2021

DEVELOPMENT OF TOOLS FOR ATOM-LEVEL INTERPRETATION OF STABLE ISOTOPE-RESOLVED METABOLOMICS DATASETS

Huan Jin
University of Kentucky, jinhuan0905@gmail.com
Author ORCID Identifier:
https://orcid.org/0000-0001-5886-7481

Digital Object Identifier: https://doi.org/10.13023/etd.2021.303

Recommended Citation

Jin, Huan, "DEVELOPMENT OF TOOLS FOR ATOM-LEVEL INTERPRETATION OF STABLE ISOTOPE-RESOLVED METABOLOMICS DATASETS" (2021). *Theses and Dissertations--Toxicology and Cancer Biology*. 37.
https://uknowledge.uky.edu/toxicology_etds/37

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student’s advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student’s thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Huan Jin, Student

Dr. Hunter N. B. Moseley, Major Professor

Dr. Isabel Mellon, Director of Graduate Studies

DEVELOPMENT OF TOOLS FOR ATOM-LEVEL INTERPRETATION OF STABLE ISOTOPE-RESOLVED METABOLOMICS DATASETS

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Medicine
at the University of Kentucky

By
Huan Jin
Lexington, Kentucky
Director: Dr. Hunter N.B. Moseley, Professor of Molecular and Cellular Biochemistry
Lexington, Kentucky
2021

ABSTRACT OF DISSERTATION

DEVELOPMENT OF TOOLS FOR ATOM-LEVEL INTERPRETATION OF STABLE ISOTOPE-RESOLVED METABOLOMICS DATASETS

Metabolomics is the global study of small molecules in living systems under a given state, merging as a new ‘omics’ study in systems biology. It has shown great promise in elucidating biological mechanism in various areas. Many diseases, especially cancers, are closely linked to reprogrammed metabolism. As the end point of biological processes, metabolic profiles are more representative of the biological phenotype compared to genomic or proteomic profiles. Therefore, characterizing metabolic phenotype of various diseases will help clarify the metabolic mechanisms and promote the development of novel and effective treatment strategies.

Advances in analytical technologies such as nuclear magnetic resonance and mass spectroscopy greatly contribute to the detection and characterization of global metabolites in a biological system. Furthermore, application of these analytical tools to stable isotope resolved metabolomics experiments can generate large-scale high-quality metabolomics data containing isotopic flow through cellular metabolism. However, the lack of the corresponding computational analysis tools hinders the characterization of metabolic phenotypes and the downstream applications.

Both detailed metabolic modeling and quantitative analysis are required for proper interpretation of these complex metabolomics data. For metabolic modeling, currently there is no comprehensive metabolic network at an atom-resolved level that can be used for deriving context-specific metabolic models for SIRM metabolomics datasets. For quantitative analysis, most available tools conduct metabolic flux analysis based on a well-defined metabolic model, which is hard to achieve for complex biological system due to the limitations in our knowledge.

Here, we developed a set of methods to address these problems. First, we developed a neighborhood-specific coloring method that can create identifier for each atom in a specific compound. With the atom identifiers, we successfully harmonized compounds and reactions across KEGG and MetaCyc databases at various levels. In addition, we evaluated the atom mappings of the harmonized metabolic reactions. These results will contribute to the construction of a comprehensive atom-resolved metabolic network. In addition, this method can be easily applied to any metabolic database that provides a molfile representation of compounds, which will greatly facilitate future expansion. In addition, we developed a moiety modeling framework to deconvolute metabolite isotopologue profiles using moiety models along with the analysis and selection of the best moiety model(s) based on the experimental data. To our knowledge, this is the first method that can analyze datasets involving multiple isotope tracers. Furthermore, instead of a single predefined metabolic model, this method allows the comparison of multiple metabolic models derived from a given metabolic profile, and we have demonstrated the robust performance of the moiety modeling framework in model selection with a ^{13}C-labeled UDP-GlcNAc isotopologue dataset. We further explored the data quality requirements and the factors that affect model selection. Collectively, these methods and tools help interpret SIRM metabolomics datasets from metabolic modeling to quantitative analysis.

KEYWORDS: Metabolomics, Atom-Resolved Metabolic Network, Maximum Common Isomorphism, Metabolic Database Harmonization, Atom Identifier

Huan Jin
(Name of Student)

07/21/2021
Date

DEVELOPMENT OF TOOLS FOR ATOM-LEVEL INTERPRETATION OF
STABLE ISOTOPE-RESOLVED METABOLOMICS DATASETS

By
Huan Jin

Dr. Hunter N.B. Moseley
Director of Dissertation

Dr. Isabel Mellon
Director of Graduate Studies

07/21/2021
Date

DEDICATION

To my parents:

Binyun Jin and Hong Liu

ACKNOWLEDGMENTS

The following dissertation, while an individual work, would not have been possible without the insights and direction of several people. First, I would like to thank my advisor Dr. Hunter N.B. Moseley for the opportunity to work in his lab and his support developing this project. I appreciate the freedom and trust Dr. Moseley has afforded me in the lab which allows me to try bravely, fail fearlessly and learn deeply. As an amazing advisor, Dr. Moseley not only helps foster creativity and critical thinking, but also instill my confidence and courage. Furthermore, I would like to thank all my lab members for your kind help in my qualifying exam, project and life. I would also like to thank my Advisory Committee: Drs. Christine Fillmore Brainson, Yekaterina Zaytseva, Miroslaw Truszczynski, Russell Brown for the time spent on collaborations, committee meetings, and exams as well as for the great suggestions on my research projects. Lastly, I would like to give a special thanks to Dr. Patricia Rippetoe Freeman for serving as Outside Examiner of my final exam.

Additionally, I received important assistance from friends. Your company and support have made this journey much more enjoyable. I will always miss the days we spent together. Finally, I wish to thank my parents. It is because of your unconditional support that I can pursue my dream in a distant country.

TABLE OF CONTENTS

ACKNOWLEDGMENTS .... **iii**

LIST OF TABLES .... **vii**

LIST OF FIGURES .... **ix**

LIST OF ADDITIONAL FILES .... **xi**

CHAPTER 1. Background **1**

1.1 Metabolomics *1*

1.2 Metabolic Reprogramming *2*

1.3 Metabolic Network and Metabolic Model *4*

1.4 Stable Isotope Resolved Metabolomics *6*

1.5 Metabolic Flux Analysis *9*

1.6 Challenges in Interpreting SIRM Derived Metabolome *11*

1.7 Overview of Dissertation *12*

CHAPTER 2. Neighborhood-specific coloring method for creating atom identifier in a compound 14

2.1 Introduction *14*

2.2 Materials and Methods *17*

2.2.1 Compound and metabolic reaction data 17
2.2.2 Overview of major analysis steps 17
2.2.3 Molfile parser 18
2.2.4 Aromatic substructure detection 18
2.2.5 Identification of double bond stereochemistry 19
2.2.6 Neighborhood-specific graph coloring method 19
2.2.7 Atom coloring validation and recolor 21
2.2.8 Creation of compound coloring identifiers based on atom coloring identifiers 22
2.2.9 Prediction of possible compound correspondence via metabolic reactions 23

2.3 Results *23*

2.3.1 Overview of KEGG and MetaCyc databases 23
2.3.2 Aromatic substructure detection 24
2.3.3 Generating identifiers for atoms using a graph coloring method 27
2.3.4 Detection of correspondences between KEGG and MetaCyc compounds via coloring identifiers. 29
2.3.5 Compound representation errors and issues detected in the KEGG and MetaCyc databases. 32

2.3.6 Estimating the error rate of the graph coloring method. 35

2.4 Discussion 39

CHAPTER 3. Hierarchical Harmonization of metabolic reactions across metabolic databases 42

3.1 Introduction 42

3.2 Materials and Methods 45

3.2.1 Compound and Metabolic Reaction Data 45
3.2.2 Curation of molfile 45
3.2.3 Identification of double bond stereochemistry 45
3.2.4 Flowchart of steps in the compound and reaction harmonization process 46
3.2.5 Validation of tautomers 46
3.2.6 Validation of generic compound pairs of compounds with different chemical formula. 47
3.2.7 Validation of compound pairs with linear and circular representations. 47
3.2.8 Parse of KEGG RCLASS RDM patterns. 48

3.3 Results 48

3.3.1 Overview of KEGG and MetaCyc databases 48
3.3.2 Results of compound harmonization across KEGG and MetaCyc databases 50
3.3.3 Results of reaction harmonization across KEGG and MetaCyc databases. 54
3.3.4 Comparison of KEGG RCLASS and RPAIR data 57
3.3.5 Evaluation of atom mappings between KEGG and MetaCyc databases. 65

3.4 Discussion 69

CHAPTER 4. Moiety modeling framework for deriving moiety abundances from mass spectrometry measured isotopologues 73

4.1 Introduction 73

4.2 Implementation 77

4.2.1 Overview of the moiety modeling framework 77
4.2.2 The moiety_modeling Python package implementation 81
4.2.3 SAGA-optimize Python package implementation 86

4.3 Results 87

4.3.1 The package interface 87
4.3.2 Dataset and model 89
4.3.3 Model optimization and selection 90
4.3.4 Generation of simulated single-tracer and multi-tracer datasets 92
4.3.5 Model optimization of simulated multi-tracer and single-tracer datasets and comparison of results 95

4.4 Discussion 96

4.4.1 Advantage of JSONized representation for MS isotopologue data and analysis results 96
4.4.2 Advantages and limitations of the SAGA-optimize and moiety-modeling packages 97
4.4.3 Difficulty in generating simulated datasets and comparing multi-tracer to single-tracer moiety modeling results 98

4.5 Conclusions 99

CHAPTER 5. Robust moiety model selection using mass spectrometry measured isotopologues 101

5.1 Introduction...101

5.2 Materials and Methods...103

5.2.1 UDP-GlcNAc time course MS isotopologue datasets 103
5.2.2 Objective functions... 104
5.2.3 Optimization methods 104
5.2.4 Model Selection Estimators.. 105

5.3 Results 106

5.3.1 UDP-GlcNAc moiety model construction. 106
5.3.2 A simple comparison of two moiety models 107
5.3.3 Effects of optimization method on model selection 110
5.3.4 Over-optimization leads to failure in model selection. 111
5.3.5 Effects of selection criterion on model selection 112
5.3.6 Effects of selection criterion on model selection 114
5.3.7 Effects of information quantity on model selection.. 122

5.4 Discussion 126

CHAPTER 6. Conclusion... 128

APPENDICES 133

APPENDIX 1. SOFTWARE 133

APPENDIX 2. DATA AVAILABILITY 134

REFERENCES 135

VITA 143

LIST OF TABLES

Table 2.1. Generation of atom identifiers for compound C00047 via graph coloring method. 21
Table 2.2. KEGG and MetaCyc databases........ 23
Table 2.3. Correspondences between KEGG and MetaCyc compounds........ 24
Table 2.4. Incomplete detection of aromatic substructures by BASS and Indigo 26
Table 2.5. Compounds with aromatic substructure 27
Table 2.6. Matched compounds detected by the compound coloring identifiers 30
Table 2.7. Matched compounds detected by the compound coloring identifiers 31
Table 2.8. Correspondences between KEGG and MetaCyc compounds verified by reactions 31
Table 2.9. Compounds with the same coloring identifier, excluding R groups........ 36
Table 2.10. Compounds with the same coloring identifiers, which includes R groups.... 36
Table 2.11. Compounds gaining asymmetry after addition of extra information in the atom naming........ 38
Table 3.1. Summary of KEGG and MetaCyc compound databases......... 48
Table 3.2. Summary of KEGG and MetaCyc atom-resolved metabolic reaction databases 49
Table 3.3. Quality check of atom-resolved reactions in KEGG and MetaCyc......... 49
Table 3.4.Harmonization of specific compound pairs......... 52
Table 3.5. Harmonization of generic compounds......... 53
Table 3.6. Harmonization of compounds with changeable representations. 53
Table 3.7. Summary of compound harmonization between KEGG and MetaCyc......... 54
Table 3.8. Harmonization of specific reactions between KEGG and MetaCyc. 55
Table 3.9. Harmonization of generic reactions between KEGG and MetaCyc. 55
Table 3.10. Loose harmonization of reactions between KEGG and MetaCyc......... 56
Table 3.11. Summary of reaction harmonization between KEGG and MetaCyc......... 57
Table 3.12. Hardly interpretable compound pairs. 62
Table 3.13. First-round evaluation of atom mappings of compound pairs between KEGG RCLASS and RPAIR......... 62
Table 3.14. Second-round evaluation of atom mappings of compound pairs between KEGG RCLASS and RPAIR......... 64
Table 3.15. Evaluation of atom mappings between KEGG and MetaCyc. 65
Table 4.1. Different forms of objective function 80
Table 4.2. Different forms of a model selection estimator 86
Table 4.3. Common creation patterns for the moiety_modeling library........ 88
Table 4.4. Common patters for using the moiety_modeling as a command-line tool 89
Table 4.5. Common patterns for using ‘SAGA’ module as a library. 89
Table 4.6. Model selection results of UDP-GlcNAc isotopologue data........ 91
Table 4.7. Single-tracer ^{13}C moiety states and values for UDP-GlcNAc biosynthesis 92
Table 4.8. Multi-tracer ^{13}C/^{18}O moiety states and values for UDP-GlcNAc biosynthesis 93

Table 4.9. Multi-tracer $^{13}C/^{18}O$ moiety states and values for UDP-GlcNAc biosynthesis 94
Table 5.1. Objective functions. 104
Table 5.2. Model selection estimators. 105
Table 5.3. Comparison of optimization methods in model selection. 111
Table 5.4. Over optimization experiments with ‘SLSQP’ method. 112
Table 5.5. Over optimization experiments with SAGA-optimize method. 112
Table 5.6. Comparison of mode rank based on different model selection criteria. 113
Table 5.7. Model selection test with absolute difference objective function. 116
Table 5.8. Model selection test with square difference objective function. 116
Table 5.9. Model selection test with absolute difference of logs objective function. 117
Table 5.10. Model selection test with difference of AIC objective function. 117
Table 5.11. Model selection test with absolute difference objective function. 121
Table 5.12. Model selection test with square difference objective function. 122
Table 5.13. Model selection test with absolute difference of logs objective function.... 122
Table 5.14. Model selection test with difference of AIC objective function. 122
Table 5.15. Inclusion of less informative dataset can lead to failure in model selection. 125

LIST OF FIGURES

Figure 1.1. Stable isotope-resolved metabolomics (SIRM) pipelines. 7
Figure 1.2. Isotopologues and isotopomers. 8
Figure 2.1. Overview of major compound harmonization steps...................................... 18
Figure 2.2. Flow chart of atom coloring. 20
Figure 2.3. KEGG Compound C00047.. 21
Figure 2.4. Flow chart of atom coloring. 22
Figure 2.5. Correspondence between KEGG and MetaCyc compound entries with different molecular representations. 25
Figure 2.6. Example aromatic substructure that cannot be detected by Indigo. 27
Figure 2.7. Example of compound with same atom identifier for asymmetric atoms using an overly simplistic coloring approach. 28
Figure 2.8. Example of charge inconsistency of compound representations between databases. 29
Figure 2.9. Compound with incomplete KEGG aromatic atom types. 33
Figure 2.10. KEGG compound with S-containing aromatic ring. 33
Figure 2.11. Compound with different aromatic representations. 34
Figure 2.12. Example of inconsistent compound representations between KEGG and MetaCyc. 35
Figure 2.13. Representative compounds that cannot be distinguished by coloring identifier. 36
Figure 2.14. Example compounds gaining asymmetry after the addition of tight atom coloring information. 38
Figure 2.15. Compound with linear and circular representations. 39
Figure 2.16. Compound with different resonance structures. 39
Figure 3.1. Flowchart of compound and reaction harmonization. 46
Figure 3.2. Compound pair of generic and specific compounds. 50
Figure 3.3. Compound pair of generic compounds... 51
Figure 3.4. Example harmonized compound pair of compounds with inconsistent chemical details. 52
Figure 3.5. Reaction pair with mismatch of last EC number. 56
Figure 3.6. Reaction pair with missing 4th-level EC number designation. 57
Figure 3.7. Example of RCLASS entry. 60
Figure 3.8. RCLASS with combinatorial issue caused by multiple possible mappings of RDM descriptions. 61
Figure 3.9. Scatter plot of changed one-bond atom color fraction for KEGG RCLASS versus RPAIR atom mappings in paired compounds. 63
Figure 3.10. Example of atoms with interchangeable mappings. 63
Figure 3.11. Comparison of KEGG RCLASS and RPAIR description for compound pair C01255 and C02378. 64
Figure 3.12. Example of reaction pair with the same atom mappings. 65

Figure 3.13. Scatter plot of changed one-bond atom color fraction for KEGG versus MetaCyc atom mappings in paired reactions. 67
Figure 3.14. Example of MetaCyc reaction with incorrect atom mappings. 67
Figure 3.15. Example of KEGG reaction with incorrect mappings. 68
Figure 3.16. Atom mappings of MetaCyc reaction RXN-14940. 68
Figure 4.1. Example complex metabolite UDP-GlcNAc and associated expert-derived moiety model. 76
Figure 4.2. Workflow of the moiety modeling framework. 78
Figure 4.3. JSONized moiety model description. 79
Figure 4.4. A unified modeling language (UML) class diagram of a Moiety Model. 80
Figure 4.5. Organization of the moiety_modeling package represented with UML diagrams. 85
Figure 4.6. ‘SAGA-optimize’ package represented with a UML class diagram with dependencies. 87
Figure 4.7. The ‘moiety_modeling’ package command line interface. 88
Figure 4.8. Histograms of simulated intensities for the largest representative isotopologue. 95
Figure 4.9. Histograms of the acetyl and uracil optimized moiety state values derived from simulated datasets. 96
Figure 5.1. Workflow for Moiety Modeling. 103
Figure 5.2. Example complex metabolite UDP-GlcNAc and associated moiety models. 107
Figure 5.3. Optimized results for 6_G1R1A1U3 and 7_G0R3A1U3_g3r2r3_r4 models. 110
Figure 5.4. Error analysis in FT-ICR-MS datasets. 118
Figure 5.5. Error analysis in LC-MS datasets. 119
Figure 5.6. Error analysis in FT-ICR-MS datasets. 120
Figure 5.7. Error analysis in LC-MS datasets. 121
Figure 5.8. Comparison of the log optimization steps where model selection with different datasets begins to fail. 125
Figure 5.9. Comparison of the log of optimization steps where model selection with different datasets begins to fail with absolute difference of logs objective function. 126

LIST OF ADDITIONAL FILES

Supplemental Spreadsheet 2.1 Compound Pairs Across KEGG MetaCyc…. [CVS 427 KB]

Supplemental Spreadsheet 2.2 Inconsistency between KEGG and MetaCyc…[CVS 13 KB]

Supplemental Spreadsheet 3.1 Harmonized Reactions with Inconsistent Mappings……………………………………………………………………… [CVS 9 KB]

Supplemental Data 4.1 Moiety Model Description……….. ……..……......[JSON 36 KB]

Supplemental Data 4.2 Dataset for Moiety Modeling……….. ……….........[JSON 10 KB]

Supplemental Data 4.3 Optimization Results……….. ……….................[JSON 724 KB]

Supplemental Data 4.4 Analysis Results…..……….. ……….................[JSON 124 KB]

Supplemental Data 4.5 Optimization Parameters…..……….. ………......[JSON 822 KB]

Supplemental Table 4.1 Model Rank Comparison Between Moiety Modeling And GAIMS…………………….……... …………… …………..….........[DOC 22 KB]

Supplemental Table 4.2 Comparison of Optimized Model Parameters Between Moiety Modeling and GAIMS. ……….……….. ……………...……………….......[DOC 17 KB]

Supplemental Table 4.3 Comparison of Model Rank Between Different Repetitions. ……….……….. …………… ...………………........................[DOC 22 KB]

Supplemental Figure 4.1 Comparison of Optimized Model Parameters Between Moiety Modeling And GAIMS. ……….……….. ……………..…………….............[PDF 36 KB]

Supplemental Figure 4.2 Multi Tracer Optimization Results for Simulated Datasets. ……….……….. ……………..…………………...…….........[PDF 210 KB]

Supplemental Figure 4.3 Single Tracer Optimization Results for Simulated Datasets. ……….……….. ……………..…………………...…….........[PDF 195 KB]

CHAPTER 1. BACKGROUND

1.1 Metabolomics

Metabolomics can be generally defined as the comprehensive study of small-molecule metabolites (<1500 Da) in a biological system (cell, tissue or organism) in a given state[1]. It is marked as the new ‘omics’, joining genomics, transcriptomics, and proteomics to achieve a better understanding of global systems biology[2]. Metabolites represent a wide range of molecules, such as lipids, amino acids, nucleic acids, vitamins and carbohydrates, playing various functions in biological systems, including energy production, macromolecules synthesis, and pathway signaling[3]. The quantitative collection of all metabolites present in a cell or organism involved in metabolic reactions is called the metabolome[4]. Compared to other ‘omics’ studies, metabolomics has a number of unique advantages[5]. First, metabolites are considered to be the molecular endpoint of many biological processes and an important molecular midpoint of most other biological processes. As the downstream of transcriptome and proteome, the metabolome can reflect the functional level of a cell more appropriately[6], suggesting metabolic profiles are more proximal to a biological phenotype than either genetic or proteomic profiles. Also, changes in the metabolome are supposed to be magnified relative to proteome or transcriptome. Additionally, the metabolic profiles are combinational results of gene expression and environmental stresses[7]. Therefore, metabolomics can provide great insights into the interaction of a biological system with its living environment.

It is becoming increasingly clear that metabolomics research can help open up many previously inaccessible biological fields, including health, disease, nutrition, environment and agriculture[8, 9]. One important area affected by metabolomics is prokaryotic genome annotation. At least 30%-50% of genes in a bacterial genome are hardly or incorrectly annotated[10], and the function of 20% genes in the best understood bacteria genome is actually unknown[11]. Small molecule signatures would allow functional hypotheses and/or determination of those unknown genes. In addition, system wide analysis and understanding of metabolic processes have proven valuable in devising strategies of metabolic engineering for microorganisms[12]. More importantly, metabolomic studies can help with the identification of biomarkers for metabolic-related diseases[13] as well as the discovery and development of therapeutic agents[14, 15]. This will be discussed in more details in the next section.

1.2 Metabolic Reprogramming

Metabolic reprogramming is a critical feature in cancer and many other diseases. It has been recognized as one of the 10 hallmarks of cancer[16]. Back to 100 years ago, Dr. Warburg discovered that certain cancer cells predominantly produce energy through the glycolytic pathway rather than via the tricarboxylic acid (TCA) cycle even under normoxic conditions[17]. Despite the exact roles of metabolic reprogramming in cancers remaining unclear, advances in cancer metabolism research over the past decade have improved our understanding of how aerobic glycolysis and other metabolic alterations support the growth, proliferation and metastasis needs of cancer cells[18].

For example, glycolysis not only provides cancer cells with energy but also necessary precursors for biosynthesis (lipids, proteins, and carbohydrates). Several glycolytic metabolites, like glucose-6-phosphate, dihydroxyacetone phosphate, also take part in other metabolic pathways. Glucose-6-phosphate can be consumed by pentose phosphate pathway to synthesize nucleotides. Dihydroxyacetone phosphate can be involved in lipid production[19].

Apart from glycolysis, many cancer cells also rely on glutaminolysis for cellular bioenergetics. Glutaminolysis is composed of a set of reactions converting glutamine into glutamate, α-ketoglutarate, etc. These products are indispensable for TCA cycles in the cancer cells, and the intermediates of TCA cycles are the building blocks of lipids, amino acids and other important metabolites[20, 21].

Increased lipid metabolism is another critical feature of cancer metabolism. Several enzymes, such as ATP citrate lyase (ACLY), acetyl-CoA carboxylase (ACC), and fatty acid synthase (FASN), are involved in the multi-step lipid biosynthesis. Cancer cells need enhanced *de novo* fatty acid biosynthesis to meet their increased demands for lipids[22-24] FASN is upregulated in breast, prostate and other types of cancers[25-27]. In addition, cancer cells often have higher lipid accumulation as lipid droplets compared to normal cells[28].

In addition, upregulation of mitochondrial biogenesis, pentose phosphate pathway as well as other biosynthetic and bioenergetic pathways is also observed in certain cancer cells. The altered metabolism provides cancer cells with necessary energy as well as crucial materials to support rapid proliferation, survival, and invasion. Given the indispensable role of reprogrammed metabolism for cancers, it is promising to develop anti-cancer therapies targeting cancer bioenergetics. A lot of compounds have been studied and tested to selectively and effectively suppress metabolic enzymes that are essential to cancer cells.

One of the most popular anti-cancer metabolism therapeutic strategies is to inhibit enzymes that are most or even exclusively expressed in cancer cell, which will effectively kill tumors while causing little harm to normal cells. Inhibitors for Glutaminase 1 (GLS1), a glutaminase isoform that is highly upregulated in cancer cells, have proved to be effective in cancer treatment by blocking GLS1[29, 30]. Glycolysis inhibitors are also of great interest to many groups of researchers. For example, 2-deoxyglucose (2-DG) can block glycolysis by reversely inhibiting hexokinase, which is among the most advanced cancer metabolism inhibitors in clinical trials[31-33].

Furthermore, metabolism therapies can be combined with other therapies to achieve better efficacy. Metabolic adaptation is often involved in the resistance to cancer treatment[34]. For instance, increased glycolysis has been associated with the resistance of breast cancer cells to HER-2-targeting trastuzumab. Synergistical combination of trastuzumab with glycolysis inhibitors have proven to be effective in trastuzumab-sensitive and trastuzumab-resistant breast cancers both in vitro and in vivo[35].

Recent remarkable advances in metabolic reprogramming have inspired exciting research in the development of anti-cancer metabolism therapies, which is expected to play important roles in the future clinical oncology. Large-scale characterization of metabolic phenotypes for various cancers will further help clarify the mechanisms of metabolic disease and promote the development of novel and effective treatment strategies.

1.3 Metabolic Network and Metabolic Model

Metabolomics data represents complex interaction among metabolites on a global scale rather than a collective of individual components[36]. Therefore, analyzing data from a network perspective is more likely to capture meaningful information that can be missed via differential analysis of individual metabolites. At an abstract but computationally useful level, metabolic network can be represented as mathematical graphs, where each node is a specific metabolite and the edges describe the biotransformation pathways[37]. The representation of metabolic network as graph allows systematical investigation of the biological system via well-understood graph theoretical concepts and algorithms. Therefore, a high-quality reconstruction of metabolic network is of great interest to the community of scientific researchers working in the systems biology of metabolism.

Huge efforts have been devoted to the reconstruction, such as Recon 1[38], the Edinburgh Human Metabolic Network (EHMN)[39], and Recon 2[40]. These are genome-scale metabolic networks constructed by combining various sources of ‘omics’ and literature data. However, due to the limitations in our knowledge of the complexity of human genome, current reconstructions of the global metabolic network are not complete. For example, only about 1000 genes that are common to human Recon 1 and EHMN[41], promoting to the reconstruction of Recon 2 by harmonizing metabolic information in four different resources (EHMN[39], HepatoNet1[42], Ac-FAO module[43] and the human small intestinal enterocyte reconstruction[44]) to Recon 1[40]. In total, 7440 reactions and 2626 unique metabolites are collected in the Recon 2. Even though EHMN, Recon, and Recon 2 are limited to human metabolism, compared to the KEGG (11427 reactions and 18636 metabolites) and MetaCyc (17203 reactions and 20264 metabolites) repositories which cover known metabolism of many organisms, it can be seen that the reconstruction of comprehensive metabolic network still has a long way to go. However, efficient integration of metabolic data across different databases remains a big challenge. Non-uniform compound identifiers and reaction names are two main blocks for the integration.

Clearly, many reactions in a genome-scale metabolic network are not active under a particular condition[45]. When it comes to analysis and interpretation of a specific metabolic profile, an essential step is to derive a predictive metabolic model for the corresponding context based on the global metabolic network. Such context-specific metabolic models are proven to exhibit better explanatory power and predictive performance than generic models[46, 47]. On the other hand, a comprehensive metabolic network is the premise for deriving context-specific metabolic models with high biological interpretability.

1.4 Stable Isotope Resolved Metabolomics

Since a metabolite can participate in many interweaved reactions, especially for some hot spots like glutamate, it is practically impossible to discern the contributions of each pathway segment only based on the metabolome[48]. To overcome this issue, isotopic tracers (^{2}H, ^{13}C, ^{14}C, ^{15}N and others) can applied in the metabolic studies, where an isotopically enriched precursor is applied to a biological system such that its metabolic transformations can be traced via the labeled atoms[49, 50]. ^{13}C is the most frequently used stable isotope in cancer metabolism research. A lot of ^{13}C-enriched precursors, like several isotopomers of D-glucose, are commercially available. Double element isotope-labeled metabolites like [U-^{13}C,^{15}N]-glutamine are also common. The wide variety of stable isotope-labeled sources greatly facilitate the experimental design for raveling the intricate metabolic segments.

The typical stable isotope resolved metabolomics (SIRM) pipelines are shown in Figure 1.1[48]. Stable isotope tracers such as ^{13}C-labeled glucose are first administrated to the experimental targets via addition to the culture medium or through injection into the whole organism. The absorbed tracers will be used for synthesizing new metabolites. After certain period of incubation, the isotopically enriched metabolites will be extracted for quantification.

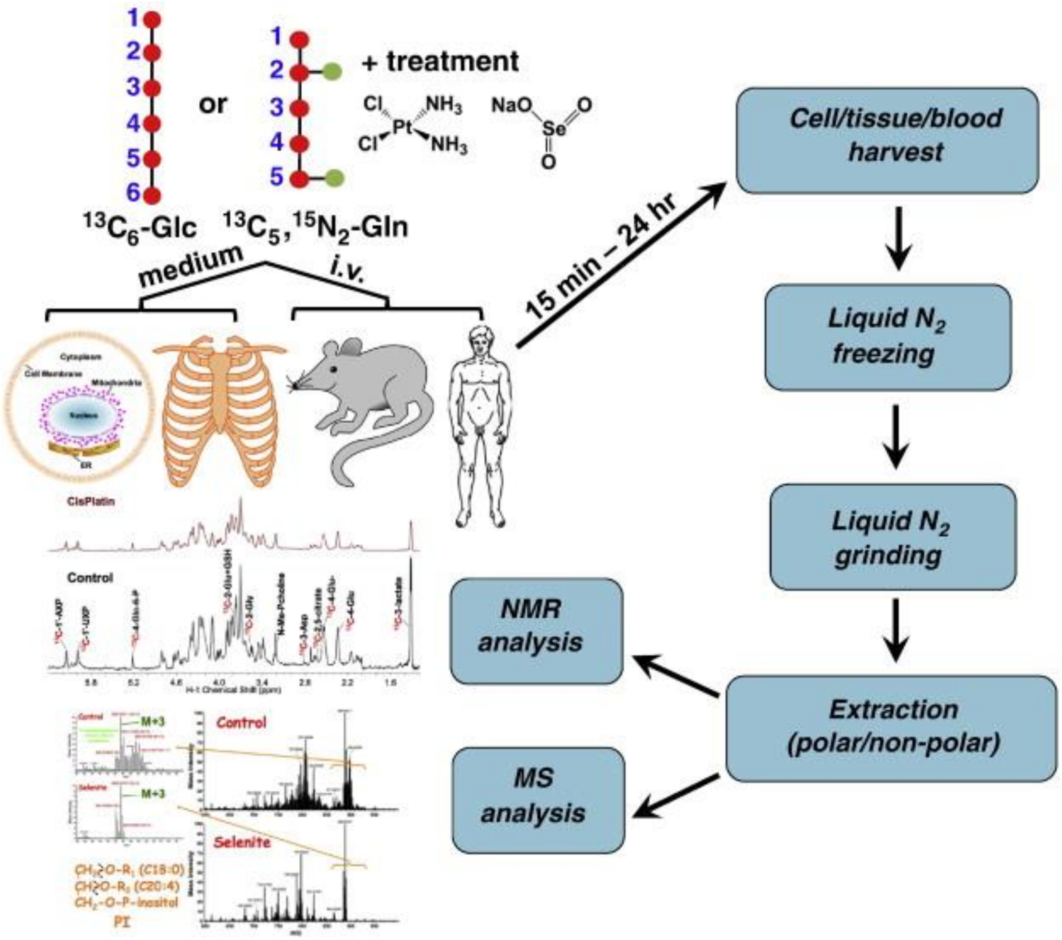


Figure 1.1. Stable isotope-resolved metabolomics (SIRM) pipelines.

Two chemical analytical features are derived from the incorporation of stable isotopes: isotopologues and isotopomers. Isotopologues are molecules that differ only in their isotopic composition[51]. Isotopomers are molecules having the same number of each isotope of each element but differing in their positions. Using alanine as an example (Figure

1.2), incorporation of ^{13}C can generate 4 sets of distinct isotopologues and 8 distinct isotopomers.

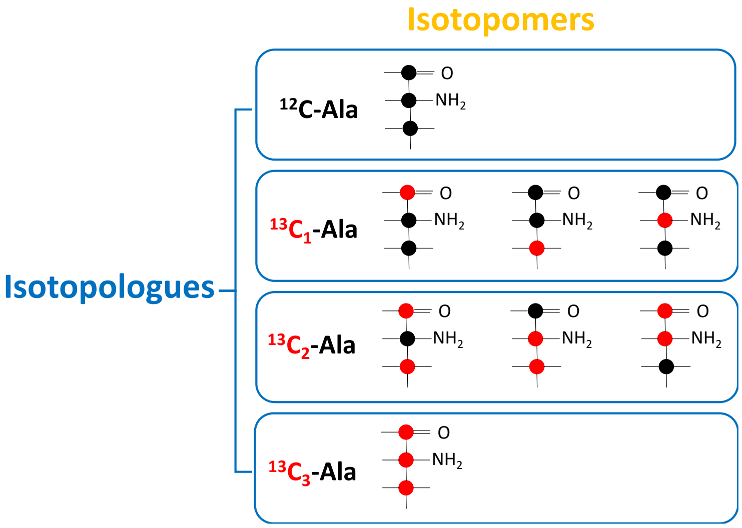


Figure 1.2. Isotopologues and isotopomers.

Nuclear magnetic resonance (NMR) and mass spectroscopy (MS) are quintessential analytical tools for metabolite detection and characterization. While NMR is extremely powerful for elucidating organic structures at the atom position-specific level (i.e. isotopomer), it is disadvantaged by its poor sensitivity. Only metabolites present in relatively high concentrations can be reliably detected and quantified by NMR[1]. MS can measure isotopologue-specific data with higher sensitivity; however, an isotopologue represents a set of mass-equivalent isotopomers.

When MS and NMR are applied in SIRM experiments, detailed sub-metabolite features representing isotopic flux through cellular metabolism can be detected on potentially thousands of metabolites in a biological system. However, it also brings another big challenge of how to properly interpret the sophisticated data.

1.5 Metabolic Flux Analysis

Merely detecting all the metabolites in a biological system cannot directly reveal the most biologically and dynamically relevant aspect of metabolism, metabolic fluxes (i.e. the flow of materials through metabolism) [52]. For example, we can hardly know what lead to the accumulation or depletion of the target metabolites just based on their concentration. Quantitative and qualitative knowledge of metabolic fluxes over a metabolic model can help understand the contribution of each pathway segment, which provide insights into the regulation of metabolism[53, 54]. The metabolic flux analysis (MFA) can be defined as a collective set of techniques to derive the rates of metabolic reactions [55]. Initially, MFA depended solely on balancing fluxes around metabolites in the predefined metabolic model. As we discussed above, the stoichiometric constraints cannot provide enough information to calculate the fluxes of interest, especially for complex systems. Therefore, we incorporate stable isotopes into the biological system so that metabolites with distinct isotope-specific patterns can be produced. The isotope measurements provide plenty of additional independent constraints for MFA[56].

Several software packages have been developed to facilitate flux analysis. Earlier metabolic flux analysis focused on local fluxes[57-62]. To serve this purpose, only a small subset of metabolic features was used to calculate the predefined analytic formulas. This type of approaches is mathematically simple and rapid with significant limitations. Since these formulas are derived with strong intuition and tacit assumptions, only a dozen can be applied to interpret the central metabolism of microbes with single carbon tracer. Recently, more efforts have been devoted to achieving systematic metabolic fluxes analysis in an iterative fitting manner [55, 56, 63-66]. All the measured metabolic features are used to estimate metabolic fluxes across the entire system. In this case, the fitting process of metabolic flux to detected metabolic data can be mathematically cumbersome. For complex system, it can take extensive computational time. Furthermore, this approach is not quite suitable for high-throughput analyses since it requires quantification of each sample individually. In addition, for complex metabolomics dataset, expert knowledge is required for quality control. The final results heavily depend on the correctness of the metabolic model, the assumptions made, as well as the precision of detected data.

Some machine learning algorithms also have been implemented for analyzing flux ratios based on ^{13}C-labeled data[67]. In the SUMOFLUX machine learning model, the measured ^{13}C isotope labeling patterns of the metabolites are the input features and the metabolic fluxes are the dependent variables to predict. Simulated data that fulfills the stoichiometric constraints of the metabolic network was generated to train the random forest model. Once the random forest predictor is constructed, it will be very efficient in estimating the metabolic fluxes for real data.

The methods mentioned above do facilitate the interpretation of complex metabolomics datasets. However, they also show some obvious limitations. First, they all highly depend on a predefined metabolic model, which is feasible for well-understood metabolism. For complex biological systems especially for non-model organisms, the metabolic model is far from complete. In addition, these tools can only deal with ^{13}C-data, which limits most of the current isotope tracer experiments to ^{13}C tracer. In Chapter 4, we developed the moiety modeling framework to address these problems.

1.6 Challenges in Interpreting SIRM Derived Metabolome

To better decipher the complex metabolomics data, both detailed modeling and quantitative analysis are required. As we discussed above, a comprehensive metabolic network is the premise for deriving context-specific metabolic models. In a traditional metabolic network, each node is a specific metabolite, where information at atom level is not representative. Here, we can see that the gap of descriptions between metabolic network (metabolite level) and SIRM metabolic profiles (atom level). In this case, metabolic features containing isotopic details derived from SIRM experiments cannot be fully utilized in the construction of context-specific metabolic models. To bridge this gap, we need a metabolic network at atom level, where each node represents an atom from a specific metabolite rather than the whole metabolite. However, currently there are no relatively complete atom-resolved databases of metabolic networks available for human metabolism.

In addition, we have discussed that it is difficult to construct a well-defined metabolic model for a complex system given our current level of knowledge of metabolism, particularly at the atom-resolved level. Since the current metabolic network is far from complete, multiple plausible metabolic models can be derived for a specific metabolic profile based on partial information. This causes another problem of model selection. However, this issue has not been fully studied yet. Here, we can see that issues exist in metabolic modeling for SIRM datasets at various stages.

The metabolic model is the cornerstone for downstream quantitative analysis. Our preliminary analysis indicated that a metabolomics dataset can have dramatically distinct metabolic fluxes interpretation under different metabolic models. Therefore, rigorous method should be developed to facilitate initial model construction and ensure optimal model selection.

1.7 Overview of Dissertation

Even though a large volume of high-quality metabolomics data has been generated recently, there is still a lack of computational tools and methods for analyzing and interpreting those data, especially for metabolomics datasets derived from SIRM experiments. The overall goal for this dissertation is to develop computational tools and methods for analyzing and interpreting metabolomics data at the atom-resolved level. Here, we focused on two major parts, construction of a relatively complete atom-resolved metabolic network and development of moiety model framework for SIRM datasets.

To construct an atom-resolved metabolic network, the very first step is to distinguish every node (i.e. an atom from a specific metabolite) in the network. In Chapter 2, we developed a neighborhood-specific coloring method for creating atom identifiers in a specific compound. In addition, compound identifiers derived from atom identifiers can be used for compound harmonization across various metabolic databases. Furthermore, we achieved hierarchical integration of metabolic reactions in KEGG and MetaCyc via an iterative combination of compound and reaction harmonization steps in Chapter 3.

After constructing an atom-resolved network, the next step is to derive metabolic models based on the metabolic profile and metabolic network and achieve metabolic flux analysis. In Chapter 4, we described a moiety modeling framework for deconvoluting SIRM isotopologue datasets. Usually, multiple models can be derived from a specific metabolic profile. This method helps with the selection of optimal model for the downstream analysis. In Chapter 5, we further analyzed how various factors (optimization method, optimizing degree, objective function and selection criterion) affect model selection.

CHAPTER 2. NEIGHBORHOOD-SPECIFIC COLORING METHOD FOR CREATING ATOM IDENTIFIER IN A COMPOUND

2.1 Introduction

Metabolic flux analysis is an essential approach to access metabolic phenotypes[53, 65] that requires both reliable metabolic profiles as well as reliable metabolic models[68-70]. Advances in analytical technologies like mass spectrometry (MS) and nuclear magnetic resonance (NMR) greatly contribute to the detection of thousands of metabolites from biofluids, cells, and tissues[71]. Application of those analytical techniques to stable isotope resolved metabolomics (SIRM) experiments facilitates production of high-quality metabolomics datasets capturing isotopic flux through cellular and systemic metabolism[48, 72]. Now, the challenge is to construct meaningful metabolic models from the corresponding metabolic profiles for downstream metabolic flux analysis. A metabolic network is usually represented by compounds connected via biotransformation routes[37]. Obviously, information at the atom level is not represented in such metabolic networks, making it impractical to derive appropriate metabolic models for SIRM datasets. Prior work demonstrated an atom-resolved metabolic network that included both central and intermediate metabolism in Escherichia coli that allowed atom-to-atom tracing[73, 74]. However, currently there are no relatively complete atom-resolved databases of metabolic networks available for human metabolism that can be used to trace individual atoms[72].

To construct an atom-resolved metabolic network, compounds and metabolic reactions with detailed documentation at the atom level are required. One approach is to reconstruct a hypothetical atom-resolved metabolic network from generalized reaction descriptions that are atom-specific[75]. However, it is unclear the level of validation and curation that such an approach would require to construct a reasonably accurate atom-resolved metabolic network for generating metabolic models usable in the analysis of SIRM datasets. An alternative is to use curated metabolic databases currently available, in particular the Kyoto Encyclopedia of Genes and Genomes (KEGG) and the MetaCyc metabolic pathway database. The popular molfile description of a compound is a text-based chemical table file format developed by MDL Information Systems and contains information about atoms, bonds, connectivity, and coordinates[76], which is available in most databases including KEGG and MetaCyc. For atom-resolved metabolic reactions, the KEGG reaction pair (RPAIR) database stores patterns of transformations occurring between two reactants in a single reaction[77]. In addition, MetaCyc contains direct atom mappings for every metabolic reaction[78]. Previous work only made use of atom mappings in either the KEGG RPAIR database[79, 80] or MetaCyc[81] for atom tracing. However, both databases cover metabolism for many common organisms, clearly indicating that these two databases are not independent of each other. A necessary first step for constructing a more comprehensive network is to integrate compounds from different databases without redundancy[82].

In an atom-resolved metabolic network, each node should include information at both molecule-specific and atom-specific levels. To name each atom in a compound, two rules need to be obeyed: 1) different atoms must have different identifiers; 2) and symmetric atoms must share the same identifier. Previous work used the atom index in the molfile associated with a compound in finding atom-specific metabolic pathways without considering molecular symmetry[80, 81]. Likewise, molecular symmetry has been ignored in prior atom-resolved metabolic network reconstruction approaches[75]. One group tried to assign a unique name for every atom in the compound based on the compound's International Union of Pure and Applied Chemistry (IUPAC) International Chemical Identifier (InChI) representation[83], which does not apply to this scenario since symmetric atoms can share the same routes in the metabolic network. Also, any InChI-based approach cannot handle the compound entries with R-groups. To our knowledge, no appropriate method has been previously published that provides each atom in a compound with a useful identifier for the explicit purpose of constructing an atom-resolved metabolic network, either because the identifier was not unique or because it was not consistent for symmetric atoms.

Here, we developed a novel neighborhood-specific graph coloring method that creates a unique identifier for each atom in a compound by expanding the type (color) of each atom based on its "neighborhood" of atoms (nodes) bonded (edges) to it. This approach is related to but distinct from atom typing performed in chemoinformatics, which determines an augmented atom type based on the local chemical environment, especially the directed bonded atoms[84]. Atom coloring creates an augmented atom type based on both directly and indirectly bonded atoms that are part of the graph neighborhood around a given atom. Moreover, the method is guaranteed to generate the same coloring identifier for symmetric atoms. Furthermore, compound coloring identifiers derived from the corresponding atom coloring identifiers can be used for compound harmonization across metabolic databases. In this context, only molecular configuration (i.e. changes requiring the breaking of a bond) and not molecular conformation (i.e. changes not requiring the breaking of a bond like a bond rotation) are considered in the generation of these identifiers. To our knowledge, this is the first attempt to create unique atom and compound identifiers that are consistent with respect to molecular symmetry and for the explicit purpose of harmonizing compounds across the KEGG and MetaCyc databases, ultimately to facilitate the construction of an integrated atom-resolved metabolic network.

2.2 Materials and Methods

2.2.1 Compound and metabolic reaction data

All data were downloaded directly from the corresponding databases. The KEGG COMPOUND and KEGG REACTION data is from the version available from KEGG on May 2019 via its REST interface. MetaCyc compound and reaction data is in version 23.0, downloaded from BioCyc.

2.2.2 Overview of major analysis steps

A compound can be represented as a graph where each node is an atom in the compound and each edge between atoms is a chemical bond. Based on the molfile, we are able to create a graph representation for the corresponding compound. After we detect the aromatic substructures for a compound, we can change the bonds within the aromatic substructures to aromatic type (molfile[76] bond designation 4). After curation of aromatic substructures and double bond stereochemistry, we performed atom coloring and validation to guarantee that symmetric atoms share the same identifier and different atoms have different identifiers. Each set of atom identifers for a compound is used to derive the corresponding compound coloring identifier. Finally, we detect corresponded pairs of compounds across two databases using ordered compound identifiers for each compound in each database. The flowchart of the overall compound harmonization procedure is shown in Figure 2.1.

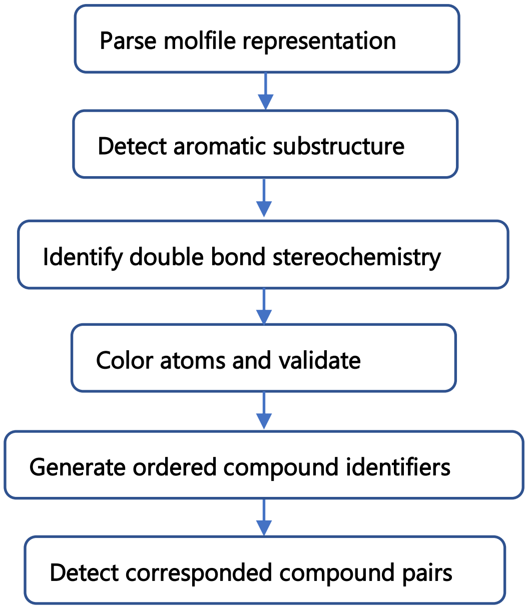

Parse molfile representation
Detect aromatic substructure
Identify double bond stereochemistry
Color atoms and validate
Generate ordered compound identifiers
Detect corresponded compound pairs

Figure 2.1. Overview of major compound harmonization steps.

2.2.3 Molfile parser

We used a modified ctfile Python 3 package[85] to parse a molfile into atom and bond blocks, and save them into the JavaScript Object Notation (JSON) format[86], facilitating access and modification.

2.2.4 Aromatic substructure detection

We used two methods in aromatic substructure detection. One is based on common subgraph isomorphism detection, and the other is an automatic aromatic atom detection method in Indigo packages[87]. In the KEGG database, aromatic atoms in a compound are specified in its KEGG Chemical Function (KCF) file[88]. Based on the aromatic atoms, we were able to extract the aromatic substructures present within a compound, and then saved every substructure into a separate molfile. If several aromatic rings are connected, we would fuse them together as one substructure. Then, we built a set of all aromatic substructures detected from the KEGG compounds without duplication. Furthermore, we manually inspected the set of aromatic substructures to ensure data quality. With this curated set of reference aromatic substructures, we tested each compound in a database for the presence of any of these aromatic substructures using the BASS method[89]. We analyzed KEGG to validate the aromatic substructure detection method itself. Then, we analyzed MetaCyc and labeled the bonds of detected aromatic substructures as aromatic. Furthermore, valid aromatic substructures in MetaCyc compounds can be detected by Indigo and other IDs. Finally, we created 366 KEGG-derived and 21 MetaCyc-derived aromatic substructures in the reference aromatic substructure set.

2.2.5 Identification of double bond stereochemistry

The C=C double bond stereochemistry is not clearly specified in the molfile in both databases. To distinguish cis/trans stereoisomers, we adopted a method for automated identification of double bond stereochemistry[90]. This method requires fully hydrogenated compounds. Therefore, we first used Open Babel[91] to add hydrogen atoms for every compound, and then performed the calculation.

2.2.6 Neighborhood-specific graph coloring method

Our neighborhood-specific graph coloring method is based on a breadth first search algorithm[92]. This method names each atom based on its own and neighbors' chemical information, which can include atom type, atom charge, atom stereochemistry, isotope, bond type, and bond stereochemistry. The method is flexible in adjusting the chemical information included in the atom coloring. A flowchart of the graph coloring method is shown in Figure 2.2. First, the method named each atom with its own chemical information, which will be saved as the 0_layer identifier and the start of the current atom identifier. Then, the method builds a dictionary that relates each atom with its 0_layer identifier and directly linked atoms. Directly bonded atoms of each atom are initialized as its neighbors.

The method continues to extend the name of each atom, adding information about its neighbors into the 0_layer dictionary to its current identifier, and updating neighbors with neighbors' neighbors that have not been used in extending the name of that atom. The method first repeats this process 3 times for all the atoms to avoid early stopping that can lead to non-unique compound coloring identifiers. Then, the method checks if an atom has a unique identifier. Atom naming will continue for those atoms that still share the same identifiers with other atoms until all the atoms in the compound have been used in name extension. Finally, the current name for each atom will be its coloring identifier. Compound C00047 in the KEGG database (Figure 2.3) is used as an example to illustrate how the method works (Table 2.1).

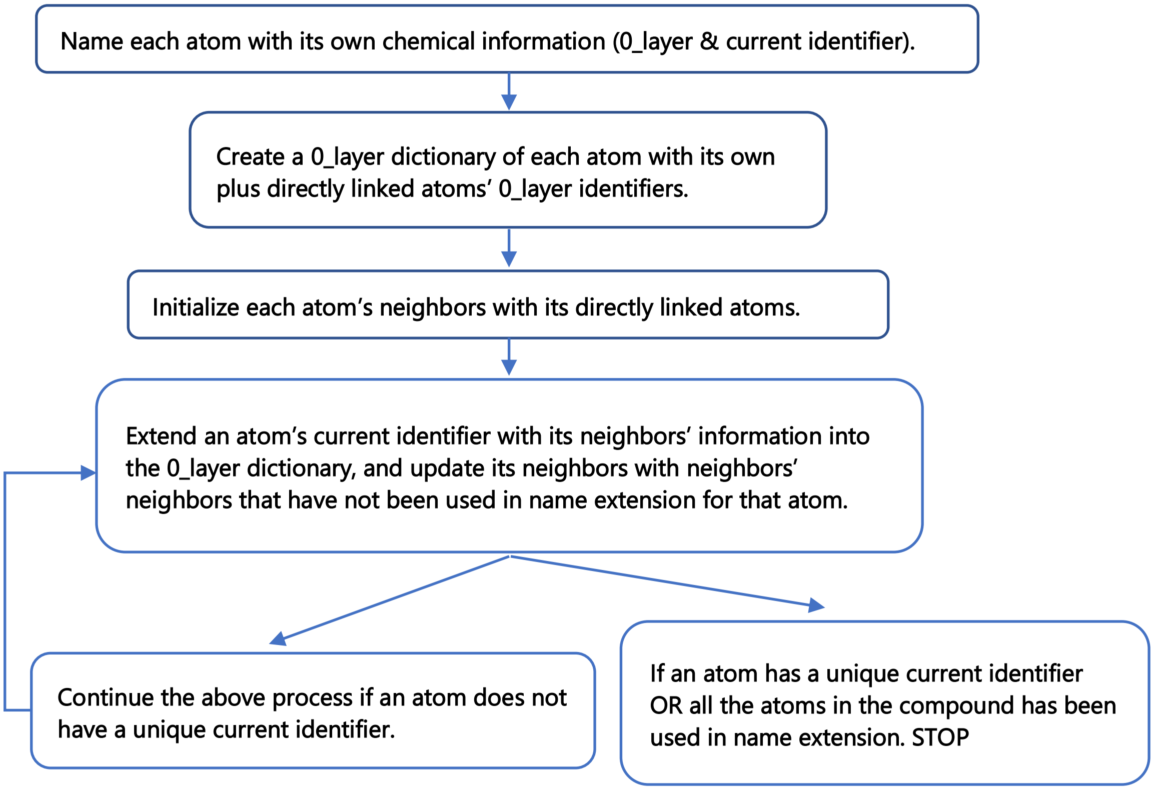

Name each atom with its own chemical information (0_layer & current identifier).
Create a 0_layer dictionary of each atom with its own plus directly linked atoms' 0_layer identifiers.
Initialize each atom's neighbors with its directly linked atoms.
Extend an atom's current identifier with its neighbors' information into the 0_layer dictionary, and update its neighbors with neighbors' neighbors that have not been used in name extension for that atom.
Continue the above process if an atom does not have a unique current identifier.
If an atom has a unique current identifier OR all the atoms in the compound has been used in name extension. STOP

Figure 2.2. Flow chart of atom coloring.

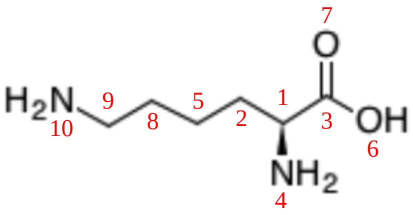


Figure 2.3. KEGG Compound C00047.

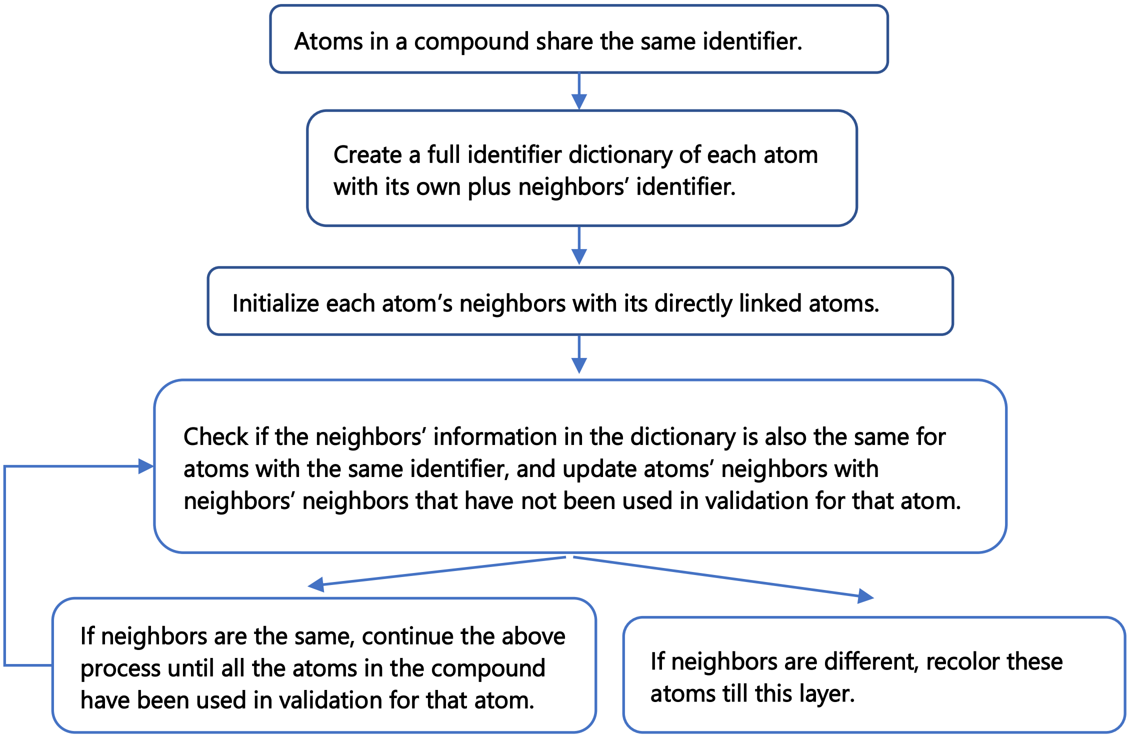
Table 2.1. Generation of atom identifiers for compound C00047 via graph coloring method.

Round	Atom identifier	Atom index
1	C	1, 2, 3, 5, 8, 9
	N	4, 10
	C	6, 7
2	C(C(C,1)(C,1)(N,1))	1
	C(C(C,1)(C,1))	2, 5, 8
	C(C(C,1)(O,1)(O,2))	3
	N(N(C,1))	4, 10
	O(O(C,1))	6
	O(O(C,2)	7
	C(C(C,1)(N,1))	9
3	C(C(C,1)(C,1)(N,1))(C(C,1)(C,1)C(C,1)(O,1)(O,2)N(C,1))	1
	C(C(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)(N,1))	2
	C(C(C,1)(O,1)(O,2))(C(C,1)(C,1)(N,1)O(C,1)O(C,2))	3
	N(N(C,1))(C(C,1)(C,1)(N,1))	4
	C(C(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1))	5
	O(O(C,1))(C(C,1)(O,1)(O,2))	6
	O(O(C,2))(C(C,1)(O,1)(O,2))	7
	C(C(C,1)(C,1))(C(C,1)(C,1)C(C,1)(N,1))	8
	C(C(C,1)(N,1))(C(C,1)(C,1)N(C,1))	9
	N(N(C,1))(C(C,1)(N,1))	10

Only chemical information of atom type and bond type is included in atom naming. The first three rounds of naming are shown above.

2.2.7 Atom coloring validation and recolor

The atom coloring validation and recoloring are also based on a breadth first search algorithm. The atom coloring validation flowchart is shown in Figure 2.4. For atoms with the same coloring identifier, we checked if neighbors of these atoms are also the same, layer by layer, until all the atoms in the compound have been tested. Then, the recoloring method will correct atoms with the same identifier that don't have the same neighbors. The recoloring process is similar to the graph coloring method. Instead of creating a 0_layer identifier dictionary, we will use a full identifier dictionary. In addition, we only color atoms to where they have different neighbors to distinguish between them.



Atoms in a compound share the same identifier.
Create a full identifier dictionary of each atom with its own plus neighbors' identifier.
Initialize each atom's neighbors with its directly linked atoms.
Check if the neighbors' information in the dictionary is also the same for atoms with the same identifier, and update atoms' neighbors with neighbors' neighbors that have not been used in validation for that atom.
If neighbors are the same, continue the above process until all the atoms in the compound have been used in validation for that atom.
If neighbors are different, recolor these atoms till this layer.

Figure 2.4. Flow chart of atom coloring.

2.2.8 Creation of compound coloring identifiers based on atom coloring identifiers

Once we create the identifiers for all the atoms in a compound, we can combine the number of atoms with the same identifier along with the atom coloring identifier. We sorted all the substrings, and then concatenated them together to form an ordered coloring identifier for the compound. The formulation is shown in Equation 1, which represents the order of string concatenation with n_k being the number of atoms with coloring a_k. The parenthesis and bracket characters are included in the resulting string.

$$Compound\ color\ identifier\ =\ (n_1)[a_1](n_2)[a_2](n_3)[a_3]\ \ldots.(n_k)[a_k] \qquad (1)$$

2.2.9 Prediction of possible compound correspondence via metabolic reactions

We connected each compound with the metabolic reactions it is a part of. For matched compounds between KEGG and MetaCyc, we tested if the compound shares at least one metabolic reaction indicated by the EC number in both databases.

2.3 Results

2.3.1 Overview of KEGG and MetaCyc databases

The numbers of compounds and atom-resolved reactions in KEGG and MetaCyc databases are summarized in Table 2.2. MetaCyc has 1.09 times as many compound entries as KEGG and 1.53 times as many atom-resolved reaction entries.

Table 2.2. KEGG and MetaCyc databases

Data types	**KEGG**	**MetaCyc**	**[a]MetaCyc/KEGG**
Compounds	18636	20264	1.09
Reactions	11427	17203	1.51
Atom-resolved reactions	10282	15909	1.53

[a]Ratio of MetaCyc entries to KEGG entries

To initially evaluate the level of overlap between KEGG and MetaCyc databases, we used existing identifiers in each database to find the correspondences between KEGG and MetaCyc compounds. Not all compounds in either database have all the chemical identifiers listed in Table 2.3. Some compounds in MetaCyc have a direct identifier to the corresponding KEGG compound[81]. We can see that the number of matched compounds (correspondences) detected by different identifiers are not consistent, with the total less than 5700. We also generated InChI identifiers based on the molfile provided for each entry in each database using Open Babel[91], which utilizes the InChI software library provided by the InChI Trust[93]. We were able to generate 16530 InChI from KEGG and 15765

InChI from MetaCyc, providing 3103 correspondences. When combined with ChEBI and KEGG Compound IDs, a total of 5929 consistent correspondences were detected. Two issues may appear when applying these identifiers to compound integration across various databases. On the one hand, there is no easy way to check if some correspondences are missing. Besides, it is difficult to tell if the results generated by those identifiers are correct, since errors can exist in every database[83, 94]. Such errors are illustrated by the 964 out of 13216 KEGG compound entries with InChI that are inconsistent with the InChI generated from their associated molfile, representing 7.3% of the InChI-containing entries in KEGG. Likewise, 55 out of 15076 MetaCyc compound entries have InChI that are inconsistent with the InChI generated from their associated molfile, representing 0.4% of the InChI-containing entries in MetaCyc.

Table 2.3. Correspondences between KEGG and MetaCyc compounds

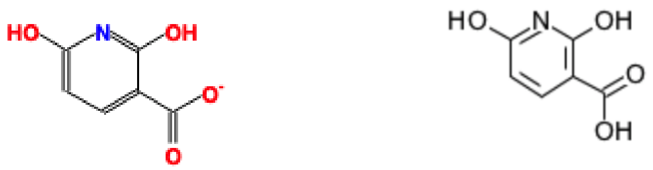
Identifiers	**KEGG**	**MetaCyc**	**Correspondences**
InChI	13216 (70.9%)	15076 (74.4%)	2336
ChEBI	15353 (82.4%)	8404 (41.5%)	3106
KEGG	18636 (100%)	5402 (26.7%)	5402
Either-ID	18636 (100%)	15216 (75.1%)	5681

InChI: IUPAC International Chemical Identifier.
ChEBI: Chemical Entities of Biological Interest.

Therefore, a reliable systematic naming method for chemical compounds that solves problems at the atom-level as well as the compound-level is required for constructing an atom-resolved metabolic network. Towards this end, we have developed a neighborhood-specific graph coloring method that derives unique identifiers for atoms as well as compounds.

2.3.2 Aromatic substructure detection

The neighborhood-specific graph coloring method is very sensitive to the specific structural representation. Moreover, aromatic substructures are not consistently represented in both databases. Instead of being directly labeled as an aromatic bond type, single and double bonds are used alternatively to depict the aromatic substructure. CPD-6962 in MetaCyc has a direct reference KEGG compound C15523 (Figure 2.5). We can see that the positions of double bonds and single bonds within the benzene ring vary between these two representations, which can lead to two different sets of atom identifiers. Therefore, we needed to ensure that compound representation is consistent across databases so that each compound will have a single set of atom identifiers. In this case, we first detect the aromatic substructures in all compounds from both databases, and change the single and double bonds within the aromatic substructure to aromatic bond.



MetaCyc: CPD-6962

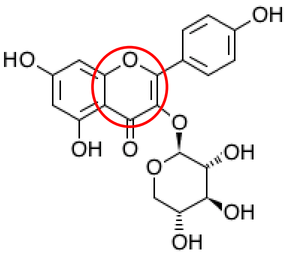
KEGG: C15523

Figure 2.5. Correspondence between KEGG and MetaCyc compound entries with different molecular representations.

Two independent aromatic detection methods were used in aromatic substructure detection: our Biochemically Aware Substructure Search (BASS) method[89] which uses neighborhood-specific graph coloring[95] to greatly improve subgraph isomorphism detection[96] and the aromatic detection facilities in the Indigo package[87]. First, we compared the aromatic substructures derived by these two methods. As shown in Table 2.4, Indigo appears more conservative than BASS in detecting aromatic substructures, detecting roughly 85% of what the BASS method does. Figure 2.6 shows an example aromatic substructure that can be missed by Indigo. We assume that Indigo cannot detect aromatic substructures with a double bond connected to atoms outside of the ring. This is not surprising, since BASS leverages the curated set of aromatic substructures in KEGG and has very high precision (99.9%) in the detection of aromatic substructures in KEGG compounds, while Indigo uses a set of simplified aromatic detection heuristics along with hard-coded algorithmic limitations of ring sizes being searched. However, we had concerns that some valid aromatic substructure representations in MetaCyc compounds may not exist in the reference aromatic substructure set derived from the KEGG database, which would be missed by the BASS method. This was confirmed by Indigo detecting 30 additional MetaCyc compounds with aromatic substructures not detected by the BASS method. Therefore, we combined the KEGG aromatic substructures with additional Indigo-detected substructures from MetaCyc. By using both methods, we were able to detect aromatic substructures in about half of the compounds in each database (Table 2.5). When an aromatic substructure was detected, all bonds for the aromatic substructure were changed to an aromatic bond type and the modified molfile was saved. All analyses were performed on a desktop computer with a i7-6850K CPU (6-core with HT), 64GB RAM, and 512GB solid state drive. On this hardware, the aromatic substructure detection took less than 5 minutes for KEGG and roughly 15 minutes for MetaCyc in terms of execution time.

Table 2.4. Incomplete detection of aromatic substructures by BASS and Indigo

Databases	BASS	Indigo
KEGG	0	~1500
MetaCyc	30	~1700



KEGG: C20727

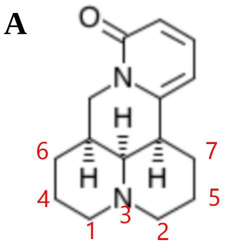
Figure 2.6. Example aromatic substructure that cannot be detected by Indigo.

Table 2.5. Compounds with aromatic substructure

Databases	**Count**
KEGG	9204 (49.4%)
MetaCyc	8292 (40.9%)

2.3.3 Generating identifiers for atoms using a graph coloring method

Since symmetric atoms share the same neighbors, the graph coloring method is guaranteed to create the same identifier for them. Our concern is whether atoms with the same identifier are actually symmetric. In our graph coloring method, we only include 0_layer identifiers in atom coloring to avoid long name strings. In some extreme cases, this shortcut can assign the same identifier to atoms that are asymmetric. An example is shown in Figure 2.7 A. We can see that this compound does not contain any symmetric atoms. Without considering the upper right ring, the bottom two rings are symmetric. Therefore, atoms 1 and 2 have the same 0_layer identifier, which is the same for atom pairs 4 & 5 and 6 & 7. In addition, once atoms 1 and 2 reach atom 3, they will share the same route to the upper right substructure. Finally, atoms 1 and 2 will share the same coloring identifier (Figure 2.7 B) even though they are not symmetric. To deal with this problem, atom coloring validation and recoloring is performed. We can see that atoms 1 and 2 have distinct identifiers after recoloring (Figure 2.7 C).



KEGG: C10782

B

```
'C(C(C,1)(N,1))(C(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)
(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)C(C,1
)(C,1)(N,1)C(C,1)(C,1)(N,1)C(C,1)(C,4)(N,4)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1
)C(C,4)(C,4)N(C,1)(C,4)(C,4)N(C,1)(C,4)(C,4))(C(C,1)(C,4)(N,4)C(C,1)(C,4)(N,4)C(C,1)(N,1)C(C,4)(C,4)C(C,4)(N,4)(O,1))(C(C,4)
(C,4)C(C,4)(C,4)C(C,4)(C,4)N(C,1)(C,4)(C,4)O(C,1))(C(C,4)(C,4)C(C,4)(N,4)(O,1)C(C,4)(N,4)(O,1))': [1, 2]
```

C

```
'C(C(C,1)(N,1))(C(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(
C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)C(C,1)
(C,1)(N,1)C(C,1)(C,1)(N,1)C(C,1)(C,4)(N,4)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)
C(C,4)(C,4)N(C,1)(C,4)(C,4)N(C,1)(C,4)(C,4))(C(C,1)(C,4)(N,4)C(C,1)(C,4)(N,4)C(C,1)(N,1)C(C,4)(C,4)C(C,4)(N,4)(O,1))(C(C,4)(
C,4)C(C,4)(C,4)C(C,4)(C,4)N(C,1)(C,4)(C,4)O(C,1))(C(C,4)(C,4)C(C,4)(N,4)(O,1)C(C,4)(N,4)(O,1))(C,1)(N,1)C(C(C,1)(C,1))(C(C,1
)(C,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)(N,1)C(C,1)(C,1)(N,1)C(C,1)(
N,1)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,
1)(C,4)(C,4))(C,1)(C,1)N(N(C,1)(C,1)(C,1))(C(C,1)(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,
1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)(N,1)C(
C,1)(C,1)(N,1)C(C,1)(C,4)(N,4)C(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1
)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,4)(C,4)N(C,1)(C,4)(C,4)N(C,1)(C,4)(C,4))(C,1)(C,1)(C,1)': [1]
```

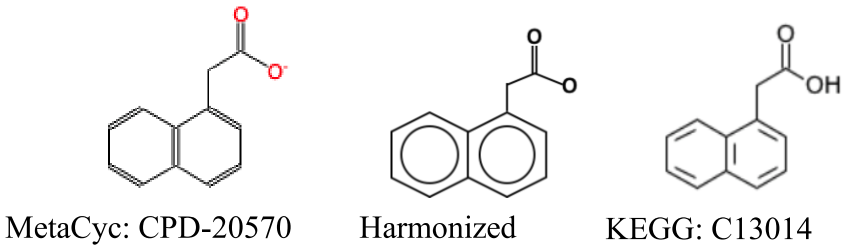
```
'C(C(C,1)(N,1))(C(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(
C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)C(C,1)
(C,1)(N,1)C(C,1)(C,1)(N,1)C(C,1)(C,4)(N,4)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)
C(C,4)(C,4)N(C,1)(C,4)(C,4)N(C,1)(C,4)(C,4))(C(C,1)(C,4)(N,4)C(C,1)(C,4)(N,4)C(C,1)(N,1)C(C,4)(C,4)C(C,4)(N,4)(O,1))(C(C,4)(
C,4)C(C,4)(C,4)C(C,4)(C,4)N(C,1)(C,4)(C,4)O(C,1))(C(C,4)(C,4)C(C,4)(N,4)(O,1)C(C,4)(N,4)(O,1))(C,1)(N,1)C(C(C,1)(C,1))(C(C,1
)(C,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)(N,1)C(C,1)(C,1)(N,1)C(C,1)(
C,4)(N,4)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,4)(C,4)N(C,1)(C,1)(C,1)N(C,1
)(C,1)(C,1)N(C,1)(C,4)(C,4))(C,1)(C,1)N(N(C,1)(C,1)(C,1))(C(C,1)(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1
)(C,1)(C,1)C(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1)N(C,1)(C,1)(C,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)C(C,
1)(C,1)(N,1)C(C,1)(C,1)(N,1)C(C,1)(C,4)(N,4)C(C,1)(N,1)C(C,1)(N,1)C(C,1)(N,1))(C(C,1)(C,1)C(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(
C,1)(C,1)C(C,1)(C,1)(C,1)C(C,1)(C,1)(C,1)C(C,4)(C,4)N(C,1)(C,4)(C,4)N(C,1)(C,4)(C,4))(C,1)(C,1)(C,1)': [2]
```

Figure 2.7. Example of compound with same atom identifier for asymmetric atoms using an overly simplistic coloring approach.
A) KEGG compound C10782; B) The atom identifiers for atoms 1 and 2 before symmetry validation; C) The atom identifiers for atoms 1 and 2 after symmetry curation.

We validate symmetry after a first round of coloring, recolor the compound if asymmetric atoms have the same identifier, and verify symmetry again. After this coloring-validation-recoloring-validation process, our results indicated that the graph coloring method is able to generate the same identifier for symmetric atoms and asymmetric atoms have unique identifiers for all compounds in both KEGG and MetaCyc databases.

2.3.4 Detection of correspondences between KEGG and MetaCyc compounds via coloring identifiers.

After creating a single set of atom identifiers for each compound, we were able to derive ordered compound coloring identifiers at different levels of chemical specificity, which can be used to harmonize compounds across databases. Since KEGG and MetaCyc can include different numbers of H (hydrogen atoms) in the molfile, we exclude H in coloring at this point. We first tried to include information of bond stereochemistry, atom charge, atom stereochemistry, and isotope stereochemistry in coloring to ensure each compound has a unique name. With the relatively specific coloring identifiers, 1763 correspondences between KEGG and MetaCyc compounds can be detected (see Table 2.6), which is not satisfactory compared to 5681 pairs discovered by other identifiers (e.g. KEGG, CHEBI, and InChI as shown in Table 2.3). This lack of correspondence is due to the inconsistencies in bond stereochemistry, atom charge, atom stereochemistry, and isotope stereochemistry information between these two databases. An example shown in Figure 2.8, where compound CPD-20570 in MetaCyc has a direct reference to KEGG compound C13014.



MetaCyc: CPD-20570 Harmonized KEGG: C13014

Figure 2.8. Example of charge inconsistency of compound representations between databases.
The middle harmonized compound representation enables loose coloring that facilitates compound harmonization.

For the following analysis, we only included information of atom type and bond type to keep the backbone of a compound in atom naming. It took less than 10 minutes of execution time on a desktop computer with a i7-6850K CPU (6-core with HT), 64GB RAM, and 512GB solid state drive to generate these coloring identifiers for all compound entries in the KEGG and MetaCyc databases. About 8865 correspondences between KEGG and MetaCyc are detected (see Table 2.6 and spreadsheets in supplementary material), and 5451 of them can be confirmed by other identifiers. With both tight and loose compound coloring identifiers, about 95.95% compounds pairs detected by other chemical IDs can be discovered. We manually checked the compound pairs that were discordant with other chemical IDs and found that none of them are caused by an inconsistency between the coloring identifier and the compound representation. The question then becomes how to validate the remaining 3414 possible pairs. Matched compounds are supposed to take part in the same metabolic reactions. The Enzyme Commission (EC) number is a numerical classification scheme for enzymes, playing a key role in classifying enzymatic reactions[97, 98]. We expected matched compounds to take part in metabolic reactions with similar EC numbers.

Table 2.6. Matched compounds detected by the compound coloring identifiers

Identifiers	**Color matched pairs**	**ID verified pairs**
Tight coloring identifier	1763	1448
Loose coloring identifier	8865	5451

Then, we analyzed the metabolic reactions in KEGG and MetaCyc databases (see Table 2.7). We can see that the documentation of EC number in KEGG is more complete compared to MetaCyc, but the number of metabolic reactions in MetaCyc is 50% larger than in KEGG. Around 80% of reactions in both databases can be related to at least a 3-leveled EC number.

Table 2.7. Matched compounds detected by the compound coloring identifiers

EC types	KEGG (count / percentage)	MetaCyc (count / percentage)
No EC	1263 / 11.05%	3427 / 19.92%
1-leveled EC	24 / 0.21%	11 / 0.06%
2-leveled EC	126 / 1.10%	67 / 0.39%
3-leveled EC	1081 / 9.46%	2958 / 17.19%
4-leveled EC	8933/ 78.17%	10740 / 62.43%

Next, we tested how well EC numbers work in the validation of correspondences between KEGG and MetaCyc compounds (See Table 2.8). We first identified color-harmonized pairs that both take part in some reactions in their respective database. There are 4227 ID confirmed pairs and 2292 possible pairs involved in the metabolic reactions. We further investigated if those pairs participate into the same type of reaction indicated by EC number. If we used the first 3 levels of the sectioned EC number as the standard, 3810 (90.13%) ID-confirmed pairs are verified by 3-leveled EC numbers. In addition, 3580 of them can be further confirmed by 4-leveled EC numbers. Furthermore, 1848 and 1540 possible pairs are confirmed by 3-leveled and 4-leveled EC numbers, respectively. These results suggest that EC numbers may be useful in validating possible pairs that have slight coloring deviations. All of the detected compound pairs are list in Supplementary Spreadsheet 2.1.

Table 2.8. Correspondences between KEGG and MetaCyc compounds verified by reactions

Conditions	ID-confirmed pairs	Possible pairs
Pairs not in reaction	1224	1122
Pairs in reactions	4227	2292
Verified by 3-leveled EC	3810	1848
Verified by 4-leveled EC	3580	1540

2.3.5 Compound representation errors and issues detected in the KEGG and MetaCyc databases.

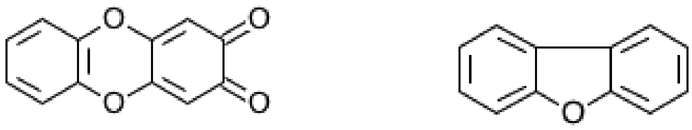
When harmonizing compounds between KEGG and MetaCyc databases, we found that there are various compound representation issues and errors existing in both databases, which can be grouped into several categories like mismatch between compound image and molfile, incorrect cross-referencing, and different bonds attached to metal ions. Here, we give a brief description with some examples, and all the detected inconsistency is documented in Supplementary Spreadsheet 2.1.

2.3.5.1 Incomplete KEGG aromatic atom types.

KEGG atom types annotate every atom in every compound of the KEGG Compound database. The KEGG atom type of an atom maps that atom to a unique chemical substructure and these substructures often map to functional groups (e.g. the atom type “O1a” represents an oxygen of a hydroxyl group). However, the set of KEGG atom types are not complete, especially with regard to aromatic heterocycle atoms. In particular, there are no oxygen and sulfur aromatic KEGG atom types defined, which prevents full automation of aromatic substructure determination based on KEGG atom type alone. We used a simple heuristic method (i.e a simple deterministic decisioning approach) to consider oxygen and sulfur atoms as aromatic when they are part of a ring where all other carbon and nitrogen atoms are labeled as aromatic, based on KEGG atom types. But this aromatic substructure detection approach has limitations that requires some manual inspection, as highlighted in Figure 2.9. KEGG Compound entry C03861 contains a 1,4-

dioxin flanked by aromatic rings. The 1,4-dioxin is not aromatic. In a counter-example, KEGG Compound entry C07729 contains an aromatic pyridine substructure flanked by benzyl rings. The presence of both examples illustrates why aromatic substructure detection cannot be fully automated based on the current set of KEGG aromatic atom types. In addition, Figure 2.10 shows a KEGG compound with an S-containing aromatic ring.

As an aside, the quinoid fragment in KEGG Compound entry C03861 is likely mislabeled as aromatic, since quinoid fragments are standardly antiaromatic[99]. This quinoid fragment was likely mislabeled as aromatic due to the whole three-ring structure obeying Huckel’s rule. While we treated KEGG-identified aromatic substructures as completely correct, this example does indicate the presence of some error in KEGG’s aromatic substructure detection methods. Comparison of Indigo to KEGG may provide a means for detecting suspect KEGG aromatic substructures, but a manual inspection of all suspect substructures is not practical, especially from an automated analysis perspective. Moreover, aromatic mislabeling should not impact compound harmonization if applied consistently across databases.



KEGG: C03861 KEGG: C07729

Figure 2.9. Compound with incomplete KEGG aromatic atom types.
The middle ring of compound C03861 (left) is not aromatic while the middle ring of compound C07729 (right) is aromatic.

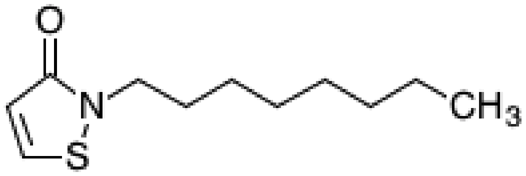
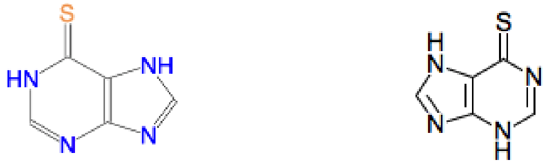


Figure 2.10. KEGG compound with S-containing aromatic ring.

2.3.5.2 Inconsistent compound representations.

Using ID-based compound harmonization, we found that there are about 10 MetaCyc compounds that contain valid aromatic substructures not detected by either the BASS or Indigo methods (Figure 2.11). To deal with this problem, we incorporated those valid aromatic substructures into the reference aromatic substructure set.

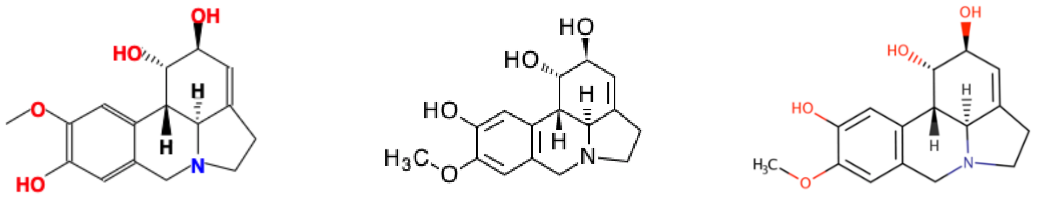


MetaCyc: CPD-15916 KEGG: C02380

Figure 2.11. Compound with different aromatic representations.
These two corresponding compound entries across KEGG and MetaCyc have two different aromatic substructure representations.

2.3.5.3 Incorrect cross-referencing

There are some matched compounds detected by other identifiers that don’t have the same coloring identifier. Compound CPD-19437 in MetaCyc has a direct reference to KEGG compound C12187, but their coloring identifiers are different (see Figure 2.12). We can see that the compound representation in MetaCyc is not consistent with its counterpart in KEGG. In addition, CPD-19437 and C12187 have the same ChEBI reference compound 32074, and the representation in ChEBI is the same with that of KEGG, suggesting the representation in MetaCyc may be incorrect.



MetaCyc: CPD-19437 KEGG: C12187 ChEBI: 32074

Figure 2.12. Example of inconsistent compound representations between KEGG and MetaCyc.

2.3.6 Estimating the error rate of the graph coloring method.

2.3.6.1 Ambiguous coloring identifiers.

During the compound harmonization process, tight atom and compound coloring was loosened (see Figure 2.8 for an example), keeping only atom type and bond type in the atom coloring for the final steps in compound harmonization. With the loose coloring, multiple compounds in one database can have the same coloring identifier. We first tested if a compound can have a unique coloring identifier when all information is included in the atom coloring with hydrogen (H) atoms excluded (Table 2.9). Here, we did not count compounds with a generic R group representing ambiguous functional groups and substructures; however, the results that include all compounds are described in Table 2.10. Several types of compounds cannot be distinguished by the tight coloring identifier except for those duplicates (Figure 2.13). When we only include atom type and bond type in the atom coloring, many more compounds share the same coloring identifier. After compound harmonization, we are able to detect compounds with the same coloring identifier from the source database.

Table 2.9. Compounds with the same coloring identifier, excluding R groups

Databases	Tight coloring identifier	Loose coloring identifier
KEGG	99 (0.5%)	968 (4.8%)
MetaCyc	117 (0.6%)	1144 (5.6%)

Table 2.10. Compounds with the same coloring identifiers, which includes R groups.

Databases	Tight coloring identifier	Loose coloring identifier
KEGG	209 (1.1%)	1132 (6.1%)
MetaCyc	449 (2.4%)	1638 (8.1%)

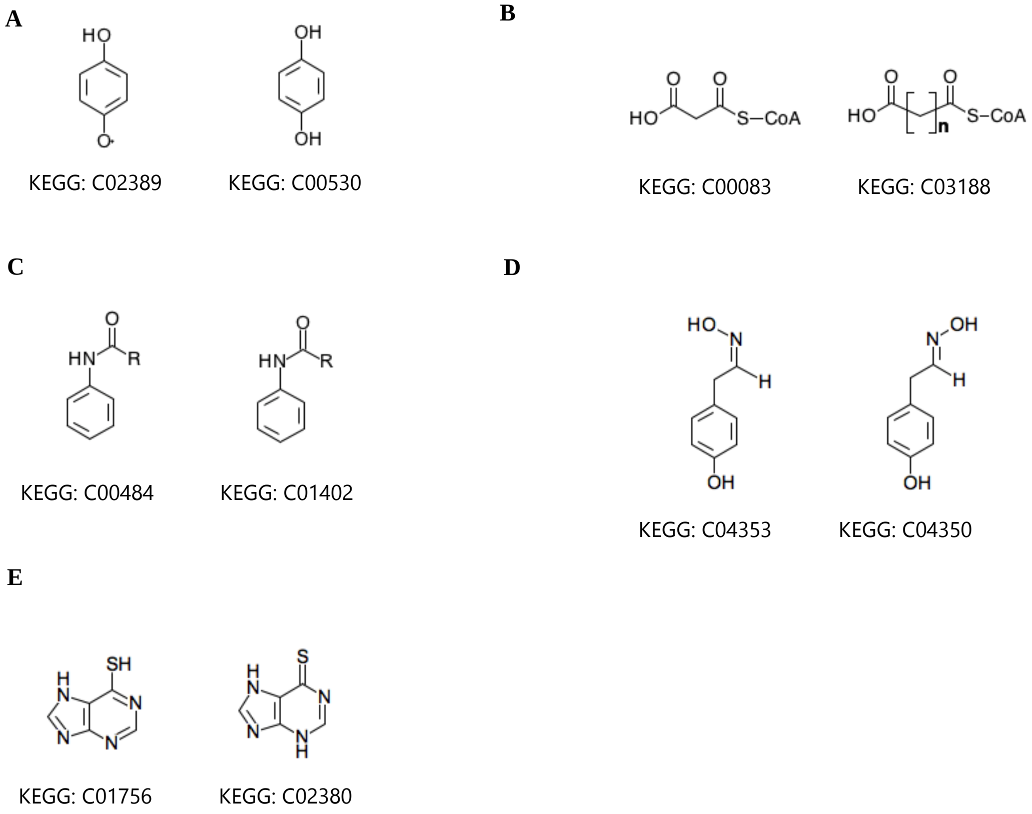


Figure 2.13. Representative compounds that cannot be distinguished by coloring identifier. A) Compound and its radical form; B) Compound containing repeated substructure; C) Compound with R representing a generic group; D) Isomers containing C=N; E) Compounds after curation of aromatic substructures.

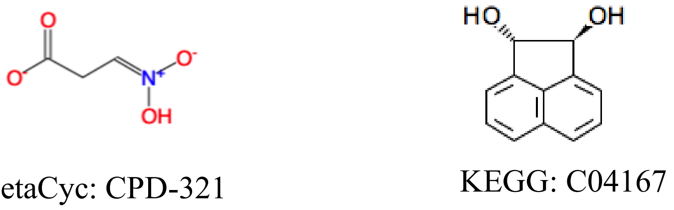
When the compound identifier is ambiguous, a compound in one database can be mapped to several different compounds in the other database during compound harmonization. For ID confirmed pairs, 28 MetaCyc compounds can be linked to more than one KEGG compound, which is caused by inconsistency of different ID references. Also, about 478 MetaCyc compounds have several KEGG correspondences among the 1848 pairs verified by 3-leveled EC. This highlights the value in leveraging metabolic reactions and the corresponding atom mappings to disambiguate multiple possible mappings while constructing an integrated metabolic network.

2.3.6.2 Pseudosymmetric atoms.

Omitting information in the atom coloring can also lead to pseudosymmetric atoms. We tested if incorporation of atom charge, atom stereochemistry, or bond stereochemistry in the atom coloring will erase some symmetric atoms (Table 2.11). After addition of atom charge, 148 MetaCyc and 38 KEGG compounds lose symmetry. Most of them are caused by terminal atoms, like CPD-321 (Figure 2.14). Since either symmetric atom can be labeled with charge, asymmetry caused by atom charge can be ignored in constructing metabolic network. In addition, both databases contain compounds affected by bond and atom stereochemistry. We need to take bond and atom stereochemistry into consideration, since some enzymes are stereochemically specific. A heuristic method could be used to test if symmetric atoms are affected by bond and atom stereochemistry, and then atom coloring identifiers incorporated with bond and atom stereochemistry will be generated to overcome this issue. However, more complex molecular symmetries like that illustrated by KEGG C04167 will require the use of algorithms that can detect all possible molecular symmetries (i.e. automorphisms induced by rotations and reflections of the $\Re^3$ embedded graph) using a 3-dimensional representation of the compound[100].

Table 2.11. Compounds gaining asymmetry after addition of extra information in the atom naming

Databases	**Atom stereochemistry**	**Atom charge**	**Bond stereochemistry**
KEGG	232	38	169
MetaCyc	219	148	227



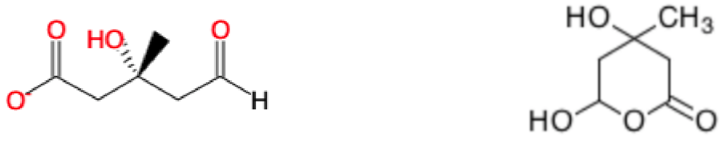
MetaCyc: CPD-321

KEGG: C04167

Figure 2.14. Example compounds gaining asymmetry after the addition of tight atom coloring information.
For CPD-321, the two oxygens bound to the nitrogen are asymmetric with tight atom coloring and symmetric with loose atom coloring.

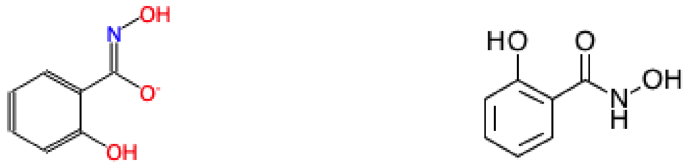
2.3.6.3 Changeable graph representation.

There are two types of matched compounds that cannot be detected by coloring identifiers. One group of compounds can have either linear or circular representations (see Figure 2.15), and there are about 26 examples in this category. The other group is caused by resonance structures (see Figure 2.16), and we discovered about 46 similar cases. Artificial sets of atom mappings can be created to represent chemical transformations that are spontaneous.



MetaCyc: MEVALDATE KEGG: C00772

Figure 2.15. Compound with linear and circular representations.



MetaCyc: CPD-6543 KEGG: C11343

Figure 2.16. Compound with different resonance structures.

2.4 Discussion

Here, we have developed a graph coloring method that creates unique identifiers for each atom in a compound with consideration for molecular symmetry. The atom-specific identifiers can capture additional cross-reaction atom mappings caused by symmetric atoms, which will contribute to the construction of a more complete atom-resolved metabolic network requiring information at both the compound and atom levels. Towards this overall goal, the ordered compound coloring identifiers derived from the corresponding atom coloring identifiers facilitate compound harmonization across metabolic databases, which is an essential first step in cross-database network integration. Different databases can have distinct preference in compound representations, especially for aromatic substructures. To overcome inconsistent aromatic representations between databases, we devised a pragmatic BASS method[89] for aromatic substructure detection that leverages the labeled aromatic substructures in KEGG. Application of BASS to KEGG validated the method, providing confidence in its application to the MetaCyc database. The automatic aromatic atom detection method in Indigo[87] further validated the comprehensiveness of our BASS aromatic substructure detection method that leverages KEGG's curated aromatic substructures, and the combination of BASS and Indigo can achieve good performance in aromatic substructure detection. This was further augmented by detecting additional aromatic substructure representations in MetaCyc through ID-based compound harmonization. In addition, compound states such as atom charge are not always the same between KEGG and MetaCyc. Therefore, identifiers like InChI that include these details to achieve an unambiguous label are not a good choice for maximizing cross-database compound harmonization in this situation. Furthermore, InChI cannot handle the compound entries that contain R-groups. However, InChI is very useful for validation of the presented methods development. Simplified molecular-input line-entry system (SMILES) identifiers and its derivatives are not a good option, because SMILES and its derivatives are not guaranteed to generate a unique identifier. Also, neither InChI nor SMILES deal with the unique naming of atoms that is consistent for symmetric atoms. While the molecular graph coloring method has similarities to molecular canonicalization methods[93, 101, 102], it was designed to facilitate harmonization of compounds between metabolic databases. The graph coloring method is flexible in adjusting information used in atom coloring, which can help detect more possible matched compounds with a higher false positive rate. With the coloring identifiers, we were able to detect 8865 correspondences between KEGG and MetaCyc compounds, and 5451 of them can be confirmed by other identifiers. In addition, commonality in EC numbers associated with reactions and compounds provided another avenue for both validating and predicting possible correspondence pairs. This method validated 1848 pairs unconfirmed by other identifiers. While harmonizing compounds between KEGG and MetaCyc, we detected various issues and errors in the databases by coloring identifiers which are enumerated in the supplemental material, suggesting that this method can also be used for curation of current metabolic databases. Furthermore, the graph coloring method and compound harmonization approach can be used to integrate any metabolic database that provides a molfile representation of compounds, greatly facilitating future construction of more complete integrated metabolic networks.

CHAPTER 3. HIERARCHICAL HARMONIZATION OF METABOLIC REACTIONS ACROSS METABOLIC DATABASES

3.1 Introduction

Metabolic models describe the inter-conversion of metabolites via biochemical reactions catalyzed by enzymes, providing snapshots of the metabolism under a given genetic or environmental condition[103, 104] . Metabolic models of metabolism have proven to be an important tool in studying systems biology and have been successfully applied to various research fields, ranging from metabolic engineering to system medicine [105-109]. Advances in analytical methodologies like mass spectroscopy and nuclear magnetic resonance greatly improve the high-throughput detection of thousands of metabolites, enabling the generation of large volumes of high-quality metabolomics datasets [48, 110] that greatly facilitate metabolic research. As a next major step, incorporating reaction atom-mappings into metabolic models enables metabolic flux analysis of isotope-labeled metabolomics datasets[53, 65, 69, 70], which will contribute to the large-scale characterization of metabolic flux molecular phenotypes and prediction of potential targets for gene manipulation[106]. Building reliable metabolic models heavily depend on the completeness of metabolic network databases. However, a relatively complete metabolic network, especially at an atom-resolved level, is practically not available [72].

Therefore, to construct an atom-resolved metabolic network, the very first major step is to integrate metabolic data from various metabolic databases without redundancy[82], which remains extreme labor-intensive. This is partially due to problems in the individual databases[111]. Common issues include non-unique compound identifiers, reactions with unbalanced atomic species, and enzyme catalyzing more than one reaction[112]. Moreover, incompatibilities of data representations (like compound identifiers) and incomplete atomistic details (like the presence of R groups and lack of atom and bond stereochemistry) across databases are key bottlenecks for the rapid construction of high-quality metabolic networks[113]. Great efforts have been made to map different compound identifiers across metabolic databases[114, 115]. Some algorithms use logistic regression to compute the similarity between strings generated by concatenating a variety of compound features, which requires careful selection of compound features that can well characterize a string pair by capturing the similarity between different variations as well as underlining the difference between descriptions which are not synonymous[108]. Alternatively, utilization of unique chemical identifier independent from a particular database, like InChI[93, 116] or SMILES[117], have been suggested as an important step in harmonizing metabolic databases[118]. However, several tricky cases still remain unresolved. For example, InChI cannot handle the compound entries that contain R-groups.

Our neighborhood-specific graph coloring method can derive atom identifiers for every atom in a specific compound with consideration of molecular symmetry, facilitating the construction of an atom-resolved metabolic network[119]. Furthermore, a unique compound coloring identifier can be generated based on the atom identifiers, which can be used for compound harmonization across metabolic databases. The results derived from the compound coloring identifiers were quite promising. However, issues like incomplete atomistic details were not completely handled in that prior work.

To put this paper into context with our prior published work, we first developed the subgraph isomorphism detection algorithm CASS (Chemically Aware Substructure Search) in 2014[89] and have made multiple improvements to this code base over the years and now call it BASS (Biochemically Aware Substructure Search). In developing our neighborhood-specific graph coloring method, we further enhanced BASS to efficiently detect aromatic substructures which was required for that work. In this chapter, we further optimized the subgraph isomorphism detection algorithm CASS (Chemically Aware Substructure Search)[89] to aid in the validation of generic compound pairs. In addition, we solved inconsistent atomistic characteristics across databases by defining a set of harmonization relationship types between compounds, aiming to capture chemical details while maintain compound pairs at various levels. Furthermore, we used the classification of compound pairs and EC (Enzyme Commission) numbers to harmonize metabolic reactions across Kyoto Encyclopedia of Genes and Genomes (KEGG) and MetaCyc metabolic pathway databases via establishing hierarchical harmonization relationships between metabolic reactions. We further made use of the atom identifiers to evaluate atom mapping consistency of these harmonized reactions. Through this analysis, we detected some issues that cause the inconsistency of reaction atom mappings both within and across databases. The generalization of metabolic reactions can be applied to various interesting topics including but not limited to predicting biotransformation of newly discovered metabolites[120], devising novel synthetic pathways of essential metabolites[121], and bridging gaps in the current metabolic network[122]. Furthermore, expanding the existing metabolic network by integrating other metabolic databases can be easily achieved when the molfile representations[76] of compounds are provided.

3.2 Materials and Methods

3.2.1 Compound and Metabolic Reaction Data

All data were downloaded directly from KEGG and MetaCyc databases. MetaCyc compound and reaction data downloaded from BioCyc is in version 23.0. The KEGG COMPOUND, KEGG REACTION and KEGG RCLASS data is from the version available from KEGG on April 2021 via its REST interface. KEGG RPAIR data was downloaded from KEGG database in 2016.

3.2.2 Curation of molfile

The documentation of atom stereochemistry in the molfiles is not complete. We used Open Babel[91] to curate the original molfiles and add stereospecific information.

3.2.3 Identification of double bond stereochemistry

We previously adopted a method for automated identification of double bond stereochemistry[90]. One limitation of this method is that only double bonds between two carbon atoms can be handled. For example, double bonds connected by heterogenous atoms, like N=C, cannot be processed by the method. Here, we designed a new algorithm to distinguish cis/trans stereoisomers. The same criteria are applied to assign priority to each group attached to the double bond. If one side of the double bond only has one group, this group will be prioritized. Next, the 2D plane of the compound representation is divided into two parts with line crossing the double bond. If the prioritized groups of both sides are on the same part of the divided plane, the double bond is labeled as cis; otherwise, it is trans.

3.2.4 Flowchart of steps in the compound and reaction harmonization process

The flowchart of steps in compound and reaction harmonization is shown in Figure 3.1. The initial compound pair list is composed of compound pairs detected by the loose compound coloring identifiers. Next, reaction harmonization is conducted with the compound pair list. Two criteria are obeyed in reaction harmonization: the two reactions should share at least one EC number and all compounds in the two reactions are paired unless one reaction has an extra compound entity, like H+. Apart from valid reaction pairs, reaction pairs with the same EC number and some unmatched compounds are also extracted. We hypothesized that those unmatched compounds are likely to be compound pairs. Validation is conducted for the unmatched compounds, and the valid compound pairs are added to the compound pair list. Every time the compound pair list is updated, the above process is repeated until no new compound pairs are discovered.

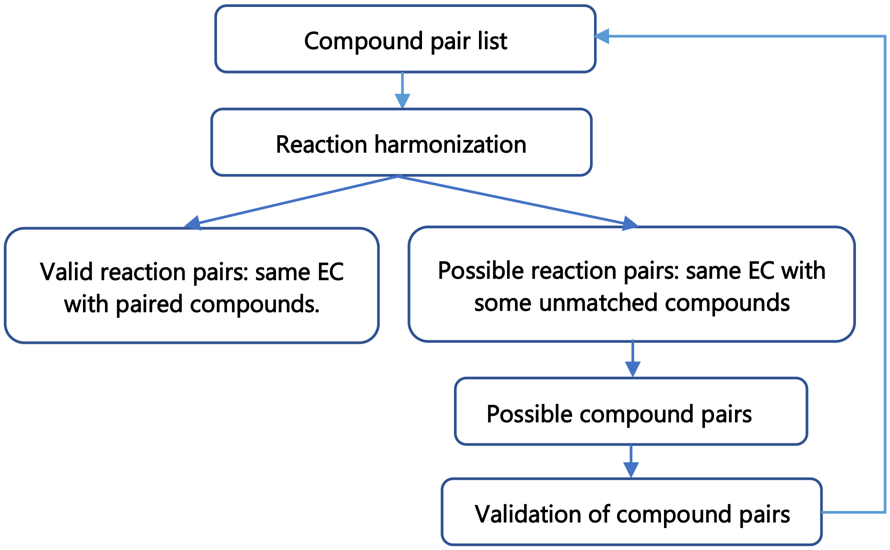

Compound pair list
Reaction harmonization
Valid reaction pairs: same EC with paired compounds.
Possible reaction pairs: same EC with some unmatched compounds
Possible compound pairs
Validation of compound pairs

Figure 3.1. Flowchart of compound and reaction harmonization.

3.2.5 Validation of tautomers

Most common form of tautomerization involves a hydrogen changing places with a double bond. Based on this transformation, the following steps are performed to validate if two compounds with same chemical formula are tautomers. To eliminate the difference caused by single and double bonds in the structural representation, all the double bonds are converted into single bonds, and the subgraph isomorphism detection algorithm[89] is used to check if two structural representations are the same after modification. Next, double bonds at unmatched positions are examined. If all the mismatches are caused by possible tautomerization, the compound pair is considered valid. Finally, other chemical details not related to atoms in the changeable positions are compared to classify the relationship between valid pairs.

3.2.6 Validation of generic compound pairs of compounds with different chemical formula.

For two compounds A and B, the subgraph isomorphism detection algorithm[89] is used to verify if the graph representation of A (ignoring R and H) is contained in the graph representation of B. Then, each unmatched branch in B is examined if it corresponds to an R group in A. Compound pairs that meet both criteria are considered valid. Next, the chemical details (atom and bond stereochemistry) in the two compounds are compared for relationship type classification. If the chemical details of compound A are included in compound B, then A has a generic-specific relationship to B; otherwise, A and B have a loose relationship.

3.2.7 Validation of compound pairs with linear and circular representations.

The compound with changeable linear and circular structures are common in small molecule carbohydrate metabolites, like glucose. This conversion occurs due to the ability of aldehydes and ketones to react with alcohols. To validate the compound pairs with linear and circular representations, we first locate the bond in the circular structure that is formed by connecting the C in the aldehyde (keto) group and O in the hydroxy group. The following steps include breaking the newly formed bond and restoring the C=O bond in the aldehyde (keto) group. Then, a new compound coloring identifier is generated for the modified circular representation. If the updated compound coloring identifiers match, the compound pair is considered valid.

3.2.8 Parse of KEGG RCLASS RDM patterns.

Based on the RDM patterns, we first identified the possible atoms that can be mapped to each reaction center. Then we derived the possible combinations of atoms for all the reaction centers for each compound. We paired cases in either compound, removed changed bond in the compound according to different region, and detected the maximum common subgraph of the remaining structures. We examined all the combinations and derived the optimal mappings with the maximum number of mapped atoms and least ratio of changed atoms.

3.3 Results

3.3.1 Overview of KEGG and MetaCyc databases

The compounds in the KEGG and MetaCyc databases are summarized in Table 3.1. Based on the atomic composition, we divided compounds into two groups: *specific compounds* (no R group) and *generic compounds* (with presence of R group(s)). About 8.02% KEGG compounds and 21.72% MetaCyc compounds contain R groups.

Table 3.1. Summary of KEGG and MetaCyc compound databases.

Compound Type	**KEGG**	**MetaCyc**
specific compounds	16529 (91.98%)	15859 (78.28%)
generic compounds	1441 (8.02%)	4400 (21.72%)
Total	17970 (100%)	20259 (100%)

According to the classification of compounds, we also categorized the atom-resolved metabolic reactions into two sets: specific reactions where all compounds in the reaction are specific compounds and generic reactions which contain at least one generic compound. Here, we only considered reactions with relatively complete EC numbers[97, 123] since consistent EC number is one essential component in reaction harmonization. From Table 3.2, we can see that about 15% KEGG reactions and 34% MetaCyc reactions are generic reactions.

Table 3.2. Summary of KEGG and MetaCyc atom-resolved metabolic reaction databases

Reaction Type	**KEGG**	**MetaCyc**
specific reactions (4-leveled EC)	6780 (75.26%)	6397 (49.93%)
specific reactions (3-leveled EC)	886 (9.83%)	2022 (15.78%)
generic reactions (4-leveled EC)	1244 (13.81%)	3572 (27.88%)
generic reactions (3-leveled EC)	99 (1.10%)	822 (6.42%)
Total	9009 (100%)	12813 (100%)

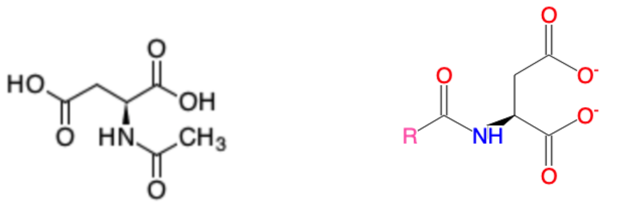
We further did a simple quality check of the atom-resolved reactions in KEGG and MetaCyc databases (Table 3.3). KEGG contains about 7.5% incomplete reactions where the number of atoms on both sides of the reaction is different. For MetaCyc, less than 0.5% reactions have incorrect atom mappings caused by mapping different atoms of different elements. In addition, a large amount of reactions only have part of atoms mapped in both KEGEG and MetaCyc databases. This level of incompleteness prevents their effective use in mass balanced metabolic modeling.

Table 3.3. Quality check of atom-resolved reactions in KEGG and MetaCyc.

Database	**Incomplete Reaction**	**Incorrect Atom Mappings**	**Incomplete Atom mappings**
KEGG	772 (7.53%)	0	7213 (70.36%)
MetaCyc	0	54 (0.37%)	6130 (41.87%)

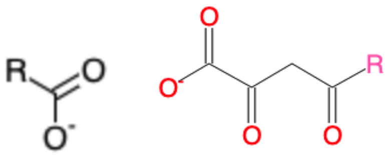
3.3.2 Results of compound harmonization across KEGG and MetaCyc databases

With the loose compound coloring identifiers generated by the neighborhood-specific graph coloring method, about 8865 compound pairs were detected, including both generic and specific compound pairs [119]. However, some cases were not solved perfectly by the loose compound coloring identifiers. First, chemical details like atom and bond stereochemistry were ignored in the loose compound coloring identifies. Second, a compound pair can involve a generic compound and a specific compound (Figure 3.2), which cannot be discovered by the loosing compound coloring identifiers. The methyl group in KEGG compound C01042 can be a specification of the R group in MetaCyc compound CPD-576. What makes things more complicated is that a compound pair can be composed of two generic compounds with different atom composition. In Figure 3.3, even though both compounds contain an R group, the MetaCyc compound 3-Acyl-pyruvates can be regarded as a subgroup of compounds belonging to KEGG compound C00060. In addition, compound pairs with different structural representations, like tautomers, were missed by the loose compound coloring identifiers.



KEGG: C01042 MetaCyc: CPD-576

Figure 3.2. Compound pair of generic and specific compounds.



KEGG: C00060 MetaCyc: 3-Acyl-pyruvates

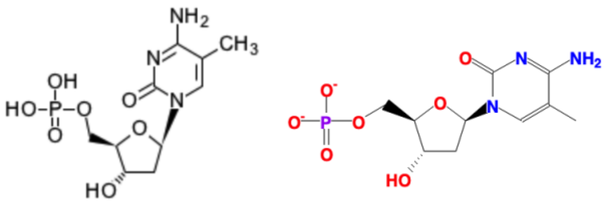
Figure 3.3. Compound pair of generic compounds.

3.3.2.1 Harmonization of specific compounds.

We first incorporated the chemical details, including atom stereochemistry and bond stereochemistry to evaluate the specific compound pairs detected by loose compound coloring identifiers. Incorporation of the chemical details can lead to three scenarios: 1) the paired compounds have the same set of chemical details; 2) the chemical details of one compound are the subset of the other compound; 3) the chemical details of the two compounds cannot be fully matched. Based on the above cases, we decided to classify the relationship between compound pairs as an equivalence relationship, a generic-specific relationship, or a loose relationship. With this classification, a compound in one database can be paired with multiple compounds in the other database with an appropriate relationship. With these improvements incorporated into specific compound harmonization (Table 3.4), we can see that the majority of specific compound pairs have a loose relationship, which is not surprising since the criteria for the loose relationship were less strict. Another explanation is that the chemical details for the same compound can be inconsistent across databases. The MetaCyc compound CPD-399 has a direct KEGG compound reference C03495 (Figure 3.4), but stereochemistry of some atoms in the two compound representations are not the same.

Table 3.4.Harmonization of specific compound pairs.

Relationship Type	**Count**
equivalence	3636 (27.99%)
generic-specific	1712 (13.18%)
loose	7642 (58.83%)
Total	12990 (100%)



KEGG: C03495 MetaCyc: CPD-399

Figure 3.4. Example harmonized compound pair of compounds with inconsistent chemical details.

3.3.2.2 Harmonization of generic compounds.

Generic compounds further complicate relationships between compounds. A generic compound can be related to generic and/or specific compounds (Figure 3.2 & 3.3). For a compound pair of two *generic compounds* with the same atom composition, we classify them based on the same criteria of *specific compounds*. Harmonization of *generic compound* pairs of compounds with different chemical formulas is much more complicated, involving detection and validation steps. All chemical identifiers fail in detecting the possible pairs, including the loose compound coloring identifiers. On the other hand, it will be very time-consuming and unnecessary to do brute-force search of all compounds across databases.

Here, we made use of the metabolic reactions across databases to detect the possible compound pairs with a different atom composition. We first extracted reaction pairs that can contain at least one *generic reaction* and share at least one EC number. Next, compounds with R group(s) in one reaction were paired with all the compounds in the other reaction. The validation method is described in the Materials and Methods section. Results of harmonization are summarized in Table 3.5. Most of the *generic compound* pairs have generic-specific relationships. This may be explained by the assumption that chemical details in a compound with less atoms are more likely to be included in the compound containing more atoms.

Table 3.5. Harmonization of generic compounds.

Relationship Type	**Count**
equivalence	126 (4.72%)
generic-specific	2543 (95.28%)
loose	0
Total	2669 (100%)

3.3.2.3 Harmonization of compounds with changeable representations.

Harmonization of compounds with changeable representation (e.g. linear vs circular sugar representations) also requires detection and validation. Again, metabolic reactions were used to detect the possible compound pairs via an iterative approach. Two criteria should be obeyed when extracting the reaction pairs: 1) the two reactions should share at least one EC number; 2) at least a pair of compounds in the two reactions can be matched. For those unmatched compounds with the same chemical formula, they will be added to the possible list. The validation methods are described in the Materials and Methods section. About 45 such compound pairs were discovered after two rounds of iteration (Table 3.6).

Table 3.6. Harmonization of compounds with changeable representations.

Relationship Type	**Count**
equivalence	20 (44.44%)
generic-specific	0
loose	25 (55.56%)
Total	45 (100%)

3.3.2.4 Summary of compound harmonization.

All compound pairs detected above were summarized in Table 3.7. In total, 15,704 compound pairs were discovered, and more than 80% of them were specific compound pairs, roughly in agreement with the proportion of generic compounds in the database. More importantly, about 2,669 generic compound pairs were detected, which cannot be achieved by any existing chemical identifier.

Table 3.7. Summary of compound harmonization between KEGG and MetaCyc.

Compound Pair Type	**Count**
specific compound pairs	12990 (82.72%)
generic compound pairs	2669 (16.99%)
changeable compound pairs	45 (0.29%)
Total	15704 (100%)

3.3.3 Results of reaction harmonization across KEGG and MetaCyc databases.

With the harmonized compounds, we performed reaction harmonization across KEGG and MetaCyc databases. Two criteria should be followed in reaction harmonization: 1) the two reactions should share at least an EC number; and 2) all compounds in the two reactions should be paired unless one reaction has an extra compound entity, like H+. Reaction pairs were further categorized into the following three relationship types based on the classification of their compound pairs: 1) equivalence relationship when a reaction pair included only equivalently paired compounds; 2) generic-specific relationship when a reaction pair only included equivalently paired compounds and at least one generic-specific compound pair that are consistently in the same general-to-specific direction; 3) loose relationship when a reaction pair included loosely paired compounds or generic-specific paired compounds with inconsistent general-to-specific direction.

We first harmonized the specific metabolic reactions where both reactions are specific reaction (Table 3.8). We can see that reaction pairs in group 3 take up more than 70%, which is quite consistent with the classification of specific compound pairs. About 60% of specific compound pairs are loosely matched (Table 3.4), and a reaction pair only requires one loosely matched compound pair to be classified into group 3.

Table 3.8. Harmonization of specific reactions between KEGG and MetaCyc.

Relationship Type	**Count**
equivalence	718 (24.00%)
generic-specific	68 (2.27%)
loose	2205 (73.72%)
Total	2991 (100%)

We also analyzed the generic reaction pairs where at least one reaction is generic. Above 70% generic reaction pairs are in group 2 (Table 3.9), which can also be well explained by the previous result that around 95% generic compound pairs have a generic-specific relationship.

Table 3.9. Harmonization of generic reactions between KEGG and MetaCyc.

Relationship Type	**Count**
equivalence	29 (6.03%)
generic-specific	344 (71.51%)
loose	108 (22.45%)
Total	481 (100%)

Since the EC information is not very complete in both databases, some reaction pairs can be ignored due to the mismatch or miss of the last level EC. To avoid missed pairs, we relaxed the first criterion in reaction harmonization to “the two reactions should have at lease a pair of EC numbers that share the first 3 levels”. The newly discovered reaction pairs are summarized in Table 3.10, including both specific and generic reaction pairs. Either mismatch or miss of last EC occur in some reaction pairs. Specific examples are shown in Figure 3.5 & 3.6.

Table 3.10. Loose harmonization of reactions between KEGG and MetaCyc.

Relationship Type	**Count**
equivalence	49 (12.76%)
generic-specific	96 (25.00%)
loose	239 (62.24%)
Total	384 (100%)

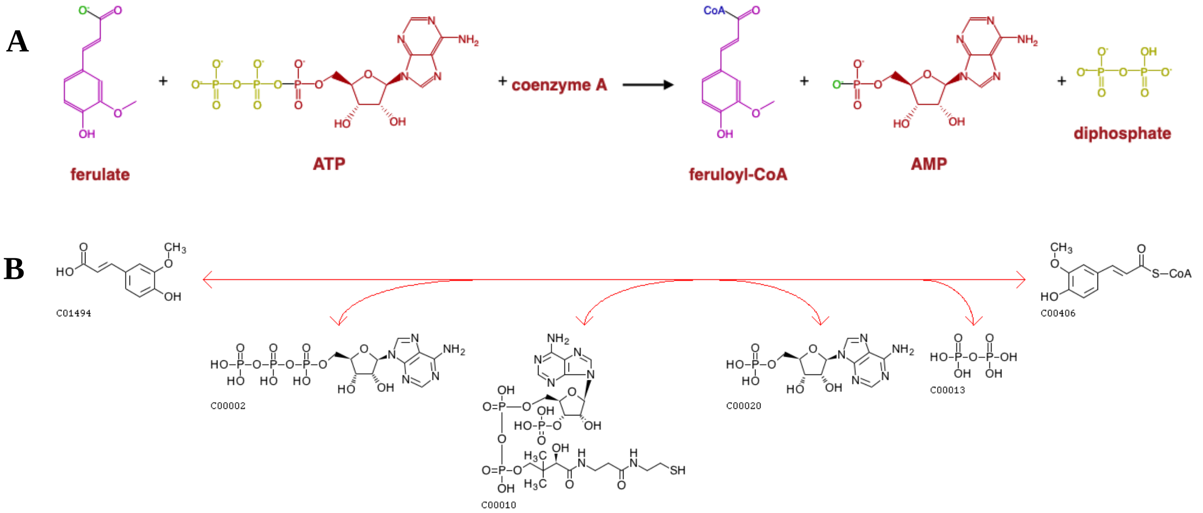


Figure 3.5. Reaction pair with mismatch of last EC number.
A) MetaCyc reaction 6.2.1.34-RXN with EC number 6.2.1.34 (https://metacyc.org/META/NEW-IMAGE?object=6.2.1.34-RXN&&redirect=T); B) KEGG reaction R02194 with EC number 6.2.1.12 (https://www.genome.jp/entry/R02194).

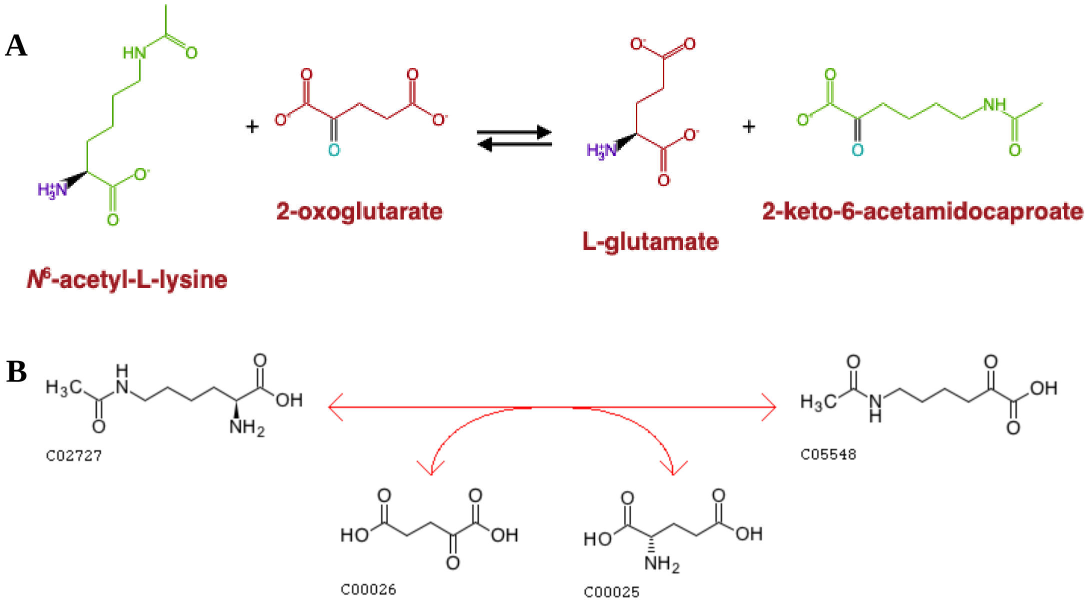


Figure 3.6. Reaction pair with missing 4th-level EC number designation.
A) MetaCyc reaction ACETCAPR-RXN with EC number 2.6.1.- (https://metacyc.org/META/NEW-IMAGE?object=ACETCAPR-RXN&&redirect=T);
B) KEGG reaction R04029 with EC number 2.6.1.65 (https://www.genome.jp/entry/R04029).

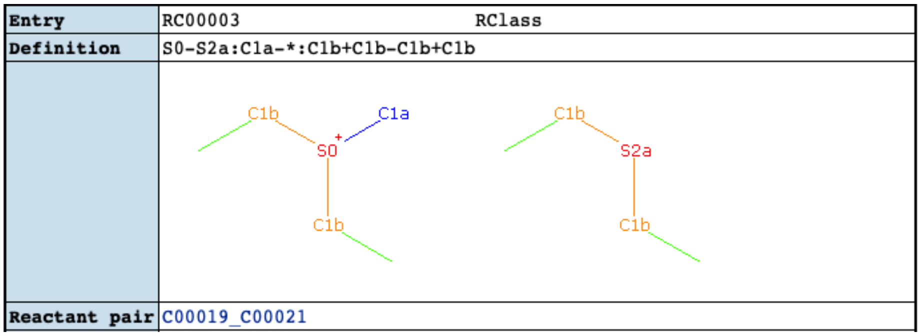
The results of reaction harmonization are shown in Table 3.11. Overall, 3,856 reaction pairs were detected via EC numbers and integrated compound pairs. The majority of reaction pairs are specific. About 10% of reactions pairs can be missed due to incomplete and inconsistent EC numbers.

Table 3.11. Summary of reaction harmonization between KEGG and MetaCyc.

Relationship Type	**Count**
specific	2991 (77.57%)
generic	481 (12.47%)
loose EC	384 (9.96%)
Total	3856 (100%)

3.3.4 Comparison of KEGG RCLASS and RPAIR data

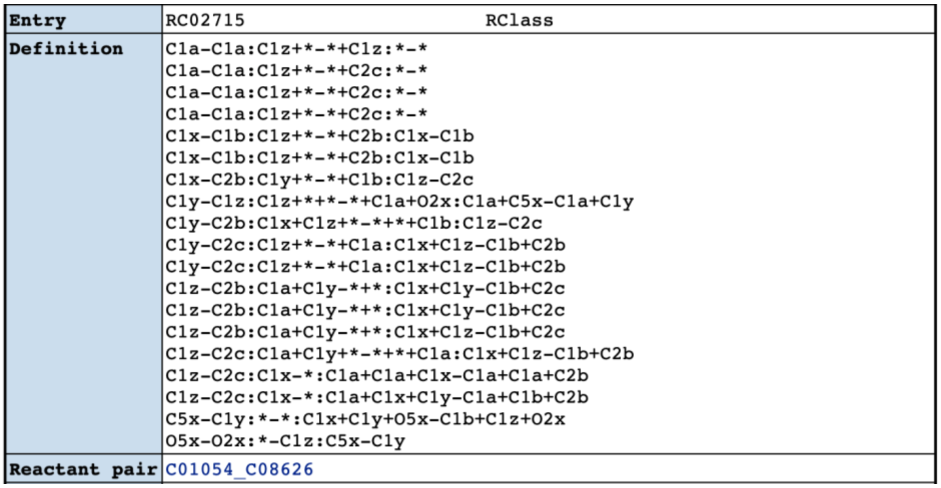
For the KEGG database, the RCLASS data describes the chemical transformation of substrate-product in the RDM pattern [124]. A RDM description can be divided into three parts: reaction center (R), the different region (D), and the matched region (M). In order to distinguish functional groups and microenvironment of atoms, KEGG classified atomic species of C, N, O, S, and P into 68 types (KEGG atom types)[88], which are implemented in the RDM description. As shown in Figure 3.7, The RCLASS entry RC00003 contains one RDM description. The S atom is the reaction center, the C1a in the first substructure belongs to the different region, and those C1b atoms are in the matched region. Based on the RDM pattern, we derived the atom mappings for specific reactant-product compound pairs based on a common graph isomorphism search between the two compounds limited by RDM description. We successfully parsed atom mappings for 10,212 (out of 10,313) compound pairs. There are 76 compound pairs that cannot be deciphered due to the incorrect or missing descriptions of reaction centers. For complicated compound pairs with multiple reaction centers, each reaction center can be mapped to several different atoms, which in a few instances causes a serious combinatorial issue that is impossible to address in a reasonable amount of time. An example is shown in Figure 3.8. Roughly 10^{13} possible cases can be derived based on the RDM descriptions. In total, 25 compound pairs cannot be processed owing to this combinatorial problem (Table 3.12). KEGG used to archive the atom mappings between the reactant-product compound pairs in the RPAIR database, where the mapped atoms are specified by the atom numbering for a compound pair. Here, we evaluated the atom mappings derived from RCLASS and an older version of KEGG RPAIR. The majority (great than 86%) of atom mappings between RCLASS and RPAIR are the same (Table 3.13). To further validate the results, we calculated the fraction of atom mappings with changed local bonded chemical environment across the mapping (i.e., atom mappings with changed one-bond atom color) in terms of the total number of mapped atoms in the reaction. The expectation is that this fraction represents the fraction of reaction center atoms present where a chemical bond is changed or broken. Then, we generated a scatter plot of changed local atom color fraction for KEGG RPAIR versus RCLASS atom mappings. From Figure 3.9 A, we can see that the majority compound pairs have the same fraction of changed local atom color (concentrated on the diagonal line). In addition, more atom mappings derived from RPAIR have a higher ratio of changed atoms. We figured out that the majority of the inconsistency is due to the interchangeable mappings of resonant atoms, like the O atoms in the carboxyl group (Figure 3.10). After further curation to handle resonant atoms (Table 3.14), about 94% compound pairs have the same atom mappings. From Figure 3.9 B, we can see that quite large portion of compound pairs with higher ratio of changed atoms in RPAIR disappear. The remaining 557 inconsistent mappings appear to come from two different issues. One, more than 93% of the remaining inconsistent mappings (517 out of 557) are likely caused by the updating of the KCF (molfile like) files or associated molfiles in KEGG database from continual curation. For example, the RDM description for compound pair C01255_C02378 has been updated in the RCLASS (Figure 3.11). We also plotted the changed one-bond atom color fraction for compound pairs with RDM update (Figure 3.9 C). Compound pairs in either RCLASS or RPAIR can have higher changed atom ratio. The fraction of changed local atom color appears to equally distributed above and below the diagonal red line, which is interesting since the update of RDM descriptions is a correction process in KEGG database and may reflect both changes in specific mapped atoms and changes in the overall proportion of atoms mapped. Two, we found that a compound representation can vary across different compound pairs. Therefore, we hypothesize that a lack of synchronization between compound and compound_pair representations over time has caused the observed atom mapping inconsistencies detected in most of the other 40 compound pairs. For this part, RPAIR compound pairs show an increased fraction of changed local atom color (Figure 3.9 D) versus its equivalent RCLASS, demonstrating that this metric has value in evaluating atom mappings.



Entry	RC00003 RClass
Definition	S0-S2a:C1a-*:C1b+C1b-C1b+C1b
Reactant pair	C00019_C00021

Figure 3.7. Example of RCLASS entry.
RCLASS RC00003 (https://www.genome.jp/dbget-bin/www_bget?rc:RC00003).

A



Entry	RC02715	RClass
Definition	C1a-C1a:C1z+*-*+C1z:*-* C1a-C1a:C1z+*-*+C2c:*-* C1a-C1a:C1z+*-*+C2c:*-* C1a-C1a:C1z+*-*+C2c:*-* C1x-C1b:C1z+*-*+C2b:C1x-C1b C1x-C1b:C1z+*-*+C2b:C1x-C1b C1x-C2b:C1y+*-*+C1b:C1z-C2c C1y-C1z:C1z+*+*-*+C1a+O2x:C1a+C5x-C1a+C1y C1y-C2b:C1x+C1z+*-*+*+C1b:C1z-C2c C1y-C2c:C1z+*-*+C1a:C1x+C1z-C1b+C2b C1y-C2c:C1z+*-*+C1a:C1x+C1z-C1b+C2b C1z-C2b:C1a+C1y-*+*:C1x+C1y-C1b+C2c C1z-C2b:C1a+C1y-*+*:C1x+C1y-C1b+C2c C1z-C2b:C1a+C1y-*+*:C1x+C1z-C1b+C2c C1z-C2c:C1a+C1y+*-*+*+C1a:C1x+C1z-C1b+C2b C1z-C2c:C1x-*:C1a+C1a+C1x-C1a+C1a+C2b C1z-C2c:C1x-*:C1a+C1x+C1y-C1a+C1b+C2b C5x-C1y:*-*:C1x+C1y+O5x-C1b+C1z+O2x O5x-O2x:*-C1z:C5x-C1y	
Reactant pair	C01054_C08626	

B

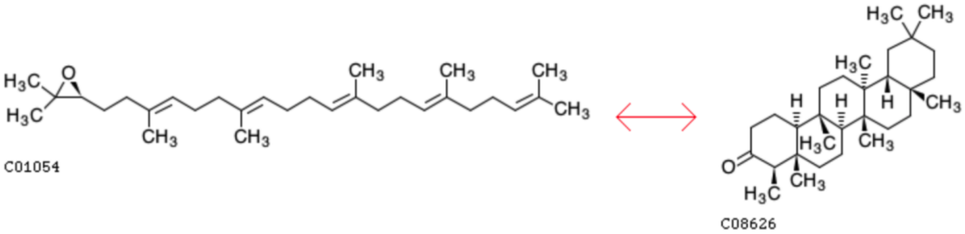


Figure 3.8. RCLASS with combinatorial issue caused by multiple possible mappings of RDM descriptions.

A) RDM description of RCLASS RC02715 (https://www.genome.jp/dbget-bin/www_bget?rc:RC02715); B) Compound pair C01054_C08626 that follows the RC02715 RDM pattern (https://www.genome.jp/Fig/reaction/R09910.gif).

Table 3.12. Hardly interpretable compound pairs.

RCLASS	Compound pair
RC02715	C01054_C08626
RC00871	C01051_C02463
RC01850	C00751_C06309
RC01579	C00751_C06083
RC01851	C00751_C06310
RC01582	C01054_C01902
RC02163	C00751_C08627
RC02496	C12354_C18337
RC01862	C01054_C08615
RC01616	C05773_C05774
RC02632	C01054_C08637
RC03124	C01054_C08797
RC02708	C01054_C17966
RC01863	C01054_C08616
RC02603	C01054_C19819
RC01864	C01054_C08628
RC02714	C01054_C20188
RC02619	C01054_C19833
RC02620	C01054_C19801
RC02621	C00751_C19834
RC01901	C02094_C15943
RC02716	C01054_C20189
RC02717	C01054_C20191
RC02720	C01054_C20194
RC02722	C01054_C20200

Table 3.13. First-round evaluation of atom mappings of compound pairs between KEGG RCLASS and RPAIR.

Condition	Count
same atom mappings	8017 (86.1%)
inconsistent atom mappings	1294 (13.9%)
Total	9311 (100%)

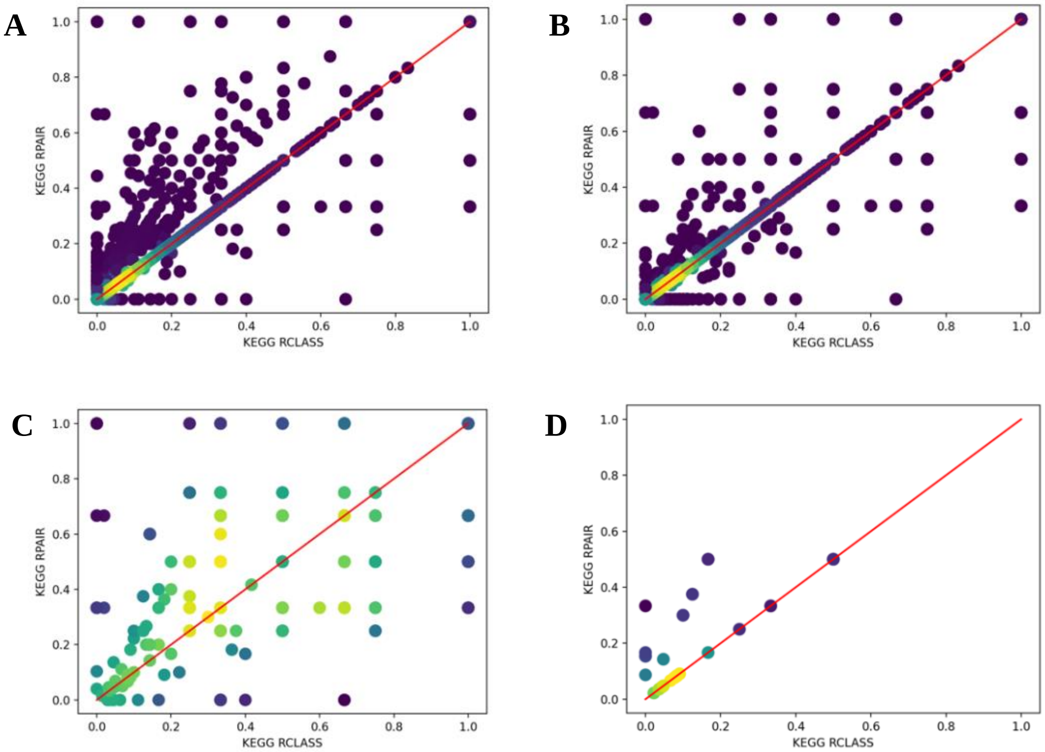


Figure 3.9. Scatter plot of changed one-bond atom color fraction for KEGG RCLASS versus RPAIR atom mappings in paired compounds.

(A) All compound pairs before correcting resonant atoms; (B) All compound pairs after correcting resonant atoms; (C) Compound pairs with inconsistent atom mappings caused by RDM update; (D) Compound pairs with inconsistent atom mappings caused by unsynchronized representations.

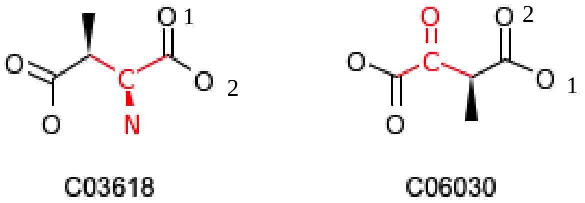
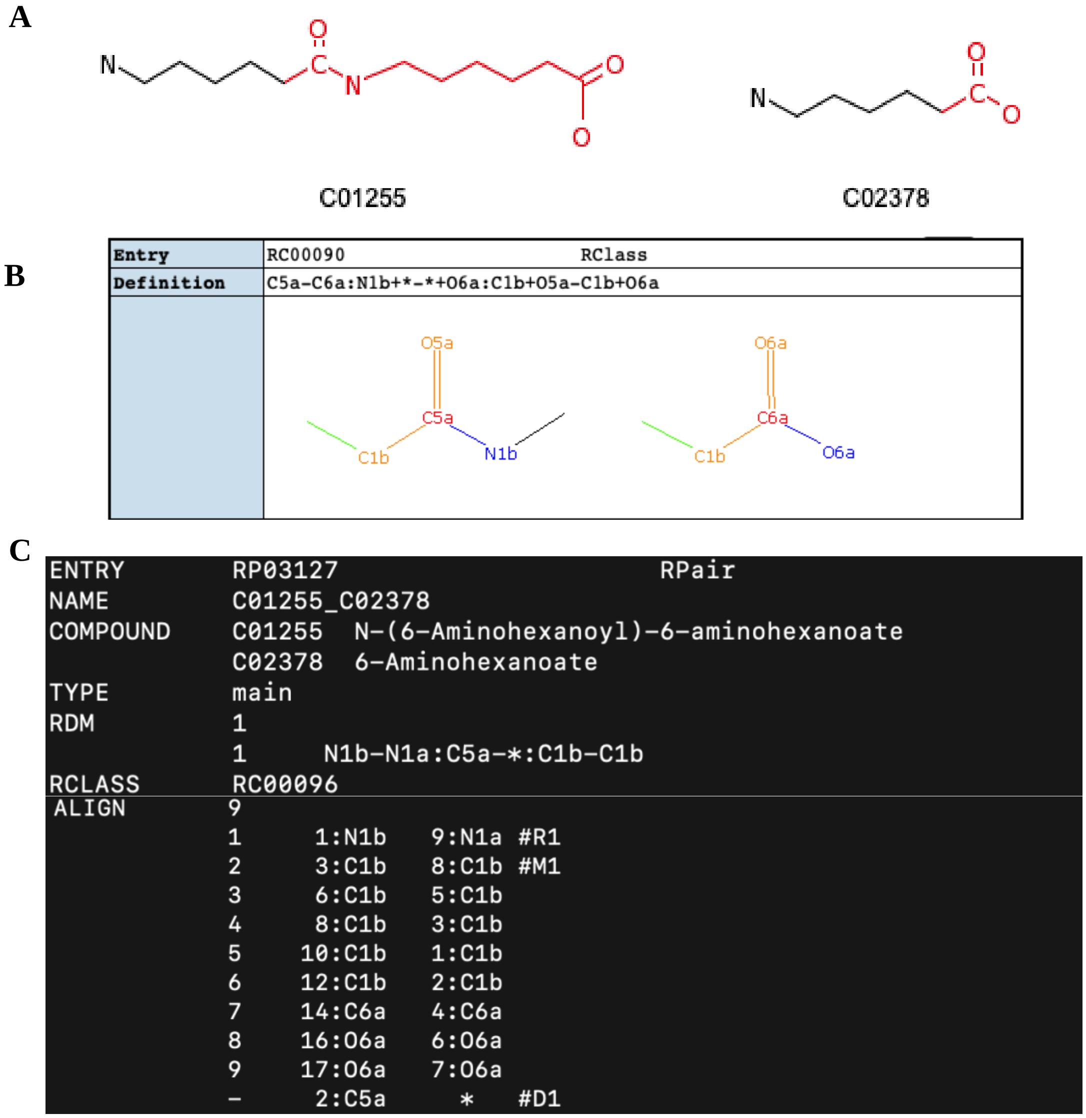


Figure 3.10. Example of atoms with interchangeable mappings.

KEGG RPAIR maps atom 1 in C03618 to atom 2 in C06030 and atom 2 in C03618 to atom 1 in C06030.

Table 3.14. Second-round evaluation of atom mappings of compound pairs between KEGG RCLASS and RPAIR.

Condition	Count
same atom mappings	8333 (95.19%)
inconsistent atom mappings	422 (4.8%)
Total	8755 (100%)



Entry	RC00090 RClass
Definition	C5a-C6a:N1b+*-*+O6a:C1b+O5a-C1b+O6a

```
ENTRY          RP03127                          RPair
NAME           C01255_C02378
COMPOUND       C01255   N-(6-Aminohexanoyl)-6-aminohexanoate
               C02378   6-Aminohexanoate
TYPE           main
RDM            1
               1      N1b-N1a:C5a-*:C1b-C1b
RCLASS         RC00096
 ALIGN         9
               1      1:N1b    9:N1a  #R1
               2      3:C1b    8:C1b  #M1
               3      6:C1b    5:C1b
               4      8:C1b    3:C1b
               5     10:C1b    1:C1b
               6     12:C1b    2:C1b
               7     14:C6a    4:C6a
               8     16:O6a    6:O6a
               9     17:O6a    7:O6a
               -      2:C5a      *    #D1
```

Figure 3.11. Comparison of KEGG RCLASS and RPAIR description for compound pair C01255 and C02378.

A) Compound C01255 and C02378 (https://www.kegg.jp/kegg-bin/rpair_image?entry=RC00090&cpair=C01255_C02378); B) KEGG RCLASS description for the compound pair (https://www.genome.jp/dbget-bin/www_bget?rc:RC00090); C) KEGG RPAIR description for the compound pair along with the atom mappings.

3.3.5 Evaluation of atom mappings between KEGG and MetaCyc databases.

The atom mappings for each reaction in the MetaCyc database are specified based on the atom numbering of each compound in their molfile representation. For the KEGG database, we used the atom mappings for compound pairs parsed from the RCLASS entries. Here, we evaluated the atom mappings in about 3000 specific reaction pairs with the same compound representations (Table 3.15). About 88% of the reaction pairs have consistent atom mappings between the two databases. A consistent example is shown in Figure 3.12.

Table 3.15. Evaluation of atom mappings between KEGG and MetaCyc.

Condition	Count
same atom mappings	2685 (88.0%)
inconsistent atom mappings	366 (12.0%)
Total	3051 (100%)

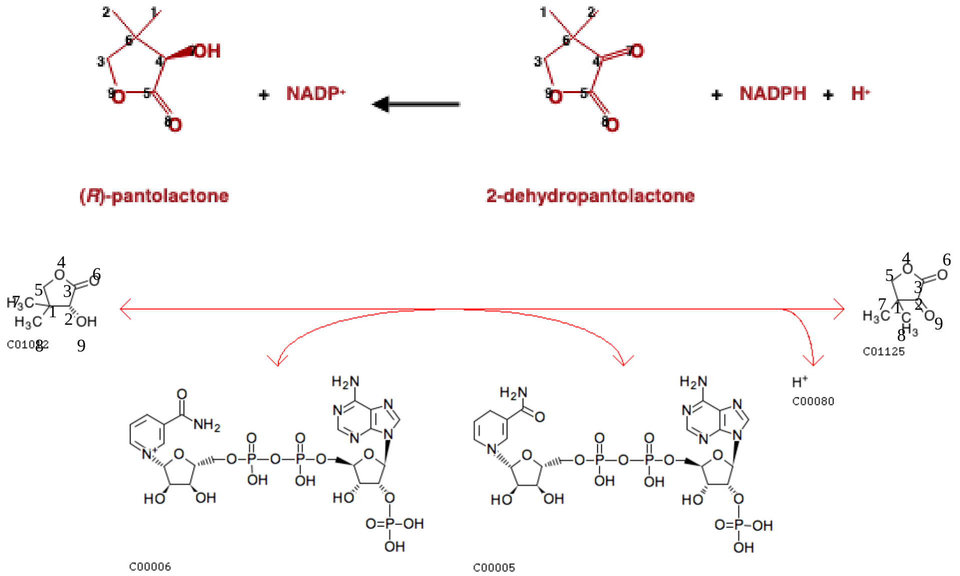


Figure 3.12. Example of reaction pair with the same atom mappings.
Reaction pair MetaCyc 1.1.1.168-RXN (https://metacyc.org/META/NEW-IMAGE?object=1.1.1.168-RXN&&redirect=T) and KEGG R03155 (https://www.genome.jp/entry/R03155) have the same atom mappings.

We also generated a scatter plot of changed atom color fraction between paired KEGG and MetaCyc reactions (Figure 3.13 A). For some reactions, only part of the compounds are mapped in either database (Table 3.3). For MetaCyc, atoms are normally mapped at a reaction level. Since the KEGG RCLASS database maps atoms at a compound level, multiple RCLASS atom mappings must be evaluated together for a given KEGG reaction. We also just visualized paired reactions with inconsistent atom mappings (Figure 3.13 B). We can see that the MetaCyc reactions have a higher ratio of changed local atom color. However, the number of mapped atoms in the paired reactions are not always the same, which can cause the fraction of changed local atom color can deviate from the diagonal. This issue makes a direct interpretation for specific reaction pairs more difficult, but the observed trend above the diagonal has interpretable value.

Through these comparisons, we see that both databases can contain distinct issues with their atom mappings. Some MetaCyc reactions can have incorrect atom mappings. An example is shown in Figure 3.14. For some KEGG reactions with single compound involving in several compound pairs, one atom can be mapped to multiple atoms and leave some atoms unmatched. For the KEGG reaction R10579 shown in Figure 3.15 A, based on the RDM descriptions in the two compound pairs (Figure 3.15 B & 3.15 C), atom 1 in compound C00251 is mapped to atom 1 in compound C00022 and atom 1 in compound C00578, leaving atom 2 in C00251 unmapped. Compared with the corresponding MetaCyc reaction RXN-14940 (Figure 3.16), the RDM description of KEGG RCLASS RC03212 appears incorrect. The harmonized reactions with different atom mappings are shown in Supplementary Spreadsheet 3.1.

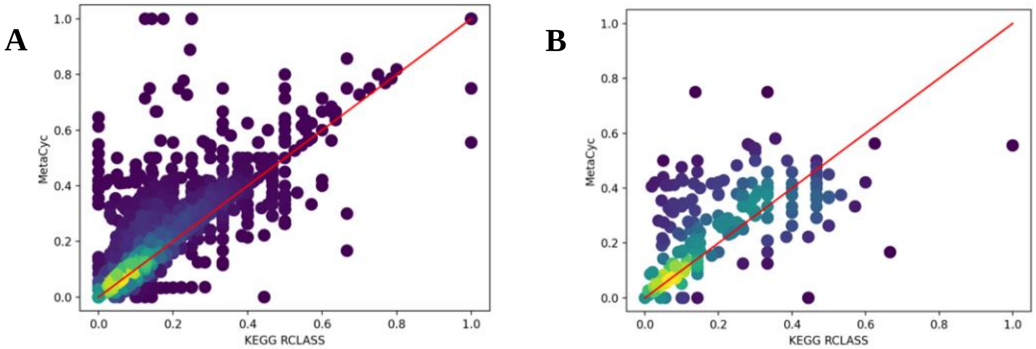


Figure 3.13. Scatter plot of changed one-bond atom color fraction for KEGG versus MetaCyc atom mappings in paired reactions.

Lighter colors represent higher overlapped point density. (A) All paired reactions are included; (B) Reaction pairs with inconsistent atom mappings are included.

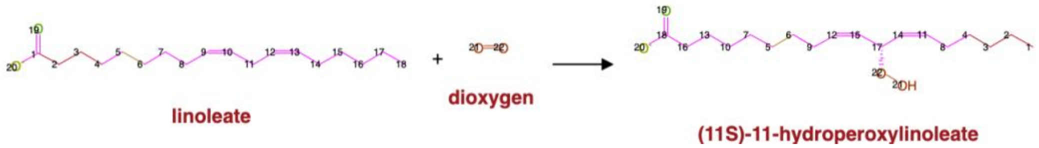
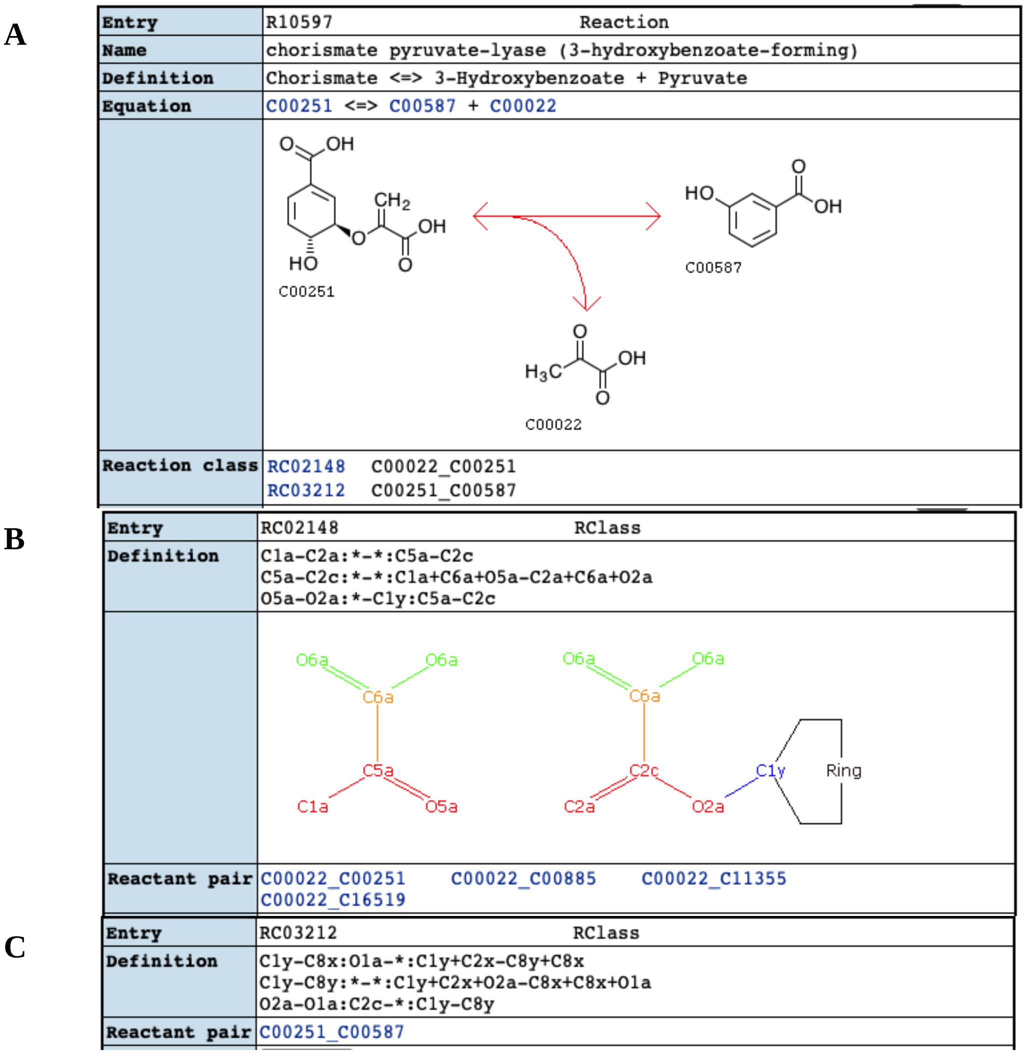


Figure 3.14. Example of MetaCyc reaction with incorrect atom mappings.

MetaCyc 1.13.11.45-RXN (https://metacyc.org/META/NEW-IMAGE?object=1.13.11.45-RXN&&redirect=T).



Entry	R10597 Reaction
Name	chorismate pyruvate-lyase (3-hydroxybenzoate-forming)
Definition	Chorismate <=> 3-Hydroxybenzoate + Pyruvate
Equation	C00251 <=> C00587 + C00022
Reaction class	RC02148 C00022_C00251 RC03212 C00251_C00587

Entry	RC02148 RClass
Definition	C1a-C2a:*-*:C5a-C2c C5a-C2c:*-*:C1a+C6a+O5a-C2a+C6a+O2a O5a-O2a:*-C1y:C5a-C2c
Reactant pair	C00022_C00251 C00022_C00885 C00022_C11355 C00022_C16519

Entry	RC03212 RClass
Definition	C1y-C8x:O1a-*:C1y+C2x-C8y+C8x C1y-C8y:*-*:C1y+C2x+O2a-C8x+C8x+O1a O2a-O1a:C2c-*:C1y-C8y
Reactant pair	C00251_C00587

Figure 3.15. Example of KEGG reaction with incorrect mappings.

A) KEGG reaction R10579 (https://www.kegg.jp/entry/R10597); B) KEGG RCLASS RC02148 (https://www.kegg.jp/entry/RC02148); C) KEGG RCLASS 03212 (https://www.kegg.jp/entry/RC03212).

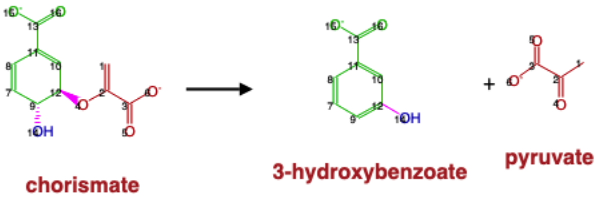


Figure 3.16. Atom mappings of MetaCyc reaction RXN-14940.

RXN-14940 (https://metacyc.org/META/NEW-IMAGE?object=RXN-14940&&redirect=T).

3.4 Discussion

Effective integration of compound and reaction from various sources is hard to achieve due to incomplete and inconsistent atom-level and bond-level details, like R groups and stereochemistry, across databases. First, we categorized compounds into specific and generic compounds based on the presence of R groups. Meanwhile, metabolic reactions were classified into specific and generic reactions according to the presence of generic compounds. To overcome inconsistent atomistic characteristics, a set of relationships between compounds were defined to both keep chemical details and conserve compound pairs at various levels. According to the degree of consistency, compound pair relationships are classified into three types: equivalence, generic-specific, and loose relationships. The majority (around 60%) of specific compound pairs have loose relationships, confirming the inconsistent issues in the databases to some extent. To our knowledge, no chemical identifier can be used to directly harmonize generic compounds across databases. Here, we further optimized a subgraph isomorphism detection algorithm to validate generic compound pairs. We first made use of the metabolic reactions to discover possible generic compound pairs. After validation, 2669 generic compound pairs remained. In addition, we developed pragmatic methods to validate tautomers and compounds with linear and circular representations. We discovered 45 compound pairs of compounds with the same chemical formula but fundamentally different structures, for example linear versus circularized chemical representations. In total, 15,704 harmonized compound pairs were detected, which dwarfs our prior best published compound harmonization result of 8865 harmonized compound pairs and 5681 harmonized compound pairs identified by prior identifiers and methods. Next, we mapped atom-resolved metabolic reactions across

KEGG and MetaCyc via compound pairs and EC numbers. Reaction pairs were also catalogued into hierarchical relationships in accordance with the classification of compound pairs. About 3856 harmonized reaction pairs were detected, and 10% of them can be missed by mismatched EC numbers (Figure 3.5 & 3.6), strongly suggesting that curation of EC numbers is of great importance in reaction harmonization. A prior systematic comparison of KEGG and MetaCyc had detected only 1961 shared reactions; however, this comparison was published in 2013[125]. The BRaunschweig ENzyme Database (BRENDA) indicates in a 2019 paper that 6115 reactions are harmonizable between KEGG and MetaCyc[126]. However, BRENDA uses a combination of text mining and prediction algorithms to build their database from primary literature, likely making their harmonization results not as chemically specific as the results presented here which directly analyzes molfiles provides by KEGG and MetaCyc.

Furthermore, we made use of the atom identifiers derived from our neighborhood-specific graph coloring method to evaluate the consistency of atom mappings across harmonized reactions. About 88% of reaction pairs have consistent atom mappings. For the 12% of harmonized and comparable reactions that are inconsistent, we do not have ground truth for determining which version of the reaction is correct. However, the fraction of changed local atom color provides a uses metric for suggesting which version has higher confidence. Additionally, given that these reaction descriptions represent reactions across thousands of organisms, it is possible that both versions are correct in different organisms. Additionally, we determined that both databases contain issues leading to inconsistency. For example, atoms in some MetaCyc reactions are not mapped correctly. For KEGG, we detected unsynchronized atom numbering in the older KEGG RPAIR representation, which is likely the reason that KEGG removed RPAIRS from their public version of the database. In contrast, the KEGG RCLASS provides a concise RDM representation of reaction atom mappings between a reaction-product compound pair, which appears highly resistant to consistency errors. This resistance to consistency error is due to a decoupling of the atom mappings from the specific atom order in the molfile representations. This allows the molfile representations to be minorly updated without having to update the RDM descriptions. However, there are also some issues with RDM descriptions. About 76 compound pairs cannot be parsed due to the incorrect description of reaction centers, and parsed compound pairs can be unreasonable at reaction level. In addition, a few KEGG RCLASS entries are computationally difficult to decipher due to a combinatorial issue caused by the several factors: multiple reaction centers in a single reaction, symmetric compounds, and reaction descriptions involving multistep reactions. This combination of factors introduces a large number of possibilities with matching a list of RDM descriptions to specific reaction center atoms. One way to prevent this combinatorial problem is to represent multiple reaction center atoms with their associated difference atoms and match atoms within a paired substructure representation instead of a list of RDM descriptions. Figure 3.15 B illustrates this paired substructure representation for the KEGG RCLASS RC02148. This kind of paired RDM substructure representation would allow the use of an efficient subgraph isomorphism detection method to derive the atom mappings and could be represented as a pair of molfiles along with a mapping of atoms between the two molfiles, all stored within a single sdfile. Additionally, our compound harmonization method for harmonizing changeable compound pairs would be useful for updating the paired RDM substructures when the compound representations dramatically change.

In addition, the methods we developed can be easily applied to integrate other metabolic databases that provide molfile representations of compounds, facilitating the expansion of the existing metabolic networks. Moreover, this hierarchical framework for relating compounds and reactions is a possible first step towards creating a systematic organization of all reaction descriptions at a desired chemical specificity to fit a given application. Such a systematic organization of reaction descriptions would augment the current Enzyme Commission number system and be useful to a wide range of possible applications from metabolic modeling, metabolite and reaction prediction, and network incorporation of newly discovered metabolites.

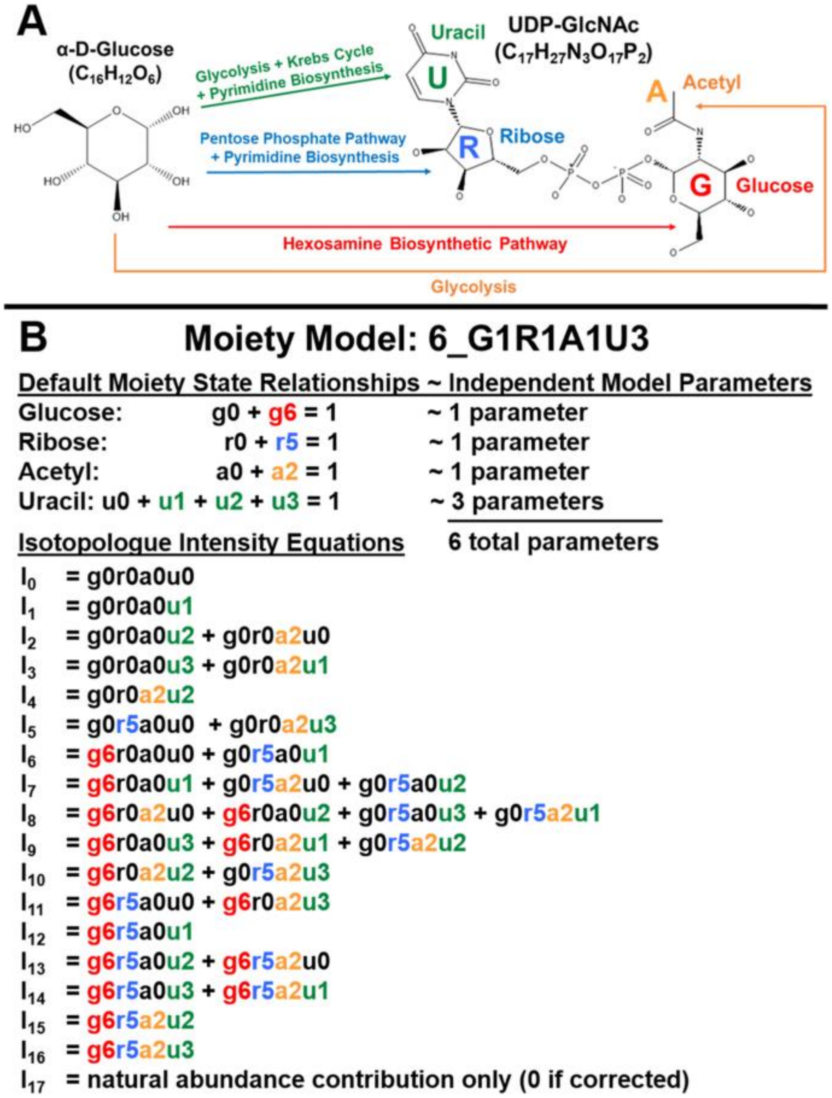
CHAPTER 4. MOIETY MODELING FRAMEWORK FOR DERIVING MOIETY ABUNDANCES FROM MASS SPECTROMETRY MEASURED ISOTOPOLOGUES

4.1 Introduction

Recent work indicates that many human diseases involve metabolic reprogramming that disturbs normal physiology and causes serious tissue dysfunction[127]. Advances in analytical technologies, especially mass spectroscopy (MS) and nuclear magnetic resonance (NMR), have made metabolic analysis of human diseases a reality[48]. Stable isotope tracing is a powerful technique that enables the tracing of individual atoms through metabolic pathways. Stable isotope-resolved metabolomics (SIRM) uses advanced MS and NMR instrumentation to analyze the fate of stable isotopes traced from enriched precursors to metabolites, providing richer metabolomics datasets for metabolic flux analyses. NMR can measure isotopomer-specific metabolite data, but is typically limited by sensitivity. Often a single piece of NMR data only provides information on the presence of stable isotopes in just a part of a metabolite, which represents a partial isotopomer. In some cases, multiple partial isotopomer information can be interpreted in terms of a full isotopomer. MS can measure isotopologue-specific data; however, an isotopologue represents a set of mass-equivalent isotopomers. Comprehensive metabolic analysis often relies on MS metabolic datasets or a combination of MS and NMR metabolic datasets. Even though large amounts of metabolomics datasets have been generated recently, it is still a big challenge to acquire meaningful biological interpretation from MS raw data, especially for complex metabolites composed of multiple subunits or moieties.

To better interpret complex isotopologue profiles of large composite metabolites, both quantitative analysis as well as detailed modeling are required. Several methods have been developed for quantitative flux analysis of specified pathways based on the stable isotope incorporated data, like the elementary metabolite units (EMU) framework[56]. These methods rely heavily on well-curated metabolic networks to accomplish the metabolic flux analysis. However, models of cellular metabolism, even for human, are far from complete.

To deconvolute the relative isotope incorporation fluxes of complex metabolites, first a plausible model of isotope incorporation should be built based on a relevant metabolic network, which is often incomplete. For example, the complex metabolite uridine diphosphose N-acetyl-D-glucosamine (UDP-GlcNAc), illustrated in Figure 4.1 A, has four distinct moieties in which ^{13}C isotopes incorporate through a metabolic network from an isotope labeling source like ^{13}C-labeled glucose. Based on the well-studied metabolic pathways that trace from glucose to UDP-GlcNAc in human metabolism, the expected (expert-derived) moiety model of ^{13}C isotope incorporation from ^{13}C-labeled glucose is illustrated in Figure 4.1 B, which includes ^{13}C incorporation states for each moiety. For example, the g6 state represents the incorporation of $^{13}C_6$ into the glucose moiety. Furthermore, the sum of moiety states for a given moiety is equal to 1. With this moiety model, a UDP-GlcNAc isotopologue profile can be deconvoluted into relative ^{13}C isotope incorporation into each UDP-GlcNAc moiety: glucose, ribose, uracil, and acetyl. The deconvolution occurs by minimizing an objective function that compares calculated isotopologues based on moiety isotope incorporation (enrichment) state parameters from the model to the directly observed, experimentally-derived isotopologues. From a mathematics perspective, the minimization represents a highly non-linear inverse problem, since the experimental intensities are compared to calculated values from nonlinear equations that use model parameters being optimized (Figure 4.1 B). With a time-series of isotopologue profiles, relative isotope fluxes for each moiety can be derived and used for the interpretation of isotope flux through specific metabolic pathways associated with each moiety. However, when multiple models are plausible, development of a robust model selection method is essential for successful isotopologue deconvolution, especially for non-model organisms. This basic approach to isotopologue deconvolution was demonstrated in a prototype Perl program called GAIMS for the metabolite UDP-GlcNAc using a MS isotopologue profile derived from a prostate cancer cell line[68]. This demonstration derived relative ^{13}C isotope fluxes for several converging biosynthetic pathways of UDP-GlcNAc under non-steady-state conditions. This demonstration also inspired the development of MAIMS, a software tool for metabolic tracer analysis[128], which further validates the robustness of the moiety model deconvolution method. However, the MAIMS software handles only ^{13}C single isotope tracer data and does not address model selection, which is crucial for addressing incomplete knowledge of cellular metabolic networks.


A
α-D-Glucose
($C_{16}H_{12}O_6$)
Glycolysis + Krebs Cycle
+ Pyrimidine Biosynthesis
Pentose Phosphate Pathway
+ Pyrimidine Biosynthesis
Uracil
U
UDP-GlcNAc
($C_{17}H_{27}N_3O_{17}P_2$)
R
Ribose
A
Acetyl
G
Glucose
Hexosamine Biosynthetic Pathway
Glycolysis

B Moiety Model: 6_G1R1A1U3

Default Moiety State Relationships ~ Independent Model Parameters

Glucose:	g0 + g6 = 1	~ 1 parameter
Ribose:	r0 + r5 = 1	~ 1 parameter
Acetyl:	a0 + a2 = 1	~ 1 parameter
Uracil:	u0 + u1 + u2 + u3 = 1	~ 3 parameters
		6 total parameters

Isotopologue Intensity Equations

I_0 = g0r0a0u0
I_1 = g0r0a0u1
I_2 = g0r0a0u2 + g0r0a2u0
I_3 = g0r0a0u3 + g0r0a2u1
I_4 = g0r0a2u2
I_5 = g0r5a0u0 + g0r0a2u3
I_6 = g6r0a0u0 + g0r5a0u1
I_7 = g6r0a0u1 + g0r5a2u0 + g0r5a0u2
I_8 = g6r0a2u0 + g6r0a0u2 + g0r5a0u3 + g0r5a2u1
I_9 = g6r0a0u3 + g6r0a2u1 + g0r5a2u2
I_{10} = g6r0a2u2 + g0r5a2u3
I_{11} = g6r5a0u0 + g6r0a2u3
I_{12} = g6r5a0u1
I_{13} = g6r5a0u2 + g6r5a2u0
I_{14} = g6r5a0u3 + g6r5a2u1
I_{15} = g6r5a2u2
I_{16} = g6r5a2u3
I_{17} = natural abundance contribution only (0 if corrected)

Figure 4.1. Example complex metabolite UDP-GlcNAc and associated expert-derived moiety model.

A) Major human metabolic pathways leading from glucose to the four moieties of UDP-GlcNAc. B) The representative moiety model is based on the expected metabolic tracing from ^{13}C-labeled glucose to UDP-GlcNAc, with the exception of one carbon in the uracil moiety that traces from carbon dioxide. The moiety states variables are identified by a lowercase moiety letter followed by a number representing the ^{13}C isotope content. The moiety state variables (model parameters) are used to calculate specific components of the relative isotopologue intensity.

In addition, the simultaneous use of multiple stable isotopes in SIRM experiments can provide much more data than a single tracer. However, incorporation of multiple stable isotopes also complicates the analysis of metabolite isotopologue profiles, which limits most of the current isotope tracer experiments to a single tracer. The lack of data analysis tools greatly impedes the application of the multiple-labeled SIRM experiments. Therefore, we have developed a new moiety modeling framework for deconvoluting MS isotopologue profiles for both single and multiple-labeled SIRM MS datasets. This moiety modeling framework not only solves the non-linear deconvolution problem, but also facilitates selection of the optimal model describing the relative isotope fluxes for a specific metabolite(s) from a set of plausible models.

4.2 Implementation

4.2.1 Overview of the moiety modeling framework

The workflow of the moiety modeling framework is composed of four major steps, model and data representation, model (parameter) optimization, analysis of optimization results, and model selection (Figure 4.2). For the model and data representation step, the moiety_modeling package creates an internal representation of a moiety model from a given JSONized moiety model description (see Figure 4.3). In this representation illustrated by a unified modeling language (UML) class diagram in Figure 4.4, the package first dissembles a complex metabolite into a list of moieties, i.e. metabolic subunits. Each moiety may contain different number of labeling isotopes, representing the flow of isotope from the labeling source to the moiety. A moiety with a specific number of labeled isotopes is represented as an isotope enrichment state of the moiety (i.e. moiety state). As specified in the JSONized model description, non-default mathematical relationships may exist between moiety states, even from different moieties and/or molecules. Molecules, their moieties, the possible moiety states, and relationships between moiety states work together to represent a particular moiety model, and the proportion for each possible moiety state is an optimizable parameter of the model. Each mass spectrum's worth of isotopologue data is represented as a separate dataset, which holds the set of isotopologues associated with each molecule. Typically, multiple mass spectra are included. Often each mass spectrum represents a single time point in a time series experiment.

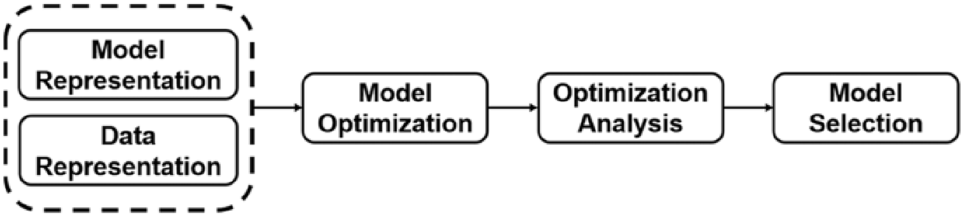


Figure 4.2. Workflow of the moiety modeling framework.

Moiety

```
{
   “name”: “ribose”,
   “nickname”: “r”,
   “ranking”: 1,
   “maxIsotopeNum”: {“13C”: 5},
   “isotopeStates”: [1, 129],
   “states”: [ “13C0”, “13C2”, “13C3”, “13C5” ] ,
   “py/object”: “moiety_modeling.model.Moiety”
 }
```

Molecule

```
{
   “name”: “UDP_GlcNAC”,
   “moieties”: [{“py/id”: 13}, …],
   “standardStates”: {“13C0”: [ [ “ribose[13C0]”, “glucose[13C0]”, “acetyl[13C0]”,
          “uracil[13C0]”]], “13C10”: [ [“ribose[13C0]”, “glucose[13C6]”, “acetyl[13C2]”,
          “uracil[13C2]”], [“ribose[13C5]”, “glucose[13C0]”, “acetyl[13C2]”,
          “uracil[13C3]”], …], …},
   “allStates”: [ “13C0”, “13C1”, “13C2”, “13C3”, “13C4”, “13C5”, “13C6”, “13C7”,
“13C8”,
          “13C9”, “13C10”, “13C11”, “13C12”, “13C13”, “13C14”, “13C15”, “13C16”,
“13C17”],
   “py/object”: “moiety_modeling.model.Molecule”
}
```

Relationship

```
{
   “moiety”: {“py/id”: 13},
   “moietyState”: “13C0”,
   “varName”: “glucose[13C0]”,
   “equivalentMoiety”: {“py/id”: 10},
   “equivalentMoietyState”: “13C0”,
   “equivalentVarName”: “ribose[13C0]”,
   “multiplier”: 1,
   “py/object”: “moiety_modeling.model.Relationship”
}
```

Figure 4.3. JSONized moiety model description.

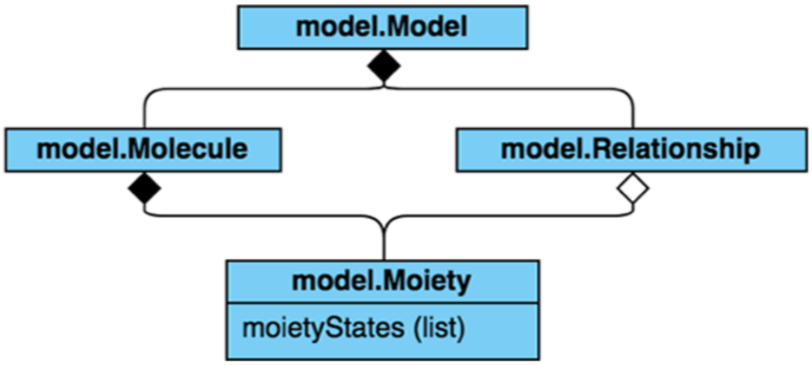


Figure 4.4. A unified modeling language (UML) class diagram of a Moiety Model.

The next major step, moiety model (parameter) optimization, involves deriving an optimal set of model parameters, i.e. moiety state fractional abundances ($moiety\_state_{j,i}$ for moiety j and state i) that are used to calculate relative isotopologue abundances ($I_{x,calc}$ from Equation 1) that best match experimental isotopologue profiles ($I_{x,obs}$) as compared by an objective function (see Table 4.1). In Equation 1, ic_a is a component of the isotopologue intensity with an isotope content x. Figure 4.1B lists these isotopologue components for each isotopologue based on the expert-derived moiety model.

$$I_{x,calc} = \sum_{ic_a \in IC_x} ic_a \;\; ; \; IC_x = \{ic_v | isotope\_content(ic_v) = x\} \; ; \; ic_v = \prod_j moiety\_state_{j,v_j} \qquad (1)$$

Table 4.1. Different forms of objective function

Loss function	**Equation**
Absolute difference	$\Sigma\lvert I_{x,obs} - I_{x,calc}\rvert$
Log difference	$\Sigma\lvert \log(I_{x,obs}) - \log(I_{x,calc})\rvert$
Square difference	$\Sigma(I_{x,obs} - I_{x,calc})^2$

The moiety_modeling package implements several optimization methods, including a combined simulated annealing and genetic algorithm (SAGA) based on the

‘Genetic Algorithm for Isotopologues in Metabolic Systems’ (GAIMS) Perl implementation[68] , a truncated Newton algorithm (TNC)[130], a SLSQP algorithm using Sequential Least Squares Programming[131], and a L-BFGS-B algorithm[132]. For the latter three algorithms ‘TNC’, ‘SLSQP’, and ‘L-BFGS-B’, the moiety_modeling package uses the implementation from the scipy.optimize Python module. In addition, we have the option to optimize the datasets together or separately.

The third major step involves the analysis of the results from the model optimization. The moiety_modeling package provides facilities for generating summative statistics and graphical visualizations for a set of optimizations performed on one or more moiety models. The final major step, model selection, tries to find the model that best fits the experimental isotopologue profiles from a set of provided moiety models that have been optimized in step two. Several forms of the Akaike information criterion (AIC)[133] and Bayesian information criterion (BIC)[134] are used as the estimator of the relative quality of moiety models for the set of isotopologue data.

4.2.2 The moiety_modeling Python package implementation

As shown in Figure 4.5, the moiety_modeling Python package consists of several modules: ‘model.py’, ‘modeling.py’, ‘analysis.py’, and ‘cli.py’. The ‘model.py’ module contains class definitions for the basic elements in the moiety model. It is composed of ‘Moiety’, ‘Relationship’, ‘Molecule’ and ‘Model’ classes. The ‘Moiety’ object represents a specific moiety, the labeling isotopes present in the moiety, and their corresponding states within the moiety. The ‘Relationship’ class describes the non-default mathematical dependencies between moiety states, where the default dependency for a given moiety is that the sum of its states is equal to 1 (see Figure 1B for example default relationships). A

‘Molecule’ object represents an individual metabolite made up of a list of ‘Moiety’ objects. The ‘Model’ class simulates the flow of isotope from labeling sources into each moiety of specific metabolites, which is initialized by lists of ‘Moiety’ objects, ‘Molecule’ objects, and ‘Relationship’ objects. A moiety model is generated and stored in a JSONized representation using the jsonpickle Python package[135]. This JSONized representation (see Supplementary Data 4.1), stored in a file, is then used as the input file for later model optimizations. The ‘modeling.py’ module is responsible for model optimization. It is composed of the ‘Dataset’ class, several model optimization classes, and the ‘OptimizationManager’ class. The ‘Dataset’ class organizes a single MS isotopologue profile dataset into a dictionary-based data structure. ‘Dataset’ objects are stored in a JSONized representation (see Supplementary Data 4.2) and used as the input for later model optimizations. Currently, no relationship between Dataset objects like a time-dependence is captured. In the abstract ModelOptimization class, we included several different objective functions (see Table 4.1). In addition, there are four specific model optimization classes in the ‘modeling’ module that utilize different optimization methods and approaches for combining datasets. The ‘SAGAoptimization’ and ‘SAGAseparateOptimization’ classes use the SAGA-optimize Python package described in the next section for either combined optimization of model parameters across all datasets or separate optimizations of model parameters for each dataset. ‘ScipyOptimization’ and ‘ScipySeparateOptimization’ classes make use of optimization methods (‘TNC’, ‘SLSQP’, and ‘L-BFGS-B’) in the scipy.optimize module to conduct optimizations in either a combined or separate manner. The ‘OptimizationManager’ class is responsible for the management of the optimization process based on the input optimization parameters. The results for a model optimization are stored in a JSONized representation (see Supplementary Data 4.3) for further analysis. A text file is used to store the filepaths to all of the optimized models with certain optimization parameters. The filepath file is then used as the input for the ‘analysis.py’ module. The ‘analysis.py’ module has five classes: ‘ResultsAnalysis’, ‘ModelRank’, ‘ComparisonTable’, ‘PlotMoietyDistribution’ and ‘PlotIsotopologueIntensity’. The ‘ResultsAnalysis’ class is responsible for generating standard statistics from the results for a set of optimizations for a given model. The mean, standard deviation, minimum, and maximum value of each model parameter are calculated from a set of model optimizations performed on the same model. The calculated isotopologue intensities and their statistics based on the sets of optimized parameters are also generated. Furthermore, several quality estimators of each model, including different forms of the ‘AIC’ (Table 4.2), are computed for model selection. The AIC tends to select the model that has too many parameters when the sample size is small, leading to overfitting. The sample size corrected AIC (AICc) was developed to address this overfitting problem[136]. The Bayesian information criterion (BIC) is another commonly used criterion for model selection[137]. The ‘ResultsAnalysis’ objects with results for each model are stored in a JSONize representation (see Supplementary Data 4.4) for further analysis, along with a text report for readability. Also, an analysis filepath file containing the filepaths to the analysis JSON files of all models with the same optimization parameters is created. Next, the ‘ModelRank’ class object uses this analysis filepath file to compare and select the model that best reflects the observed isotopologue profile. The ‘ComparisonTable’ class compares the model selection results with different optimization parameters. The ‘PlotMoietyDistribution’ class and ‘PlotIsotopologueIntensity’ class are responsible for the visualization of the optimization results for a set of optimizations performed on a single model. The ‘cli.py’ module provides the command-line interface to perform model optimization, model optimization analysis, and model selection, which is implemented with the ‘docopt’ Python library[78].

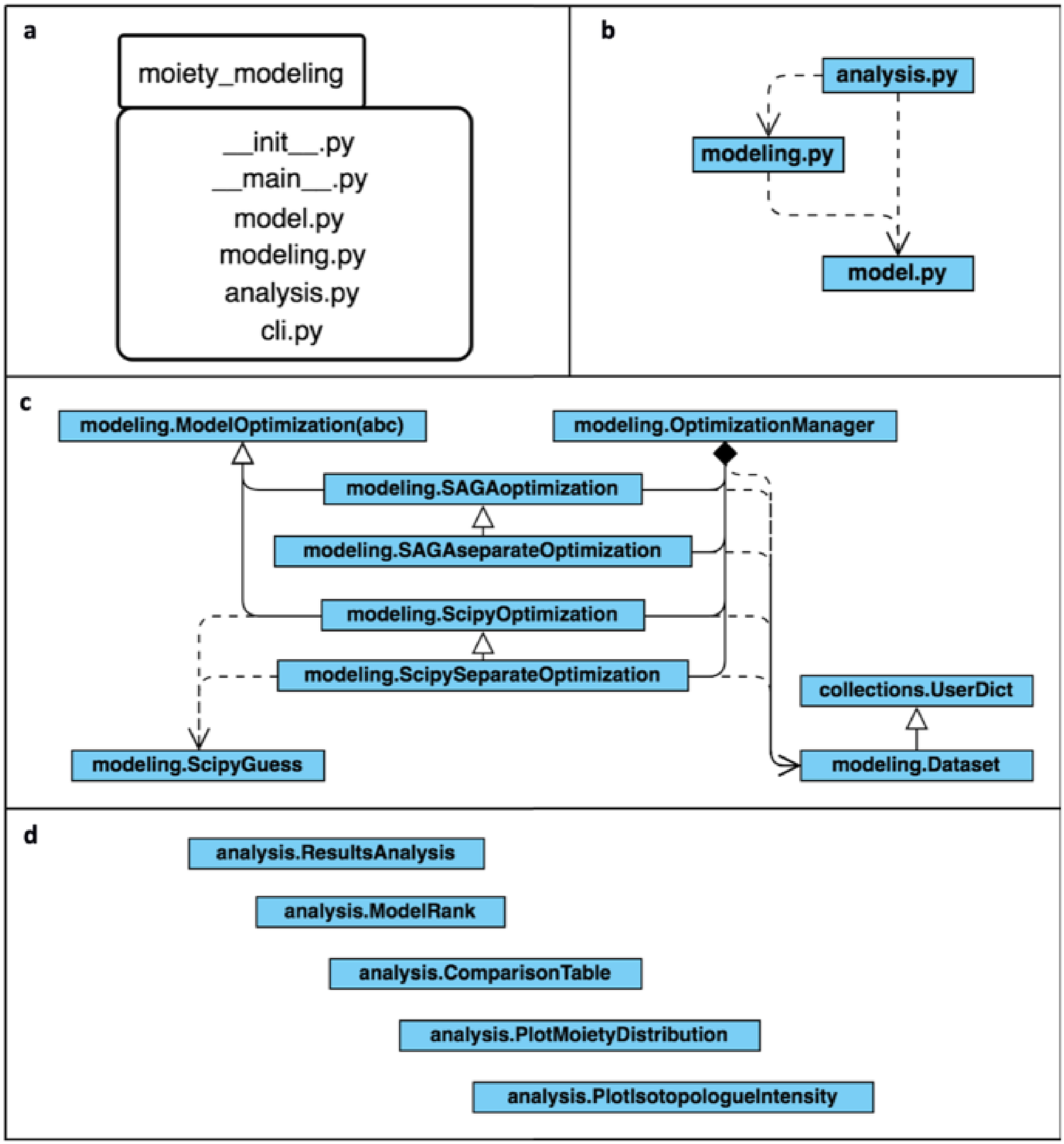

a
moiety_modeling
__init__.py
__main__.py
model.py
modeling.py
analysis.py
cli.py
b
analysis.py
modeling.py
model.py
c
modeling.ModelOptimization(abc)
modeling.OptimizationManager
modeling.SAGAoptimization
modeling.SAGAseparateOptimization
modeling.ScipyOptimization
modeling.ScipySeparateOptimization
collections.UserDict
modeling.ScipyGuess
modeling.Dataset
d
analysis.ResultsAnalysis
analysis.ModelRank
analysis.ComparisonTable
analysis.PlotMoietyDistribution
analysis.PlotIsotopologueIntensity

Figure 4.5. Organization of the moiety_modeling package represented with UML diagrams.

A) UML package diagram of the moiety_modeling Python library; B) Subpackage dependencies diagram; C) UML class diagram of the 'modeling.py' module with dependency relationships; D) UML class diagram of the 'analysis.py' module, which contains a set of classes with no relationships.

Table 4.2. Different forms of a model selection estimator

Selection Criterion	Equation
Akaike Information Criterion (AIC)	$2\mathrm{k} + \mathrm{n}\ln(\mathrm{RSS}/\mathrm{n})$
Sample size corrected AIC (AICc)	$\mathrm{AIC} + (2k^2 + 2\mathrm{k})/(\mathrm{n} - \mathrm{k} - 1)$
Bayesian Information Criterion (BIC)	$\mathrm{n}\ln(\mathrm{RSS}/\mathrm{n}) + \mathrm{k}\ln(\mathrm{n})$

k is the number of parameters.
n is the number of data points.
RSS is the residual sum of squares: RSS = $\sum_{i=1}^{n}(I_{obs} - I_{calc})^2$.

4.2.3 SAGA-optimize Python package implementation

The SAGA-optimize Python package is a novel type of combined simulated annealing and genetic algorithm[68] used to find the optimal solutions to a set of parameters based on the minimization of a given energy (objective) function calculated using the set of parameters. In this context, the energy function represents a comparison of calculated and experimentally-observed isotopologue relative intensities, with the calculated intensities based on the moiety model parameters being optimized. As shown in Figure 4.6, it is composed of ‘ElementDescription’, ‘Guess’, ‘Population’ and ‘SAGA’ classes. An ‘ElementDescription’ object describes an individual parameter of the moiety model. In the expert derived moiety model (Figure 4.1B), the g6 model parameter would be represented by a single ‘ElementDescription’ object. The ‘ElementDescription’ object is bound by a range and several mutation methods are available to change the value of the ‘ElementDescription’ object. A ‘Guess’ object contains lists of all the parameters (‘ElementDescription’ objects) and their corresponding values for a particular moiety model. In addition, it also stores the energy calculated based on this set of parameters. A ‘Population’ object contains information of a list of ‘ElementDescription’ objects, a list of ‘Guess’ objects, the range of each ‘ElementDescription’ among all the ‘Guess’ objects, the highest and lowest energy for the list of ‘Guess’ objects, and the best ‘Guess’ object. The ‘ElementDescription’, ‘Guess’ and ‘Population’ classes are the building blocks of the

‘SAGA’ class, which is the main class that provides the interface for optimization. Furthermore, several distinct crossover functions are available for creating new Guess objects from the cross-over of two other Guess objects.

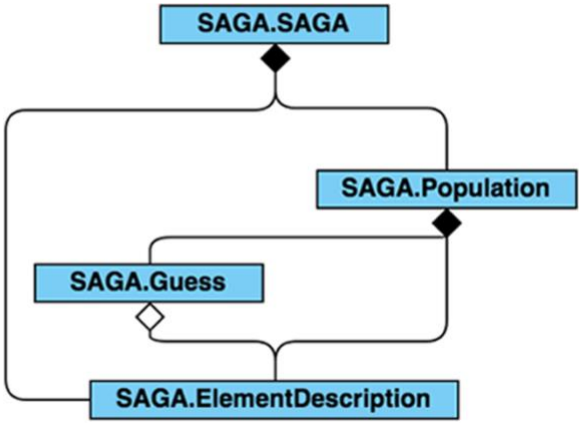


Figure 4.6. ‘SAGA-optimize’ package represented with a UML class diagram with dependencies.

4.3 Results

4.3.1 The package interface

The moiety_modeling package can be used in two main ways: (i) as a library within Python scripts for accessing and manipulating moiety models and isotopologue datasets stored in JSON files, or (ii) as a command-line tool to perform model optimization, model analysis, and model selection.

To use the moiety_modeling package as a library within Python scripts, it should be imported with a Python program or an interactive interpreter interface. Next, ‘Moiety’, ‘Relationship’ and ‘Molecule’ objects can be created to construct a moiety model. ‘Dataset’ objects are also built with the moiety_modeling package. Table 4.3 summarizes common patterns for using moiety_modeling package as a library in construction of a moiety model and related datasets.

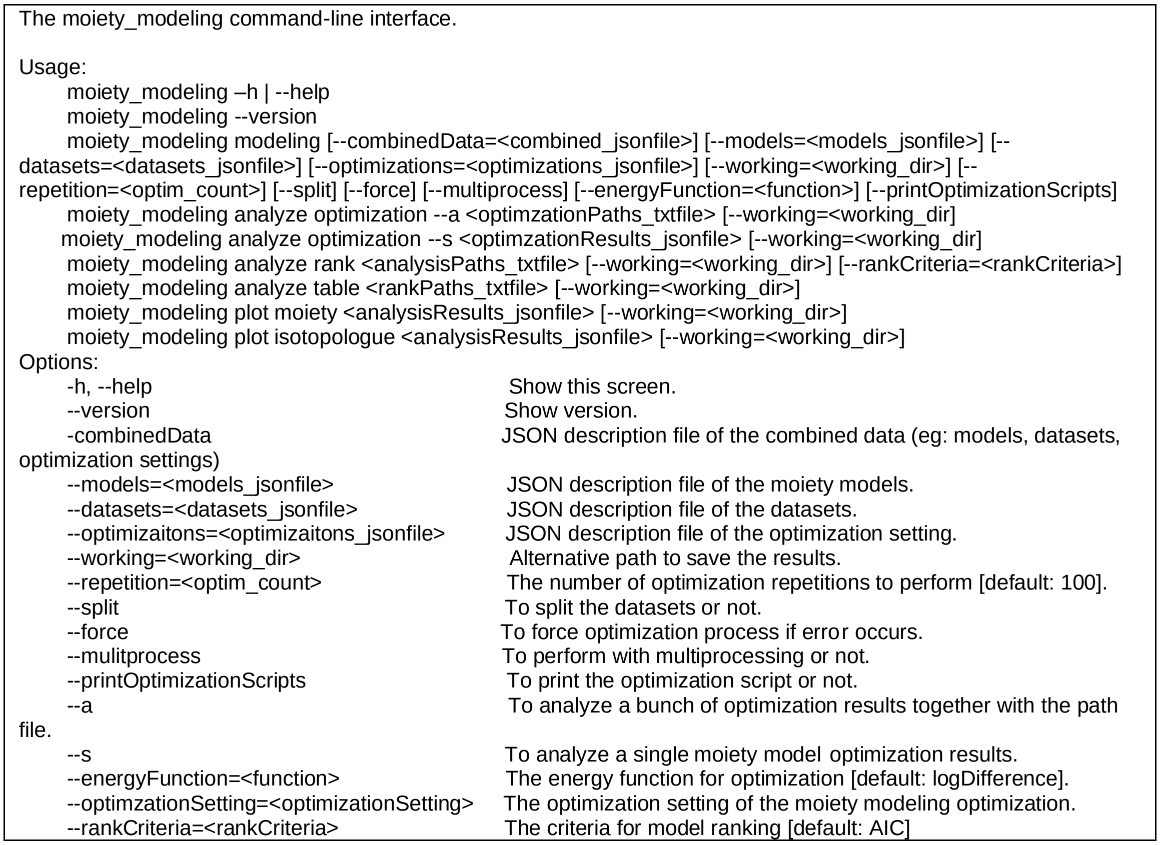
The moiety_modeling package also provides a simple command-line interface to perform model optimization, selection, and visualization. Figure 4.7 shows version 1.0 of the command-line interface, and Table 4.4 summarizes common pattern for using moiety_modeling as a command-line tool. The common patterns for using SAGA-optimize as a library are shown in Table 4.5.

Table 4.3. Common creation patterns for the moiety_modeling library

Entity	**Example**

Moiety	glucose = moiety_modeling.Moiety(‘glucose’, {‘13C’: 6}, isotopeStates={‘13C’: [1, 3, 5]}, nickname= ‘g’)
	acetyl = moiety_modeling.Moiety(‘acetyl’, {‘13C’: 2}, isotopeStates={‘13C’: [0, 1, 2]}, nickname= ‘a’)
	uracil = moiety_modeling.Moiety(‘uracil’, {‘13C’: 4}, isotopeStates={‘13C’: [1, 2, 4]}, nickname= ‘u’)
	ribose = moiety_modeling.Moiety(‘ribose’, {‘13C’: 5}, isotopeStates={‘13C’: [0, 3, 5]}, nickname= ‘r’)

Relationship	relationship = moiety_modeling.Relationship(glucose, ‘13C0’, acetyl, ‘13C2’, ‘*’, 2)
Molecule	UDP-GlcNAc = moiety_modeling.Molecule(‘UDP-GlcNAc’, [glucose, uracil, acetyl, ribose])
Model	model1 = moiety_modeling.Model(‘model1’, [glucose, uracil, acetyl, ribose], [UDP_GlcNAc], [relationship])
Dataset	dataset = moiety_modeling.Dataset(‘12h’, ‘UDP_GlcNAc’: [{‘labelingIsotopes’:’13C_0’, ‘height’: 0.0175, ‘heightSE’: 0 }, {‘labelingIsotopes’:’13C_1’, ‘height’: 0.0075, ‘heightSE’: 0 }, …])



```
The moiety_modeling command-line interface.

Usage:
        moiety_modeling –h | --help
        moiety_modeling --version
        moiety_modeling modeling [--combinedData=<combined_jsonfile>] [--models=<models_jsonfile>] [--
datasets=<datasets_jsonfile>] [--optimizations=<optimizations_jsonfile>] [--working=<working_dir>] [--
repetition=<optim_count>] [--split] [--force] [--multiprocess] [--energyFunction=<function>] [--printOptimizationScripts]
        moiety_modeling analyze optimization --a <optimzationPaths_txtfile> [--working=<working_dir]
       moiety_modeling analyze optimization --s <optimzationResults_jsonfile> [--working=<working_dir]
        moiety_modeling analyze rank <analysisPaths_txtfile> [--working=<working_dir>] [--rankCriteria=<rankCriteria>]
        moiety_modeling analyze table <rankPaths_txtfile> [--working=<working_dir>]
        moiety_modeling plot moiety <analysisResults_jsonfile> [--working=<working_dir>]
        moiety_modeling plot isotopologue <analysisResults_jsonfile> [--working=<working_dir>]
Options:
        -h, --help                                       Show this screen.
        --version                                        Show version.
        -combinedData                                    JSON description file of the combined data (eg: models, datasets,
optimization settings)
        --models=<models_jsonfile>                       JSON description file of the moiety models.
        --datasets=<datasets_jsonfile>                   JSON description file of the datasets.
        --optimizaitons=<optimizaitons_jsonfile>         JSON description file of the optimization setting.
        --working=<working_dir>                          Alternative path to save the results.
        --repetition=<optim_count>                       The number of optimization repetitions to perform [default: 100].
        --split                                          To split the datasets or not.
        --force                                          To force optimization process if error occurs.
        --mulitprocess                                   To perform with multiprocessing or not.
        --printOptimizationScripts                       To print the optimization script or not.
        --a                                              To analyze a bunch of optimization results together with the path
file.
        --s                                              To analyze a single moiety model optimization results.
        --energyFunction=<function>                      The energy function for optimization [default: logDifference].
        --optimzationSetting=<optimizationSetting>       The optimization setting of the moiety modeling optimization.
        --rankCriteria=<rankCriteria>                    The criteria for model ranking [default: AIC]
```

Figure 4.7. The ‘moiety_modeling’ package command line interface.

Table 4.4. Common patters for using the moiety_modeling as a command-line tool

Command	Description	Example
modeling	Perform model optimization	% python3 –m moiety_modeling modeling --models=models.json --datasets=dataset.json --optimizations=optimization_settings.json
analyze	Analyze the optimization results	% python3 –m moiety_modeling analyze optimizations --a optimizationPaths.txt
plot	Plot the distribution of calculated moiety modeling parameters.	% python3 –m moiety_modeling plot moiety analysisResults.json

Table 4.5. Common patterns for using 'SAGA' module as a library.

Usage	Example

SAGA	saga = SAGA.SAGA(stepNumber=100, temperature=10, startTemperature=0.5, alpha=1, energyfunction=targertedEnergyFunction)
	saga.addElmentDescriptions(SAGA.ElementDecription(low=0, high=1))

Population	population = saga.optimize()
Guess	bestGuess = population.bestGuess

4.3.2 Dataset and model

We used the timecourse (34h, 48h, and 72h) of ^{13}C isotopologue data for UDP-GlcNAc generated from [U-^{13}C]-glucose in human prostate cancer LnCaP-LN3 cells to evaluate the robustness of the moiety modeling framework. An expert-derived moiety model of UDP-GlcNAc (6_G1R1A1U3) was created based on known human biochemical pathways (Figure 4.1A) and corroborated by NMR data. Also, 40 hypothetical moiety models of the isotopic flow into UDP-GlcNAc were crafted as simple perturbations of the original expert-derived model. These perturbations include the inclusion of different and/or additional moiety states and non-default moiety state relationships (e.g. g6 = r5). For example, model 7_G2R1A1U3_g5 includes an extra $^{13}C_5$ g5 glucose moiety state for a total of 7 independent model parameters, 2 for glucose, 1 for ribose, 1 for acetyl, and 3 for uracil. We tested whether the expert-derived moiety model could be selected from all the other models.

4.3.3 Model optimization and selection

The incorporation of ^{13}C from [U-^{13}C]-glucose into UDP-GlcNAc leads to a total of 17 isotopologues plus one due to ^{13}C natural abundance from carbon dioxide (I_0, …, I_{17}). We applied the moiety modeling framework to the observed UDP-GlcNAc isotopologue data with each built model to test whether the expert-derived moiety model could be selected above the other models. We used the SAGA optimization method with a log difference objective function (see Table 4.1). The optimization was repeated 100 times for each model. These analyses were performed on a desktop computer with i7-6850K CPU (6 core with HT), 64GB RAM and 512GB SSD. On this hardware, the analyses for all 40 models took roughly 3 hours of total execution time. The results are list in the Table 4.6. From these results, we can see that the expert-derived moiety model can be selected successfully among all the moiety models using the AICc (see Table 4.2), which demonstrates the robustness of the moiety modeling framework. Model selection criteria like the AICc help to address model overfitting; however, the use of a log difference objective function with multiple time points of data in the form of separate sets of observed isotopologues makes the model selection very robust against most of the model overfitting[68]. We also compared the optimization results generated by the moiety-modeling package to results generated by GAIMS (see Supplementary Table 4.1 & 4.2, Supplementary Figure 4.1). For this comparison, an absolute difference objective function was used with the moiety-modeling package to match the objective function available in the GAIMS software. Also, there are some small differences in the implementation of optimization method between the two software packages. The SAGA-optimize package implements a true simulated annealing, while GAIMS implements a modified annealing with steepest decent qualities. Also, both optimization methods are stochastic as demonstrated by replicate moiety-modeling analyses shown in Supplementary Table 4.3. Therefore, the results are not identical; however, they are reasonably comparable. But neither method is able to select the expert-derived model with an AICc model selection method, due to issues of overfitting with the absolute difference objective function.

Table 4.6. Model selection results of UDP-GlcNAc isotopologue data

Model[a]	**Estimator (AICc)**
6_G1R1A1U3 (expert-derived model)	-229.2918
6_G1R1A1U3_r4	-227.5208
6_G1R1A1U3_u4	-225.0006
6_G0R2A1U3_g3r2r3_g6r5	-223.1633
6_G1R1A1U3_g5	-215.9565
7_G1R2A1U3_r1	-212.4727
7_G2R1A1U3_g1	-212.1217
7_G1R2A1U3_r3	-210.9640
7_G1R1A2U3	-210.0952
7_G2R1A1U3_g5	-208.1346
7_G1R2A1U3_g3r2r3	-207.6523
7_G1R2A1U3_r2	-207.4187
7_G2R1A1U3_g4	-206.6430
7_G2R1A1U3_g2	-206.5609
7_G0R2A2U3_g3r2r3_g6r5	-205.0569
7_G2R1A1U3_g3	-204.8797
7_G0R3A1U3_g3r2r3_g6r5_g5r4	-204.2729
7_G1R1A1U4	-203.3710
7_G1R2A1U3_r4	-202.6782
6_G1R1A1U3_a1	-199.5560
8_G2R1A2U3_g1	-195.9713
7_G1R1A1U3C1	-195.5788
8_G1R2A2U3_r1	-195.4893
7_G0R3A1U3_g3r2r3_g6r5_r4	-192.4980
8_G1R2A2U3_r2r3	-187.3342

8_G1R2A2U3_r3	-186.8810
8_G2R1A2U3_g5	-186.2693
8_G1R2A2U3_r2	-186.2562
8_G2R1A2U3_g2	-185.6112
8_G2R1A2U3_g4	-184.9444
8_G1R2A2U3_g3r2r3	-184.2929
8_G1R2A2U3_g3r2r3_g6r5_g5	-183.2154
8_G2R1A2U3_g3	-183.1467
8_G1R2A2U3_r4	-182.1334
8_G1R1A2U3C1	-177.5013
9_G2R2A2U3_r2r3_g1	-170.3323
9_G2R2A2U3_r2r3_g2	-161.5770
9_G2R2A2U3_r2r3_g3	-160.7823
9_G2R2A2U3_r2r3_g6r5_g3_g5	-160.6917
9_G2R2A2U3_r2r3_g4	-160.4500
9_G2R2A2U3_r2r3_g5	-158.8733

Optimization settings: method = ’SAGA’, SAGA_parameters = {‘stepNumber’: 100000, ‘temperatureStepSize’: 100, ‘alpha’: 1, ‘crossoverRate’: 0.05, ‘mutationRate’: 3, ‘populationSize’: 20, ‘startTemperature’: 0.5}, repetition=100, split, objective function=log difference.

[a]The first number in the model name is the total number of free model parameters followed by the number of free parameters for each moiety and perturbations from the expert-derived model.

4.3.4 Generation of simulated single-tracer and multi-tracer datasets

In addition, we generated simulated single tracer and multi-tracer datasets to test, compare, and evaluate multi-tracer optimization functionality. First, we created a set of rounded moiety state values for the single-tracer expert derived model roughly based on the optimized model state values derived from the experimental UDP-GlcNAc 48h dataset (Table 4.7).

Table 4.7. Single-tracer ^{13}C moiety states and values for UDP-GlcNAc biosynthesis

Moiety states	Moiety value	Moiety states	Moiety value
glucose[13C_0]	0.1	ribose[13C_5]	0.9
glucose[13C_6]	0.9	uracil[13C_0]	0.2
acetyl[13C_0]	0.7	uracil[13C_1]	0.2
acetyl[13C_2]	0.3	uracil[13C_2]	0.5
ribose[13C_0]	0.1	uracil[13C_3]	0.1

We then used ^{13}C and ^{18}O labeled glucose ($^{13}C_6H_{12}{}^{18}O_6$) as a hypothetical isotope labeling source for UDP-GlcNAc biosynthesis. Following the expert derived model and with the aid of atom-mapping information of relevant human biochemical reactions from MetaCyc[78], we traced the incorporation of oxygen and carbon atoms from glucose to each moiety to derived a multi-tracer model. For glucose, acetyl and ribose, oxygen atoms incorporated into the moiety with their directly bonded carbon atom. However, during the biosynthesis of uracil, some ^{18}O-^{13}C bonds are sometimes broken, creating a more varied set of moiety states. Next, we derived rounded multi-tracer moiety state values that are equivalent to the rounded single-tracer values (Table 4.8).

Table 4.8. Multi-tracer $^{13}C/^{18}O$ moiety states and values for UDP-GlcNAc biosynthesis

Moiety states	**Moiety value**	**Moiety states**	**Moiety value**
glucose[13C_0.18O_0]	0.1	uracil[13C_0.18O_0]	0.2
glucose[13C_6.18O_5]	0.9	uracil[13C_1.18O_0]	0.2
acetyl[13C_0.18O_0]	0.7	uracil[13C_2.18O_0]	0.25
acetyl[13C_2.18O_1]	0.3	uracil[13C_2.18O_1]	0.25
ribose[13C_0.18O_0]	0.1	uracil[13C_3.18O_0]	0.05
ribose[13C_5.18O_4]	0.9	uracil[13C_3.18O_1]	0.05

Next, we generated the base single-tracer and multi-tracer simulated datasets by calculating the set of relative isotopologue intensity values using Equation 1 with the respective moiety state values. Finally, we created simulated datasets with added normally distributed error that is subsequently thresholded to zero based on a minimum hypothetical detection limit (0.005) and then renormalized to a sum of 1. We generated three sets of 100 simulated datasets for both single and multi-tracer models by adding error from a normal distribution with increasing standard deviations of 0.001, 0.01 and 0.1. We then estimated the effects of error propagation by calculating the average sum of isotopologues across 100 simulated datasets after error addition and thresholding, but before renormalization (Table 4.9).

Table 4.9. Multi-tracer $^{13}C/^{18}O$ moiety states and values for UDP-GlcNAc biosynthesis

σ of Added Error	**Average Sum of Isotopologues**	
	Single-tracer	**Multi-tracer**
0.1	1.50	9.97
0.01	1.02	1.73
0.001	0.99	0.98

Based on this calculation, the single-tracer datasets and the multi-tracer datasets have comparable levels of propagated error when normal error with a 0.001σ is added. However, this quickly deviates with larger amounts of additive error as shown by single-tracer datasets with a 0.1σ added normal error having slightly less propagated error than the multi-tracer datasets with a 0.01σ added normal error. The multi-tracer datasets with a σ=0.1 added normal error are practically useless due to the level of propagated error being roughly nine (i.e. $9.97 - 1.00 = 8.97 \approx 9$) times the original signal on average. Using histograms of simulated intensities for the largest respective isotopologue in both the single-tracer and multi-tracer simulated datasets, Figure 4.8 illustrates these error propagation effects due to thresholding and renormalization. It is clear from this figure the loss of intensity information in the multi-tracer simulated dataset with σ=0.1 added normal error.

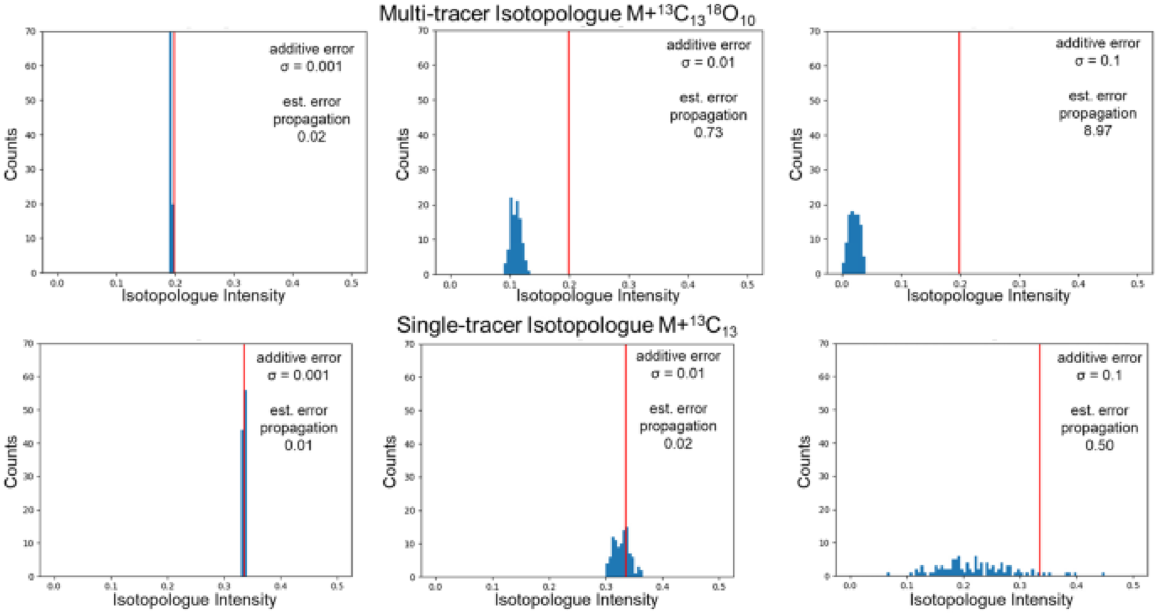


Figure 4.8. Histograms of simulated intensities for the largest representative isotopologue.

4.3.5 Model optimization of simulated multi-tracer and single-tracer datasets and comparison of results

For each simulated dataset consisting of a single time point, the respective model was optimized 100 times (i.e. in 100 separate repetitions), each using 5000 steps of SAGA with an absolute objective function. This generated 10,000 separate optimizations for each set of simulated datasets at a given added level of error. Using histograms, Figure 4.9 visualizes the distribution for the acetyl and uracil moiety state values for the multi-tracer dataset with 0.01σ added normal error and for the single-tracer datasets with σ=0.1 and σ=0.01 added normal error. The full set of histograms are in Supplementary Figure 4.2 for the multi-tracer results and Supplementary Figure 4.3 for the single tracer results. When comparing multi-tracer and single-tracer experiments with equivalent added normal error (σ=0.01), the propagated error leads to wider variances in the multi-tracer moiety state values and some additional skewness of their distributions. However, some of the single-tracer moiety state value distributions are bimodal. When comparing multi-tracer and single-tracer experiments with comparable propagated error levels, the multimodality in the single-tracer distributions become very pronounced, especially in the acetyl moiety states.

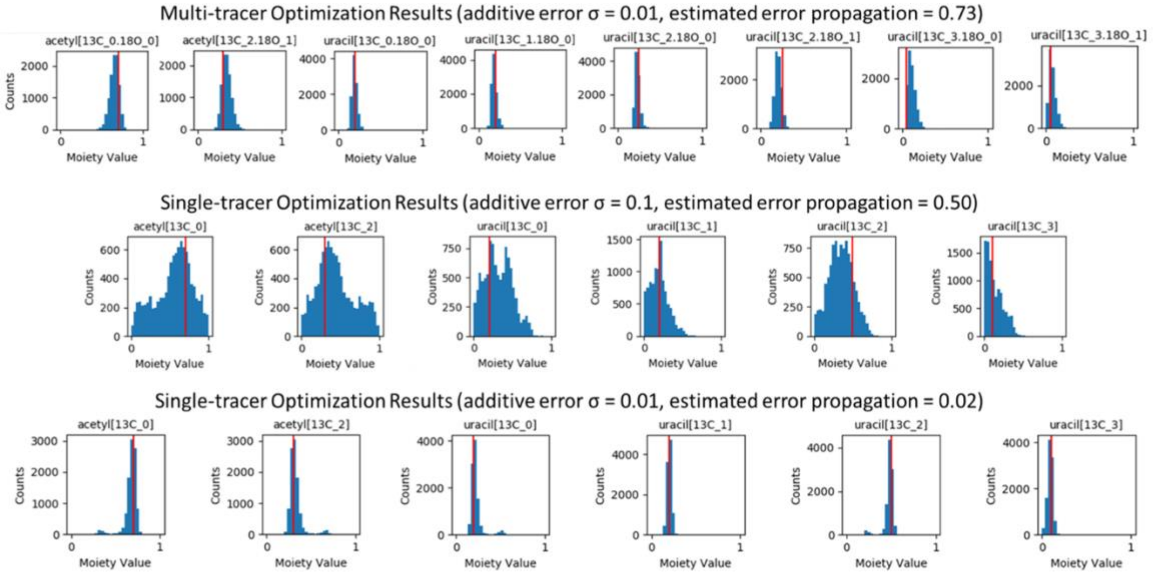


Figure 4.9. Histograms of the acetyl and uracil optimized moiety state values derived from simulated datasets.

4.4 Discussion

4.4.1 Advantage of JSONized representation for MS isotopologue data and analysis results

JavaScript object notation (JSON)[138] is an open-standard file format using human-readable text to collect data in pair-value and array structures, widely used by different programming language. Complex Python objects, like ‘Moiety’ and ‘Molecule’ objects mentioned above, can be serialized to JSON format with the jsonpickle Python library. The moiety model and dataset constructed with moiety_modeling package as well as optimization parameters are the input files for the moiety modeling, all of which are saved in JSON format using jsonpickle (see Supplementary Data 4.1, 4.2 and 4.5). The use of JSON format makes the moiety modeling framework easily accessible to other programming languages and naturally extendible. In addition, the optimization and analysis results are also stored in a JSON file (see Supplementary Data 4.3 & 4.4).

4.4.2 Advantages and limitations of the SAGA-optimize and moiety-modeling packages

The SAGA-optimize package provides certain advantages to the model optimization versus the other optimization methods from scipy and even a similar implementation in GAIMS. The level and steepness of optimization can be precisely tuned with the specification of the annealing length and schedule. Also, this novel implementation of a combined simulated annealing and genetic algorithm incorporates the annealing processing directly into the mutation step itself, attenuating the level of mutation as the annealing temperature drops. The moiety-modeling package provides a range of objective functions and can split each independent set of isotopologues into individual moiety model optimizations, which neither the GAIMS nor MAIMS packages can do. Moreover, both the SAGA-optimize and moiety-modeling packages have multiprocessing facilities that enable an efficient utilization of all CPU cores. As demonstrated in the Table 4.6 results, the combination of advantages allows the moiety-modeling package to optimize and accurately select the expert-derived model in roughly one tenth of the execution time of the original GAIMS package, i.e. with 100,000 steps of optimization in moiety-modeling versus 1,000,000 steps in GAIMS. Also, both the SAGA-optimize and moiety-modeling packages contain over 2200 lines of code implemented in major version 3 of the Python language with a fully object-oriented design and Pythonic style. Every module, class, method, and function have documentation strings (docstrings) written in the reStructuredText markup language. Variables, data members, methods, functions, and classes have descriptive names as demonstrated in Figures 4.4, 4.5, and 4.6. Documentation is automatically generated using the Sphinx Python Document Generator and made available on ReadTheDocs. This documentation includes a user guide, installation instructions, tutorial, and application programming interface (API) reference. Both packages are available on GitHub, utilize Travis CI for continuous integration, and are distributed via the Python Package Index. Code coverage from unit testing is above 65% for moiety-modeling and above 73% for SAGA-optimize. These packages enable researchers to perform moiety model isotopologue deconvolution using JSON representations of moiety models, datasets, and optimization method selection and settings provided by the user. At this time, the moiety-modeling package has no facilities for automatic moiety model generation.

4.4.3 Difficulty in generating simulated datasets and comparing multi-tracer to single-tracer moiety modeling results

The generation of realistic simulated biophysical datasets is always a non-trivial task[138]. Even the addition of normal additive error can create non-intuitive propagation of error, especially through inverse problems[139]. This is illustrated in Table 4.8 and Figure 4.8, where thresholding creates a positive bias in accumulated error and the renormalization creates a proportional-like error component from this positive accumulated error. The thresholding is required to keep the simulated data within the physical boundaries of the analytical detection, i.e. all non-negative values. The renormalization keeps the simulated data within mathematical boundaries, i.e. the sum of the isotopologue values is equal to 1. Neither step can be avoided with the inclusion of normal additive error. This created error propagation problem is quite dramatic for the simulated multi-tracer datasets, because there are 324 possible isotopologues in the multi-tracer datasets as compared to only 18 isotopologues in the single-tracer datasets. This problem simply increases in magnitude with the number of isotopologues present in a dataset. With a σ=0.1 added normal error, the isotopologue intensity information is effectively lost for the multi-tracer datasets (see Figure 4.8) and these datasets become effectively unusable (see Supplementary Figure 4.2). However, the lower additive error datasets are usable and illustrate the power of multi-tracer datasets to reduce multimodality in optimized moiety state values as compared to the single-tracer datasets.

4.5 Conclusions

Here, we present a moiety modeling framework for the deconvolution of metabolite isotopologue profiles using moiety models along with the analysis and selection of the best moiety model(s) based on the experimental data. This framework can analyze datasets involving single and multiple isotope tracers as demonstrated on simulated datasets for multiple tracer models and both simulated and experimental datasets on single tracer models. With a ^{13}C-labeled UDP-GlcNAc isotopologue dataset, we further demonstrate the robust performance of the moiety modeling framework for model selection on real experimental datasets. The selection of correct moiety models is required for generating deconvolution results that can be accurately interpreted in terms of relative metabolic flux. Furthermore, the JSON formats of moiety model, isotopologue data, and optimization results facilitate the inclusion of these tools in data analysis pipelines. Future work will explore the data quality requirements of model selection and validation of multiple isotope tracing model optimization and selection.

CHAPTER 5. ROBUST MOIETY MODEL SELECTION USING MASS SPECTROMETRY MEASURED ISOTOPOLOGUES

5.1 Introduction

While the first observations of metabolic alterations in cancer were made about a century ago[140], metabolomics is a relatively new field of ‘omics’ technology aiming to systematically characterize metabolites being created and/or utilized in cells, tissues, organisms, and ecosystems[141]. This combined consumption and biosynthesis of metabolites can be represented as flux through specific metabolic paths within cellular metabolism, reflecting specific physiological and pathological states in biomedically useful detail and in ways that are distinct and often more sensitive than other omics methods. It is increasingly recognized that metabolomics biomarkers have great utility in characterizing and monitoring diseases with significant metabolic reprogramming like cancer[127]. Therefore, better regulatory understanding of specific metabolic flux phenotypes of metabolic diseases will aid in developing new therapeutic strategies.

Stable isotope resolved metabolomics (SIRM) experiments utilize stable isotopes from a labeling source to isotopically enrich detected metabolite analytical features, providing more complex but data-rich metabolomics datasets for metabolic flux analysis. Advances in mass spectrometry (MS) and nuclear magnetic resonance spectroscopy (NMR) greatly contribute to the generation of high-quality SIRM datasets[48]. However, computational methods are required to gain biologically meaningful interpretation from such complex datasets, especially in terms of metabolic flux through specific metabolic paths in cellular metabolism. Most current metabolic flux analysis methods heavily depend on a predetermined metabolic network and are mostly focused on the analysis of ^{13}C tracer experiments[53, 56, 65, 142]. However, large numbers of ‘unknown’ metabolites in the metabolomics datasets strongly indicate that current metabolic network are far from complete, especially for secondary metabolism and central metabolism of non-model organisms[129, 143, 144]. Without an accurate and reasonably-defined metabolic network, it is challenging to conduct meaningful metabolic flux analyses. Even worse, assuming a metabolic model is accurate compromises the scientific rigor of the metabolic modeling and can lead to misinterpretation of results[139].

Our newly developed moiety deconvolution package called moiety_modeling is a novel method for analyzing time series SIRM MS isotopologue profiles that can involve single or multiple isotope tracers[69]. This package integrates facilities for moiety (i.e. biochemical functional group) model and data representation, model (parameter) optimization, analysis of optimization results, and model selection under a single moiety modeling framework. A typical data analysis workflow for this moiety modeling framework is shown in Figure 5.1. Moiety modeling deconvolutes isotopologue intensity data of a metabolite into pseudo-isotopomers based on a given moiety description of the metabolite. Moiety modeling is an early step in certain metabolic flux analysis approaches that can allow the comparison of different moiety models for model selection. First, plausible and hypothetical moiety models of an interesting metabolite are provided by a user based on a relevant metabolic network. After the optimization of each moiety model during isotopologue deconvolution, the optimal model can be selected based on the optimized results of model parameters, which can be directly used for downstream metabolic flux analysis and interpretation.

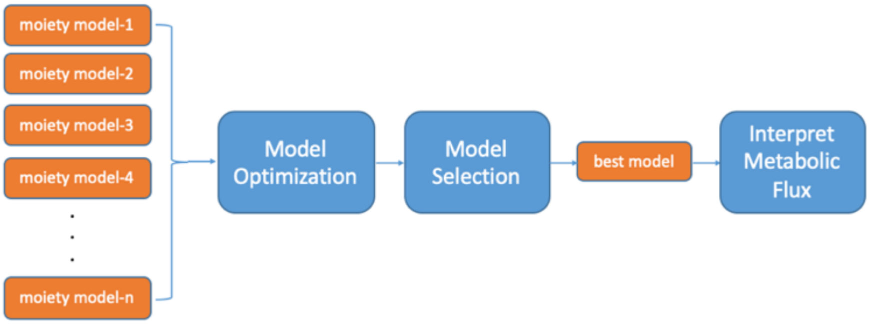


Figure 5.1. Workflow for Moiety Modeling.

In this chapter, we use this moiety modeling framework to investigate the effects of the optimization method, optimizing degree, objective function and selection criterion on model selection to identify modeling criteria that promote robust model selection. To our knowledge, this is the first attempt to investigate how all of these factors can affect model selection in metabolic modeling.

5.2 Materials and Methods

5.2.1 UDP-GlcNAc time course MS isotopologue datasets

Two UDP-GlcNAc time course MS isotopologue datasets were used to test the robustness of model selection mechanism. The first is a direct infusion Fourier transform MS (FTMS) UDPGlcNAc ^{13}C isotopologue dataset derived from LnCaP-LN3 human prostate cancer cells with [U-^{13}C]-glucose as isotope labeling source and collected on an Advion Nanomate nanoelectrospray inline connected to a Thermo 7T LTQ Fourier transform ion cyclotron resonance MS (FT-ICR-MS). This dataset includes 3 time points: 34h, 48h, and 72h[68]. The second is a liquid chromatography-MS (LC-MS) UDP-GlcNAc ^{13}C isotopologue dataset derived from human umbilical vein endothelial cells with [U-^{13}C]-glucose as the isotope labeling source and collected on a ThermoFisher Dionex UltiMate

3000 LC System in-line connected to a ThermoFisher Q-Exactive Orbitrap MS. This dataset has 5 time points: 0h, 6h, 12h, 24h and 36h[128].

5.2.2 Objective functions

We used four distinct forms of the objective function (Table 5.1) that compares the observed isotopologues and corresponding calculated isotopologues derived from model parameters obtained from model optimization. The first is a summation of absolute differences between observed and calculated isotopologues, which is generally expected to work well with data where the dominant type of error is additive. The second is a summation of the absolute differences between the log of observed and calculated isotopologues, which is generally expected to work well with data where the dominant type of error is proportional. The third is a summation of square of differences between observed and calculated isotopologues. The fourth one tries to mimic the effect of model selection criteria.

Table 5.1. Objective functions.

Objective function	**Equation**
Absolute difference	$\Sigma\lvert I_{n,obs} - I_{n,calc}\rvert$
Absolute difference of logs	$\Sigma\lvert \log(I_{n,obs}) - \log(I_{n,calc})\rvert$
Square difference	$\Sigma(I_{n,obs} - I_{n,calc})^2$
Difference of AIC	$2k + n\ln(RSS/n)$

k is the number of parameters.
n is the number of data points.
RSS is the residual sum of squares: RSS = $\sum_{i=1}^{n}(I_{obs} - I_{calc})^2$.

5.2.3 Optimization methods

From a mathematics perspective, model optimization is actually a non-linear inverse problem. Several different optimization methods were used to solve this problem, including the SAGA-optimize method[68], and three other optimization methods

(‘TNC’[130], ‘SLSQP’[131], and ‘L-BFGS-B’[132]) available in the scipy.optimize Python module. The SAGA-optimize is a combination of simulated annealing (SA) and genetic algorithm (GA) that gains advantages of both SA and GA, making it able to produce better quality results in small amount of time. The ‘TNC’ method is designed for optimizing non-linear functions with large numbers of independent variables[130]. The SLSQP method uses Sequential Least Squares Programming, which is an iterative method for constrained nonlinear optimization[131]. ‘L-BFGS-B’ is a limited-memory algorithm for solving large nonlinear optimization problems subject to simple bounds on the variables[132].

5.2.4 Model Selection Estimators

We used three different quality estimators (Table 5.2) in model selection: the Akaike Information Criterion (AIC)[133], the sample size corrected Akaike Information Criterion (AICc)[136], and the Bayesian Information Criterion (BIC)[134]. The Akaike information criterion (AIC) is biased to select models with more parameters when the sample size is small, which can lead to overfitting[133]. The sample size corrected AIC (AICc) was developed to handle this bias and prevent overfitting[136]. The Bayesian information criterion (BIC) is another criterion commonly used in model selection[134].

Table 5.2. Model selection estimators.

Selection Criterion	**Equation**
Akaike Information Criterion (AIC)	$2\mathrm{k} + \mathrm{n}\ln(\mathrm{RSS}/\mathrm{n})$
Sample size corrected AIC (AICc)	$\mathrm{AIC} + (2k^2 + 2k)/(n - k - 1)$
Bayesian Information Criterion (BIC)	$\mathrm{n}\ln(\mathrm{RSS}/\mathrm{n}) + \mathrm{k}\ln(\mathrm{n})$

k is the number of parameters.
n is the number of data points.
RSS is the residual sum of squares: RSS = $\sum_{i=1}^{n}(I_{obs} - I_{calc})^2$.

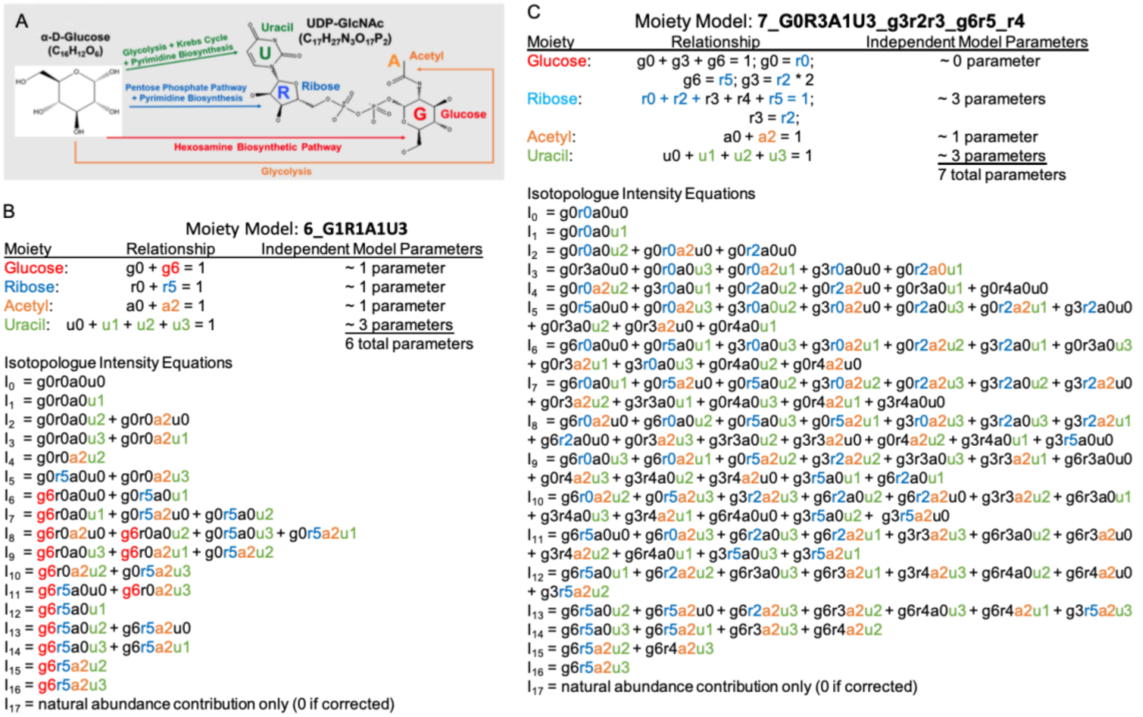
5.3 Results

5.3.1 UDP-GlcNAc moiety model construction.

UDP-GlcNAc can be divided into four distinct moieties: glucose, ribose, acetyl, and uracil, in which isotopes incorporate through a metabolic network from an isotope labeling source. The expected (expert-derived) moiety model of ^{13}C isotope incorporation from ^{13}C-labeled glucose to UDP-GlcNAc (see Figure 5.2 B) is built based on well-studied human central metabolism pathways that converge in UDP-GlcNAc biosynthesis, which is corroborated with NMR data[68]. This expert-derived model is labeled as 6_G1R1A1U3, representing six optimizable parameters, one for the glucose moiety (G1), one for the ribose moiety (R1), one for the acetyl moiety (A1), and 3 for the uracil moiety (U3), for each moiety state equation representing the fractional ^{13}C incorporation for each moiety. For example, the g6 state represents the incorporation of $^{13}C_6$ into the glucose moiety, whereas the g0 state represents no incorporation of ^{13}C. Since both g0 and g6 must sum to 1, there is only one parameter that needs to be optimized for this moiety state equation. The set of isotopologue intensity equations are derived using the moiety model parameters and Equation 1, as illustrated for the expert-derived model in Figure 5.2 B. Figure 5.2 C shows an alternative hypothetical moiety model 7_G0R3A1U3_g3R2R3_g6r5_r4 along with the isotopologue intensity equations generated from the model.

$$I_{x,calc} = \sum_{ic_a \in IC_x} ic_a \;\; ; IC_x = \{ic_v | isotope\_content(ic_v) = x\} \; ; ic_v = \prod_j moiety\_state_{j,v_j} \qquad (1)$$

We also manually crafted 40 hypothetical moiety models to capture isotope flow from [U-^{13}C]-glucose into each moiety. This set of models provides a mechanism for testing how robustly the expert-derived model can be selected from all the other models.


A
α-D-Glucose
($C_{16}H_{12}O_6$)
Glycolysis + Krebs Cycle
+ Pyrimidine Biosynthesis
Pentose Phosphate Pathway
+ Pyrimidine Biosynthesis
Uracil
UDP-GlcNAc
($C_{17}H_{27}N_3O_{17}P_2$)
U
R
Ribose
A
Acetyl
G
Glucose
HO
OH
Hexosamine Biosynthetic Pathway
Glycolysis

B

Moiety Model: **6_G1R1A1U3**

Moiety	Relationship	Independent Model Parameters
Glucose:	g0 + g6 = 1	~ 1 parameter
Ribose:	r0 + r5 = 1	~ 1 parameter
Acetyl:	a0 + a2 = 1	~ 1 parameter
Uracil:	u0 + u1 + u2 + u3 = 1	~ 3 parameters
		6 total parameters

Isotopologue Intensity Equations
I_0 = g0r0a0u0
I_1 = g0r0a0u1
I_2 = g0r0a0u2 + g0r0a2u0
I_3 = g0r0a0u3 + g0r0a2u1
I_4 = g0r0a2u2
I_5 = g0r5a0u0 + g0r0a2u3
I_6 = g6r0a0u0 + g0r5a0u1
I_7 = g6r0a0u1 + g0r5a2u0 + g0r5a0u2
I_8 = g6r0a2u0 + g6r0a0u2 + g0r5a0u3 + g0r5a2u1
I_9 = g6r0a0u3 + g6r0a2u1 + g0r5a2u2
I_{10} = g6r0a2u2 + g0r5a2u3
I_{11} = g6r5a0u0 + g6r0a2u3
I_{12} = g6r5a0u1
I_{13} = g6r5a0u2 + g6r5a2u0
I_{14} = g6r5a0u3 + g6r5a2u1
I_{15} = g6r5a2u2
I_{16} = g6r5a2u3
I_{17} = natural abundance contribution only (0 if corrected)

C

Moiety Model: **7_G0R3A1U3_g3r2r3_g6r5_r4**

Moiety	Relationship	Independent Model Parameters

Glucose:	g0 + g3 + g6 = 1; g0 = r0;
	g6 = r5; g3 = r2 * 2
		~ 0 parameter

Ribose:	r0 + r2 + r3 + r4 + r5 = 1;
	r3 = r2;
		~ 3 parameters

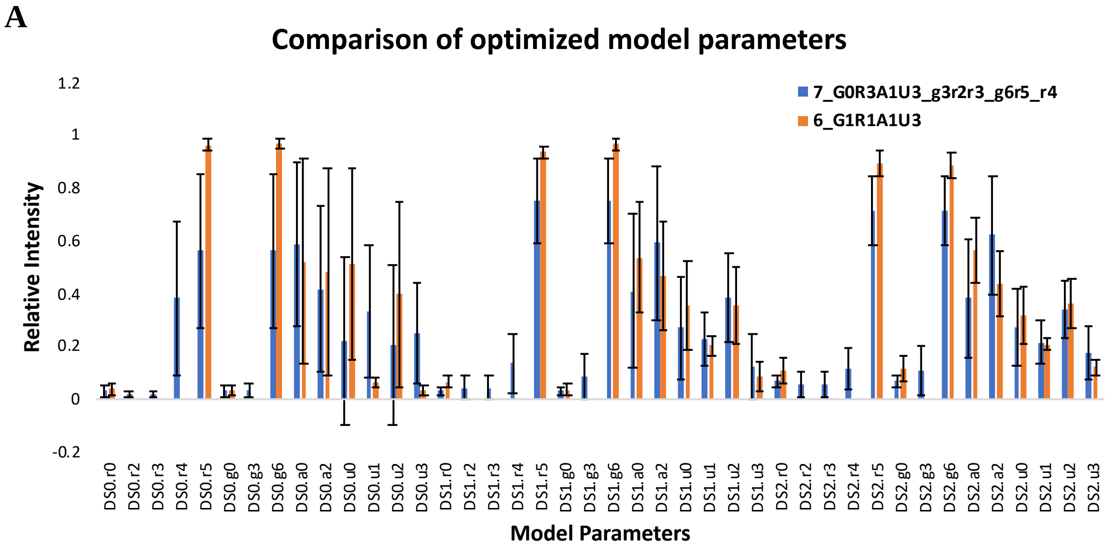
Acetyl:	a0 + a2 = 1	~ 1 parameter
Uracil:	u0 + u1 + u2 + u3 = 1	~ 3 parameters
		7 total parameters

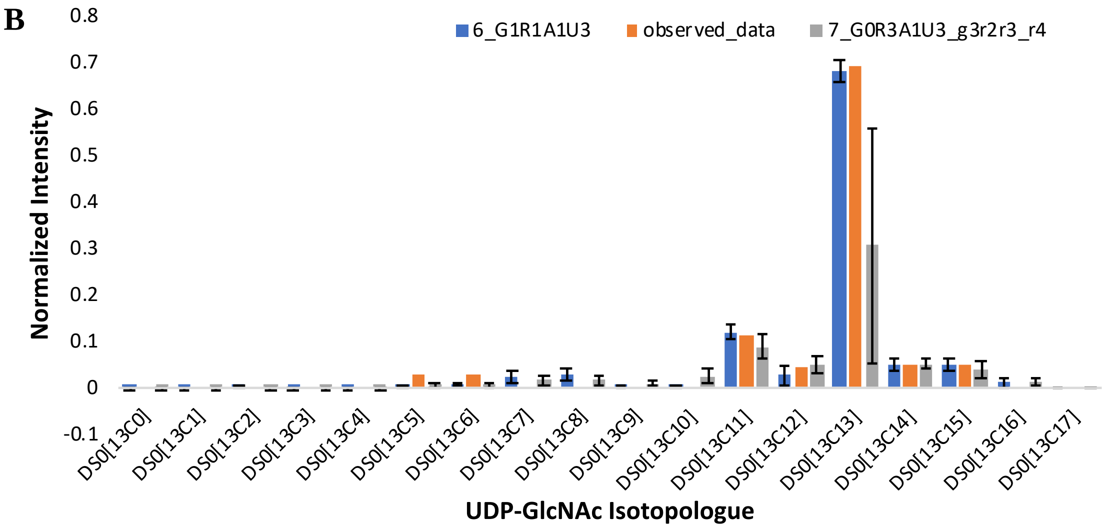
Isotopologue Intensity Equations
I_0 = g0r0a0u0
I_1 = g0r0a0u1
I_2 = g0r0a0u2 + g0r0a2u0 + g0r2a0u0
I_3 = g0r3a0u0 + g0r0a0u3 + g0r0a2u1 + g3r0a0u0 + g0r2a0u1
I_4 = g0r0a2u2 + g3r0a0u1 + g0r2a0u2 + g0r2a2u0 + g0r3a0u1 + g0r4a0u0
I_5 = g0r5a0u0 + g0r0a2u3 + g3r0a0u2 + g3r0a2u0 + g0r2a0u3 + g0r2a2u1 + g3r2a0u0
+ g0r3a0u2 + g0r3a2u0 + g0r4a0u1
I_6 = g6r0a0u0 + g0r5a0u1 + g3r0a0u3 + g3r0a2u1 + g0r2a2u2 + g3r2a0u1 + g0r3a0u3
+ g0r3a2u1 + g3r0a0u3 + g0r4a0u2 + g0r4a2u0
I_7 = g6r0a0u1 + g0r5a2u0 + g0r5a0u2 + g3r0a2u2 + g0r2a2u3 + g3r2a0u2 + g3r2a2u0
+ g0r3a2u2 + g3r3a0u1 + g0r4a0u3 + g0r4a2u1 + g3r4a0u0
I_8 = g6r0a2u0 + g6r0a0u2 + g0r5a0u3 + g0r5a2u1 + g3r0a2u3 + g3r2a0u3 + g3r2a2u1
+ g6r2a0u0 + g0r3a2u3 + g3r3a0u2 + g3r3a2u0 + g0r4a2u2 + g3r4a0u1 + g3r5a0u0
I_9 = g6r0a0u3 + g6r0a2u1 + g0r5a2u2 + g3r2a2u2 + g3r3a0u3 + g3r3a2u1 + g6r3a0u0
+ g0r4a2u3 + g3r4a0u2 + g3r4a2u0 + g3r5a0u1 + g6r2a0u1
I_{10} = g6r0a2u2 + g0r5a2u3 + g3r2a2u3 + g6r2a0u2 + g6r2a2u0 + g3r3a2u2 + g6r3a0u1
+ g3r4a0u3 + g3r4a2u1 + g6r4a0u0 + g3r5a0u2 + g3r5a2u0
I_{11} = g6r5a0u0 + g6r0a2u3 + g6r2a0u3 + g6r2a2u1 + g3r3a2u3 + g6r3a0u2 + g6r3a2u0
+ g3r4a2u2 + g6r4a0u1 + g3r5a0u3 + g3r5a2u1
I_{12} = g6r5a0u1 + g6r2a2u2 + g6r3a0u3 + g6r3a2u1 + g3r4a2u3 + g6r4a0u2 + g6r4a2u0
+ g3r5a2u2
I_{13} = g6r5a0u2 + g6r5a2u0 + g6r2a2u3 + g6r3a2u2 + g6r4a0u3 + g6r4a2u1 + g3r5a2u3
I_{14} = g6r5a0u3 + g6r5a2u1 + g6r3a2u3 + g6r4a2u2
I_{15} = g6r5a2u2 + g6r4a2u3
I_{16} = g6r5a2u3
I_{17} = natural abundance contribution only (0 if corrected)

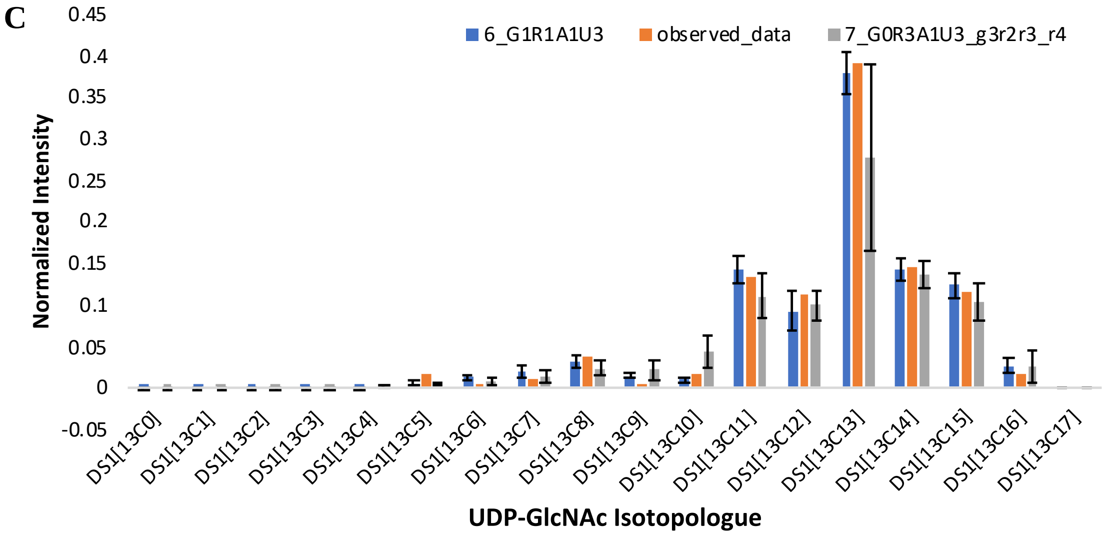
Figure 5.2. Example complex metabolite UDP-GlcNAc and associated moiety models. A) Major human metabolic pathways from glucose to the four moieties of UDP-GlcNAc. B) The expert-derived moiety model based on known human central metabolism pathways with corroborating NMR data. C) An alternative hypothetical moiety model with simple perturbations of the original expert-derived model.

5.3.2 A simple comparison of two moiety models

Model optimization aims to minimize an objective function that compares calculated isotopologues based on moiety state parameters from the model to the directly observed, experimentally-derived isotopologues. Figure 5.3 A shows the comparison of optimized model parameters between the expert-derived moiety model (6_G1R1A1U3) and the hypothetical moiety model (7_G0R3A1U3_g3r2r3_r4) for three time points of isotopologue intensity data, i.e. three sets of isotopologue intensities. In these model optimizations, the SAGA-optimize method and absolute difference objective function were used and DS0, DS1, and DS2 correspond to the 34h, 48h, and 72h time points in the FT-ICR-MS UDP-GlcNAc dataset. We can easily tell that the relative intensity of the corresponding model parameters between these two models are quite different, suggesting that the moiety-specific ^{13}C isotopic incorporation derived from the same MS isotopologue profile varies from one model to another. Furthermore, experiment-derived and model parameter-calculated isotopologue profiles are shown in Figure 5.3 B, illustrating how much better the expert-derived model vs an inaccurate model is able to reflect the observed data.




B
6_G1R1A1U3
observed_data
7_G0R3A1U3_g3r2r3_r4
Normalized Intensity
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
DS0[13C0]
DS0[13C1]
DS0[13C2]
DS0[13C3]
DS0[13C4]
DS0[13C5]
DS0[13C6]
DS0[13C7]
DS0[13C8]
DS0[13C9]
DS0[13C10]
DS0[13C11]
DS0[13C12]
DS0[13C13]
DS0[13C14]
DS0[13C15]
DS0[13C16]
DS0[13C17]
UDP-GlcNAc Isotopologue


C
6_G1R1A1U3
observed_data
7_G0R3A1U3_g3r2r3_r4
Normalized Intensity
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
DS1[13C0]
DS1[13C1]
DS1[13C2]
DS1[13C3]
DS1[13C4]
DS1[13C5]
DS1[13C6]
DS1[13C7]
DS1[13C8]
DS1[13C9]
DS1[13C10]
DS1[13C11]
DS1[13C12]
DS1[13C13]
DS1[13C14]
DS1[13C15]
DS1[13C16]
DS1[13C17]
UDP-GlcNAc Isotopologue

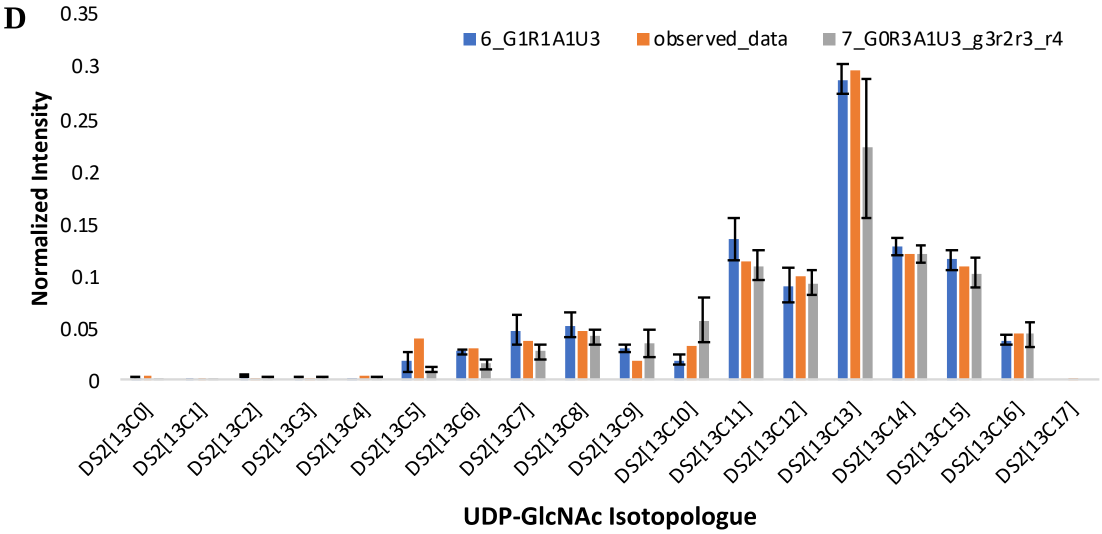


Figure 5.3. Optimized results for 6_G1R1A1U3 and 7_G0R3A1U3_g3r2r3_r4 models. Each model optimization was conducted 100 times. A) Comparison of mean of optimized model parameters with standard deviation. B-D) Reconstruction of the isotopologue distribution of UDP-GlcNAc from model parameters. Observed isotopologue data was compared with the mean of calculated isotoplogue data with standard deviation from the optimized parameters for each model.

5.3.3 Effects of optimization method on model selection

The first question we were interested in was whether the optimization method could affect the model selection results. As in the previous analysis, we used 3 time points from the FT-ICR-MS dataset, the AICc criterion, and an absolute difference objective function in the initial trial. The optimization for each model was conducted 100 times, and we used the average of the 100 optimization results in the analysis (see Table 5.3). Most optimization methods can select the expert-derived model except for ‘SLSQP’. What interested us most was that the ‘SLSQP’ method failed in model selection with the lowest loss value (value returned from objective function) and is generally considered to be the fastest converging of the optimization methods we tested.

We repeated the experiment with the ‘SLSQP’ method 10 times and found that model selection fails when the loss value approaches 0.3, suggesting strong instability of model selection at a critical point. Model optimization aims to minimize the objective function, which is actually a non-linear inverse problem, and one inherited issue in solving a non-linear inverse problem is overfitting (i.e. fitting to error in the data). Therefore, we developed the hypothesis that over-optimization of model parameters can lead to failure in model selection.

Table 5.3. Comparison of optimization methods in model selection.

Optimization method	**Loss value**	**AICc**	**Selected model**
SAGA	0.469	-401.760	Expert-derived model
SLSQP	**0.320**	**-408.341**	**7_G2R1A1U3_g5**
L-BFGS-B	0.763	-342.164	Expert-derived model
TNC	0.870	-327.344	Expert-derived model

Dataset: FT-ICR-MS (combined); Selection criterion: AICc; Objective function: Absolute difference.

5.3.4 Over-optimization leads to failure in model selection.

To test the above hypothesis, we first tried to increase the stop criterion of ‘SLSQP’ method to control over-optimization. The results are shown in Table 5.4. When optimization stops earlier, the expert-derived model can be selected, which supports our hypothesis.

The SAGA-optimize method is more flexible in controlling the degree of optimization simply by adjusting the number of optimization steps. The more steps, the lower the average loss value reached by the optimization. Next, we performed a set of experiments using the SAGA-optimize method with increasing number of optimization steps to further validate the hypothesis. The results are summarized in the Table 5.5. We can see that the loss value decreases as optimization step increases. When the loss value reaches a certain critical point, the expert-derived model cannot be selected, further supporting the hypothesis that over-optimization can lead to failure in model selection. Furthermore, the selected model can change with increasing degrees of over-optimization.

Based on the above results, we conclude that it is not the optimization method but the degree of optimization that affects model selection, which is explained by overfitting to error in the data when solving a non-linear inverse problem. When optimization reaches a certain critical point, successful model selection cannot be guaranteed. Therefore, proper control of the degree of optimization is of great importance in model selection.

Table 5.4. Over optimization experiments with ‘SLSQP’ method.

Optimization method	**Loss value**	**AICc**	**Selected model**	**Stop criterion**
SLSQP	0.320	-408.341	7_G2R1A1U3_g5	‘ftol': *1e-06*
SLSQP	0.514	-393.934	Expert-derived model	‘ftol': *1e-05*

Dataset: FT-ICR-MS (combined); Selection criterion: AICc; Objective function: Absolute difference.

Table 5.5. Over optimization experiments with SAGA-optimize method.

Optimization steps	**Loss value**	**AICc**	**Selected model**
500	2.070	-219.488	Expert-derived model
1000	1.754	-235.728	Expert-derived model
2000	1.377	-260.654	Expert-derived model
5000	0.941	-305.651	Expert-derived model
10000	0.664	-375.192	Expert-derived model
25000	0.469	-401.760	Expert-derived model
50000	0.408	-414.737	Expert-derived model
75000	**0.328**	**-418.228**	**7_G2R1A1U3_g5**
100000	0.316	-424.924	7_G1R2A1U3_r4

Dataset: FT-ICR-MS (combined); Selection criterion: AICc; Objective function: Absolute difference.

5.3.5 Effects of selection criterion on model selection

Next, we investigated whether selection criterion could affect model selection. We compared the model selection results generated by SAGA-optimize with different model selection criteria (see Table 5.6). From these results, we can see that the rank of top models is quite consistent across different selection criteria, suggesting that these model selection criteria have little effect on robust model selection, at least under this model selection context. Since our previous experiments used AICc as the selection criterion, we will stick with AICc in the following experiments.

Table 5.6. Comparison of mode rank based on different model selection criteria.

Models	AICc	rank	AIC	rank	BIC	rank
Expert-derived model	**-401.7597**	**1**	**-421.3026**	**1**	**-385.5009**	**1**
7_G1R1A2U3	-384.3075	2	-413.1825	2	-371.4139	2
7_G2R1A1U3_g5	-381.2868	3	-410.1618	3	-368.3932	3
7_G1R2A1U3_r3	-379.2657	4	-408.1407	4	-366.3720	4
7_G1R2A1U3_r4	-378.8969	5	-407.7719	5	-366.0033	5
7_G2R1A1U3_g4	-375.9538	6	-404.8288	6	-363.0601	6
6_G1R1A1U3_g5	-374.9694	7	-394.5122	10	-358.7105	8
6_G1R1A1U3_r4	-374.1820	8	-393.7249	11	-357.9231	9
7_G1R1A1U4	-373.4563	9	-402.3313	7	-360.5626	7
6_G1R1A1U3_u4	-370.0716	10	-389.6145	13	-353.8127	11
7_G2R1A1U3_g1	-367.8353	11	-396.7103	8	-354.9416	10
7_G2R1A1U3_g2	-360.1668	12	-389.0418	14	-347.2732	13
7_G1R1A1U3C1	-360.0296	13	-388.9046	15	-347.1360	14
7_G1R2A1U3_r1	-354.8814	14	-383.7564	16	-341.9878	16
8_G1R2A2U3_r3	-354.4480	15	-395.8273	9	-348.0917	12
8_G2R1A2U3_g4	-351.9886	16	-393.3679	12	-345.6323	15
6_G0R2A1U3_g3r2r3_g6r5	-345.1277	17	-364.6706	21	-328.8689	17
8_G2R1A2U3_g1	-334.2882	18	-375.6675	17	-327.9319	18
7_G2R1A1U3_g3	-332.9148	19	-361.7898	22	-320.0211	19
7_G1R2A1U3_r2	-332.3262	20	-361.2012	23	-319.4326	21
8_G1R2A2U3_r1	-325.9344	21	-367.3137	18	-319.5781	20
8_G1R1A2U3C1	-324.5196	22	-365.8989	19	-318.1633	22
8_G2R1A2U3_g5	-324.5004	23	-365.8797	20	-318.1441	23
7_G0R2A2U3_g3r2r3_g6r5	-324.0749	24	-352.9499	26	-311.1813	25
7_G1R2A1U3_g3r2r3	-324.0721	25	-352.9471	27	-311.1784	26
8_G2R1A2U3_g2	-318.5771	26	-359.9564	24	-312.2208	24
6_G1R1A1U3_a1	-318.2498	27	-337.7927	31	-301.9910	28
8_G1R2A2U3_r4	-317.3169	28	-358.6962	25	-310.9606	27
8_G2R1A2U3_g3	-302.7897	29	-344.1690	28	-296.4334	29
8_G1R2A2U3_g3r2r3_g6r5_g5	-297.7429	30	-339.1222	30	-291.3866	30
8_G1R2A2U3_r2r3	-295.0078	31	-336.3871	32	-288.6515	31

8_G1R2A2U3_r2	-294.7900	32	-336.1693	33	-288.4337	32
8_G1R2A2U3_g3r2r3	-292.7867	33	-334.1660	34	-286.4304	33
9_G2R2A2U3_r2r3_g6r5_g3_g5	-281.8920	34	-340.0458	29	-286.3433	34
7_G0R3A1U3_g3r2r3_g6r5_g5r4	-279.0349	35	-307.9099	37	-266.1412	36
9_G2R2A2U3_r2r3_g4	-273.5807	36	-331.7345	35	-278.0320	35
9_G2R2A2U3_r2r3_g5	-254.4087	37	-312.5625	36	-258.8599	37
9_G2R2A2U3_r2r3_g3	-248.2277	38	-306.3815	38	-252.6789	38
9_G2R2A2U3_r2r3_g2	-242.9984	39	-301.1522	39	-247.4497	39
9_G2R2A2U3_r2r3_g1	-242.4110	40	-300.5648	40	-246.8623	40
7_G0R3A1U3_g3r2r3_g6r5_r4	-226.7271	41	-255.6021	41	-213.8334	41

Dataset: FT-ICR-MS (combined); Optimization method: SAGA-optimize (25000 steps); Objective function: Absolute difference.

5.3.6 Effects of selection criterion on model selection

Considering that the dominant type of error existing in metabolomics datasets may vary from dataset to dataset, different forms of objective function may affect model optimization and then influence the results of model selection. Here, we test the effects of four objective functions in the context of model selection: absolute difference, absolute difference of logs, square difference, and difference of AIC. To speed up optimization, we first split the FT-ICR-MS dataset based on time point (34h, 48h, 72h) into separate model optimizations executed on their own CPU core, and then combine the optimization results for the model selection. This functionality is provided by the moiety_modeling package. We set a series of experiments for each objective function with SAGA-optimize method. The results are shown in Table 5.7 to Table 5.10. In comparing Table 5.7 to Table 5.5, the number of optimizations per time point provides roughly the same degree of optimization as three times the number of optimization steps used on a combined optimization.

From these tables, we can see that optimization with the absolute difference of logs objective function is less likely to fail (> 250000 steps) in the model selection compared to the other three objective functions (10000 - 20000 steps). One interpretation from these results is that the FT-ICR-MS dataset is dominated by proportional error instead of additive error. However, the AICc produced with the absolute difference of logs is significantly higher (less negative) than that produced by the other objective functions. Therefore, this objective function may simply be hindering efficient optimization, especially if the dataset is dominated by an error structure that is not as compatible with this objective function. From this alternative viewpoint, additive error may actually dominate this dataset. We used a graphical method to visualize errors in both FT-ICR-MS and LC-MS datasets (Figure 5.4 and 5.5). For the plots of FT-ICR-MS datasets, we used another dataset generated from the same procedure, which included two replicates at 0, 3h, 6h, 11h, 24h, 34h, and 48h time points. For two replicates with proportional error, a scatter plot of each replicate against the other will show an increasing spread of values with increasing signal, and the log-transformed data will collapse into a line. Plot of two replicates with additive error can be viewed as uniformly deviated from the line of identity, but once log-transformed will show an increasing spread of values with decreasing signal. The original plots of raw data indicate existence of proportional error in both FT-ICR-MS and LC-MS datasets (Figure 5.4 A and 5.5 A). However, the original plots of normalized data almost collapsed to a straight line (Figure 5.4 C and 5.5 C), suggesting that normalization somehow removes the proportional error in the raw data. In addition, from the log-transformed plots, we can see that additive error does not exist in the normalized FT-ICR-MS datasets (Figure 5.4 D), but does exist in the normalized LC-MS datasets (Figure 5.5 D). The replicate plots of all time points (Figure 5.6 & 5.7) show similar tendency with selected optimized datasets. Based on the above results, the absolute difference of logs objective function can hinder efficient optimization in FT-ICR-MS datasets. We also compared four objective functions in the context of model selection with LC-MS datasets (Table 5.11-5.14). From these tables, we can see that model selection fails earlier with absolute difference of logs objective function compared to other objective functions, also suggesting that additive error may dominate in the normalized LC-MS datasets. Based on the above results, the objective function clearly affects model selection and the selection of certain objective functions for model optimization is able to increase resistance to failure in model selection caused by over-optimization; however, this is likely due to less efficient model optimization caused by the selection of an objective function not appropriate for the type of error in the data.

Table 5.7. Model selection test with absolute difference objective function.

Optimization steps	Loss value	AICc	Selected model
500	1.045	-293.540	Expert-derived model
1000	0.819	-330.411	Expert-derived model
2000	0.651	-361.038	Expert-derived model
5000	0.459	-408.167	Expert-derived model
10000	0.392	-422.516	Expert-derived model
15000	0.359	-431.276	Expert-derived model
20000	**0.290**	**-434.468**	**7_G1R1A2U3**
25000	0.285	-436.909	7_G1R1A2U3

Dataset: FT-ICR-MS (split); Selection criterion: AICc; Objective function: absolute difference.

Table 5.8. Model selection test with square difference objective function.

Optimization steps	Loss value	AICc	Selected model
500	0.085	-298.516	Expert-derived model
1000	0.047	-330.096	Expert-derived model
2000	0.023	-367.279	Expert-derived model
5000	0.011	-404.509	Expert-derived model
10000	0.007	-425.695	Expert-derived model
15000	**0.005**	**-429.869**	**7_G2R1A1U3_g5**
20000	0.005	-435.348	7_G1R2A1U3_r4

Dataset: FT-ICR-MS (split); Selection criterion: AICc; Objective function: square difference.

Table 5.9. Model selection test with absolute difference of logs objective function.

Optimization steps	**Loss value**	**AICc**	**Selected model**
500	31.647	-221.501	Expert-derived model
1000	29.628	-223.363	Expert-derived model
2000	28.164	-224.330	Expert-derived model
5000	27.096	-225.911	Expert-derived model
10000	26.631	-227.499	Expert-derived model
15000	26.469	-227.690	Expert-derived model
20000	26.398	-227.780	Expert-derived model
25000	26.271	-228.178	Expert-derived model
50000	26.126	-228.892	Expert-derived model
100000	25.949	-228.926	Expert-derived model
150000	25.865	-229.926	Expert-derived model
250000	25.777	-230.232	Expert-derived model

Dataset: FT-ICR-MS (split); Selection criterion: AICc; Objective function: absolute difference of logs.

Table 5.10. Model selection test with difference of AIC objective function.

Optimization steps	**Loss value**	**Selected model**
500	-345.559	Expert-derived model
1000	-371.852	Expert-derived model
2000	-398.570	Expert-derived model
5000	-436.582	Expert-derived model
10000	**-458.064**	**7_G1R1A2U3**
15000	-467.960	7_G2R1A1U3_g5

Dataset: FT-ICR-MS (split); Selection criterion: AICc; Objective function: difference of AIC.

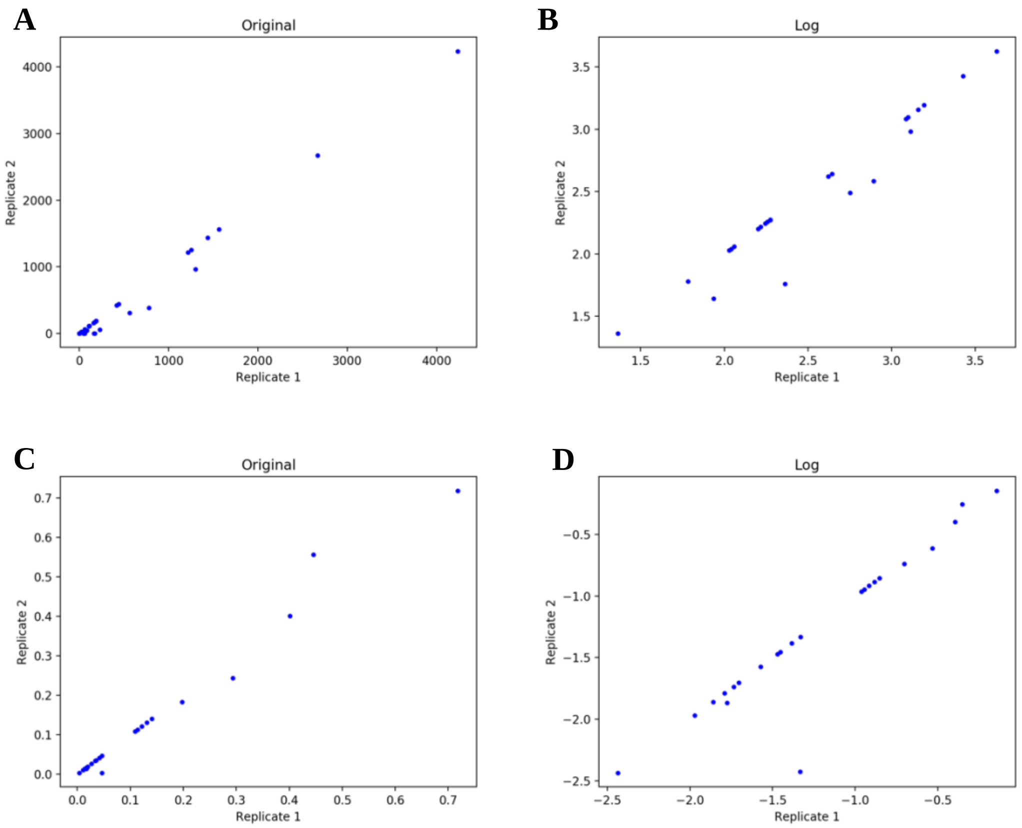


Figure 5.4. Error analysis in FT-ICR-MS datasets.

A and B are plots of raw data. C and D are plots of renormalized data after natural abundance correction. All these plots contain 3 time points (12 – 36h).

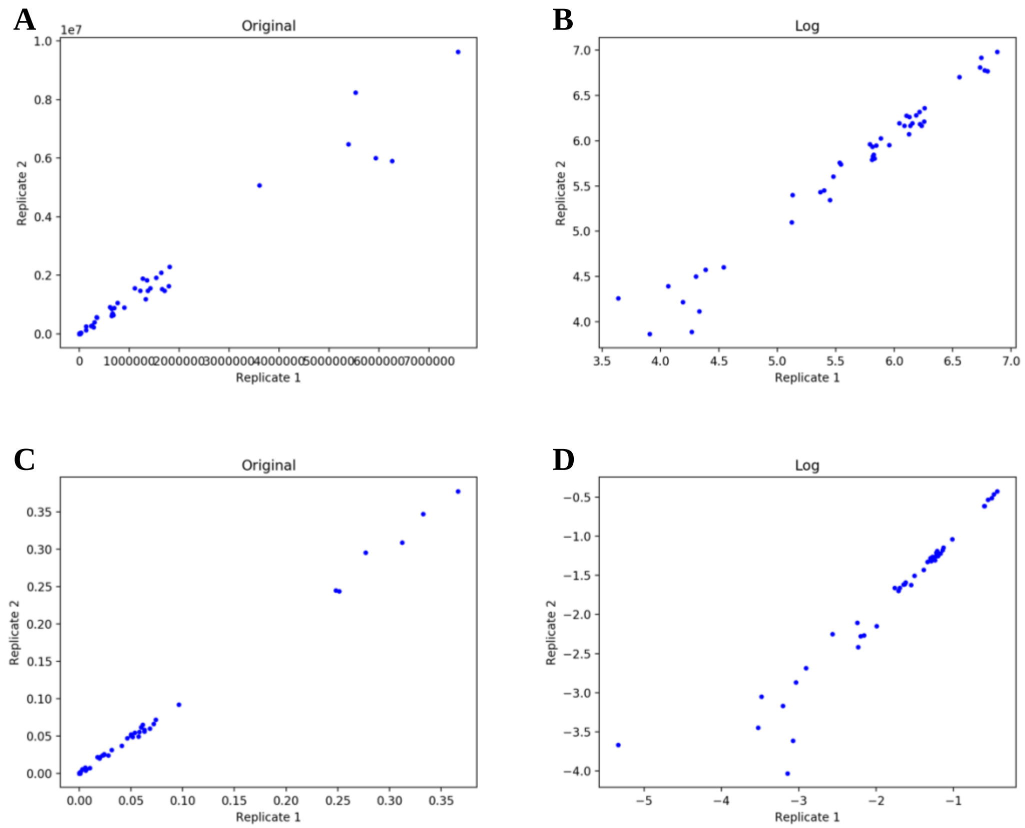


Figure 5.5. Error analysis in LC-MS datasets.

A and B are plots of raw data. C and D are plots of renormalized data after natural abundance correction. All these plots contain 3 time points (12 – 36h).

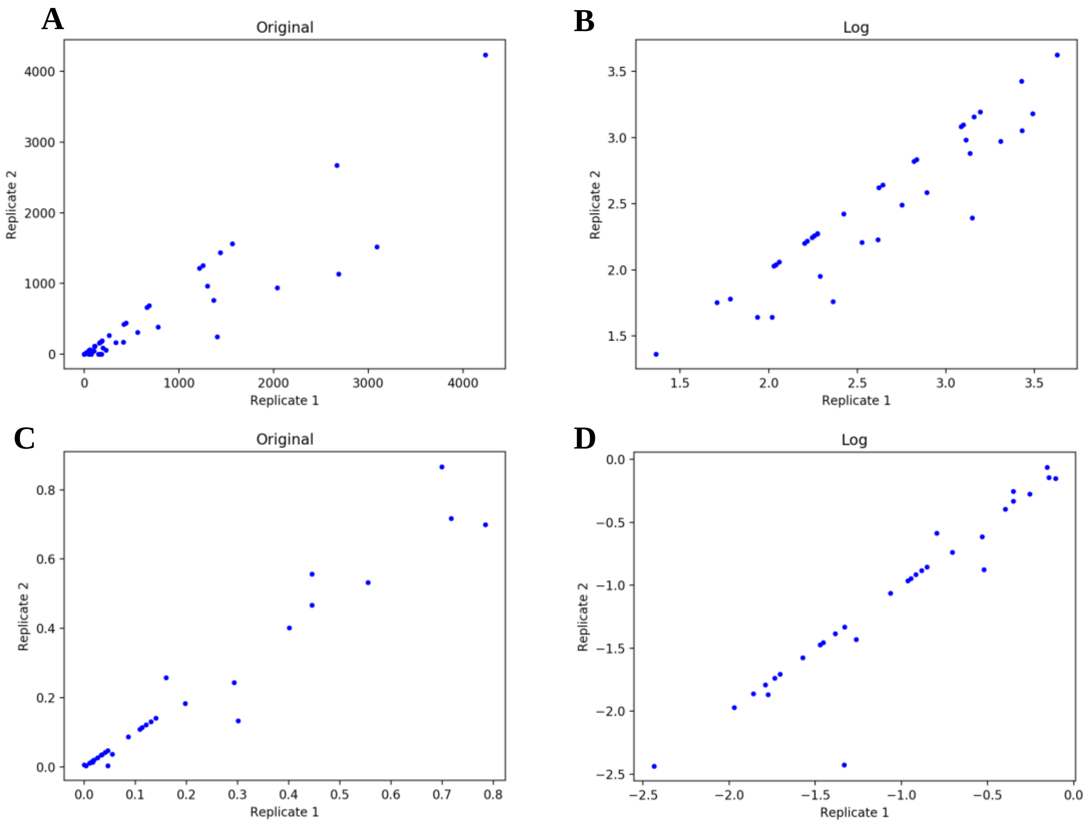


Figure 5.6. Error analysis in FT-ICR-MS datasets.

A and B are plots of raw data. C and D are plots of renormalized data after natural abundance correction. All these plots contain all time points.

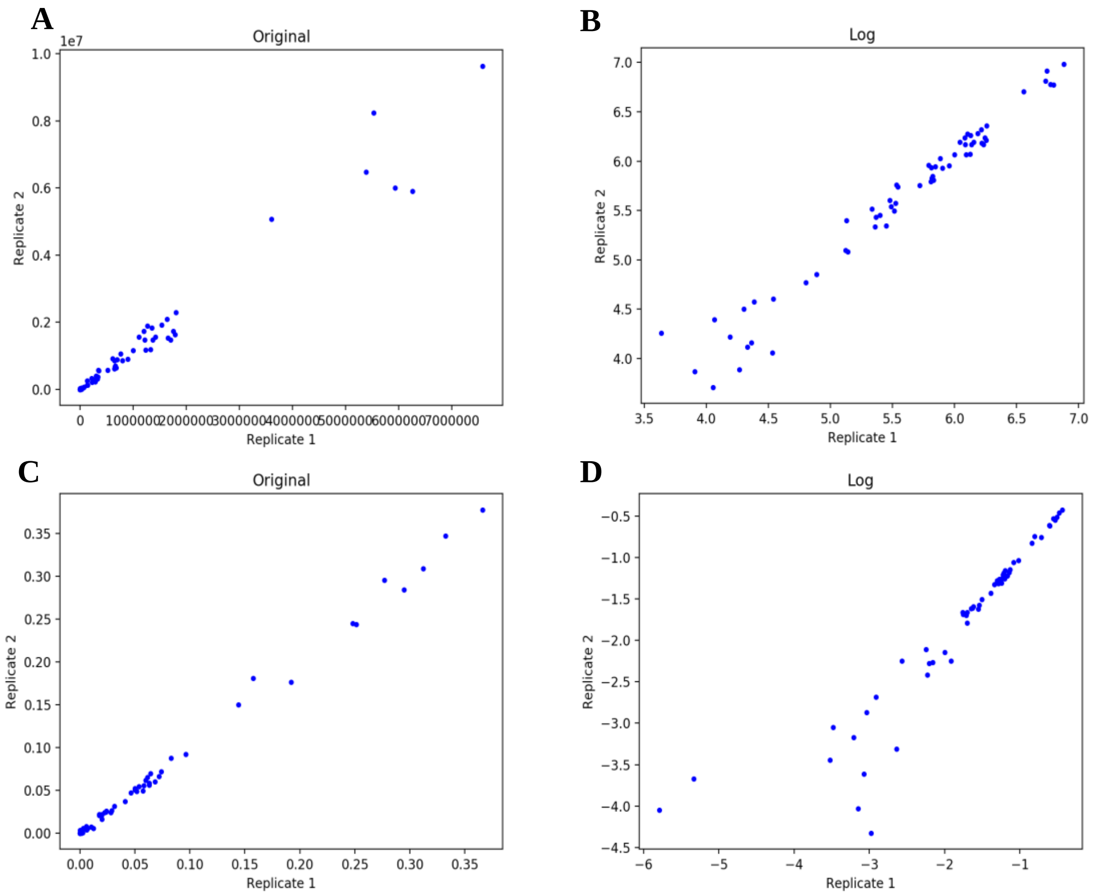


Figure 5.7. Error analysis in LC-MS datasets.

A and B are plots of raw data. C and D are plots of renormalized data after natural abundance correction. All these plots contain all time points.

Table 5.11. Model selection test with absolute difference objective function.

Optimization steps	Loss value	AICc	Selected model
500	0.840	-344.734	Expert-derived model
1000	0.682	-368.696	Expert-derived model
2000	0.580	-386.000	Expert-derived model
5000	0.492	-398.243	Expert-derived model
10000	0.447	-402.611	Expert-derived model
15000	0.430	-405.722	Expert-derived model
25000	0.458	-407.414	6_G1R1A1U3

Dataset: LC-MS (split); Selection criterion: AICc; Objective function: absolute difference.

Table 5.12. Model selection test with square difference objective function.

Optimization steps	Loss value	AICc	Selected model
500	0.031	-348.250	Expert-derived model
1000	0.021	-368.818	Expert-derived model
2000	0.015	-387.196	Expert-derived model
5000	0.011	-404.563	Expert-derived model
10000	0.010	-411.177	Expert-derived model
15000	0.010	-413.499	Expert-derived model
25000	0.009	-415.498	Expert-derived model

Dataset: LC-MS (split); Selection criterion: AICc; Objective function: square difference.

Table 5.13. Model selection test with absolute difference of logs objective function.

Optimization steps	Loss value	AICc	Selected model
500	50.595	-315.616	Expert-derived model
1000	47.213	-319.026	Expert-derived model
2000	44.137	-323.619	Expert-derived model
5000	40.811	-320.949	Expert-derived model
10000	39.474	-328.551	Expert-derived model
15000	57.856	-331.700	6_G1R1A1U3

Dataset: LC-MS (split); Selection criterion: AICc; Objective function: absolute difference of logs.

Table 5.14. Model selection test with difference of AIC objective function.

Optimization steps	Loss value	Selected model
500	-365.957	Expert-derived model
1000	-389.987	Expert-derived model
2000	-409.322	Expert-derived model
5000	-427.064	Expert-derived model
10000	-435.618	Expert-derived model
15000	-437.970	Expert-derived model
25000	-439.533	Expert-derived model

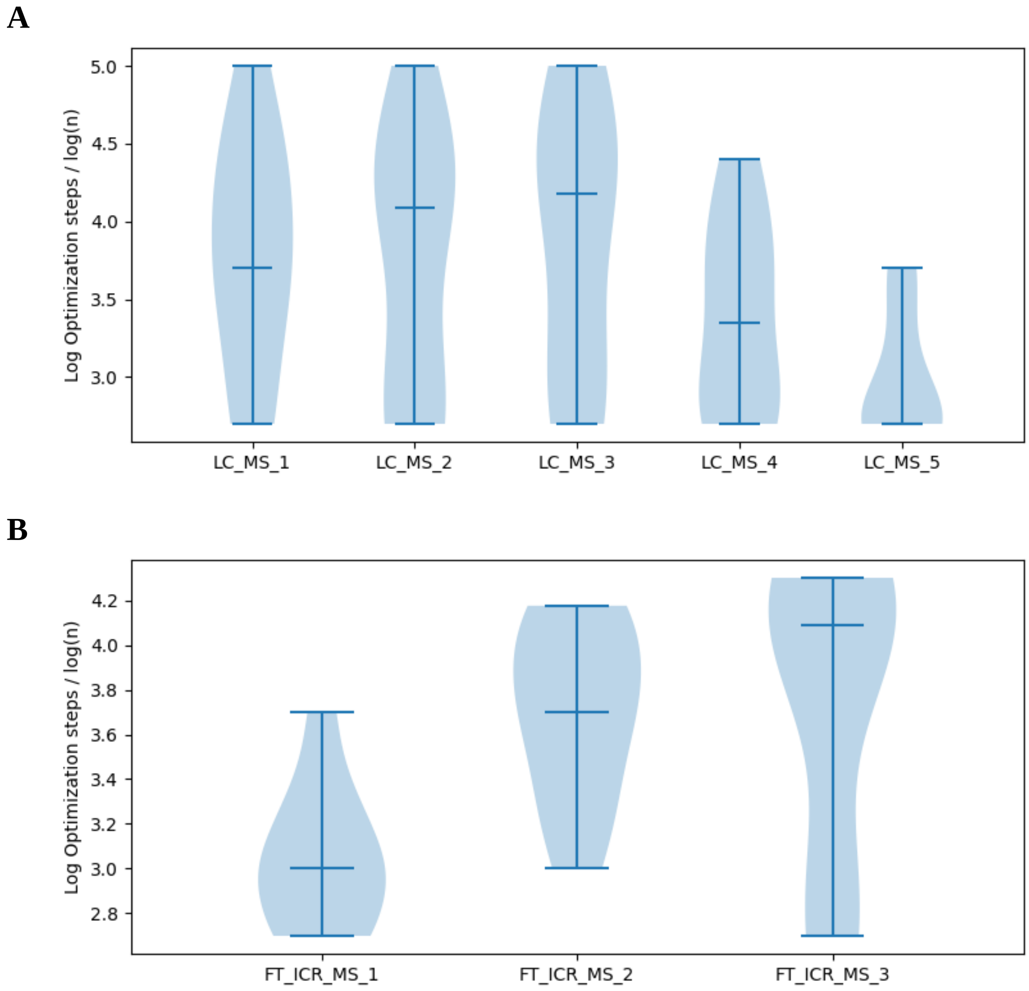
Dataset: LC-MS (split); Selection criterion: AICc; Objective function: difference of AIC.

5.3.7 Effects of information quantity on model selection

From the above experiments, we found that over-optimization is a primary cause for failure in model selection and this is affected by the objective function used. The next question is whether the quantity of information affects model selection. One basic approach is to utilize more datasets in order to overcome the effects of over-optimization. In the following experiments, we repeated single model optimization 10 times in order to pragmatically finish these computational experiments. Every experiment was conducted 10 times using the AICc criterion and the absolute difference objective function. We used the SAGA-optimize method to test where model selection starts to fail.

First, we used decreasing number of time points of the LC-MS dataset to test whether data quantity affects model selection (Figure 5.8 A, Table 5.15). However, model selection failed with few optimization steps when all five time points were included and when only one time point was included, with the most robust model selection occurring with 3 time points. Initially, these results were not expected, until we realized that the relative isotopologue intensity of the 0 and 6h time points is concentrated within the $^{13}C_0$ isotopologue with zero ^{13}C tracer. Thus, these datasets are less informative with respect to capturing the isotope flow from labeling source to each moiety in the metabolite. When the 0 and 6h time points are removed, the selection results improved significantly. Likewise, when information-rich time points are removed, the model selection robustness decreases as well. Similar results were obtained when testing the FT-ICR-MS dataset (Figure 5.8 B). Taken together, the addition of information-rich data contributes to successful model selection while the addition of information-poor data detracts from successful model selection.

To further test this concept, we investigated whether combining FT-ICR-MS (34h, 48h, 72h) and LC-MS (12h, 24h, 36h) datasets can prevent failure in model selection (Figure 5.8 C). From the comparison, we can see that combining information-rich FT-ICR-MS and LC-MS datasets is much more resistant to failure of model selection than just using the information-rich FT-ICR-MS or LC-MS dataset, strongly supporting our previous conclusions that utilizing more information-rich datasets can prevent failure in model selection. Similar results were obtained with absolute difference of logs objective function (Figure 5.9). These datasets were collected at different times, on very different mass spectrometry platforms. One used chromatographic separation while the other utilized direct infusion. However, the really surprising part is that the datasets were derived from different human cell cultures: LnCaP-LN3 human prostate cancer cells and human umbilical vein endothelial cells.



C

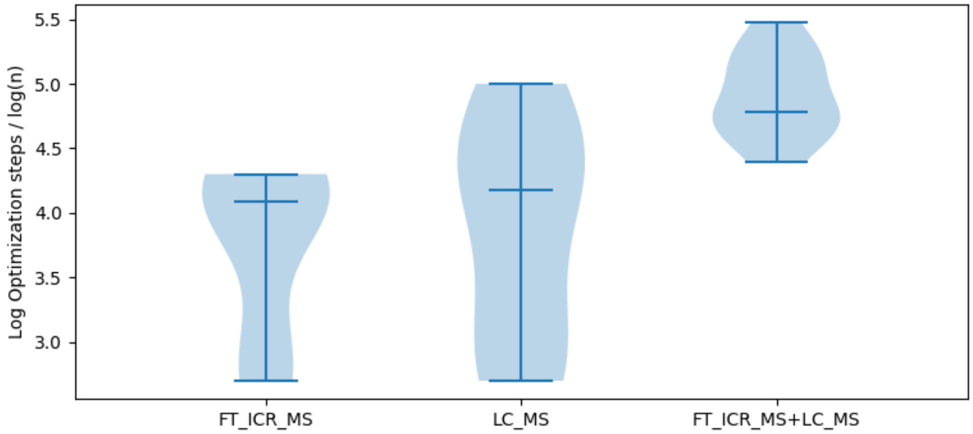


Figure 5.8. Comparison of the log optimization steps where model selection with different datasets begins to fail.

A) Test with LC-MS datasets. LC-MS_1 to LC-MS_5 represent LC-MS datasets with 36h, 24-36h, 12-36h, 6-36h and 0-36h. B) Test with FT-ICR-MS datasets. FT-ICR-MS_1 to FT-ICR-MS_3 represent FT-ICR-MS datasets with 48h, 48-72h and 34-72h. C) Test with combination of LC-MS (12h, 24h, 36h) and FT-ICR-MS (34h, 48h, 72h) datasets. The median values are indicated in the plots.

Table 5.15. Inclusion of less informative dataset can lead to failure in model selection.

Optimization steps	**Selected model**				
	5 time points (0-36h)	**4 time points (6-36h)**	**3 time points (12-36h)**	**2 time points (24-36h)**	**1 time point (36h)**
500	ED model	ED model	ED model	ED model	ED model
1000	ED model	ED model	ED model	ED model	ED model
2000	ED model	ED model	ED model	ED model	ED model
5000	6_G1R1A1U3_u4	6_G1R1A1U3_u4	ED model	ED model	7_G1R2A1U3_r1
10000	6_G1R1A1U3_u4	6_G1R1A1U3_u4	ED model	7_G1R2A1U3_r1	7_G1R2A1U3_r1
15000	6_G1R1A1U3_u4	6_G1R1A1U3_u4	6_G1R1A1U3_u4	7_G1R2A1U3_r1	7_G1R2A1U3_r1
25000	6_G1R1A1U3_u4	6_G1R1A1U3_u4	6_G1R1A1U3_u4	7_G1R2A1U3_r1	7_G1R2A1U3_r1

Dataset: LC-MS (split); Objective function: log difference; Selection criterion: AICc; Optimization method: SAGA-optimize.

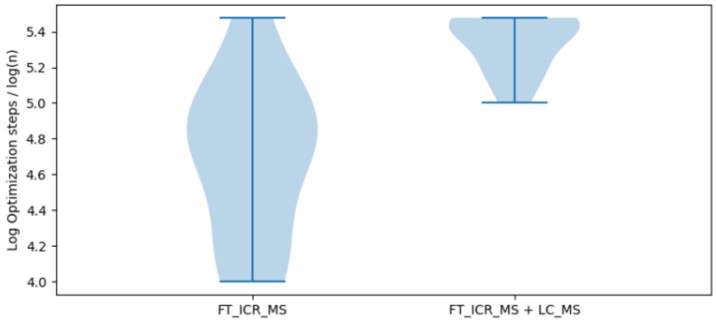


Figure 5.9. Comparison of the log of optimization steps where model selection with different datasets begins to fail with absolute difference of logs objective function.

5.4 Discussion

Here, we discussed the importance of model selection in isotopic flux analysis as a proxy for metabolic flux analysis and factors that affect robust model selection. We found that it is not the optimization method per se, but the degree of optimization that influences model selection, due to the effects of over-optimization, i.e. fitting of model parameters based on the error in the data. Overfitting is a known problem typically due to the ill-conditioning of the nonlinear inverse problem that is partially ill-posed. Moreover, the objective function in model optimization is also of great importance in model selection. Proper selection of an objective function can help increase resistance to failure in model selection. This may mean that different objective functions should be used for model selection versus parameter optimization for flux interpretation. Most SIRM experimental datasets have few collected replicates and time points due to the cost and effort required to acquire these datasets. The lack of replicates makes it impractical to directly estimate error in many of these datasets. Also, the presence of different types of systematic error like ion suppression can limit the overall effectiveness of replicate-based error analysis. With our moiety modeling framework, we are able to conduct a set of gradient experiments with varying amounts of optimization (i.e, number of optimization steps) using the SAGA-optimize method to estimate the failure point in model selection caused by overfitting. Furthermore, we found that incorporation of less informative datasets can hinder successful model selection since they cannot properly represent the incorporation of isotopes simulated by moiety models, which can lead to increased errors in model selection. On the other hand, combination of informative datasets (i.e. time points with significant isotope incorporation) can help control failure in model selection, which suggests that informative datasets in public metabolomics repositories can be combined to facilitate robust model selection. Moreover, these datasets do not need to come from identical biological systems, just biological systems that utilize the same part of metabolism being measured and modeled. The implication is that SIRM datasets in public repositories of reasonable quality can be combined with newly acquired datasets to improve model selection. Furthermore, curation efforts of public metabolomics repositories to maintain high data quality and provide metrics of measurement error could have a huge impact on future metabolic modeling efforts.

CHAPTER 6. CONCLUSION

In this project, we worked on the development of computational methods and tools for analyzing and interpreting metabolomics data derived from SIRM experiments. Despite SIRM experiments which can generate enriched metabolic features containing isotopic flow through cellular metabolism, the lack of corresponding analytic tools and methods greatly hinders the characterization of metabolic phenotypes and their downstream applications. Both detailed metabolic modeling and quantitative analysis methods are required for better interpretation of the complex metabolomics data. Currently, there is no relatively comprehensive metabolic network at an atom-resolved level that can be used for deriving context-specific metabolic models for a given metabolic profile. In addition, most existing software packages conduct metabolic flux analysis based on a predefined metabolic model, where novel metabolic mechanism can hardly be detected. Besides, a well-defined metabolic model is hard to achieve for complex biological system given the limitations in our knowledge. Here, we developed a set of methods and tools to help address those problems.

In Chapter 2, we developed a graph coloring method that creates unique identifiers for unique atom as well as same identifier for symmetric atoms in a compound. Therefore, additional cross-reaction atom mappings caused by symmetric atoms can be captured, contributing to the construction of a more complete atom-resolved metabolic network. In addition, the ordered compound coloring identifiers derived from the corresponding atom coloring identifiers facilitate compound harmonization across metabolic databases, which is an essential first step in cross-database network integration. While harmonizing compounds between KEGG and MetaCyc, the graph coloring method also detected various issues and errors in both databases, suggesting that this method can also be used for curating current metabolic databases. More importantly, the graph coloring method and compound harmonization approach can be used to integrate any metabolic database that provides a molfile representation of compounds, which will greatly facilitate future expansion.

In addition to harmonizing compounds with specified atomic compositions, we further integrated compounds with R group(s) via an optimized common subgraph isomorphism algorithm in Chapter 3. We also addressed the issue of inconsistent atomistic characteristics across databases by defining a set of relationships between compounds. Meanwhile, the hierarchical relationships between metabolic reactions were created in accordance with the classification of compound pairs. This hierarchical framework for relating compounds and reactions can be an essential step to creating a comprehensive organization of all reaction descriptions at a desired chemical specificity to fit a given application. Such a comprehensive organization of reaction descriptions would be useful to a wide range of possible applications, including metabolic modeling, metabolite and reaction prediction, and network incorporation of newly discovered metabolites.

In addition, we made use of the atom identifiers derived from the neighborhood-specific graph coloring method to evaluate the consistency of atom mappings across harmonized reactions. Through the evaluation, we figured out that the documentation of atom mappings in either database can contain issues. Compared to other representations, KEGG RCLASS provides a concise RDM description of reaction atom mappings between a reaction-product compound pair, which appears more resistant to consistency errors. This resistance to consistency error is due to a separation of the atom mappings from the specific atom order in the molfile representations, which allows update molfile representations without affecting the RDM descriptions. However, a few KEGG RCLASS entries are computationally hard to parse due to a combinatorial issue caused by the several factors: multiple reaction centers in a single reaction, symmetric compounds, and reaction descriptions involving multi-steps. This combination of factors introduces a large number of possibilities when match reaction center atoms to the RDM descriptions. To prevent this combinatorial problem, one possible solution is to represent multiple reaction center atoms with their match atoms and associated difference atoms within a paired substructure representation, like sdfile.

The above work helps to build a comprehensive atom-resolved metabolic network. However, this is just the first step in interpreting metabolomics datasets. In Chapter 4, we presented a moiety modeling framework for deconvoluting metabolite isotopologue profiles using moiety models along with the analysis and selection of the best moiety model(s) based on the experimental data. This moiety modeling framework successfully integrates model representation, model optimization, and model selection together. To our knowledge, this is the first framework that can analyze datasets involving single and multiple isotope tracers as demonstrated on simulated datasets for multiple tracer models and both simulated and experimental datasets on single tracer models. Furthermore, rather than a single predefined metabolic model, this method allows comparison of multiple metabolic models derived from a given metabolic profile. In addition, we demonstrated the robust performance of the moiety modeling framework in model selection on real experimental datasets with a ^{13}C-labeled UDP-GlcNAc isotopologue dataset. The selection of correct moiety models is the premise for generating deconvolution results that can be accurately interpreted in terms of relative metabolic flux.

In Chapter 5, we further explored the data quality requirements and the factors that affect model selection. We have demonstrated the importance of model selection in isotopic flux analysis as a proxy for metabolic flux analysis. Here, we found that it is not the optimization method per se, but the degree of optimization that influences model selection, due to the effects of overfitting. Moreover, the objective function in model optimization also plays important role in model selection. Our results indicated that proper selection of an objective function can help increase resistance to failure in model selection, which suggests that different objective functions should be used for model selection versus parameter optimization for flux interpretation. It is often difficult to acquire SIRM experimental datasets with enough replicates at multiple time points due to the cost and effort, making it impractical to directly estimate error in these datasets. In addition, different types of systematic error can limit the effective replicate-based error analysis. To address this problem, we can conduct a set of gradient experiments with varying optimization degree using the SAGA-optimize method to estimate the failure point in model selection caused by overfitting. We also found that incorporation of less informative datasets can hinder successful model selection since they lack enough information representing the incorporation of isotopes simulated by moiety models, leading to increased errors in model selection. On the other hand, combination of informative datasets can help control failure in model selection, suggesting that informative datasets in public metabolomics repositories can be combined to facilitate robust model selection. More importantly, these datasets do not need to come from the same biological systems. We just need to make sure that the biological systems utilize the same part of metabolism being measured and modeled. The implies that SIRM datasets in public repositories of reasonable quality can be combined with newly acquired datasets to improve model selection. Therefore, curation efforts of public metabolomics repositories to maintain high-quality data and provide metrics of measurement error could have a huge impact on future metabolic modeling.

In conclusion, compound and reaction harmonization through the neighborhood-specific coloring method can help build a more comprehensive metabolic network, which will help derive the proper context-specific metabolic models for metabolic flux analysis. The moiety model modeling framework has demonstrated its robust performance in isotopologue deconvolution and moiety model selection. We also devised a strategy to deal with failure in model selection caused by over-optimization.

APPENDICES

APPENDIX 1. SOFTWARE

moiety_modeling software:

GitHub - https://github.com/MoseleyBioinformaticsLab/moiety_modeling

PyPI - https://pypi.org/project/moiety-modeling/

SAGA-optimization package:

GitHub - https://github.com/MoseleyBioinformaticsLab/SAGA_optimize

PyPI - https://pypi.org/project/SAGA-optimize/

APPENDIX 2. DATA AVAILABILITY

Atom Identifiers Generated by a Neighborhood-Specific Graph Coloring Method Enable Compound Harmonization across Metabolic Databases: https://doi.org/10.6084/m9.figshare.12894008.

Hierarchical Harmonization of Atom-Resolved Metabolic Reactions across Metabolic Databases: https://doi.org/10.6084/m9.figshare.14703999.

Moiety Modeling Framework for Deriving Moiety Abundances from Mass Spectrometry Measured Isotopologues: https://figshare.com/articles/moiety_modeling_framework/7886135.

Robust Moiety Model Selection Using Mass Spectrometry Measured Isotopologues: https://figshare.com/articles/moiety_model_selection/10279688.

REFERENCES

1. Horgan, R.P. and L.C. Kenny, *'Omic' technologies: genomics, transcriptomics, proteomics and metabolomics.* Obstetrician & Gynaecologist, 2011. **13**(3): p. 189-195.
2. Rochfort, S., *Metabolomics Reviewed: A New "Omics" Platform Technology for Systems Biology and Implications for Natural Products Research.* Journal of Natural Products, 2005. **68**(12): p. 1813-1820.
3. Yan, M. and G. Xu, *Current and future perspectives of functional metabolomics in disease studies–A review.* Analytica Chimica Acta, 2018. **1037**: p. 41-54.
4. Hollywood, K., D.R. Brison, and R. Goodacre, *Metabolomics: Current technologies and future trends.* PROTEOMICS, 2006. **6**(17): p. 4716-4723.
5. Ma, Y., et al., *Metabolomics in the fields of oncology: a review of recent research.* Molecular Biology Reports, 2012. **39**(7): p. 7505-7511.
6. Worley, B. and R. Powers, *Multivariate Analysis in Metabolomics.* Current Metabolomics, 2013. **1**(1): p. 92-107.
7. Johnson, H.E., et al., *Metabolic fingerprinting of salt-stressed tomatoes.* Phytochemistry, 2003. **62**(6): p. 919-928.
8. Idle, J.R. and F.J. Gonzalez, *Metabolomics.* Cell Metabolism, 2007. **6**(5): p. 348-351.
9. Kell, D.B., *Metabolomics and systems biology: making sense of the soup.* Current Opinion in Microbiology, 2004. **7**(3): p. 296-307.
10. Siew, N., Y. Azaria, and D. Fischer, *The ORFanage: an ORFan database.* Nucleic Acids Research, 2004. **32**(suppl_1): p. D281-D283.
11. Liang, P., B. Labedan, and M. Riley, *Physiological genomics of Escherichia coli protein families.* Physiological Genomics, 2002. **9**(1): p. 15-26.
12. Lee, S.Y., J.M. Park, and T.Y. Kim, *Application of metabolic flux analysis in metabolic engineering.* Methods Enzymol, 2011. **498**: p. 67-93.
13. Vinayavekhin, N., E.A. Homan, and A. Saghatelian, *Exploring Disease through Metabolomics.* ACS Chemical Biology, 2010. **5**(1): p. 91-103.
14. Wilcoxen, K.M., et al., *Practical metabolomics in drug discovery.* Expert Opinion on Drug Discovery, 2010. **5**(3): p. 249-263.
15. Powers, R., *NMR metabolomics and drug discovery.* Magnetic Resonance in Chemistry, 2009. **47**(S1): p. S2-S11.
16. Hanahan, D. and Robert A. Weinberg, *Hallmarks of Cancer: The Next Generation.* Cell, 2011. **144**(5): p. 646-674.
17. Yoshida, G.J., *Metabolic reprogramming: the emerging concept and associated therapeutic strategies.* Journal of Experimental & Clinical Cancer Research, 2015. **34**(1): p. 111.
18. Ward, Patrick S. and Craig B. Thompson, *Metabolic Reprogramming: A Cancer Hallmark Even Warburg Did Not Anticipate.* Cancer Cell, 2012. **21**(3): p. 297-308.
19. Phan, L.M., S.-C.J. Yeung, and M.-H. Lee, *Cancer metabolic reprogramming: importance, main features, and potentials for precise targeted anti-cancer therapies.* Cancer biology & medicine, 2014. **11**(1): p. 1-19.
20. DeBerardinis, R.J. and T. Cheng, *Q's next: the diverse functions of glutamine in metabolism, cell biology and cancer.* Oncogene, 2010. **29**(3): p. 313-324.

21. Dang, C.V., *Glutaminolysis: Supplying carbon or nitrogen or both for cancer cells?* Cell Cycle, 2010. **9**(19): p. 3884-3886.

22. Santos, C.R. and A. Schulze, *Lipid metabolism in cancer.* The FEBS Journal, 2012. **279**(15): p. 2610-2623.

23. Medes, G., A. Thomas, and S. Weinhouse, *Metabolism of Neoplastic Tissue. IV. A Study of Lipid Synthesis in Neoplastic Tissue Slices <em>in Vitro</em>.* Cancer Research, 1953. **13**(1): p. 27-29.

24. Kuhajda, F.P., et al., *Fatty acid synthesis: a potential selective target for antineoplastic therapy.* Proceedings of the National Academy of Sciences, 1994. **91**(14): p. 6379-6383.

25. Yoon, S., et al., *Up-regulation of Acetyl-CoA Carboxylase α and Fatty Acid Synthase by Human Epidermal Growth Factor Receptor 2 at the Translational Level in Breast Cancer Cells*.* Journal of Biological Chemistry, 2007. **282**(36): p. 26122-26131.

26. Swinnen, J.V., et al., *Selective activation of the fatty acid synthesis pathway in human prostate cancer.* International Journal of Cancer, 2000. **88**(2): p. 176-179.

27. Menendez, J.A. and R. Lupu, *Fatty acid synthase and the lipogenic phenotype in cancer pathogenesis.* Nature Reviews Cancer, 2007. **7**(10): p. 763-777.

28. Accioly, M.T., et al., *Lipid Bodies Are Reservoirs of Cyclooxygenase-2 and Sites of Prostaglandin-E<sub>2</sub> Synthesis in Colon Cancer Cells.* Cancer Research, 2008. **68**(6): p. 1732-1740.

29. DeLaBarre, B., et al., *Full-Length Human Glutaminase in Complex with an Allosteric Inhibitor.* Biochemistry, 2011. **50**(50): p. 10764-10770.

30. Wang, J.-B., et al., *Targeting Mitochondrial Glutaminase Activity Inhibits Oncogenic Transformation.* Cancer Cell, 2010. **18**(3): p. 207-219.

31. Maher, J.C., et al., *Hypoxia-inducible factor-1 confers resistance to the glycolytic inhibitor 2-deoxy-<span class="sc">d</span>-glucose.* Molecular Cancer Therapeutics, 2007. **6**(2): p. 732-741.

32. Pelicano, H., et al., *Glycolysis inhibition for anticancer treatment.* Oncogene, 2006. **25**(34): p. 4633-4646.

33. Pajak, B., et al., *2-Deoxy-d-Glucose and Its Analogs: From Diagnostic to Therapeutic Agents.* International Journal of Molecular Sciences, 2020. **21**(1): p. 234.

34. Morandi, A. and S. Indraccolo, *Linking metabolic reprogramming to therapy resistance in cancer.* Biochimica et Biophysica Acta (BBA) - Reviews on Cancer, 2017. **1868**(1): p. 1-6.

35. Zhao, Y., et al., *Overcoming Trastuzumab Resistance in Breast Cancer by Targeting Dysregulated Glucose Metabolism.* Cancer Research, 2011. **71**(13): p. 4585-4597.

36. Perez De Souza, L., et al., *Network-based strategies in metabolomics data analysis and interpretation: from molecular networking to biological interpretation.* Expert Review of Proteomics, 2020. **17**(4): p. 243-255.

37. Aittokallio, T., *Graph-based methods for analysing networks in cell biology*, in *Briefings in Bioinformatics*. 2006. p. 243-255.

38. Duarte, N.C., et al., *Global reconstruction of the human metabolic network based on genomic and bibliomic data.* Proceedings of the National Academy of Sciences, 2007. **104**(6): p. 1777-1782.

39. Hao, T., et al., *Compartmentalization of the Edinburgh Human Metabolic Network.* BMC Bioinformatics, 2010. **11**(1): p. 393.

40. Thiele, I., et al., *A community-driven global reconstruction of human metabolism.* Nature Biotechnology, 2013. **31**(5): p. 419-425.

41. Wu, M. and C. Chan, *Human Metabolic Network: Reconstruction, Simulation, and Applications in Systems Biology.* Metabolites, 2012. **2**(1): p. 242-253.

42. Gille, C., et al., *HepatoNet1: a comprehensive metabolic reconstruction of the human hepatocyte for the analysis of liver physiology*. Molecular Systems Biology, 2010. **6**(1): p. 411.

43. Sahoo, S., et al., *A compendium of inborn errors of metabolism mapped onto the human metabolic network.* Molecular bioSystems, 2012. **8**(10): p. 2545.

44. Sahoo, S. and I. Thiele, *Predicting the impact of diet and enzymopathies on human small intestinal epithelial cells.* Human Molecular Genetics, 2013. **22**(13): p. 2705-2722.

45. Becker, S.A. and B.O. Palsson, *Context-Specific Metabolic Networks Are Consistent with Experiments.* PLOS Computational Biology, 2008. **4**(5): p. e1000082.

46. Jerby, L., T. Shlomi, and E. Ruppin, *Computational reconstruction of tissue-specific metabolic models: application to human liver metabolism.* Molecular Systems Biology, 2010. **6**(1): p. 401.

47. Bordbar, A., et al., *Model-driven multi-omic data analysis elucidates metabolic immunomodulators of macrophage activation.* Molecular Systems Biology, 2012. **8**(1): p. 558.

48. Fan, T.W., et al., *Stable isotope-resolved metabolomics and applications for drug development.* Pharmacol Ther, 2012. **133**(3): p. 366-91.

49. Lane, A.N., R.M. Higashi, and T.W.M. Fan, *NMR and MS-based Stable Isotope-Resolved Metabolomics and applications in cancer metabolism.* TrAC Trends in Analytical Chemistry, 2019. **120**: p. 115322.

50. Niedenführ, S., W. Wiechert, and K. Nöh, *How to measure metabolic fluxes: a taxonomic guide for 13C fluxomics.* Current Opinion in Biotechnology, 2015. **34**: p. 82-90.

51. Muller, P., *Glossary of terms used in physical organic chemistry (IUPAC Recommendations 1994).* Pure and Applied Chemistry, 1994. **66**(5): p. 1077-1184.

52. Dai, Z. and J.W. Locasale, *Understanding metabolism with flux analysis: From theory to application.* Metabolic Engineering, 2017. **43**: p. 94-102.

53. Antoniewicz, M.R., *Methods and advances in metabolic flux analysis: a mini-review.* J Ind Microbiol Biotechnol, 2015. **42**(3): p. 317-25.

54. Sauer, U. and N. Zamboni, *From biomarkers to integrated network responses.* Nature Biotechnology, 2008. **26**(10): p. 1090-1092.

55. Quek, L.-E., et al., *OpenFLUX: efficient modelling software for 13C-based metabolic flux analysis.* Microbial Cell Factories, 2009. **8**(1): p. 25.

56. Antoniewicz, M.R., J.K. Kelleher, and G. Stephanopoulos, *Elementary metabolite units (EMU): A novel framework for modeling isotopic distributions*, in *Metabolic Engineering*. 2007. p. 68-86.

57. Mancuso, A., et al., *Examination of primary metabolic pathways in a murine hybridoma with carbon-13 nuclear magnetic resonance spectroscopy.* Biotechnology and Bioengineering, 1994. **44**(5): p. 563-585.

58. Sauer, U., et al., *Metabolic flux ratio analysis of genetic and environmental modulations of Escherichia coli central carbon metabolism.* J Bacteriol, 1999. **181**(21): p. 6679-88.

59. Szyperski, T., et al., *Bioreaction Network Topology and Metabolic Flux Ratio Analysis by Biosynthetic Fractional 13C Labeling and Two-Dimensional NMR Spectroscopy.* Metabolic Engineering, 1999. **1**(3): p. 189-197.

60. Emmerling, M., et al., *Metabolic flux responses to pyruvate kinase knockout in Escherichia coli.* J Bacteriol, 2002. **184**(1): p. 152-64.

61. Fischer, E. and U. Sauer, *Metabolic flux profiling of Escherichia coli mutants in central carbon metabolism using GC-MS.* Eur J Biochem, 2003. **270**(5): p. 880-91.

62. Fischer, E. and U. Sauer, *Large-scale in vivo flux analysis shows rigidity and suboptimal performance of Bacillus subtilis metabolism.* Nat Genet, 2005. **37**(6): p. 636-40.

63. Wiechert, W., et al., *A Universal Framework for 13C Metabolic Flux Analysis.* Metabolic Engineering, 2001. **3**(3): p. 265-283.

64. Zamboni, N., et al., *13C-based metabolic flux analysis.* Nature Protocols, 2009. **4**(6): p. 878-892.

65. Young, J.D., *INCA: a computational platform for isotopically non-stationary metabolic flux analysis.* Bioinformatics, 2014. **30**(9): p. 1333-5.

66. Weitzel, M., et al., *13CFLUX2—high-performance software suite for 13C-metabolic flux analysis.* Bioinformatics, 2012. **29**(1): p. 143-145.

67. Kogadeeva, M. and N. Zamboni, *SUMOFLUX: A Generalized Method for Targeted 13C Metabolic Flux Ratio Analysis.* PLOS Computational Biology, 2016. **12**(9): p. e1005109.

68. Moseley, H.N., et al., *A novel deconvolution method for modeling UDP-N-acetyl-D-glucosamine biosynthetic pathways based on 13C mass isotopologue profiles under non-steady-state conditions*, in *BMC Biology*. 2011. p. 37.

69. Jin, H. and H.N.B. Moseley, *Moiety Modeling Framework for Deriving Moiety Abundances from Mass Spectrometry Measured Isotopologues*, in *bmc bioinformatics*. 2019.

70. Jin, H. and H.N.B. Moseley, *Robust Moiety Model Selection Using Mass Spectrometry Measured Isotopologues*, in *Metabolites*. 2020. p. 118.

71. Rathahao-Paris, E., et al., *High resolution mass spectrometry for structural identification of metabolites in metabolomics*, in *Metabolomics*. 2016. p. 10.

72. Chokkathukalam, A., et al., *Stable isotope-labeling studies in metabolomics: new insights into structure and dynamics of metabolic networks.* Bioanalysis, 2014. **6**(4): p. 511-24.

73. Arita, M., *In Silico Atomic Tracing by Substrate-Product Relationships in Escherichia coli Intermediary Metabolism*, in *Genome Research*. 2003. p. 2455-2466.

74. Arita, M., Y. Fujiwara, and Y. Nakanishi, *Map Editor for the Atomic Reconstruction of Metabolism (ARM)*, in *Plant Metabolomics*. Springer-Verlag: Berlin/Heidelberg. p. 129-139.
75. Hadadi, N., et al., *Reconstruction of biological pathways and metabolic networks from in silico labeled metabolites*, in *Biotechnology Journal*. 2017. p. 1600464.
76. Dalby, A., et al., *Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited*, in *Journal of Chemical Information and Modeling*. 1992. p. 244-255.
77. Kotera, M., et al., *RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions*, in *Genome Informatics*. 2004. p. P062.
78. Latendresse, M., et al., *Accurate Atom-Mapping Computation for Biochemical Reactions*, in *Journal of Chemical Information and Modeling*. 2012. p. 2970-2982.
79. Pitkänen, E., P. Jouhten, and J. Rousu, *Inferring branching pathways in genome-scale metabolic networks*, in *BMC Systems Biology*. 2009. p. 103.
80. Heath, A.P., G.N. Bennett, and L.E. Kavraki, *Finding metabolic pathways using atom tracking.* Bioinformatics, 2010. **26**(12): p. 1548-55.
81. Latendresse, M., M. Krummenacker, and P.D. Karp, *Optimal metabolic route search based on atom mappings*, in *Bioinformatics*. 2014. p. 2043-2050.
82. Altman, T., et al., *A systematic comparison of the MetaCyc and KEGG pathway databases.* BMC Bioinformatics, 2013. **14**(1): p. 112.
83. Dashti, H., et al., *Unique identifiers for small molecules enable rigorous labeling of their atoms*, in *Scientific Data*. 2017. p. 170073.
84. Willighagen, E.L., et al., *The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching*, in *Journal of Cheminformatics*. 2017. p. 33.
85. Smelter, A. *ctfile*. [cited 2019 Nov 26]; Available from: https://github.com/MoseleyBioinformaticsLab/ctfile.
86. T. Bray, E., *The JavaScript Object Notation (JSON) Data Interchange Format.* 2014.
87. *Indigo Toolkit*. [cited 2020 Apr 30]; Available from: https://lifescience.opensource.epam.com/indigo/index.html
88. Hattori, M., et al., *Development of a Chemical Structure Comparison Method for Integrated Analysis of Chemical and Genomic Information in the Metabolic Pathways*, in *Journal of the American Chemical Society*. 2003. p. 11853-11865.
89. Mitchell, J.M., et al., *Development and in silico evaluation of large-scale metabolite identification methods using functional group detection for metabolomics*, in *Frontiers in Genetics*. 2014.
90. Teixeira, A.L., J.P. Leal, and A.O. Falcao, *Automated Identification and Classification of Stereochemistry: Chirality and Double Bond Stereoisomerism*. 2013.
91. O'Boyle, N.M., et al., *Open Babel: An open chemical toolbox.* Journal of Cheminformatics, 2011. **3**(1): p. 33.
92. Cormen, T.H., et al., *Introduction to Algorithms*. 2001, MIT Press and McGraw-Hill. p. 531-539.
93. Heller, S., et al., *InChI - the worldwide chemical structure identifier standard*, in *Journal of Cheminformatics*. 2013. p. 7.

94. Dashti, H., et al., *Automated evaluation of consistency within the PubChem Compound database*, in *Scientific Data*. 2019. p. 190023.

95. Ramar, R. and S. Venkatasubramanian, *Neighbourhood distinguishing coloring in graphs*, in *Innovations in Incidence Geometry: Algebraic, Topological and Combinatorial*. 2013. p. 135-140.

96. Li, Y., A. Razborov, and B. Rossman, *On the AC^0 Complexity of Subgraph Isomorphism*, in *SIAM Journal on Computing*. 2017. p. 936-971.

97. McDonald, A.G., S. Boyce, and K.F. Tipton, *ExplorEnz: the primary source of the IUBMB enzyme list*, in *Nucleic Acids Research*. 2009. p. D593-D597.

98. Danchin, A., *Enzyme nomenclature, recommendations (1992) of the nomenclature committee or the international union of biochemistry and molecular biology*, in *Biochimie*. 1993. p. 501.

99. Szatylowicz, H., et al., *Why 1,2-quinone derivatives are more stable than their 2,3-analogues?*, in *Theoretical Chemistry Accounts*. 2015. p. 35.

100. Ivanov, J. and G. Schüürmann, *Simple Algorithms for Determining the Molecular Symmetry*, in *Journal of Chemical Information and Computer Sciences*. 1999. p. 728-737.

101. Tinhofer, G. and M. Klin. *Algebraic Combinatorics in Mathematical Chemistry. Methods and Algorithms. III. Graph Invariants and*. 1999.

102. Schneider, N., R.A. Sayle, and G.A. Landrum, *Get Your Atoms in Order—An Open-Source Implementation of a Novel and Robust Molecular Canonicalization Algorithm*, in *Journal of Chemical Information and Modeling*. 2015. p. 2111-2120.

103. Pham, N., et al., *Consistency, Inconsistency, and Ambiguity of Metabolite Names in Biochemical Databases Used for Genome-Scale Metabolic Modelling.* Metabolites, 2019. **9**(2).

104. Ryu, J.Y., H.U. Kim, and S.Y. Lee, *Reconstruction of genome-scale human metabolic models using omics data*, in *Integrative Biology*. 2015. p. 859-868.

105. Contreras, A., et al., *Mapping the Physiological Response of Oenococcus oeni to Ethanol Stress Using an Extended Genome-Scale Metabolic Model.* Frontiers in Microbiology, 2018. **9**(291).

106. Lee, D.S., et al., *Comparative genome-scale metabolic reconstruction and flux balance analysis of multiple Staphylococcus aureus genomes identify novel antimicrobial drug targets.* J Bacteriol, 2009. **191**(12): p. 4015-24.

107. Patil, K.R., M. Akesson, and J. Nielsen, *Use of genome-scale microbial models for metabolic engineering.* Curr Opin Biotechnol, 2004. **15**(1): p. 64-9.

108. Radrich, K., et al., *Integration of metabolic databases for the reconstruction of genome-scale metabolic networks.* BMC Syst Biol, 2010. **4**: p. 114.

109. Zhang, C. and Q. Hua, *Applications of Genome-Scale Metabolic Models in Biotechnology and Systems Medicine.* Front Physiol, 2015. **6**: p. 413.

110. Creek, D.J., et al., *Stable isotope-assisted metabolomics for network-wide metabolic pathway elucidation.* Anal Chem, 2012. **84**(20): p. 8442-7.

111. Ginsburg, H., *Caveat emptor: limitations of the automated reconstruction of metabolic pathways in Plasmodium.* Trends Parasitol, 2009. **25**(1): p. 37-43.

112. Poolman, M.G., et al., *Challenges to be faced in the reconstruction of metabolic networks from public databases.* Syst Biol (Stevenage), 2006. **153**(5): p. 379-84.

113. Saha, R., A. Chowdhury, and C.D. Maranas, *Recent advances in the reconstruction of metabolic models and integration of omics data.* Curr Opin Biotechnol, 2014. **29**: p. 39-45.

114. Qi, X., Z.M. Ozsoyoglu, and G. Ozsoyoglu, *Matching metabolites and reactions in different metabolic networks.* Methods, 2014. **69**(3): p. 282-97.

115. van Heck, R.G., et al., *Efficient Reconstruction of Predictive Consensus Metabolic Network Models.* PLoS Comput Biol, 2016. **12**(8): p. e1005085.

116. Heller, S.R., et al., *InChI, the IUPAC International Chemical Identifier.* J Cheminform, 2015. **7**: p. 23.

117. Weininger, D., A. Weininger, and J.L. Weininger, *SMILES. 2. Algorithm for generation of unique SMILES notation.* Journal of Chemical Information and Computer Sciences, 1989. **29**(2): p. 97-101.

118. Lieven, C., et al., *Memote: A community driven effort towards a standardized genome-scale metabolic model test suite.* bioRxiv, 2018: p. 350991.

119. Jin, H., J.M. Mitchell, and H.N.B. Moseley, *Atom Identifiers Generated by a Neighborhood-Specific Graph Coloring Method Enable Compound Harmonization across Metabolic Databases.* Metabolites, 2020. **10**(9).

120. Jeffryes, J.G., et al., *MINEs: open access databases of computationally predicted enzyme promiscuity products for untargeted metabolomics.* Journal of Cheminformatics, 2015. **7**(1): p. 44.

121. Hadadi, N., et al., *ATLAS of Biochemistry: A Repository of All Possible Biochemical Reactions for Synthetic Biology and Metabolic Engineering Studies.* ACS Synth Biol, 2016. **5**(10): p. 1155-1166.

122. Frainay, C., et al., *Mind the Gap: Mapping Mass Spectral Databases in Genome-Scale Metabolic Networks Reveals Poorly Covered Areas.* Metabolites, 2018. **8**(3): p. 51.

123. Barrett, A.J., *Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme Nomenclature. Recommendations 1992. Supplement 4: corrections and additions (1997).* Eur J Biochem, 1997. **250**(1): p. 1-6.

124. Kotera, M., et al., *Computational Assignment of the EC Numbers for Genomic-Scale Analysis of Enzymatic Reactions.* Journal of the American Chemical Society, 2004. **126**(50): p. 16487-16498.

125. Altman, T., et al., *A systematic comparison of the MetaCyc and KEGG pathway databases*, in *BMC Bioinformatics*. 2013. p. 112.

126. Jeske, L., et al., *BRENDA in 2019: a European ELIXIR core data resource.* Nucleic Acids Research, 2018. **47**(D1): p. D542-D549.

127. DeBerardinis, R.J. and C.B. Thompson, *Cellular Metabolism and Disease: What Do Metabolic Outliers Teach Us?*, in *Cell*. 2012. p. 1132-1144.

128. Verdegem, D., et al., *MAIMS: a software tool for sensitive metabolic tracer analysis through the deconvolution of 13C mass isotopologue profiles of large composite metabolites*, in *Metabolomics*. 2017. p. 123.

129. Wilken, S.E., et al., *Linking 'omics' to function unlocks the biotech potential of non-model fungi*, in *Current Opinion in Systems Biology*. 2019. p. 9-17.

130. Nash, S., *Newton-Type Minimization via the Lanczos Method*, in *SIAM Journal on Numerical Analysis*. 1984, Society for Industrial and Applied Mathematics. p. 770-788.
131. Boggs, P.T. and J.W. Tolle, *Sequential Quadratic Programming*, in *Acta Numerica*. 1995. p. 1.
132. Zhu, C., et al., *Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization*, in *ACM Transactions on Mathematical Software*. 1997. p. 550-560.
133. Akaike, H., *Information Theory and an Extension of the Maximum Likelihood Principle*. 1998. p. 199-213.
134. Schwarz, G., *Estimating the Dimension of a Model*, in *The Annals of Statistics*. 1978. p. 461-464.
135. Aguilar, D., *jsonpickle*.
136. Cavanaugh, J.E., *Unifying the derivations for the Akaike and corrected Akaike information criteria*, in *Statistics & Probability Letters*. 1997. p. 201-208.
137. Wit, E., E.v.d. Heuvel, and J.-W. Romeijn, *'All models are wrong...': an introduction to model uncertainty*, in *Statistica Neerlandica*. 2012. p. 217-236.
138. Smelter, A., E.C. Rouchka, and H.N.B. Moseley, *Detecting and accounting for multiple sources of positional variance in peak list registration analysis and spin system grouping*, in *Journal of Biomolecular NMR*. 2017. p. 281-296.
139. Moseley, H.N.B., *ERROR ANALYSIS AND PROPAGATION IN METABOLOMICS DATA ANALYSIS*, in *Computational and Structural Biotechnology Journal*. 2013. p. e201301006.
140. Pavlova, N.N. and C.B. Thompson, *Perspective The Emerging Hallmarks of Cancer Metabolism*, in *Cell Metabolism*. 2016, Elsevier Inc. p. 27-47.
141. Wishart, D.S., *Applications of Metabolomics in Drug Discovery and Development*, in *Drugs in R & D*. 2008. p. 307-322.
142. Sauer, U., *Metabolic networks in motion: 13 C ‐ based flux analysis*, in *Molecular Systems Biology*. 2006. p. 62.
143. Krumsiek, J., et al., *Mining the Unknown: A Systems Approach to Metabolite Identification Combining Genetic and Metabolic Information*, in *PLoS Genetics*, M.I. McCarthy, Editor. 2012. p. e1003005.
144. Basu, S., et al., *Sparse network modeling and Metscape-based visualization methods for the analysis of large-scale metabolomics data*, in *Bioinformatics*. 2017. p. btx012.

VITA

Huan Jin

EDUCATION

BS in Food Science and Engineering from Guangxi University, Guangxi, Nanning, China, Sep 2010 – Jun 2014

MS in Nutrition and Food Safety from China Agricultural University, Beijing, China, Sep 2014 – Jun 2016

ACADEMIC EMPLOYMENT

Graduate Research Assistant, Department of Toxicology and Cancer Biology, University of Kentucky, Lexington, KY, 2016-Present

Student Research Assistant, College of Food Science and Nutritional Engineering, China Agricultural University, Beijing, China, Sep 2014 – Jun 2016

SCHOLASTIC AND PROFESSIONAL HONORS

Metabolomics Association of North America Conference Travel Award, 2019

Excellent Graduate, China Agricultural University, Beijing, 2016

Excellent Dissertation, China Agricultural University, Beijing, 2016

National Scholarship, Guangxi University, Nanning, Guangxi, 2012

PROFESSIONAL PUBLICATIONS

Jin H and Moseley HNB. Hierarchical Harmonization of Atom-Resolved Metabolic Re-actions Across Metabolic Databases. ***Metabolites***, 11: 431, 2021.

Jin H, Mitchell JM, and Moseley HNB. Atom Identifiers Generated by a Neighborhood-Specific Graph Coloring Method Enable Compound Harmonization across Metabolic Databases. ***Metabolites***, 10: 368, 2020.

Jin H and Moseley HNB. Robust Moiety Model Selection Using Mass Spectrometry Measured Isotopologues. ***Metabolites***, 10: 118, 2020.

Jin H and Moseley HNB. Moiety Modeling Framework for Deriving Moiety Abundances from Mass Spectrometry Measured Isotopologues. ***BMC Bioinformatics***, 20: 524, 2019.

Kamelgarn M, Chen J, Kuang L, Jin H, Kasarskis EJ, and Zhu H. ALS mutations of FUS suppress protein translation and disrupt the regulation of nonsense-mediated decay. ***Proceedings of the National Academy of Sciences***. 115: 11904, 2018.

Jin H, Yin S, Song X, et al. p53 activation contributes to patulin-induced nephrotoxicity via modulation of reactive oxygen species generation. ***Sci. Rep***. 6: 24455, 2016.