ such that $model2$ is the second model
Output: $e$ such that $e$ is the total number of equal metabolites
begin
    $M1 \leftarrow model1_{metabolites}$
    $M2 \leftarrow model2_{metabolites}$
    $e \leftarrow 0$
    $m1 \in M1$
    $A1 \leftarrow m1_{annotations}$
    for $m2 \in M2$ do
        $A2 \leftarrow m2_{annotation}$
        $\sigma_{found} \leftarrow false$
        for $a2 \in A2$ do
            for $a1 \in A1$ do
                if $a1 = a2$ then
                    $\sigma_{found} \leftarrow true$
                end
            end
        end
        if $\sigma_{found}$ then
            $e{+}{+}$
        end
    end
    return $e$
end
```

3.6 Web-service implementation

A dockerized web application was implemented to provide BOIMMG's information and services online. BOIMMG's framework can be divided into two complementary modules: the database and the automatic integration in GSM models. Hence, the web-service was divided into two modules: the navigation and submissions modules. The *back-end* was essentially implemented in *Django* and *Flask*, whereas the *front-end* was implemented in Javascript, Cascading Style Sheet (CSS) and HyperText Markup Language (HTML). This general architecture is illustrated in Figure 25 and will be briefly explained next.

The navigation module complemented the GSM models integration, allowing users to visualize eventually newly introduced structures. Accordingly, this module was implemented on top of *Django*'s framework, ensuring project's scalability, robustness and easy access to BOIMMG's database. Herein, *rdkit* was used to depict all the compounds' structures.

The submissions module was developed to provide BOIMMG services. In turn, the user's submissions are handled in a *Flask* application. Due to *Flask*'s flexibility, a queue system of submissions was implemented, allowing the service to process each request sequentially.

Each HyperText Transfer Protocol (HTTP) request (either navigation or services), is received and processed by each module, separately. The responses are then retrieved and rendered in the *front-end*.

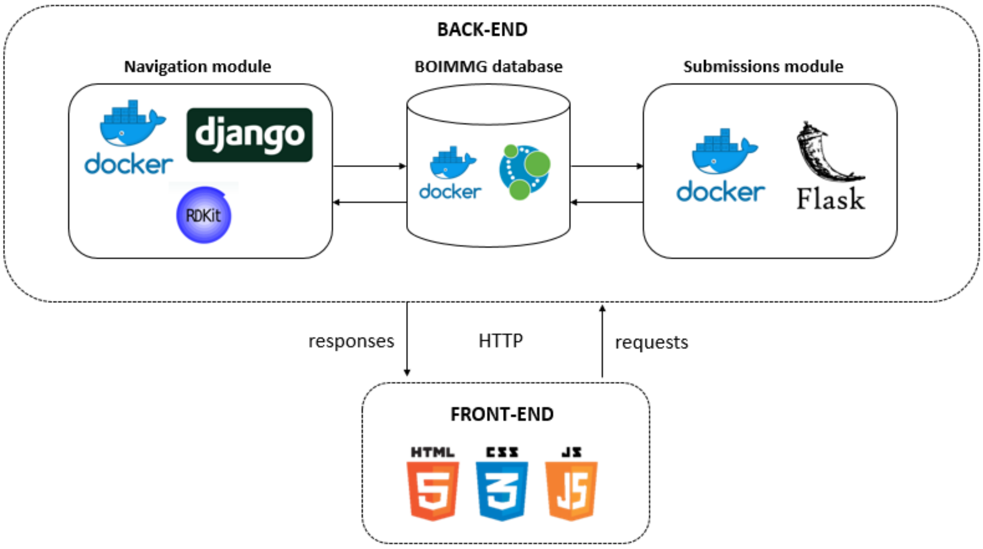


Figure 25: BOIMMG's web-service holistic architecture. The *back-end* is composed by two main modules: navigation and submissions. The navigation module is implemented on top of *Django* and allows the navigation in BOIMMG's database. On the other hand, the submissions module is implemented on top of *Flask* and provides BOIMMG services. Although the two are interacting with the database, the purposes are different. The submissions module contains BOIMMG services, which are database-dependent. In contrast, the navigation module's purpose is specifically to render and provide database's information. The HTTP requests are handled separately by each module, enhancing the two modules' clear functional separation. After retrieving HTTP responses, the result is rendered in the *front-end*, which is implemented in HTML, CSS and Javascript.

3.6.1 Navigation module implementation

The navigation module is composed by the following sub-modules: URL patterns, Views renderer and the Database access layer (Figure 26). The former will be responsible for recognizing the HTTP requests and then passing it to the Views renderer. Therein, the request is processed and the database access is established. In the Database access layer queries are made to the database, and the results are processed and retrieved back to the Views renderer. Finally, the results are rendered in the *front-end*. It is worth noting that all compounds' images are being generated using *rdkit*.

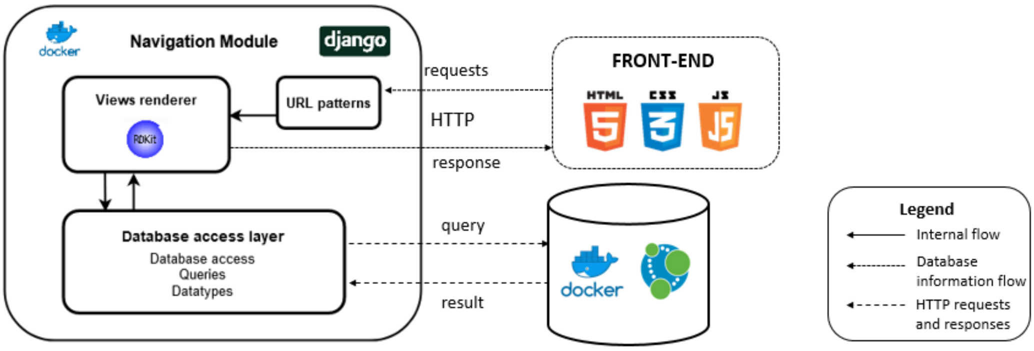


Figure 26: Navigation module architecture. The URL patterns' submodule matches the requests, which are then processed by the Views renderer. Specific information regarding compound's structures and relationships is requested to the database access layer. Afterwards, the database is queried, and the results stored in their respective datatypes containers. The Views renderer will then use *rdkit* for the depiction of compounds, as well as render their information in an HTML page.

3.6.2 Submissions module

The submissions module engine is on top of a queue system that includes two main entities and three types of submission ”states”. The first entity is the module management, whose function is to handle HTTP requests from the outside, ensuring that all the submissions will be processed, and bring the results back to the user. The module workers are Docker containers with BOIMMG services embedded. They can run BOIMMG services, and retrieve information about the processing status. Furthermore, there are three main submission status: the ”submission”, ”processing”, and ”results”. These status are folders that contain information about each submission.

The *modus operandi* of this module is depicted in Figure 27. It starts with the submission of a GSM model and several parameters. These are then stored in the ”submissions” folder, while an available worker is requested to operate. When at least one is available, all the files are transferred to the ”processing” folder. Then, as the files are ready to be processed and at least one worker is available, the submission starts to be processed. When the worker generates the results, they are transferred into the ”results” folder. The management system is always checking whether new results are available in the ”results” folder, in which case, the results are sent back to the user. Lastly, it is worth noting that the workers module can deploy as many workers as required to deal with submission requests, which is particularly relevant if the user-base increases significantly, as more workers can be assigned to manage more requests.

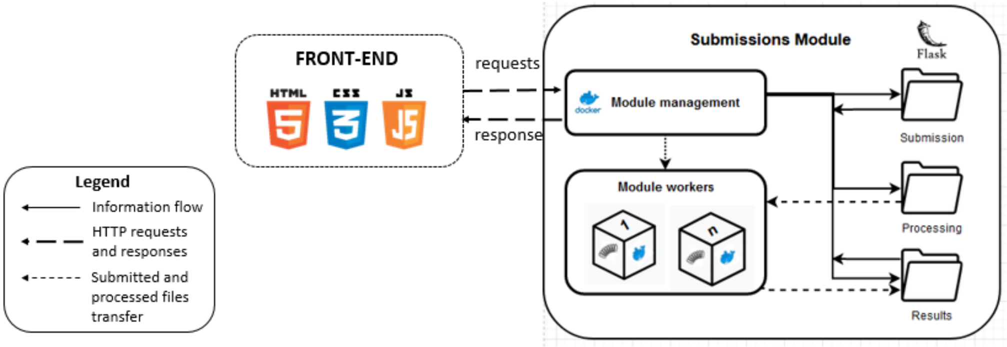


Figure 27: Submissions module's *modus operandi*. It starts with the submission of the requested input (GSM model and several parameters). On the right is shown the Submissions module's main entities and submission states: Module management, and Module workers; "submission", "processing", and "results" states, respectively. Whereas on the left is depicted the user and the *front-end* implementation. The main interactions between all the dockerized entities and the user are herein illustrated.

4

RESULTS AND DISCUSSION

In this chapter, the results regarding the three phases of BOIMMG's pipeline will be presented and discussed. The databases' integration requires evaluating certain parameters, such as the number of metabolites that were present in different databases. Furthermore, the knowledge expansion will be assessed by the number of correctly characterized reactions. Lastly, the integration in GSM models will be validated with the comparison of different already curated models.

4.1 Databases integration assessment

The first task of BOIMMG's pipeline is the integration of the databases. Correspondingly, the information present in SLM, LMSD and ModelSEED was integrated into a graph-based database using *Neo4j*. It is worth noting that these reference databases include information on both generic and structurally defined lipids. Although LMSD contains generic compounds' information, it does not include their structure representation (e.g. SMARTS) in the TSV file. For this reason, only the generic compounds from SLM and ModelSEED were integrated into BOIMMG's database.

4.1.1 Generic compounds integration

Regarding the generic compounds, the integration involved using the canonical SMILES of each compound, which resulted in 71 compounds between SLM and ModelSEED, whereas 876 exclusively extracted from SLM and 203 exclusively extracted from ModelSEED (Figure 28).

The overlap between both databases focused mainly on the most well-annotated lipids such as the glycerophospholipids, few sphingolipids and glycerolipids classes. These results suggest that these classes are the most commonly used, revised, and the biosynthetic pathways are well-characterized in GSM models.

The Model SEED's unique generic lipids were mainly prenol lipids, such as quinones and their biosynthetic precursors, as these are incompletely and poorly characterized in the SLM database.

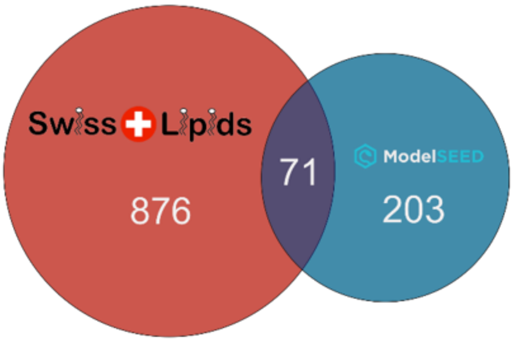


Figure 28: Venn diagram with the number of generic compounds that have been integrated from each database.

On the other hand, as expected, the SLM's unique classes were mostly the ones further away from the SLM's hierarchy leaves, as ModelSEED does not provide an ontology or hierarchy of compounds.

4.1.2 Structurally defined compounds integration

Concerning the integration of structurally defined compounds, only 579 were found in all the integrated databases. On the other hand, 12555 compounds exclusively from LMSD and SLM have matched. Moreover, 27 were found exclusively in ModelSEED and LMSD, whereas SLM share 219 compounds with only ModelSEED. Lastly, 406 from ModelSEED, 580062 from SLM and 30921 from LMSD did not match with any other compound in any other database. This information is present in Figure 29.

Altogether, the results were the following (Figure 29):

- LIPID MAPS: 30 921
- ModelSEED: 406
- SWISS LIPIDS: 580 062
- LIPID MAPS and ModelSEED: 27
- LIPID MAPS and SWISS LIPIDS: 12 555
- SWISS LIPIDS and ModelSEED: 219
- SWISS LIPIDS, ModelSEED, and LIPID MAPS: 579

Regarding the results presented in Figure 29, it is clear that an effective integration was performed. It shows that there was redundant information across databases; however, simultaneously, unique data was found in all the different sources.

When it comes to the overlap between data retrieved from all databases, the integration occurred mostly in structurally defined glycerophospholipids, fatty acyls, and derivatives (Figure 30).

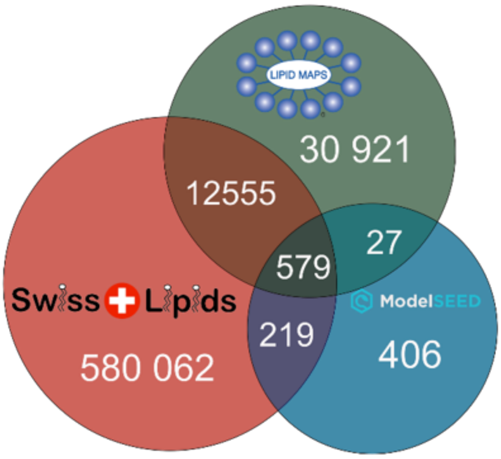


Figure 29: Venn diagram with the number of structurally defined compounds that have been integrated from each database.

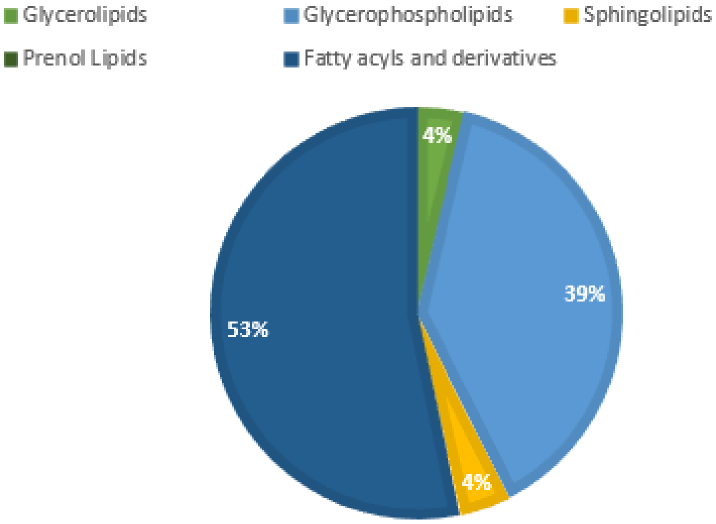


Figure 30: The pie chart shows the percentage of structurally defined lipids with overlap in all databases. Results are shown per class of lipid.

Unsurprisingly, the results for Model SEED's integrated compounds were similar to those in Figure 30. As Model SEED compiles information extracted from GSM models, the fact that most of the overlap occurred in classes such as glycerophospholipids, fatty acyls and derivatives suggests that these are the most widely represented classes of lipids in GSM models.

On the other hand, as shown in Figure 31, an analysis performed over the generated hierarchy revealed that the most common lipids shared by SLM and LMSD belong to glycerolipids and glycerophospholipids' classes.

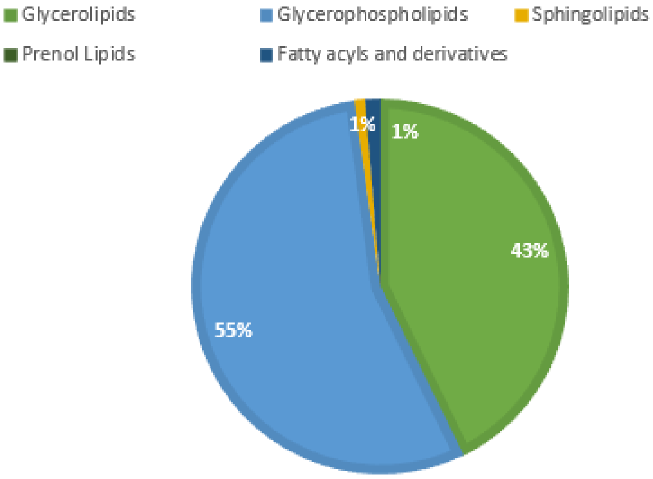


Figure 31: The pie chart shows the percentage of structurally defined lipids with overlap in SLM and LMSD structure database. Results are shown per class of lipid.

4.1.3 Lipids integration in SLM's hierarchy

This stage's last step was to establish "component_of" and "is_a" relationships between LMSD, ModelSEED compounds and the previously loaded SLM's structures. The results of this integration are presented in Table 19.

Table 19: Results of LMSD and Model SEED compounds integration in SLM's hierarchy. This table shows the number of relationships before and after the integration and the total of new established relationships.

Relationship	Before	After	Total new relationships
is_a	481 800	488 549	6749
component_of	1 443 920	1 455 977	12057

As shown in Table 19, the integration of LMSD and ModelSEED's compounds in SLM's hierarchy was successful: 6749 "is_a" and 12057 "component_of" relationships were generated.

Table 20: Integration of LMSD and Model SEED compounds in SLM hierarchy. Number of relationships established between each databases' compound and other already integrated in the hierarchy. This table shows the ones which have been integrated from exclusively one database and the ones present in both.

Relationship	LMSD only	Model SEED only	Both
is_a	6105	607	27
component_of	11 949	103	0

The next step of BOIMMG's pipeline aims at establishing biosynthetic relationships between compounds. As a limited set of lipid subclasses had at least one well-annotated biosynthesis pathway in MetaCyc database, the scope of this integration was limited to several subclasses of glycerophospholipids, few of glycerolipids, sphingolipids and fatty acyls (Table 21). Predictably, the obtained results, expressed in Venn diagram in Figure 29 and Table 20, show that not all LMSD and Model SEED compounds were integrated.

On the other hand, prenol lipids were integrated manually because the SLM's hierarchy lacked information about prenol lipids, mainly on electron-transfer quinones' precursors.

Table 21: Number of integrated subclasses per class of lipid

Main class	Number of integrated subclasses
Glycerolipids	3
Glycerophospholipids	22
Sphingolipids	2
Prenol Lipids	15
Fatty acyls and derivatives	2

4.2 Final database topology and general statistics

The final database topology is the result of the databases' integration and the knowledge expansion process. When transformed into a graph and loaded into the database, several nodes were disconnected. Nevertheless, manual, semi- and fully automated methods were employed in order to curate and integrate all the redundant data. Moreover, as the database and new information was generated, new relationships were created, as well.

Correspondingly, the database final topology will be such as in Figure 32. Basically, the set of edge labels $\mathcal{E}_{\mathcal{L}}$ will now be defined as $\mathcal{E}_{\mathcal{L}} = \{$ "is_db_link_of", "component_of", "is_a", "precursor_of", "conjugated_acid_of", and "conjugated_base_of". $\}$.

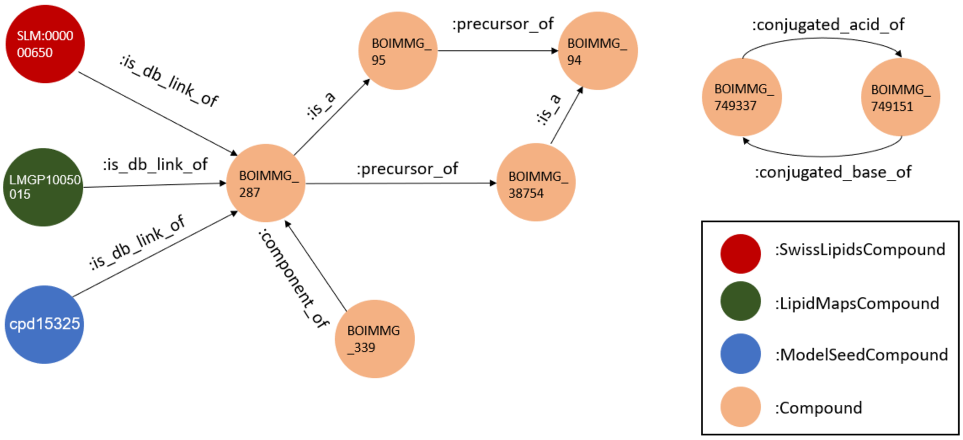


Figure 32: The final topology of the database is shown here. The arrows represent the edges and the circles correspond to the nodes. Each circle colour corresponds to a specific node label. One internal identifier is assigned to each "consensual node" (nodes with the ":Compound" label).

As shown in Figure 32, one internal identifier is assigned to each "consensual node" (nodes with the ":Compound" label). Moreover, the relationships will be represented by edges with specific labels. These labels will only be in the subset of $\mathcal{E}_{\mathcal{L}}$ composed by {"component_of", "is_a", "precursor_of", "conjugated_acid_of", and "conjugated_base_of", "is_db_link_of}. All the previous relationship's types will be crucial for the biochemical complex data integration in GSM models. The importance of each relationship will be now enumerated:

- **:component_of**: These relationships are important to build the structurally defined lipids requested by the user. Receiving as input the produced structural components (e.g. fatty acids), and the structural parent, one can construct the defined structures only consulting BOIMMG's database.
- **:is_a**: These relationships will be crucial both for generalizing and granulating chemical species. The successors in edges with the label ":is_a" allow generalizing one chemical species. On the other hand, the predecessors in the same type of relationship allow granulating chemical species.
- **:precursor_of**: These relationships will be of paramount importance to granulate reactions and characterize complete pathways. Generalizing reactions is substantially easier than granulating (as it only requires using "is_a" relationships). Considering that each generic compound represents a whole set of molecules, a generic reaction's granulation will require both structurally defined compounds and their correctly assigned precursors. The latter condition is addressed by querying this relationships in BOIMMG's database. Moreover, applying graph algorithms such as Depth First Search or Breadth First Search, all the biosynthetic intermediates of a structurally defined lipid can be obtained.
- **:conjugated_acid_of** and **:conjugated_base_of**: These relationships will be particularly relevant when managing the generalization or granulation of electron-transfer chemical species. A reaction that transfer electrons is also a typical conjugated acid and base reaction. As referred in Chapter 2, generalizing or granulating electron-transfer chemical species implies that their conjugated acid or base are also generalized or granulated. This condition can be addressed by querying these relationships in BOIMMG's database.

4.2.1 General statistics

In this section, the general database's statistics will be presented and discussed.

As explained in the previous section, different relationships were generated. Although most of the "is_a" and "component_of" relationships were loaded from the SLM database, others were automatically generated to integrate LMSD and ModelSEED's lipids. Moreover, "precursor_of" relationships are certainly considered the novelty of the present work and represent a relevant feature regarding knowledge expansion, as no database nor ontology includes such information. The number of relationships per type is indicated at Table 22.

As discussed in section 3.3.2, while establishing the "precursor_of" relationships, if the precursors were not listed, their structure would be automatically generated. Figure 33 exhibits a Venn diagram with the number of listed compounds per database and the ones not listed.

The generation of theoretical lipid structures has been described here and elsewhere [9, 98]. These approaches aim at leading to theoretical hypothesis and to the elucidation of lipids'

Table 22: Number of relationships per type in BOIMMG's database

Relationship	Number
is_a	515 527
precursor_of	749 784
component_of	1 509 351
conjugated_acid_of	128
conjugated_base_of	128

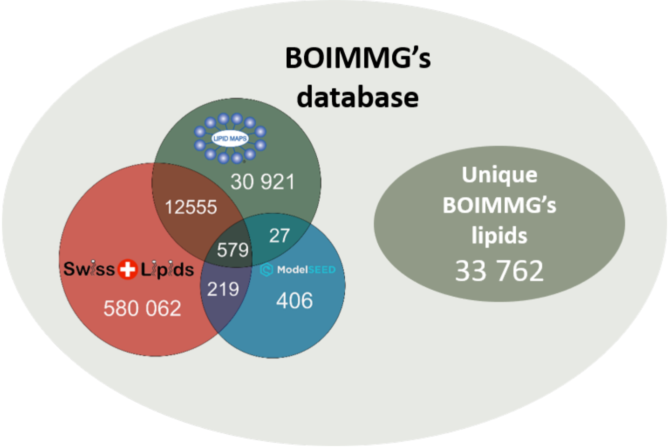


Figure 33: Venn diagram with the integrated and listed lipids as well as the ones not listed (Unique BOIMMG's lipids)

metabolism and biosynthesis. However, it is evident that these methods can be generating structures that do not exist in nature, thus, the set of false positive structures could eventually be large. The SLM's approach [9] generates theoretically feasible lipid structures combining SMILES of chemical substructures. However, their approach do not account for these compounds' biosynthesis, thus several precursors were absent. BOIMMG's approach, while generating valuable knowledge, has contextualized both SLM's theoretical structures and experimentally validated ones within their biosynthetic context. In this context, theoretical reactions and pathways can be enumerated, driving to several hypothesis waiting to be validated.

4.3 Knowledge Expansion

The Knowledge Expansion was the second step of BOIMMG's pipeline, which included the identification of biosynthetic relationships between lipids. This section will include the integrated assessment of the established relationships by evaluating whether the created relationships could generate balanced reactions.

4.3.1 Capacity of generating new reactions

As mentioned in section 3.5.1, 70 generic reactions extracted from MetaCyc were evaluated. The percentage of complete, wrongly derived reactions, was calculated, along with the number of unestablished relationships.

The wrongly generated reactions' and the number of unestablished relationships' percentages are depicted in the blue and orange bars in Figure 34, respectively. The results show that only two generic reactions out of 70 have passed the mark of 10%. Out of these two, only one had a total percentage of wrongly generated reactions of 100 %.

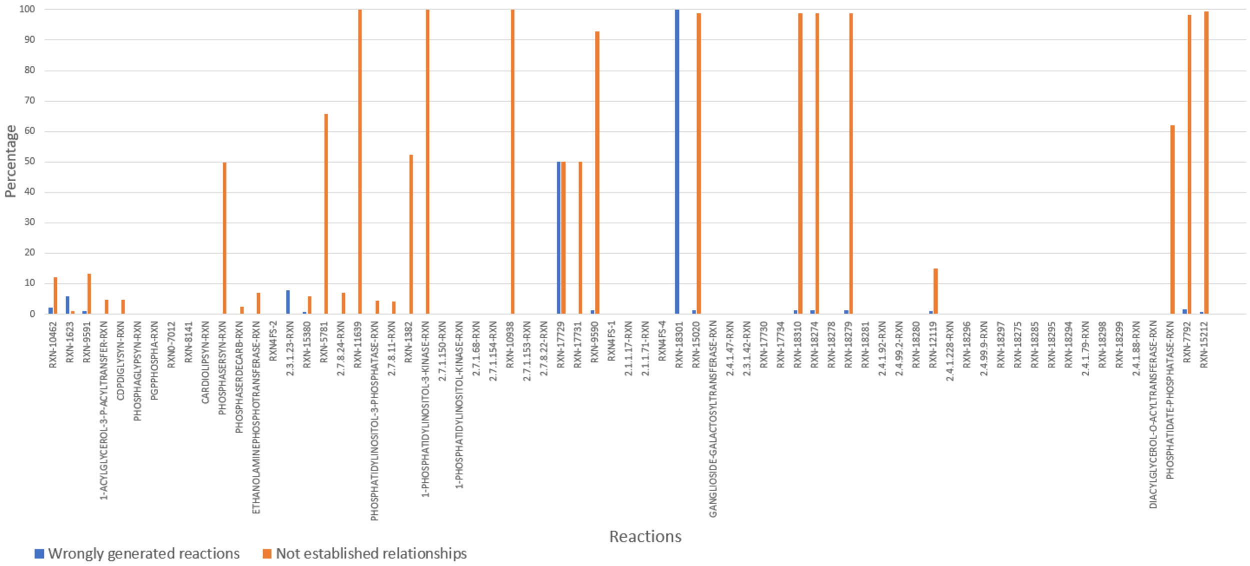


Figure 34: Wrongly generated reactions (blue bars) and unestablished relationships (orange bars) percentage, in which 70 reactions were evaluated to assess the efficacy in the knowledge expansion process. Each bar corresponds to a given generic reaction and shows the percentage of wrongly complete reactions derived from it, taking into account the previously established "precursor_of" relationships.

The percentage of wrongly generated reactions derived from "RXN-18301", and "RXN-17729" was of 100%, as Condition 1 was violated. Hence, all the structurally defined compounds possibly involved in these reactions were not associated with the correct structural parents. The problem resided in the SMARTS transformation definition, which was wrongly associated with either an incorrect generic product or reactant. Correspondingly, all the possible reactions derived from "RXN-18301" and "RXN-17729" were wrongly generated.

On the other hand, the wrongly generated reactions that did not reach the mark of 10 % were also cases in which the Condition 1 was violated, but for a different reason. Most of these reactions have at least two generic reactants. Consequently, if one of these reactants' structural children had no relationship with a structural child of that reaction's product, the creation of the complete reaction would be impaired. In other words, at least one precursor was not related to the reaction's products, compromising the reaction's creation.

Therefore, regarding all the established relationships, the whole set of correctly established relationships could generate balanced reactions (Condition 2 always met). On the other hand, this was not true for the Condition 1, enhancing the need to annotate the hierarchy further.

Concerning the number of unestablished relationships (orange bars in Figure 34), 34 out of the 70 analysed reactions had at least one relationship that was unestablished. This result indicates that several reactions were partially or not processed. A thorough analysis of the reactions with higher percentages on the "not established reactions" bar ("RXN-18279", "RXN-18310", for instance) reveals that either the SMARTS transformation was not correctly formulated or the reaction was not processed, which compromised the establishment of biosynthetic relationships between structurally defined compounds.

The establishment of the relationships was based on the MetaCyc's ontology of pathways. Each instance of the ontology was converted into separate graphs and processed individually. The processing order could have compromised or even ignored several potential relationships. Although allowing to process each pathway sequentially, this order could have ignored reactions or newly generated structures along the Knowledge Expansion process. Further improvements in the NH modules should be employed.

In conclusion, 30 out of 70 generic reactions were characterized entirely by BOIMMG's knowledge expansion method. However, 29 were characterized partially, as though these offered successfully established relationships, other links were missing. Although further annotation will be in all likelihood required, the characterization of unlisted reactions was achieved. Correspondingly, this large set of theoretical feasible reactions can represent a small step further towards the fully characterization of lipid biosynthesis and metabolism. Lastly, these results showed that, for most of the cases, the relationships are correctly established and ready to be integrated in GSM models.

4.4 BOIMMG's data integration in GSM models

This section will focus on the description and discussion regarding BOIMMG's data integration in GSM models. The strategy described in section 3.4.2. was employed to validate BOIMMG's data integration in GSM models.

4.4.1 Simple Representation Case

A comparative analysis was performed to assess the integration of BOIMMG's data in SRC type of cases. The metabolite and reaction sets of two different models (iML1515 [95] and iMM904 [96]) were compared, before and after being either generalized or granulated. Moreover, a descriptive analysis of each model with different types of quinones will be performed.

Generalization of iML1515 model

iML1515 [95] is a GSM model of *Escherichia coli* K-12 MG1655. The electron-transfer quinones available in this model are the ubiquinone and menaquinone with eight repeating isoprene units.

BOIMMG's method was applied to generalize all the electron-transfer quinones and their precursors. Menaquinone-8 and ubiquinone-8 were swapped with their generic version, which will involve modifying their precursors, conjugated acids, and associated reactions. The results of these network modifications are shown in Figure 23.

Table 23: Quinone chemical species before and after BOIMMG's network modification. On the left, the structurally defined compounds and their model identifiers are enumerated, whereas on the right are shown the generic compounds and their identifiers.

Before	Model ID	After	Model ID
Ubiquinone-8	q8	Ubiquinone-n	q
Ubiquinol-8	q8h2	Ubiquinol-n	qh2
3-demethylubiquinol-8	2dmmql8	3-demethylubiquinol-n	C_BOIMMG_749215
2-Octaprenyl-3-methyl-6-methoxy-1,4-benzoquinol	2ommbl	2-polyprenyl-3-methyl-6-methoxy-1,4-benzoquinol	C_BOIMMG_749221
2-Octaprenyl-6-methoxy-1,4-benzoquinol	2ombzl	2-polyprenyl-6-methoxy-1,4-benzoquinol	C_BOIMMG_749335
2-Octaprenyl-6-methoxyphenol	2omph	2-polyprenyl-6-methoxyphenol	C_BOIMMG_749145
2-Octaprenyl-6-hydroxyphenol	2ohph	2-polyprenyl-6-hydroxyphenol	C_BOIMMG_749139
2-Octaprenylphenol	2oph	2-polyprenylphenol	C_BOIMMG_749133
3-Octaprenyl-4-hydroxybenzoate	3ophb	3-polyprenyl-4-hydroxybenzoate	C_BOIMMG_749127
Menaquinol-8	mql8	Menaquinol-n	C_BOIMMG_749097
Menaquinone-8	mqn8	Menaquinone-n	C_BOIMMG_749415
2-Demethylmenaquinone-8	2dmmq8	2-Demethylmenaquinone-n	C_BOIMMG_749498
2-Demethylmenaquinol-8	2dmmql8	2-Demethylmenaquinol-n	C_BOIMMG_749025

The results of the generalization process are shown in Table 23. The new generic metabolite's model identifier is either in BiGG or BOIMMG's format. As the original model is in BiGG format, the swapping process will try to convert the new metabolites into the BiGG format as well. However, when no BiGG compound is found, the default BOIMMG's database format is used. Table 23 shows that the model identifier of almost all generic compounds is in BOIMMG's format. This information indicates that there is an evident absence of generic representations in BiGG's database.

Moreover, it is worth noting that all swapped metabolites are annotated with cross-references and chemical structure representations such as SMILES and InChIKey.

Granulation of iML1515 model

A previous generalization of the iML1515 model was conducted to validate the granulation of electron-transfer quinones. BOIMMG's approach was applied to the generalized model. The generic menaquinone, ubiquinone and their biosynthetic intermediates were swapped with menaquinone-6, ubiquinone-6, precursors, and conjugated acids with six isoprene units in the side chain. Afterwards, the granulated model was compared with the original iMM904 [96] model to assess whether the granulation was correctly performed. iMM904 [96] is a *Saccharomyces cerevisiae* model that contains ubiquinone-6, its conjugated acid, and precursors.

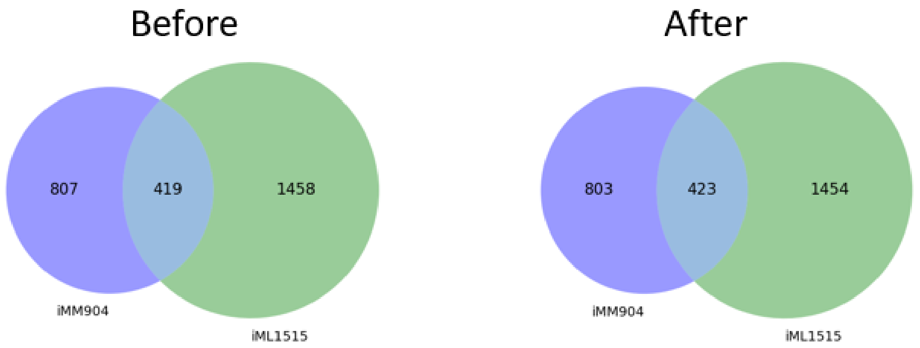


Figure 35: Venn diagram illustrating the intersection between *E. coli* and *S.cerevisiae* model's metabolite set, before and after being subjected to BOIMMG's method.

As shown in Figure 35 and Table 24, four new metabolites were part of the intersection after running BOIMMG.

A larger set of common metabolites would have been expected, as all precursors of ubiquinone-6 should have matched. However, as the original iMM904's *E. coli* model uses the conjugated acids of ubiquinone's precursors, there was no match between these compounds. Showing that the level of detail in the swapping process reaches the conjugated acids and bases of quinone's precursors. For instance, instead of swapping the 3-demethylubiquinol-n (in the granulated iML1414) with 3-demethylubiquinone-6, BOIMMG was able to select the correct chemical species (3-demethylubiquinol-6).

Likewise, the analysis of both reaction sets was performed. As shown in Figure 36, after BOIMMG's granulation, the intersection between both model's reaction sets has slightly enlarged. The intersection's difference before and after granulation was of five reactions. These reactions are all associated with electron-transfers using ubiquinone and ubiquinol. As discussed before, the model uses the conjugated acids of ubiquinone's precursors. Although this does not compromise the operation, the comparison will not match the biosynthesis reactions using ubiquinone and ubiquinol's intermediates. Nevertheless, it is worth noting that BOIMMG correctly swapped the electron-transfer reactions using different quinone species.

Another granulation was performed, in which the generic ubiquinone and their biosynthetic intermediates were swapped with unlisted chemical species (ubiquinone-5, conjugated acid and precursors). The results are shown in Table 25.

Table 24: Metabolite set match between iMM904 model and the previously modified iML1414. On the right, the metabolites in iMM904 are shown, whereas on the left the altered metabolites are presented. The match between each pair of metabolites is also shown in the last column.

iMM904	Model ID	iML1414_modified	Model ID	Match
Ubiquinone-6	q6	Ubiquinone-6	q6	✓
Ubiquinol-6	q6h2	Ubiquinol-6	q6h2	✓
3-demethylubiquinone-6	2hpmhmbq	3-demethylubiquinol-6	C_BOIMMG_749217	X
2-Hexaprenyl-3-methyl-6-methoxy-1,4-benzoquinone	2hpmmbq	2-Hexaprenyl-3-methyl-6-methoxy-1,4-benzoquinol	C_BOIMMG_749223	X
2-Hexaprenyl-6-methoxy-1,4-benzoquinone	2hp6mbq	2-Hexaprenyl-6-methoxy-1,4-benzoquinol	C_BOIMMG_749337	X
2-Hexaprenyl-6-methoxyphenol	2hp6mp	2-Hexaprenyl-6-methoxyphenol	2hp6mp	✓
3-Hexaprenyl 4,5-dihydroxybenzoate	3dh5hpb	-	-	X
3-Hexaprenyl-4-hydroxybenzoate	3ophb_5	3-Hexaprenyl-4-hydroxybenzoate	3ophb_5	✓

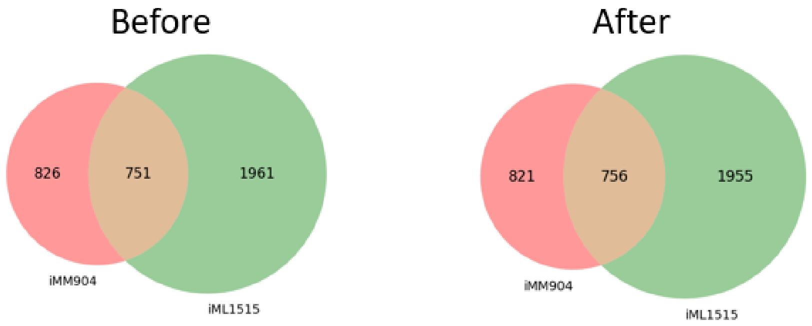


Figure 36: Venn diagram illustrating the intersection between *E. coli* and *S.cerevisiae* model's reaction set, before and after being subjected to BOIMMG's method.

This granulation was correctly performed even considering chemical species unavailable in reference databases. It is important to mention that all new compounds are well annotated with InChIKey, SMILES, and BOIMMG's identifiers, allowing further confirmation of each structure.

4.4.2 Redundant Representation Case

The Redundant Representation Case (RRC) is a problem of representation that encompasses specific classes of compounds. Chemical classes in which more than one of its kind occur at

Table 25: Quinone chemical species before and after BOIMMG's network modification. On the left are the structurally defined compounds with 8 isoprene repeating units and their model identifiers, whereas on the right, the compounds with 5 isoprene repeating units and their identifiers.

Before	Model ID	After	Model ID
Ubiquinone-8	q8	Ubiquinone-5	C_BOIMMG_749196
Ubiquinol-8	q8h2	Ubiquinol-5	C_BOIMMG_749246
3-demethylubiquinol-8	2dmmql8	3-demethylubiquinol-5	C_BOIMMG_749249
2-Octaprenyl-3-methyl-6-methoxy-1,4-benzoquinol	2ommbl	2-pentaprenyl-3-methyl-6-methoxy-1,4-benzoquinol	C_BOIMMG_749250
2-Octaprenyl-6-methoxy-1,4-benzoquinol	2ombzl	2-pentaprenyl-6-methoxy-1,4-benzoquinol	C_BOIMMG_749345
2-Octaprenyl-6-methoxyphenol	2omph	2-pentaprenyl-6-methoxyphenol	C_BOIMMG_749200
2-Octaprenyl-6-hydroxyphenol	2ohph	2-pentaprenyl-6-hydroxyphenol	C_BOIMMG_749199
2-Octaprenylphenol	2oph	2-pentaprenylphenol	C_BOIMMG_749198
3-Octaprenyl-4-hydroxybenzoate	3ophb	3-pentaprenyl-4-hydroxybenzoate	C_BOIMMG_749197

the same time in the same organism are to be assigned to RRC. In fact, all lipid classes were included in this case, except the electron-transfer quinones.

The granulation results are present in Table 26, 27 and in Tables 30,31 in the Supplementary material.

Specifically, the compounds present in the iAF1260b model and their matching species in the iJR904 granulated model version are shown in Tables 26 and 27. A quick analysis of the tables reveals that few metabolites are absent in the granulated iJR904 model, as all lipids were based on the components received as input. Therefore, predictably, the missing chemical species correspond to lipids with 12 and 18 carbon long side chains (without double bonds). Notwithstanding, it was expected that the cardiolipins using either myristic acid (14 carbons) or myristoleic acid (14 carbons and a double bond) as structural components were present in the model. However, neither of these cardiolipins were available in BOIMMG's database, so they were not added to the network.

Manual gap-filling was conducted to evaluate whether each new lipid present in the biomass reaction was being produced. As shown in Table 28, 25 new reactions, derived from five generic reactions of the original model, were inserted. These reactions match either to the first step of the phospholipids synthesis or the decomposition in fatty acids. "PASYN_EC" is originally the reaction that assembles all the Acyl-Acyl Carrier Protein (ACP)s to generate a generic phosphatidate. Whereas "PLIPA1", "LPLIPA1", "LPLIPA2", and "LPLIPA3" are reactions of glycerophospholipids and glycerolipids decomposition into fatty acids. BOIMMG's approach does not allow performing this task automatically. However, little effort has to be made, as the biosynthesis network is already correctly built. In fact, all the biomass lipids are being produced, as demonstrated by the BioISO results in Figures 46, 49, 47 and 48 (Supplementary

Table 26: Lipids present in the biomass equation of iJR904's metabolic model. These are the cardiolipins, and phosphoethanolamine. The "Compound" column represents the structurally defined compound with the following format:<abbreviation>(<number of carbons in the sidechain>:<number of double bonds>). Then, the second column indicates each compound identifier in iAF1260b. The third column indicates whether the compound is present in the altered iJR904 model. The fourth specifies the identifier in the modified model, and the fifth indicates whether the metabolite is being produced after the manual gap-filling.

Cardiolipins (CLPN)				
Compound	iAF1260b ID	In	Granulated Model ID	Produced
CLPN (12:0)	clpn120	X	-	-
CLPN (14:0)	clpn140	X	-	-
CLPN (14:1)	clpn141	X	-	-
CLPN (16:0)	clpn160	✓	C_BOIMMG_322789	✓
CPLN(16:1)	clpn161	✓	C_BOIMMG_341201	✓
CPLN(18:0)	clpn180	X	-	
CPLN(18:1)	clpn181	✓	C_BOIMMG_347940	✓
Phosphoethanolamine (PE)				
Compound	iAF1260b ID	In	Granulated Model ID	Produced
PE (12:0)	pe120	X	-	-
PE (14:0)	pe140	✓	C_BOIMMG_12604	✓
PE (14:1)	pe141	✓	C_BOIMMG_12585	✓
PE(16:0)	pe160	✓	C_BOIMMG_12595	✓
PE(16:1)	pe161	✓	C_BOIMMG_12583	✓
PE(18:0)	pe180	X	-	-
PE(18:1)	pe181	✓	C_BOIMMG_8715	✓

Table 27: Continuation. Lipids present in the biomass equation of iJR904's metabolic model. These are the phosphatidylglycerol, and phosphatidylserine. The "Compound" column represents the structurally defined compound with the following format:<abbreviation>(<number of carbons in the sidechain>:<number of double bonds>). Then, the second column indicates each compound identifier in iAF1260b. The third column indicates whether the compound is present in the altered iJR904 model. The fourth specifies the identifier in the modified model, and the fifth indicates whether the metabolite is being produced after the manual gap-filling.

Phosphatidylglycerol (PG)

Compound	iAF1260b ID	In	Granulated Model ID	Produced
PG (12:0)	pg120	X	-	-
PG (14:0)	pg140	✓	C_BOIMMG_7474	✓
PG (14:1)	pg141	✓	C_BOIMMG_7456	✓
PG(16:0)	pg160	✓	C_BOIMMG_427	✓
PG(16:1)	pg161	✓	C_BOIMMG_7454	✓
PG(18:0)	pg180	X	-	-
PG(18:1)	pg181	✓	C_BOIMMG_7408	✓

Phosphatidylserine (PS)

Compound	iAF1260b ID	In	Granulated Model ID	Produced
PS (12:0)	ps120	X	-	-
PS (14:0)	ps140	✓	C_BOIMMG_729819	✓
PS (14:1)	ps141	✓	C_BOIMMG_729870	✓
PS(16:0)	ps160	✓	C_BOIMMG_729826	✓
PS(16:1)	ps161	✓	C_BOIMMG_729871	✓
PS(18:0)	ps180	X	-	-
PS(18:1)	ps181	✓	C_BOIMMG_729851	✓

material), suggesting that the whole upstream network is also carrying flux, and thus correctly built. Moreover, ″ISA″ reactions were added for each lipid in the biomass equation, but only for flux analysis purposes.

Table 28: Added reactions to gap-fill the granulated model.

Old reaction ID	New Reaction ID	BOIMMG reactant	BOIMMG product
PASYN_EC	PASYN_EC_1	palmACP_c	C_BOIMMG_423_c
	PASYN_EC_2	myrsACP_c	C_BOIMMG_38435_c
	PASYN_EC_3	hdeACP_c	C_BOIMMG_38416_c
	PASYN_EC_4	octeACP_c	C_BOIMMG_38371_c
	PASYN_EC_5	tdeACP_c	C_BOIMMG_38417_c
PLIPA1	PLIPA1_1	C_BOIMMG_427_c	hdca_c + C_BOIMMG_314_c
	PLIPA1_2	C_BOIMMG_7454_c	hdcea_c + C_BOIMMG_16363_c
	PLIPA1_3	C_BOIMMG_7408_c	ocdcea_c + C_BOIMMG_16317_c
	PLIPA1_4	C_BOIMMG_7474_c	ttdca_c + C_BOIMMG_452_c
	PLIPA1_5	C_BOIMMG_7456_c	ttdcea_c + C_BOIMMG_16364_c
LPLIPA1	LPLIPA1_1	C_BOIMMG_314_c	hdca_c
	LPLIPA1_2	C_BOIMMG_16363_c	hdcea_c
	LPLIPA1_3	C_BOIMMG_16317_c	ocdcea_c
	LPLIPA1_4	C_BOIMMG_452_c	ttdca_c
	LPLIPA1_5	C_BOIMMG_16364_c	ttdcea_c
LPLIPA2	LPLIPA2_1	C_BOIMMG_291_c	hdca_c
	LPLIPA2_2	C_BOIMMG_27923_c	hdcea_c
	LPLIPA2_3	C_BOIMMG_27877_c	ocdcea_c
	LPLIPA2_4	C_BOIMMG_27943_c	ttdca_c
	LPLIPA2_5	C_BOIMMG_27925_c	ttdcea_c
LPLIPA3	LPLIPA3_1	C_BOIMMG_223_c	hdca_c
	LPLIPA3_2	C_BOIMMG_6942_c	hdcea_c
	LPLIPA3_3	C_BOIMMG_6896_c	ocdcea_c
	LPLIPA3_4	C_BOIMMG_319_c	ttdca_c
	LPLIPA3_5	C_BOIMMG_6943_c	ttdcea_c

Finally, the comparison between the reactions and metabolites set of the iAF1260b model and the iJR904 granulated model version was performed. The results of these comparisons are depicted in the Venn diagram illustrated in Figures 37 and 38.

In total, 58 metabolites were added to the iJR904 model. The overlap between the metabolite sets illustrated in Figure 37 reveals that 53 matched both the granulated iJR904 and iAF1260b models, leaving five metabolites without any match. This subset corresponds to the chemical species ″Acyl-glycerophosphocholine″, which was granulated into five different lipids. However, this metabolite is not available in iAF1260b model, hence, predictably, no match was found. As enumerated in Tables 26, 27, and 30,31 (Supplementary material), the newly introduced metabolites that matched (subset of 53 metabolites) correspond to the following chemical species:

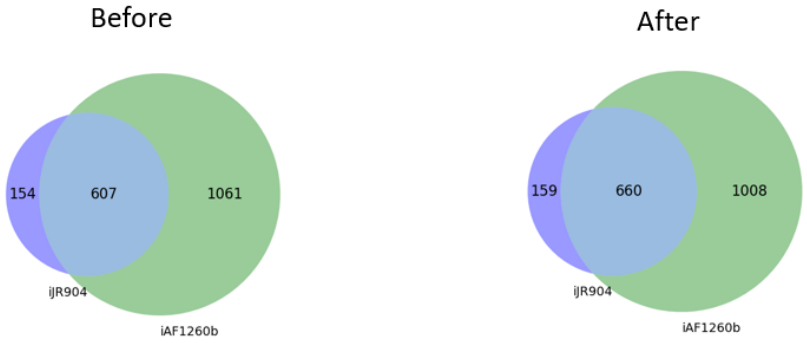


Figure 37: Venn diagram illustrating the intersection between iAF1260b and iJR904 model's metabolite set, before and after being subjected to BOIMMG's method.

- Cardiolipins - 3+
- Phosphoethanolamine - 5+
- Phosphatidylglycerol - 5 +
- Phosphatidylserine - 5+
- Phosphatidylglycerophosphate - 5+
- CDPdiacylglycerol - 5+
- Acyl-glycerophosphoglycerol - 5+
- Acyl phosphatidylglycerol - 5+
- Acyl-glycerophosphoethanolamine - 5+
- 1,2-Diacylglycerol - 5+
- Acyl phosphatidylglycerol - 5+
- Phosphatidate - 5+

These results are expected, as no other generic lipid species was found in the iJR904 model. Accordingly, all lipid species were granulated successfully, and matched other structurally defined lipid species introduced manually (in the iAF1260b model).

The diagram in Figure 38 denotes the overlap between the reaction sets, in which it is possible to verify that 91 new reactions were added to the iJR904 model. As shown in the Venn diagram and Table 29, 53 have matched with the other reaction set. These correspond to reactions present in the biosynthesis of structurally defined lipids (38) and other reactions manually added for gap-filling purposes (15). Moreover, 18 were "ISA" reactions (one for each biomass lipid). Other reactions were added without any match as these were derived from generic reactions not present in iAF1260b model.

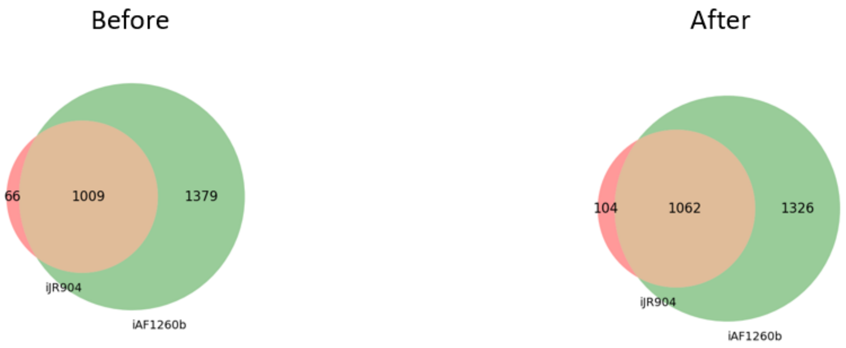


Figure 38: Venn diagram illustrating the intersection between iAF1260b and iJR904 model's reaction sets, before and after being subjected to BOIMMG's method.

Table 29: Number of new reactions in iJR904 model per type

Reaction type	Number	Overlaped	No overlap
Biosynthesis network	38	38	0
Gap Filling	25	15	10
"ISA" reactions	18	0	18
Other	10	0	10
Total	91	53	38

Given these results, it is possible to conclude that BOIMMG's approach was capable of successfully granulating the iJR904 model for chemical species assigned to the RRC.

4.5 Web-service

A web-service was implemented to enable users navigation through BOIMMG's database as well as using BOIMMG's services for swapping and granulating metabolites. The home page is rendered, as shown in Figure 39 and can be accessed in the following link: https://boimmg.bio.di.uminho.pt/. The code is available at https://gitlab.bio.di.uminho.pt/jcapela/boimmg.

4.5.1 Navigation module

The navigation module was implemented as detailed in section 3.6.1. Generally, this module renders two pages: the hierarchy and the lipid page, as depicted in Figure 40 and 41, respectively. The hierarchy web-page contains the lipid hierarchy organized in classes. The main lipid classes are on the left of the web-page and upon clicking on it, the user can navigate through its hierarchy.

Moreover, the lipid page allows getting the information about each lipid's structure, biosynthetic precursors, structural parents, and so forth. Such information can be accessed either by

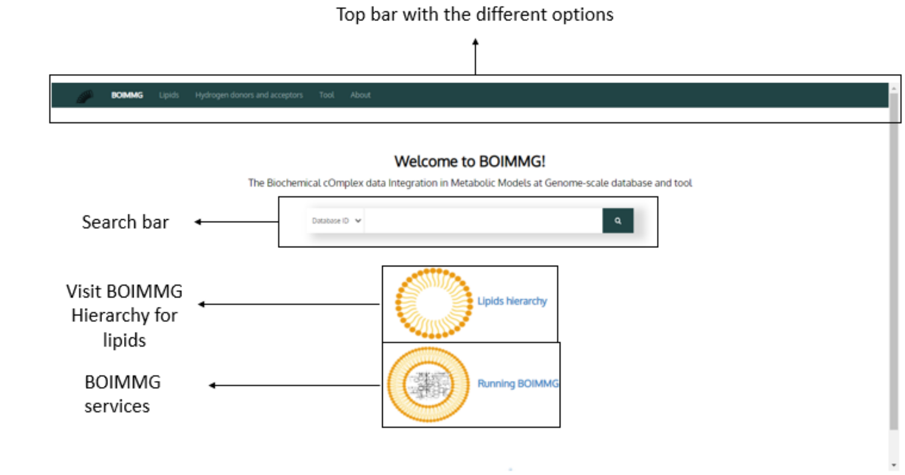

Top bar with the different options
BOIMMG
Lipids
Hydrogen donors and acceptors
Tool
About
Welcome to BOIMMG!
The Biochemical cOmplex data Integration in Metabolic Models at Genome-scale database and tool
Search bar
Database ID
Visit BOIMMG Hierarchy for lipids
Lipids hierarchy
BOIMMG services
Running BOIMMG

Figure 39: Home page from BOIMMG's web-service.

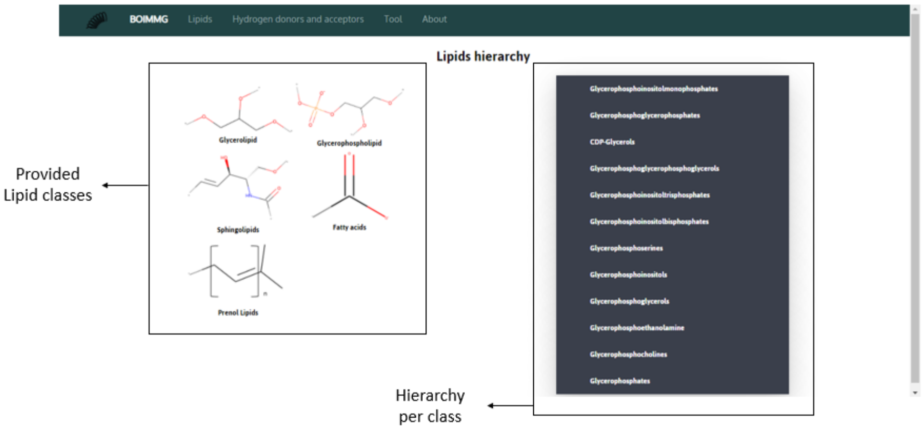


Figure 40: Lipids hierarchy in BOIMMG's web-service.

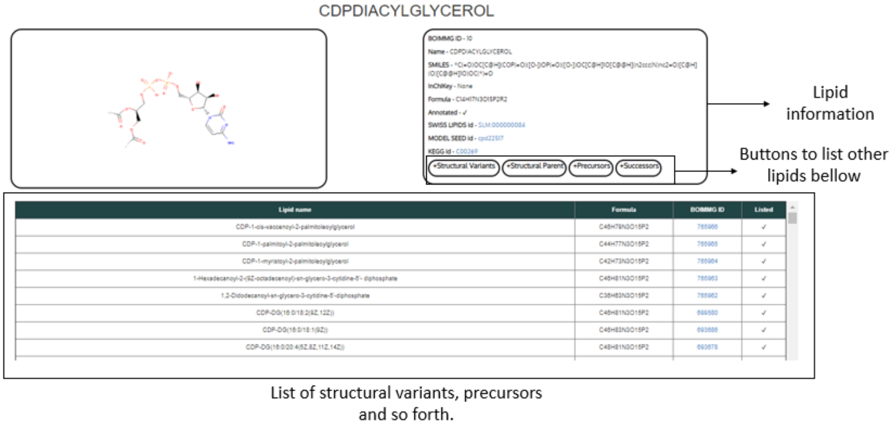
checking the box on the right or by clicking in the buttons. The buttons will then show a table with the requested information.

4.5.2 Submissions module

The submissions module was implemented as described in section 3.6.2. This module renders a menu with the two possible BOIMMG modes: RRC and SRC, as illustrated in Figure 42.

Regarding the SRC mode, the following parameters are requested:

- the quinones in the model;
- the ones to serve as replacers;



BOIMMG ID - 10

Name - CDPDIACYLGLYCEROL

SMILES - *C(=O)OC[C@H](COP(=O)([O-])OP(=O)([O-])OC[C@H]1O[C@@H](n2ccc(N)nc2=O)[C@H](O)[C@@H]1O)OC(*)=O

InChIKey - None

Formula - C14H17N3O15P2R2

Annotated - ✓

SWISS LIPIDS id - SLM:000000084

MODEL SEED id - cpd22517

KEGG id - C00269

Lipid name	Formula	BOIMMG ID	Listed
CDP-1-cis-vaccenoyl-2-palmitoleoylglycerol	C45H79N3O15P2	765966	✓
CDP-1-palmitoyl-2-palmitoleoylglycerol	C44H77N3O15P2	765965	✓
CDP-1-myristoyl-2-palmitoleoylglycerol	C42H73N3O15P2	765964	✓
1-Hexadecanoyl-2-(9Z-octadecenoyl)-sn-glycero-3-cytidine-5'- diphosphate	C45H81N3O15P2	765963	✓
1,2-Didodecanoyl-sn-glycero-3-cytidine-5'-diphosphate	C36H63N3O15P2	765962	✓
CDP-DG(16:0/18:2(9Z,12Z))	C45H81N3O15P2	699580	✓
CDP-DG(16:0/18:1(9Z))	C45H83N3O15P2	693686	✓
CDP-DG(16:0/20:4(5Z,8Z,11Z,14Z))	C45H81N3O15P2	693678	✓

Figure 41: Lipid page with information regarding structure's representations, as well as concerning its precursors, components, and so forth.

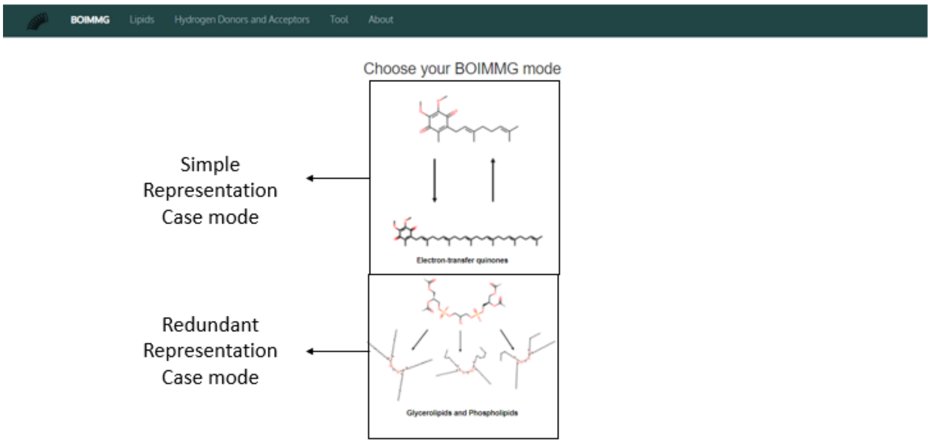

BOIMMG
Lipids
Hydrogen Donors and Acceptors
Tool
About
Choose your BOIMMG mode
Simple Representation Case mode
Redundant Representation Case mode
Electron-transfer quinones
Glycerolipids and Phospholipids

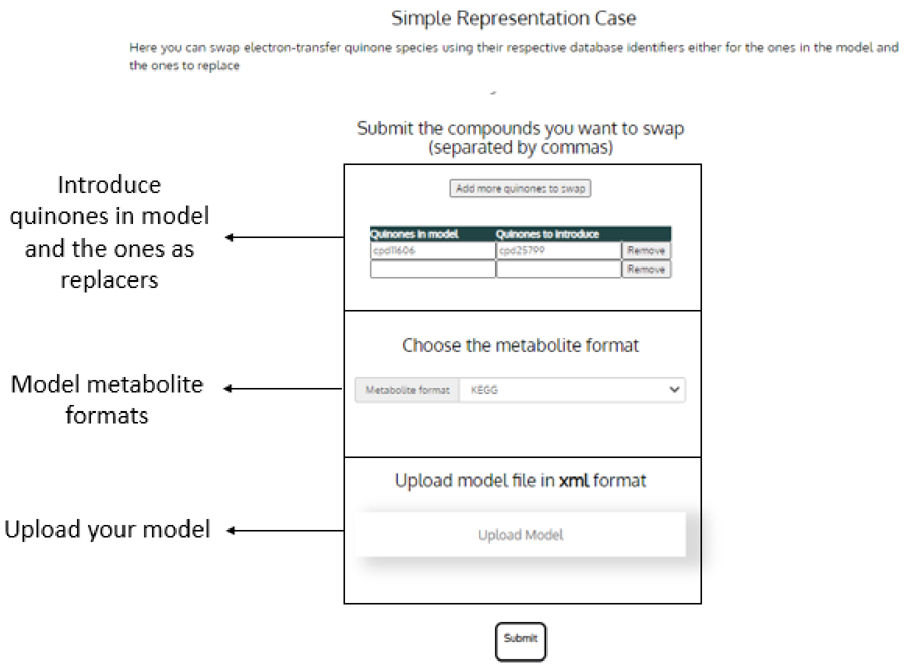
Figure 42: Submissions' module menu.

- the model metabolite formats (ModelSEED, BiGG or KEGG);
- the model in the SBML format;

This description can be followed in Figure 43.

Regarding the RRC mode, the following parameters are requested:

- the generic lipids to be granulated;
- the respective components (fatty acids);
- whether the user wants to mix components or always have the same in each lipid species;
- the model metabolite formats (ModelSEED, BiGG or KEGG);
- the model in the SBML format;



Simple Representation Case

Here you can swap electron-transfer quinone species using their respective database identifiers either for the ones in the model and the ones to replace

Submit the compounds you want to swap (separated by commas)

Quinones in model	Quinones to introduce	
cpd11606	cpd25799	Remove
		Remove

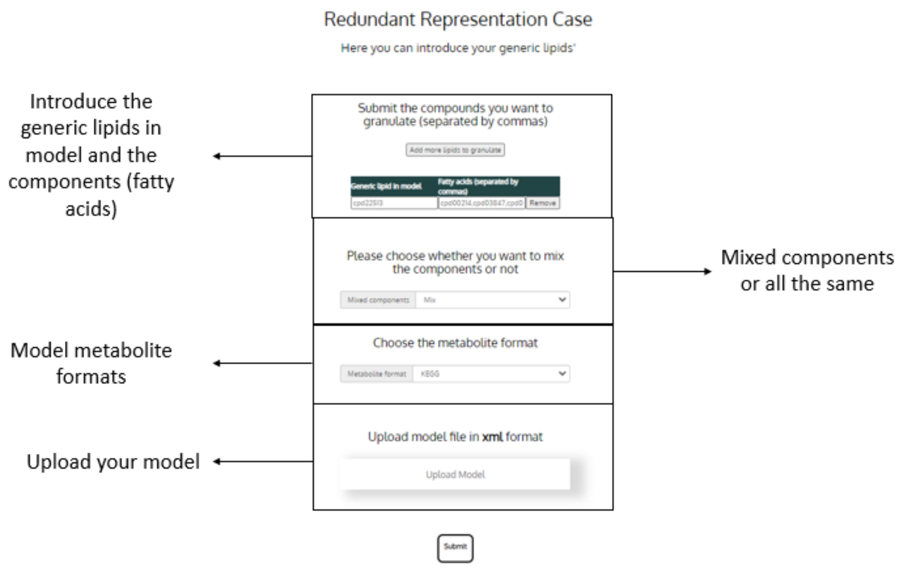
Choose the metabolite format

Metabolite format: KEGG

Upload model file in **xml** format

Introduce quinones in model and the ones as replacers
Model metabolite formats
Upload your model
Add more quinones to swap
Upload Model
Submit

Figure 43: SRC mode parameters and model submission page.



Redundant Representation Case

Here you can introduce your generic lipids'

Submit the compounds you want to granulate (separated by commas)

Generic lipid in model	Fatty acids (separated by commas)	
cpd22513	cpd00214,cpd03847,cpd0	Remove

Please choose whether you want to mix the components or not

Mixed components: Mix

Choose the metabolite format

Metabolite format: KEGG

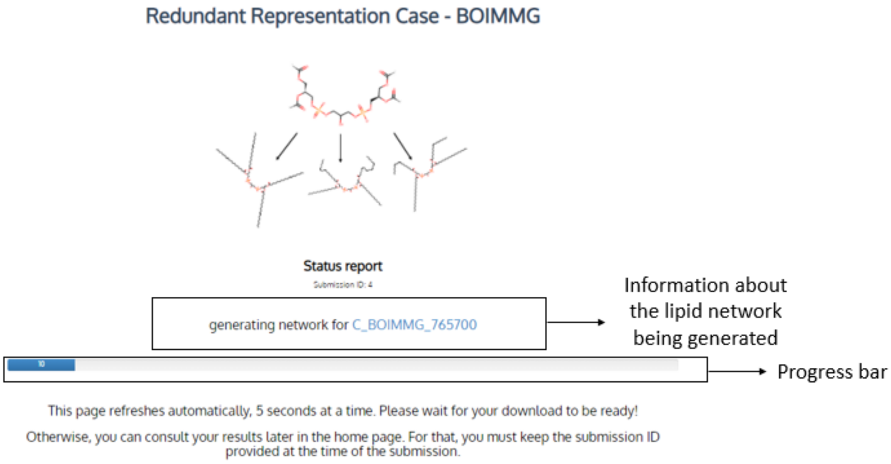
Upload model file in **xml** format

Introduce the generic lipids in model and the components (fatty acids)
Mixed components or all the same
Model metabolite formats
Upload your model
Add more lipids to granulate
Upload Model
Submit

Figure 44: RRC mode parameters and model submission page.

This description can be followed up right in Figure 44.

Furthermore, after submission, a status page is rendered, with information associated with the stage in BOIMMG's granulation or swapping process. A progress bar is shown and updated. In the case of the RRC, one message regarding the generation of a given compound's network will provide a link to the lipid web-page. Information regarding the lipids being introduced will be rendered, as shown in Figure 45. Finally, the altered model is provided in a download page.



Redundant Representation Case - BOIMMG

Status report

Submission ID: 4

generating network for C_BOIMMG_765700

This page refreshes automatically, 5 seconds at a time. Please wait for your download to be ready!

Otherwise, you can consult your results later in the home page. For that, you must keep the submission ID provided at the time of the submission.

Figure 45: Status page rendering information related to the state of the submission.

BOIMMG's web-service is implemented as a user-friendly platform, providing easy access to BOIMMG's swapping and granulation processes. Hence, no programming skills are required for the tool to be run. Moreover, each introduced lipids and precursors can be easily consulted in the navigation module.

5

CONCLUSION

5.1 Conclusion remarks

The Biochemical cOmplex data Integration in Metabolic Models at Genome-scale (BOIMMG) is an open-source, academic, hypothesis-driven framework that aims at revising the lipid metabolism in GSM models. The framework allows gathering lipid-specific information from different sources, expanding the current knowledge in theoretical lipid structures, as well as in the biosynthetic context of the already listed ones. Finally, such information can be applied to complex metabolic networks.

Three different data sources, namely SLM, LMSD and ModelSEED were integrated into a *Neo4j* graph-based database. The SLM's hierarchy was uploaded and is the cornerstone of BOIMMG's hierarchy, as all the unique ModelSEED and LMSD lipids were integrated on top of it. This integration revealed that all databases have differences and unique data, which now enrich BOIMMG's lipid set.

Regarding the knowledge expansion process, over 30 generic reactions were fully and 27 partially characterized. These reactions have resulted from the relationships established between structurally defined compounds. Over 700000 relationships were established using generic reaction information retrieved from MetaCyc (for the great majority of the lipid set), and KEGG (for the electron-transfer quinones). Moreover, more than 30000 new lipid structures were hypothesized in this work, and are waiting for further validation.

Finally, BOIMMG's approach was applied to different GSM models towards the lipids' granulation or generalization. Using BOIMMG's database information, a software capable of granulating and generalizing lipids was developed. As far as the metabolites are correctly annotated with, at least, one cross-reference or InChIKey, both generalization and granulation will run without problems. Otherwise, erroneous results will be generated.

Regarding the electron-transfer quinones generalization and granulation, the iML1515 *E. coli* GSM model was analysed. A comparison between this model and a model of *S. cerevisiae* was performed. Results show that BOIMMG conducted both granulation and generalization successfully.

For the glycerolipids and phospholipids' granulation, a model from *E. coli* (iJR904) with generic lipids was processed by BOIMMG. Afterwards, the altered model was compared to one of its published iterations (iAF1260b), in which the granulation was performed and curated manually. The comparison demonstrated that the granulation was successful, resulting in 53 more matching lipids and 38 more matching reactions. Moreover, besides the correct biochemical set, after a slight manual effort in gap-filling, BioISO's analysis demonstrated that the biomass lipids were being correctly produced.

In conclusion, BOIMMG is a framework capable of generating relevant relationships between complex macromolecules, and of generating theoretically feasible chemical structures in the context of biosynthetic pathways. Furthermore, it provides an automatic tool to integrate these complex data in genome-wide metabolic networks.

5.2 Future perspectives

BOIMMG's database will be updated and annotated regularly, and used as a reference resource for integrating GSM models. Moreover, a chemoinformatics framework may be implemented to assess the viability of the new lipid structures, reactions, and metabolic pathways predicted by this framework. Combining these works, a reference and hypothesis-driven database with existing annotated metabolic models' information would be provided.

The Knowledge Expansion module may be scaled for other molecules such as the different polyssacharides. Moreover, further improvements in the NH module will include the compilation of all MetaCyc and KEGG pathways into a single graph, along with analysing them split into separate graphs, as it could improve the efficacy in the Knowledge Expansion module.

Finally, BOIMMG framework will be integrated with *merlin*, Kbase and the iPlants project.

5.3 Scientific outcomes

- Lima, Diogo; Lagoa, D.; Cruz, Fernando; Bastos, J.; **Capela, João**; Ferreira, Eugénio C.; Rocha, Miguel; Dias, Oscar, merlin v4: an updated platform for reconstructing genome-scale metabolic models. BOD 2020 - IX Bioinformatics Open Days (Conference Book). Braga, Feb 19-21, 2020.

- Cruz, Fernando; **Capela, João**; Ferreira, Eugénio C.; Rocha, Miguel; Dias, Oscar, (manuscript in preparation). Accelerating the reconstruction of genome-scale metabolic models with BioISO.

- **Capela, João**; Rodrigues, Rúben; Lima, Diogo; Lagoa, Davide; Ferreira, Eugénio C.; Rocha, Miguel; Dias, Oscar, (manuscript in preparation for submission in Nucleic Acid Research). *merlin* v4.0: an updated platform for the reconstruction of high-quality genome-scale metabolic models.

BIBLIOGRAPHY

[1] Reed, J. L., Vo, T. D., Schilling, C. H., & Palsson, B. O. (2003). An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR). *Genome biology*.

[2] Orth, J. D., Conrad, T. M., Na, J., Lerman, J. A., Nam, H., Feist, A. M., & Palsson, B. (2011). A comprehensive genome-scale reconstruction of Escherichia coli metabolism-2011. *Molecular Systems Biology*.

[3] Feist, A. M., Henry, C. S., Reed, J. L., Krummenacker, M., Joyce, A. R., Karp, P. D., Broadbelt, L. J., Hatzimanikatis, V., & Palsson, B. Ø. (2007). A genome-scale metabolic reconstruction for ¡i¿Escherichia coli¡/i¿ K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Molecular Systems Biology*, *3*(1), 121.

[4] Palsson, B. (2006). *Systems biology: Properties of reconstructed networks*.

[5] Mukherjee, S., Stamatis, D., Bertsch, J., Ovchinnikova, G., Verezemska, O., Isbandi, M., Thomas, A. D., Ali, R., Sharma, K., Kyrpides, N. C., & Reddy, T. B. (2017). Genomes OnLine Database (GOLD) v.6: Data updates and feature enhancements. *Nucleic Acids Research*.

[6] Oberhardt, M. A., Palsson, B., & Papin, J. A. (2009). Applications of genome-scale metabolic reconstructions.

[7] Aung, H. W., Henry, S. A., & Walker, L. P. (2013). Revising the representation of fatty acid, glycerolipid, and glycerophospholipid metabolism in the consensus model of yeast metabolism. *Industrial Biotechnology*, *9*(4), 215–228.

[8] Sud, M., Fahy, E., Cotter, D., Brown, A., Dennis, E. A., Glass, C. K., Merrill, A. H., Murphy, R. C., Raetz, C. R. H., Russell, D. W., & Subramaniam, S. (2007). LMSD: LIPID MAPS structure database. *Nucleic acids research*, *35*(Database issue), D527–32.

[9] Aimo, L., Liechti, R., Hyka-Nouspikel, N., Niknejad, A., Gleizes, A., Götz, L., Kuznetsov, D., David, F. P., Van Der Goot, F. G., Riezman, H., Bougueleret, L., Xenarios, I., & Bridge, A. (2015). The SwissLipids knowledgebase for lipid biology. *Bioinformatics*.

[10] Fahy, E., Cotter, D., Sud, M., & Subramaniam, S. (2011). Lipid classification, structures and tools. *Biochimica et Biophysica Acta - Molecular and Cell Biology of Lipids*.

[11] Seaver, S. M. D., Liu, F., Zhang, Q., Jeffryes, J., Faria, J. P., Edirisinghe, J. N., Mundy, M., Chia, N., Noor, E., Beber, M. E., Best, A. A., DeJongh, M., Kimbrel, J. A., D'haeseleer, P., McCorkle, S. R., Bolton, J. R., Pearson, E., Canon, S., Wood-Charlson, E. M., . . . Henry, C. S. (2020). The ModelSEED Biochemistry Database for the integration of metabolic annotations and the reconstruction, comparison and analysis of metabolic models for plants, fungi and microbes. *Nucleic Acids Research*.

[12] Ebrahim, A., Lerman, J. A., Palsson, B. O., & Hyduke, D. R. (2013). COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Systems Biology*.

[13] Singh, R. S. (2003). Darwin to DNA, molecules to morphology: The end of classical population genetics and the road ahead.

[14] Watson, J. D., & Crick, F. H. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, *171*(4356), 737–738.

[15] Maxam, A. M., & Gilbert, W. (1977). A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America*, *74*(2), 560–564.

[16] Sanger, F., & Coulson, A. R. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology*, *94*(3).

[17] Metzker, M. L. (2010). Sequencing technologies the next generation.

[18] Craig Venter, J., Adams et al. (2001). The sequence of the human genome. *Science*, *291*(5507), 1304–1351.

[19] Kitano, H. (2002). Computational systems biology.

[20] Kitano, H. (2001). *Foundations of systems biology*. MIT Press.

[21] Edwards, J. S., & Palsson, B. O. (1999). Systems properties of the Haemophilus influenzae Rd metabolic genotype. *Journal of Biological Chemistry*, *274*(25), 17410–17416.

[22] King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O., & Lewis, N. E. (2016). BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, *44*(D1), D515–D522.

[23] Thiele, I., & Palsson, B. (2010). A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature Protocols*, *5*(1), 93–121.

[24] Agren, R., Liu, L., Shoaie, S., Vongsangnak, W., Nookaew, I., & Nielsen, J. (2013). The RAVEN Toolbox and Its Use for Generating a Genome-scale Metabolic Model for Penicillium chrysogenum. *PLoS Computational Biology*, *9*(3).

[25] Dias, O., Rocha, M., Ferreira, E. C., & Rocha, I. (2015). Reconstructing genome-scale metabolic models with merlin. *Nucleic Acids Research*.

[26] Pitkänen, E., Jouhten, P., Hou, J., Syed, M. F., Blomberg, P., Kludas, J., Oja, M., Holm, L., Penttilä, M., Rousu, J., & Arvas, M. (2014). Comparative Genome-Scale Reconstruction of Gapless Metabolic Networks for Present and Ancestral Species. *PLoS Computational Biology*, *10*(2).

[27] Dias, O., Pereira, R., Gombert, A. K., Ferreira, E. C., & Rocha, I. (2014). iOD907, the first genome-scale metabolic model for the milk yeast Kluyveromyces lactis. *Biotechnology Journal*, *9*(6), 776–790.

[28] Mendoza, S. N., Olivier, B. G., Molenaar, D., & Teusink, B. (2019). A systematic assessment of current genome-scale metabolic reconstruction tools. *Genome Biology*, *20*(1), 158.

[29] Hucka, M. et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, *19*(4), 524–531.

[30] Machado, D., Herrgård, M. J., & Rocha, I. (2016). Stoichiometric Representation of Gene–Protein–Reaction Associations Leverages Constraint-Based Analysis from Reaction to Gene-Level Phenotype Prediction. *PLoS Computational Biology*, *12*(10).

[31] Aite, M., Chevallier, M., Frioux, C., Trottier, C., Got, J., Cortés, M. P., Mendoza, S. N., Carrier, G., Dameron, O., Guillaudeux, N., Latorre, M., Loira, N., Markov, G. V., Maass, A., & Siegel,

A. (2018). Traceability, reproducibility and wiki-exploration for “à-la-carte” reconstructions of genome-scale metabolic models (J. Nielsen, Ed.). *PLOS Computational Biology*, *14*(5), e1006146.

[32] Machado, D., Andrejev, S., Tramontano, M., & Patil, K. R. (2018). Fast automated reconstruction of genome-scale metabolic models for microbial species and communities. *Nucleic Acids Research*, *46*(15), 7542–7553.

[33] Karlsen, E., Schulz, C., & Almaas, E. (2018). Automated generation of genome-scale metabolic draft reconstructions based on KEGG. *BMC Bioinformatics*, *19*(1).

[34] Wang, H., Marcišauskas, S., Sánchez, B. J., Domenzain, I., Hermansson, D., Agren, R., Nielsen, J., & Kerkhoven, E. J. (2018). RAVEN 2.0: A versatile toolbox for metabolic network reconstruction and a case study on Streptomyces coelicolor (C. A. Ouzounis, Ed.). *PLOS Computational Biology*, *14*(10), e1006541.

[35] Karp, P. D., Latendresse, M., Paley, S. M., Krummenacker, M., Ong, Q. D., Billington, R., Kothari, A., Weaver, D., Lee, T., Subhraveti, P., Spaulding, A., Fulcher, C., Keseler, I. M., & Caspi, R. (2016). Pathway Tools version 19.0 update: software for pathway/genome informatics and systems biology. *Briefings in Bioinformatics*, *17*(5), 877–890.

[36] Olivier, B. G. (2018). Metadraft. *https://systemsbioinformatics.github.io/cbmpy-metadraft/*.

[37] Pabinger, S., Snajder, R., Hardiman, T., Willi, M., Dander, A., & Trajanoski, Z. (2014). MEMOSys 2.0: an update of the bioinformatics database for genome-scale models and genomic data. *Database*, *2014*.

[38] Henry, C. S., Dejongh, M., Best, A. A., Frybarger, P. M., Linsay, B., & Stevens, R. L. (2010). High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nature Biotechnology*, *28*(9), 977–982.

[39] Boele, J., Olivier, B. G., & Teusink, B. (2012). FAME, the Flux Analysis and Modeling Environment. *BMC Systems Biology*, *6*(1), 8.

[40] Arkin, A. P., & other. (2018). KBase: The United States Department of Energy Systems Biology Knowledgebase. *Nature Biotechnology*.

[41] Reed, J. L. (2012). Shrinking the Metabolic Solution Space Using Experimental Datasets (J. A. Papin, Ed.). *PLoS Computational Biology*, *8*(8), e1002662.

[42] O’Brien, E. J., Monk, J. M., & Palsson, B. O. (2015). Using genome-scale models to predict biological capabilities.

[43] Lewis, N. E., Hixson, K. K., Conrad, T. M., Lerman, J. A., Charusanti, P., Polpitiya, A. D., Adkins, J. N., Schramm, G., Purvine, S. O., Lopez-Ferrer, D., Weitz, K. K., Eils, R., König, R., Smith, R. D., & Palsson, B. (2010). Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models. *Molecular Systems Biology*, *6*.

[44] Mahadevan, R., & Schilling, C. H. (2003). The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic Engineering*, *5*(4), 264–276.

[45] Segrè, D., Vitkup, D., & Church, G. M. (2002). Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(23), 15112–15117.

[46] Shlomi, T., Berkman, O., & Ruppin, E. (2005). Regulatory on/off minimization of metabolic flux changes after genetic perturbations. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(21), 7695–7700.

[47] Vlassis, N., Pacheco, M. P., & Sauter, T. (2014). Fast Reconstruction of Compact Context-Specific Metabolic Network Models. *PLoS Computational Biology*, *10*(1).

[48] Kim, B., Kim, W. J., Kim, D. I., & Lee, S. Y. (2014). Applications of genome-scale metabolic network model in metabolic engineering. *Journal of Industrial Microbiology and Biotechnology*, *42*(3), 339–348.

[49] Gu, C., Kim, G. B., Kim, W. J., Kim, H. U., & Lee, S. Y. (2019). Current status and applications of genome-scale metabolic models.

[50] Zwier, K. R. (2011). John Dalton's puzzles: From meteorology to chemistry. *Studies in History and Philosophy of Science Part A*, *42*(1), 58–66.

[51] Silberberg, M. S. (S. (2013). *Principles of general chemistry*. McGraw-Hill.

[52] Hastings, J., Magka, D., Batchelor, C., Duan, L., Stevens, R., Ennis, M., & Steinbeck, C. (2012). *Structure-based classification and ontology in chemistry* (tech. rep.).

[53] Warr, W. A. (2011). Representation of chemical structures. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *1*(4), 557–579.

[54] Gasteiger, J. (, & Engel, T. (2003). *Chemoinformatics : a textbook*. Wiley-VCH.

[55] Polanski, J., & Gasteiger, J. (2017). Computer representation of chemical compounds. *Handbook of computational chemistry* (pp. 1997–2039). Springer International Publishing.

[56] Morgan, H. L. (1965). The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation*, *5*(2), 107–113.

[57] Wipke, W. T., & Dyott, T. M. (1974). Stereochemically unique naming algorithm. *Journal of the American Chemical Society*, *96*(15), 4834–4842.

[58] Bauerschmidt, S., & Gasteiger, J. (1997). *Overcoming the Limitations of a Connection Table Description: A Universal Representation of Chemical Species* (tech. rep.).

[59] Dalby, A., Nourse, J. G., Hounshell, W. D., Gushurst, A. K., Grier, D. L., Leland, B. A., & Laufer, J. (1992). Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited. *Journal of Chemical Information and Computer Sciences*, *32*(3), 244–255.

[60] Hastings, J., Owen, G., Dekker, A., Ennis, M., Kale, N., Muthukrishnan, V., Turner, S., Swainston, N., Mendes, P., & Steinbeck, C. (2016). ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Research*, *44*(D1), D1214–D1219.

[61] O'boyle, N. M. (2012). *Towards a Universal SMILES representation-A standard method to generate canonical SMILES based on the InChI* (tech. rep.).

[62] Vollmer, J. J. (1983). Wiswesser line notation: an introduction. *Journal of Chemical Education*, *60*(3), 192.

[63] Weininger, D. (1988). SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules. *Journal of Chemical Information and Computer Sciences*, *28*(1), 31–36.

[64] Lin, T.-S., Coley, C. W., Mochigase, H., Beech, H. K., Wang, W., Wang, Z., Woods, E., Craig, S. L., Johnson, J. A., Kalow, J. A., Jensen, K. F., & Olsen, B. D. (2019). BigSMILES: A Structurally-Based Line Notation for Describing Macromolecules.

[65] Ash, S., Cline, M. A., Homer, R. W., Hurst, T., & Smith, G. B. (1997). SYBYL Line Notation (SLN): A versatile language for chemical structure representation. *Journal of Chemical Information and Computer Sciences*, *37*(1), 71–79.

[66] Rohbeck, H.-G. (1991). Representation of Structure Description Arranged Linearly. *Software development in chemistry 5* (pp. 49–58). Springer Berlin Heidelberg.

[67] Heller, S. R., McNaught, A., Pletnev, I., Stein, S., & Tchekhovskoi, D. (2015). InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics*, *7*(1).

[68] Lipkus, A. H., Yuan, Q., Lucas, K. A., Funk, S. A., Bartelt, W. F., Schenck, R. J., & Trippe, A. J. (2008). Structural diversity of organic chemistry. A scaffold analysis of the CAS Registry. *Journal of Organic Chemistry*, *73*(12), 4443–4451.

[69] Probst, D., & Reymond, J. L. (2018). SmilesDrawer: Parsing and Drawing SMILES-Encoded Molecular Structures Using Client-Side JavaScript. *Journal of Chemical Information and Modeling*, *58*(1), 1–7.

[70] Daylight Theory: SMIRKS - A Reaction Transform Language [https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html, accessed in 2019-12-31]. (n.d.).

[71] Heller, S., McNaught, A., Stein, S., Tchekhovskoi, D., & Pletnev, I. (2013). InChI - The worldwide chemical structure identifier standard.

[72] Kanehisa, M., Sato, Y., Furumichi, M., Morishima, K., & Tanabe, M. (2019). New approach for understanding genome variations in KEGG. *Nucleic Acids Research*, *47*(D1), D590–D595.

[73] Caspi, R., Billington, R., Fulcher, C. A., Keseler, I. M., Kothari, A., Krummenacker, M., Latendresse, M., Midford, P. E., Ong, Q., Ong, W. K., Paley, S., Subhraveti, P., & Karp, P. D. (2018). The MetaCyc database of metabolic pathways and enzymes. *Nucleic Acids Research*, *46*(D1), D633–D639.

[74] Devoid, S., Overbeek, R., DeJongh, M., Vonstein, V., Best, A. A., & Henry, C. (2013). Automated genome annotation and metabolic model reconstruction in the SEED and model SEED. *Methods in Molecular Biology*, *985*, 17–45.

[75] Placzek, S., Schomburg, I., Chang, A., Jeske, L., Ulbrich, M., Tillack, J., & Schomburg, D. (2017). BRENDA in 2017: New perspectives and new tools in BRENDA. *Nucleic Acids Research*, *45*(D1), D380–D388.

[76] Moretti, S., Martin, O., Van Du Tran, T., Bridge, A., Morgat, A., & Pagni, M. (2016). MetaNetX/MNXref - Reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. *Nucleic Acids Research*, *44*(D1), D523–D526.

[77] Bernard, T., Bridge, A., Morgat, A., Moretti, S., Xenarios, I., & Pagni, M. (2014). Reconciliation of metabolites and biochemical reactions for metabolic networks. *Briefings in Bioinformatics*, *15*(1), 123–135.

[78] Jassal, B., Matthews, L., Viteri, G., Gong, C., Lorente, P., Fabregat, A., Sidiropoulos, K., Cook, J., Gillespie, M., Haw, R., Loney, F., May, B., Milacic, M., Rothfels, K., Sevilla, C.,

Shamovsky, V., Shorser, S., Varusai, T., Weiser, J., . . . D'Eustachio, P. (2019). The reactome pathway knowledgebase. *Nucleic acids research*.

[79] Fahy, E., Subramaniam, S., Brown, H. A., Glass, C. K., Merrill, A. H., Murphy, R. C., Raetz, C. R., Russell, D. W., Seyama, Y., Shaw, W., Shimizu, T., Spener, F., Van Meer, G., VanNieuwenhze, M. S., White, S. H., Witztum, J. L., & Dennis, E. A. (2005). A comprehensive classification system for lipids. *Journal of Lipid Research*.

[80] Nowicka, B., & Kruk, J. (2010). Occurrence, biosynthesis and function of isoprenoid quinones.

[81] Ibrahim, R. K., & Muzac, I. (2000). Chapter Eleven The methyltransferase gene superfamily: A tree with multiple branches. *Recent advances in phytochemistry* (pp. 349–384). Elsevier Inc.

[82] Oettmeier, W., & Trebst, A. (1983). INHIBITOR AND PLASTOQUINONE BINDING TO PHOTOSYSTEM II. *The oxygen evolving system of photosynthesis* (pp. 411–420). Elsevier.

[83] Verberne, M. C., Muljono, R. A., & Verpoorte, R. (1999). Chapter 13 Salicylic acid biosynthesis. *New comprehensive biochemistry* (pp. 295–312). Elsevier.

[84] Fujimoto, N., Kosaka, T., & Yamada, M. (n.d.). *Menaquinone as Well as Ubiquinone as a Crucial Component in the Escherichia coli Respiratory Chain* (tech. rep.).

[85] Campbell, A. R., Titus, B. R., Kuenzi, M. R., Rodriguez-Perez, F., Brunsch, A. D., Schroll, M. M., Owen, M. C., Cronk, J. D., Anders, K. R., & Shepherd, J. N. (2019). Investigation of candidate genes involved in the rhodoquinone biosynthetic pathway in Rhodospirillum rubrum. *PLoS ONE*, *14*(5).

[86] van Oostende, C., Widhalm, J. R., Furt, F., Ducluzeau, A. L., & Basset, G. J. (2011). Vitamin K1 (Phylloquinone): Function, enzymes and genes. *Advances in botanical research* (pp. 229–261). Academic Press Inc.

[87] Sánchez, B. J., Li, F., Kerkhoven, E. J., & Nielsen, J. (2019). SLIMEr: Probing flexibility of lipid metabolism in yeast with an improved constraint-based modeling framework. *BMC Systems Biology*.

[88] Poupin, N., Vinson, F., Moreau, A., Batut, A., Chazalviel, M., Colsch, B., Fouillen, L., Guez, S., Khoury, S., Dalloux-Chioccioli, J., Tournadre, A., Le Faouder, P., Pouyet, C., Van Delft, P., Viars, F., Bertrand-Michel, J., & Jourdan, F. (2020). Improving lipid mapping in Genome Scale Metabolic Networks using ontologies. *Metabolomics*.

[89] Cottret, L., Frainay, C., Chazalviel, M., Cabanettes, F., Gloaguen, Y., Camenen, E., Merlet, B., Heux, S., Portais, J. C., Poupin, N., Vinson, F., & Jourdan, F. (2018). MetExplore: Collaborative edition and exploration of metabolic networks. *Nucleic Acids Research*.

[90] Zhukova, A., & Sherman, D. J. (2014a). Knowledge-based Generalization of Metabolic Models. *Journal of Computational Biology*, *21*(7), 534–547.

[91] Zhukova, A., & Sherman, D. J. (2014b). Knowledge-based generalization of metabolic networks: A practical study. *Journal of Bioinformatics and Computational Biology*, *12*(2).

[92] Zhukova, A., & Sherman, D. J. (2015). Mimoza: Web-based semantic zooming and navigation in metabolic networks. *BMC Systems Biology*, *9*(1).

[93] King, B., Farrah, T., Richards, M. A., Mundy, M., Simeonidis, E., & Price, N. D. (2018). ProbAnnoWeb and ProbAnnoPy: Probabilistic annotation and gap-filling of metabolic reconstructions. *Bioinformatics*, *34*(9), 1594–1596.

[94] Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., . . . Scopatz, A. (2017). Sympy: Symbolic computing in python. *PeerJ Computer Science*, *3*, e103.

[95] Monk, J. M., Lloyd, C. J., Brunk, E., Mih, N., Sastry, A., King, Z., Takeuchi, R., Nomura, W., Zhang, Z., Mori, H., Feist, A. M., & Palsson, B. O. (2017). iML1515, a knowledgebase that computes Escherichia coli traits.

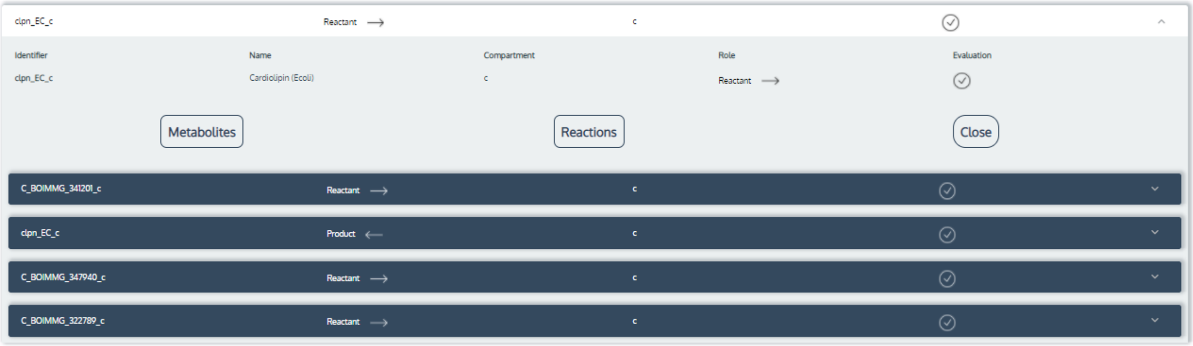
[96] Mo, M. L., Palsson, B., & Herrgård, M. J. (2009). Connecting extracellular metabolomic measurements to intracellular flux states in yeast. *BMC Systems Biology*.

[97] Cruz, F. (2017). *Genome-Scale Metabolic Network Reconstruction of the dairy bacterium Streptococcus thermophilus* (MSc Thesis). Universidade do Minho.

[98] Foster, J. M., Moreno, P., Fabregat, A., Hermjakob, H., Steinbeck, C., Apweiler, R., Wakelam, M. J. O., & Vizcaíno, J. A. (2013). LipidHome: A Database of Theoretical Lipids Optimized for High Throughput Mass Spectrometry Lipidomics (M. Oresic, Ed.). *PLoS ONE*, *8*(5), e61951.

6

SUPPLEMENTARY MATERIAL

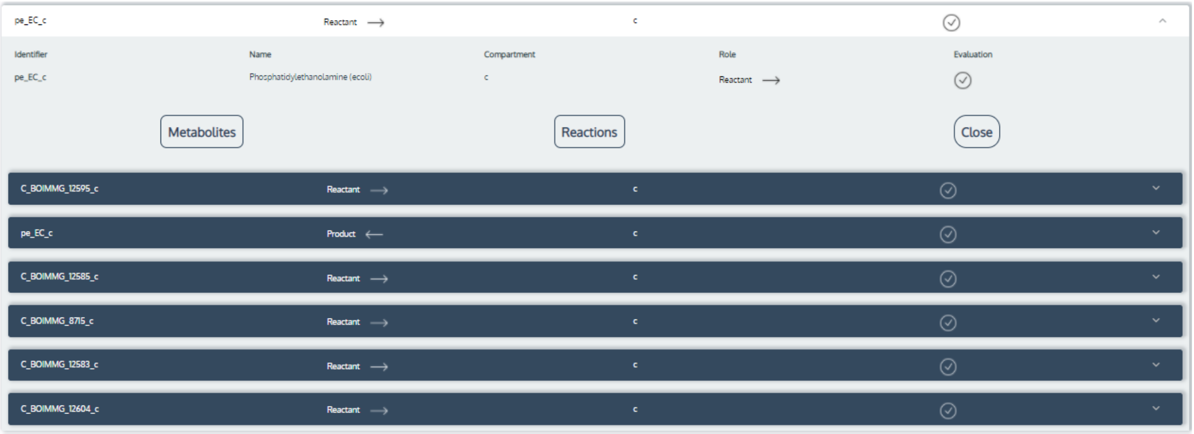


clpn_EC_c	Reactant →	c	✓

Identifier	Name	Compartment	Role	Evaluation
clpn_EC_c	Cardiolipin (Ecoli)	c	Reactant →	✓

C_BOIMMG_341201_c	Reactant →	c	✓
clpn_EC_c	Product ←	c	✓
C_BOIMMG_347940_c	Reactant →	c	✓
C_BOIMMG_322789_c	Reactant →	c	✓

Figure 46: BioISO flux analysis for the complete cardiolipins. The checkmark associated to each cardiolipin indicates that it is carrying flux.

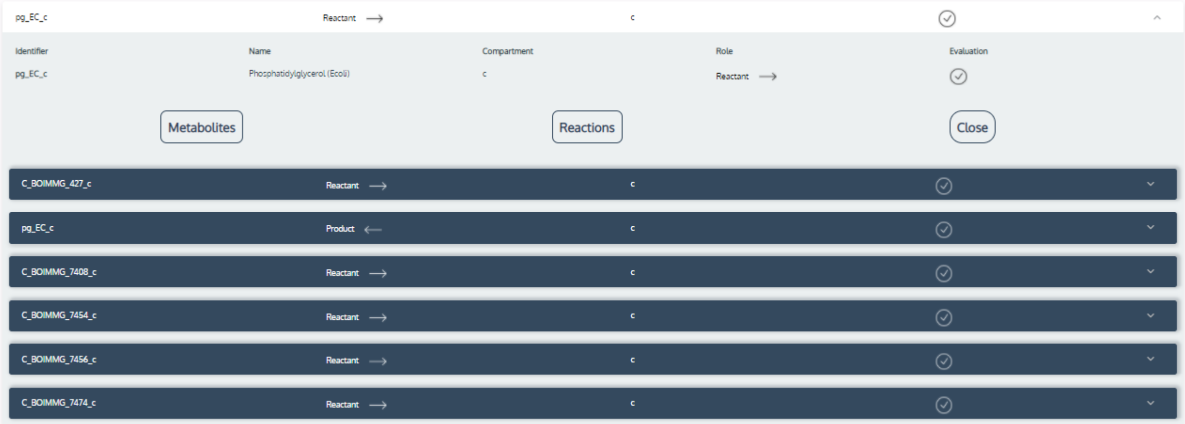


pe_EC_c	Reactant →	c	✓

Identifier	Name	Compartment	Role	Evaluation
pe_EC_c	Phosphatidylethanolamine (ecoli)	c	Reactant →	✓

C_BOIMMG_12595_c	Reactant →	c	✓
pe_EC_c	Product ←	c	✓
C_BOIMMG_12585_c	Reactant →	c	✓
C_BOIMMG_8715_c	Reactant →	c	✓
C_BOIMMG_12583_c	Reactant →	c	✓
C_BOIMMG_12604_c	Reactant →	c	✓

Figure 47: BioISO flux analysis for the complete phosphatidylethanolamines. The checkmark associated to each phosphatidylethanolamines indicates that it is carrying flux.

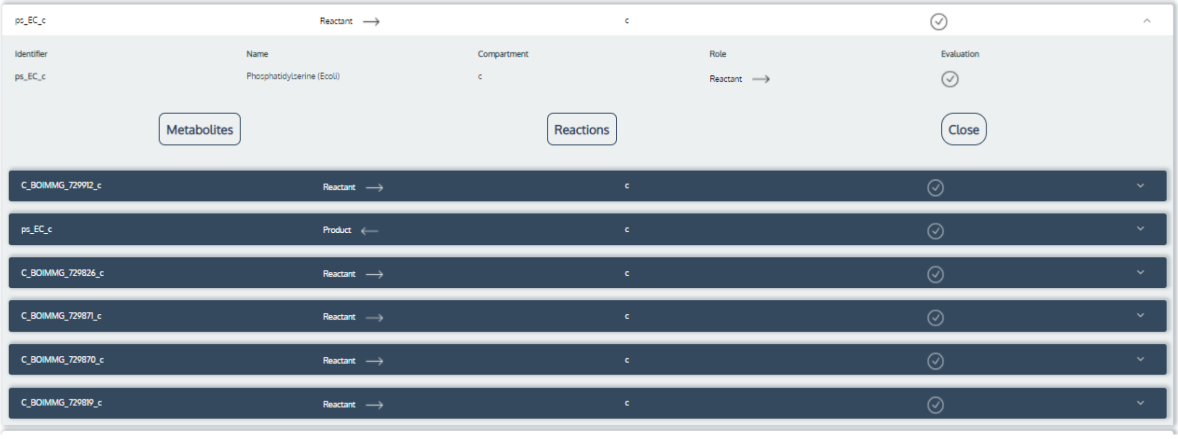


pg_EC_c	Reactant ⟶	c	✓

Identifier	Name	Compartment	Role	Evaluation
pg_EC_c	Phosphatidylglycerol (Ecoli)	c	Reactant ⟶	✓

C_BOIMMG_427_c	Reactant ⟶	c	✓
pg_EC_c	Product ⟵	c	✓
C_BOIMMG_7408_c	Reactant ⟶	c	✓
C_BOIMMG_7454_c	Reactant ⟶	c	✓
C_BOIMMG_7456_c	Reactant ⟶	c	✓
C_BOIMMG_7474_c	Reactant ⟶	c	✓

Figure 48: BioISO flux analysis for the complete phosphatidylglycerol. The checkmark associated to each phosphatidylglycerols indicates that it is carrying flux.



ps_EC_c	Reactant ⟶	c	✓

Identifier	Name	Compartment	Role	Evaluation
ps_EC_c	Phosphatidylserine (Ecoli)	c	Reactant ⟶	✓

C_BOIMMG_729912_c	Reactant ⟶	c	✓
ps_EC_c	Product ⟵	c	✓
C_BOIMMG_729826_c	Reactant ⟶	c	✓
C_BOIMMG_729871_c	Reactant ⟶	c	✓
C_BOIMMG_729870_c	Reactant ⟶	c	✓
C_BOIMMG_729819_c	Reactant ⟶	c	✓

Figure 49: BioISO flux analysis for the complete phosphatidylserine. The checkmark associated to each phosphatidylserine indicates that it is carrying flux.

6.1 Added lipids and their correspondence to the iAF1260b model

Table 30: Added lipids and their correspondence to the iAF1260b model

Phosphatidylglycerophosphate (PGP)					Acyl-glycerophosphoglycerol (AGPG)				
Compound	iAF1260b ID	In	Granulated Model ID	Produced	Compound	iAF1260b ID	In	Granulated Model ID	Produced
PGP (12:0)	pgp120	X	-	-	AGPG (12:0)	1agpg120	X	-	-
PGP (14:0)	pgp140	✓	C_BOIMMG_296532	✓	AGPG (14:0)	1agpg140	✓	C_BOIMMG_452	✓
PGP (14:1)	pgp141	✓	C_BOIMMG_296513	✓	AGPG (14:1)	1agpg141	✓	C_BOIMMG_16364	✓
PGP (16:0)	pgp160	✓	C_BOIMMG_296523	✓	AGPG (16:0)	1agpg160	✓	C_BOIMMG_239898	✓
PGP(16:1)	pgp161	✓	C_BOIMMG_296512	✓	AGPG (16:1)	1agpg161	✓	C_BOIMMG_314	✓
PGP(18:0)	pgp180	X	-	-	AGPG (18:0)	1agpg180	X	-	-
PGP(18:1)	pgp181	✓	C_BOIMMG_293736	✓	AGPG (18:1)	1agpg181	✓	C_BOIMMG_16317	✓
CDPdiacylglycerol (CDPdag)					Acyl phosphatidylglycerol (APG)				
Compound	iAF1260b ID	In	Granulated Model ID	Produced	Compound	iAF1260b ID	In	Granulated Model ID	Produced
CDPdag (12:0)	cdpdddecg	X	-	-	APG (12:0)	1apg120	X	-	-
CDPdag (14:0)	cdpdtdecg	✓	C_BOIMMG_574421	✓	APG (14:0)	1apg140	✓	C_BOIMMG_765781	✓
CDPdag (14:1)	cdpdtdec7eg	✓	C_BOIMMG_574506	✓	APG (14:1)	1apg141	✓	C_BOIMMG_765700	✓
CDPdag (16:0)	cdpdhdecg	✓	C_BOIMMG_239898	✓	APG (16:0)	1apg160	✓	C_BOIMMG_765924	✓
CDPdag(16:1)	cdpdhdec9eg	✓	C_BOIMMG_524486	✓	APG (16:1)	1apg161	✓	C_BOIMMG_765716	✓
CDPdag(18:0)	cdpdodecg	X	-	-	APG (18:0)	1apg180	X	-	-
CDPdag(18:1)	cdpdodec11eg	✓	C_BOIMMG_540741	✓	APG (18:1)	1apg181	✓	C_BOIMMG_765860	✓

Table 31: Added lipids and their correspondence to the iAF1260b model - rest of the lipids

Acyl-glycerophosphoethanolamine (AGPE)					Phosphatidate (PA)				
Compound	iAF1260b ID	In	Granulated Model ID	Produced	Compound	iAF1260b ID	In	Granulated Model ID	Produced
AGPE (12:0)	1agpe120	X	-	-	PA (12:0)	pa120	X	-	-
AGPE (14:0)	1agpe140	✓	C_BOIMMG_27943	✓	PA (14:0)	pa140	✓	C_BOIMMG_38417	✓
AGPE (14:1)	1agpe141	✓	C_BOIMMG_27925	✓	PA (14:1)	pa141	✓	C_BOIMMG_38435	✓
AGPE (16:0)	1agpe160	✓	C_BOIMMG_291	✓	PA (16:0)	pa160	✓	C_BOIMMG_423	✓
AGPE (16:1)	1agpe161	✓	C_BOIMMG_27923	✓	PA (16:1)	pa161	✓	C_BOIMMG_38416	✓
AGPE (18:0)	1agpe180	X	-	-	PA (18:0)	pa180	X	-	-
AGPE (18:1)	1agpe181	✓	C_BOIMMG_27877	✓	PA (18:1)	pa181	✓	C_BOIMMG_38371	✓

1,2-Diacylglycerol (12DGR)				
Compound	iAF1260b ID	In	Granulated Model ID	Produced
12DGR (12:0)	12dgr120	X	-	-
12DGR (14:0)	12dgr140	✓	C_BOIMMG_53297	✓
12DGR (14:1)	12dgr141	✓	C_BOIMMG_53244	✓
12DGR (16:0)	12dgr160	✓	C_BOIMMG_61409	✓
12DGR(16:1)	12dgr161	✓	C_BOIMMG_61238	✓
12DGR(18:0)	12dgr180	X	-	-
12DGR(18:1)	12dgr181	✓	C_BOIMMG_70100	✓