RICE UNIVERSITY

Computational Discovery and Analysis of Metabolic Pathways

by

Allison Park Heath

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Lydia Kavraki

Lydia E. Kavraki (Chair)
Professor of Computer Science
Rice University

Bennett

George N. Bennett
Professor of Biochemistry and Cell Biology
Rice University

Luay K. Nakhleh
Assistant Professor of Computer Science
Rice University

Gábor Balázsi
Assistant Professor of Systems Biology
The University of Texas MD Anderson
Cancer Center

Houston, Texas
June, 2010

Abstract

Finding novel or non-standard metabolic pathways, possibly spanning multiple species, has important applications in fields such as metabolic engineering, metabolic network analysis, and metabolic network reconstruction. Traditionally, this has been a manual process, but the large volume of metabolic data now available has created a need for computational tools to automatically identify biologically relevant pathways. This thesis presents new algorithms for automatically finding biologically meaningful linear and branched metabolic pathways in multi-genome scale metabolic networks. These algorithms utilize atom mapping data, which provides the correspondence between atoms in the substrates to atoms in the products of a chemical reaction, to find pathways which conserve a given number of atoms between desired start and target compounds.

The first algorithm presented identifies atom conserving linear pathways by explicitly tracking atoms during an exploration of a graph structure constructed from the atom mapping data. The explicit tracking of atoms enables finding branched pathways because it provides automatic identification of the reactions and compounds through which atoms are lost or gained. The thesis then describes two algorithmic approaches for identifying branched metabolic pathways based upon atom conserving linear pathways. One approach takes one linear pathway at a time and attempts to add branches that connect loss and gain compounds. The other approach takes a group of linear pathways and attempts to merge pathways that move mutually exclusive sets of atoms from the start to the target compounds.

Comparisons to known metabolic pathways demonstrate that atom tracking causes the algorithms to avoid many unrealistic connections, often found in previous approaches, and return biologically meaningful pathways. While the theoretical complexity of finding even linear atom conserving pathways is high, by choosing the appropriate representations and heuristics, and perhaps due to the structure of the underlying data, the algorithms in this thesis have practical running times on real data. The results also demonstrate the potential of the algorithms to find novel or non-standard pathways that may span multiple organisms.

Acknowledgments

The completion of this document marks the end of my adventure as a graduate student. It has been an incredible journey that would have been impossible without the support, guidance, and friendship of many great people. First and foremost, I thank my advisor, Dr. Lydia Kavraki, for intellectual inspiration and guiding my way even when I veered off the beaten path. Dr. George Bennett has been a wonderful co-advisor and provided much of the motivation for the work contained in this thesis. I never expected biochemistry to be so fascinating! The quality of both my research and my experience as a graduate student has benefited greatly from the advice and feedback of Drs. Luay Nakhleh and Gábor Balázsi. My gratitude goes out to all of the members of the Physical and Biological Computing Group, whose positive impact on me and my work over the last eight years is unquantifiable.

I am grateful for my loving parents, Jim and Oksook Heath, who encouraged my curiosity, kept my bookshelf full, and taught me always to do things honestly and to the best of my ability: Thank you. And to my brother, Jason, for preventing me from ever being too serious. Special thanks goes to my husband, Joseph Fisk, for your unwavering support and ability to keep things in perspective. I am looking forward our next adventure together.

Finally, this work was made possible by various sources of funding that enable scientific pursuits. I have been supported in part by a training fellowship from the Biomedical Discovery Training Program of the W. M. Keck Center for Interdisciplinary Bioscience Training of the Gulf Coast Consortia (NIH Grant No. T90 DA022885 and R90 DA023418) as well as a National Science Foundation (NSF) Graduate Research Fellowship. Additional support came from NSF (CCF-0523908) and a Hamill Innovation Award from the Institute of Biosciences and Bioengineering at Rice University and Rice University funds. Computational resources were provided by the Shared University Grid at Rice, funded by NSF under Grant EIA-0216467 and partnership between Rice University, Sun Microsystems, and Sigma Solutions, Inc.

Contents

Illustrations

Tables

Chapter 1
Introduction

Metabolism refers to the chemical reactions, typically catalyzed by enzymes, that occur to sustain life. Experimental studies, coupled with advancements in molecular biology and improved computational methods, have generated increasingly large amounts of data on metabolic reactions and pathways in the last few decades. As a result, many specialized databases have been created in order to store and organize this metabolic data [56, 21]. One of the major goals of these databases is to make the data easily navigable and interpretable by their users. The data is usually presented as many small subpathways that are manually divided based on function and sometimes by organism. However, it can be difficult to navigate these subpathways to find connections between compounds, especially for discovering novel or non-standard pathways that may span multiple organisms. These type of pathways can be important for a number of applications such as metabolic engineering [9], understanding the metabolic scope of multi-species communities [36, 119] and metabolic network reconstruction [92, 35]. Combining parts of pathways existing in different organisms can lead to new ways of considering how metabolism works in complex communities and provide novel ways to synthesize important and useful compounds.

Computational tools can provide a way to automatically find biologically interesting pathways in metabolic data and reveal pathways that may have not been identified by manual means. The primary problem that computational methods for metabolic pathfinding try to solve is given a start and target compound, find "biologically meaningful" or "realistic" pathways of enzymatic reactions that make the target compound from the start compound. In many ways, methods for metabolic pathfinding can be thought of as "search engines" which try to find metabolic pathways, similar to finding websites, of interest from a large amount of data. In both cases, it is not always clear what the best results are, pathways or websites, to return for a particular query. Instead, it is dependent on the specific application that the user has in mind. While "biologically meaningful" and "realistic" are not well-defined, it is important to provide an initial validation of metabolic path finding tools based on their ability to find known metabolic pathways or by manual analysis of the results.

Graphs provide well-suited computational framework for modeling metabolism as a network, where the compounds and reactions are the vertices of the graph and the edges connect the compounds to the reactions they are involved in. Additionally, finding paths in graphs is a well-studied problem with a number of efficient algorithmic solutions for finding shortest and *k* shortest pathways. Unfortunately, as further discussed in the next chapter, paths found in standard graph representations of metabolic networks often do not correspond to biologically meaningful pathways [6, 35]. The work in this thesis is based on the observation that these graph-based methods do not work because they remove information about what is really going in a reaction biochemically. Each edge connecting the compounds to the reaction are considered the same, but biochemically one compound may be just providing the energy to drive the transformation of another chemical. One must look at the atomic level to understand which substrate chemicals are being transformed into which product chemical.

In the last few years the availability of atomic level information on which atoms in the substrate compounds correspond to which atoms in the product compounds, called *atom mapping data*, has been steadily increasing. The algorithms presented in this thesis harness atom mapping data to track atoms through metabolic networks to find pathways which conserve atoms from a given start compound to a give target compound. Atom tracking greatly increases the complexity of finding pathways in metabolic networks as compared to standard ways of finding paths in graphs. The general problem of finding linear pathways that conserve a given number of atoms, using atom mapping data, from the start to the target compound has been proven to be PSPACE-complete, and NP-complete when pathways are constrained to only use a compound once [17]. Fortunately, experimental results demonstrate that linear atom conserving pathways can be found in efficiently in practice. This is likely due to the underlying structure of metabolic networks combined with the fact that the number of atoms being tracked is often low. Atom tracking is a crucial feature, when using graph-based methods, for finding meaningful metabolic pathways because it essentially eliminates spurious connections and reactions that do not correspond to useful or real biochemical pathways or reactions. Furthermore, atom tracking enables finding branched metabolic pathways.

Branched metabolic pathways refer to pathways that contain linear pathways that interact biochemically to produce one molecule of the target compound. For example, a target compound may be made of three molecules of glucose, a six carbon compound, that undergo different modifications before participating in reactions to combine the molecules into one larger final compound. In this case, the number of atoms conserved is the total number of atoms that go from the starting molecules of glucose to the target compound. If all carbon atoms are conserved from the molecules of glucose, the branched pathway would be said to conserve 12 carbon atoms. Finding branched pathways that conserve a give number of atoms while using less than a given number of reactions has been shown to be NP-hard by reduction from the minimum set cover problem [91]. However, unlike the linear atom conserving pathways, finding branched pathways in practice appears to be closer to the theoretical complexity of the general problem. Therefore, algorithmic solutions require the use of heuristics in order to obtain branched, atom conserving, metabolic pathways.

1.1 Contributions

The main contribution of this thesis are three novel algorithms for finding both linear and branched metabolic pathways by using atom mapping data to track the movement of atoms through metabolic networks. The goal of these algorithms is to find biologically meaningful, possibly novel, metabolic pathways that may utilize reactions from multiple organisms without focusing on a particular organism or prior information on the environments in which they function. The algorithms all take as input atom mapping data, a start compound, a target compound, a minimum number of atoms to conserve and a maximum number of pathways to return. A set of metabolic pathways, that conserve at least given number of atoms from the start compound to the target compound, are returned. The first algorithm, LPAT, guarantees finding the *k* shortest linear metabolic pathways that conserve a given number of number of atoms. Even though this problem is of high theoretical complexity, experimental results show that the computational time required for LPAT is on the order of minutes in practice. The other algorithms, BPAT-S and BPAT-M, tackle the problem of finding branched metabolic pathways. Development of these algorithms revealed that finding branched metabolic pathways is a more complex problem in practice than finding linear metabolic pathways, even though they are of similarly high theoretical complexity. Therefore, BPAT-S and BPAT-M take two different heuristic approaches for finding branched metabolic pathways. This thesis also contributes extensive experimental testing of the algorithms, using metabolic data spanning multiple organisms, to demonstrate their practical performance. The results of the heuristic approaches of BPAT-S and BPAT-M are compared and highlight the different advantages and disadvantages of the approaches. The analysis of experimental results demonstrate that the algorithms in this thesis can find biologically meaningful and interesting pathways and improve on the current state of the art algorithms for find metabolic pathways.

1.2 Thesis Outline

The rest of the thesis proceeds by first providing background information and an overview of related work in Chapter 2. Chapter 3 describes LPAT, the algorithm developed for finding linear metabolic pathways using atom tracking. The computational complexity of finding branched metabolic pathways requires the use of heuristics, and so Chapter 4 presents two different algorithms, BPAT-S and BPAT-M, utilizing different heuristic approaches, for accomplishing this task. Chapter 5 describes the experimental setup and testing of all of the algorithms presented. The experimental setup describes how metabolic data was obtained and processed and the hardware used, and the experimental testing revolves around understanding how well the algorithms find known metabolic pathways. Chapter 6 is the final chapter, containing the conclusions and future directions for this work.

Chapter 2

Background

The study of metabolism, the chemical reactions that occur in cells to maintain life, has a long and rich history now spanning hundreds of years. The textbook version of metabolism breaks it down into relatively small and easy to understand metabolic pathways, such as glycolysis and the citric acid cycle [69]. While there is no denying that understanding the central metabolic pathways is important, in the last few decades, advances in molecular biology have greatly increased our knowledge and revealed a more complex picture of metabolism. The current view of metabolism is a network, rather than a set of straightforward pathways, containing all of the biochemical reactions which occur within an organism. A visualization of the metabolic network of *E. coli* can be found in Figure 2.1. This network view demonstrates the complexity found in just the metabolic network of one organism. A closer look reveals that a number of alternative pathways can exist to perform the same metabolic function or multiple pathways typically thought of as separate may be interconnected by using the same chemical compounds or enzymes. Finding interconnections and alternative pathways is important for understanding metabolism overall as well as how to make useful modifications to metabolic networks.

Modern molecular biology has not only enabled better understanding of metabolic networks, but has also brought the ability to manipulate metabolic networks through genetic engineering techniques. This has created a whole area of research, termed metabolic engineering, that is focused on the rational design, or redesign, of the genetic and regulatory processes in an organism, in order to increase production of a desired substance [9].

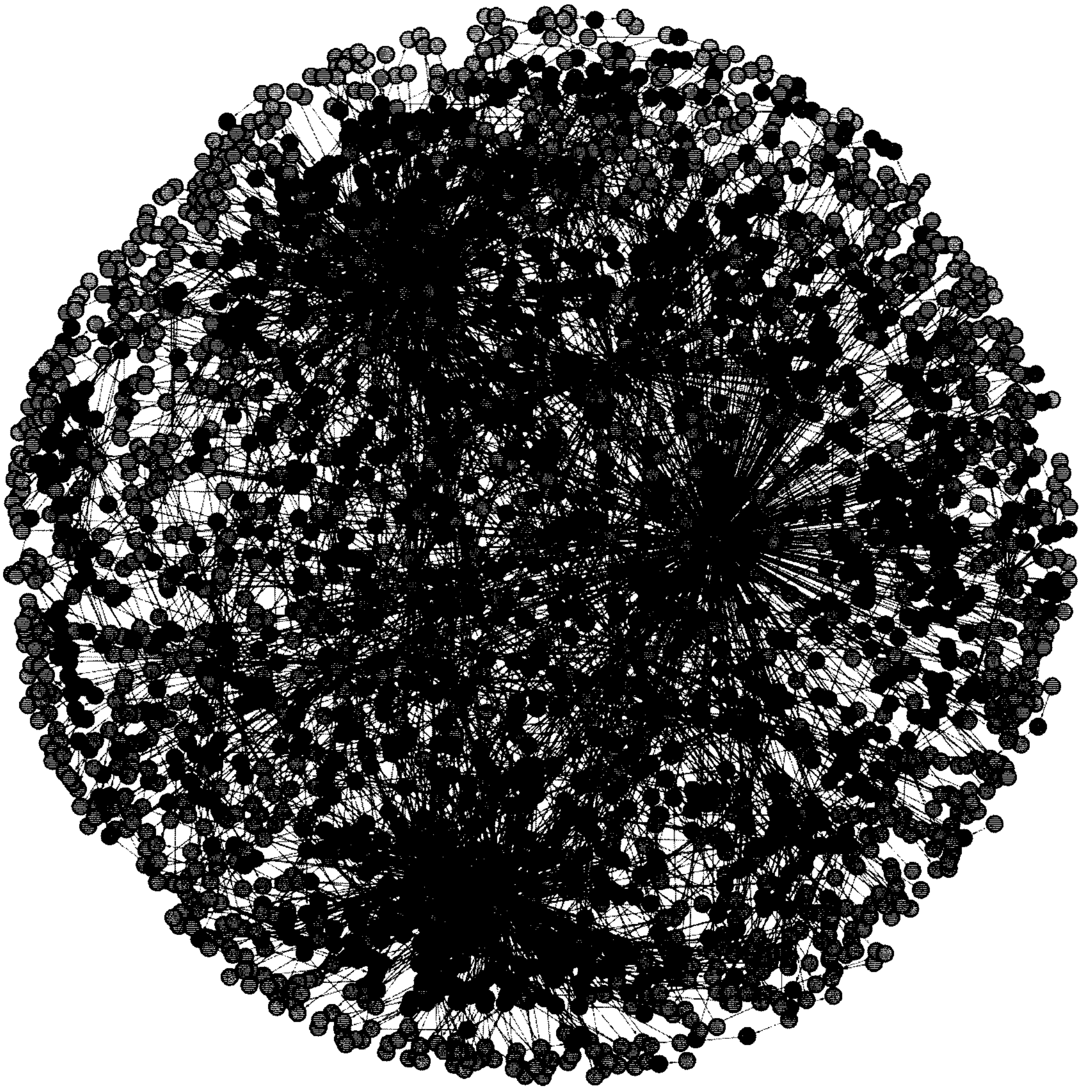


Figure 2.1 : Visualization of the metabolic network of *Escherichia coli* obtained from KEGG. The magenta nodes represent enzymatic reactions and the green nodes are chemical compounds. The network contains 2570 nodes and 5238 edges. The figure was made using Cytoscape [109].

Metabolic engineering has resulted in the successful biosynthesis of a number of compounds such as amino acids [71], drugs and drug precursors [68], and biofuels [67]. Metabolic engineering is a complex task which often requires extensive experimental trial and error. The metabolic engineering process begins by selecting a desired chemical product. Metabolic engineers then must carefully search multiple databases and scientific literature for metabolic reactions involving the desired product and possible precursors. These reactions usually come from more than one organism. This is a lengthy process with a goal of discovering a set of efficient reactions. To test this candidate network, specific genes are inserted into a host organism, incorporating the relevant reactions. Therefore, there is need for creating computational tools to help assist and accelerate the metabolic engineering process.

Computational techniques are well suited for automatically finding pathways in multi-genome scale metabolic networks that could be potentially used for metabolic engineering. Automated identification of metabolic pathways also has applications in areas such as metagenomics and metabolomics to assist with metabolic network reconstruction and understanding the metabolic scope of communities of organisms composed of multiple species [36, 119]. The algorithms in this thesis are general enough that they can be used for finding metabolic pathways for other applications, but the experimental results are geared towards metabolic engineering applications.

There have been two major areas of previous work on computationally finding metabolic pathways that differ primarily on the type of model used: stoichiometric, also known as constraint-based, and graph-based, also known as pathfinding, approaches [92, 65]. Both approaches have advantages and disadvantages and therefore, as highlighted by a recent pair of papers, the choice of approach is related to the specific biological questions that one wants to answer [34, 27]. Stoichiometric approaches require that the stoichiometry of metabolic reactions be satisfied and require the specification of the compounds that exist in the system. Stoichiometric approaches are better utilized for the analysis of small metabolic networks, frequently from a single organism, to understand the properties and behavior of the network. Graph-based methods model metabolic networks as graphs and typically ignore stoichiometric constraints. This enables graph based approaches to analyze much larger metabolic networks and find pathways without knowledge of the compounds that exist in the system. However, ignoring stoichiometric constraints also means that graph based approaches can return spurious connections and pathways. Therefore, a number of heuristics have been introduced in order to overcome this problem. The work in this thesis solves the problem of spurious connections by explicitly tracking atoms from the input compounds to the output compounds of a reaction.

The rest of this chapter will first provide an overview of the current state of available metabolic data, which is required for any metabolic path finding method. This will be followed by a brief discussion of stoichiometric methods, as they were the first computational methods for finding metabolic pathways, but not closely related to the work in this thesis. The chapter will conclude with a discussion of graph based approaches, which are more directly related to the work in this thesis, for finding both linear and branched metabolic pathways.

2.1 Metabolic Data Sources

A number of databases have been created in recent years to store and organize the ever increasing amount of data on metabolism. In April 2010, Pathguide, a listing of biological pathway resources, contained 58 currently available biological pathway resources related to metabolic data. These resources vary in their focus, size, curation levels, availability and popularity [8]. The current trend reveals that both the quality and quantity of metabolic data is increasing and becoming more difficult to search through and analysis manually.

Therefore, methods which can automatically identify relevant pieces of data are also becoming increasingly important. The goal of metabolic path finding applications is to find relevant pathways by using data about enzymatic reactions and the input and output chemical compounds. There are two major types of online databases that contain this type of the data. The first are metabolic pathway databases, which organize the reactions into pathways and focuses on the relationship of these pathways. The second are enzyme databases that focus on individual enzymes and their properties.

Two of the largest online databases focusing on metabolic pathway data are the Kyoto Encyclopedia of Genes and Genomes (KEGG) PATHWAY [56] database and MetaCyc [21], part of the larger BioCyc Database Collection [58]. MetaCyc and KEGG both contain similar data about metabolic pathways and their components and are freely available for academic use, but they vary in how this data is organized, curated and made available.

The KEGG PATHWAY database was initially released in December 1995 and contained a selected number of metabolic pathways. KEGG PATHWAY is manually curated and drawn based on the literature. The database in April 2010 contained 159 reference pathways under the metabolism category. Each reference pathway represents a metabolic function (e.g., glycolysis) and contains the enzymatic reactions and chemical compounds which make up the pathway. The reference pathways contain the union of the reactions and compounds found across organisms, one can then select a particular organism of interest and the reactions and compounds found in that organism will be highlighted on the pathway. Each of the reactions and compounds found in KEGG are given a unique identifier, either starting with "R" for reaction or "C" for compound followed by five digits. KEGG is continually expanding, in April 2010, KEGG contained 107,071 pathways generated from 356 reference pathways, including pathways not related to metabolism, and information on 129 eukaryotes, 1005 bacteria and 79 archaea.

KEGG also hosts databases which provide further information about the enzymes and compounds, along with cross linking to other databases via GenomeNet [55]. All of the data found in KEGG is free for academic users and can be downloaded via FTP in different flat file formats or accessed through web services. The reference metabolic pathways are also provided in KEGG Markup Language (KGML), an XML-based format created by KEGG. More recently, KEGG PATHWAY has been expanding to include other biological pathways such as those involved in genetic information processing, environmental information processing, cellular processes, human diseases and drug development. However, the metabolic pathway maps still remain the most popular feature [56].

MetaCyc is also an independent metabolic pathway database which contains manually curated information from scientific literature. MetaCyc was first released in 1999, in conjunction with EcoCyc, a database containing pathway and genome information about *Escherichia coli*. In April 2010, MetaCyc consisted of more than 1,470 metabolic pathways, across more than 1,800 organisms. It also contains information about the compounds and enzymes in these pathways and takes great care to provide the primary literature sources that the data was obtained from. The MetaCyc data is freely available to all users in different file formats, including BioPAX and SBML. There is also a BioCyc software package, free for academic usage, called Pathway Tools which provides analysis and visualization capabilities.

MetaCyc and KEGG are highly similar resources in that their overall goal is to store and organize metabolic pathway data. However, there are some important philosophical differences in how they accomplish this task. The main being that the MetaCyc pathways attempt to capture a metabolic pathway that functions in an individual organism while the KEGG pathways are the union of pathways that can be found in multiple organisms. This means that the KEGG pathways are typically larger than the corresponding MetaCyc pathways. This is highlighted by comparing the lysine biosynthesis pathways as represented

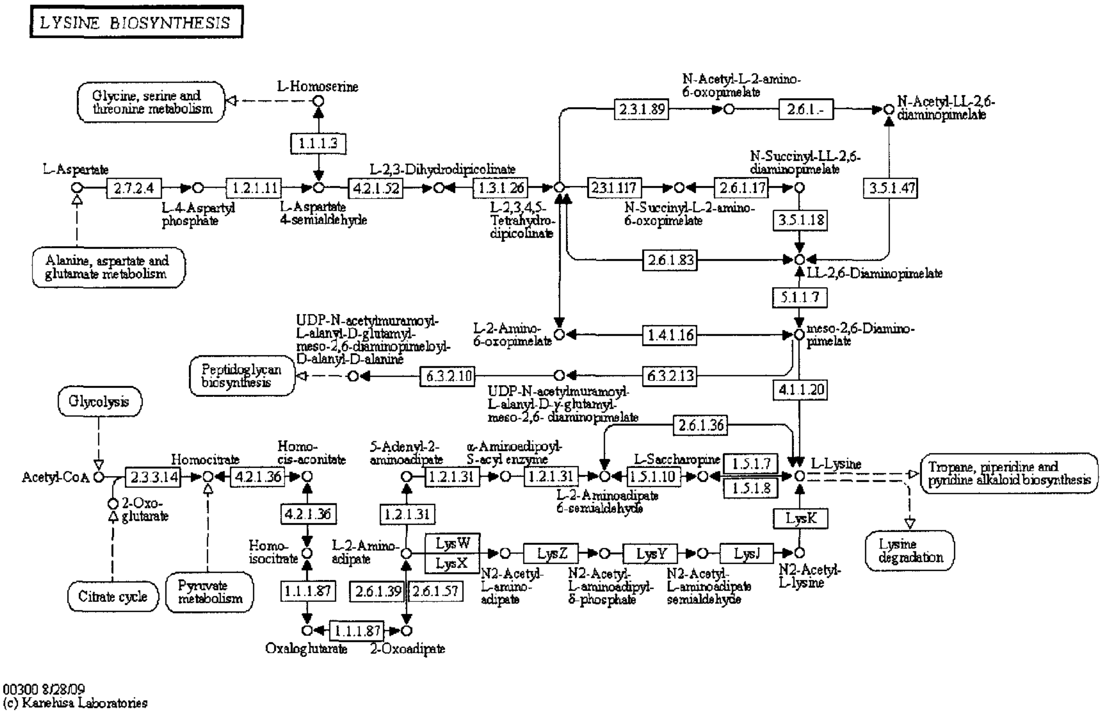

LYSINE BIOSYNTHESIS
Glycine, serine and threonine metabolism
L-Homoserine
1.1.1.3
L-Aspartate
2.7.2.4
L-4-Aspartyl phosphate
1.2.1.11
L-Aspartate 4-semialdehyde
4.2.1.52
L-2,3-Dihydrodipicolinate
1.3.1.26
L-2,3,4,5-Tetrahydro-dipicolinate
Alanine, aspartate and glutamate metabolism
2.3.1.89
N-Acetyl-L-2-amino-6-oxopimelate
2.6.1.-
N-Acetyl-LL-2,6-diaminopimelate
2.3.1.117
N-Succinyl-L-2-amino-6-oxopimelate
2.6.1.17
N-Succinyl-LL-2,6-diaminopimelate
3.5.1.47
3.5.1.18
2.6.1.83
LL-2,6-Diaminopimelate
5.1.1.7
L-2-Amino-6-oxopimelate
1.4.1.16
meso-2,6-Diamino-pimelate
UDP-N-acetylmuramoyl-L-alanyl-D-glutamyl-meso-2,6-diaminopimeloyl-D-alanyl-D-alanine
Peptidoglycan biosynthesis
6.3.2.10
6.3.2.13
UDP-N-acetylmuramoyl-L-alanyl-D-γ-glutamyl-meso-2,6- diaminopimelate
4.1.1.20
2.6.1.36
Glycolysis
Acetyl-CoA
2.3.3.14
Homocitrate
4.2.1.36
Homo-cis-aconitate
2-Oxo-glutarate
4.2.1.36
Homo-isocitrate
1.1.1.87
Citrate cycle
Pyruvate metabolism
5-Adenyl-2-aminoadipate
1.2.1.31
α-Aminoadipoyl-S-acyl enzyme
1.2.1.31
1.2.1.31
L-2-Aminoadipate 6-semialdehyde
L-Saccharopine
1.5.1.10
1.5.1.7
1.5.1.8
L-Lysine
Tropane, piperidine and pyridine alkaloid biosynthesis
Lysine degradation
LysK
L-2-Amino-adipate
LysW
LysX
LysZ
LysY
LysJ
N2-Acetyl-L-amino-adipate
N2-Acetyl-L-aminoadipyl-δ-phosphate
N2-Acetyl-L-aminoadipate semialdehyde
N2-Acetyl-L-lysine
2.6.1.39
2.6.1.57
1.1.1.87
Oxaloglutarate
2-Oxoadipate
00300 8/28/09
(c) Kanehisa Laboratories

Figure 2.2 : KEGG's reference pathway for lysine biosynthesis. The small circles represent individual chemical compounds and the rectangles represent enzymes and are labeled with EC numbers. The solid directed edges connect the substrate and products to their enzymatic reactions. The rounded ovals are other metabolic pathways and the dotted edges connect chemical compounds to other metabolic pathways.

by KEGG in Figure 2.2 and MetaCyc in Figure 2.3. MetaCyc also provides the user with several different layers of detail to view each pathway, as shown in Figure 2.3 while there is only one level of detail provided for the KEGG pathways. On both the KEGG and MetaCyc websites each individual element is clickable and takes the user to a page with more information about that element.

Additionally, from the beginning MetaCyc has put heavy emphasis on providing the primary literature sources coupled with descriptions of each pathway that often discuss the variants of the pathway in different organisms. In contrast, KEGG has only recently begun

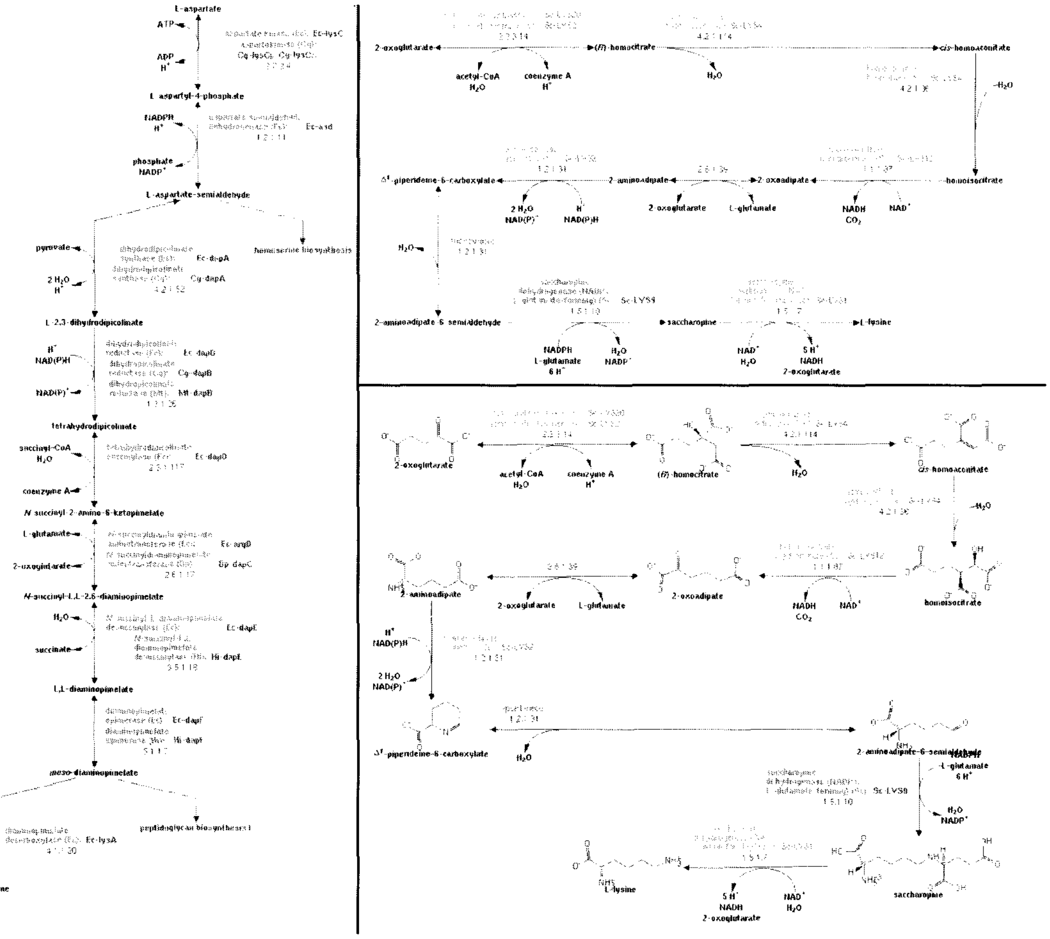


Figure 2.3 : Three MetaCyc pathways for lysine biosynthesis. The pathway on the left is lysine biosynthesis I and the one in the top right is lysine biosynthesis IV. Both of these pathways contain just one pathway found in the corresponding KEGG pathway in Figure 2.2. MetaCyc also offers several levels of detail to display, the pathway in the top right and bottom right are both lysine biosynthesis IV displayed with two different levels of detail. The chemical compounds are connected by directed edges which are labeled with the enzyme's name and EC number which catalyzes the reactions.

to provide literature sources and typically provides fewer for a given pathway than MetaCyc. KEGG has begun to include one paragraph pathway descriptions for their pathways, but this is currently only available for a few of the central pathways such as glycolysis and citrate cycle and just gives a brief overview of the chemical reactions.

The differences between KEGG and MetaCyc means that determining which one is better depends on the specific application in mind. MetaCyc takes a more organism specific role while KEGG takes a broader view of metabolism. Both resources are valuable for researchers interested in metabolism and in the ideal case the data from both sources could be integrated together. Due to the independent nature of the databases, data integration is non-trivial, but MetaCyc has made an effort to cross link to KEGG in order to make data integration easier. In version 13.6 of MetaCyc 4292 out of 8334 compounds and 3267 out of 8540 reactions had corresponding KEGG IDs. While this is a good start, integrating the data from the two sources is still no easy task and becomes even more difficult when considering different data sources such as enzyme databases.

Enzyme databases, such as such as BRENDA (BRaunschweig ENzyme DAtabase) [22] and ExPASy-ENZYME [11], are also an important source of metabolic data that can be used for metabolic path finding. These databases differ from metabolic pathway databases such as KEGG and MetaCyc as they focus on data on specific enzymes and do not organize the reactions in pathways. Enzyme databases provide more detailed information about each reaction, such as functional and kinetic parameters, that may be important in understanding the feasibility and dynamics of a metabolic pathway.

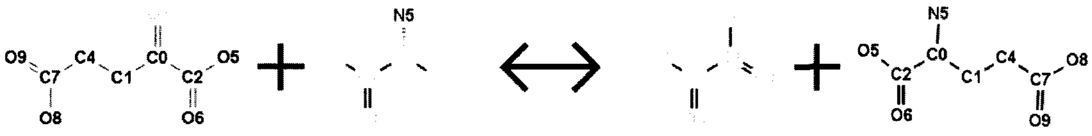
BRENDA was originally created in 1987 and has seen steady growth and improvements since its beginning. BRENDA is manually curated and currently contains data for over 5,000 enzymes acquired from more than 79,000 primary literature references [22]. BRENDA is also supplemented by data from FRENDA and AMENDA which contain data from automated text-mining of the literature. The first release of ExPASy-ENZYME was in 1993 and it has also been greatly expanded and regularly updated through the years. It is primarily concerned with storing information on the nomenclature of enzymes and so it does not contain as much information as BRENDA does about the properties of each enzyme. However, it does provide cross links to a number of other databases and is part of the broader ExPASy (EXpert Protein Analysis SYstem) server [40] and could play an important role in data integration.

One of the major issues with integrating the enzyme data with metabolic pathway data is that since the enzyme databases are not as concerned with the relationships between the reactions, the chemical compounds are not given unique identifiers. Rather, they use human-friendly names for the substrates and products of the enzyme and even within the database these names may not be consistent. This makes it a challenge to map these names to specific compounds found in databases like KEGG and MetaCyc.

The focus of this thesis is the development of algorithms for metabolic path finding, not data integration, and therefore in the current work all of the data used comes from KEGG. Additionally, as described in Section 2.1.1, KEGG is the only database that contains corresponding atom mapping data which is required for the algorithms in this work. Integrating other data sources is an important direction for future work and is further discussed in the conclusions chapter.

2.1.1 Atom Mapping Data

Even as metabolic databases have greatly increased the number of reactions and compounds they contain, they have traditionally lagged behind in providing atom mapping data. Atom mapping data provides a mapping between each atom in the substrate compounds of a reaction to the corresponding atom in the product compounds of the reactions. An illustration



2-Oxoglutarate C00026 + L-Alanine C00041 ⟷ Pyruvate C00022 + L-Glutamate C00025

Figure 2.4 : Atom mapping data obtained from KEGG for the reaction catalyzed by alanine transaminase (KEGG ID: R00258, EC Number: 2.6.1.2). Underneath each compound is its common name and its KEGG ID. The colors together with the atom labels indicate which atoms in the substrates map to which atoms in the products. For example, the carbons in alanine, C1, C2 and C3, become C1, C2 and C3 in pyruvate and the carbons in 2-oxoglutarate, C0, C1, C2, C3, C5 and C7, become C0, C1, C2, C3, C5 and C7 in L-glutamate. No hydrogens are illustrated as they are not included in the atom mapping data.

of the reaction catalyzed by alanine transaminase (KEGG ID: R00258, EC Number: 2.6.1.2) using atom mapping data from KEGG can be found in Figure 2.4. The atomic level detail of reactions provided by atom mapping data enables explicit tracking of atoms through metabolic networks. The ability to track atoms through metabolic networks has a number of applications in understanding and manipulating metabolism, such as metabolic path finding and radioisotope-tracer experiments [48, 16, 6, 5]. The algorithms presented in this thesis are based upon atom mapping data, which has recently become increasingly available.

The large scale availability of atom mapping data is a result of coupling computational methods with manual curation. The problem of finding atom mappings can be formalized by representing the chemical structure of the compounds as a graph and attempt to find a set of cuts that produce isomorphic components between the substrate and product compounds of the reactions. In the general case, finding a set of cuts of size C is NP-hard [1]. However, since chemical compounds are graphs of bounded degree, if the number of substrate compounds, p, product compounds, q, and C is held constant then the problem becomes $O((n^C)^{(p+q)})$ [1]. While this does cover a number of reactions, it is difficult to tell without manual intervention if this is the case for a particular reaction. Therefore, a common approach is to use heuristic or approximate algorithms to iteratively find maximum common substructures (MCS), an NP-hard problem [39], between the substrate and product compounds and then manually fix errors. This approach typically requires an extensive amount of manual curation and was used to build the ARM database which contains data for carbon, phosphorus and nitrogen atoms for about 3000 reactions [7]. A similar approach was taken to build the KEGG RPAIR database, arguably the largest source of atom mapping data currently available [57, 62].

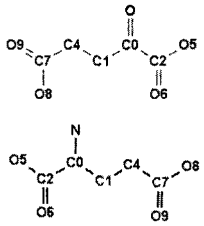
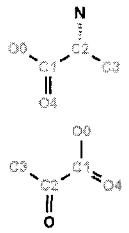
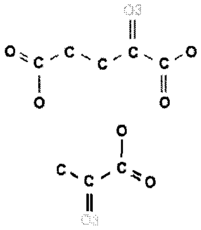
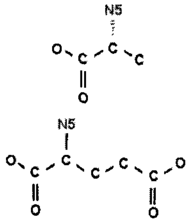
The experiments in this thesis use data acquired from the KEGG RPAIR database [57, 62]. The KEGG RPAIR database was built by using a combination of manual curation and the computational method SIMCOMP [45, 46]. SIMCOMP differs from the computational methods used to create the ARM database in that it does not take into account a whole reaction, but rather only finds the MCS between two compounds. The RPAIR entries are created by manually examining reactions in KEGG and splitting them into substrate-product compounds pairs. These pairs are then used as input to SIMCOMP and the output of SIMCOMP is then checked manually and corrected if there are errors. The corrected atom mapping of the compound pairs become the RPAIR entries and are assigned a unique identifier starting with "RP" followed by five digits.

Since the RPAIR entries only contain atom mapping between a pair of compounds, the atom mapping for most reactions is determined using several RPAIR entries. For example, the atom mapping for the reaction in Figure 2.4 is determined by using four different RPAIR entries corresponding to the four different colors. Table 2.1 contains the atom mapping data obtained from the four different RPAIRs combined together to determine the overall atom mapping data for the entire reaction. In turn, each RPAIR entry may be associated with multiple reactions. For example, in the RPAIR entries downloaded from KEGG in February 2010, the RPAIR entry between 2-oxoglutarate and L-Glutamate (RP000014) was associated with 69 reactions. In the same data, there were 12,079 RPAIR entries involved in 7525 reactions out of a total 8098 reactions stored in KEGG. While the KEGG RPAIR database has a high level of coverage of the reactions in KEGG, there has been interest in developing better computational tools for automatically identifying atom mapping in order to reduce the high level of manual curation currently required.

Computationally solving the atom mapping problem has two major issues, one being the complexity, as discussed previously in this section [1]. The other is that the formal definition of the atom mapping problem involves finding a set of cuts of size C, but it does not provide an optimization criteria. Typically, this is to minimize C, but this is a heuristic that does not always correspond to the actual chemistry. Additionally, for the minimal C there frequently several solutions and there needs to be criteria to choose between them. One solution, proposed by Blum and Kohlbacher, is to cluster reactions by their EC number and identify the frequency of different chemical transformations in each cluster [16]. When multiple atom mappings exist for the minimal C the one that corresponds to the most common chemical transformation in the reaction's EC cluster is chosen.

Other approaches strive to minimize the number of bonds broken and formed in the reaction. Crabtree and Mehta provided five algorithms to find atom mappings based upon this criteria, one which is exhaustive and the other four based upon different heuristics [25]. They demonstrated that the best performing of their heuristic algorithms, evaluated on several benchmarks, finds the optimal solution 84-91% of the time. A similar approach by Heinonen et al. uses an A* based algorithm that finds the atom mapping that minimizes the number of broken and formed bonds [48]. Additionally, their algorithm can incorporate weights on the bonds, which can be used to represent chemical criteria such as the energy

Table 2.1 : The four RPAIR entries needed to obtain the atom mapping for the reaction in Figure 2.4. The chemical drawing depicts the two compounds in the RPAIR; the atoms mapped in each RPAIR entry are colored and labeled the same as in Figure 2.4; the other atoms are black and are not labeled.

RPAIR ID	Substrate ID	Atom ID	Product ID	Atom ID	Chemical Drawing
RP00014	C00026	C0	C00025	C0	 O O9 C4 C0 O5 C7 C1 C2 O8 O6 N O5 C0 C4 O8 C2 C1 C7 O6 O9
	C00026	C1	C00025	C1	
	C00026	C2	C00025	C2	
	C00026	C4	C00025	C4	
	C00026	O5	C00025	O5	
	C00026	O6	C00025	O6	
	C00026	C7	C00025	C7	
	C00026	O8	C00025	O8	
RP00021	C00041	O0	C00022	O0	 N O0 C2 C1 C3 O4 O0 C3 C1 C2 O4 O
	C00041	C1	C00022	C1	
	C00041	C2	C00022	C2	
	C00041	C3	C00022	C3	
	C00041	O4	C00022	O4	
RP05853	C00026	O3	C00022	O3	
RP08469	C00041	N5	C00025	N5	

required to break or form a particular bond. This recent work on computational methods for improving the automated determination of atom mappings bodes well for the increasing availability of atom mapping data.

2.2 Stoichiometric Approaches for Metabolic Pathfinding

The first approaches for computationally finding metabolic pathways were based upon stoichiometric models of metabolic networks [92]. These approaches typically focus on one organism or system, and require the user to define which compounds are present or available to the cell. Stoichiometric models utilize the steady-state assumption and therefore require explicit labeling of internal and external compounds. The internal compounds are balanced, that is their creation and consumption are equivalent while the external compounds are free to be created or consumed as required. This can be a disadvantage as there are feasible biochemical pathways that do not obey the steady-state assumption and/or a compound that labeled as internal could easily be provided as an external compound [92]. The algorithms found in this thesis strive to find interesting metabolic pathways that may span multiple organisms without information such as what organisms or what compounds exist in the environment. These algorithms should be perceived as complementary to stoichiometric methods; in the future the paths found by the path finding algorithms may be incorporated with stoichiometric models in order to further understand how new paths may behave when introduced into an organism. However, stoichiometric approaches are widely used and it is relevant to understand the differences between stoichiometric and graph based approaches for metabolic pathfinding.

In 1986, Seressiotis and Bailey published the first method, Metabolic Pathway Synthesis (MPS) for finding pathways that convert one chemical compound into another in metabolic networks [107, 108]. They also introduced the concept of *genetically independent* pathways.

A pathway is genetically independent if no subset of the pathway's reactions is also a pathway from the same source and target compound. MPS takes in a source compound, a target compound and a metabolic network and then performs an exhaustive search to find all stoichiometric balanced, genetically independent pathways. MPS was tested on a network containing about 90 reactions and 120 compounds. In the early 1990s, Mavrovouniotis et al. presented a branch and bound algorithm for finding genetically independent pathways that satisfy stoichiometric constraints [77, 76]. This algorithm allowed for specifying multiple compounds as sources and targets and operates by iteratively satisfying constraints. It was tested on a network containing about 250 reactions and 400 compounds.

MPS and Mavrovouniotis' algorithm are both exhaustive algorithms that are exponential in the worst case. They were tested on networks that are relatively small compared to metabolic networks of interest today that contain thousands to tens of thousands of reactions and compounds. While the criteria of genetic independence and the structure of metabolic networks helps limit the scope of the search, the exhaustive search of both algorithms becomes too computationally expensive to apply to these larger networks. Additionally, it may be of interest to identify non-genetically independent pathways because the additional reactions may have other implications such as energy production or regulation.

Stoichiometric models subsequently became a popular way to computationally study metabolic networks. Flux balance analysis (FBA) emerged as a powerful, experimentally validated, and widely used way to analyze the fluxes through metabolic networks [121, 32, 96, 86]. It differs from path finding approaches in that it's primary goal is to provide a global view of the fluxes through a metabolic network. FBA sets up the stoichiometric constraints as a set of linear equations and then creates a heuristic optimization function based upon the idea that the cell is trying to optimize its growth. Solving this linear programming problem results in a set of fluxes for each reaction which can then be analyzed to find high flux pathways or compared to the fluxes that result after a reaction is added or removed. Elementary flux modes (EFMs) are a related approach that is directed more towards finding "fundamental" or "ideal" pathways in metabolic networks [104, 102, 103]. An EFM is a minimal, or non-decomposable, set of reactions that operates at steady state, while also respecting any irreversible reactions. The set of EFMs is unique and finite for a metabolic network, however the number of EFMs increases combinatorially in relation to the size of the network [60]. Therefore, while there is evidence that EFMs can be useful for identifying metabolic pathways, it is currently limited to analysis of relatively small networks.

Other stoichiometric models have been geared more towards identifying new pathways for metabolic engineering applications. One such example is OptStrain, whose goal is to identify the minimum number of reactions to add to a host organism to produce a desired compound while also maximizing the theoretical yield of the compound [90]. This is done by first creating a "super" metabolic network, which in the study contained about 4800 compounds and 5700 reactions. This network is then used to calculate the theoretical maximum yield of the desired product. The reactions are labeled as native or non-native based upon whether they occur in the host organism and the problem is formulated as a mixed integer linear program where the number of non-native reactions are minimized while still maintaining the theoretical maximum yield. Several computational results were presented on hydrogen production in three different microbes as well as the production of vanillin from glucose in *E. coli.* OptStrain is the most closely related of the pure stoichiometric models to the algorithms found in this thesis, as its goal is to find new metabolic pathways that can be utilized for metabolic engineering purposes. However, it still is hobbled by the disadvantages that face all stoichiometric models in that it does not allow for unbalanced pathways.

Petri nets lie on the boundary of stoichiometric and graph-based path finding methods because of their flexibility in specifying firing rules [23, 61, 133, 63]. Petri nets are an attractive modeling tool for metabolic networks because of their rich theoretical foundations [82] and the straightforward analogy of compounds as tokens and reactions as transitions. The firing rules of petri nets are typically based upon stoichiometric constraints that are highly similar to stoichiometric models, and a number of concepts in petri net analysis correspond to independently developed stoichiometric analysis methods. For example, the T-invariant of such a petri net is the same as the previously discussed elementary modes [105]. However, unlike stoichiometric models, petri net firing rules can be relaxed so that they do not follow the stoichiometric constraints, thus making them more similar to graph-based approaches. Work by Kuffner et al. demonstrates how the pathways found in petri nets vary depending on the constraints used [63]. They constructed a large petri net containing data from KEGG, MetaCyc and BRENDA. Results were reported for pathways from glucose to pyruvate that were constrained to having a maximum length of nine. They found over 500,000 unrestricted pathways in the network, when stoichiometric constraints were applied this was reduced to about 80,000. They also investigated restricting the cut width of the pathways to 2 and 1, which resulted in 541 and 170 pathways respectively. Their results demonstrated the large number of alternative pathways in multi-genome scale metabolic networks.

Stoichiometric models are adept at analyzing the metabolic networks under steady state conditions and this analysis can be used to identify pathways in metabolic networks. However, graph-based approaches arguably provide a more natural framework for directly identifying pathways in metabolic networks, especially in large networks and in non-steady state conditions.

2.3 Graph-Based Approaches for Metabolic Pathfinding

Graphs provide a solid framework, backed with a rich theoretical foundation, for modeling objects and the relationships between them. More formally, a graph is a pair (V, E) where V is a finite set, called the vertex set, and E, called the edge set, is a binary relation on V. In particular, finding paths in a graph is a well defined problem that a number of efficient algorithms have been developed to solve. Therefore, the problem of metabolic path finding becomes finding a correct way to associate compounds, reactions and their relationships to V and E in an appropriate graph model and applying the proper path finding algorithm.

The most commonly used graph models for representing metabolic networks are compound graphs, reactions graphs, bipartite graphs and hypergraphs [65, 29]. Examples for each of these graph models are depicted in Figure 2.5. The graphs used are typically directed graphs in order to take the direction of reactions into account, but may be undirected if the direction of reactions is unimportant or unknown. In a compound graph, the vertices correspond to compounds and an edge is drawn from one compound to another if one compound is a substrate and the other compound is a product in a reaction. The reaction graph is a similar concept except the vertices correspond to reactions and an edge is drawn from one reaction to another if a product of one reaction is used as the substrate as the other reaction. Compound and reaction graphs have been used in analysis related to the topological properties of metabolic networks [122, 73, 51]. However, they are limited as they only contain information about compounds or reactions and therefore can also be ambiguous. That is, two different set of metabolic reactions can create the same reaction or compound graph [29]. For example, the graph in Figure 2.5(b) could represent two different reactions, one being C1 → C2 and the other being C1 → C3 instead of R1 as found in Figure 2.5(a). One way to eliminate the ambiguity is to add edge labels based upon the reactions, but a more popular approach is to use a bipartite graph to represent the network.

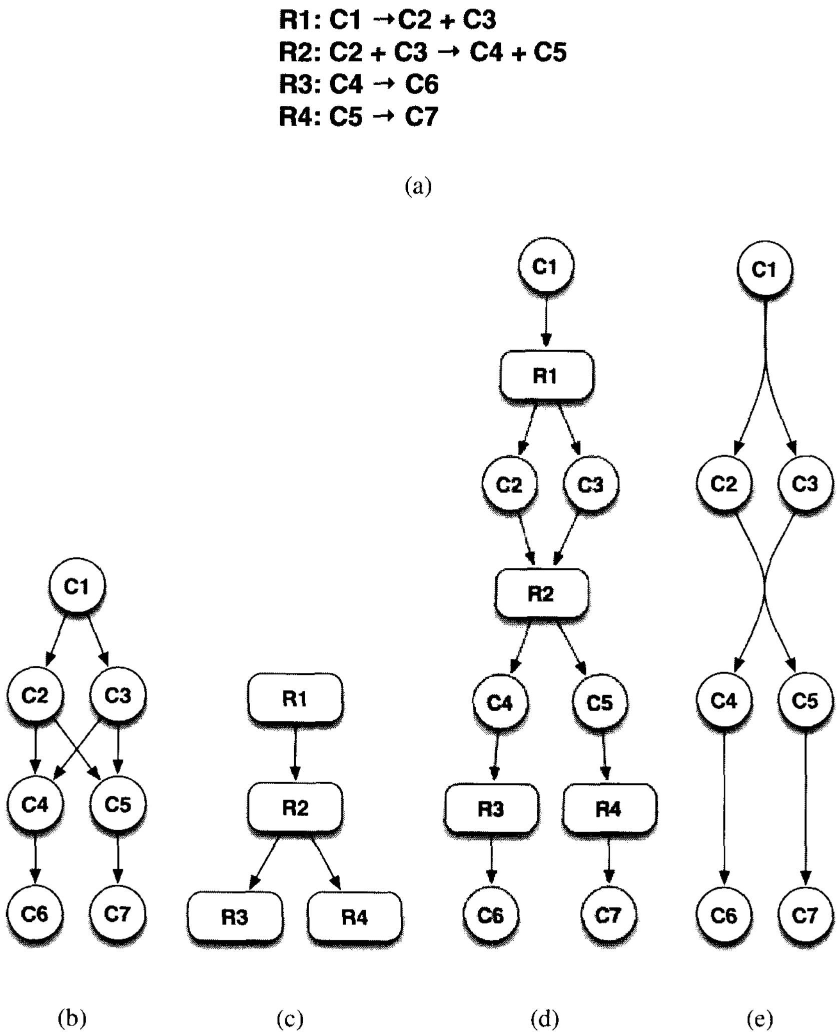


Figure 2.5 : The generic metabolic reactions, R1, R2, R3 and R4 using the generic compounds C1, C2, C3, C4, C5, C6 and C7 in (a) represented using four different graph models: (b) compound graph (c) reaction graph (d) bipartite graph (e) hypergraph.

In a bipartite graph representation, there are vertices representing reactions and vertices representing compounds. A bipartite graph is one where V can be partitioned into two disjoint sets, V_1 and V_2, such that each edge has one node in V_1 and the other in V_2. When modeling a metabolic network as a bipartite graph, the two disjoint sets correspond to the set of compounds and the set of reactions. There are no edges in the graph from compound to compound or reaction to reaction. Edges are drawn from substrate compounds to the reaction they participate in and from the reaction to the resulting product compounds. This means the bipartite graph is not ambiguous, as is the case with compound or reaction graphs. Hypergraphs provide another unambiguous model of metabolic networks. A hypergraph is still a pair (V,E), but now the edge set is no longer restricted to connecting two individual vertices. Instead, it contains hyperedges that can connect two arbitrary subsets of vertices. For modeling metabolic networks, the vertices of the hypergraph correspond to chemical compounds and the hyperedges correspond to the reactions. A hypergraph representation of metabolic network is equivalent to the bipartite representation, as a hyperedge corresponds to a reaction vertex in the bipartite graph [29].

While graph models have a number of advantages for finding pathways in metabolic networks, the main disadvantage is that they can contain spurious connections through currency metabolites. Therefore, most of the research in the last decade has been on developing methods and heuristics to avoid these spurious connections. A discussion of these approaches follows in Section 2.3.1. Another disadvantage is that graphs do not typically model the splitting and joining of chemical compounds, which can result in branched metabolic pathways. The work in this thesis overcomes these disadvantages by explicitly tracking the atoms in the network which helps avoid spurious connections as well as enables finding branched pathways. According to the current literature, there is only one other graph based method, discussed in Section 2.3.2, that is able to find branched pathways.

2.3.1 Finding Linear Metabolic Pathways

The early 2000's saw a rise of interest in graph-based analysis of biological networks. Most of the work in this field has focused on identifying linear pathways, as they correspond to paths, as defined by graph theory, in graph representations of metabolic networks. One of the seminal works focused on analyzing the topology of metabolic networks and brought the concepts of scale-free and small world to biological networks [52]. In their work, Jeong et al. modeled the metabolic networks of 43 organisms, from the now defunct WIT database, as directed bipartite graphs. The major conclusion of this work was that topology of the metabolic networks are not random but rather *scale-free* and *small-world* [52]. A scale-free network is one whose degree distribution is characterized by a power law of the form $P(k) \sim k^{-\gamma}$ [12]. A small-world network is one where any two vertices in the network can be connected by a relatively short path.

The scale-free property of metabolic networks means that there are a few highly connected compounds and most of the compounds have low degree. Coupled with the small-world property this means that a few compounds are responsible for the overall connectivity of the network. Jeong et al. found that irrespective of size, all of the metabolic networks they tested had highly similar diameters, between 3 and 4. A diameter of a network is defined as the average over all shortest paths between all pairs of vertices. They demonstrated that by removing the highly connected compounds one at a time from the network the diameter of the networks increases dramatically. These properties of networks are of further interest because a number of real-world networks, such as the Internet and citation networks, appear to have similar properties [3]. A number of other papers found that other biological networks, such as protein-protein interaction, transcription and signaling networks also are scale-free and small-world and there have been number of reviews written about the subject [2, 13, 3, 113].

The small diameter of metabolic networks was especially surprising as known biosynthetic and degradative metabolic pathways are typically much longer. Further investigation revealed that, in graph representations of metabolic networks, many of the shortest pathways may be biologically meaningless because they route through highly connected *currency* metabolites [73, 5, 6]. The term currency metabolites refers to compounds that take part in a large number of reactions, often as cofactors instead of part of the main chemical transformation, such as ATP and NADH. Therefore, while the structural analysis of the bipartite graph representation of metabolic networks provides a number of interesting insights, the actual paths found do not correspond to biochemical pathways for synthesizing or degrading compounds. Subsequent studies of metabolic networks attempted to correct these issues by either removing edges between current metabolites and reactions or the current metabolites themselves from the graphs [122, 98, 73, 41].

Wagner and Fell analyzed undirected compound and reaction graph representation of the metabolic network of *E. coli* [122]. They compared the properties of full graphs to graphs where ATP, ADP, NAD^+, NADH, $NADP^+$ and NADPH were removed. The mean degree and standard deviation of degree of the graphs dropped when the compounds were removed. They found that the degree distribution of the compound graph, with or without compounds removed, follows the power law distribution while the reaction graph did not. This highlights how different representations of the same metabolic network can have different structural properties. Additionally, they found that both graphs were small-world, with or without compounds removed, although the shortest paths in the graphs with compounds removed had slightly longer path lengths. In the work of Ma and Zeng, they analyzed the metabolic networks of 80 organisms, represented as directed compound graphs [73]. They manually examined about 3,000 reactions to determine whether the compounds involved were part of the main chemical transformation; if they were not, the corresponding edge was removed in the graph. This resulted in more than half of the edges being removed from the graphs, but they still found the small-world property held for these graphs.

The work of Arita was the first to challenge the notion of metabolic networks being small-world and introduced the idea that "the biochemical link between metabolites . . . depends on the conserved structural moieties in the adjacent reactions" [6]. Therefore, a valid metabolic pathway between two compounds is redefined as a sequence of reactions that conserves at least one carbon atom from the starting to the target compound. In order to automatically identify these pathways atom mapping data is required. Arita manually created a database of atom mapping for over 2,500 reactions contained in the KEGG database [6, 5]. Valid pathways were identified by running a k-shortest pathway algorithm until the shortest path that conserved at least one carbon atom was found. Using the length of these pathways, it was revealed that the small-world property no longer held [6]. This analysis showed the importance of not ignoring the biochemical underpinnings of metabolic networks and provided evidence that graph models need to be used with care in order to find valid metabolic pathways.

Biochemical intuition says that pathways which move a high percentage of atoms from start to finish compounds will be biologically relevant. Boyer and Viari proved that this problem is PSPACE-complete by formulating the problem as finding a maximal composition of partial injections [17]. They also state, but do not prove that when a compound can only be used once in a pathway, the problem is NP-complete. Despite the complexity, they were able to develop an algorithm that found linear atom conserving pathways efficiently in practice, given a number of atoms to conserve and a maximum pathway length. The algorithm by Boyder and Viari returns all pathways less than the maximum pathway length that conserve the given number of atoms. The algorithm was tested with atom mapping data obtained by using the SIMCOMP program on data from KEGG, resulting in a network containing 2920 compounds and 3721 reactions. They present two pathway results, one for chorismate to tryptophan, conserving 6 carbon atoms with a maximum pathway length of 6, and the other for -D-glucose 6-phosphate to phosphoenolpyruvate, conserving 9 heavy atoms with a maximum pathway length of 8. Both pathway results contain the known metabolic pathways, along with a several other pathways. In the case of tryptophan the other pathways are mainly catabolic pathways, which is expected as they considered all reactions as reversible. In the case of glycolysis, the other pathways included reactions from the pentose phosphate pathway. These results provide evidence that the intuition that atom conserving pathways are biologically relevant may be correct. However, in the next few years, work on path finding in metabolic networks mostly focused on finding realistic linear pathways without using atom tracking, perhaps due to the theoretical complexity and the perceived unavailability of atom mapping data. These approaches focused on finding realistic metabolic pathways with minimal manual intervention.

One approach was to use measures of structural or chemical similarity instead of explicit atom mapping to avoid avoid spurious connections when finding metabolic pathways [78, 95]. The PathMiner program uses a predefined set of 145 descriptors based on atom type and bond type [78]. Each compound is then represented by a vector of size 145 where each entry corresponds to the number of atoms or bond of each type that the compound contains. The cost of a transition from one compound to another is then calculated as the difference of the vectors for the two compounds. PathMiner uses the data from the manually curated KEGG pathway maps, which typically contain only the compounds involved in the main chemical transformation, resulting in 3890 compounds and 2917 transformations. PathMiner was tested on four different metabolic pathways using breadth-first and depth-first search. As expected, it was observed that the BFS finds the shortest path in terms of number of reactions and the DFS finds very long pathways. Additionally, an A* search was tested, using a cost function that was the sum of transition costs along the pathway. The A* approach showed promise, compared to BFS and DFS, but they note that due to the cost function, it is best used for pathways that contain small chemical modifications and may not perform as well on pathways that require the addition of large functional groups.

The Pathway Hunter Tool (PHT) takes a similar approach to PathMiner by creating a bit vector based upon the chemical structure of a compound [95]. This was done by using the "fingerprinting" algorithm contained in the Chemistry Development Kit [111, 112]. The similarity between two compounds is then computed by combining the atom mass value and the Tanimoto algorithm [130]. PHT then differs from PathMiner in that it automatically maps substrate compounds to product compounds, instead of relying on the manually curated KEGG pathway maps. This mapping is done by iteratively assigning substrate-product pairs that have the highest similarity. The substrate-product pairs are then used to determine valid connections during a search of the graph. PHT does not use the similarity scores in the search of the graph, but rather just returns the k-shortest paths based upon the number of reactions in the pathway. In the article, they show that the shortest pathway that PHT finds, using data obtained from KEGG, from beta-D-glucose to pyruvate is of nine reactions in length and corresponds to the known reactions of glycolysis. However, there are no guarantees that PHT will find the correct mappings and if multiple substrates and products have high similarity it is unclear how the mapping will behave. PHT is currently available via a web server located at http://pht.tu-bs.de/PHT/.

Another approach by Croes et al. avoided using chemical structures by focusing on the observation that currency metabolites have high degree in metabolic networks [26]. Therefore, they created a weighted, directed, bipartite graph where the compound vertices were assigned a weight equal to their degree. They built one graph from the KEGG database that contained 4,756 compounds and 5,985 reactions and another graph from the EcoCyc database containing 1,348 reactions and 1,329 compounds. The weighted graph is then searched to find the k paths, where k is a user input, with minimal total weight, which they term *lightest paths*. They compared the resulting lightest pathways to shortest paths in the unweighted graph as well as shortest pathways in an unweighted graph where 36 manually selected currency metabolites were removed. Surprisingly, this relatively simple weighting scheme performed quite well in finding known metabolic pathways.

Croes et al. introduced a quantitative performance measure for the known annotated pathways to the computed pathways; true positives (TP) are compounds found in both pathways; false negatives (FN) are compounds in the known pathway which are not in the computed pathway; false positives (FP) are compounds not in the known pathway which are in the computed pathway. Sensitivity (Sn) and positive predictive value (PPV) are used to calculate the accuracy (Ac); Sn = TP/(TP+FN), PPV = TP/(TP+FP) and accuracy Ac = (Sn+PPV)/2. They built a test set containing 56 pathways from the aMAZE database [70] and 104 pathways from the EcoCyc database [58]. In contrast, previous studies took more a qualitative approach to evaluating performance by just analyzing a handful of pathways. Using this measure, they found that the average accuracy for pathways from the unweighted graphs was less than 30%, filtering manually chosen currency metabolites increased the performance greatly to about 65% and the lightest paths performed the best with 85%. They additionally provided more in depth analysis of three metabolic pathways: heme biosynthesis, methionine/adenosyl-methionine biosynthesis and lysine biosynthesis. This analysis highlights that there is typically not just one correct metabolic pathway, especially when using a network constructed from reactions from multiple organisms, and that in the case of methionine and lysine biosynthesis feasible alternative pathways were found in the top five lightest pathways.

Follow up work to the work on degree weighting by Faust et al. coupled using the weighted graph with information obtained from the KEGG RPAIR database [35]. Faust et al. performed a qualitative analysis, based on the measures described in the previous paragraph, of the different combinations of parameters that can be used to construct metabolic graphs. They tested three types of graphs, the standard bipartite graph containing compound and reactions nodes, a bipartite *RPAIR graph* containing nodes corresponding to a RPAIR entry and nodes corresponding to compounds and a *reaction-specific RPAIR*, similar to the RPAIR graph, except it contains a node for each RPAIR/reaction combination. When all reactions or RPAIRs are considered reversible, the graph can be directed and contain two nodes representing each direction, or undirected and only containing one node for each reaction. The graphs could also be unfiltered, containing all compounds, or filtered where the 36 currency metabolites identified from the previous work were removed. The RPAIR graphs could also be filtered based upon the RPAIR class, one where it contains all of the RPAIRs, one where only RPAIRs classified as "main" are included and one where only "main" and "trans" RPAIRs are included. Finally, they tested different weighting schemes, the degree based weighting from the previous work and weights based on the RPAIR classes where the node weights were based on the RPAIR class: main=1; trans=5; cofac=10; ligase=15 and leave=20. It should be noted that in this work, they did not explicitly use the atom mapping data contained the KEGG RPAIR data, but rather used it as either a way of constructing or weighting the graphs.

Faust et al. then tested all feasible combinations of these parameters, 104 combinations in all, using a set of 55 annotated pathways from the aMAZE database and the accuracy score developed in their previous work. The top performing combination used the RPAIR graph and the degree based weighting scheme, resulting in an average accuracy of 83% across all pathways, 93% for *E. coli* pathways, 66% for *S. cerevisiae* pathways and 70% on human pathways. One of the most significant results was that the RPAIR graph always produced the most accurate pathways and the degree weighting always performed better than no weighting, regardless of the other parameters. They also provided a more detailed analysis of several pathways to highlight specific examples where the RPAIR annotation helps to remove spurious connections in the metabolic graph. The pathfinding tools used in the analysis are freely available through the Network Analysis Tools web server at http://rsat.ulb.ac.be/neat/ [19].

MetaRoute combined the idea of weighting by degree along with the idea that a valid biochemical path should transfer at least one atom, typically a carbon, from the start compound to the target compound [15]. MetaRoute uses a different weighting scheme, termed *combined weight*, that takes into account both the degree of the compounds along the pathway and the degree of the compounds along the pathway that participate in the pathway's reactions. The intuition is that compounds along the pathway should have low degree, as in the work of Croes et al., and the other compounds involved in the reaction, but not in the pathway, should have high degree so that they're more likely to be available currency metabolites. MetaRoute also utilizes computationally computed atom mappings from the work of Blum and Kohlbacher [16]. These atom mappings are used to create a graph structure where each reaction is broken down into substrate/product pairs (S_i, P_j) where at least one atom is conserved from the substrate to the product. Each substrate/product pair is then a node in the graph, edges are created in the graph between two nodes (S_i, P_j) and (S_k, P_l) if $P_j = S_k$ and at least one atom is conserved from S_i to S_k. Pathways are then found by applying Eppstein's k-shortest path algorithm to the graph using the combined weight. The k pathways are then filtered to only return pathways that conserve at least one atom from the start to the target compound. MetaRoute was tested on 137 pathways from *E. coli* and demonstrated improved performance over their previous work which used degree weighting instead of combined weighting. MetaRoute is freely available via a web server at http://www-bs.informatik.uni-tuebingen.de/Services/MetaRoute/.

The previous research done on using graph based methods for linear metabolic pathways has helped to shape the philosophy found in this thesis that it is important to explicitly track atoms to find biologically meaningful pathways. Atom tracking also enables the use of graph based methods to find branched metabolic pathways, a topic that has not been explored until the last few years.

2.3.2 Finding Branched Metabolic Pathways

Graph-based methods for finding metabolic pathways have focused on finding linear metabolic pathways because on the compound/reaction level it is unclear how to determine the relationships between the different pathways. However, the introduction of atom mapping data and atom tracking allows the algorithm to automatically identify where atoms are lost and gained along a linear pathway and thus this information can be used to build branched pathways. Branched pathways play an important role in metabolic networks as the atoms from the source compounds often go through separate reaction paths in order synthesize the final target compounds. Furthermore, branched pathways help highlight the relationships between different compounds and reactions that may not be as easily found by looking at a set of linear pathways. This may play an important role in the choice of metabolic pathways for applications such as metabolic engineering. An example of a branched pathway between sn-glycero-3-phosphocholine and L-threonine is depicted in Figure 2.6. The branched pathway contains two valid linear pathways, one through acetaldehyde and the other through glycine. However, only by realizing the two linear pathways interact biochemically can the full branched pathway utilized be revealed.

A recently introduced algorithm, ReTrace, was the first to take a graph based approach to find branched pathways [91]. Similar to the algorithms presented in this thesis, ReTrace

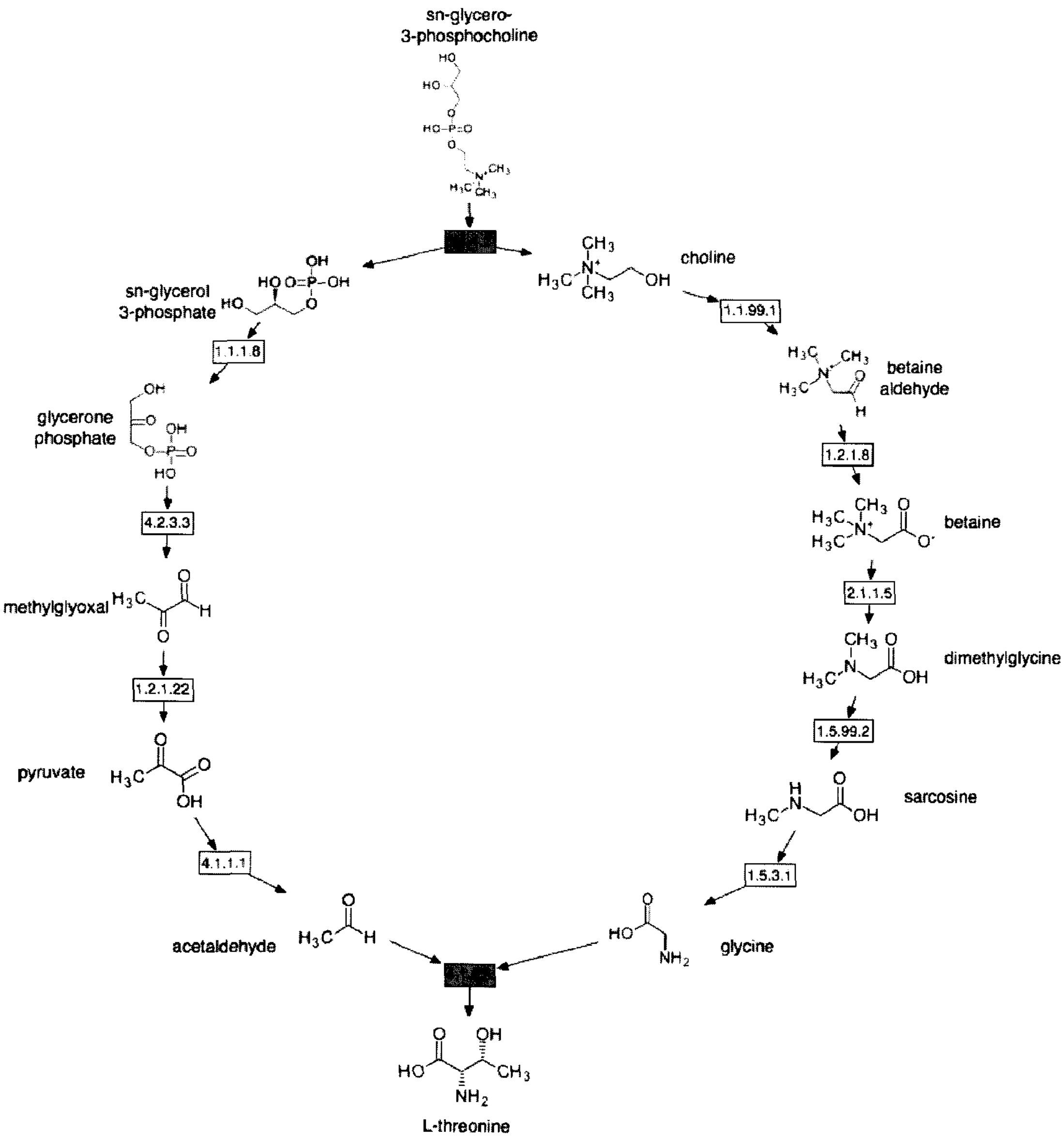


Figure 2.6 : A branched pathway between sn-glycero-3-phosphocholine and L-threonine, with the chemical structures of the compounds depicted and the reactions that function as branch points highlighted in magenta. This pathway corresponds to a result found in Figure 5.9.

achieves this by using atom mapping data to explicitly track atoms. In this formulation, the goal is to find a set of reactions that maximize the number of atoms that are conserved from start to target compound while minimizing the number of reactions. More formally, in the terminology of Pitkänen et al., $\mathscr{R}$ is the set of reactions, $\mathscr{A}$ is the set of atom positions for all compounds, an atom mapping is a bijective relation $\Gamma : \mathscr{R} \rightarrow \mathscr{A} \times \mathscr{A}$ describing which subtrate atoms become which product atoms. When finding pathways, a pathway is a subset of reaction $P = r_1, \ldots, r_k \subseteq \mathscr{R}$, the set of start atoms is S, the set of target atoms is T and Z_O is the fraction of atoms conserved by P from S to T. This gives rise to the main *Find-Branching-Pathways* problem which is definied by Pitkänen et al. as "Given a set of reactions, sets S, $T \subseteq \mathscr{A}$, $l \in \mathbb{Z}^+$ and $w \in \mathbb{R}$, find all pathways, $P \subseteq \mathscr{R}$ such that $Z_O(P, S, T) \geq w$ and $|P| < l$". Subsequently, they proved that *Find-Branching-Pathways* is NP-hard and developed ReTrace as a heuristic algorithm for finding branching pathways.

ReTrace is so named because it first identifies linear paths that conserve at least one atom from start to target atoms and then recursively attempts to find linear paths that can be combined that conserve more atoms, thus resulting in branched pathways. ReTrace was tested on a atom graph built from 11,265 KEGG RPAIR entries involved in 7,781 reactions. They provide a nice analysis of pairwise shortest distances when at least one atom was required to be conserved. Their results show that the distance distribution for carbon contains a long tail with a mean distance of 21.2, much longer than the mean distance of 8.4 found in previous work [6]. For branched pathways, they looked at 13 compounds and found pathways between all pairs of the compounds. Their results demonstrate that computation time and number of pathways found is highly dependent on the compounds specified as the start and target sets. They provide a more detailed analysis of the branched pathway found from Glucose to 5'-inosine monophosphate (IMP). ReTrace has also been successfully used to reconstruct the metabolic network of *Trichoderma reesei* [54].

The branched pathway algorithms in this thesis, developed independently, also augment linear pathways to find branched pathways that maximize the number of atoms conserved from the start to the target compound. Since because both methods use various heuristics and cutoffs to overcome the high complexity of finding branched pathways, the selection of which method to use may be dependent on the specific application or compounds being studied. One key difference is that the algorithms in this thesis explicitly use linear pathways that conserve at least a given number of atoms. In contrast, ReTrace finds pathways that conserve one atom and requires weighting heuristics to help find paths that conserve a larger number of atoms. As these methods are adopted, future work should analyze the practical performance of these methods to understand the affect of their heuristics and identify areas for improvement.

Chapter 3

Finding Linear Metabolic Pathways

This chapter presents a new algorithm for finding atom conserving linear pathways in metabolic networks, called LPAT for Linear Pathfinding with Atom Tracking. LPAT takes as input a start compound, a target compound, the minimum number of atoms to be conserved, the number of pathways to return, k, and a special data structure, termed an *atom mapping graph* (G_{am}). LPAT is then guaranteed to return the k shortest pathways between the start and target compounds that conserve at least a given number of atoms. Optionally, LPAT can utilize weighting schemes, for example those discussed in Section 2.3.1, that associate weights with compounds and reactions. If a weighting scheme is used, LPAT will return the k shortest pathways where the "length" of the path is equal to the sum of the weights of the nodes used in the pathway. LPAT is efficient enough that maximal atom conserving linear pathways can be found by starting with the number of atoms in the smaller of the start and target compounds and decrementing by one until pathways are found or a minimal number of atoms is reached.

Several general steps are used to find linear metabolic pathways. The first is to construct G_{am}, described in Section 3.1, which contains the atom mapping information in a minimalistic way and whose structure important for the efficiency of the algorithm. After the construction of G_{am}, the next series of steps are performed by LPAT, as described in Section 3.2.

3.1 Construction of G_{am}

A frequently used representation of metabolic networks is a directed graph where there are reaction nodes and compound nodes and edges are drawn between the compounds and the reactions they participate in. However, tracking atoms through this representation typically results in unreasonably high computational cost, because of compounds, such as cofactors, participating in a large number of reactions. Therefore, the *atom mapping graph*, G_{am}, is built using the observation that the same atom mapping pattern between two compounds often appears in multiple reactions [5]. For example, ATP to ADP occurs in many reactions, but the atom mapping remains the same between the two compounds. When searching G_{am}, only one node representing the ATP to ADP atom mapping needs to be explored. This is more efficient than explicitly exploring all of the reactions containing the atom mapping. In practice, this representation is important to help reduce the computational cost required to find atom conserving pathways.

G_{am} is a directed bipartite graph containing *compound nodes* and *mapping nodes*. Building G_{am} starts by providing a list of atom mappings between pairs of compounds and the associated reactions that utilize the atom mappings. For each compound, a compound node is added to G_{am} that has a unique identifier as well as a unique identifier for each of the atoms in the compound. A mapping node is added for each atom mapping entry and two directed edges are created, one from the first compound to the mapping node and one from the mapping node to the second compound. The mapping nodes contain atom mapping information, such that the atom identifiers from the substrate compound are associated with the atom indices in the product compound. There are can be multiple mapping nodes for the same pair of compounds. Reversibility is handled by adding another mapping node to enable the reverse direction; it has the same edges created but in the reverse direction. Typically, if no reversibility information is available then all atom mappings are considered reversible. If reversibility information is provided, it is incorporated into G_{am} by only allowing the corresponding edges to be added to the graph. Additionally, molecular symmetry can be handled by adding multiple mapping nodes that correspond to the different possible symmetries.

The mapping information contained in G_{am} allows for explicit tracking of an individual atom through the graph. Figure 3.1 shows a small subgraph of G_{am}. In this subgraph, a linear path from C00231 to C05345 through RP00080 and RP13340 conserves three carbon atoms. An important property of G_{am} is that all mapping nodes only have one input edge and one output edge connected to two different compound nodes. The compound nodes have the same number of outgoing and incoming edges equal to the number of atom mapping entries they participate in. Therefore, the degree of the nodes of G_{am} is less than the more traditional compound and reaction directed graph, which in turn contributes to the efficiency of the path finding methods.

3.2 LPAT: Linear Pathfinding with Atom Tracking

LPAT begins with a depth-first exploration of G_{am}. The exploration, described in Section 3.2.1, begins with the atoms of the start compound and finds all parts of the atom mapping graph that are reachable with at least the given number of minimal atoms. The result of the exploration is then used to build an auxiliary graph, described in Section 3.2.2, that has the property that all pathways from the start compound conserve the given number of minimal atoms. The last step is to run a standard k shortest path algorithms, the choice of which is discussed in Section 3.2.3, on the auxiliary graph and the resulting pathways correspond to the atom conserving linear pathways.

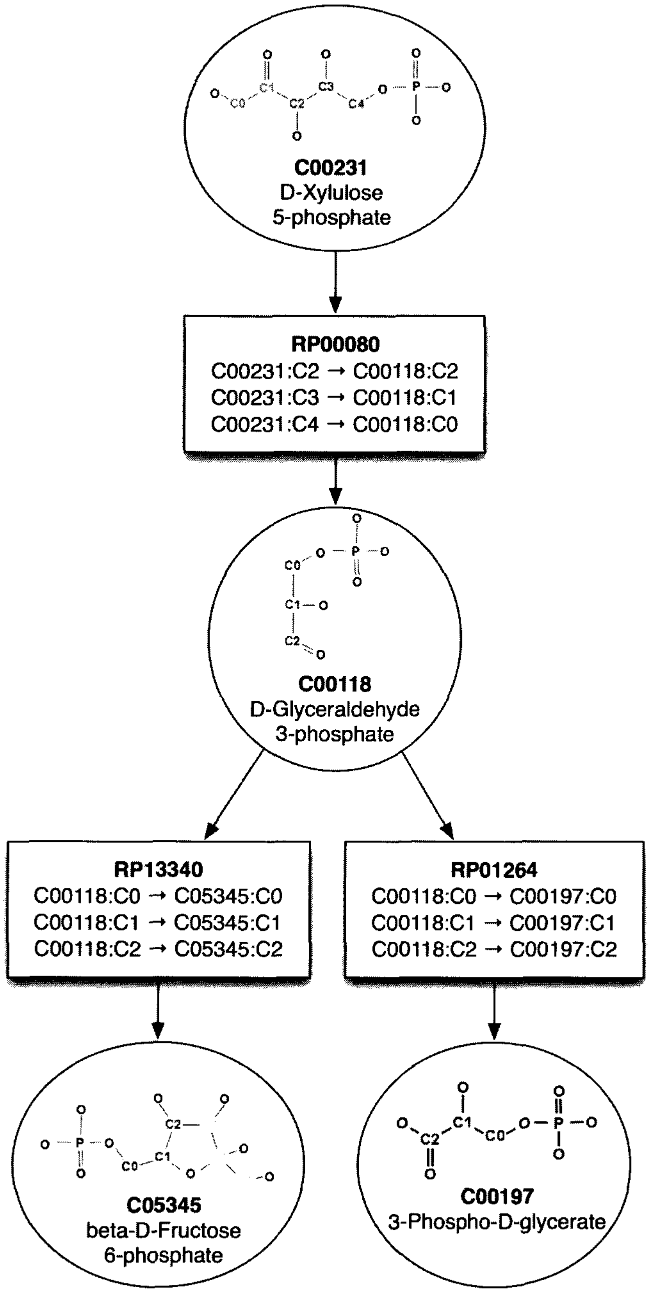


Figure 3.1 : An illustration of a small subgraph of G_{am} built from KEGG RPAIR data containing four compound nodes and three mapping nodes. The identifiers for each node are in bold, the compound nodes are also labeled with the common name of the compound and the mapping nodes contain the mapping between the carbon atoms. The KEGG RPAIR database contains information for all non-hydrogen atoms, but only carbons are depicted for clarity. The carbons colored in blue are mapped in the mapping nodes while the green and pink colored carbons are not in this subgraph.

3.2.1 Exploration of G_{am}

After the construction of G_{am}, LPAT begins with an exploration step that traverses G_{am} in a depth-first manner while explicitly tracking where each atom from the starting compound goes along the way. The exploration is a modified version of a standard depth-first search because the semantics of G_{am} are different. These semantics require that the exact set of atoms visited in each compound are recorded, and not just the fact that the node itself is visited as in traditional graph traversals. Therefore, a string containing the compound identifier along with an ordered list of atom identifiers is termed an *atom marking*. When the exploration moves through a mapping node, the input atom marking is used to compute the output atom marking based upon the mapping contained in the mapping node. The queue for the depth-first search then contains mapping nodes along with their input atom marking. For example, in Figure 3.1 the atom marking for all carbon atoms in D-xylulose 5-phosphate would be "C00231 C0,C1,C2,C3,C4", then taking the mapping node RP00080 would result in the atom marking "C00118 C0,C1,C2".

Since each possible atom marking is considered a different state during the exploration, each compound node can be visited $\sum_{k=m}^{n} \binom{n}{k}$ times where n is the number of atoms in the compound and m is the minimum number of atoms to conserve. Theoretically, this means there is the potential for a combinatorial explosion in the number of states to visit during an exploration of G_{am}. One standard way to reduce the amount of exploration time is to limit the depth of the search. However, in practice the cost of the exploration is dependent on a number of interelated factors such as the number of atoms in the start compound, the minimum number of atoms to conserve and the connectivity of the graph. In all of the linear pathway experiments for this thesis, a complete exploration of G_{am} was obtained in a reasonable amount of time and space without a limit on the depth.

The states visited during the exploration are recorded using an object containing the input atom marking, the identifier of the mapping node taken and the resulting output atom marking is termed a *transition history*. Using the atom markings and transition history, we can now introduce Atom Tracking Depth-First Search in Algorithm 3.2.1. This search starts from the starting compound and explores G_{am} to find all reachable states that conserve the given number of atoms. The result of the search is a list of generated transition histories, L, which is then used to build an auxiliary graph. An example of how Algorithm 3.2.1 operates on a generic G_{am} is illustrated in Figures 3.2 and 3.3.

3.2.2 Auxiliary Graph Construction

The resulting list of transition histories, L, from the previous exploration step of G_{am}, implicitly contains all of the linear atom conserving pathways. It is an implicit representation because the search backtracks when it reaches a previously explored state, instead of explicitly storing the path consisting of the new exploration path combined with the old exploration path. The purpose of the auxiliary graph is to combine all of the states explored such that the auxiliary graph can be used as input to standard k shortest path algorithms to obtain the final linear pathways. The auxiliary graph can almost be thought of as a filtered version of G_{am} which only contains paths from the start compound that conserve at least the given number of atoms. However, G_{am} can not be filtered directly by the exploration because of the atom tracking behavior. The atom tracking causes each compound node to take on several atom marking states which are not represented explicitly in G_{am}. Therefore, the auxiliary graph must be constructed separately from G_{am} so that it explicitly contains all of the atom marking states and mapping nodes found during the exploration.

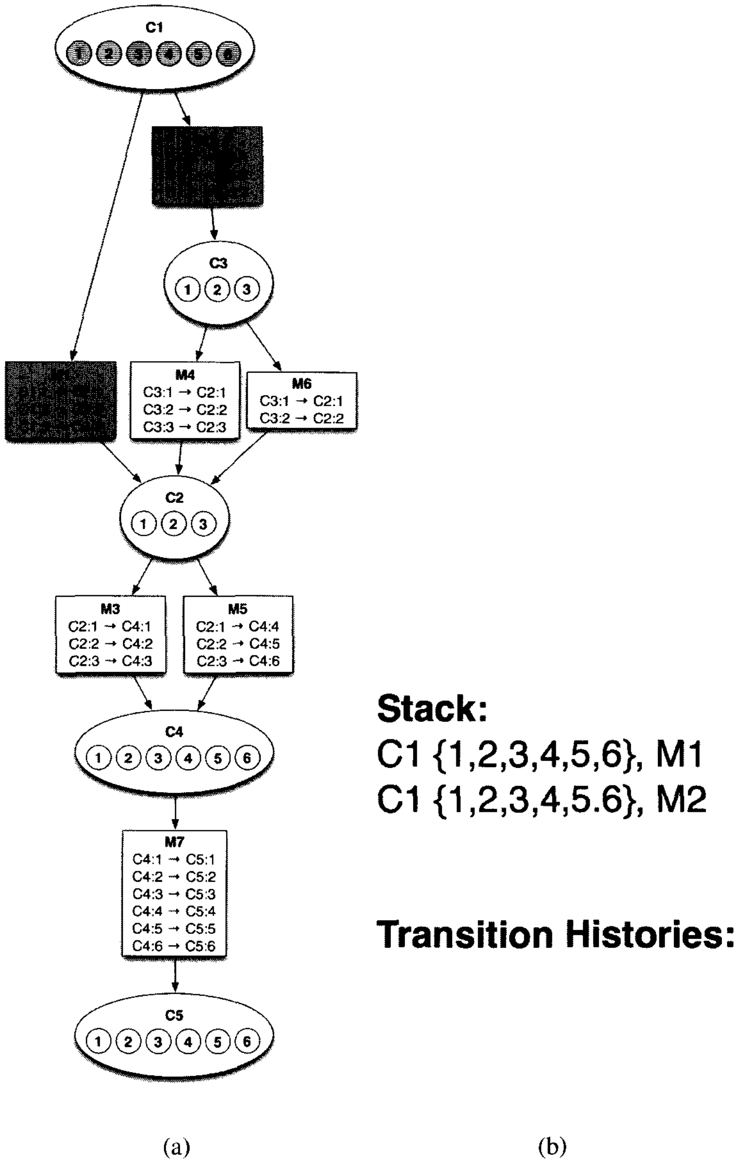
The construction of the auxiliary graph begins with the resulting list of transition histories, L, from the previous exploration step of G_{am}. The auxiliary graph contains a node labeled with each atom marking found in L. For each transition history, a new node containing the

Algorithm 3.2.1 Atom Tracking Depth-First Search (LPAT)

Input: Input Compound Atom Marking c_{in}, Minimum Number of Atoms to Conserve n, Atom Mapping Graph G_{am}

Output: List of Transition Histories L

```
$V \leftarrow \{c_{am}\}$ Set of visited atom markings
$S \leftarrow \{\}$ Stack of partial transition histories containing a mapping node and an input atom marking
$L \leftarrow \{\}$
for each successor mapping node $m$ from $c_{in}$ in $G_{am}$ do
    Add $\{m, c_{in}\}$ to $S$
while $S$ is not empty do
    Pop $s$ from $S$
    $c_{out} \leftarrow$ the output atom marking of traversing the mapping node $s_m$ of $s$ using the atom marking $s_{am}$ of $s$
    Add transition history $\{s_{am}, s_m, c_{out}\}$ to $L$
    if $c_{out}$ is not in $V$ then
        Add $c_{out}$ to $V$
        if $c_{out}$ contains $n$ or more atoms then
            for each successor mapping node $m$ from $c_{out}$ in $G_{am}$ do
                Push $\{m, c_{out}\}$ on to $S$
```



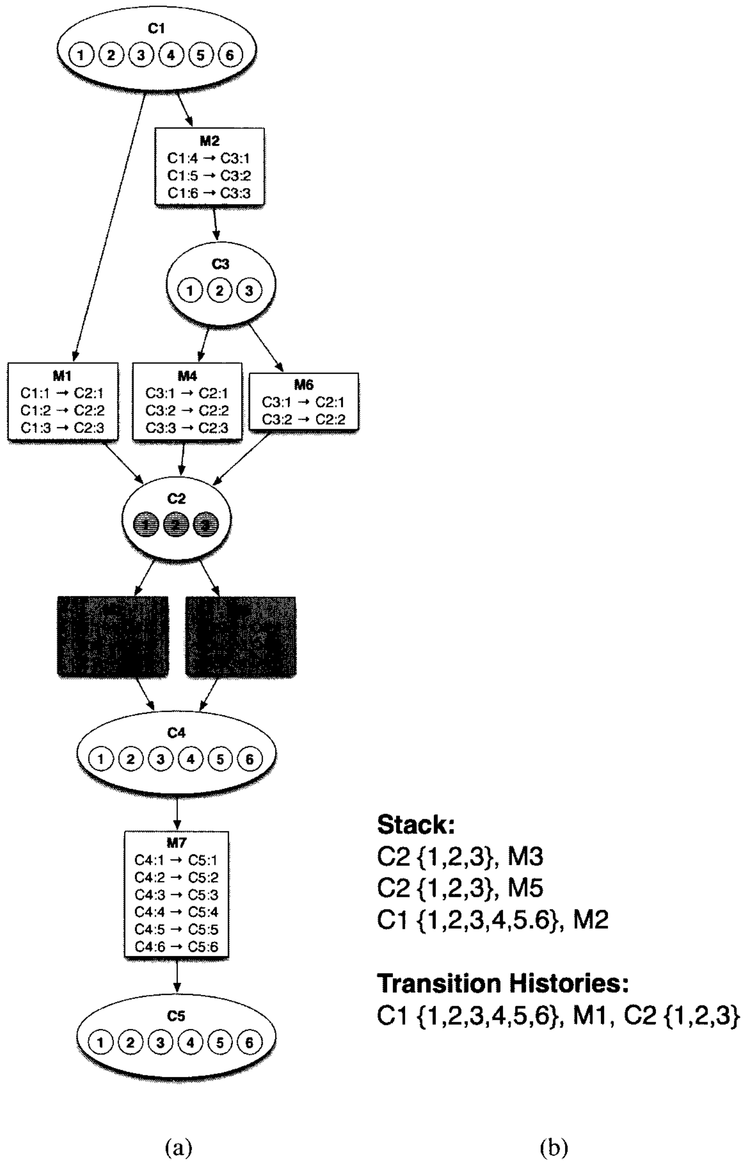
Stack:

C1 {1,2,3,4,5,6}, M1

C1 {1,2,3,4,5.6}, M2

Transition Histories:

Figure 3.2 : A generic G_{am} is drawn in (a), the compound nodes are ovals containing circles representing each atom of the compound and the mapping nodes are rectangles. When an exploration of this G_{am} is started at C1, its atom marking contains all of its atoms, as indicated by coloring them green. The exploration then identifies all adjacent mapping nodes, which are colored magenta. The state of the stack and the list of transition histories is shown in (b). Since this is the start of the exploration there are no transition histories yet. The next step of the exploration is shown in Figure 3.3.



Stack:
C2 {1,2,3}, M3
C2 {1,2,3}, M5
C1 {1,2,3,4,5.6}, M2

Transition Histories:
C1 {1,2,3,4,5,6}, M1, C2 {1,2,3}

(a) (b)

Figure 3.3 : The next step of the exploration of generic G_{am}, continued from Figure 3.2. The first item is popped off of the stack from Figure 3.2(b) and the atom marking of C1 with the atom mapping in M1 is used to compute the resulting atom marking of C2, highlighted in green in (a). The transition history generated by this traversal is added to the list of transition histories in (b). The adjacent mapping nodes to C2 are then added to the stack with the atom marking of C2, highlighted in magenta in (a) and the state of the stack is shown in (b). The exploration continues in this manner until the stack is empty.

mapping node identifier is added. Since the same mapping node can be traversed using different atom markings, the mapping node identifier is appended with a counter that is incremented every time the same mapping node is added to the auxiliary graph. Then an edge is drawn from the input atom marking node to the node representing the mapping node and then from the node representing the mapping node to the output atom marking node. Figure 3.4 illustrates the construction of the auxiliary graph from the generic G_{am} found in Figures 3.2 and 3.3. Figure 3.4(a) contains the resulting L from the full exploration of the generic G_{am} and Figure 3.4(b) shows the auxiliary graph constructed from this L.

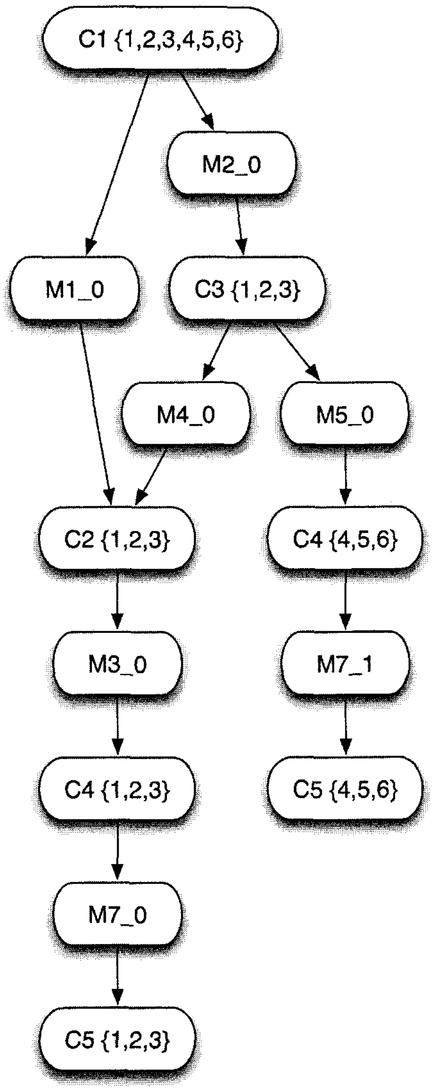
3.2.3 Finding k shortest paths

The final step of LPAT finds the linear atom conserving pathways by using a standard k shortest path algorithm run on the auxiliary graph. The auxiliary graph only contains one node for the start compound, but it may contain several different nodes for the target compound with different atom markings. If a specific atom marking is desired for the target compound, then the corresponding node can be used as the target node for the k shortest path algorithm. For example, there are two different atom markings of C5 in Figure 3.4(b), but they can be easily merged into one node as illustrated in Figure 3.5. Typically, the atom markings of the target compound do not matter for linear pathway finding and so all nodes corresponding to the target compound are merged into one node. Then, the k shortest path algorithm gets as input the auxiliary graph, the node corresponding to the start compound and the node corresponding to the target compound and the resulting set of pathways are linear atom conserving pathways.

The implementation of LPAT in this thesis use Eppstein's k shortest path algorithm [33]. Eppstein's algorithm was chosen because of its low computational cost, $O(m + nlogn + kn)$, where n is the number of verticies, m is the number of edges and k is the number of paths

Stack:

Transition Histories:
C1 {1,2,3,4,5,6}, M1, C2 {1,2,3}
C2 {1,2,3}, M3, C4 {1,2,3}
C4 {1,2,3}, M7, C5 {1,2,3}
C1 {1,2,3,4,5,6}, M2, C3 {1,2,3}
C3 {1,2,3}, M6, C2 {1,2}
C3 {1,2,3}, M4, C2 {1,2,3}
C3 {1,2,3}, M5, C4 {4,5,6}
C4 {4,5,6}, M7, C5 {4,5,6}



(a) (b)

Figure 3.4 : The resulting list of transition histories, L, from the exploration of the generic G_{am}, conserving three atoms, found in Figures 3.2 and 3.3 is shown in (a). L is then used to construct the auxiliary graph found in (b), with nodes created for each atom marking and mapping node and edges from input atom markings to the mapping node and from the mapping node to the output atom marking. Notice that the traversal of M6 is stored in the list of transition histories so that it is not visited again in the exploration, but it does not need to be contained in the auxiliary graph because it cannot be part of a path conserving three atoms. Also notice that M7 is traversed two separate times utilizing different atom markings, this is taken into account by appending the identifier of each mapping node with a counter that distinguishes the two traversals.

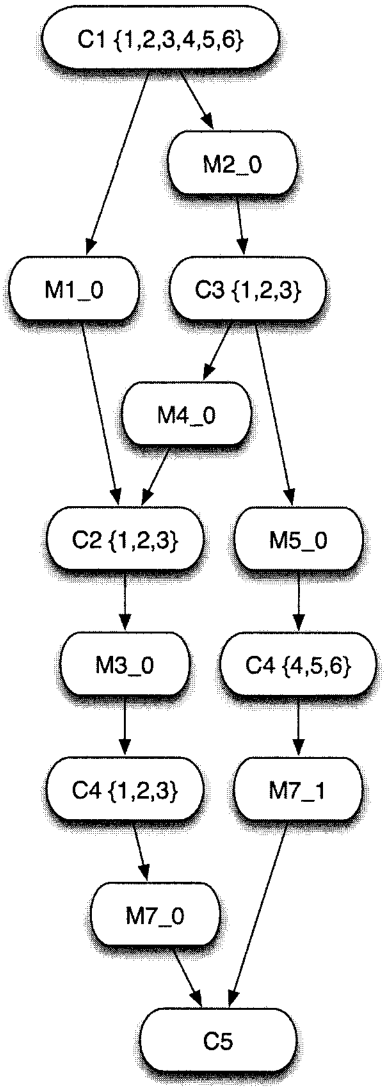


Figure 3.5 : Auxiliary graph from Figure 3.4(b) with the nodes corresponding to C5 merged together for finding pathways between C1 and C5 conserving three carbons. This graph can now be used with *k* shortest path algorithms using "C1 {1,2,3,4,5,6}" as the starting node and "C5" as the target node and the returned paths will correspond to linear pathways conserving three atoms between C1 and C5.

to return. However, Eppstein's algorithm considers all paths, not just simple paths, and so the paths returned can contain cycles. For some applications, finding paths with cycles may be useful, but for the most part metabolic paths with cycles are undesirable. Cycles are especially undesirable when all reactions are considered reversible, as many pathways will contain many cycles from the substrates to the products and back again through the reverse direction. This is highly unlikely biochemical behavior and in the current implementation of LPAT, *k* is typically set to a high value and the resulting pathways are filtered to remove paths with cycles. Future implementations may investigate using *k* shortest path algorithms that only consider simple paths, such as Yen's algorithm that takes $O(kn(m+nlogn))$ [131], which have slower run times but do not require the filtering step.

Chapter 4
Finding Branched Metabolic Pathways

In addition to eliminating inappropriate transitions, atom tracking identifies where atoms are lost and gained along a linear pathway and enables finding branched pathways. This chapter presents two algorithms for Branched Pathfinding using Atom Tracking (BPAT) that both begin by identifying linear pathways, using LPAT, between the start and target compounds. Due to the high complexity of this problem, as discussed in Section 2.3.2, the BPAT algorithms utilize different heuristics and cutoffs to identify biologically relevant pathways. Since BPAT-S and BPAT-M are heuristic, they can return different top ranking results. As further explored in Chapter 5, BPAT-S and BPAT-M both have their own advantages and disadvantages.

Both algorithms take as input the atom mapping graph, G_{am} whose construction is described in Section 3.1, a correspondence between the atom mappings and the reactions they occur in, a start compound, a target compound and a minimal number of atoms to conserve and returning a set of branched pathways, ranked first by the number of atoms conserved and then by the total length, or weight, of the linear pathways used to construct it. The algorithms both begin by passing the appropriate inputs to LPAT and then take two different approaches for finding branched pathways from the linear pathways. The first algorithm, BPAT-S (S for Seed pathways), is described in Section 4.1. BPAT-S works by taking each linear pathway and replacing mapping nodes through which atoms are lost and gained with specific reactions to form seed pathways. Branches can then be added to the seed pathways, resulting in branched pathways. The second algorithm, BPAT-M (M for

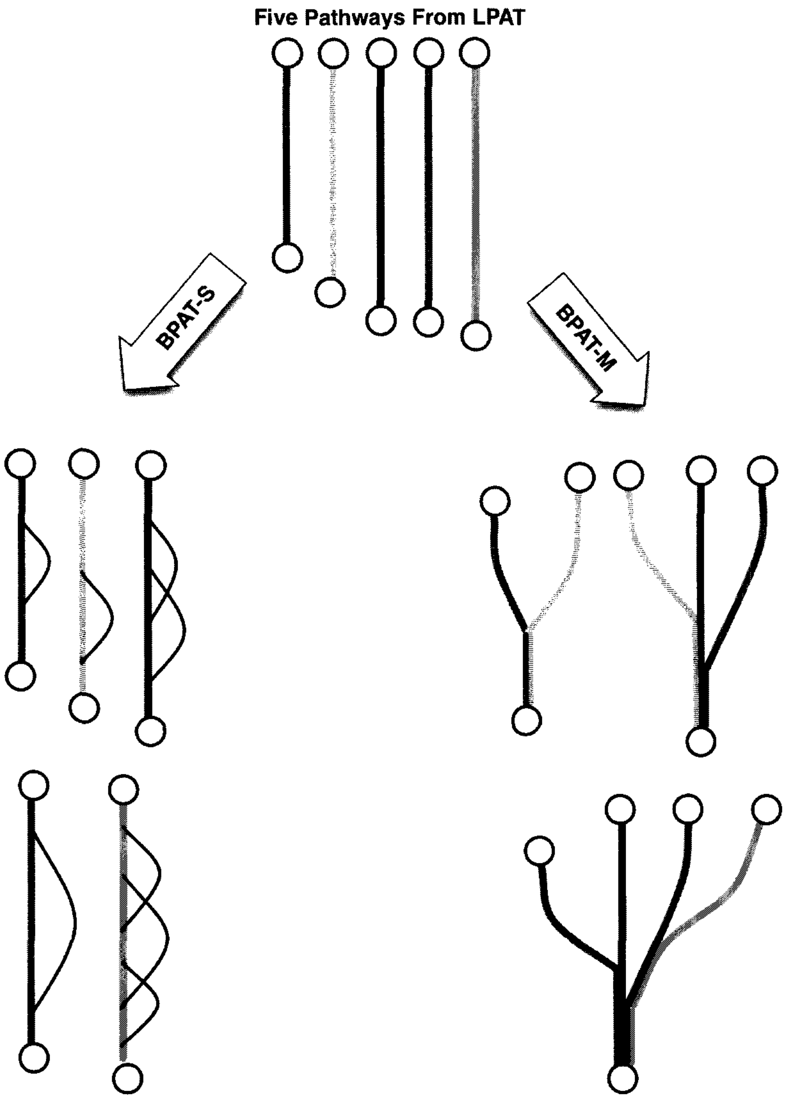


Figure 4.1 : A cartoon depicting the approaches taken by BPAT-S and BPAT-M to find branched pathways. The linear pathways from LPAT are represented as the colored lines and the circles represent the start and target compounds. BPAT-S takes each pathway in turn and finds new branches, represented by the narrower black lines, that can be attached to the linear pathway. BPAT-M does not find new branches, but rather tries to merge the linear pathways into branched pathways.

Merging), is described in Section 4.2. BPAT-M attempts to merge linear pathways together to form branched pathways. An high level depiction of the general approaches for BPAT-S and BPAT-M can be found in Figure 4.1.

4.1 BPAT-S: Branched Pathfinding Using Seed Pathways

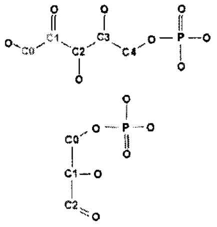
This section describes how BPAT-S finds branched pathways in metabolic networks. The first step of BPAT-S is to use LPAT, described in the previous chapter, to obtain a set of linear pathways between the desired start and target compounds. The linear pathways are then examined to find the compounds through which atoms are lost or gained through. This examination gives rise to the creation of *seed pathways*, where mapping nodes that potentially lose or gain atoms are replaced by specific reactions. Potential branches are then identified between reactions through which atoms are lost and gained. All valid combinations of these branches are systematically attached to the seed pathways to obtain the resulting set of branched pathways.

4.1.1 Generation of Seed Pathways

BPAT-S starts by using LPAT to obtain a set of linear atom conserving pathways, P_l. Each pathway in P_l will potentially give rise to a number of seed pathways. While the atom mapping graph, G_{am}, contains enough information to find linear pathways, it is missing information about which substrate and product compounds not along the pathway through which the atoms can be lost or gained. Identifying these compounds is important because they are the start and target compounds for potential branches. In order to obtain this information, the actual reactions associated with the atom mapping must be examined. Therefore, a correspondence is stored between mapping nodes and reactions.

For the experiments in this thesis, the reaction data is obtained from KEGG REACTION

Table 4.1 : RPAIR entry RP00080 between D-xylulose 5-phosphate (C00231) and D-glyceraldehyde 3-phosphate (C00118).

RPAIR ID	Substrate ID	Atom ID	Product ID	Atom ID	Chemical Drawing
RP00080	C00231	C2	C00118	C2	
	C00231	C3	C00118	C1	
	C00231	C4	C00118	C0	

database and KEGG also provides the correspondence between RPAIR and REACTION entries. For example, RP00080 in Table 4.1 is associated with six different reactions in KEGG, which use and produce different compounds in addition to D-xylulose 5-phosphate and D-glyceraldehyde 3-phosphate. Two of the reactions are illustrated in Figure 4.2. In one reaction, D-xylulose 5-phosphate reacts with formaldehyde to produce D-glyceraldehyde 3-phosphate and glycerone. In the other, D-xylulose 5-phosphate reacts with orthophosphate to produce D-glyceraldehyde 3-phosphate and acetyl phosphate. The important difference is that in the first reaction the C0 and C1 carbons of D-xylulose 5-phosphate end up in glycerone and in the second reaction they end up in acetyl phosphate. Hence, the starting compound of the branch is different depending which reaction is used.

By using the corresponding reactions, for each $p \in P_l$ two types of mapping nodes are identified: loss mapping nodes (LMNs) and gain mapping nodes (GMNs). LMNs are mapping nodes where the atom mapping does not map all of the atoms in the input compound and GMNs are mapping nodes where the atom mapping does not map all of the atoms in the output compound. For example, RP00080 in Table 4.1 would be considered a LMN.

The construction of seed pathways begins by (a) replacing each LMN in p with all

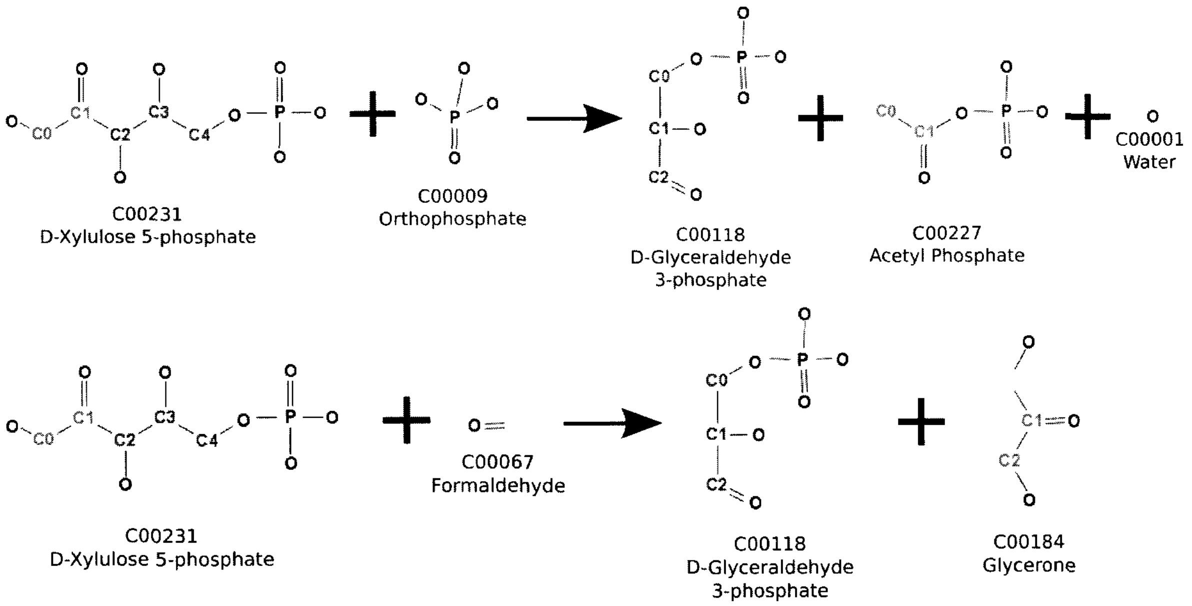


Figure 4.2 : The reactions R01621 (top) and R01440 (bottom) contain the atom mapping, RP00080, found in Table 4.1. If RP00080 was in a linear pathway, it would be considered a LMN, R01621 and R01440 would be two of the corresponding LRNs, with R01621 resulting in acetyl phosphate being the LCN and R01440 resulting in glycerone being the LCN.

possible corresponding reactions - called loss reactions nodes (LRNs), and (b) replacing each GMN in p with all possible corresponding reactions - called gain reaction nodes (GRNs). Since a mapping node can have multiple corresponding reactions, one seed pathway is created for each possible combination of LRNs and GRNs along the pathway. For example, since RP00080 in Table 4.1 is found in six reactions in KEGG at least six seed pathways would be created, each one containing one of the reactions. Depending on the reactions corresponding to the other LMN and GNMs, more seed pathways may be generated corresponding to all possible combinations of LRNs and GRNs.

There exists the theoretical possibility of a combinatorial explosion, as the number of seed pathways is equal to the number of possible combinations of reactions along the pathway. However, in practice this has never been a problem because most pathways have few LMNs and GMNs, which also typically correspond to only a few reactions. Additionally, reactions are considered the same if the compound through which atoms could be lost and gained through are the same. For example, there are many reactions that include the atom mapping from ATP to ADP, but many of these reactions are identical in that the phosphorus atom is lost through phosphate. Therefore, all of these reactions can be considered as the same LRN to reduce the number of seed pathways.

The final step in constructing seed pathways is to attach the compound nodes through which atoms are lost or gained for each LRN and GRN. For each seed pathway, its LRNs are examined and the compound nodes through which the atoms are lost, LCNs, are identified and attached to the seed pathway. This is done by creating a directed edge from the LRN to the LCN. The GRNs are then examined and the compounds nodes through which atoms could be gained, GCNs, are also identified and attached to the seed pathway. This is done by creating a directed edge from the GCN to the GRN. The final resulting seed pathways contain possible attachment points for branches and are used to obtain branched pathways.

4.1.2 Attaching Branches to Seed Pathways

The seed pathways can then be used as input for Algorithm 4.1.1 which returns branched pathways first sorted by the total number of atoms conserved and then by the total number of nodes. Since the algorithm is based on LPAT, as described in Chapter 3, weighting schemes can be easily incorporated and the results sorted by the total weight of the branched pathway. Additionally, utilizing multiple molecule of the start compound is a natural extension that can be done in two ways. One is for the user to specify the number of start molecules to use. Then, a new compound node corresponding to the multiple molecules is added and used as the start compound. This new compound node is attached to the graph by creating new mapping nodes, one for each molecule desired. These new mapping nodes then correspond to a new reaction so that each of the molecules can be used as the starting point for branches. This the the method used for the experimental results found in this thesis. Another, more general, extension is to allow any branch to start from a molecule of the start compound. Either way, the only thing that changes is that the LCNs corresponding to the start compound are considered as contributing additional atoms to the pathway when the number of atoms conserved are calculated.

Algorithm 4.1.1 proceeds by examining each augmented seed pathway and obtaining potential branches by finding the shortest maximal atom conserving linear pathways between all pairs of LCNs and GCNs (lines 7-10). Then, all combinations of possible branches are tried systematically for attachment to the augmented seed pathway (lines 11-15). Testing all combinations of branches is necessary because adding a branch to an augmented seed pathway may affect the atom tracking down the pathway.

The brute force nature of trying all combinations in Algorithm 4.1.1 may result in unreasonable running times, although this was rarely encountered in practice. This is partially due to the observation that the start and target compounds for branches were often

Algorithm 4.1.1 Find branched pathways using seed pathways (BPAT-S)

```
Input: Set of seed pathways $\mathcal{S}$, Atom mapping graph $G_{am}$, max number of branches $b$
Output: Sorted list of branched pathways $\mathcal{P}$, sorted first by number of atoms conserved, then by
    total number of nodes
$\mathcal{P} \leftarrow \{\}$
$M_b \leftarrow \{\}$ ($M_b$ is map between a pair of compound nodes, $c_x, c_y$ and the shortest maximal atom
    conserving linear pathway in $G_{am}$ from $c_x$ to $c_y$)
for each $p$ in $\mathcal{S}$ do
  $C_l \leftarrow$ all compound nodes from $p$ through which atoms may be lost
  $C_g \leftarrow$ all compound nodes from $p$ through which atoms may be gained
  $B \leftarrow \{\}$
  for each $c_l, c_g$ in $C_l \times C_g$ do
    if $M_b(c_l, c_g)$ has not yet been set then
      $M_b(c_l, c_g) \leftarrow$ the shortest maximal atom conserving linear pathway in $G_{am}$ from $c_l$ to $c_g$
    Add $M_b(c_l, c_g)$ to $B$
  for each $n = 1$ to $b$ do
    for each combination $V$ of cardinality $n$ from $B$ do
      if all paths in $V$ start from different $c_l \in C_l$ and end at different $c_g \in C_g$ then
        $p_b \leftarrow$ branched pathway created by attaching all branches in $V$ to $p$
        Determine the number of atoms conserved along $p_b$
        Add $p_b$ to $\mathcal{P}$
```

the same across a number of pathways. This observation resulted in taking extra care to maintain any previously computed branches in a global data structure, M_b, so that branches can be reused for multiple pathways to reduce computation time. Experimentation also led to the empirical observation that trying all combinations can lead to long run times without substantially improving biological value. Therefore, there is an option to limit the maximum number of branches that can be added by b, although this was never used for the results presented in this thesis.

4.2 BPAT-M: Branched Pathfinding by Merging Linear Pathways

This section describes another algorithm, Branched Pathfinding using Atom Tracking and Merging (BPAT-M), for finding branched pathways by combining the linear pathways returned by LPAT. The idea of merging the linear pathways is motivated by the observation that a significant portion of time is spent finding the branches in BPAT-S, but these branches may already be contained in the set of linear pathways found by LPAT. Therefore, better performance may be achieved by merging the linear pathways into branched pathways, instead of re-finding the same pathways. Furthermore, BPAT-M also has the advantage that all pathways are considered equal, there is no division between seed and branch pathways. This means that BPAT-M can potentially find pathways with a wider variety topologies, as compared to BPAT-S where there is the central seed pathway and the branches must start and end on the seed pathway.

In general, it is difficult to determine how to merge the pathways to obtain better performance because there are typically thousands of linear pathways and testing all combinations would be infeasible. Fortunately, the number of possibilities is greatly reduced because of biochemical constraints. Only pathways that interact properly biochemically, resulting in more atoms conserved in the target compound, can be merged together to make a realistic branched pathway. BPAT-M takes advantage of these constraints by processing the linear pathways to construct three data structures Q, C, and M, that are important for performance and their construction is detailed in the next two sections. The last section describes Algorithm 4.2.1, which takes Q, M and C as inputs, along with k number of pathways to return, and returns k branched pathways ranked first by the number of atoms conserved and second by the number of reactions. There is an additional user specified parameter w for Algorithm 4.2.1 that can be used to further limit the run time and/or storage needs as necessary.

4.2.1 Construction of Q and C Based on Target Atom Markings (TAMs)

A target atom marking (TAM) of a linear pathway is a set of indices corresponding to the specific atoms in the target compound that have been conserved from the start compound. So for a target compound containing six carbon atoms, linear pathways that conserve at least three carbons could potentially result in 42 different TAMs as $\sum_{k=3}^{6}\binom{6}{k} = 42$. Typically, the number of TAMs found is much less than the theoretical maximum number due to chemical constraints. Figure 4.3 displays four linear pathways from α-D-glucose 6-phosphate to stachyose and their associated TAMs. TAMs play a central role in the performance of BPAT-M because they allow a quick way to determine which linear pathways can not be merged together. Two pathways can not be merged together if the intersection of their TAMs is nonempty, that is if they contain the same atom index or indices. This is because merging two linear pathways that have overlapping TAMs would require that two atoms, one from each pathway, would arrive at the same atom position in the target compound and this is physically impossible. If two pathways have disjoint TAMs, they can not necessarily be merged because it must be checked whether they share a common reaction, as described in the next section. However, if two pathways are mergable, then the TAM of the merged pathway is the union of the TAMs of the pathways.

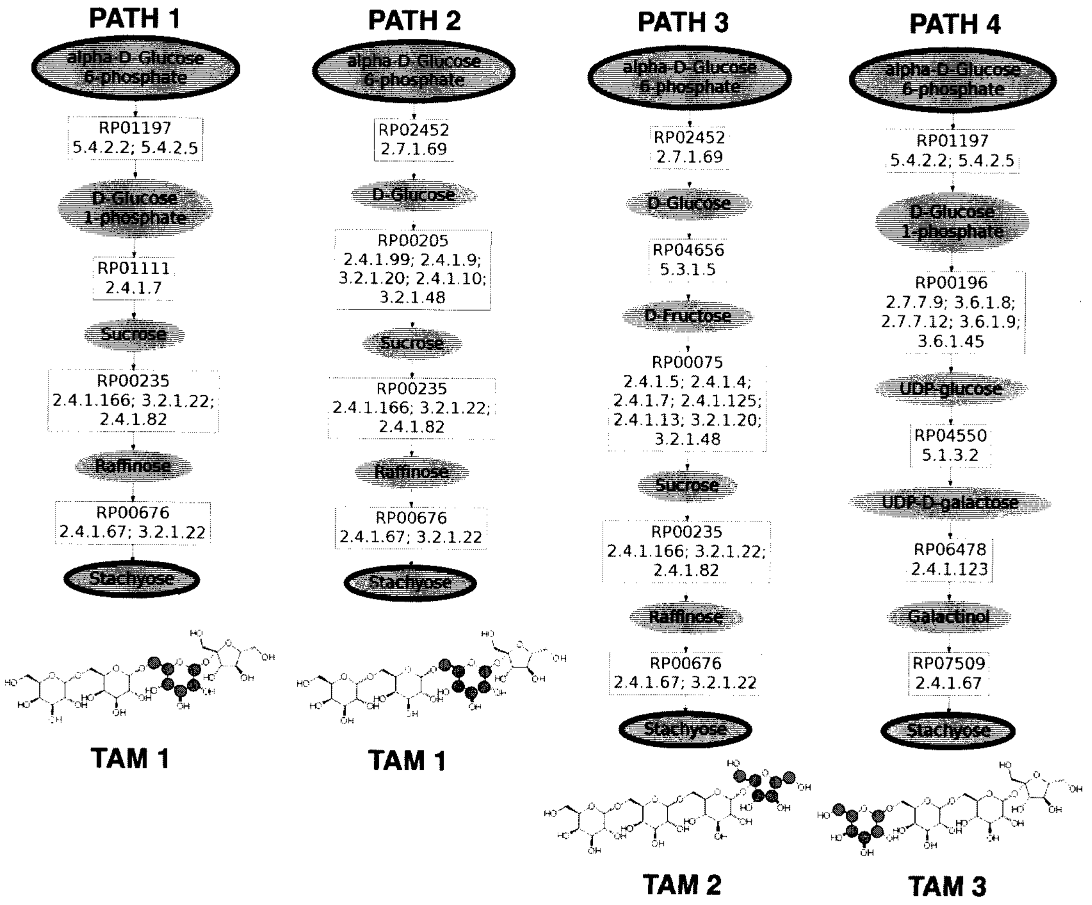


Figure 4.3 : Four linear pathways from α-D-glucose 6-phosphate to stachyose and their associated carbon TAMs, as indicated by the magenta circles.

The ability to use the TAMs to quickly determine if pathways are not mergable motivates the construction of the data structure, Q, which maps a TAM to a set of linear pathways containing that TAM. For a particular TAM, t, this means that $Q[t]$ returns all linear pathways whose TAM is equal to t. For example, using the pathways depicted in Figure 4.3, Q[TAM 1] would return the pathways labeled PATH 1 and PATH 2. After Q is constructed, all disjoint combinations of the TAMs from the linear pathways are computed and stored in a list C. For example, for the TAMs, $t_1 = \{0,1,2\}$, $t_2 = \{0,1\}$, $t_3 = \{2,3\}$ and $t_4 = \{4,5\}$, C would contain $\{t_1,t_4\}$, $\{t_2,t_3,t_4\}$, $\{t_2,t_3\}$, $\{t_2,t_4\}$ and $\{t_3,t_4\}$ as all disjoint combinations. C is then sorted in decreasing order by the total size of the combination. In this example, $\{t_2,t_3,t_4\}$ would be first entry in C because it is of size six. Sorting C this way is important because the goal is to find pathways that conserve the maximal number of atoms. For each combination $c \in C$, the TAMs are accessed by their indices, so if $c = \{t_2,t_3,t_4\}$, $c[1] = t_2$, $c[2] = t_3$ and $c[3] = t_4$. C is then used to dictate how the search proceeds to try and merge combinations of linear pathways to obtain branched pathways. Since C is sorted and the TAM of the merged pathway is the union of the TAMs of the linear pathways; if k pathways have been found and the next combination in C is smaller than the number of atoms conserved in the kth pathway, the search can be stopped and the pathways returned because any further pathway found would result in fewer atoms being conserved. The use of Q and C greatly improves the performance of BPAT-M by providing a quick way to identify pathway combinations that can potentially be merged.

4.2.2 Construction of M

The data structure M maps all pairs of mergable linear pathways to a tuple containing the reaction used to merged the pathways and the number of mapping nodes from the target compound that the merge point occurs at in each pathway. Two linear pathways, p_1 and

p_2, are considered *mergable* if they both contain a mapping node that can be replaced by the same reaction, r, and the atoms conserved along p_1 and p_2 are all incorporated into the same product molecule by r and subsequently into the same target compound molecule. Otherwise, the pathways are not mergable. This means that all mapping nodes from the target compound to r in each pathway must be the same. Therefore, the mergability of p_1 and p_2 starts by identifying the first different atom mapping nodes, m_1 in p_1 and m_2 in p_2, from the target compound and then testing whether m_1 and m_2 can be replaced by the same reaction, r, and merged. The tuple containing r, m_1 and m_2 is called a *merge point.*

All pairs of pathways that have disjoint TAMs are tested whether they are mergable; if they are, the corresponding entry in M is set to be the merge point, that is, $M[p_1, p_2] = (r, m_1, m_2)$. Figure 4.4 contains two mergable pathways from α-D-glucose 6-phosphate to stachyose that have disjoint TAMs. For example, since the pathways in Figure 4.4 are mergable, an entry for them would be created in M that contains $(\text{R00803}, 2, 3)$, because the merge point occurs at the second mapping node from stachyose in PATH 1 and at the third mapping node from stachyose in PATH 2. The construction of M essentially results in branched pathways containing two linear pathways. The next section describes how BPAT-M uses M, together with Q and A, to find branched pathways containing more than two linear pathways by incrementally merging additional linear pathways.

4.2.3 Finding Branched Pathways Using Q, C and M

After processing the linear pathways to construct Q, C and M, they are given as input to Algorithm 4.2.1 along with k number of branched pathways to return and w. Algorithm 4.2.1 then returns the final result of BPAT-M, the top k branched pathways it finds ranked first by the number of atoms conserved and then by the number of reactions. Even with the lengths taken to reduce the number of combinations of linear pathways that need to be tested, there

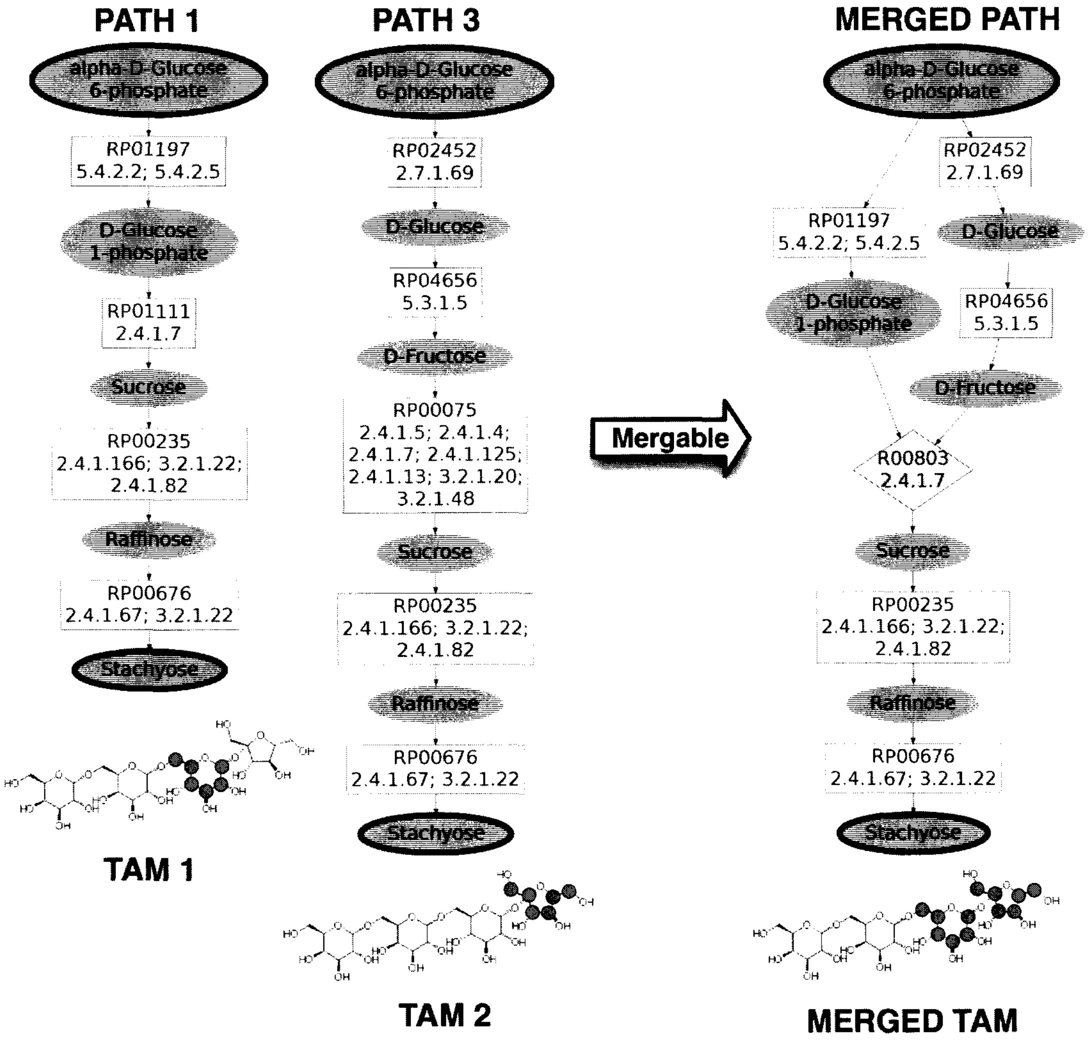


Figure 4.4 : Two mergable linear pathways from Figure 4.3 and their associated carbon TAMs, as indicated by in magenta circles.

are still an unreasonable number and a cutoff parameter for the search, w, is introduced. Exactly how w limits the search is explained later in the section and its introduction means that BPAT-M does not guarantee finding the optimal combination, but the results presented in Chapter 5 demonstrate that it performs well in practice.

Algorithm 4.2.1 works by taking each combination of TAMs $c \in C$ in turn and using them to build branched pathway combinations. For each combination c, the first two TAMs, $c[1]$ and $c[2]$, are used to obtain the set of associated pathways for each TAM from Q and all pairs of pathways are tested for mergability using M (lines 6-8). If a pair of pathways are mergable, then they are stored in the set of intermediate branched pathways (IBPs), $\mathscr{T}$. The IBPs store a list of mergable linear pathways and their merge points. Then, for each subsequent TAM in C, all of the pathways associated with the TAM $c[n]$, $p_q \in Q[C[n]]$ are retrieved (line 9). Each $p_q \in Q[C[n]]$ is then tested for mergability with each linear pathway in each IBP (lines 12-15), since retrieving p_q from $Q[C[n]]$ just means p_q's TAM is disjoint from the TAMs of the pathways already contained in the IBP, which does not guarantee mergability. If p_q is mergable with a linear pathway, p_l in IBP, that is $M[p_q, p_l]$ contains a merge point, p_q can potentially be merged with the IBP to create a branched pathway that conserves more atoms. However, because p_l has already been merged with other pathways, it must be verified that the merge point between p_q and p_l is still valid.

A merge point is always valid if p_l has not been merged with another pathway in the IBP at the same mapping node it would use to merge with the new pathway, p_q. However, if p_l has been previously merged at the same point with another pathway in the IBP, the merge point with p_q can still be valid if the reaction in the merge point of p_q and p_l is the same reaction used previously and the substrate in p_q is not contained in the other pathways. Otherwise, the merge point is invalid. For example, if there is a reaction r that takes the substrate compounds a, b and c. If in the IBP p_l was merged with another pathway at r with

p_l containing a and the other containing b, then if the merge point between p_l and p_q is also at the same point but p_q contains c then the merge point is still valid. However, if p_q contained b then the merge point would now be invalid.

If the merge point is valid, p_q is merged with the IBP and the resulting branched pathway is stored as another IBP (line 16). Therefore, each IBP gives rise to a number of new IBPs equal to the number of p_q that have valid merge points with the IBP. This means that there is a theoretical combinatorial explosion of IBPs for each C_i and unfortunately, unlike previous situations, an unreasonable number of IBPs is typically generated. This results in the introduction of the parameter r to limit the number of combinations generated. After adding the pathways for each TAM, only the top w IBPs, sorted by number of atoms conserved and the sum of the length of the linear pathways, are carried over for each subsequent TAM (lines 17-18). As previously noted, by sorting C by the size of each combination the k top branched pathways can be returned before trying every combination in C if the next combination is smaller than the TAM of the kth pathway (lines 3-4).

The final way in which the run time and/or space required by BPAT-M is reduced is by limiting the number of pathways that are kept for each TAM in Q. This is done because it was observed that in some cases a few clusters can be very large and generate unreasonable number of IBPs, even with a low limit on r. The hope is that by removing longer pathways with the same TAMs, these are linear pathways that would not be contained in the final branched pathway set. However, future work is needed to investigate the impact of different limits to the number of IBPs and size of the pathway clusters. In Chapter 5, a number of results for branched pathways are presented that demonstrate that even with the heuristic limits, the search performs well and proves to be complimentary to BPAT-S.

Algorithm 4.2.1 Final Step of BPAT-M

Input: Pathways organized by their TAMs, Q; Sorted list of all combinations of disjoint TAMs, C; Mergable pairs of paths, M; Number of pathways to return, k; Limit on IBPs, w;

Output: Sorted list of branched pathways $\mathscr{P}$, containing linear pathways and merge points, sorted first by number of atoms conserved, then by total number of nodes

```
$\mathscr{P} \leftarrow \{\}$
for each $c$ in $C$ do
    if $\mathscr{P}$ contains more than $k$ pathways and the $k$th pathway conserves more atoms than the size of $c$ then
        Break
    $\mathscr{T} \leftarrow \{\}$ //for storing the IBPs
    for each pair of linear pathways $(p_i, p_j)$ in $(Q[c[1]] \times Q[c[2]])$ do
        if $M(p_i, p_j)$ exists then
            Add IBP containing $p_i, p_j, M(p_i, p_j)$ to $\mathscr{T}$
    for $n = 3$ to size of $c$ do
        $\mathscr{N} \leftarrow \mathscr{T}$
        $\mathscr{T} \leftarrow \{\}$
        for each IBP $P$ in $\mathscr{N}$ do
            for each linear pathway $p_q$ in $Q[c[n]]$ do
                for each linear pathway $p_l$ in $P$ do
                    if $M(p_q, p_l)$ exists and is a valid merge point in $P$ then
                        Add new IBP containing $P$ merged with $p_q$ and $M(p_q, p_l)$ to $\mathscr{T}$
        if $\mathscr{T}$ contains more than $w$ pathways then
            Truncate $\mathscr{T}$ to $w$ pathways
Add all pathways in $\mathscr{T}$ to $\mathscr{P}$ and return $\mathscr{P}$
```

Chapter 5

Experimental Setup and Results

5.1 KEGG Data Collection and Processing

Until recently the development of automated ways to track atoms through large metabolic networks has been hindered by a lack of atom mapping data. Fortunately, as discussed in Section 2.1.1, large scale curation efforts have resulted in the increased availability of atom mapping data for chemical reactions. Progress has also been made in computational tools for automatically generating correct atom mappings, which can be used to fill in the gaps of the manual curation process [1, 15, 16]. The experiments in this thesis use data from the Kyoto Encyclopedia of Genes and Genomes (KEGG) due to the existence of the curated KEGG RPAIR database [56]. Each KEGG RPAIR entry contains structural information for each compound, an alignment mapping atoms between the two compounds and a list of associated reactions [57].

The algorithms presented in this thesis require that there is a universal index for each atom in each compound. This would usually be provided by the KEGG LIGAND database that contains the chemical structures for each compound in a MDL Molfile with a unique index for each atom [42]. However, the KEGG RPAIR database was developed independently from the the KEGG LIGAND database and RPAIR entries contain their own KEGG Chemical Function (KCF) description of the structures being mapped. As such, the atom indices contained in the RPAIR data are not necessarily the same indices contained in the LIGAND data. This would not normally create a problem except that the RPAIR data is inconsistent with itself, two different RPAIR entries containing the same compound may have different indices for the same atoms in the same compound. Since it is important to utilize as much of the atom mapping data as possible, several steps are taken to discard as little data as possible.

The processing of the RPAIR data begins by identifying compounds whose atoms are consistently indexed and RPAIR entries that contain two consistent compounds are used as downloaded. In cases with inconsistency, an attempt is made to determine universal indices for the atoms using the structure in the LIGAND database as the reference structure. This is done by first checking whether there exists an isomorphism between the structural data found in the RPAIR entry and the data found in the LIGAND database. If there exists only one isomorphism, this mapping is used to produce a consistent RPAIR. If there is more than one isomorphism mapping, the RMSD is calculated between the two mappings and if it is equal to zero, that mapping is used. Otherwise, the RPAIR is removed from the data set. Additionally, any RPAIRs which consist of an atom mapping where the atom types being mapped do not match are discarded, except in the generic compound cases containing "R". KEGG contains generic compounds such as "alcohol" where the structure contains a "R". Mappings with generic mappings where the "R" is not mapped to a specific atom are included, otherwise the mapping is discarded. The overall processing of the RPAIR data results in discarding about one percent of the entries.

All KEGG data used in the experiments of this thesis was downloaded on February 10, 2010. The processed data contained 11,892 consistent RPAIR entries involving 6,002 compounds and corresponding to 7,510 reactions.

5.2 Reaction Reversibility Information

A difficulty that has plagued computational approaches to finding metabolic pathways is the lack of easily available information on the reversibility of enzymatic reactions. This is partially because the direction of the reactions is dependent on the thermodynamics of the system, which can change based on the environmental conditions such as pH, temperature and concentration. Still, many reactions are more favorable in one direction and are never or very rarely observed in biological conditions. Unfortunately, most metabolic databases, including the KEGG REACTION database, do not currently include systematic information about reversibility of the reactions. However, in the manually drawn KEGG pathway maps one will notice that a number of arrows drawn are unidirectional. Therefore, the pathway maps contain manually curated reversibility information. This information can be extracted from the XML representations of the pathway maps, distributed in the the KEGG Markup Language (KGML), an XML representation.

A KGML file is obtained for each pathway map for a total of 162 files. The KGML language contains a tag "reaction" representing a reaction element that has two attributes: "name" corresponding to the reaction's KEGG ID and "type" which can be "reversible" or "irreversible". A reaction has two possible child elements, "substrate" and "product" which contain the substrate and product compounds and their ID's. All of this information is extracted from the KGML and used to identify irreversible reactions. However, similar to the RPAIR data, when a reaction occurs more than once in the pathway maps, which are described in Section 2.1, there are sometimes inconsistencies in the KGML data that need to be reconciled. The two major inconsistencies are when the substrate or products do not match and/or the reversibility information does not match. If the inconsistencies cannot be resolved, the default is to label the reaction as reversible. If only one of the either the substrates or products does not match previously seen reactions, then the direction of the reaction is considered correct. If neither the substrates or products match, the reverse direction is tried and if that matches then the direction of the reaction is considered the opposite given. Otherwise, the reaction is considered reversible. If the substrate and products do match but the reversibility information does not, the reaction is then considered reversible.

The processing of the KGML pathway maps resulted in 4,360 reactions being labeled irreversible. Once the reaction direction is determined, this information is then used to label RPAIR entries reversible or irreversible, which is used in the atom mapping graph construction. This is slightly complicated by the fact that one RPAIR entry can be associated with multiple reactions. Therefore, for each RPAIR entry all associated reactions have to be checked for directionality. If all of the reactions are irreversible and consistent in the labeling of the compounds as substrates and products then the RPAIR entry is considered irreversible. Otherwise, the entry is labeled as reversible. This results in 6,386 RPAIR entries being considered irreversible.

5.3 Implementation Details and Hardware Specification

The implementation was done in Java using the Chemical Development Kit [112] and the Java Universal Network/Graph Framework (http://jung.sourceforge.net/). All result figures are drawn using Graphviz (http://www.research.att.com/sw/to
ols/graphviz/). All experiments were run on the Shared University Grid at Rice (SUG@R), using a single core from a 2.83GHz Intel Xeon E5440 with access to 16GB of RAM for each pathway. For all experiments, except where noted, the input *k* for Eppstein's *k* shortest paths algorithm was set to one million.

5.4 Linear Pathway Results

Properly evaluating metabolic path finding methods can be a difficult task, even for linear pathways, because there is no standard test set. Quantitative evaluation can be especially troublesome when the goal is to discover previously unknown pathways or pathways composed of reactions from multiple organisms. To overcome these difficulties, results are presented on two separate test sets composed of known metabolic pathways. One is a set of pathways from a previous study that includes a variety of different pathways from *E. coli*, *S. cerevisiae* and *H. sapiens* [35]. The other is a set of amino acid biosynthesis pathways that was manually curated and constructed from a variety of sources to capture the diversity of pathways found in different organisms.

The construction of the test sets and the resulting evaluation of linear pathway finding methods is presented and discussed in Section 5.4.1. Further probing of the performance of atom tracking and weighting methods is done by analyzing two specific examples that highlight general themes in the results. Section 5.4.2 presents results for pyruvate to 3-hydroyxpropanoate (3HP), which is a compound of high interest as a target of metabolic engineering because of its numerous industrial applications and cost [49, 53]. Lastly, Section 5.4.3 investigates using clustering techniques to understand the overall diversity of linear pathways found by LPAT.

5.4.1 Evaluation Using Known Linear Metabolic Pathways

Validation and evaluation of LPAT, as described in Chapter 3, is accomplished by utilizing two tests set of known metabolic pathways. One set is constructed by obtaining a test set of known metabolic pathways from a recent evaluation on linear pathways [35]. The original set was based on metabolic pathways in the aMAZE database, which provides cross references to KEGG, and contained 55 pathways from *E. coli*, *S. cerevisiae* and *H.*

sapiens. The set was then further processed to remove any pathways that were missing atom mapping data or no carbons are conserved from the start to target compounds. The amino acid biosynthesis pathways were also removed to avoid redundancy with the second test set. This resulted in 43 known metabolic pathways contained in the test set, referred to as the *NeAT test set*, after the Network Analysis Tools (NeAT) used in the previous evaluation [19]. A list of all of the pathways in the NeAT set can be found in Table A.1.

While the NeAT set contains a variety of metabolic pathways, it mostly lacks alternative pathways between start and target compounds because it focuses on pathways from only three organisms. Finding alternative pathways or pathways composed of reactions from multiple organisms is an important application for the algorithms in this thesis. Therefore, a second test set, referred to as the *amino acid test set*, was built with the purpose of capturing the diversity of metabolic pathways. The amino acid set is composed of amino acid biosynthesis pathways because they have been well studied across many organisms and can often be synthesized by several different routes [81, 128]. Additionally, essential amino acids, such as lysine, threonine, and tryptophan, have been important successes in metabolic engineering efforts and are now primarily produced by engineered strains of bacteria [71]. The amino acid set was manually constructed by scouring textbooks, online resources such as KEGG and MetaCyc and literature to identify as many known amino acid biosynthesis pathways as possible [79, 69, 56, 58, 81, 128, 71]. The resulting amino acid set contained 71 pathways for the 20 common amino acids between 39 pairs of start and target compounds. A list of all of the pathways in the amino acid set can be found in Table A.2.

The test sets are used to evaluate four different carbon tracking methods in conjunction with three different weighting methods. All of the different methods are tested on the atom mapping graph built using the processed KEGG RPAIR data, both with the reversibility information as described in Section 5.2 or without, where all reactions are considered reversible. The different carbon tracking methods include:

- *no carbon* tracking, ignores the atom mapping data and finds the k-shortest paths in the graph
- *one carbon* tracking, where the k-shortest paths are filtered to only include pathways where one carbon is conserved
- *half carbon* tracking, finds k-shortest paths that conserve half of the theoretical maximum number of carbons (rounded up)

For example, if the start compound contained 6 carbons and the target compound contained 5 carbons, the max carbon tracking would find pathways that conserved 5 carbons and decrement by one until pathways are found. In contrast, the half carbon tracking would identify 5 as the theoretical maximum number of carbons and find pathways that conserved 3 carbons, again decrementing by one until pathways are found. Both the max carbon and half carbon tracking utilize LPAT and will be labeled LPAT-max and LPAT-half in the following results. The three different weighting methods include:

- *short*, where every node gets weight equal to 1 and is the same as finding the shortest path
- *degree*, where the compound nodes are given weight equal to their degree
- *context*, which is the weighting scheme used by MetaRoute as described in Section 2.3.1 [15]

For each combination of carbon tracking and weighting methods, the start and target compounds of the known pathways are used as inputs and the accuracy of the computed pathways is calculated using measures from previous evaluations of metabolic path finding methods [26, 15, 16, 93, 35]. True positives (TP) are compounds found in both pathways; false negatives (FN) are compounds in the known pathway which are not in the computed pathway; false positives (FP) are compounds not in the known pathway which are in the computed pathway. For each pathway, the sensitivity (Sn), positive predictive value (PPV) and accuracy (Ac) are calculated as follows:

$$Sn = TP/(TP+FN)$$

$$PPV = TP/(TP+FP)$$

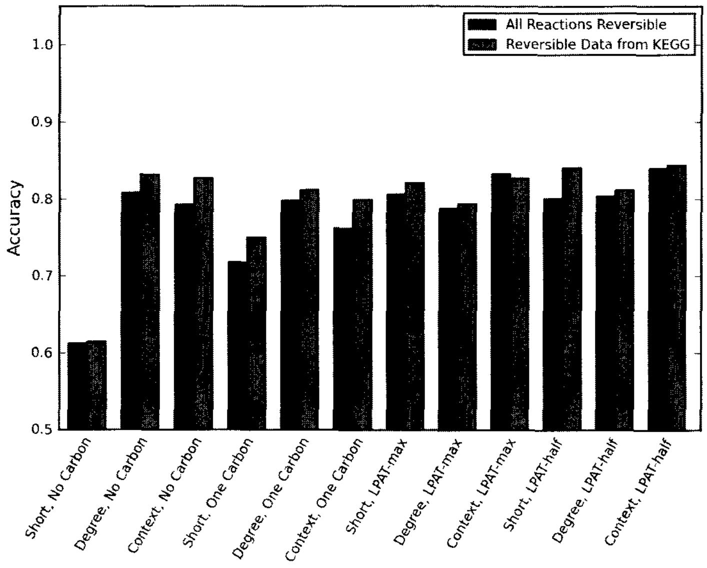
$$Ac = (Sn+PPV)/2$$

Positive predictive value is used instead of specificity because true negatives do not exist in this comparison. For the NeAT test set, each start and target compound is only associated with one known pathway. However, since each start and target compound can be associated with more than one pathway in the amino acid test set, for evaluating the accuracy, the known pathway is considered the union of all of the known pathways.

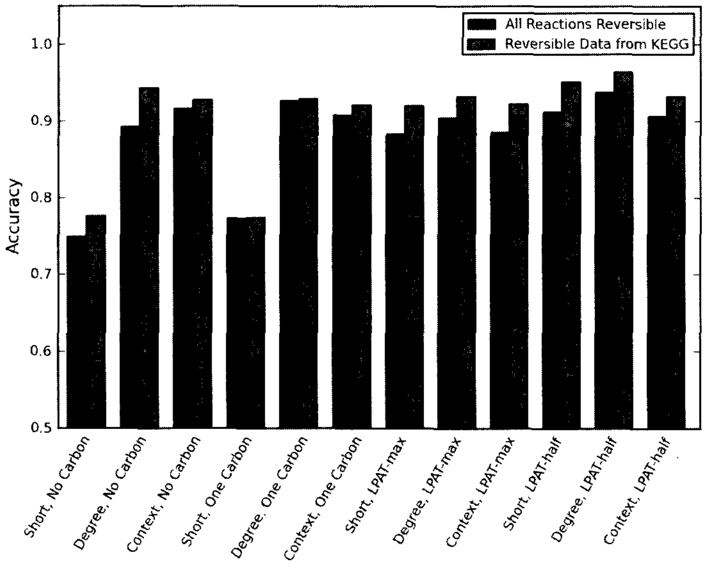
Figures 5.1(a) and 5.1(b) contain the average accuracy results for the NeAT test set and the amino acid test set, respectively. The average accuracy values are obtained by identifying the pathway with the best accuracy from the top 10 computed pathways and averaging the best accuracies for all of the pathways in the test set.The major observation from the results is that all linear pathway finding methods, except for short, no carbon and short, one carbon perform similarly on average. For all of the path finding methods, the average accuracy was higher on the amino acid test set versus the NeAT test set. This is likely due to two factors, one is the construction of the amino acid test set to contain multiple pathways for each amino acid coupled with the atom mapping graph being built using data for all organisms in KEGG. The other factor is that the amino acid test set contains pathways that only consist of one reaction and these are easy to find. While the relative accuracy of the amino acid test set was higher, similar conclusions can be drawn from both test sets.

Using the reversible data from KEGG generally improved the results for all pathway finding methods, although the amount of improvement depended on the particular method. This is an expected result as information about the reversibility of a reaction eliminates finding "short cuts" by going through a reaction in the opposite direction it would normally occur. An example is the pathways found from chorismate to tryptophan. Without the reversibility information, the shortest path includes a series of reactions that are used in tryptophan degradation to make formylanthranilate and L-formylkynurenine. However, it is likely infeasiable for these reactions to function in this direction and the KEGG reversibility data includes this information. The correct biosynthesis pathway, which is one reaction longer than the pathway using the degradation reactions, is found as the shortest path when the reversibility data is used. The only exception is in the NeAT test set where the "context, LPAT-max" method has a slight decrease in accuracy. This is due to a decrease in accuracy for three pathways, L-Homoserine to 2-Oxobutanoate, Cholesterol to Glycochenodeoxycholate and L-Lysine to Acetyl-CoA. All of the other pathways have the same accuracy in both cases. This decrease occurs because the degree of the compounds change when reversibility information is incorporated and thus the weighting changes and the top 10 pathways change, in this case causing worse performance.

The use of LPAT-half was motivated by an observed disadvantage with LPAT-max. While LPAT-max works in many cases, it occasionally results in long and unwieldy pathways because they conserve more carbons than the known pathway. Thus, the idea behind LPAT-half is to identify pathways that transfer a significant portion of carbon atoms while allowing flexibility for carbons to leave the pathway. When LPAT-half is used for pyruvate to L-leucine, the known pathway is recovered. A comparison of using the max versus half carbon



(a) NeAT Test Set



(b) Amino Acid Test Set

Figure 5.1 : Average accuracies of linear pathway finding methods evaluated using (a) NeAT Test Set and (b) Amino Acid Test Set.

approach, while holding the weighting scheme fixed, reveals a slight improvement for using the the half carbon in the NeAT test set and a larger improvement in the amino acid test set. A further breakdown of the differences between LPAT-max and LPAT-half performances can be found in Table 5.1. Only two pathways had reduced accuracy when using LPAT-half compared to LPAT-max, while 16 pathways had improved accuracy. This result demonstrates a common theme in metabolic pathfinding that while different approaches may on average improve the performance, it is rare that the improvement occurs across the board.

Looking at the improvement of accuracy compared to the "short, no carbon" method, the results demonstrate that the improvement using LPAT-max and LPAT-half is similar to using the weighting schemes. Interestingly, there is no further systematic improvement when the weighting schemes are used in LPAT. Further analysis of the results points towards two factors that may account for this observation. The first is that if one excludes the short, no carbon and short, one carbon methods, most pathways are found with an accuracy of one. Therefore, the maximal performance values may be dominating the average accuracy values and minor differences may not be as evident. This is especially true for the amino acid test set, where 74% and 80% of pathway experiments find pathways of accuracy one without and with reversibility data, respectively. In the NeAT test set these values are lower at 58% and 60%, respectively. The second is that the purpose of the weighting schemes and the LPAT approaches is to avoid spurious connections. However, if these connections are already eliminated using the carbon tracking, the weighting scheme will not necessarily further improve the results. Additionally, if the weighting scheme causes the search to avoid a highly connected compound in the known pathway the carbon tracking will not correct this. While no clear cut examples of the weighting decreasing performance occurred in the test sets, a case study of pyruvate to 3-hydroyxpropanoate in Section 5.4.2 highlights the potential disadvantages of using weighting schemes.

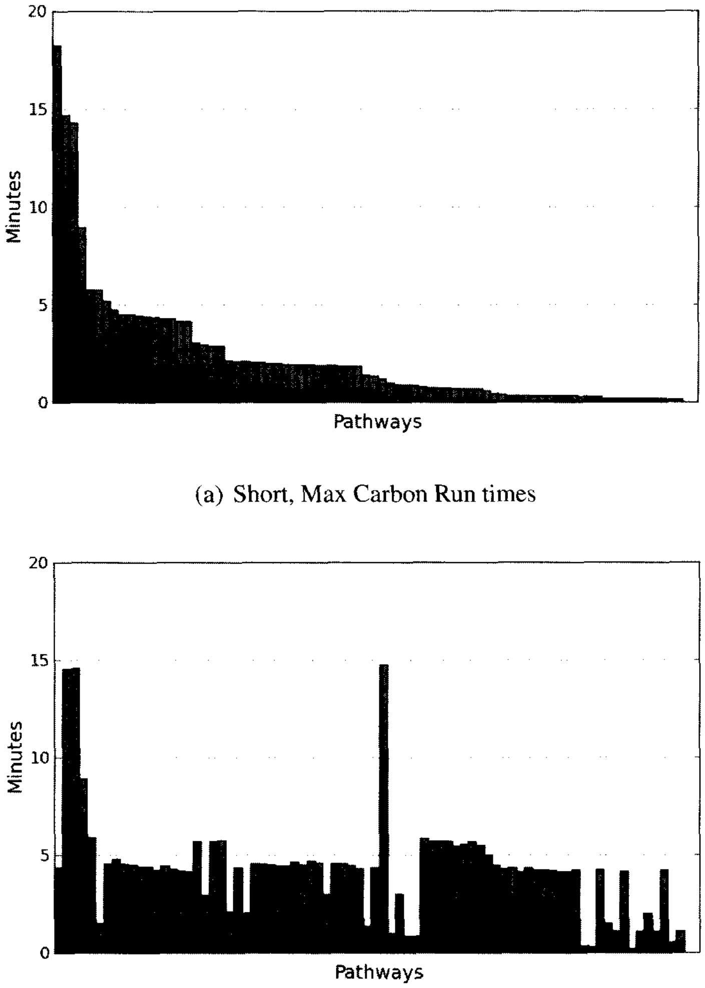
Table 5.1 : Pathways where there was a change in accuracy when using LPAT-half versus LPAT-max, when both use the reversibility data. Results above the double line are from the amino acid test set; below the line are from the NeAT test set. Most of the pathways saw a large increase in accuracy when using LPAT-half versus LPAT-max. Only two pathways, highlighted in gray, resulted in a decrease of accuracy when using LPAT-half versus LPAT-max.

Start Compound	Target Compound	Weight Type	Max Carbon Accuracy	Half Carbon Accuracy
L-Phenylalanine	L-tyrosine	context	1	0.476
Pyruvate	L-Leucine	short	0.582	0.8
Pyruvate	L-Leucine	degree	0.166	1
Pyruvate	L-Leucine	context	0.546	1
Pyruvate	L-Isoleucine	short	0.602	1
Pyruvate	L-Isoleucine	degree	0.602	1
Pyruvate	L-Isoleucine	context	0.556	1
L-Glutamine	L-Arginine	short	0.505	0.752
L-Aspartate	L-Threonine	short	0.287	0.636
L-Lysine	Acetyl-CoA	context	0.146	0.478
L-Lysine	Acetyl-CoA	degree	0.144	0.239
L-Lysine	Acetyl-CoA	short	0.37	0.597
alpha-D-Glucose 6-phosphate	Pyruvate	short	0.543	0.747
L-Arginine	Succinate	degree	0.264	0.543
L-Homoserine	2-Oxobutanoate	context	0.778	0.762
Pyruvate	Acetyl-CoA	short	0.274	0.686
Pyruvate	Acetyl-CoA	degree	0.274	0.686
Pyruvate	Acetyl-CoA	context	0.274	0.686

The experiments run on the linear pathways also demonstrate the efficiency of LPAT. While there could be a theoretical combinatorial explosion in the search, this is never seen in practice and all of the test cases run on the order of minutes or less. Two bar graphs corresponding to the run times for both test sets using the "short, LPAT-max" and "short, LPAT-half" approaches can be found in Figure 5.2. While it is difficult to tease out all of the factors affecting run time, one general principle is that for a given start and target compound the run time is inversely proportional to the number of carbons tracked. This is made clear by sorting the run times of "short, LPAT-max", shown in Figure 5.2(a), and then plotting the run times of "short, LPAT-half", in the same order, shown in Figure 5.2(b). For almost every pathway, the runtime increased when using half the number of carbons as the minimal number of atoms to conserve in LPAT. In the few pathways where there this did not occur, such as the first bar, the first run of LPAT, given the maximum number of carbons to conserve, could not find any pathways. Therefore, the number of carbons was decremented by one and LPAT was run again to find pathways. Since the half carbon approach started out with less carbons, it only had to call LPAT once, resulting in shorter run times.

5.4.2 Pyruvate to 3-hydroyxpropanoate

While the weighting schemes have similar performance to the LPAT-max and LPAT-half approaches, they are heuristics that can perform poorly if the pathways require the use of highly connected compounds. The results of finding pathways between pyruvate and 3-hydroyxpropanoate (3HP) demonstrate the pitfalls of relying on weighting schemes. While weighting schemes perform poorly on this pathway, it was chosen as a test case initially because the high interest in the production of 3HP using metabolic engineering approaches makes it an excellent test case for metabolic pathfinding approaches [49]. 3HP is used in a number of important industrial applications such polymer production as well as being a



(a) Short, Max Carbon Run times

(b) Short, Half Carbon Run times

Figure 5.2 : Run times over both test sets for (a) short, LPAT-max and (b) short, LPAT-half. The short, LPAT-max pathways are sorted by run time and the corresponding run times for the pathways using short, LPAT-half are plotted in the same order. The green represents the total time used for the exploration of G_{am} and the magenta represents the total time for obtaining the k shortest paths from the auxiliary graph.

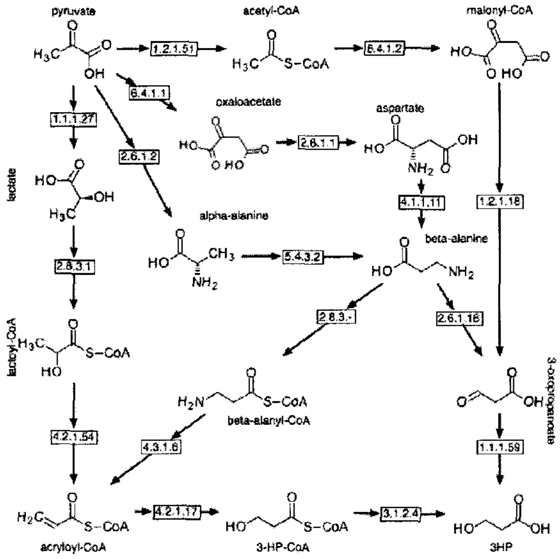


Figure 5.3 : Known pathways for 3HP obtained from [49].

precursor to number of other valuable compounds such as acrylic acid or acrylamide [120]. Therefore, a number of pathways for 3HP biosynthesis in several different microorganisms, often starting from pyruvate, have been discovered and analyzed. Several of these pathways, constructed in microorganisms using recombinant techniques, have resulted in patents [115, 106]. Figure 5.3 depicts the known pathways between pyruvate and 3HP obtained from a recent study [49].

Pyruvate and 3HP were used as the start and target for all of the linear pathway approaches described in the previous section. In recovering the known pathways, the short, LPAT-half performed the best; three different known pathways were contained in the top ten pathways returned. The pathway through oxaloacetate and beta-alanyl-CoA was the 100th pathway returned. None of the approaches were able to identify the pathways through alpha-alanine because the reaction between alpha-alanine and beta-alanine are not contained in KEGG. The LPAT-max approach did not perform as well, only finding the pathway

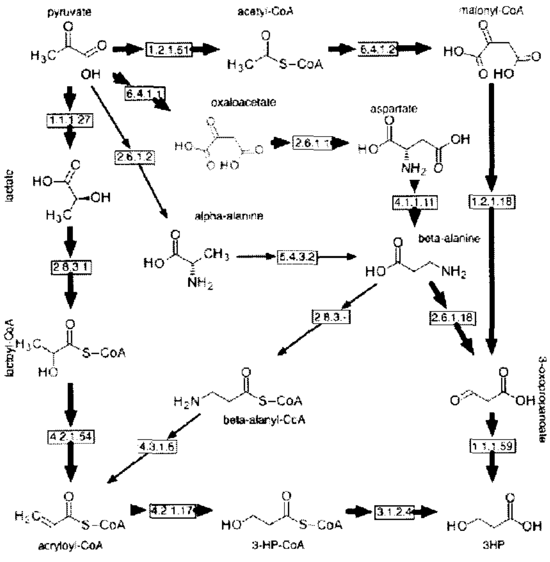


Figure 5.4 : Pathways for pyruvate to 3HP found in the top ten pathways returned by "short, LPAT-half".

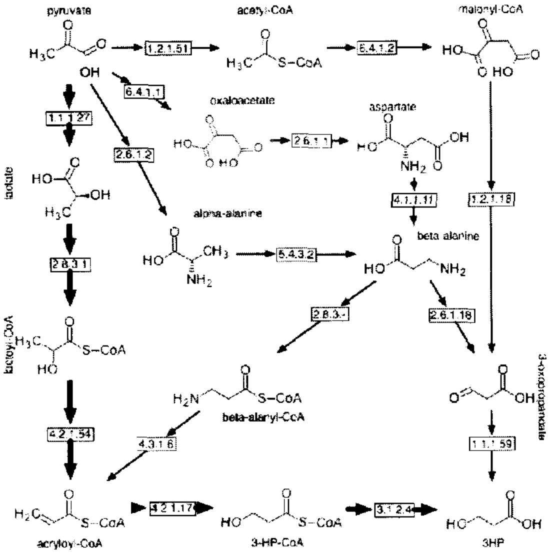


Figure 5.5 : Pathway from pyruvate to 3HP through lactate. This is the only pathway found using LPAT-max, as well in the top ten pathways when degree or context weighting in conjuction with any carbon tracking approach.

through lactate highlighted in Figure 5.5. This result occurs because the pathways through lactate conserves three carbons, the maximum amount, but the other pathways only conserve two carbons, due to a release of carbon dioxide. As this issue with maximizing the number of carbons was also observed in the test sets, it lends credence to preferring the half carbon approach over the max carbon approach.

When either the degree or context weighting were used, irregardless of the carbon tracking approach they were paired with, the only known pathway found in the top ten results was through the lactate pathway, shown in Figure 5.5. Using degree weighting, not only were the other known pathways not found in the top ten pathways, but they were not found in the entire returned set of pathways. The context weighting scheme, unlike the degree weighting, was able to find the pathway through oxaloacetate and 3-oxopropanoate and caused them to be ranked 92, 88 and 58 using no, one and LPAT-half respectively. The context weighting scheme arguably performed better finding the pathway through oxaloacetate and beta-alanyl-CoA as compared to the "short, LPAT-half" approach as it returned it ranked 74, 71 and 43 using no, one and LPAT-half respectively. However, the context weighing scheme did not return the pathway through acetyl-CoA in any of the pathways returned using context weighting.

Both the degree and context weighting scheme fail to find the pathway through acetyl-CoA because acetyl-CoA is involved in a large number of reactions. For example, using the "degree, LPAT-half" approach, the last path in the set of 323 return pathways had a weight of 184. However, the degree of acetyl-CoA is 264. The computed 3HP pathways demonstrate that when intermediate compounds are of high weight, the atom tracking approach is better equipped to find the known pathways. The atom tracking approach is able to avoid spurious connections without relying on a heuristic weighting scheme, but is flexible enough to be used with one if so desired. Furthermore, by utilizing atom tracking branched pathways can be identified using graph-based approaches, as demonstrated in the following sections.

5.4.3 Linear Pathway Cluster Analysis

For most analysis of pathfinding results for linear pathways, only the top ranking ones are analyzed. However, it is interesting that such a large number of pathways are found and raises a number of questions about the pathways not manually examined. One way to begin answering these questions is to cluster the linear pathways to understand how similar or different the pathways are. Applying a clustering method requires a distance measure between pathways. A version of levenshtein distance, or edit distance, was chosen because it incorporates the notion of order along the pathway and it is relatively quick to compute [44].

To calculate the levenshtein distance between two pathways, only the compounds along the pathway were used and the "edits" are on the compound level. This can be thought of as computing the levenshtein distance of two strings representing each pathway built by assigning a letter to each unique compound and adding the letters to the string in the order they appear in the pathway from start to target compound. So in the case of KEGG, the "alphabet" would need to contain 6,002 letters, each one representing a different compound. No chemical similarity information was taken into account, each edit is considered as having the same weight.

The "agnes" agglomerative clustering method from the R language was used to cluster the linear pathways between a given start and target compound [94], using the levenshtein distance as described. To validate the idea of clustering the set of linear pathways three examples were chosen: pyruvate to L-lysine conserving 2 carbons, G6P to L-tryptophan conserving 4 carbons and pyruvate to 3HP conserving 2 carbons and. The pathways from G6P to L-tryptophan are expected to have the least amount of variation because it is made primarily through the shikimate pathway and is not known to have any major variations. As discussed in the previous section, pyruvate to 3HP should have more variation because several alternative pathways. The pyruvate to L-lysine pathways should also contain a lot of variation because there are two general pathways for L-lysine, through asparate and homocitrate, and each of those have more minor variants. The clustered distance matrices for each set are plotted as heat maps in Figures 5.6 for pyruvate to L-lysine, 5.7 for G6P to L-tryptophan, and 5.8 for pyruvate to 3HP, with red meaning the most dissimilar and blue meaning the most similar.

The distance matrices for each set of linear pathways, even with a relatively simple clustering technique, clearly shows that there is structure in each set of linear pathways and that the pathways for L-lysine and 3HP are more variable than those found for L-tryptophan. To further assist in the interpretation of the clustered results, the known pathways were identified and are labeled on the distance matrices. Then, the pathways belonging to the same cluster as the known pathways can be extracted and manually examined and analyzed to understand how they relate to the known pathway.

LPAT returned 318 pathways conserving 2 carbons from pyruvate to lysine and they were all used in the cluster analysis. The clustering of pyruvate to L-lysine pathways is illustrated in Figure 5.6 and has a relatively clear interpretation when the known pathways are labeled. Even without the known pathway labels, the distance matrix reveals that there are two major classes of pathways, which then have variations within them. Using the known pathways, it is found that the large cluster in the upper right of the distance matrix corresponds to pathways related to the aspartate pathways. The middle, smaller cluster corresponds to pathways related to the homocitrate pathway. The other small clusters appear to contain pathways that take very long routes between pyruvate and asparate or homocitrate.

LPAT returned 14,533 pathways conserving 4 carbons from G6P to L-tryptophan and the

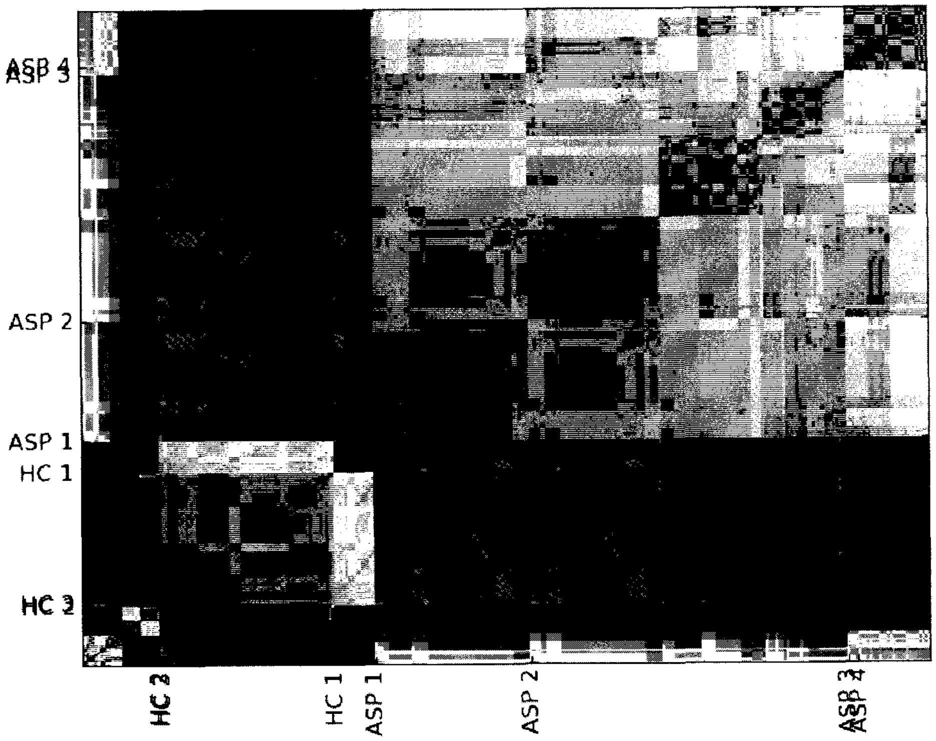


Figure 5.6 : Clustered distance matrix for 318 pathways found using LPAT from pyruvate to L-lysine that conserves at least two carbons. ASP 1-4 corresponds to the four known pathways through aspartate and HC 1-3 corresponds to the three known pathways through homocitrate.

top 1,000 pathways were taken for the cluster analysis. The clustered distance matrix for the tryptophan pathways, illustrated in Figure 5.7, reveals that there is a high level of similarity between the pathways. This is the expected result because tryptophan is not known for having a wide variation of biosynthetic pathways. The known tryptophan biosynthesis through the shikimate pathway is labeled on the distance matrix and it belongs to the smaller cluster in the top right. Inspection of the pathways belonging to the larger cluster in the lower left indicates that they mostly correspond to pathways that find a two reaction shortcut from 3-dehydroshikimate through protocatechuate and 4-hydroxybenzoate to chorismate; rather than the four reactions typically utilized by the shikimate pathway from 3-dehydroshikimate to chorismate.

The top 1,000 pathways out of 1,464 from pyruvate to 3HP conserving 2 carbons, found by LPAT, were used for the cluster analysis. The resulting clustered distance matrix is illustrated in Figure 5.8. While there are identifiable clusters around each of the known pathways, the clustered distance matrix for 3HP is not as well structured as the ones for the lysine and tryptophan pathways, and thus the interpretation is not as clear. This result may be due to several different factors such as, the 3HP pathways may not be as well suited for agglomerative clustering, the distance measure, the increased number of pathways used for clustering or there really are many distinct pathways for making 3HP. Initial manual inspection seems to indicate that there is likely a better clustering of the pathways than is provided by the agglomerative pathways using the levenshtein distance. For example, many of the pathways containing acetyl-CoA that appear similar are not clustered together, which may be the cause of in the relatively unstructured clusters in the lower left.

The initial clustering results presented in this section should indicate that it may be worthwhile to analyze the resulting linear pathways as a whole, instead of just inspecting the top ranking pathways. By clustering the pathways, interesting patterns and pathways may be

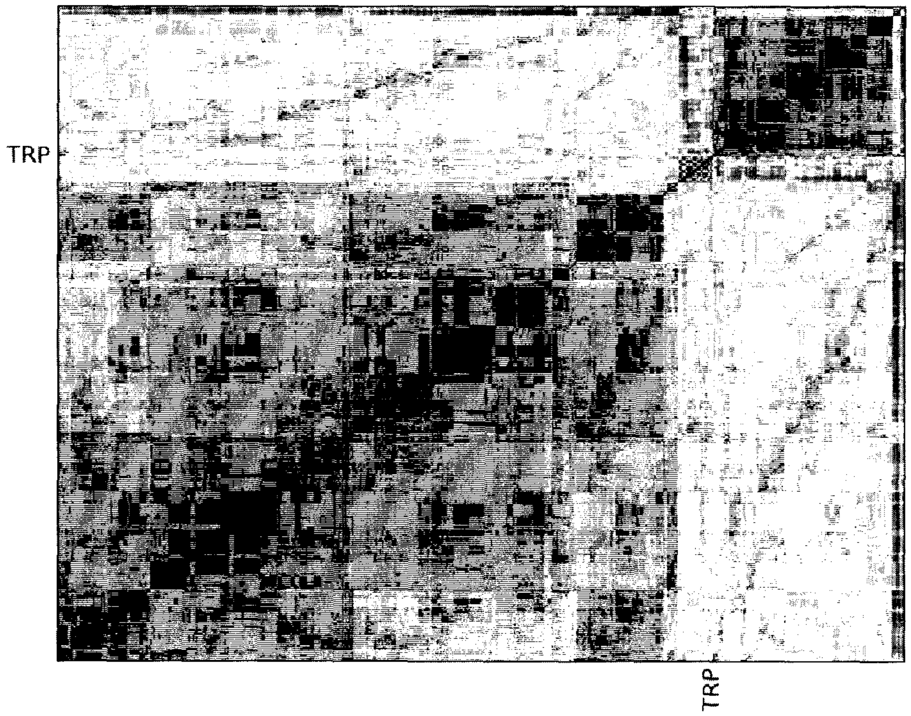


Figure 5.7 : Clustered distance matrix for 1,000 shortest pathways found using LPAT from G6P to L-tryptophan that conserves at least four carbons. TRP corresponds to the known tryptophan pathway through shikimate.

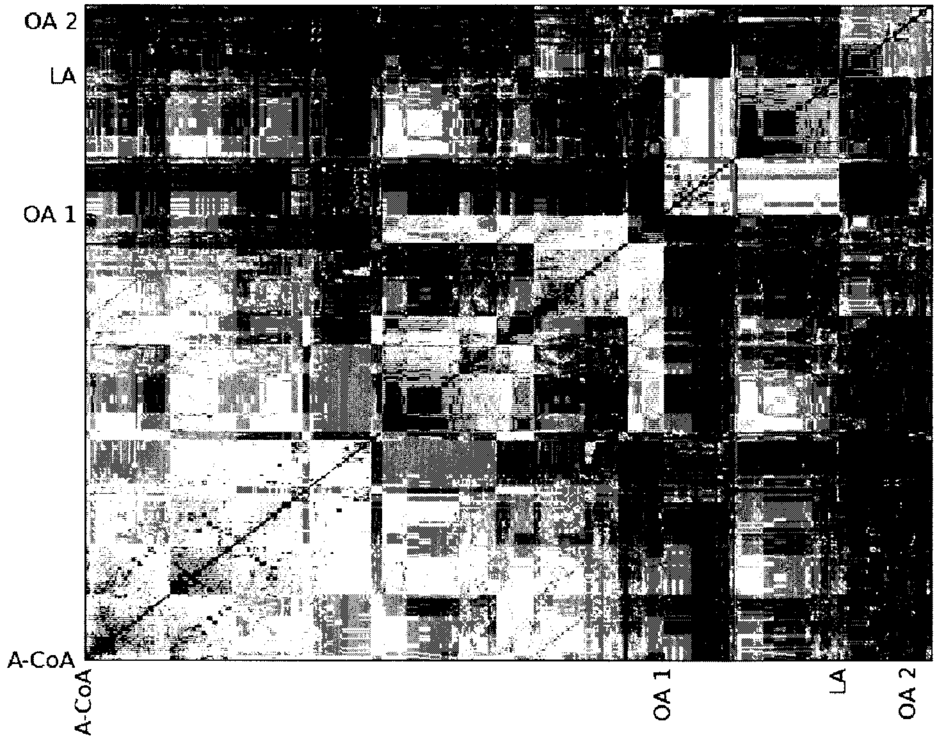


Figure 5.8 : Clustered distance matrix for 1,000 shortest pathways found using LPAT from pyruvate to 3HP that conserve at least two carbons. A-CoA corresponds to the known pathway through acetyl-CoA, OA 1 corresponds to the known pathway through oxaloacetate and 3-oxopropanoate, LA corresponds to the known pathway through lactate and OA 2 corresponds to the known pathway through oxaloacetate and beta-alanyl-CoA.

revealed that would not have been found by examining each pathway individually. While it is difficult to tease out all of the different properties that the cluster analysis may reveal, the analysis lends itself well to interactive inspection of the resulting set of pathways as a whole and may be important for use in future visualizations. Additionally, there are many clustering methods and distance measures to choose from and one future direction, highlighted by the 3HP case, is to study the affect of using different methods and measures. Furthermore, extending the cluster analysis to branched pathways is likely to provide interesting results, but what distance measure to use for branched pathways is an open question and provides fodder for future work.

5.5 Branched Pathway Results

Due to the emphasis on finding linear pathways in metabolic networks, there are very few test cases for branched metabolic networks in the literature. Therefore, the performance of the branched pathway finding algorithms is evaluated on a manually selected set of branched metabolic pathways. The pathways were selected based on biological interest, especially target compounds with industrial or medical applications, and how well studied they are to enable evaluation of the quality of the pathways returned by the algorithms. All of the pathways were tested with both BPAT-S and BPAT-M, described in Chapter 4, utilizing the reversibility data from KEGG as described in Section 5.2. A qualitative analysis is presented for the computed pathways based on what is known about the functioning of the branched pathways. For a number of pathways, α-D-Glucose 6-Phosphate is used as the starting compound because it is what D-Glucose, a common carbon source, is converted into upon being imported into the cell. A list of the start and target compounds tested, along with the number of carbons conserved and the total size of the top ranked pathway found by BPAT-S or BPAT-M can be found in Table 5.2. The resulting pathways found by BPAT-S and BPAT-

Table 5.2 : Branched pathways, listed in the order they appear in this section, with the number of carbons conserved and the total size of the top ranking pathway returned by BPAT-S or BPAT-M.

		BPAT-S Top Pathway		BPAT-M Top Pathway	
Start Compound	Target Compound	Number of Carbons Conserved	Total Size	Number of Carbons Conserved	Total Size
Sn-glycero-3-phosphocholine	L-threonine	4	10	4	10
Chorismate	(*S*)-norcoclaurine	16	10	16	10
G6P	Stachyose	24	15	24	15
G6P	L-Tryptophan	11	28	n/a	n/a
G6P	(-)-Carvone	9	37	6	26
G6P	Inosine Monophosphate	8	24	6	22
G6P	Streptomycin	14	36	12	23
G6P	Penicillin	9	30	6	21
G6P	Cephalosporin C	15	48	9	39
G6P	Erythromycin	34	95	33	86
G6P	Lycopene	21	102	24	136

M demonstrate the ability of the algorithms to find known branched metabolic pathways and highlight a number of biologically interesting properties. The results sometimes contain biologically unrealistic reactions, likely due to both the high computational and biological complexity, but this provides inspiration for future improvements to the algorithms.

Unless otherwise specified, BPAT-S was used without limits on the number of branches for each seed pathway; BPAT-M was run with *r* set to 1,000 and the number of pathways associated with each target atom marking was limited to 5,000. In all of the branched pathway figures, compound nodes are ovals, mapping nodes are boxes containg the KEGG RPAIR ID and associated EC numbers and reactions are diamonds containing the KEGG REACTION ID and associated EC numbers. Each edge in the branched pathway figures corresponds to one molecule, for example if there are four edges from G6P that means that the pathway utilizes four molecules of G6P. Many of the pathways are long and have been split into two in order to fit on the page better, in these cases numbers are provided on the figures to help follow the pathways. The following results are roughly sorted by the complexity of the pathways, from least to most complex. The section concludes with a discussion of the branched pathway run times.

5.5.1 Sn-glycero-3-phosphocholine to L-threonine

Threonine is an essential amino acid primarily manufactured by using engineered strains of bacteria, and therefore there has been a major focus toward improving the yield [71]. A strategy to increase yield may include using pathways which transform degradation products, which otherwise might be lost, into the desired product. Sn-glycero-3-phosphocholine is a common degradation intermediate of triglycerides containing eight carbons [47]. Figure 5.9 depicts two resulting branched pathways from the BPAT algorithms starting with sn-glycero-3-phosphocholine to threonine which contains four carbons. In this case, both BPAT-S and BPAT-M returned the same pathways for the top ranked pathways. BPAT-S was given as input to conserve at least two carbons from start to finish and generated 29661 seed pathways, with the shortest seed pathway containing six reactions. The linear pathways used by BPAT-M also conserved two carbons, resulting in three different target atom markings and only one mutually exclusive combination.

Many of the resulting branched pathways split sn-glycero-3-phosphocholine into sn-

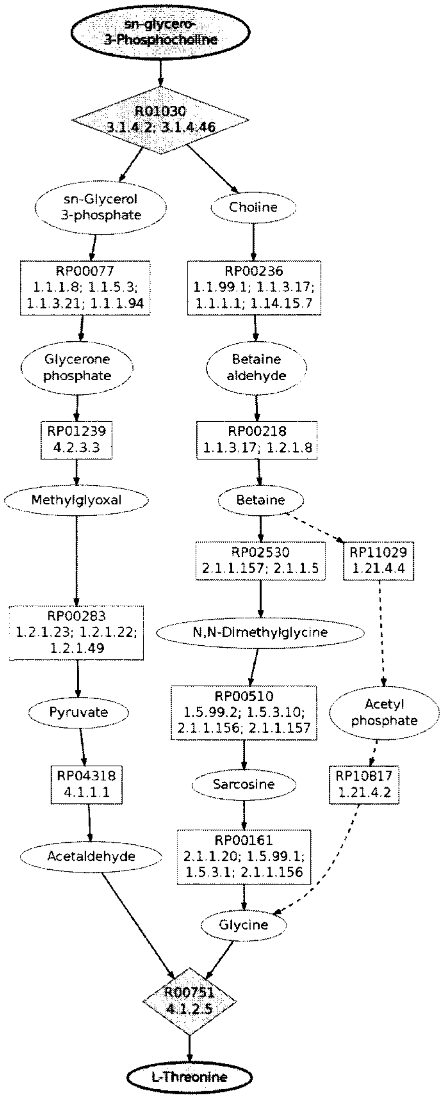


Figure 5.9 : The solid edges shows the known pathway, containing 11 reactions, ranked third in both the BPAT-S and BPAT-M results. The solid pathway is the shortest pathway for sn-glycero-3-phosphocholine to threonine, containing 10 reactions.

glycerol-3-phophate and choline, which begin paths that conserve two carbons and end at acetaldehyde and glycine, which then join to make the four carbon threonine. This general branching scheme is an expected result and no linear pathways were found that conserved all four carbons. The top ranked result is depicted by the dashed edges in Figure 5.9. This result takes an unusual, likely infeasible, shortcut through acetyl phosphate. The reaction from acetyl phosphate to glycine is only observed in the reverse direction, but this is not captured by the reversibility information because the reaction is not contained in any of the pathway maps. However, the reaction from betaine to acetyl phosphate is a feasible reaction that may not typically be considered and could lead to other interesting pathways. Therefore, interesting paths and reactions may be automatically revealed that might normally not be foremost to those familiar with specific subpathways. Additionally, the known pathway from choline to glycine via demethylation is in the next longest set of pathways, with 11 reactions, and is depicted by the solid edges in Figure 5.9. The pathway from sn-glyerol-3-phosphate to acetaldehyde demonstrates the difficulty in finding the balance between finding unusual but likely shortcuts and very unlikely shortcuts. In this pathway, most likely pyruvate is generated from glycerone phosphate via glycolysis instead of through methylglyoxal. However, the overall scheme returned by the branched pathway search is correct and potential ways to help address shortcuts around standard pathways like glycolysis are discussed in the last chapter.

5.5.2 Chorismate to (*S*)-norcoclaurine

(*S*)-norcoclaurine is a key intermediate in the formation of benzylisoquinoline alkaloids, leading to more complex molecules such as morphine and codeine [80]. Chorismate, produced by the shikimate pathway, is an important precursor for a number of aromatic secondary metabolites, including (*S*)-norcoclaurine [50]. Each molecule of chorismate contains 10 carbons and (*S*)-norcoclaurine contains 16 carbons. Both branched pathway searches conserve 7 carbons and BPAT-S is given two molecules of chorismate as the start compound. This resulted in 1534 seed pathways of length ranging from 4 to 14 reactions. The merge pathway search identified 6 different target atom markings which resulted in 5 different mutually exclusive combinations. The top four ranked branched pathways found by both BPAT-S and BPAT-M correspond to the four known pathways for the synthesis of (*S*)-norcoclaurine. An illustration of these pathways can found in Figure 5.10.

All of the pathways share the same path from chorismate to (*S*)-norcoclaurine through 4-hydroxyphenylacetaldehyde, which is the shortest linear pathway. The other branch through dopamine is two reactions longer and much further down in the list of linear pathways. This means that these pathways may be missed if just the shortest linear pathways are reviewed. However, by searching for branched pathways these longer pathways are incorporated into the branched pathway and rise to the top of results over other unlikely linear pathways. This illustrates that identifying branched pathways may help to eliminate undesirable pathways and reveal important relationships.

5.5.3 α-D-Glucose 6-Phosphate to Stachyose

Stachyose is part of the raffinose family oligosaccharides (RFOs), which are found in many plant species. RFOs can be used for carbon transport and storage and can make up 25-80% of the dry weight of certain plants [110]. Notably, raffinose and stachyose are found in high concentration in soybeans, an important staple crop [30]. Other functions of RFOs are unclear, but studies have indiciated that they may act as protective agents for seeds and against cold temperatures [118, 10]. RFOs have have traditionally been perceived as anti-nutritional because humans and other monogastric organisms lack the enzymes to break them down. However, it has been suggested that RFOs be beneficial to gut flora [117].

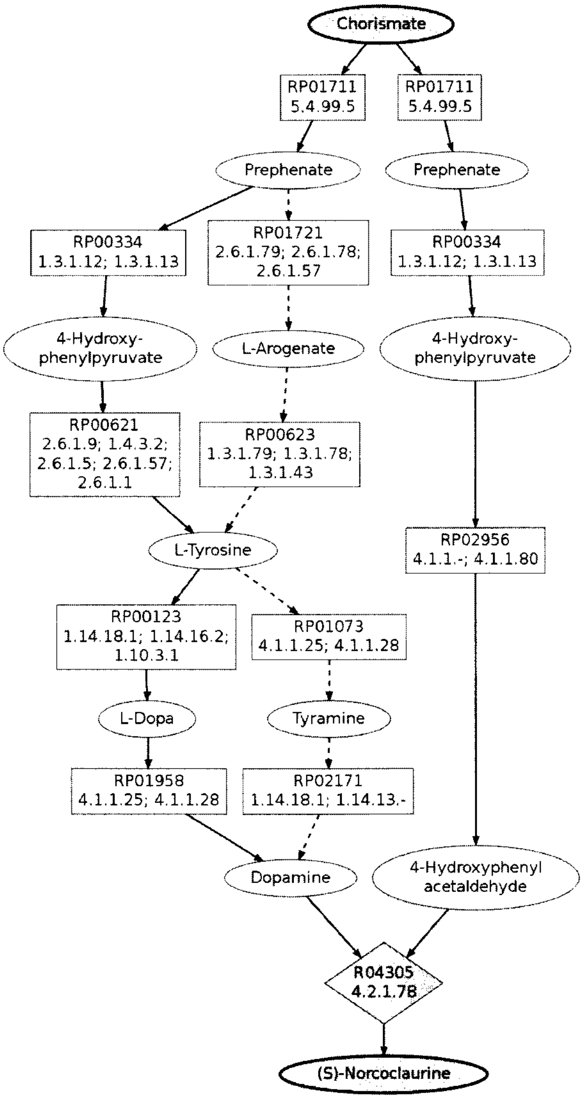


Figure 5.10 : The top four pathways for two molecules of chorismate to (*S*)-norcoclaurine, as identified by both BPAT-S and BPAT-M, merged together with the dashed lines showing how the pathways differ.

Additionally, outcome of lung transplantation was improved when raffinose was included in the lung storage solution, although the specific mechanisms are still being studied [37].

The starting compound for the stachyose branched pathway search was chosen as α-D-Glucose 6-Phosphate (G6P), even though the pathway is typically viewed as starting from sucrose. The top ranking pathway returned using either BPAT-S or BPAT-M was the same and is depicted in Figure 5.11. The computed pathway corresponds to the known stachyose biosynthesis pathway. While the pathway used to produce sucrose may vary in different organisms, the top ranked pathways also primarily vary in the paths from G6P to sucrose. The stachyose pathway is the most topologically complex pathway where the top ranking results from the merge and seed approaches are the same. Both algorithms were given as input to conserve at least 3 carbons for the linear pathway search. BPAT-M identified 14 different target atom markings in the linear pathways, resulting in 19 mutually exclusive combinations. BPAT-S started from four molecules of G6P and tested 8746 seed pathways containing 4 to 11 reactions.

5.5.4 α-D-Glucose 6-Phosphate to L-Tryptophan

Tryptophan, similar to threonine, is an essential amino acid mainly produced by microbial fermentation [71]. The tryptophan biosynthesis pathway is relatively complex, with a number of places where carbons are lost and gained along the way. BPAT-S was initialized with two molecules of α-D-glucose 6-phosphate (G6P). The minimum number of carbons to conserve for the linear pathways was 4, resulting in 14079 seed pathways of length between 13 and 21 reactions. BPAT-M was tried with 3 and 4 carbons to conserve, but neither search resulting in any returned branched pathways. The lack of resulting branched pathways occurs because the known pathway conserves four carbons through the shikimate pathway, which is relatively long, with carbons being added three carbons along the way. Therefore,

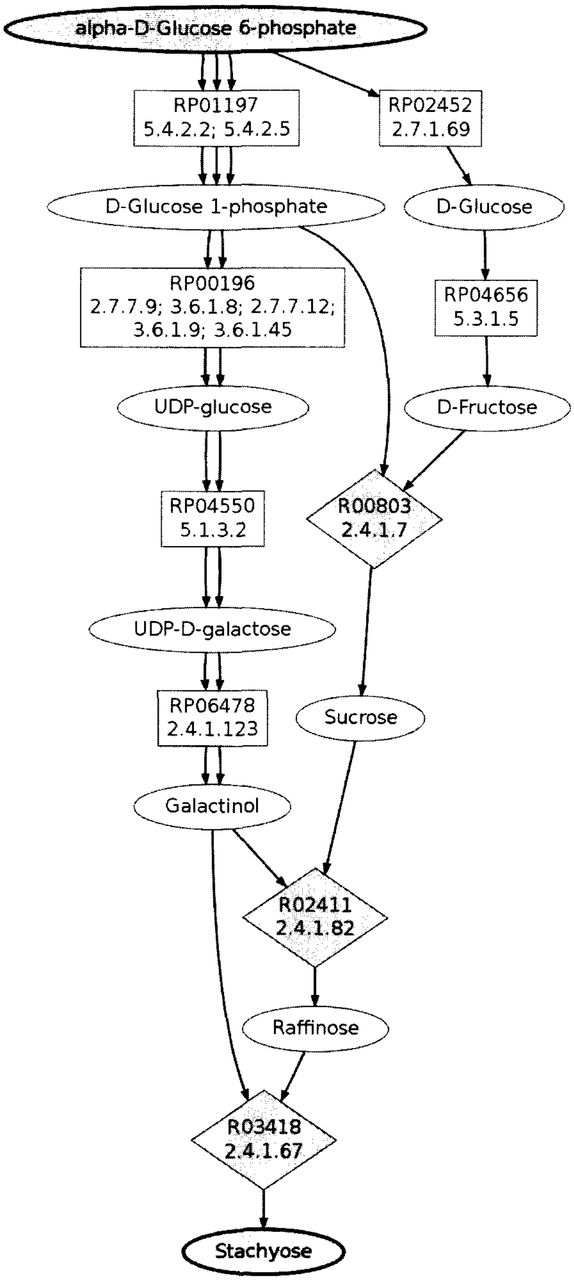


Figure 5.11 : Top ranking pathway returned by both BPAT-S and BPAT-M from G6P to Stachyose, with the linear pathways conserving 3 carbons.

the search with four carbons results in linear pathways that go through the shikimate pathway and have the same target atom marking, and thus cannot be merged. When BPAT-M uses 3 as the minimal number of carbons to conserve, the linear pathways found are so much shorter that the shikimate pathway is not in the set of linear pathways returned and again none of the pathways can be merged. This demonstrates how the performance of the two approaches can widely vary based on the underlying structure of the metabolic pathways. In this case, the tryptophan pathway is more easily discovered by the seed pathway approach.

The results for the tryptophan pathway are greatly improved by using the reversibility information. When considering all reactions reversibile, there are a number of "shortcuts" revealed by the search. Many of these shortcuts utilized reactions from tryptophan degradation that are unlikely to occur in the direction used by the pathway, such as ones through L-formylkynurenine to tryptophan and catechol to anthranilate. However, the top ranked result returned by BPAT-S takes a shorter route from 3-dehydroshikimate to chorismate that is 3 reactions long, rather than the standard 4 reactions of the shikimate pathway. The shorter route uses a reaction from 4-hydroxybenzoate to chorismate that is drawn as reversible on the KEGG pathway maps, but primarily goes from chorismate to 4-hydroxybenzoate.

The shortest pathway that utilizes the reactions in the standard shikimate pathway is ranked forth in the returned pathways, but this pathway utilizes a degradation reaction that typically uses tryptophan as the substrate in the reverse direction as an attachment point for a branch. Therefore, the best result in the top ten pathways returned by BPAT-S is the seventh result, which captures major features of the known pathway, and is depicted in Figure 5.12. As with other results, the two molecules of phosphoenolpyruvate would typically be created via glycolysis, but here they are created by a different scheme because pathways conserving the same number of carbons are ordered by length. Additionally, only one molecule of G6P would be required which cannot be found because the seed pathway approach currently only finds branches off of the seed pathway. At the same time, this result illuminates a number of interesting properties of the tryptophan pathway, such as serine and 5-phospho-α-D-ribose 1-diphosphate both can be made from compounds further upstream. These complex relationships, automatically discovered by the algorithm, are of importance for metabolic pathway analysis and design and may not typically be considered.

5.5.5 α-D-Glucose 6-Phosphate to (-)-Carvone

Plants have evolved to produce a wide variety of secondary metabolites, often used as defense mechanisms to against herbivory or infection [14]. Many of these metabolites have medicinal and other important uses and thus are the target of metabolic engineering approaches to produce them on a larger scale. The monoterpene (-)-carvone is a component of spearmint essential oil and has potential antimicrobial, anticancer and insecticial effects [88, 135, 66]. The known (-)-carvone biosynthesis pathway functions by combining isopentenyl diphosphate (IPP) and dimethylallyl diphosphate (DMAPP) to form geranyl diphosphate. IPP and DMAPP can be produced either by the mevalonate (MVA) pathway or the methylerythritol phosphate (MEP) pathway, which was more recently discovered [72]. The MVA pathway is known to function in eukaryotes, archeabacteria and in the cytosol of plants and the MEP pathway is known to function in eubacteria, green algae and chloroplasts of plants [64]. Three more reactions then transform geranyl diphosphate into (-)-carvone, which contains 10 carbons [74]. The enzyme forming geranyl diphosphate and three enzymes that subsequently drive the chemical reactions to form (-)-carvone have been succesfully installed into *E. coli* in a study to understand it's capability to produce foreign metabolites [20].

BPAT-S and BPAT-M were used to find the (-)-carvone biosynthesis pathway by giving as the start compound α-D-Glucose 6-Phosphate, (-)-carvone as the target compound and 3

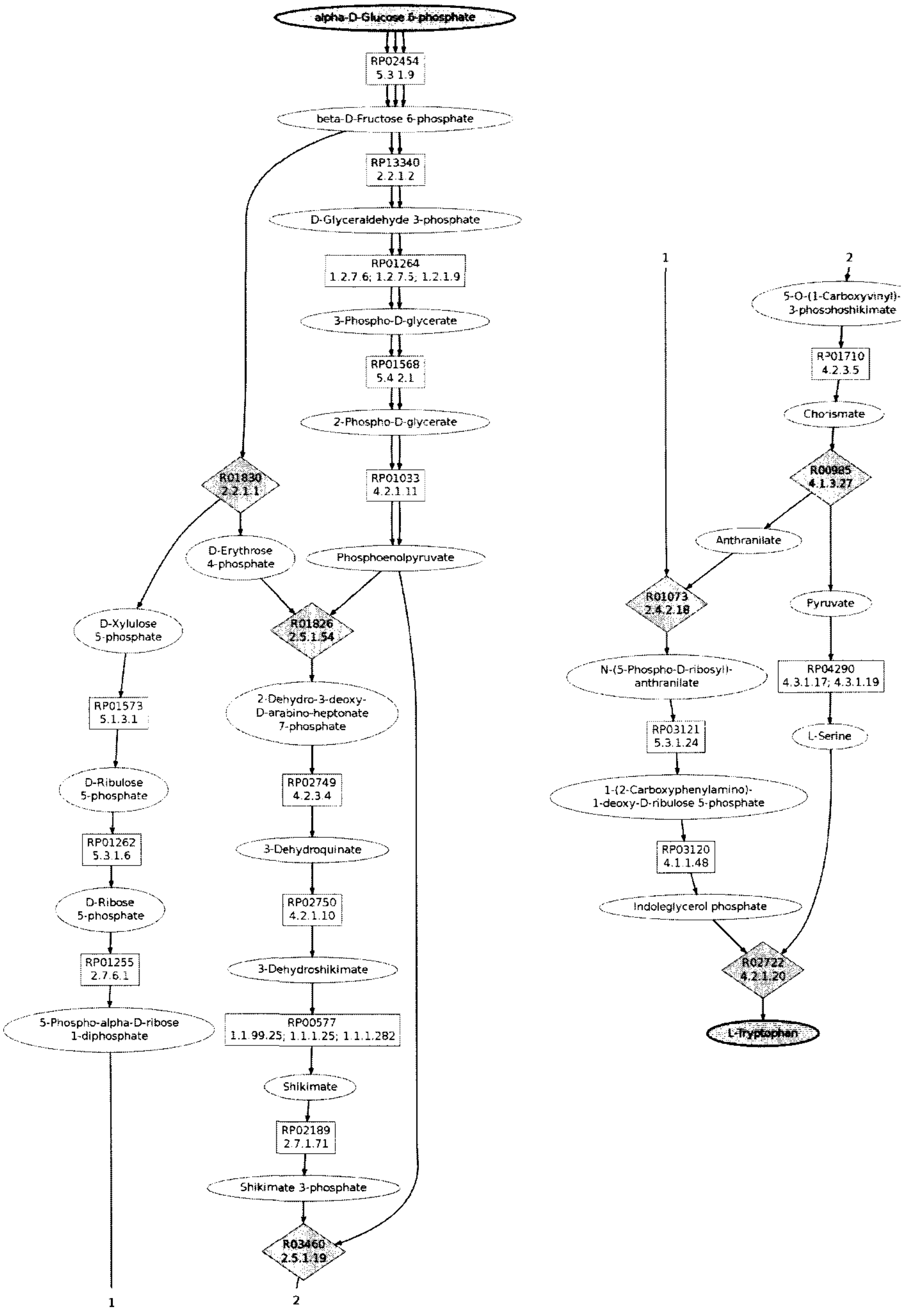


Figure 5.12 : The seventh branched pathway returned by BPAT-S reveals the interesting topology of the tryptophan biosynthesis pathway. The numbers are to assist the reader in following the pathway, as the figure is split in half to fit on the page. BPAT-M is unable to find a branched pathway for G6P to tryptophan.

as the minimal number of carbons to conserved. This resulted in 21976 seed pathways for BPAT-S ranging in length of 14 to 21. With the default limits on the number of pathways in each target atom marking cluster and r for BPAT-M, the search would run out of memory and the results were obtained by setting r to 500 and limit the number of pathways in each cluster is 2500. BPAT-M identified four different target atom markings from the linear pathways that resulted in three different mutually exclusive combinations.

The resulting branched pathways for (-)-carvone are the first example where both BPAT-S and BPAP-M find branched pathways but the resulting top pathways are different. The BPAT-S pathway contained the MEP pathway to make DMAPP and this is used as the seed pathway, the MVA pathway is found as a branch to make IPP. However, the branch is only realistic from acetoacetyl-CoA. This is because acetoacetyl-CoA contributes four carbons to isopentenyl disphosphate through the mevalonate pathway and the branch tries to find a path of four carbons from G6P. Therefore a long, unrealistic linear pathway is found that does conserve four carbons instead of the more realistic branched pathway where acetoacetyl-CoA is synthesized by two molecules of acetyl-CoA. On the other hand, the top ranking pathway from BPAT-M only utilizes the MEP pathway, as it is the shorter of the two pathways, to make both IPP and DMAPP. Overall, this is a more realistic pathway, but it does not reveal the variety of pathways that can be used like the BPAT-S result does.

5.5.6 α-D-Glucose 6-Phosphate to Inosine Monophosphate

Inosine monophosphate (IMP) is an important intermediate in the formation of purine nucleotides and nucleosides. The de novo biosynthesis of IMP from glucose proceeds by first forming the ribose component (PRPP) from D-ribose 5-phosphate, a product of the pentose phosphate pathway and incrementally building the purine ring from a number of donor compounds, with glycine contributing the largest component of two

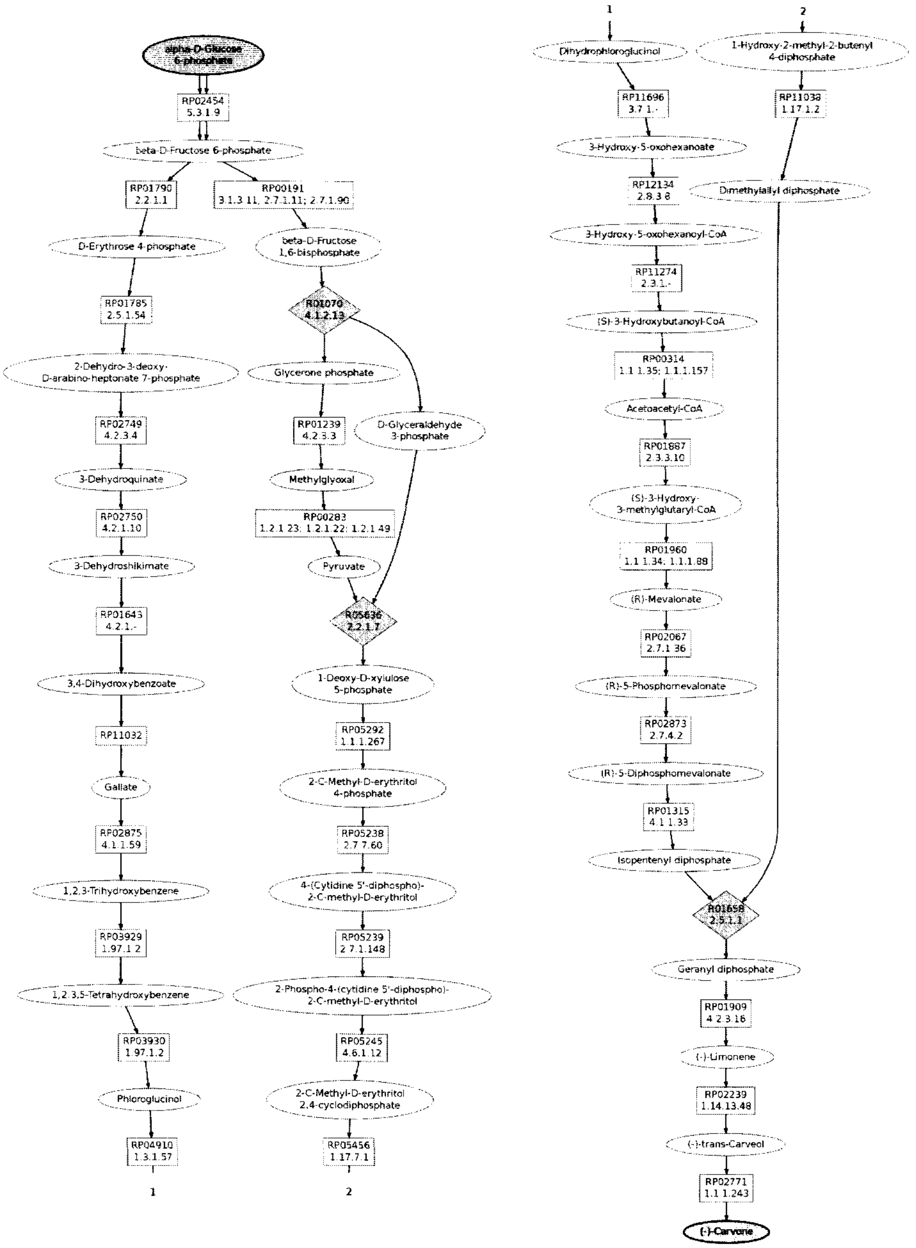


Figure 5.13 : Top ranked pathway returned by BPAT-S for G6P to (-)-Carvone, with the linear pathways conserving 3 carbons.

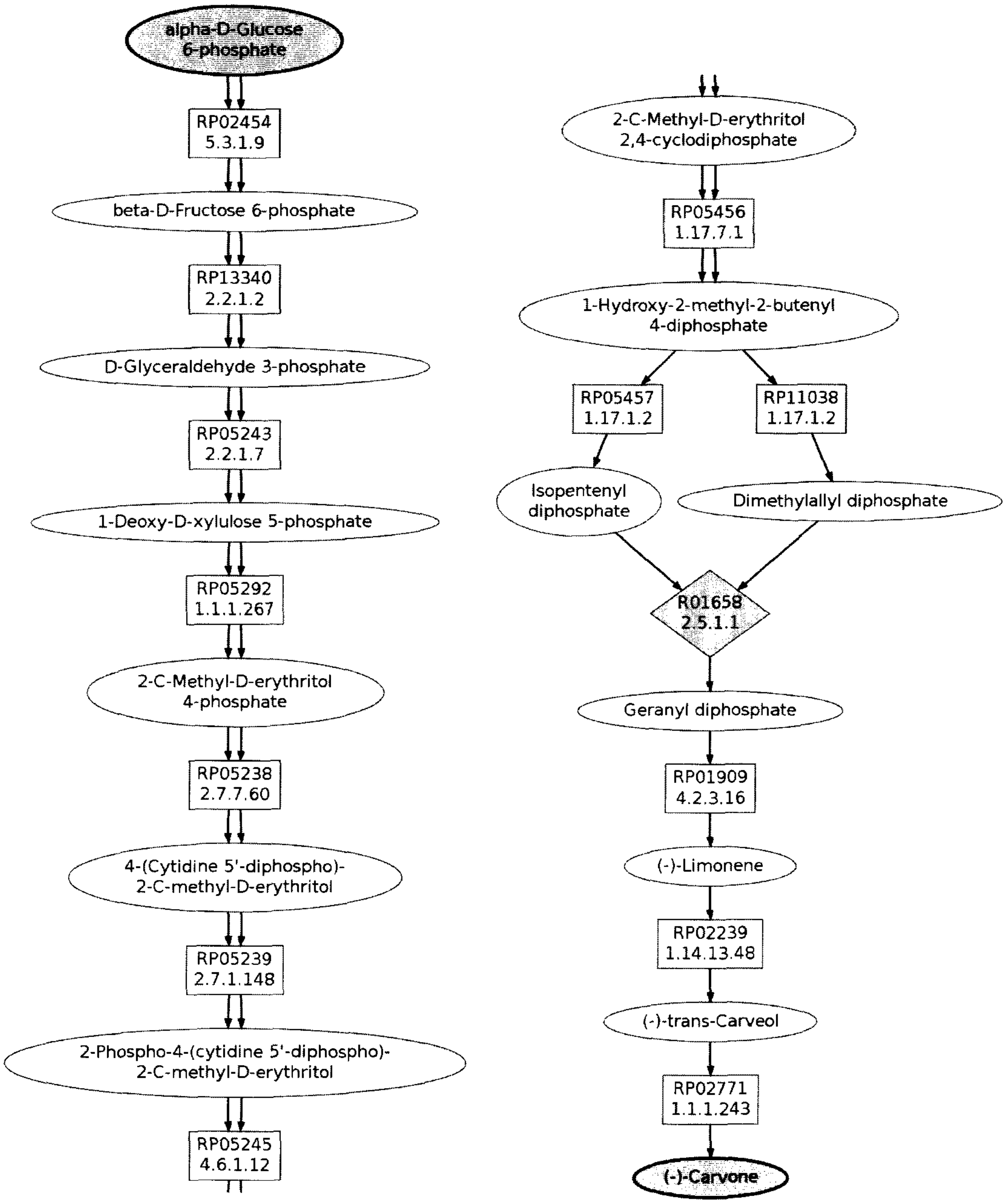

alpha-D-Glucose 6-phosphate
RP02454 5.3.1.9
beta-D-Fructose 6-phosphate
RP13340 2.2.1.2
D-Glyceraldehyde 3-phosphate
RP05243 2.2.1.7
1-Deoxy-D-xylulose 5-phosphate
RP05292 1.1.1.267
2-C-Methyl-D-erythritol 4-phosphate
RP05238 2.7.7.60
4-(Cytidine 5'-diphospho)-2-C-methyl-D-erythritol
RP05239 2.7.1.148
2-Phospho-4-(cytidine 5'-diphospho)-2-C-methyl-D-erythritol
RP05245 4.6.1.12
2-C-Methyl-D-erythritol 2,4-cyclodiphosphate
RP05456 1.17.7.1
1-Hydroxy-2-methyl-2-butenyl 4-diphosphate
RP05457 1.17.1.2
RP11038 1.17.1.2
Isopentenyl diphosphate
Dimethylallyl diphosphate
R01658 2.5.1.1
Geranyl diphosphate
RP01909 4.2.3.16
(-)-Limonene
RP02239 1.14.13.48
(-)-trans-Carveol
RP02771 1.1.1.243
(-)-Carvone

Figure 5.14 : Top ranked pathway returned by BPAT-M for G6P to (-)-Carvone, with the linear pathways conserving 3 carbons.

carbons and one nitrogen [79]. The de novo incremental construction begins with 5-phosphoribosylamine and proceeds through 5'-phosphoribosylglycinamide (GAR), 5'-phosphoribosyl-N-formylglycinamide (FGAR), 2-(formamido)-N1-(5'-phosphoribosyl)acetamidine (FGAM), aminoimidazole ribotide (AIR), 1-(5-phospho-D-ribosyl)- 5-amino-4-imidazolecarboxylate (CAIR), 1-(5'-phosphoribosyl)-5-amino-4-(N-succinocarboxamide)-imidazole (SAICAR), 1-(5'-phosphoribosyl)-5-amino-4-imidazolecarboxamide (AICAR) and 1-(5'-Phosphoribosyl)-5-formamido-4-imidazolecarboxamide (FAICAR) which is made into IMP. Plants, animals and microrganisms all perform de novo synthesis of IMP in a similar manner [136]. In addition to the de novo pathway, because of the importance and energenic costs of synthesizing purine nucleotides and nucleosides, there are a number of purine salvage pathways that go through IMP [83]. These salvage pathways makes finding the de novo pathway a challenging case for computational metabolic pathfinding methods.

In the recent work by Pitkänen et al., the ReTrace algorithm is tested on the IMP pathway, using D-glucose as the start compound [91]. The presented pathway found by ReTrace has a general branching scheme that combines hypoxanthine and D-ribose 5-phosphate to form IMP. However, this corresponds to a known salvage pathway and is not the de novo pathway, which the authors note that ReTrace had difficulty finding. While the hypoxanthine pathway is a valid branching scheme, it is unlikely to be the one from glucose. Furthermore, the reactions in the depicted branched pathway between glycine and hypoxanthine are typically degradation reactions that would proceed in the opposite direction needed by the computed pathway. This occurs because no reversibility information was included when testing ReTrace and demonstrates the importance of using reversibility information. Since BPAT-S and BPAT-M are tested using reversibility information it is impossible to do a direct comparison with the ReTrace results.

For the IMP pathway, BPAT-S obtained better results than BPAT-M. The methodology of

BPAT-S enables it to start with a seed pathway for the ribose component and then search for a branch that corresponds to the construction of the purine base. BPAT-S used 25310 linear pathways ranging in length from 5 to 13 reactions. The top ranked pathway found by BPAT-S is depicted in Figure 5.15. In this pathway, due to the use of reversibility information, one of the branches contains the proper sequence of reactions through glycine to AICAR. However, because the original seed pathway utilized a reversible salvage reaction, a strange connection occurs at the join point of this branch. Typically AICAR already has the ribose component incorporated, but in order to attach the branch, the ribose component is removed from AICAR to make 5-amino-4-imidazolecarboxyamide, which is then combined with PRPP from the seed pathway to make FAICAR.

BPAT-M was thwarted by the fact that there are a number of short salvage pathways that conserve carbons from G6P to IMP through the ribose component and therefore the set of linear pathways found by LPAT do not contain the pathways that build the purine base of IMP. BPAT-M was only able to find branched pathways when it used one carbon as the minimal number of atoms to conserve. When conserving one carbon, BPAT-M found 14 different target atom markings resulting in 190 different mutually exclusive combinations. The top ranking result from BPAT-M conserving one carbon is depicted in Figure 5.16. This pathway contains the same seed pathway used by BPAT-S to find the pathway in Figure 5.15 and also the pathway through formate which contributes one carbon. The pathway through formate is found in both the BPAT-S and the BPAT-M result and is interesting because the standard pathway for IMP biosynthesis is folate dependent, typically 10-formyltetrahydrofolate, as the source for the formyl group added to AICAR to form FAICAR [79]. However, archea perform this reaction without folate or modified folates and utilize formate instead [129, 134]. The automatic finding of the formate reactions from archea demonstrates how searching over all organisms can retrieve interesting or non-standard reactions that might

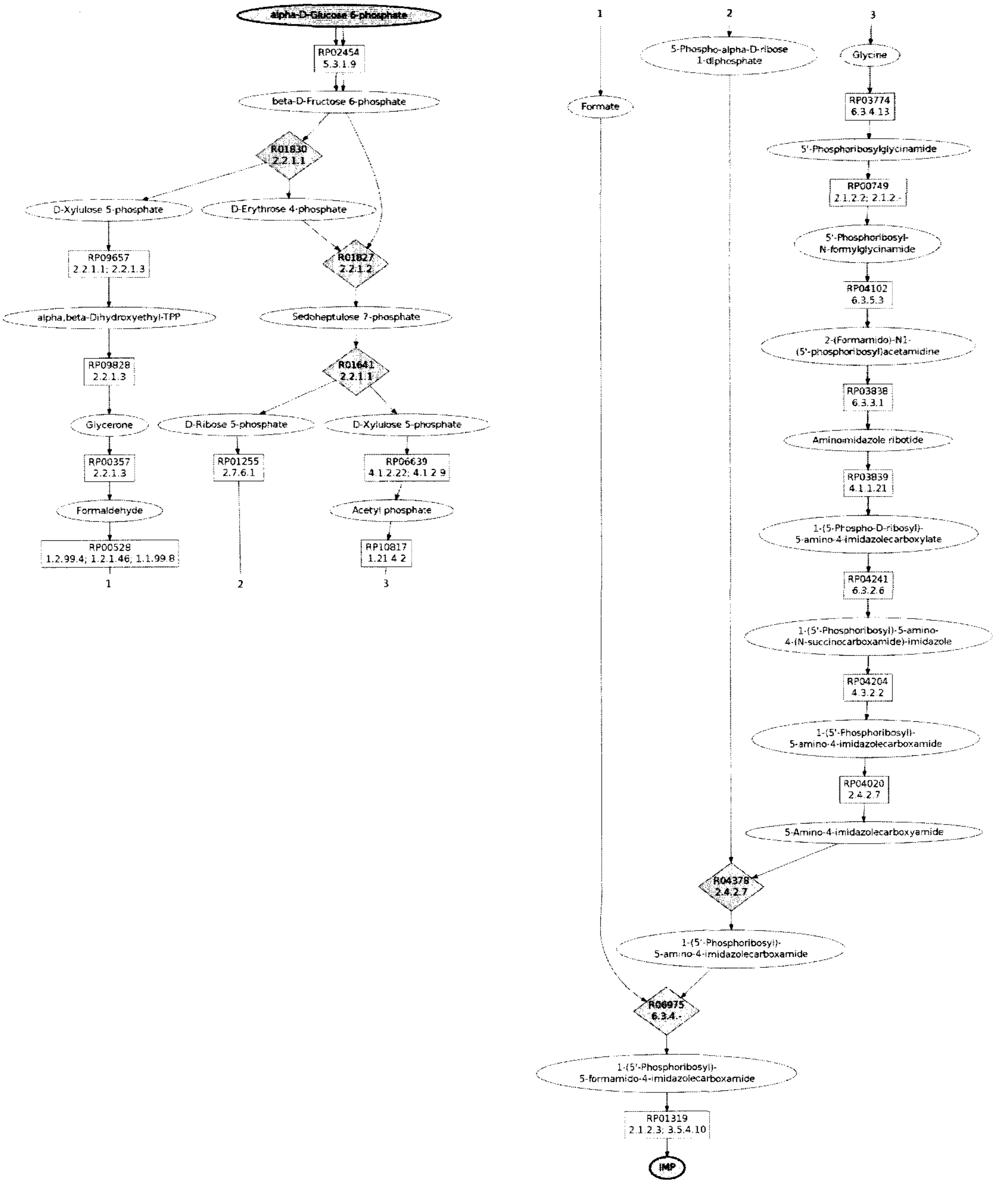


Figure 5.15 : Top ranked pathway returned by BPAT-S for G6P to IMP, with the linear pathways conserving 3 carbon.

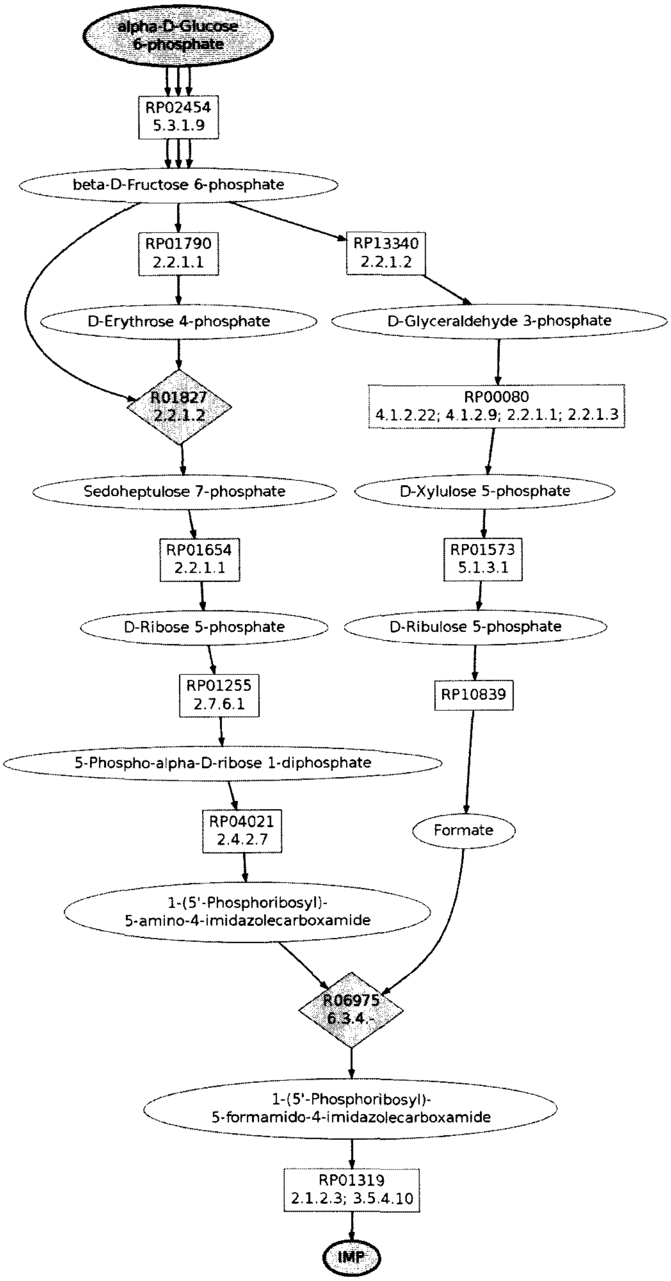

alpha-D-Glucose 6-phosphate
RP02454 5.3.1.9
beta-D-Fructose 6-phosphate
RP01790 2.2.1.1
RP13340 2.2.1.2
D-Erythrose 4-phosphate
D-Glyceraldehyde 3-phosphate
R01827 2.2.1.2
RP00080 4.1.2.22; 4.1.2.9; 2.2.1.1; 2.2.1.3
Sedoheptulose 7-phosphate
D-Xylulose 5-phosphate
RP01654 2.2.1.1
RP01573 5.1.3.1
D-Ribose 5-phosphate
D-Ribulose 5-phosphate
RP01255 2.7.6.1
RP10839
5-Phospho-alpha-D-ribose 1-diphosphate
RP04021 2.4.2.7
Formate
1-(5'-Phosphoribosyl)-5-amino-4-imidazolecarboxamide
R06975 6.3.4.-
1-(5'-Phosphoribosyl)-5-formamido-4-imidazolecarboxamide
RP01319 2.1.2.3; 3.5.4.10
IMP

Figure 5.16 : Top ranked pathway returned by BPAT-M for G6P to IMP, with the linear pathways conserving 1 carbon.

not be considered otherwise.

The IMP biosynthesis pathway underscores the challenges that branched pathway finding algorithms face because of the complexities and nuances that occur in nature. The results of BPAT-S show that a number of these challenges can be overcome to produce reasonable results, but there is still room for future improvement as discussed in the final chapter of this thesis. The IMP pathway is particularily difficult because its synthesis is highly intertwined with a number of important compounds. Both BPAT-S and BPAT-M perform much better on the rest of the pathways included in this thesis because, while the pathway topology may be more complicated, they are secondary metabolites.

5.5.7 α-D-Glucose 6-Phosphate to Streptomycin

Quite a bit of research has been done to to understand and characterize the biosynthetic pathways related to antibiotic production, which coupled with their complex topologies makes them good test cases for branched pathway finding. Streptomycin is an aminoglycoside antibiotic that was isolated in the 1940s and subsequently became the first antibiotic treatment for tuberculosis [84]. Streptomycin, along with the majority of clinicially useful antibiotics, are synthesized by bacteria in the genus *Streptomyces* that primarily live in soil [126]. The pathway for streptomycin biosynthesis is known to be composed of three major pathways resulting in streptidine 6-phosphate, dTDP-L-dihydrostreptose and NDP-N-methyl-L-glucosamine. The three compounds are incorporated in two sequential reactions to form dihydrostreptomycin 6-phosphate, which then becomes streptomycin via two more reactions [38].

BPAT-S and BPAT-M are used to find the streptomycin biosynthesis pathway using G6P as the starting compound and conserving at least three carbons. BPAT-S used 73749 linear pathways identified by LPAT ranging in length from 10 to 18 reactions. BPAT-M found five different target atom markings that resulted in six different mutually exclusive combinations. The top ranking pathways for BPAT-S and BPAT-M are illustrated in Figures 5.17 and 5.18, respectively. The top ranking pathways found by both of the searches contains the known pathways to streptidine 6-phosphate and dTDP-L-dihydrostreptose and subsequently to streptomycin. However, the pathway to NDP-N-methyl-L-glucosamine is missing because it has not been well characterized experimentally and thus not contained in the KEGG REACTION database. Therefore, any search method using the KEGG data would fail to find this pathway and finding the pathway containing the other two pathways is the best that one could expect.

BPAT-S additionally identifies the use of arginine in the synthesis of streptomycin. In each reaction, the amidino group of arginine, containing one carbon, is attached to the precursor compound of streptomycin. While the pathways from G6P to arginine shortcut through CO2, the use of arginine is correct. BPAT-M does not find these branches because they only conserve one carbon and thus are not contained in the original set of linear pathways. This difference between BPAT-S and BPAT-M is a consistent theme through the antibiotic biosynthesis pathway test cases.

5.5.8 α-D-Glucose 6-Phosphate to Penicillin

Penicillin is a widely used beta lactam antibiotic that is synthesized by filamentous fungi and its discovery is an important landmark in the development of the biotechnology industry [31]. Initial work to characterize the biosynthesis pathways for penicillin and related compounds was stymied by difficulties with genetically analyzing the fungi and purifying the relevant enzymes [18]. However, with the great advances in molecular biology these pathways are very well characterized and have become the target of metabolic engineering techniques to improve the production of beta lactam antibiotics [116]. The biosynthesis

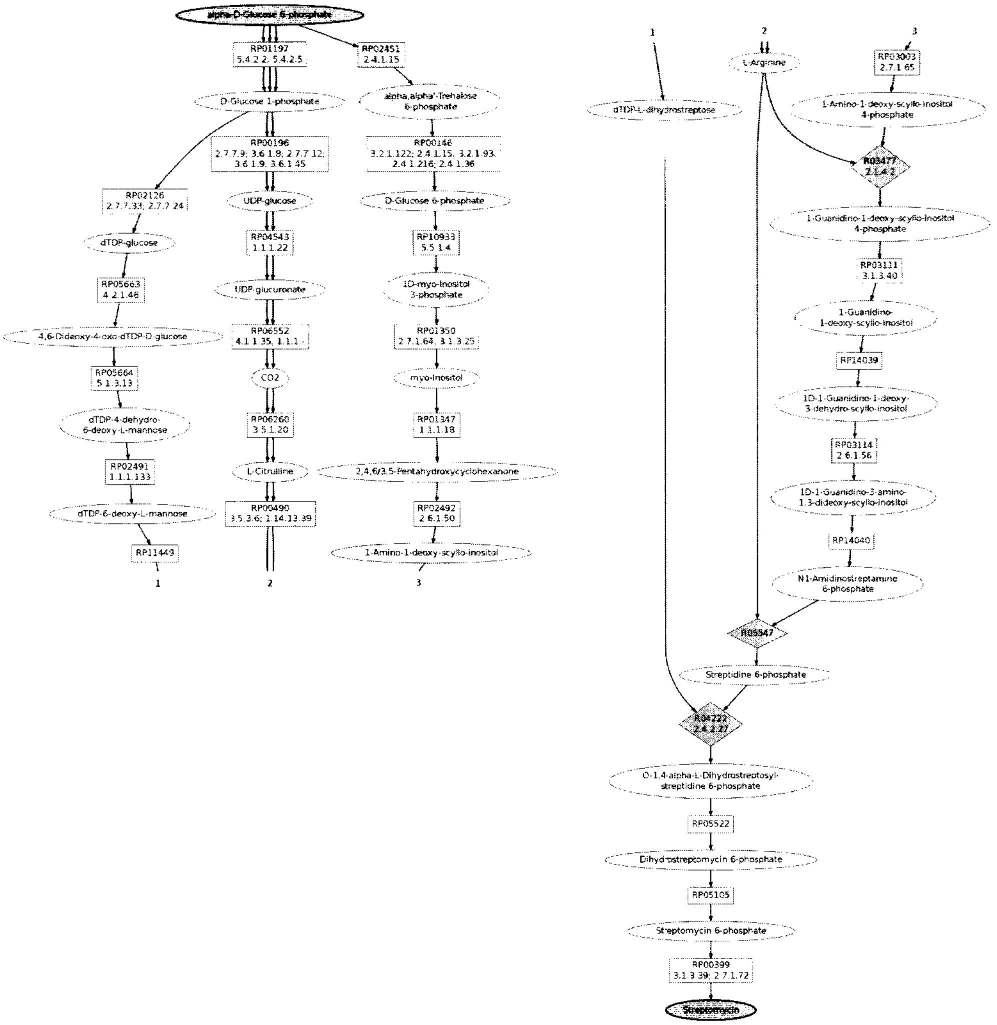


Figure 5.17 : Top ranked pathway returned by BPAT-S for G6P to Streptomycin, with the linear pathways conserving 3 carbons.

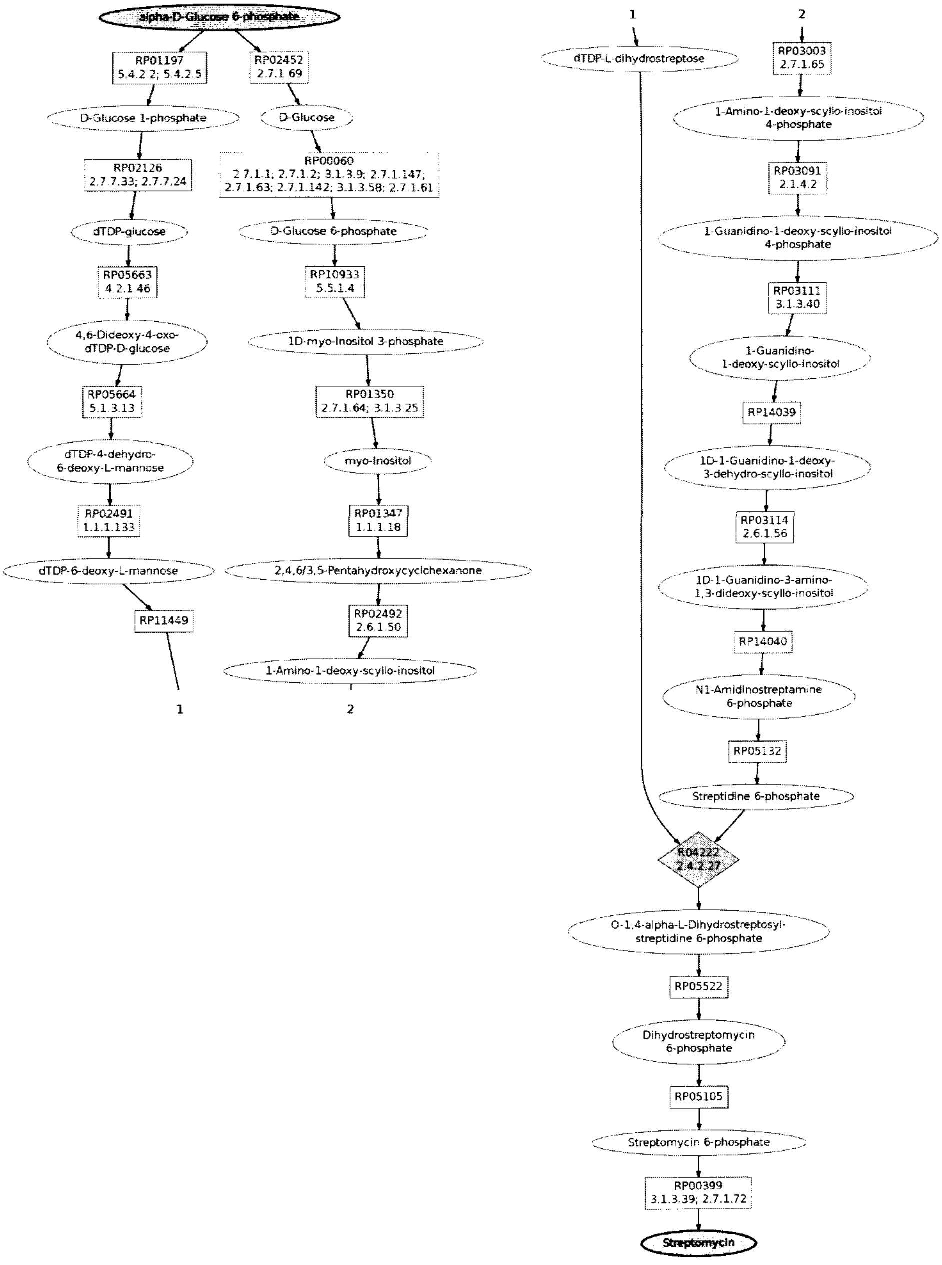


Figure 5.18 : Top ranked pathway returned by BPAT-M for G6P to Streptomycin, with the linear pathways conserving 3 carbons.

of penicillin is typically viewed as beginning with a reaction, catalyzed by ACV synthase, which condenses three amino acids, L-valine, L-cysteine and L-2-aminoadipate to form δ-(L-α-aminoadipyl)-L-cysteinyl-D-valine (ACV) [18]. ACV then becomes isopenicillin N which can subsequently made into a number of compounds in the penicillin family.

The ability of BPAT-S and BPAT-M to find the penicillin pathway was tested by using G6P as the start compound and penicillin as the target compound. The "penicillin" compound in KEGG (ID: C00395) is a generic compound that contains the chemical structure for the beta-lactam ring and the five-membered thiazolidine ring, characteristic of all penicillins. A "R" is used to represent the other chemical structures that can be attached to create different penicillin variants. BPAT-S and BPAT-M were given as input to conserve at least three carbons and the resulting top ranked pathways are depicted in Figures 5.19 and 5.20, respectively. BPAT-S used 71845 linear pathways ranging in length of 9 to 15 reactions. BPAT-M identified three different target atom markings, resulting in two different mutually exclusive combinations.

In both of the results from BPAT-S and BPAT-M, the general branching scheme through L-valine and L-cysteine are found. The pathway through L-2-aminoadipate is not found because the carbons from L-2-aminoadipate are removed in the reaction between isopenicillin N and penicillin. This pathway is identified in the next example of cephalosporin C which proceeds through a similar pathway but retains the carbons of L-2-aminoadipate. The pathways to valine and cysteine from G6P are correct starting with pyruvate, but go through the shortcuts from G6P to pyruvate instead of more likely proceeding through glycolysis. As discussed with previous results, this is a common issue that computational methods for finding metabolic pathways face.

The BPAT-S pathway, similary to the streptomycin result, presents a more complete picture of the biosynthesis of penicillin than the pathway found by BPAT-M. Since the

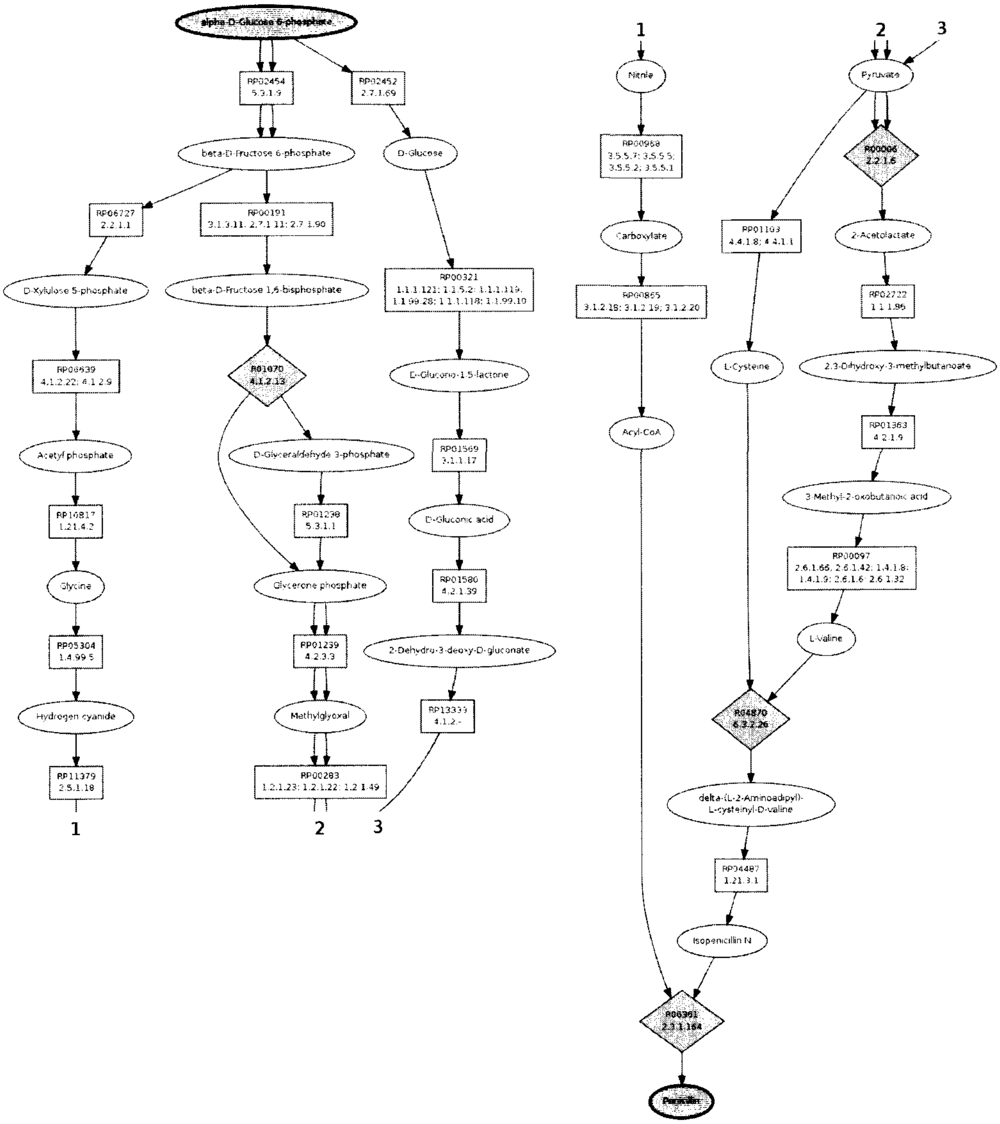


Figure 5.19 : Top ranked pathway returned by BPAT-S for G6P to Penicillin, with the linear pathways conserving 3 carbons.

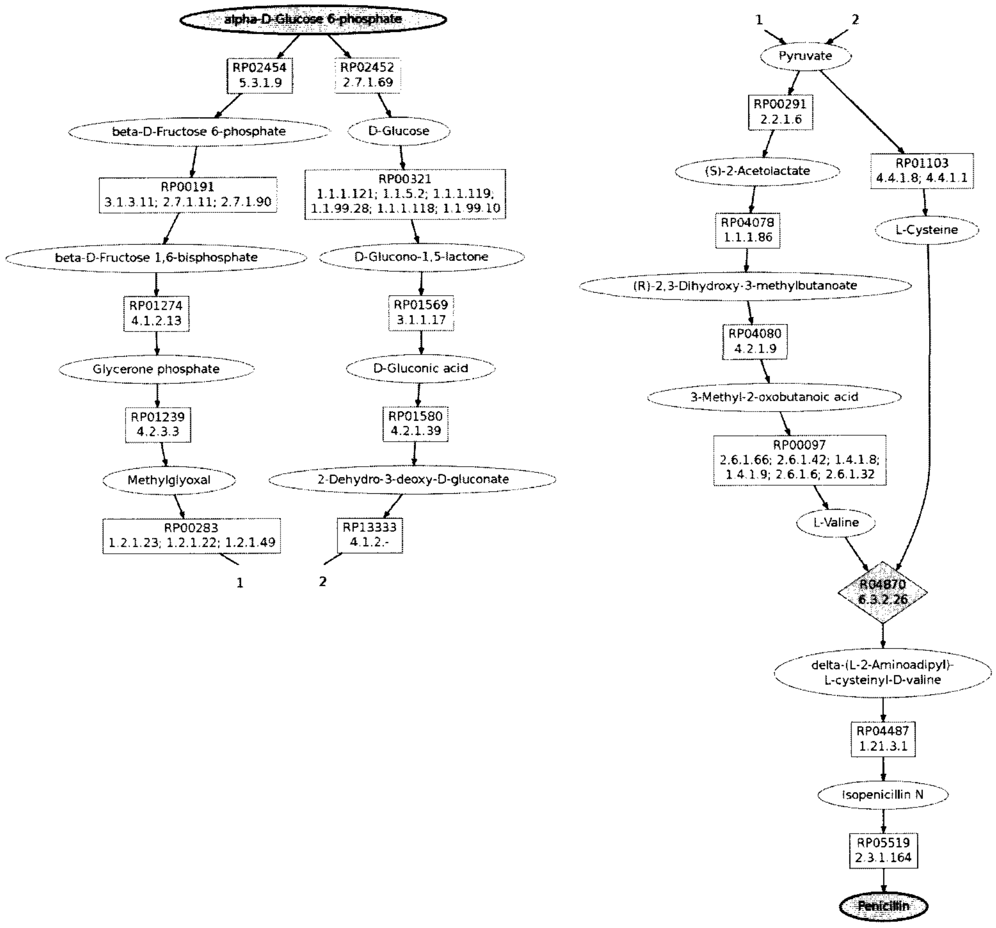


Figure 5.20 : Top ranked pathway returned by BPAT-M for G6P to Penicillin, with the linear pathways conserving 3 carbons.

seed pathway used proceeds through valine, BPAT-S is able to correctly identify that valine is produced by using two molecules of pyruvate. Notably, this pathway is 12 reactions long and is in the middle of the tens of thousands of linear pathways used by BPAT-S. Yet, because BPAT-S is able to analyze a large number of these pathways efficiently this result is pulled to the top of the branched pathways. Additionally, BPAT-S finds the pathway through acyl-CoA that contributes one carbon to penicillin along with the "R" that represents the different side chains that can be attached to penicillin. Although the pathway from G6P to acyl-CoA is unusual, and likely toxic due to proceeding through hydrogen cyanide, it is found because of the generic nature of the acyl-CoA compound. Pathway searches for more specific varieties of penicillin may have better results. BPAT-M does not identify the second pathway through valine or through acyl-CoA because they both conserve less than three carbons and are therefore not contained in the linear pathways.

5.5.9 α-D-Glucose 6-Phosphate to Cephalosporin C

Cephalosporin C is also a beta lactam antibiotic that is synthesized by certain bacteria and fungi. Cephalosporin C is a broader spectum antibiotic than penicillin, but is not used clinicially because of its low potency [28]. However, it is an important precursor for a number of related antibiotics and has also been a target for increased production using metabolic engineering approaches [116]. The biosynthetic pathway for cephalosporin C is similar to penicillin as it begins with the the synthesis of ACV from L-valine, L-cysteine and L-2-aminoadipate [18]. The pathway then proceeds through isopenicillin N which then undergoes a series of reactions resulting in cephalosporin C.

BPAT-S and BPAT-M were given as input G6P as the start compound, cephalosporin C as the target compound and three as the minimal number of carbons to conserve. BPAT-S used 31280 linear pathways of length ranging from 10 to 16 reactions. BPAT-M found 5 different target atom markings resulting in 18 different mutually exclusive combinations. The top ranked pathways for BPAT-S and BPAT-M are depicted in Figures 5.21 and 5.22, respectively. The top ranking pathways from BPAT-S and BPAT-M capture all three of the pathways to the reaction utilizing L-valine, L-cysteine and L-2-aminoadipate.

The introduction of the L-2-aminoadipate pathway causes some interesting changes in the BPAT-S results as compared to the penicillin results. Since the pathway through glycerone phophate and L-2-aminoadipate is used as a seed pathway, BPAT-S can no longer identify the branched pathway from two molecules of pyruvate that creates valine. Instead, it tries to conserve all of the carbons through valine and finds a long and unlikely pathway instead. This is a tradeoff because L-2-aminoadipate is also created via a branched pathway and therefore using a pathway through valine results in a long and unlikely pathway for L-2-aminoadipate. In this case, the seed pathway through L-2-aminoadipate resulted in a smaller overall branched pathway. BPAT-S also identifies the branch through acetyl-CoA which contributes its acetyl group to deacetylcephalosporin C to make cephalosporin C. BPAT-M does not identify this branch because it only contributes two carbons, but obtains better pathways for the other parts of the pathway.

Since the shorter pathways through valine and L-2-aminoadipate were both contained in the linear pathway set, BPAT-M is able to merge them together and obtain a more realistic pathway for cephalosporin C. The pathway correctly matches the known pathway from pyruvate and 2-oxoadipate on down. As with many of the results, the pathways from G6P to those compounds take some unusual shortcuts, in the pyruvate case it would be more likely to be made by glycolysis. It is interesting to note that BPAT-M is able to reveal more variety between the compounds as compared to BPAT-S, which will use the same branch between two different compounds.

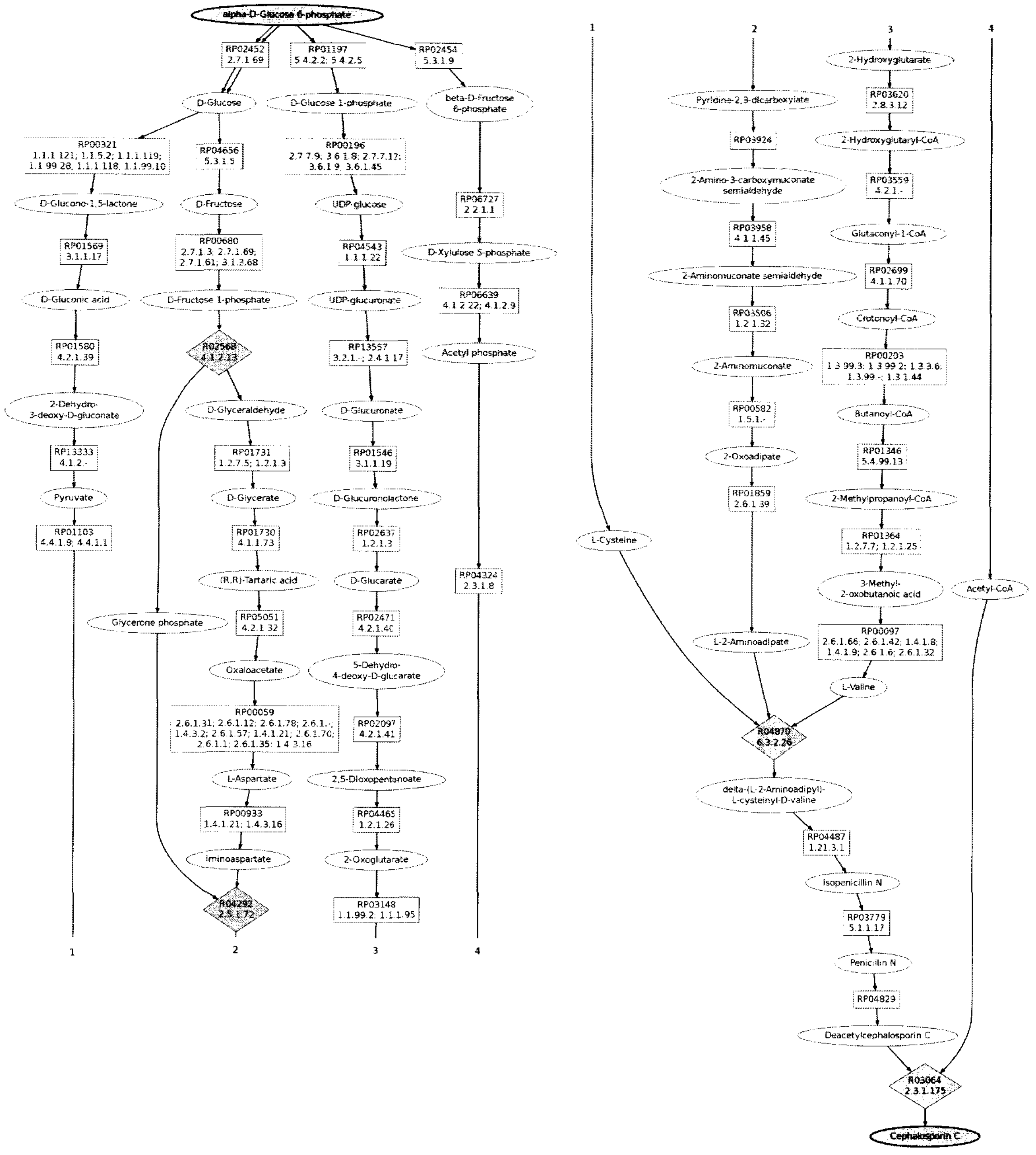


Figure 5.21 : Top ranked pathway returned by BPAT-S for G6P to cephalosporin C, with the linear pathways conserving 3 carbons.

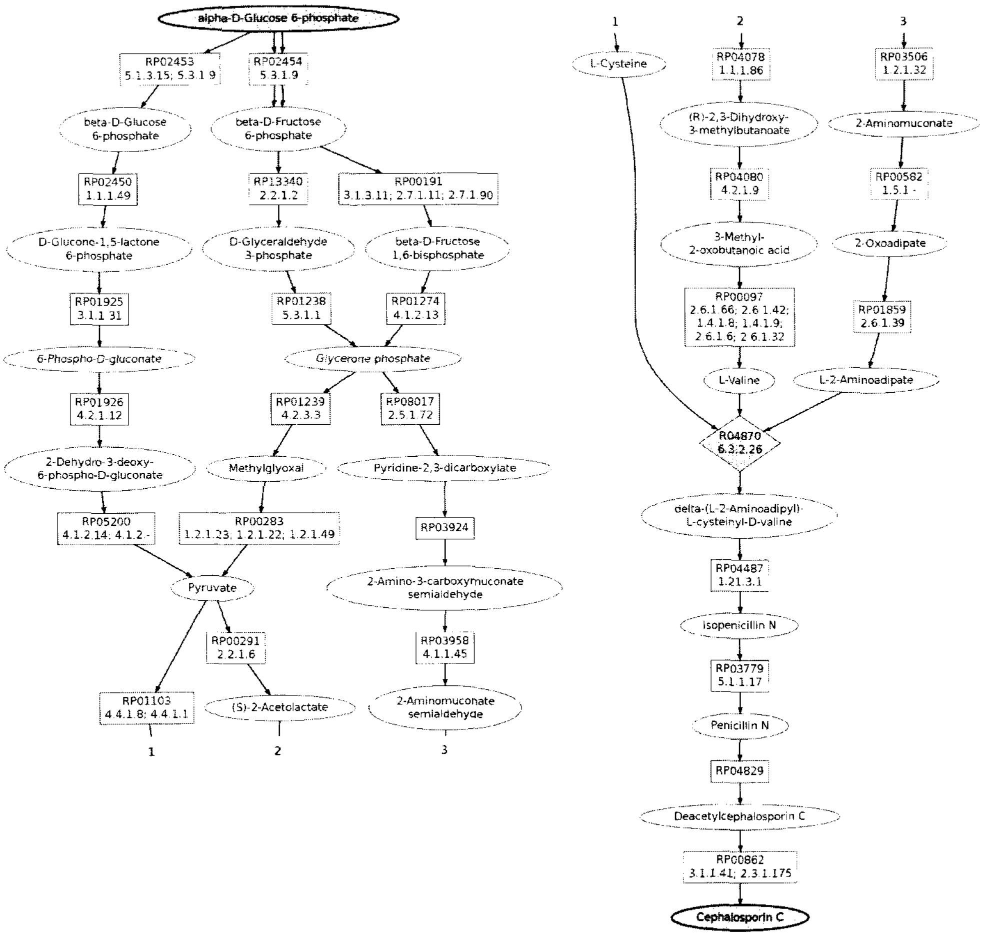


Figure 5.22 : Top ranked pathway returned by BPAT-M for G6P to cephalosporin C, with the linear pathways conserving 3 carbons.

5.5.10 α-D-Glucose 6-Phosphate to Erythromycin

Erythromycin is a highly successful, broad-spectrum, macrolide antibiotic discovered in the early 1950s and quickly became the preferred drug for a wide variety of infections [124, 125]. Erythromycin is produced by bacteria *Saccharopolyspora erythraea* and is difficult to synthesize in the laboratory [87]. Therefore, several metabolic engineering approaches have been taken to improve the production of erythromycin and erythromycin precursors, both by modifying *S. erythraea* or inserting the genes required into other microorganisms such as *E. coli* [85, 89, 24, 99]. The biosynthesis of erythromycin has been well studied and proceeds in two major steps. The first is the construction of the macrocyclic lactone intermediate, 6-Deoxyerythronolide B (6DB) by 6DB synthase from six molecules of methylmalonyl-CoA and one propanoyl-CoA [59]. 6DB then undergoes a series of modifications that includes the attachment of two sugars, L-mycarose and D-desosamine, in order to produce erythromycin [127, 114].

Both BPAT-S and BPAT-M are able to find the relatively complex branched pathway for the biosynthesis of erythromycin using G6P as the start compound and conserving three carbons. BPAT-S was initialized with eight molecules of G6P and due to the large number of branches, *k* was limited to 200000, resulting in 2902 linear pathways ranging in length from 10 to 17 reactions. The default limits were retained for BPAT-M, which found 15 different target atom markings, resulting in 4592 mutually exclusive combinations. The top ranking result for BPAT-S is found in Figure 5.23, with the seed pathway going through (*S*)-lactate. Overall, the pathway found by BPAT-S is similar to the top ranking result for BPAT-M, depicted in Figure 5.24. Both results find the proper pathway from pyruvate to the synthesis of 6DB and the addition of the two sugars to complete the synthesis of erythromycin. The branched pathways contain the known pathway for the D-desosamine and a pathway highly similar to the known pathway for the L-mycarose, except for a slight

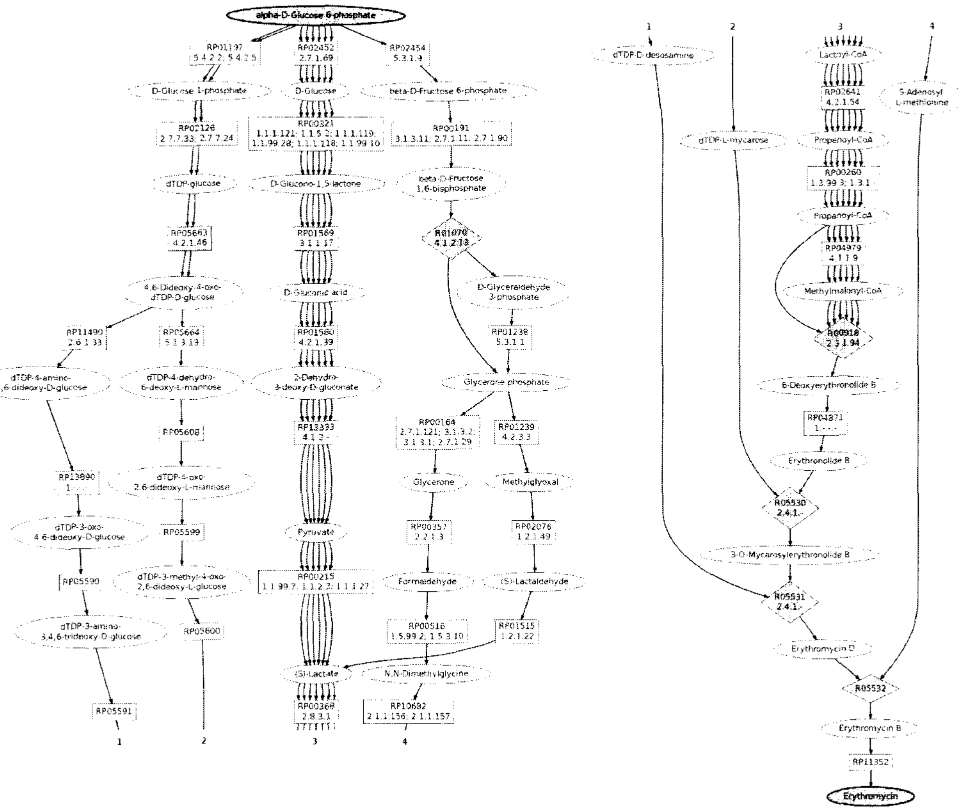


Figure 5.23 : Top ranked pathway returned by BPAT-S for G6P to erythromycin, with the linear pathways conserving 3 carbons.

shortcut through dTDP-4-dehydro-6-deoxy-L-mannose.

The differences between the two reinforce a number of general trends that appear when comparing the results of BPAT-S and BPAT-M in previous results. BPAT-S is able to find a branch that conserves less carbons than the initial seed pathway, the one through S-adenosyl-L-methionine, which does not appear in the BPAT-M results because it is not included in the original set of linear pathways. BPAT-M reveals more variety in the pathways between G6P to pyruvate, although the likely pathway through glycolysis is not found due to its length. There is a pathway from G6P to (*S*)-lactate in the BPAT-S pathway that is not in the BPAT-M pathway. BPAT-S is able to find that one molecule of G6P is able to result in

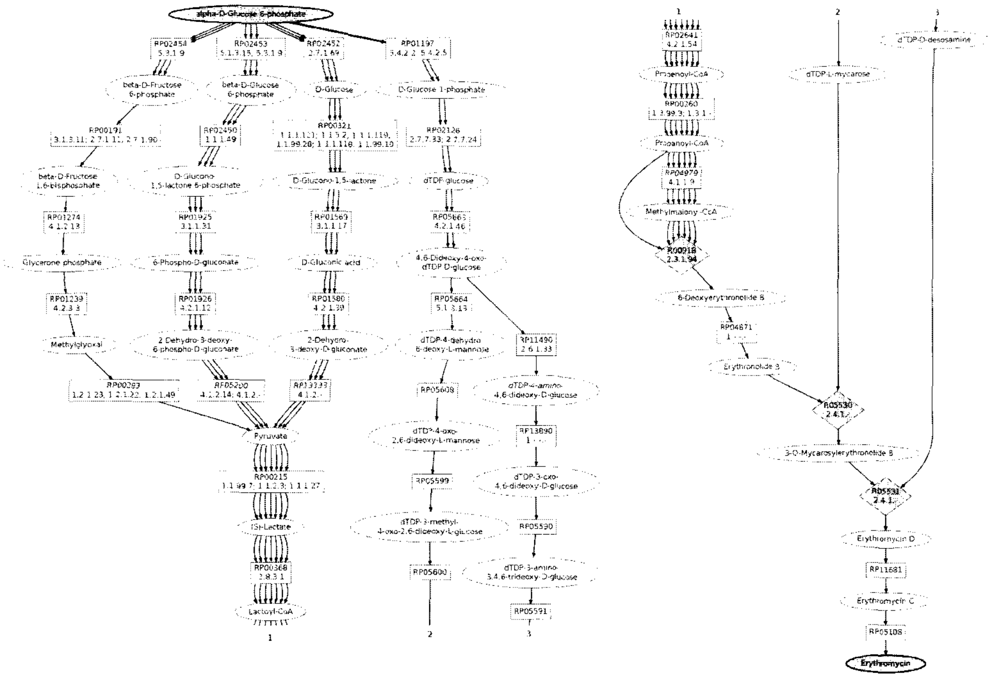


Figure 5.24 : Top ranked pathway returned by BPAT-M for G6P to erythromycin, with the linear pathways conserving 3 carbons.

one molecule of (*S*)-lactate and S-adenosyl-L-methionine. Although, due to their search methods, neither algorithm is able to identify that each molecule of G6P can result in two molecules of pyruvate. This occurs in BPAT-S because branches are not allowed to branch and in BPAT-M because pathways only merge at one point. This may be corrected in future work through either adding a post processing step or improving the algorithms to find these occurrences, which is further discussed in Section 6.1.1.

5.5.11 α-D-Glucose 6-Phosphate to Lycopene

Lycopene is a C_{40} carotenoid having a bright red color and is typically found in fruits and vegetables, such as tomatoes and watermelons. Popular interest in lycopene has grown because it is an antioxidant and studies indicate it may be helpful in the prevention and treatment of diseases such as cancer [97]. Lycopene's nutritional and pharmaceutial potential has resulted in a number of investigations on using metabolic engineering techniques to increase yield and/or produce lycopene in microbial hosts [101, 4, 132, 123]. The biosynthesis pathway of lycopene is relatively well understood and it utilizes several molecules of isopentenyl diphosphate (IPP) and dimethylallyl diphosphate (DMAPP) [100]. As discussed in result for (-)-carvone, IPP and DMAPP can be produced by either the mevalonate (MVA) pathway or the methylerythritol phosphate (MEP) pathway, depending on the organism.

Lycopene proves to be an interesting test case for the branched pathfinding algorithms because it is a case where BPAT-M performs better than BPAT-S, likely due to the underlying structure of the lycopene biosynthesis pathway. BPAT-S and BPAT-M are given as the start compound G6P, the target compound lycopene and three as the number of carbons to conserve. Several of the default parameters to be readjusted because of the large number of carbons in lycopene and the complexity of the overall pathway. For BPAT-S, *k* was reduced to 500,000 because it would reach the maximum time allowed of 24 hours at the default value. For BPAT-M, the number of pathways in each cluster was limited to 2,500 and *r* was set to 500 because it would run out of memory using the default values. BPAT-S analyzed 22,064 linear pathways of length ranging from 14 to 21 reactions and BPAT-M found 16 mutually exclusive target atom markings that generated 6,301 combinations. The top ranked results for BPAT-S and BPAT-M can be found in Figures 5.25 and 5.26, respectively.

BPAT-M correctly finds the lycopene pathway that utilizes the MEP pathway to synthesize IPP and DMAPP. This pathway has an interesting topology as IPP and DMAPP are produced using the same reactions and then the GMAPP combines with IPP to make two molecules of geranyl diphosphate which are combined with IPP in two more sequential reactions resulting in two molecules of geranylgeranyl diphosphate, which combine to make the C_{40} molecule prephytoene diphosphate that becomes lycopene. This pathway topology lends itself well to discovery by BPAT-M because it is naturally a merging of several linear pathway of similar length that result in a complex branched pathway.

BPAT-S has greater difficulty identifying the overall pathway, mainly because branches off of the seed pathway cannot further branch and this is required to generate the topology found by BPAT-M. Still, quite a bit of the general topology is contained in the top ranked pathway. Additionally, the result from BPAT-S does utilize both the MEP pathway for DMAPP and MVA pathway for IPP, thus revealing the variety available. However, similar to the (-)-carvone result, the pathway found from G6P to acetoacetyl-CoA to begin the MVA pathway is long and unwieldy because the branch is trying to conserve the maximum number of carbons without being able to branch. Lastly, the top ranked BPAT-S pathway contains cis forms of the intermediate compounds between prephyonene diphosphate and lycopene. These are valid reactions and compounds contained in the KEGG databases, and there is some evidence that they perhaps could occur, but they do not appear to be well characterized [43, 75]. The third ranked BPAT-S pathway contains the more well characterized trans forms

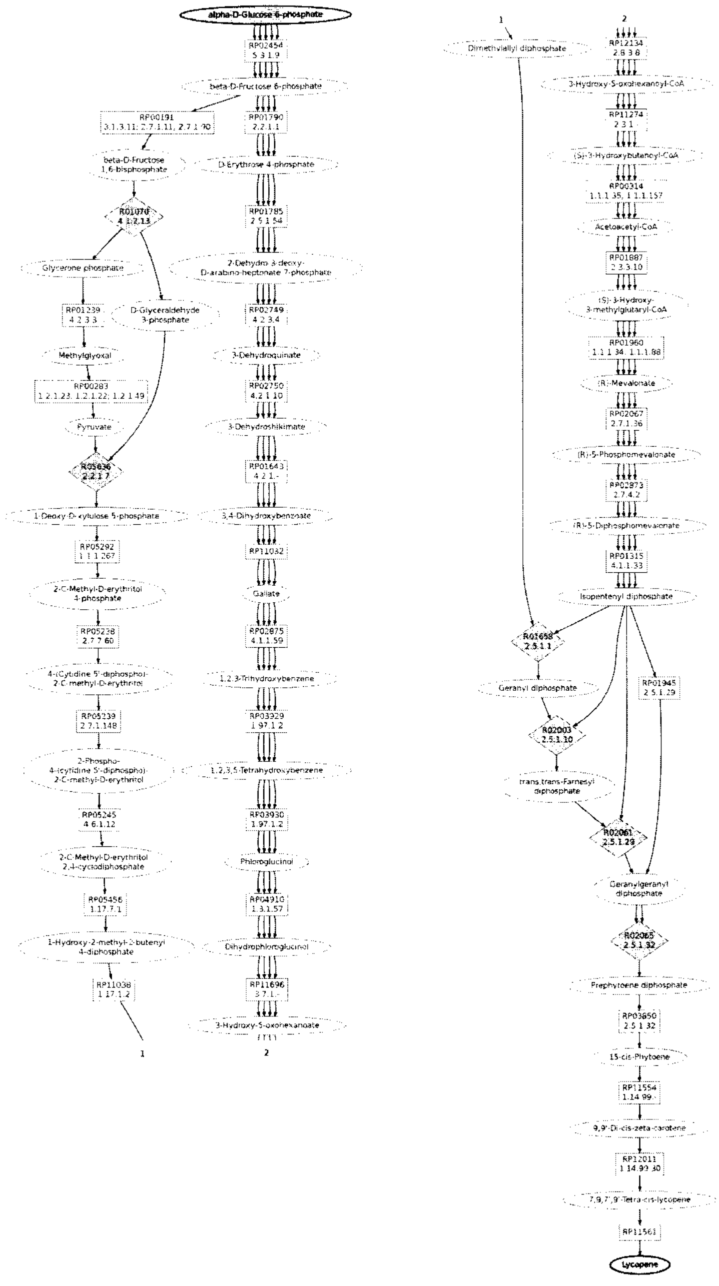


Figure 5.25 : Top ranked pathway returned by BPAT-S for G6P to lycopene, with the linear pathways conserving 2 carbons.

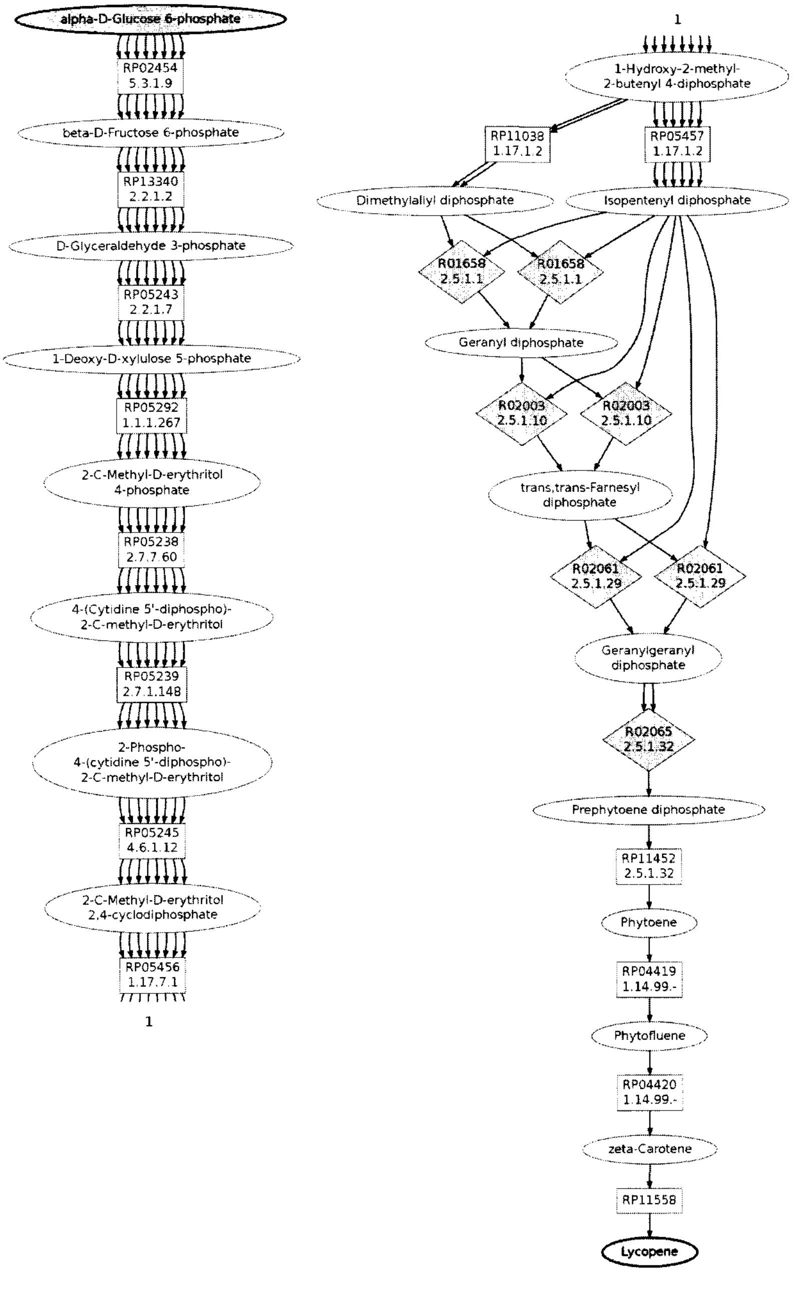

alpha-D-Glucose 6-phosphate
RP02454
5.3.1.9
beta-D-Fructose 6-phosphate
RP13340
2.2.1.2
D-Glyceraldehyde 3-phosphate
RP05243
2.2.1.7
1-Deoxy-D-xylulose 5-phosphate
RP05292
1.1.1.267
2-C-Methyl-D-erythritol 4-phosphate
RP05238
2.7.7.60
4-(Cytidine 5'-diphospho)-2-C-methyl-D-erythritol
RP05239
2.7.1.148
2-Phospho-4-(cytidine 5'-diphospho)-2-C-methyl-D-erythritol
RP05245
4.6.1.12
2-C-Methyl-D-erythritol 2,4-cyclodiphosphate
RP05456
1.17.7.1
1
1
1-Hydroxy-2-methyl-2-butenyl 4-diphosphate
RP11038
1.17.1.2
RP05457
1.17.1.2
Dimethylallyl diphosphate
Isopentenyl diphosphate
R01658
2.5.1.1
R01658
2.5.1.1
Geranyl diphosphate
R02003
2.5.1.10
R02003
2.5.1.10
trans,trans-Farnesyl diphosphate
R02061
2.5.1.29
R02061
2.5.1.29
Geranylgeranyl diphosphate
R02065
2.5.1.32
Prephytoene diphosphate
RP11452
2.5.1.32
Phytoene
RP04419
1.14.99.-
Phytofluene
RP04420
1.14.99.-
zeta-Carotene
RP11558
Lycopene

Figure 5.26 : Top ranked pathway returned by BPAT-M for G6P to lycopene, with the linear pathways conserving 2 carbons.

of the compounds from prephyonene diphosphate to lycopene and it is promising that the algorithm is capable of finding both pathways.

5.5.12 Wall Time for Finding Branched Pathway

The wall time taken on the system described in Section 5.3 to find the branched pathways by BPAT-S or BPAT-M is plotted in Figures 5.27 and 5.28, respectively. The maximum time allowed for a pathway search was 24 hours. The overall height of each bar indicates the total time and then the bars are partitioned into sections representing the times required for different stages of the algorithms. The plots reveal that for all pathways, except the one to (*S*)-norcoclaurine, BPAT-S took signficantly more time than BPAT-M. In the case of (*S*)-norcoclaurine, it took about a minute with BPAT-S and 14 seconds with BPAT-M. The wide variation of run times indicates that the time required is dependent on the start and target compound and is difficult to predict beforehand.

Without any limits on the number of atoms conserved by the branches or the number of branches per seed pathways, BPAT-S typically takes on the order of hours to run. The run time of BPAT-S does not appear to be dependent on the complexity of the expected result, as the L-threonine pathway is a relatively simple branched pathway that took a long time to compute. There is also no clear trend in the relatively amount of times spent finding branches versus trying different combinations of branches. In several cases, finding branches appear to dominate the computation time. This happens in the L-threonine pathway as well and it seems to be because there are a large number of one carbon branches, if one limits the branches to conserve at least two carbons the run time falls dramatically to about 25 minutes. Other pathways,such as erythromycin and lycopene, are dominated by the time it takes to try combinations of branches. This is to be expected because these pathways require a large number of branches to synthesize their final target compounds.

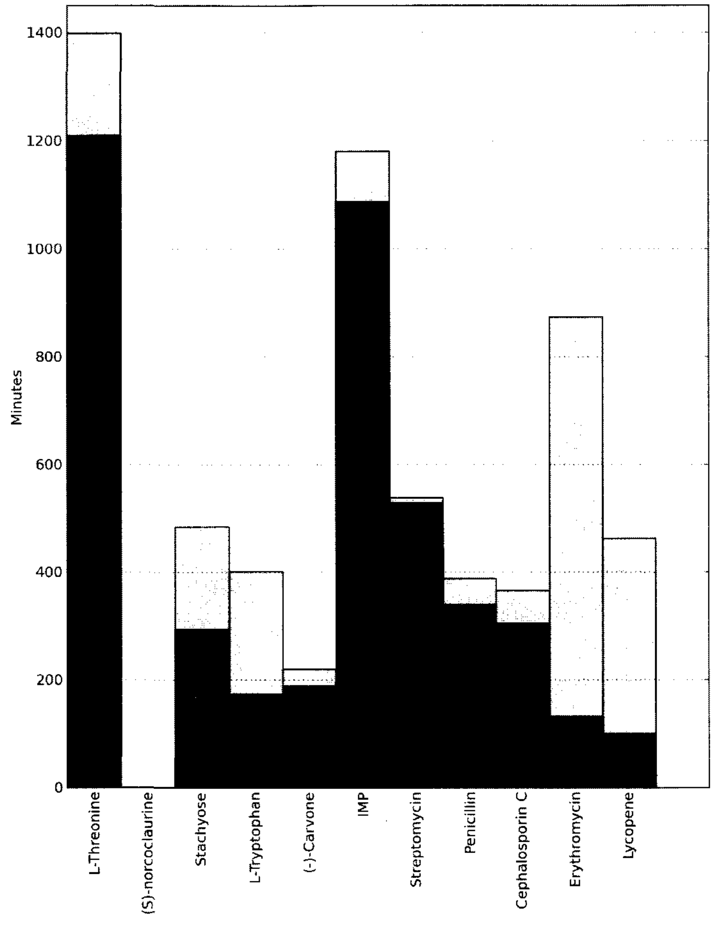


Figure 5.27 : Wall times for BPAT-S for each branched pathway as indicated by the target compound labels. The dark teal is the time required for using LPAT to find the branches. The light green is the time required to try all combination of branches. The time required for finding the original set of linear pathways is so relatively negligible that it is not visible.

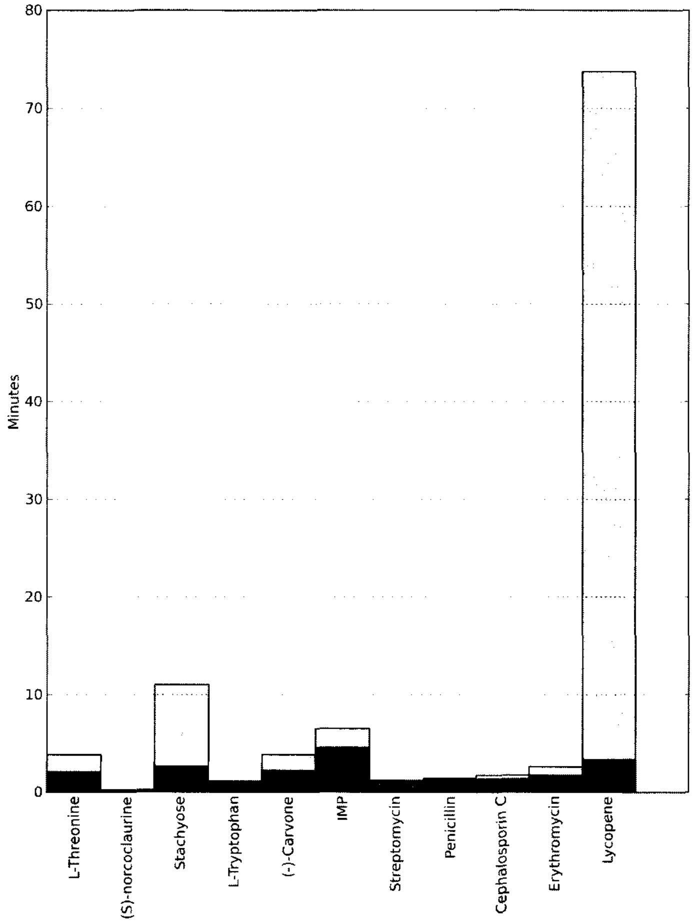


Figure 5.28 : Wall times for BPAT-M for each branched pathway as indicated by the target compound labels. The red is the time used by LPAT to find the set of linear pathways. The dark teal is the time used to construct M and the light green is the time used to merge the pathways using Q and C and M. The time required to construct Q and C is so relatively negligible that it is not visible.

The wall times for BPAT-M are significantly shorter for the same branched pathways than BPAT-S. However, this is likely because BPAT-M utilizes stricter cutoffs through the number of pathways kept for each TAM in Q and w. When these cutoffs were raised, the search would run out of memory before running out of time. Therefore, it is interesting that, at least by analyzing the top returned pathway, BPAT-M was able to perform as well as BPAT-S for many pathways. Since the times are so much shorter the time required to find the original linear pathways by LPAT is relatively more significant and shows up on the plot as the bottom, red section of the bars. For all of the pathways, the time to do the pairwise mergable comparisons of pathways to construct M took a few minutes or less. In the case of lycopene, and to a lesser extent stachyose, which took greater amounts of time to compute, the time required to merge the pathways dominated the computation. It is of note that the the lycopene pathway took the longest to compute and was the only one where BPAT-M was able to outperform BPAT-S.

5.5.13 General Summary of Branched Pathway Results

BPAT-S and BPAT-M take two different heuristic approaches for finding metabolic pathways and the experimental results reveal that their performance depends on the underlying structure of the pathway they seek to find. The approach taken by BPAT-S will perform better if there is one main linear pathway that transfers a significant portion of the mass from the start to target compound and the branches are smaller modifications. This occurs in the case of the tryptophan pathway, where four carbons are conserved through the shikimate pathway and the other carbons are lost and gained in shorter, two or three carbon branches. In other cases, BPAT-S is able to identify branches that conserve less carbons than the original linear pathways, something that BPAT-M is unable to do. However, the merging approach used by BPAT-M will perform better if all of the branches conserve at least the given number of atoms and are of similar length. If this is the case, then BPAT-M can also find more complex topologies, because it is not limited to requiring all branches to start and end from the same pathway. This is typical of large compounds made from similar components, and the lycopene result highlights this ability of BPAT-M. The lycopene pathway utilizes a complex "interwoven" pattern that BPAT-S is unable to find, but is returned as the top result by BPAT-M. While it can be difficult to identify *a priori* which method will perform better, it is reasonable to try both approaches and analyze the results. By utilizing both approaches, researchers may gain insight into different metabolic schemes. Additionally, since the algorithms are complementary, future work may improve the performance by combining them as discussed in Section 6.1.1.

Chapter 6

Conclusions and Future Directions

The metabolic path finding algorithms presented in this thesis are part of a growing set of analysis tools that will assist in understanding metabolic networks and designing novel metabolic pathways for applications such as metabolic engineering and synthetic biology. Chapter 3 described a new algorithm, LPAT, that guarantees finding linear metabolic pathways between a start and target compound that conserve a given number of atoms. Chapter 4 presented BPAT-S and BPAT-M, two new algorithms for heuristically finding branched metabolic pathways between a start and target compound. The experimental results in Chapter 5 demonstrated that while the theoretical complexity of finding even linear atom conserving pathways is high that the algorithms in this thesis have reasonable running times in practice. This is due to several factors including the appropriate representations and heuristics and the structure of the underlying metabolic network. The experimental results also provide inspiration for a number of future directions for improving metabolic pathfinding and the analysis of the resulting pathways.

6.1 Future Directions

The work presented in this thesis provides a solid foundation for future work on metabolic pathfinding. There are a number of interesting future directions to persue, some related to algorithmic improvements and other focusing on the analysis and visualization of the found pathways.

6.1.1 Improvements to BPAT-S and BPAT-M

One of the clearest algorithmic extensions to the current work is to combine the complementary approaches taken by BPAT-S and BPAT-M. Currently, BPAT-S attaches branches to a single seed pathway generated from a single linear pathway returned by LPAT. However, it may improve results to generate "seed pathways" from the branched pathway results of BPAT-M. The combined method would proceed by processing each branched pathway returned by BPAT-M in the same manner as the linear pathways from LPAT by BPAT-S to identify places where atoms can be lost and gained. Then BPAT-S would go through the same method to attach new branches to the branched pathway from BPAT-M. In this way, the combined method takes advantage of the BPAT-M's ability to identify more complex topologies with BPAT-S's ability to identify branches that do not exist in the original set of linear pathways. Since the experimental results reveal that the time needed by BPAT-M is relatively small to the amount of time required by BPAT-S, combining the methods in this manner should not greatly increase the overall time required.

Still, a combined method may not find certain topologies of branched pathways because the more complex topologies found by BPAT-M rely on the pathways being in the set of original linear pathways. An important question to ask is how common complex topologies are in biochemical systems and how useful are they for producing compounds of interest. As discussed in Chapter 2, there have been a number of studies on the average or overall topologies of metabolic networks, but there has not been as much work understanding the networks required to synthesize, or breakdown, one compound. A better understanding of the topologies that algorithms like BPAT-S and BPAT-M should be able to find can be used to guide the design of future extensions. If it turns out there are topologies that BPAT-M combined with BPAT-S cannot handle, another direction to explore would be to extend BPAT-S to allow branching from the branches, but how to proceed while still having reasonable search times is less clear. One way to reduce the wall time required is to parallelize the algorithm, and in some sense it is embarrassingly parallel as each processor can be assigned a set of seed pathways. However, storing previously identified branches is important to the efficiency of BPAT-S, and so the algorithm may be better suited to shared memory systems, which are becoming more common with the rise of multi-core processors.

6.1.2 Incorporating Known Biochemical Information

A theme found in a number of the experimental results is that the computed pathways contain unusual shortcuts around glycolysis. This is because pathways are ranked by the number of reactions they use and glycolysis is a relatively long pathway. However, glycolysis is a highly conserved and universal metabolic pathway that generates energy and therefore, while perhaps interesting, the unusual shortcuts are unlikely to be used in reality. In some way, not identifying glycolysis could be perceived as a failing of the computational methods, but the end goal of these methods is to discover new and interesting pathways, not to rediscover well known pathways. At the same time, when finding branched pathways, better overall pathways may be found if well known pathways such as glycolysis are treated differently.

One way to incorporate known metabolic pathways is to postprocess the resulting pathways. If the resulting pathways contain a different pathway between two compounds that are connected in a known pathway, then the known pathway can replace the found pathway. This may make the size of the overall pathway larger and experimentation would need to be done to understand how to count the size of the known pathway in the final ranking of the results. Another approach would be to incoporate the known metabolic pathways in the search itself. How to exactly do this raises a number of questions and would likely result in a redesign of the algorithms. For example, one way would be to search from both the start compound and target compound at the same time. If the search from the start compound and the search from the target compound arrive at compounds that are connected by a known pathway then the searches can stop and the known pathway can just be used to connect the two pathways. In the end, the decision to incorporate known pathways will likely depend on the specific application. While the known pathways may make the results look better in terms of validation and testing, it is a balancing act between returning what is already known versus finding new or unusual pathways.

6.1.3 Clustering and Analysis of Returned Pathways

The experimental results in this thesis provide an initial cluster analysis of linear pathways that reveals that it may be of interest to consider the resulting pathways as a whole, instead of just manually analyzing the top ranked pathways. The analysis presented used a relatively simple distance measure and agglomerative clustering method. Future work should investigate how the choice of measure and method change the results. For example, the distance measure used currently considers each compound as a completely independent entity, but this is not true because there is a notion of chemical similarity. Therefore, it is possible that incorporating chemical similarity when comparing two pathways may return more chemically relevant clusterings. Another important extension to understand how to cluster branched pathways. The main challenge is the development of a proper distance measure. Should the emphasis be placed on topological or chemical similarity, both, or something else entirely? Once the distance measure is developed, then based on the linear pathway results, it is likely that standard clustering methods will produce interesting results and hopefully further understanding of different metabolic schemes.

6.1.4 Visualization and Interactivity

In the end, it is highly unlikely that any algorithm or ranking scheme will be able to return a perfect list of metabolic pathways for any application. This means, along with the large number of pathways typically returned, that it is important to develop good visualization tools that will allow researchers to interact with the results in a meaningful way. Some initial work has been done in implementing a webserver for these algorithms and enabling users to filter the pathways based on their compounds and reactions. However, providing interactive and useful visualizations of large numbers of metabolic pathways, especially branched pathways, pose a number of unique challenges that deserve future investigation. The goal of finding "biologically meaningful" and "realistic" metabolic pathways is not well-defined and while much effort has been put into validating and testing the algorithms in this thesis, the true test will be experimental validation of new and interesting computational results.

Appendix A

Linear Test Pathways

Table A.1 : Pathways in the NeAT Test Set

Pathway Name	Organism	Start Compound	Target Compound	Number of Reactions
Arginine Catabolism	*E. coli*	L-Arginine	L-Glutamate	5
Glycolysis	*E. coli*	beta-D-Glucose 6-phosphate	Pyruvate	8
Rhamnose Catabolism 1	*E. coli*	L-Rhamnose	L-Lactate	4
Rhamnose Catabolism 2	*E. coli*	L-Rhamnose	Propane-1,2-diol	4
D-Glucarate Catabolism	*E. coli*	D-Glucarate	3-Phospho-D-glycerate	4
Fucose Catabolism 1	*E. coli*	L-Fucose	Propane-1,2-diol	4
Fucose Catabolism 2	*E. coli*	L-Fucose	2-Dehydro-3-deoxy-L-rhamnonate	4
Fucose Catabolism 3	*E. coli*	L-Fucose	L-Lactate	4
Galactonate Catabolism	*E. coli*	D-Galactonate	D-Glyceraldehyde 3-phosphate	3
Hexitol Degradation 1	*E. coli*	Sorbitol 6-phosphate	D-Glyceraldehyde 3-phosphate	3
Hexitol Degradation 2	*E. coli*	Sorbitol 6-phosphate	D-Glyceraldehyde 3-phosphate	3
Sorbitol Degradation	*E. coli*	Sorbitol 6-phosphate	D-Glyceraldehyde 3-phosphate	3
Threonine Degradation 1	*E. coli*	L-Threonine	Methylglyoxal	3
Threonine Degradation 2	*E. coli*	L-Threonine	(*R*)-1-Aminopropan-2-ol	3

Continued on Next Page...

Table A.1 – Continued

Pathway Name	Organism	Start Compound	Target Compound	Number of Reactions
Pyruvate Oxidation	*E. coli*	Pyruvate	Acetyl-CoA	3
Chorismate Biosynthesis	*E. coli*	2-Dehydro-3-deoxy-D-arabino-heptonate 7-phosphate	Chorismate	6
D-Galacturonate Catabolism	*E. coli*	D-Galacturonate	2-Dehydro-3-deoxy-6-phospho-D-gluconate	4
Mannitol Degradation	*E. coli*	D-Mannitol 1-phosphate	D-Glyceraldehyde 3-phosphate	3
Glycolysis	*E. coli*	alpha-D-Glucose 6-phosphate	Pyruvate	8
D-Galactarate Catabolism	*E. coli*	D-Galactarate	3-Phospho-D-glycerate	4
D-Glucuronate Catabolism	*E. coli*	beta-D-Glucuronoside	2-Dehydro-3-deoxy-6-phospho-D-gluconate	5
Arginine Utilization	*E. coli*	L-Arginine	Succinate	6
Heme Biosynthesis	*S. cerevisiae*	5-Aminolevulinate	Heme	7
Purine de novo Biosynthesis	*S. cerevisiae*	PRPP	GMP	12
Biotin Biosynthesis	*S. cerevisiae*	8-Amino-7-oxononanoate	Biotin	3
Arginine Degradation	*S. cerevisiae*	L-Arginine	CO2	3

Continued on Next Page...

Table A.1 – *Continued*

Pathway Name	**Organism**	**Start Compound**	**Target Compound**	**Number of Reactions**
Sulfur Incorporation and Transsulfuration	*S. cerevisiae*	L-Homoserine	2-Oxobutanoate	4
Purine de novo Biosynthesis	*S. cerevisiae*	PRPP	AMP	12
Heme Biosynthesis	*H. sapiens*	5-Aminolevulinate	Heme	7
Bile Acid Synthesis 1	*H. sapiens*	Cholesterol	Cholate	15
Bile Acid Synthesis 2	*H. sapiens*	Cholesterol	Glycochenodeoxycholate	11
Bile Acid Synthesis 3	*H. sapiens*	Cholesterol	Cholate	14
Bile Acid Synthesis 4	*H. sapiens*	Cholesterol	Taurochenodeoxycholate	11
Bile Acid Synthesis 5	*H. sapiens*	Cholesterol	Cholate	15
Corticosterone Biosynthesis	*H. sapiens*	Cholesterol	Corticosterone	4
Pantothenate Metabolism	*H. sapiens*	Uracil	D-4'-Phosphopantothenate	5
Guanine Biosynthesis	*H. sapiens*	IMP	Guanine	4
Fatty Acid Biosynthesis	*H. sapiens*	Holo-[carboxylase]	CoA	3
Lysine Degradation	*H. sapiens*	L-Lysine	Acetyl-CoA	9
Beta-Alanine Biosynthesis	*H. sapiens*	Uracil	beta-Alanine	3

Continued on Next Page...

Table A.1 – Continued

Pathway Name	Organism	Start Compound	Target Compound	Number of Reactions
Aldosterone Biosynthesis	*H. sapiens*	Cholesterol	Aldosterone	6

Table A.2 : Pathways in the Amino Acid Test Set

Start Compound	**Target Compound**	**Number of Pathways**	**Number of Reactions in Longest Pathway**
PRPP	L-Histidine	2	10
Pyruvate	L-Leucine	5	10
Homocitrate	L-Lysine	2	9
L-Aspartate	L-Lysine	4	9
Pyruvate	L-Isoleucine	1	9
L-Glutamate	L-Isoleucine	1	8
L-Glutamate	L-Arginine	2	8
Propanoate	L-isoleucine	1	7
L-Aspartate	L-Methionine	3	7
L-Threonine	L-Isoleucine	1	6
Pyruvate	L-Valine	2	6
Chorismate	L-Tryptophan	2	5
L-Arginine	L-Proline	2	5
L-Glutamine	L-Arginine	1	4
L-Aspartate	L-Threonine	1	4
L-Glutamate	L-Proline	2	4
CO2	L-Arginine	1	4
Chorismate	L-Tyrosine	2	3
Chorismate	L-Phenylalanine	2	3
3-Phospho-D-glycerate	L-Serine	1	3
L-Ornithine	L-Proline	3	3
Cysteine	L-Asparagine	1	2

Continued on Next Page...

Table A.2 – Continued

Start Compound	**Target Compound**	**Number of Pathways**	**Number of Reactions in Longest Pathway**
Cysteine	L-Aspartate	1	2
L-Threonine	L-Glycine	2	2
L-Serine	L-Cysteine	2	2
Pyruvate	L-Alanine	4	2
L-Asparagine	L-Aspartate	1	1
L-Cysteine	L-Alanine	1	1
L-Phenylalanine	L-Tyrosine	1	1
L-Serine	L-Glycine	1	1
L-Aspartate	L-Asparagine	2	1
L-Aspartate	L-Alanine	1	1
Glyoxylate	L-Glycine	3	1
Oxaloacetate	L-Aspartate	1	1
2-Oxoglutarate	L-Glutamate	4	1
L-Glutamate	L-Glutamine	5	1
Pyruvate	L-Serine	1	1
CO2	L-Glycine	1	1

Bibliography

[1] AKUTSU, T. Efficient extraction of mapping rules of atoms from enzymatic reaction data. *Journal of Computational Biology 11*, 2-3 (2004), 449462.

[2] ALBERT, R. Scale-free networks in cell biology. *Journal of Cell Science 118*, Pt 21 (2005), 4947–57.

[3] ALBERT, R., AND BARABÁSI, A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics 74*, 1 (2002), 47–97.

[4] ALPER, H., MIYAOKU, K., AND STEPHANOPOULOS, G. Construction of lycopene-overproducing E. coli strains by combining systematic and combinatorial gene knock-out targets. *Nature Biotechnology 23*, 5 (2005), 612–616.

[5] ARITA, M. In silico atomic tracing by substrate-product relationships in Escherichia coli intermediary metabolism. *Genome Research 13*, 11 (2003), 2455–66.

[6] ARITA, M. The metabolic world of Escherichia coli is not small. *Proceedings of the National Academy of Sciences 101*, 6 (2004), 1543–1547.

[7] ARITA, M. *Introduction to the ARM Database: Database on Chemical Transformations in Metabolism for Tracing Pathways.* Springer-Verlag, Tokyo, 2005, ch. 13, pp. 193–210.

[8] BADER, G. D., CARY, M. P., AND SANDER, C. Pathguide: a pathway resource list. *Nucleic Acids Research 34*, Database issue (2006), D504–6.

[9] BAILEY, J. Toward a science of metabolic engineering. *Science 252*, 5013 (1991), 1668–1675.

[10] BAILLY, C., AUDIGIER, C., LADONNE, F., WAGNER, M. H., COSTE, F., CORBINEAU, F., AND CÔME, D. Changes in oligosaccharide content and antioxidant enzyme activities in developing bean seeds as related to acquisition of drying tolerance and seed quality. *Journal of Experimental Botany 52*, 357 (2001), 701–8.

[11] BAIROCH, A. The ENZYME database in 2000. *Nucleic Acids Research 28*, 1 (2000), 304–305.

[12] BARABÁSI, A.-L., AND ALBERT, R. Emergence of scaling in random networks. *Science 286*, 5439 (1999), 509–12.

[13] BARABÁSI, A.-L., AND OLTVAI, Z. N. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics 5*, 2 (2004), 101–13.

[14] BENNETT, R. N., AND WALLSGROVE, R. M. Secondary metabolites in plant defence mechanisms. *New Phytologist 127*, 4 (1994), 617–633.

[15] BLUM, T., AND KOHLBACHER, O. MetaRoute: fast search for relevant metabolic routes for interactive network navigation and visualization. *Bioinformatics 24*, 18 (2008), 2108–2109.

[16] BLUM, T., AND KOHLBACHER, O. Using Atom Mapping Rules for an Improved Detection of Relevant Routes in Weighted Metabolic Networks. *Journal of Computational Biology 15*, 6 (2008), 565–576.

[17] BOYER, F., AND VIARI, A. Ab initio reconstruction of metabolic pathways. *Bioinformatics 19*, 90002 (2003), 26ii–34.

[18] Brakhage, A. A. Molecular regulation of beta-lactam biosynthesis in filamentous fungi. *Microbiology and Molecular Biology Reviews 62*, 3 (1998), 547–85.

[19] Brohée, S., Faust, K., Lima-Mendez, G., Sand, O., Janky, R., Vanderstocken, G., Deville, Y., and van Helden, J. NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Research 36*, Web Server issue (2008), W444–51.

[20] Carter, O. A., Peters, R. J., and Croteau, R. Monoterpene biosynthesis pathway construction in Escherichia coli. *Phytochemistry 64*, 2 (2003), 425–33.

[21] Caspi, R., Foerster, H., Fulcher, C. A., Kaipa, P., Krummenacker, M., Latendresse, M., Paley, S., Rhee, S. Y., Shearer, A. G., Tissier, C., Walk, T. C., Zhang, P., and Karp, P. D. The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Research 36*, Database issue (2008), D623–31.

[22] Chang, A., Scheer, M., Grote, A., Schomburg, I., and Schomburg, D. BRENDA, AMENDA and FRENDA the enzyme information system: new content and tools in 2009. *Nucleic Acids Research 37*, Database issue (2009), D588–92.

[23] Chaouiya, C. Petri net modelling of biological networks. *Briefings in Bioinformatics 8*, 4 (2007), 210–9.

[24] Chen, Y., Deng, W., Wu, J., Qian, J., Chu, J., Zhuang, Y., Zhang, S., and Liu, W. Genetic modulation of the overexpression of tailoring genes eryK and eryG leading to the improvement of erythromycin A purity and production in Saccharopolyspora erythraea fermentation. *Applied and Environmental Microbiology 74*, 6 (2008), 1820–8.

[25] CRABTREE, J. D., AND MEHTA, D. P. Automated reaction mapping. *Journal of Experimental Algorithmics 13* (2009), 1.15.

[26] CROES, D., COUCHE, F., WODAK, S. J., AND VAN HELDEN, J. Inferring meaningful pathways in weighted metabolic networks. *Journal of Molecular Biology 356*, 1 (2006), 222–236.

[27] DE FIGUEIREDO, L. F., SCHUSTER, S., KALETA, C., AND FELL, D. A. Can sugars be produced from fatty acids? A test case for pathway analysis tools. *Bioinformatics 25*, 1 (2008), 152–158.

[28] DEMAIN, A. L., AND ELANDER, R. P. The β-lactam antibiotics: past, present, and future. *Antonie van Leeuwenhoek 75*, 1 (1999), 5–19.

[29] DEVILLE, Y., GILBERT, D., VAN HELDEN, J., AND WODAK, S. J. An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics 4*, 3 (2003), 246–59.

[30] DIERKING, E. C., AND BILYEU, K. D. Raffinose and stachyose metabolism are not required for efficient soybean seed germination. *Journal of Plant Physiology 166*, 12 (2009), 1329–35.

[31] DREWS, J. Drug Discovery: A Historical Perspective. *Science 287*, 5460 (2000), 1960–1964.

[32] EDWARDS, J. S., IBARRA, R. U., AND PALSSON, B. O. In silico predictions of Escherichia coli metabolic capabilities are consistent with experimental data. *Nature Biotechnology 19*, 2 (2001), 125–30.

[33] EPPSTEIN, D. Finding the k Shortest Paths. *SIAM Journal on Computing 28*, 2 (1998), 652–673.

[34] FAUST, K., CROES, D., AND VAN HELDEN, J. In response to 'Can sugars be produced from fatty acids? A test case for pathway analysis tools'. *Bioinformatics 25*, 23 (2009), 3202–5.

[35] FAUST, K., CROES, D., AND VAN HELDEN, J. Metabolic pathfinding using RPAIR annotation. *Journal of Molecular Biology 388*, 2 (2009), 390–414.

[36] FEIST, A. M., HERRGÅ RD, M. J., THIELE, I., REED, J. L., AND PALSSON, B. O. Reconstruction of biochemical networks in microorganisms. *Nature Reviews Microbiology 7*, 2 (2009), 129–43.

[37] FISCHER, S., HOPKINSON, D., LIU, M., MACLEAN, A. A., EDWARDS, V., CUTZ, E., AND KESHAVJEE, S. Raffinose improves 24-hour lung preservation in low potassium dextran glucose solution: a histologic and ultrastructural analysis. *The Annals of Thoracic Surgery 71*, 4 (2001), 1140–5.

[38] FLATT, P. M., AND MAHMUD, T. Biosynthesis of aminocyclitol-aminoglycoside antibiotics and related compounds. *Natural Product Reports 24*, 2 (2007), 358–92.

[39] GAREY, M. R., AND JOHNSON, D. S. Computers and Intractability; A Guide to the Theory of NP-Completeness.

[40] GASTEIGER, E. ExPASy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Research 31*, 13 (2003), 3784–3788.

[41] GERLEE, P., LIZANA, L., AND SNEPPEN, K. Pathway identification by network pruning in the metabolic network of Escherichia coli. *Bioinformatics 25*, 24 (2009),

3282–8.

[42] GOTO, S., NISHIOKA, T., AND KANEHISA, M. LIGAND: chemical database of enzyme reactions. *Nucleic Acids Research 28*, 1 (2000), 380–2.

[43] GREGONIS, D. E., AND RILLING, H. C. The stereochemistry of trans-phytoene synthesis. Some observations on lycopersene as a carotene precursor and a mechanism for the synthesis of cis- and trans-phytoene. *Biochemistry 13*, 7 (1974), 1538–42.

[44] GUSFIELD, D. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1997.

[45] HATTORI, M., OKUNO, Y., GOTO, S., AND KANEHISA, M. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society 125*, 39 (2003), 11853–65.

[46] HATTORI, M., OKUNO, Y., GOTO, S., AND KANEHISA, M. Heuristics for chemical compound matching. *Genome Informatics 14* (2003), 144–53.

[47] HAYAISHI, O., AND KORNBERG, A. Metabolism of phospholipides by bacterial enzymes. *The Journal of Biological Chemistry 206*, 2 (1954), 647–63.

[48] HEINONEN, M., LAPPALAINEN, S., MIELIKAINEN, T., AND ROUSU, J. Computing Atom Mappings for Biochemical Reactions without Subgraph Isomorphism. *Journal of Computational Biology* (2010).

[49] HENRY, C. S., BROADBELT, L. J., AND HATZIMANIKATIS, V. Discovery and analysis of novel metabolic pathways for the biosynthesis of industrial chemicals: 3-hydroxypropanoate. *Biotechnology and Bioengineering 106*, 3 (2010), 462–473.

[50] HERRMANN, K. M., AND WEAVER, L. M. The Shikimate Pathway. *Annual Review of Plant Physiology and Plant Molecular Biology 50* (1999), 473–503.

[51] HORNE, A. B., HODGMAN, T. C., SPENCE, H. D., AND DALBY, A. R. Constructing an enzyme-centric view of metabolism. *Bioinformatics 20*, 13 (2004), 2050–5.

[52] JEONG, H., TOMBOR, B., ALBERT, R., OLTVAI, Z. N., AND BARABÁSI, A.-L. The large-scale organization of metabolic networks. *Nature 407*, 6804 (2000), 651–4.

[53] JIANG, X., MENG, X., AND XIAN, M. Biosynthetic pathways for 3-hydroxypropionic acid production. *Applied Microbiology and Biotechnology 82*, 6 (2009), 995–1003.

[54] JOUHTEN, P., PITKÄNEN, E., PAKULA, T., SALOHEIMO, M., PENTTILÄ, M., AND MAAHEIMO, H. 13C-metabolic flux ratio and novel carbon path analyses confirmed that Trichoderma reesei uses primarily the respirative pathway also on the preferred carbon source glucose. *BMC Systems Biology 3*, 1 (2009), 104.

[55] KANEHISA, M. The KEGG databases at GenomeNet. *Nucleic Acids Research 30*, 1 (2002), 42–46.

[56] KANEHISA, M., ARAKI, M., GOTO, S., HATTORI, M., HIRAKAWA, M., ITOH, M., KATAYAMA, T., KAWASHIMA, S., OKUDA, S., TOKIMATSU, T., AND YAMANISHI, Y. KEGG for linking genomes to life and the environment. *Nucleic Acids Research 36*, Database issue (2008), D480–4.

[57] KANEHISA, M., GOTO, S., HATTORI, M., AOKI-KINOSHITA, K. F., ITOH, M., KAWASHIMA, S., KATAYAMA, T., ARAKI, M., AND HIRAKAWA, M. From ge-

nomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research 34*, Database issue (2006), D354–7.

[58] KARP, P. D., OUZOUNIS, C. A., MOORE-KOCHLACS, C., GOLDOVSKY, L., KAIPA, P., AHRÉN, D., TSOKA, S., DARZENTAS, N., KUNIN, V., AND LÓPEZ-BIGAS, N. Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Research 33*, 19 (2005), 6083–9.

[59] KHOSLA, C., TANG, Y., CHEN, A. Y., SCHNARR, N. A., AND CANE, D. E. Structure and mechanism of the 6-deoxyerythronolide B synthase. *Annual Review of Biochemistry 76* (2007), 195–221.

[60] KLAMT, S., AND STELLING, J. Combinatorial Complexity of Pathway Analysis in Metabolic Networks. *Molecular Biology Reports 29*, 1 (2002), 233 – 236.

[61] KOCH, I., JUNKER, B. H., AND HEINER, M. Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics 21*, 7 (2005), 1219–26.

[62] KOTERA, M., HATTORI, M., OH, M., YAMAMOTO, R., KOMENO, T., GOTO, S., YABUZAKI, J., AND KANEHISA, M. RPAIR: a reactant-pair database representing chemical changes in. *Genome Informatics 15* (2004), P062.

[63] KÜFFNER, R., ZIMMER, R., AND LENGAUER, T. Pathway analysis in metabolic databases via differential metabolic display (DMD). *Bioinformatics 16*, 9 (2000), 825–836.

[64] KUZUYAMA, T. Mevalonate and nonmevalonate pathways for the biosynthesis of isoprene units. *Bioscience, Biotechnology, and Biochemistry 66*, 8 (2002), 1619–27.

[65] Lacroix, V., Cottret, L., Thébault, P., and Sagot, M.-F. An introduction to metabolic networks and their structural analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 5*, 4 (2008), 594–617.

[66] Lee, S., Tsao, R., Peterson, C., and Coats, J. R. Insecticidal activity of monoterpenoids to western corn rootworm (Coleoptera: Chrysomelidae), twospotted spider mite (Acari: Tetranychidae), and house fly (Diptera: Muscidae). *Journal of Economic Entomology 90*, 4 (1997), 883–92.

[67] Lee, S. K., Chou, H., Ham, T. S., Lee, T. S., and Keasling, J. D. Metabolic engineering of microorganisms for biofuels production: from bugs to synthetic biology to fuels. *Current Opinion in Biotechnology 19*, 6 (2008), 556–63.

[68] Lee, S. Y., Kim, H. U., Park, J. H., Park, J. M., and Kim, T. Y. Metabolic engineering of microorganisms: general strategies and drug production. *Drug Discovery Today 14*, 1-2 (2009), 78–88.

[69] Lehninger, A., Nelson, D. L., and Cox, M. M. *Lehninger Principles of Biochemistry*. W. H. Freeman, 2008.

[70] Lemer, C., Antezana, E., Couche, F., Fays, F., Santolaria, X., Janky, R., Deville, Y., Richelle, J., and Wodak, S. J. The aMAZE LightBench: a web interface to a relational database of cellular processes. *Nucleic Acids Research 32*, Database issue (2004), D443–8.

[71] Leuchtenberger, W., Huthmacher, K., and Drauz, K. Biotechnological production of amino acids and derivatives: current status and prospects. *Applied Microbiology and Biotechnology 69*, 1 (2005), 1–8.

[72] LICHTENTHALER, H. Biosynthesis of isoprenoids in higher plant chloroplasts proceeds via a mevalonate-independent pathway. *FEBS Letters 400*, 3 (1997), 271–274.

[73] MA, H., AND ZENG, A.-P. Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics 19*, 2 (2003), 270–7.

[74] MAHMOUD, S. S., AND CROTEAU, R. B. Strategies for transgenic manipulation of monoterpene biosynthesis in plants. *Trends in Plant Science 7*, 8 (2002), 366–73.

[75] MASAMOTO, K., WADA, H., KANEKO, T., AND TAKAICHI, S. Identification of a gene required for cis-to-trans carotene isomerization in carotenogenesis of the cyanobacterium Synechocystis sp. PCC 6803. *Plant and Cell Physiology 42*, 12 (2001), 1398–402.

[76] MAVROVOUNIOTIS, M. L. Identification of qualitatively feasible metabolic pathways. 325–364.

[77] MAVROVOUNIOTIS, M. L., STEPHANOPOULOS, G., AND STEPHANOPOULOS, G. Computer-aided synthesis of biochemical pathways. *Biotechnology and Bioengineering 36*, 11 (1990), 1119–32.

[78] MCSHAN, D. C., RAO, S., AND SHAH, I. PathMiner: predicting metabolic pathways by heuristic search. *Bioinformatics 19*, 13 (2003), 1692–1698.

[79] MICHAL, G. *Biochemical Pathways: An Atlas of Biochemistry and Molecular Biology*. John Wiley & Sons, Inc, New York, NY, 1999.

[80] MINAMI, H., KIM, J.-S., IKEZAWA, N., TAKEMURA, T., KATAYAMA, T., KUMAGAI, H., AND SATO, F. Microbial production of plant benzylisoquinoline alkaloids. *Proceedings of the National Academy of Sciences 105*, 21 (2008), 7393–8.

[81] MONTES, G. H., MEJIA, J. J. D., RUEDA, E. P., AND SEGOVIA, L. The hidden universal distribution of amino acid biosynthetic networks: a genomic perspective on their origins and evolution. *Genome Biology 9*, 6 (2008), R95.

[82] MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE 77*, 4 (1989), 541–580.

[83] MURRAY, A. W. The biological significance of purine salvage. *Annual Review of Biochemistry 40* (1971), 811–26.

[84] MURRAY, J. F. A century of tuberculosis. *American Journal of Respiratory and Critical Care Medicine 169*, 11 (2004), 1181–6.

[85] NIELSEN, J. The role of metabolic engineering in the production of secondary metabolites. *Current Opinion in Microbiology 1*, 3 (1998), 330–336.

[86] ORTH, J. D., THIELE, I., AND PALSSON, B. O. What is flux balance analysis? *Nature Biotechnology 28*, 3 (2010), 245–8.

[87] PAL, S. A journey across the sequential development of macrolides and ketolides related to erythromycin. *Tetrahedron 62*, 14 (2006), 3171–3200.

[88] PATTNAIK, S., SUBRAMANYAM, V. R., BAPAJI, M., AND KOLE, C. R. Antibacterial and antifungal activity of aromatic constituents of essential oils. *Microbios 89*, 358 (1997), 39–46.

[89] PFEIFER, B., HU, Z., LICARI, P., AND KHOSLA, C. Process and metabolic strategies for improved production of Escherichia coli-derived 6-deoxyerythronolide B. *Applied and Environmental Microbiology 68*, 7 (2002), 3287–92.

[90] PHARKYA, P., BURGARD, A. P., AND MARANAS, C. D. OptStrain: A computational framework for redesign of microbial production systems. *Genome Research 14* (2004), 2367–2376.

[91] PITKÄNEN, E., JOUHTEN, P., AND ROUSU, J. Inferring branching pathways in genome-scale metabolic networks. *BMC Systems Biology 3*, 1 (2009), 103.

[92] PLANES, F. J., AND BEASLEY, J. E. A critical examination of stoichiometric and path-finding approaches to metabolic pathways. *Briefings in Bioinformatics 9*, 5 (2008), 422–36.

[93] PLANES, F. J., AND BEASLEY, J. E. Path finding approaches and metabolic pathways. *Discrete Applied Mathematics* (2008).

[94] R DEVELOPMENT CORE TEAM. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009.

[95] RAHMAN, S. A., ADVANI, P., SCHUNK, R., SCHRADER, R., AND SCHOMBURG, D. Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). *Bioinformatics 21*, 7 (2005), 1189–1193.

[96] RAMAN, K., AND CHANDRA, N. Flux balance analysis of biological systems: applications and challenges. *Briefings in Bioinformatics 10*, 4 (2009), 435–49.

[97] RAO, A. V., AND RAO, L. G. Carotenoids and human health. *Pharmacological Research 55*, 3 (2007), 207–16.

[98] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A. L. Hierarchical organization of modularity in metabolic networks. *Science 297*, 5586 (2002), 1551–5.

[99] Reeves, A. R., Brikun, I. A., Cernota, W. H., Leach, B. I., Gonzalez, M. C., and Weber, J. M. Engineering of the methylmalonyl-CoA metabolite node of Saccharopolyspora erythraea for increased erythromycin production. *Metabolic Engineering 9*, 3 (2007), 293–303.

[100] Sandmann, G. Carotenoid biosynthesis and biotechnological application. *Archives of Biochemistry and Biophysics 385*, 1 (2001), 4–12.

[101] Sandmann, G., Albrecht, M., Schnurr, G., Knörzer, O., and Böger, P. The biotechnological potential and design of novel carotenoids by gene combination in Escherichia coli. *Trends in Biotechnology 17*, 6 (1999), 233–7.

[102] Schuster, S., Dandekar, T., and Fell, D. A. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends in Biotechnology 17*, 2 (1999), 53–60.

[103] Schuster, S., Fell, D. A., and Dandekar, T. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology 18*, 3 (2000), 326–32.

[104] Schuster, S., and Hilgetag, C. On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems 2*, 2 (1994), 165–182.

[105] Schuster, S., Pfeiffer, T., Moldenhauer, F., Koch, I., and Dandekar, T. Exploring the pathway structure of metabolism: decomposition into subnetworks and application to Mycoplasma pneumoniae. *Bioinformatics 18*, 2 (2002), 351–61.

[106] Selifonova, O., Jessen, H., Gort, S., Selmer, T., and Buckel, W. 3-Hydroxypropionic acid and other organic compounds, 2002.

[107] Seressiotis, A., and Bailey, J. E. MPS: An algorithm and data base for metabolic pathway synthesis. *Biotechnology Letters 8*, 12 (1986), 837–842.

[108] Seressiotis, A., and Bailey, J. E. MPS: An artificially intelligent software system for the analysis and synthesis of metabolic pathways. *Biotechnology and Bioengineering 31*, 6 (1988), 587–602.

[109] Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research 13*, 11 (2003), 2498–504.

[110] Sprenger, N., and Keller, F. Allocation of raffinose family oligosaccharides to transport and storage pools in Ajuga reptans: the roles of two distinct galactinol synthases. *The Plant Journal 21*, 3 (2000), 249–258.

[111] Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann, E., and Willighagen, E. The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics. *Journal of Chemical Information and Computer Sciences 43*, 2 (2003), 493–500.

[112] Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R., and Willighagen, E. L. Recent Developments of the Chemistry Development Kit (CDK) - An Open-Source Java Library for Chemo- and Bioinformatics. *Current Pharmaceutical Design 12*, 17 (2006), 2111–2120.

[113] Strogatz, S. H. Exploring complex networks. *Nature 410*, 6825 (2001), 268–76.

[114] SUMMERS, R. G., DONADIO, S., STAVER, M. J., WENDT-PIENKOWSKI, E., HUTCHINSON, C. R., AND KATZ, L. Sequencing and mutagenesis of genes from the erythromycin biosynthetic gene cluster of Saccharopolyspora erythraea that are involved in L-mycarose and D-desosamine production. *Microbiology 143 (Pt 1* (1997), 3251–62.

[115] SUTHERS, P., AND CAMERON, D. Production of 3-hydroxypropionic acid in recombinant organisms, 2001.

[116] THYKAER, J. Metabolic engineering of β-lactam production. *Metabolic Engineering 5*, 1 (2003), 56–69.

[117] TORTUERO, F., FERNÁNDEZ, E., RUPÉREZ, P., AND MORENO, M. Raffinose and lactic acid bacteria influence caecal fermentation and serum cholesterol in rats. *Nutrition Research 17*, 1 (1997), 41–49.

[118] TRAVERT, S., VALERIO, L., FOURASTE, I., BOUDET, A. M., AND TEULIERES, C. Enrichment in Specific Soluble Sugars of Two Eucalyptus Cell-Suspension Cultures by Various Treatments Enhances Their Frost Tolerance via a Noncolligative Mechanism. *Plant Physiology 114*, 4 (1997), 1433–1442.

[119] TURNBAUGH, P. J., AND GORDON, J. I. An invitation to the marriage of metagenomics and metabolomics. *Cell 134*, 5 (2008), 708–13.

[120] VAN MARIS, A. J. A., KONINGS, W. N., VAN DIJKEN, J. P., AND PRONK, J. T. Microbial export of lactic and 3-hydroxypropanoic acid: implications for industrial fermentation processes. *Metabolic Engineering 6*, 4 (2004), 245–55.

[121] VARMA, A., AND PALSSON, B. O. Parametric sensitivity of stoichiometric flux balance models applied to wild-type Escherichia coli metabolism. *Biotechnology and*

Bioengineering 45, 1 (1995), 69–79.

[122] WAGNER, A., AND FELL, D. A. The small world inside large metabolic networks. *Proceedings of the Royal Society B : Biological Sciences 268*, 1478 (2001), 1803–10.

[123] WANG, F., JIANG, J. G., AND CHEN, Q. Progress on molecular breeding and metabolic engineering of biosynthesis pathways of C(30), C(35), C(40), C(45), C(50) carotenoids. *Biotechnology Advances 25*, 3 (2007), 211–22.

[124] WASHINGTON, J. A., AND WILSON, W. R. Erythromycin: a microbial and clinical perspective after 30 years of clinical use (1). *Mayo Clinic Proceedings 60*, 3 (1985), 189–203.

[125] WASHINGTON, J. A., AND WILSON, W. R. Erythromycin: a microbial and clinical perspective after 30 years of clinical use (2). *Mayo Clinic Proceedings 60*, 4 (1985), 271–8.

[126] WATVE, M. G., TICKOO, R., JOG, M. M., AND BHOLE, B. D. How many antibiotics are produced by the genus Streptomyces? *Archives of Microbiology 176*, 5 (2001), 386–90.

[127] WEBER, J. M., LEUNG, J. O., MAINE, G. T., POTENZ, R. H., PAULUS, T. J., AND DEWITT, J. P. Organization of a cluster of erythromycin genes in Saccharopolyspora erythraea. *Journal of Bacteriology 172*, 5 (1990), 2372–83.

[128] WENDISCH, V. F. *Amino acid biosynthesis: pathways, regulation, and metabolic engineering*. Springer, 2007.

[129] WHITE, R. H. Purine biosynthesis in the domain Archaea without folates or modified folates. *Journal of Bacteriology 179*, 10 (1997), 3374–7.

[130] XUE, L., GODDEN, J. W., STAHURA, F. L., AND BAJORATH, J. Design and evaluation of a molecular fingerprint involving the transformation of property descriptor values into a binary classification scheme. *Journal of Chemical Information and Computer Sciences 43*, 4 (2003), 1151–7.

[131] YEN, J. Y. Finding the K Shortest Loopless Paths in a Network. *Management Science 17*, 11 (1971), 712–716.

[132] YOON, S.-H., KIM, J.-E., LEE, S.-H., PARK, H.-M., CHOI, M.-S., KIM, J.-Y., LEE, S.-H., SHIN, Y.-C., KEASLING, J. D., AND KIM, S.-W. Engineering the lycopene synthetic pathway in E. coli by comparison of the carotenoid genes of Pantoea agglomerans and Pantoea ananatis. *Applied Microbiology and Biotechnology 74*, 1 (2007), 131–9.

[133] ZEVEDEI-OANCEA, I., AND SCHUSTER, S. Topological analysis of metabolic networks based on Petri net theory. *In Silico Biology 3*, 3 (2003), 323–45.

[134] ZHANG, Y., MORAR, M., AND EALICK, S. E. Structural biology of the purine biosynthetic pathway. *Cellular and Molecular Life Sciences 65*, 23 (2008), 3699–724.

[135] ZHENG, G. Q., KENNEY, P. M., AND LAM, L. K. Anethofuran, carvone, and limonene: potential cancer chemopreventive agents from dill weed oil and caraway oil. *Planta Medica 58*, 4 (1992), 338–41.

[136] ZRENNER, R., STITT, M., SONNEWALD, U., AND BOLDT, R. Pyrimidine and purine biosynthesis and degradation in plants. *Annual Review of Plant Biology 57* (2006), 805–36.