Appeared in Journal of Computational Biology, Volume 18, Issue 11, p.1575-1597 (2011) DOI:10.1089/cmb.2011.0165

An Algorithm for Efficient Identification of Branched Metabolic Pathways

Allison P. Heath[1], George N. Bennett[2], Lydia E. Kavraki[1,3,4,*]

[1]Department of Computer Science, Rice University, Houston, TX, USA

[2]Department of Biochemistry and Cell Biology, Rice University, Houston, TX, USA

[3]Department of Bioengineering, Rice University, Houston, TX, USA

[4]Structural and Computational Biology and Molecular Biophysics,
Baylor College of Medicine, Houston, TX, USA

[*] Corresponding Author: kavraki@rice.edu

Abstract

This paper presents a new graph-based algorithm for identifying branched metabolic pathways in multi-genome scale metabolic data. The term *branched* is used to refer to metabolic pathways between compounds that consist of multiple pathways that interact biochemically. A branched pathway may produce a target compound through a combination of linear pathways that split compounds into smaller ones, work in parallel with many compounds, and join compounds into larger ones. While branched metabolic pathways predominate in metabolic networks, most previous work has focused on identifying linear metabolic pathways. The ability to automatically identify branched pathways is important in applications that require a deeper understanding of metabolism, such as metabolic engineering and drug target identification. The algorithm presented in this paper utilizes explicit atom tracking to identify linear metabolic pathways and then merges them together into branched metabolic pathways. We provide results on several well characterized metabolic pathways that demonstrate that the new merging approach can efficiently find biologically relevant branched metabolic pathways.

1 Introduction

The quantity and quality of metabolic data has greatly increased in the last few decades, as indicated by the growth of such databases as the Kyoto Encyclopedia of Genes and Genomes (KEGG) [20] and MetaCyc [8]. Gaining understanding from this vast quantity of metabolic data requires novel computational tools that enable automatic identification and thorough analysis of biologically relevant metabolic pathways. These computational tools may reveal novel or alternative metabolic pathways, spanning single or multiple species, that could not have been identified by manual means. The ability to find metabolic pathways in multi-genome scale data has important applications in areas such as metabolic engineering [4], understanding the metabolic scope of multi-species communities [15, 43] and metabolic network reconstruction [34, 14].

The central problem in computational metabolic path finding is the following: given a start and target compound, find and return *biologically relevant* or *realistic* pathways of enzymatic reactions that produce the target compound from the start compound. Previous work in this area has primarily focused on finding linear sequences of reactions between start and target compounds [34]. However, more complex metabolic pathways, termed *branched pathways*, are dominant in metabolic networks and provide a more complete picture of metabolic processes [26, 33]. Branched pathways consist of multiple pathways that interact biochemically. For example, a start compound may be split into smaller compounds which, in parallel, undergo several different chemical reactions. The resulting products can then combine to form the target compound. A diagram of a generic branched metabolic pathway is depicted in Figure 1. The identification of branched pathways enables the analysis of metabolic pro-

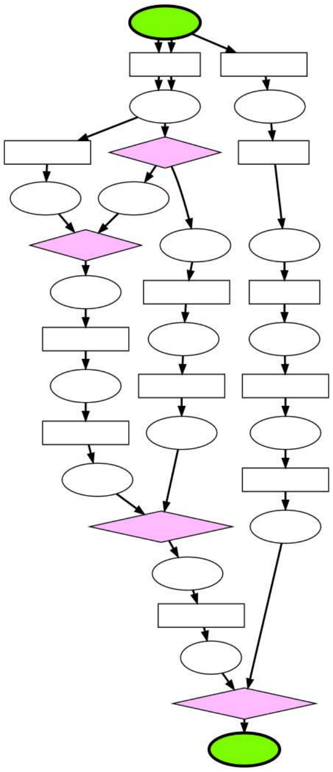

Figure 1: A generic depiction of a branched metabolic pathway. The green nodes represent the start and target compounds, the ellipses represent compounds, the boxes represent reactions along linear pathways that compose the branched pathway and the pink diamonds represent the reactions that occur at branch points. Each edge represents one molecule of each compound and the arrows indicate the direction of the reactions. In this example, three molecules of the start compound are required to produce one molecule of the target compound.

cesses with a more comprehensive perspective as compared to the limited picture provided by linear pathways.

The main contribution of this paper is a novel graph-based algorithm for identifying branched metabolic pathways by using atom tracking information to merge linear pathways. The merging approach of the presented algorithm is different from previous graph-based approaches, which start from a single linear pathway and then find new linear pathways to attach as branches [33, 18]. The results demonstrate that the new algorithm is able to efficiently find different network topologies in multi-genome scale data obtained from KEGG. This paper is an extended version of [19]. The paper proceeds as follows: Section 2 describes the relevant previous work in the area of graph-based metabolic path finding; Section 3 describes how our new algorithm merges linear pathways to find branched pathways; Section

4 contains the results for five well-characterized branched metabolic pathways; Section 5 concludes the paper.

2 Previous Work

Graph-based Models for Finding Metabolic Pathways Graphs provide a natural, well-studied computational model for identifying biologically relevant pathways in metabolic networks [13]. Graph-based metabolic path finding algorithms complement stoichiometric approaches, as they focus on different aspects of modeling and understanding metabolism [34, 14, 11]. Stoichiometric models are typically utilized for modeling specific organisms or metabolic systems [17]. Most stoichiometric models are based on the steady-state assumption and therefore require explicit labeling of internal and external compounds [22]. 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 [34]. However, both types of models are important for gaining insights into metabolic networks.

Graph-based methods have suffered from the disadvantage of finding pathways with spurious connections [3]. Several approaches have been developed to try to overcome this problem, such as removing certain currency metabolites from the graph [45, 16, 36], adding weights based on the degree of the nodes [10, 14] or using measures of structural similarity between compounds [25, 35]. We build on work that utilizes *atom-mapping data*, an approach more closely related to the underlying biochemistry [2, 3]. Atom-mapping data provides a systematic way of understanding a biochemical reaction by providing a specific mapping between each atom in the input compounds of a reaction to an atom in the output compounds. In the last few years, the availability of atom-mapping data has been steadily increasing, with one of the primary sources being the KEGG RPAIR database [21, 24].

Previous work has mainly used atom-mapping data for finding metabolic pathways by only allowing connections through reactions where at least one atom is being transfered from input to output compound [14, 27], or only returning pathways that conserve at least one atom, typically carbon [2, 3, 5, 6]. However, there are often instances where it is biochemically relevant to find pathways which conserve a high percentage of atoms from start to target compounds [7, 33]. Finding atom conserving pathways allow graph-based algorithms to account for the compounds that provide the atoms in the target compound, without having to use full stoichiometric constraints. The algorithm presented in this paper is based on our earlier work that finds atom conserving pathways by explicitly tracking multiple atoms in metabolic networks [18].

Foundation for the Presented Work The algorithm presented in this paper utilizes a graph-based structure that incorporates atom-mapping data called an *atom-mapping graph*, G_{am}, whose design is based on the observation that the same atom-mapping pattern between two compounds often appears in multiple reactions [2]. G_{am} is a directed bipartite graph containing *compound nodes* and *mapping nodes*. The compound nodes have unique identifiers for both the compound as well as its atoms. The compound nodes are connected by directed edges to mapping nodes that explicitly specify, via the unique identifiers, what atoms from the input compound become the atoms in the output compound. Each atom-mapping only maps the atoms between a pair of compounds and so the mapping nodes in G_{am} only have one input edge and one output edge connected to two different compound nodes, while the compound nodes have a degree equal to the number of mappings they participate in. Since the same atom-mapping can occur in many different reactions, a correspondence is stored between the mapping nodes and the reactions in which they occur. A more detailed description of the construction of G_{am} can be found in [18].

Previously, we developed and validated an algorithm for identifying the k shortest linear pathways in G_{am} that conserve at least a given number of atoms between desired start and target compounds [18]. This problem has been shown to be PSPACE-complete in the general case and NP-complete when a compound can only be used once in a pathway [7]. Previously unnamed, we will call the linear path finding algorithm from [18] LPAT, for Linear

Pathfinding with Atom Tracking. Based on LPAT, we also developed and validated a graph-based algorithm for identifying branched metabolic pathways. Also previously unnamed, we will call the branched path finding algorithm from [18] BPAT-S, for Branched Pathfinding using Atom Tracking and Seed pathways. The first step of BPAT-S uses LPAT to obtain a set of linear pathways between the desired start and target compounds. BPAT-S then annotates and stores the linear pathways with information about the specific reactions and compounds through which atoms are lost or gained. These annotated linear pathways are called *seed pathways* and indexed for efficient processing and attachment of branches. The branches are identified by calling LPAT to find linear pathways between compounds through which atoms are lost to compounds through which atoms are gained. These linear branches are then attached to the seed pathway to give rise to branched pathways, which are ranked first by the number of atoms they conserve and then by the total number of reactions they contain. In our previous study, we demonstrated that BPAT-S can find and return branched pathways that correspond to known branched pathways [18].

To the best of our knowledge, the only other comparable algorithm in the literature is the ReTrace algorithm [33]. The approach taken by ReTrace is similar to that of BPAT-S, but is based on pathways that only conserve one atom. In this paper, we present a novel algorithm that takes a significantly different approach from BPAT-S or ReTrace by merging linear pathways to form branched pathways. We provide results on five different metabolic pathways that demonstrate both the strengths and weaknesses of BPAT-S, ReTrace and the presented algorithm, BPAT-M.

3 BPAT-M: Branched Pathfinding by Merging Linear Pathways

This section describes a new algorithm, Branched Pathfinding using Atom Tracking and Merging (BPAT-M), for finding branched pathways by merging linear pathways returned by

LPAT. BPAT-M utilizes 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. This redundancy is eliminated in BPAT-M by carefully inventorying the linear pathways. BPAT-M takes advantage of the fact that linear pathways can only be merged together if the pathways do not have overlapping atoms in their target compounds. The atom tracking information from the linear pathways provided by LPAT are processed by BPAT-M to construct three data structures Q, C, and M. These data structures enable the efficient merging of linear pathways to find branched pathways. The construction of Q, C and M is described in Section 3.1. Section 3.2 then describes Algorithm 3.1, BPAT-M Search, which harnesses the extensive indexing of linear pathways contained in Q, M and C to find and return n branched pathways ranked first by the number of atoms conserved and second by the number of reactions.

3.1 Construction of Q, C and M from 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. Typically, the number of TAMs found is much less than the theoretical maximum number due to the biochemical nature of the pathways. On the left of Figure 2 there are three 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 for 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. If two pathways have disjoint TAMs, they can not necessarily be merged because the algorithm must check whether they share a common reaction. However, if two pathways are mergeable, then the TAM of the merged pathway is the union of the TAMs of the pathways.

The ability to use the TAMs to quickly determine if pathways are not mergeable mo-

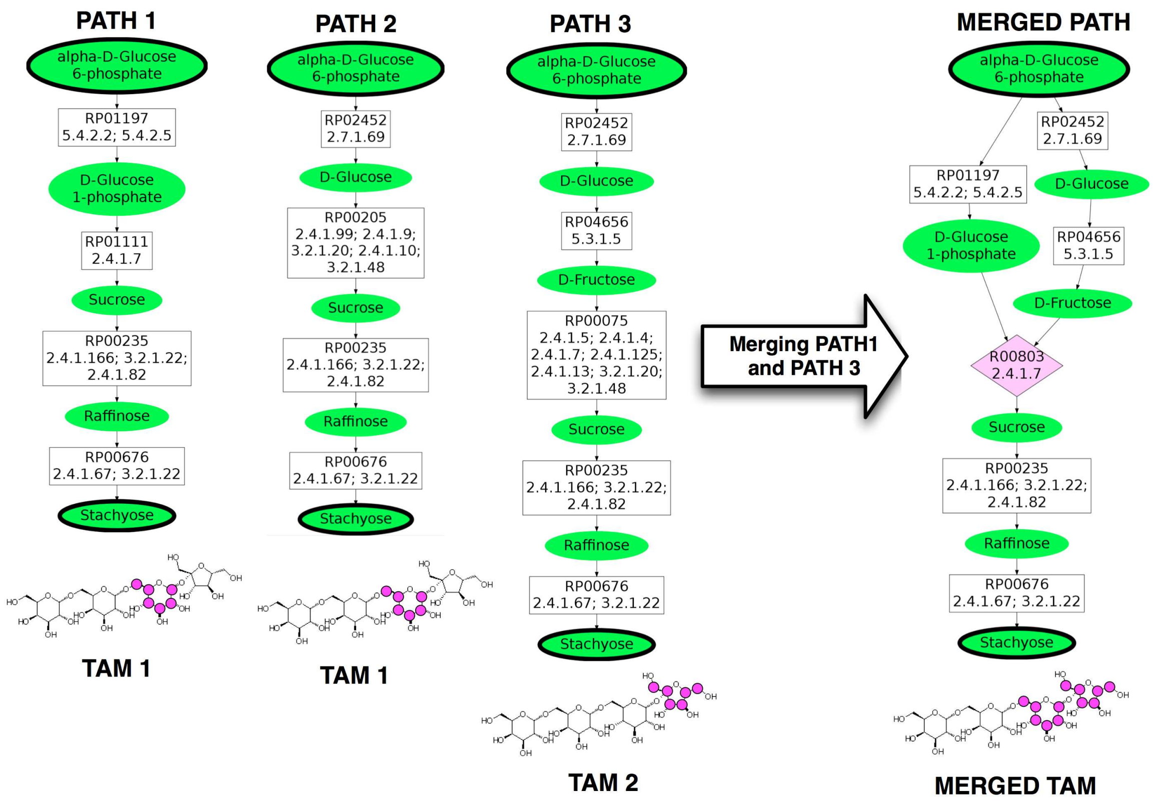
PATH 1
alpha-D-Glucose 6-phosphate
RP01197 5.4.2.2; 5.4.2.5
D-Glucose 1-phosphate
RP01111 2.4.1.7
Sucrose
RP00235 2.4.1.166; 3.2.1.22; 2.4.1.82
Raffinose
RP00676 2.4.1.67; 3.2.1.22
Stachyose
TAM 1
PATH 2
alpha-D-Glucose 6-phosphate
RP02452 2.7.1.69
D-Glucose
RP00205 2.4.1.99; 2.4.1.9; 3.2.1.20; 2.4.1.10; 3.2.1.48
Sucrose
RP00235 2.4.1.166; 3.2.1.22; 2.4.1.82
Raffinose
RP00676 2.4.1.67; 3.2.1.22
Stachyose
TAM 1
PATH 3
alpha-D-Glucose 6-phosphate
RP02452 2.7.1.69
D-Glucose
RP04656 5.3.1.5
D-Fructose
RP00075 2.4.1.5; 2.4.1.4; 2.4.1.7; 2.4.1.125; 2.4.1.13; 3.2.1.20; 3.2.1.48
Sucrose
RP00235 2.4.1.166; 3.2.1.22; 2.4.1.82
Raffinose
RP00676 2.4.1.67; 3.2.1.22
Stachyose
TAM 2
Merging PATH1 and PATH 3
MERGED PATH
alpha-D-Glucose 6-phosphate
RP01197 5.4.2.2; 5.4.2.5
RP02452 2.7.1.69
D-Glucose 1-phosphate
D-Glucose
RP04656 5.3.1.5
D-Fructose
R00803 2.4.1.7
Sucrose
RP00235 2.4.1.166; 3.2.1.22; 2.4.1.82
Raffinose
RP00676 2.4.1.67; 3.2.1.22
Stachyose
MERGED TAM

Figure 2: Three linear pathways from α-D-glucose 6-phosphate to stachyose and their associated carbon TAMs, as indicated by the magenta circles. The two potentially mergeable pairs of paths are PATH 1 with PATH 3 and PATH 2 with PATH 3. The result of merging PATH 1 and PATH 3 is displayed on the right.

tivates the construction of the data structure, Q, which maps a TAM to a list 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, sorted by their length. For example, using the pathways depicted in Figure 2, $Q[\text{TAM 1}]$ 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 hypothetical 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 a larger 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 merge combinations of linear pathways to obtain branched pathways.

Once potentially mergeable linear pathways have been identified using Q and C, they must be further compared to see if they can be merged through a common reaction r. The data structure M is constructed to store the results of comparing pairs of linear pathways for mergeability, thus the comparison is only performed once. M maps all pairs of mergeable linear pathways to a tuple containing r and the number of mapping nodes from the target compound that r occurs. M is constructed by first identifying all pairs of pathways, p_1 and p_2, with disjoint TAMs. The mapping nodes of p_1 and p_2 are compared starting from the target compound. This comparison identifies the position, m, of the mapping nodes closest to the target compound that differs between the two pathways. Only the position m is considered because if merging two paths results in a larger TAM, they must share a common reaction at this point. It is possible that two paths may also interact closer to the start compound, but this is currently not considered by the algorithm because it does not impact the TAM. In Figure 2, the comparison between PATH 1 and PATH 3 would identify RP01111 and RP00075 as the different mapping nodes closest to the target compound and m would be 2, using zero-based indexing. The final step is to look up the reactions that are associated with the two mapping nodes at m and determine if the mapping nodes share a common reaction that can be used to merge the two pathways. If there is no common reaction, then the pathways are not mergeable. In the case of PATH 1 and PATH 3 in Figure 2, both RP01111 and RP00075 are found in the reaction R00803 (EC Number 2.4.1.7) in KEGG. The right side of Figure 2 depicts PATH 1 and PATH 3 merged by R00803. This information about the mergeability of PATH 1 and PATH 3 in Figure 2 would then be stored as $M[PATH1, PATH3] = (R00803, 2)$.

3.2 Finding Branched Pathways Using Q, C and M

After processing and indexing the linear pathways to construct Q, C and M, these data structures are given as input to Algorithm 3.1 along with the number of branched pathways to return, n, and a fixed beam width, w, which can be used to bound the search. Algorithm 3.1 then returns the final result of BPAT-M, the top n branched pathways found ranked first by the number of atoms conserved and then by the number of reactions. Despite reducing the number of linear pathways combinations that need to be tested, the number of such combinations sometimes remains quite large. Therefore, the algorithm performs a beam search with a fixed beam width, w, which is provided by the user. The heuristic used for the beam search is discussed in more detail later in the section, and its usage means that BPAT-M does not guarantee finding the optimal combination. However, the results demonstrate that the search performs well in practice.

Algorithm 3.1 works by taking each combination of TAMs $c \in C$ in turn and using them to build branched pathway combinations. 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 mergeability using M (lines 7-9). If a pair of pathways are mergeable, then they are stored in the set of Intermediate Branched Pathways (IBPs), $\mathcal{T}$. The IBPs store a list of mergeable linear pathways and their merge points. Then, for each subsequent TAM in c (line 10), all of the pathways associated with the TAM $c[k]$, $p_q \in Q[c[k]]$ are retrieved (line 14). Each $p_q \in Q[c[k]]$ is then tested for mergeability with each linear pathway in each IBP (lines 13-16). If p_q is mergeable 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 (line 16).

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

Algorithm 3.1 BPAT-M Search

```
Input: Pathways organized by their TAMs, $Q$; Sorted list of all combinations of disjoint TAMs, $C$; Mergeable pairs of paths, $M$; Number of pathways to return, $n$; Limit on Intermediate Branched Pathways (IBPs), $w$;
Output: Sorted list of branched pathways $\mathcal{P}$, containing linear pathways and merge points, sorted first by number of atoms conserved, then by total number of nodes
$\mathcal{P} \leftarrow \{\}$
for each $c$ in $C$ do
    if $\mathcal{P}$ contains more than $n$ pathways and the $n$th pathway conserves more atoms than the size of $c$ then
        Truncate $\mathcal{P}$ to $n$ pathways
        Break
    $\mathcal{T} \leftarrow \{\}$ //for storing the IBPs, sorted by the same criteria as $\mathcal{P}$
    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 $\mathcal{T}$
    for $k = 3$ to size of $c$ do
        $\mathcal{N} \leftarrow \mathcal{T}$
        $\mathcal{T} \leftarrow \{\}$
        for each IBP $P$ in $\mathcal{N}$ do
            for each linear pathway $p_q$ in $Q[c[k]]$ 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 $\mathcal{T}$
        if $\mathcal{T}$ contains more than $w$ pathways then
            Truncate $\mathcal{T}$ to $w$ pathways
    Add all pathways in $\mathcal{T}$ to $\mathcal{P}$
Return $\mathcal{P}$
```

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 compound in p_q is not contained in the other pathways. Otherwise, the merge point is invalid. As an example, there could be a reaction r that takes the substrate compounds a, b and c. If two pathways, p_1 and p_2 were merged together through r, with p_1 containing a and p_2 containing b, there are two possibilities for a third pathway p_3, that is potentially mergeable with p_1 at r. If p_3 contains c, then the merge point is still valid and the resulting branched pathway would contain p_1, p_2 and p_3 merged through r. However, if p_3 contained b, then the merge point is invalid, as p_2 has already been merged through b. By checking for validity, merging multiple pathways through the same reaction is handled in a general way and only limited by the substrates used in the reaction.

In Algorithm 3.1, if the merge point is valid, p_q is merged with the IBP and the resulting branched pathway is stored as another IBP (line 17). 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 we have observed that very large numbers of IBPs can be generated in practice. This resulted in the introduction of the beam width, w, 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 18-19). Since the pathways are first ranked by the number of atoms they conserve and C is sorted by the size of each combination, the search can terminate when n pathways have been found and the next combination to try is smaller than the TAM of the nth pathway (lines 3-5).

The final way in which the 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 by sorting the pathways by length and only keeping a user specified number of the shortest pathways for each TAM. Future work is needed to investigate the impact of these parameters and develop easier ways for users to understand and select the proper limitations for their application. Our results demonstrate that even with the heuristic limits, BPAT-M performs well in practice.

4 Results

We present results from BPAT-M, BPAT-S and ReTrace (v1.03) on five representative, biologically interesting, test cases for branched pathways. The target compounds are stachyose, erythromycin, cephalosporin C, inosine monophosphate (IMP) and lycopene. The starting compound for all of the pathways is α-D-Glucose 6-Phosphate (G6P), a common form of intracellular glucose. Section 4.1 describes the data and hardware used for all of the experiments. Section 4.2 contains the results from BPAT-M for the five test cases. Section

4.3 presents a comparison of the BPAT-M results with the pathways found by ReTrace and BPAT-S.

4.1 Experimental Setup

All KEGG data used in the following experiments was downloaded on February 5, 2011, containing 12,457 RPAIR entries and 8,406 REACTION entries. Each RPAIR entry contains an atom-mapping between a pair of compounds. The chemical structure of the compounds and a list of associated reactions are also contained in each RPAIR entry. For the experiments in this paper only carbon atoms were tracked, but the algorithms are able to handle any atom type that are included in the RPAIR data. For BPAT-M and BPAT-S, the KEGG RPAIR data is processed to obtain a universal index for each atom in each compound. The universal index is obtained for most of the compounds by using the indices of the compound structure data in each RPAIR entry. However, there are sometimes inconsistencies between the indices for the same compound in different RPAIR entries. If this is the case, an attempt is made to obtain the universal indices by using the compound structure in the KEGG LIGAND database. This is done by first checking whether there exists an isomorphism between the structure data found in the RPAIR entry and the data found in the LIGAND database. If there exists only one isomorphism, the compound's atoms are indexed using the structure from the LIGAND database. If there is more than one isomorphism, the RMSD is calculated between the two structures using the isomorphism mapping. If the RMSD is equal to zero, that isomorphism is used to index the atoms. Otherwise, the RPAIR is removed from the data set. Finally, any RPAIRs which consist of an atom-mapping where the atom types being mapped do not match are discarded. This processing removed 116 RPAIR entries, resulting in a G_{am} constructed from 12,341 RPAIR entries involving 6,159 compounds and providing atom-mappings for 7,782 reactions from more than 1,500 organisms. In the results presented in this paper the full G_{am} was used, but subgraphs of G_{am} corresponding to particular organisms, reactions or compounds of interest can easily be created and searched.

Additionally, reversibility information was obtained from XML representations of the KEGG metabolic pathway maps, distributed in the KEGG Markup Language (KGML). A reaction is considered irreversible if it is consistently labeled as such across all of the KEGG metabolic pathway maps. Otherwise, the reaction is 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 construction of G_{am}. 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 resulted in 3,698 reactions and 5,420 RPAIR entries considered irreversible. The reversibility information was also provided as input to ReTrace.

The implementation of BPAT-M was done in Java using the Chemical Development Kit [40] and the Java Universal Network/Graph Framework (http://jung.sourceforge.net/). All result figures are drawn using Graphviz (http://www.research.att.com/sw/tools/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.

4.2 BPAT-M Results

The BPAT-M searches all used the same parameter values: n was 100, w was 500 and each entry in Q was limited to the 2,000 shortest pathways. Since BPAT-M begins with a search by LPAT, the value for k given to LPAT must also be specified. The value of k is typically set quite high because the k corresponds to the value given to Eppstein's k-shortest path algorithm in LPAT, which returns cycles. However, pathways with cycles are typically undesirable and removed, returning only the simple paths for use in the branched pathway finding. For these experiments, k was set to 1,000,000. In the branched pathway figures the ellipses are compounds, with the start and target compounds highlighted in green, the boxes are mapping nodes that correspond to RPAIRs and the pink diamonds contain the KEGG ID and EC numbers for reactions. Each edge in the pathway figures correspond to one molecule of the compound being used as a substrate or product of the reaction. The mapping nodes also contain EC numbers for the reactions that are associated with the RPAIR. Since pathways returned are often quite large and not always very readable on paper, the full pathway figures can be found in the online supplementary material (URL at the end of the paper) for better viewing.

4.2.1 α-D-Glucose 6-Phosphate to Stachyose

Stachyose is part of the raffinose family oligosaccharides (RFOs), which are found in many plant species [39]. Starting from G6P, the stachyose pathway has three major branching points. The first is in making sucrose, typically from some form of fructose and glucose, the second and third is the addition of galactose, provided by α-D-galactosyl-(1→3)-1D-myo-inositol, to sucrose and then to raffinose to form stachyose [31].

The top ranking pathway returned by BPAT-M is depicted in Figure 3. The initial LPAT search was given as input to conserve at least three carbons, which returned 10,760 linear pathways. The linear pathways contained 15 different TAMs, resulting in 908 mutually exclusive combinations. The top ranking pathway found by BPAT-M closely corresponds to the known stachyose biosynthesis pathway. While the pathway used to produce sucrose may vary in different organisms, the top ten returned pathways for BPAT-M also varied by the reaction and pathways leading to sucrose.

4.2.2 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 [46, 47]. Erythromycin is produced by bacteria *Saccharopolyspora erythraea* and is difficult to synthesize in the laboratory [30]. Therefore, several metabolic engineering approaches have

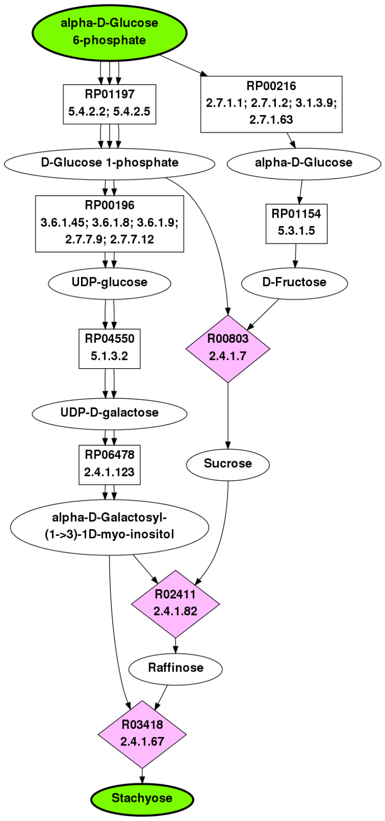

Figure 3: Top ranked pathway from G6P to Stachyose as found by BPAT-M. The ellipses are compounds, with the start and target compounds highlighted in green, the boxes are mapping nodes that correspond to RPAIRs and the pink diamonds contain the KEGG ID and EC numbers for reactions. Each edge in the pathway figures correspond to one molecule of the compound being used as a substrate or product of the reaction. This pathway uses four molecules of G6P to produce one molecule of stachyose.

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* [29, 32, 9, 37]. The biosynthesis of erythromycin has been well characterized 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 [23]. 6DB then undergoes a series of modifications that includes the attachment of two sugars, L-mycarose and D-desosamine, in order to produce erythromycin [48, 41].

The top ranked pathway returned by BPAT-M is highly similar to the known pathway for the biosynthesis of erythromycin using G6P as the start compound. This top ranked

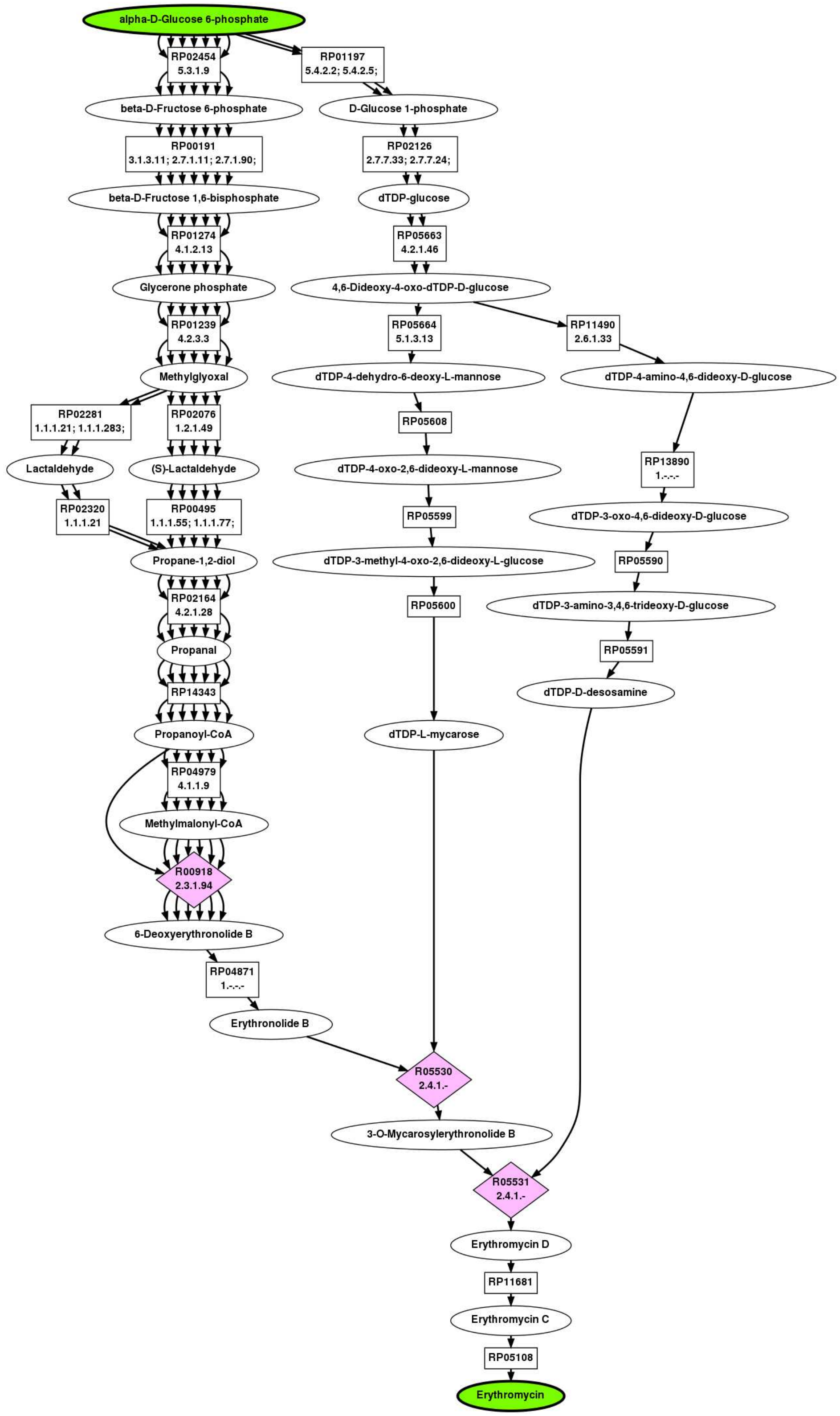

Figure 4: Full pathway returned as the top result for G6P to erythromycin by BPAT-M.

pathway is depicted in Figure 4. The initial LPAT search used by BPAT-M conserves three carbons, resulting in 11,217 linear pathways. The pathways have 16 unique TAMs which then produce 5,359 mutually exclusive combinations. The BPAT-M search took less than 4 minutes to complete.

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

Cephalosporin C is a β-lactam antibiotic, synthesized by certain bacteria and fungi, but not used clinically because of its low potency [12]. However, it is an important precursor for a number of related antibiotics and has been a target for increased production using metabolic engineering approaches [42]. The biosynthetic pathway for cephalosporin C includes an reaction that synthesizes δ-(L-α-aminoadipyl)-L-cysteinyl-D-valine (ACV) from L-valine, L-cysteine and L-2-aminoadipate. The pathway then proceeds through isopenicillin N which then undergoes a series of reactions resulting in cephalosporin C [42].

The top ranked pathway returned by BPAT-M is depicted in Figure 5. Three was the minimal number of carbons to conserve for the initial LPAT search. This resulted in 33,095 linear pathways with five unique TAMs. From the five TAMs, there were 18 mutually exclusive combinations. The pathway found by BPAT-M correctly identifies the crucial reaction catalyzed by ACV synthetase, which requires three different substrate compounds and produces ACV as the product. At the same time, the pathways from G6P to L-valine and L-cysteine would more likely use the glycolysis pathway to produce pyruvate. However, the reactions contained in the BPAT-M pathway are feasible and highlight the difficulty in judging the quality of the branched pathway. In the study of metabolism, especially when considering multiple organisms, it is often difficult to draw up just one “correct” branched pathway. Many pathways may be feasible or have interesting features, but perhaps not typically considered due to energetic or regulatory issues. In some applications, identifying these alternative or not as well characterized pathways may be preferred over returning already well studied pathways.

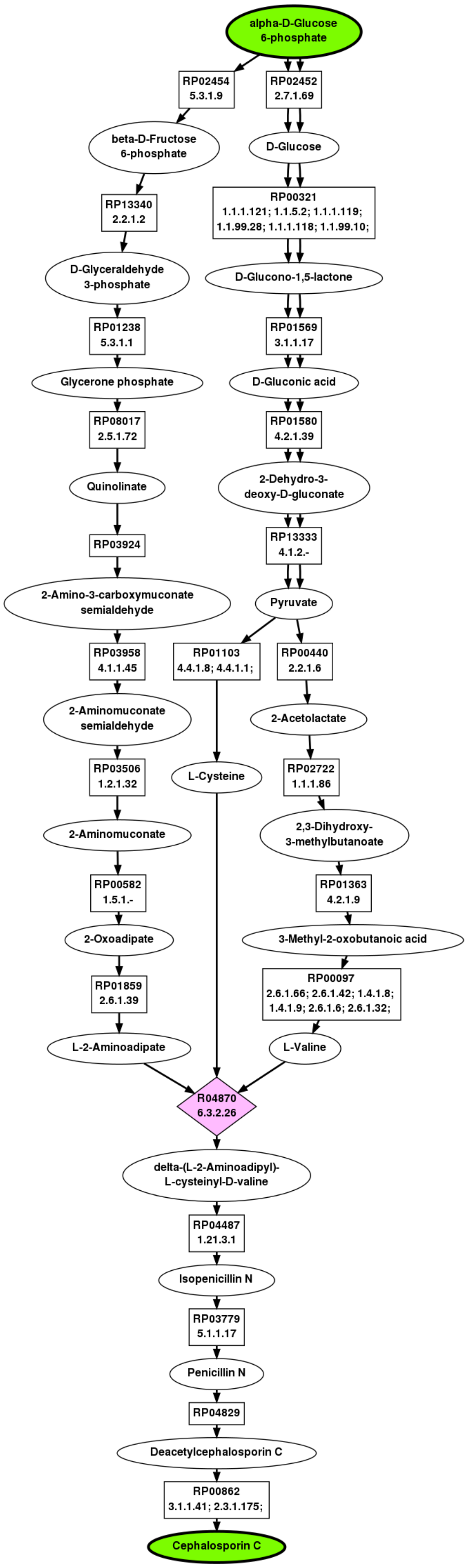

Figure 5: Top ranked branched metabolic pathways for G6P to cephalosporin C as found by BPAT-M.

4.2.4 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 carbons and one nitrogen [26]. 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 microorganisms all perform *de novo* synthesis of IMP in a similar manner [52]. In addition to the *de novo* pathway, because of the importance and energetic costs of synthesizing purine nucleotides and nucleosides, there are a number of purine salvage pathways that go through IMP [28]. These salvage pathways make finding the *de novo* pathway a challenging case.

The IMP pathway demonstrates a case where BPAT-M does not perform very well. The top ranked pathway found by BPAT-M is illustrated in Figure 6. This pathway was found by using linear pathways from LPAT that conserve one carbon. The linear pathways contained 14 unique TAMs, resulting in 190 mutually exclusive combinations. BPAT-M's performance is affected by the large number of short salvage pathways that conserve carbons from G6P to IMP through the ribose component. In the top ranked pathway, this is illustrated by the salvage reaction from 5-phospho-*alpha*-D-ribose 1-diphosphate (PRPP) to AICAR. The set of linear pathways found by LPAT for use in BPAT-M do not contain the relatively long pathways that build the purine base of IMP. The pathway through formate, found in the top ranked pathway, is of note because the pathway for IMP biosynthesis typically uses 10-

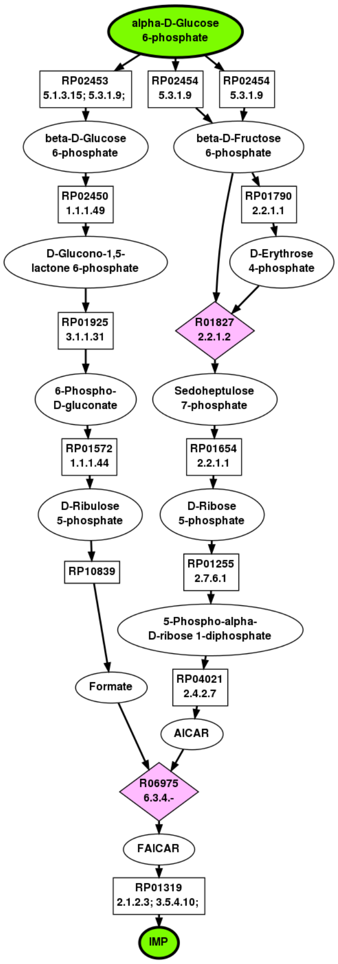

Figure 6: Top ranked pathway from G6P to IMP as found by BPAT-M.

formyltetrahydrofolate as the source for the formyl group added to AICAR to form FAICAR [26]. However, archea perform this reaction without folate or modified folates and utilize formate instead [49, 51]. The automatic finding of the formate reactions from archea demonstrates how searching over all organisms can retrieve interesting or non-standard reactions that might not be considered otherwise.

4.2.5 α-D-Glucose 6-Phosphate to Lycopene

Lycopene is a C_{40} carotenoid having a bright red color and is found in fruits and vegetables, such as tomatoes and watermelons. Lycopene's nutritional and pharmaceutical potential has resulted in a number of investigations on using metabolic engineering techniques to increase yield and/or produce lycopene in microbial hosts [1, 44, 50]. The known biosyn-

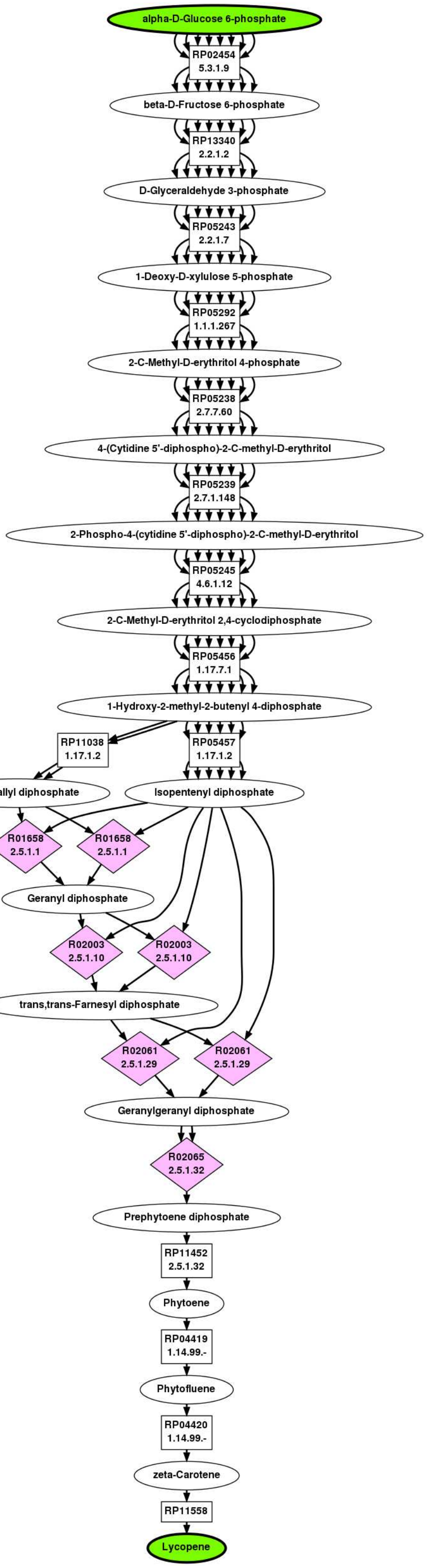
alpha-D-Glucose 6-phosphate
RP02454
5.3.1.9
beta-D-Fructose 6-phosphate
RP13340
2.2.1.2
D-Glyceraldehyde 3-phosphate
RP05243
2.2.1.7
1-Deoxy-D-xylulose 5-phosphate
RP05292
1.1.1.267
2-C-Methyl-D-erythritol 4-phosphate
RP05238
2.7.7.60
4-(Cytidine 5'-diphospho)-2-C-methyl-D-erythritol
RP05239
2.7.1.148
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
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 7: Top ranked branched metabolic pathway for G6P to lycopene as found by BPAT-M.

thesis pathway of lycopene is relatively well understood and contains an interesting "woven" topology; isopentenyl diphosphate (IPP) and dimethylallyl diphosphate (DMAPP) are produced by either the 2-C-methyl-D-erythritol 4-phosphate/1-deoxy-D-xylulose 5-phosphate (MEP/DOXP) or mevalonate (MVA) pathways, and then DMAPP 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 [38].

The top ranked pathway from G6P to lycopene returned by BPAT-M is depicted in Figure 7. The set of linear pathways used from LPAT were given as input three carbons to conserve, resulting in 31,726 linear pathways. These linear pathways contained 16 unique TAMs which were combined into 6,301 mutually exclusive combinations. BPAT-M successfully identifies the known "woven" topology of the lycopene biosynthesis pathway, starting with IPP and DMAPP. As shown in the next section, BPAT-M required significantly more time to search for the lycopene pathway compared to the other test cases. This is partially caused by having to test more mutually exclusive combinations of TAMs, which shows the importance of utilizing the biochemical constraints provided by the TAMs to reduce the running time. If the TAMs where not utilized, the number of combinations of linear pathways tested would require an unreasonable amount of computation.

4.3 Comparison of BPAT-M to Existing Algorithms for Finding Branched Metabolic Pathways

In this section, the results of BPAT-M as presented in Section 4.2 are compared to searches for the same pathways by BPAT-S [18] and ReTrace (version 1.03) [33]. BPAT-S and ReTrace were provided the same KEGG data, including the reversibility data, as described in Section 4.1. BPAT-S utilized the same G_{am} as BPAT-M. ReTrace processes the KEGG data separately and builds a different type of graph where each node corresponds to one atom. ReTrace reported 272 RPAIR errors and built a graph with 293,907 nodes and 300,115 edges.

For BPAT-S the default value of k given to the initial LPAT search was reduced to 500,000, as BPAT-S finds new linear pathways to be used as branches during the search. BPAT-S was implemented in Java and utilizes the same libraries as BPAT-M. ReTrace is implemented in Python.

The ReTrace parameter called k was set to the same values used for the IMP pathway search results reported by Pitkänen et al. [33]: 100 for the first depth, 25 for the second depth and 1 for the depths there after. The maximum depth for ReTrace was set to 10, due the large number of branches present in many of the test cases. The rest of the ReTrace parameters were left at their default values. The pathway figures for the ReTrace results contain all reactions because ReTrace selects a reaction for each RPAIR used in the search. BPAT-S and BPAT-M take a slightly different approach and return pathways that still contain the mapping nodes and only select reactions required for the branch points.

4.3.1 Wall Time for Finding Branched Metabolic Pathways

The wall times, excluding time for parsing and processing data, of the pathway searches by BPAT-M, ReTrace and BPAT-S are plotted in Figures 8(a), 8(b) and 8(c), respectively. For all cases, BPAT-M took the least amount of time, with BPAT-M and ReTrace both requiring significantly less time than BPAT-S. Due the heuristic nature of all of the algorithms, it is difficult to predict how much time a pathway search will require. Additionally, the amount of time required for each algorithm is dependent on parameter selection. Making the parameter values more restrictive will reduce the running time for all of the algorithms, at the expense of missing potential results. We chose parameters with the goal of having comparable levels of search, but this is still done in an *ad hoc* manner. Further investigation is required to fully understand the implications of different parameter values on the search and resulting pathways.

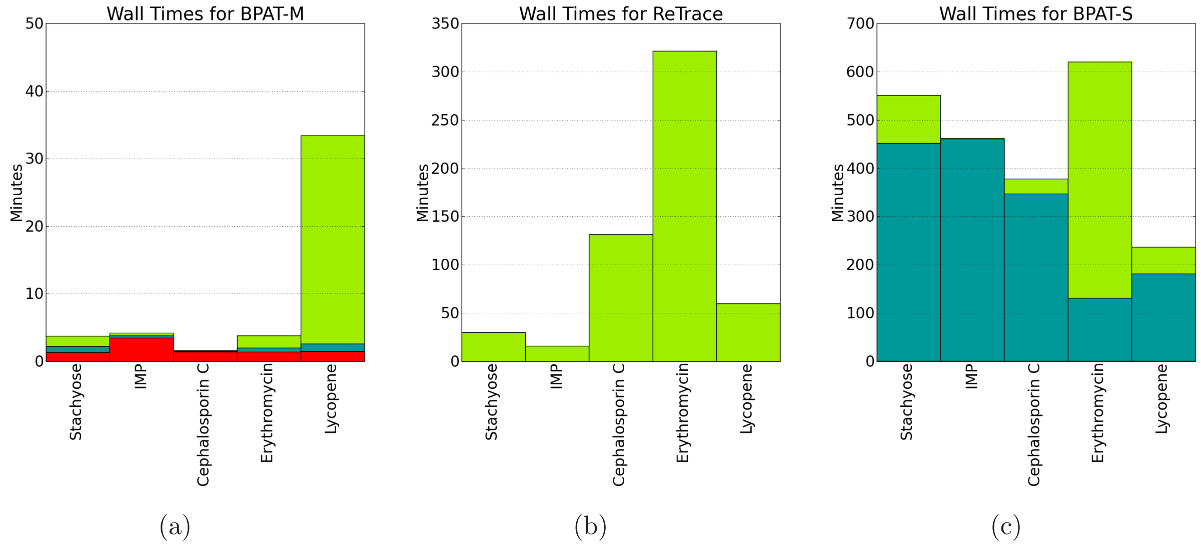

Figure 8: Computational time required for each test case for (a) BPAT-M (b) BPAT-S and (c) ReTrace. Note the different scales on the y-axis for each chart. The colors for BPAT-M and BPAT-S correspond to different stages of the algorithm. In (a) BPAT-M, red is the time used by LPAT to find the set of linear pathways; dark teal is the time used to construct M; 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 relatively negligible and is not visible. ReTrace only provides the total computational time required, which is plotted in (b). In (c) BPAT-S, dark teal is the time required for using LPAT to find the branches; light green is the time required to try all combination of branches; the time required for finding the original set of linear pathways is negligible and is not visible.

4.3.2 Comparison to BPAT-S

In general, the results found by BPAT-S do not justify the significantly longer times required, as compared to BPAT-M. For example, the top ranking pathway found by BPAT-S for stachyose is exactly the same as the top ranking pathway found by BPAT-M in Figure 3. To find the same pathway, BPAT-S took about 9 hours while BPAT-M took about 4 minutes. The one notable exception is the IMP pathway. The top ranking pathway returned by BPAT-S for IMP is illustrated in Figure 9. The methodology of BPAT-S enables it to start with a seed pathway for the ribose component of IMP and then search for a branch that corresponds to the construction of the purine base.

One of the branches that BPAT-S finds contains the proper sequence of reactions through glycine to AICAR. However, because the original seed pathway utilized a reversible salvage reaction from 5-phospho-α-D-ribose 1-diphosphate (PRPP) to AICAR, the attached branch

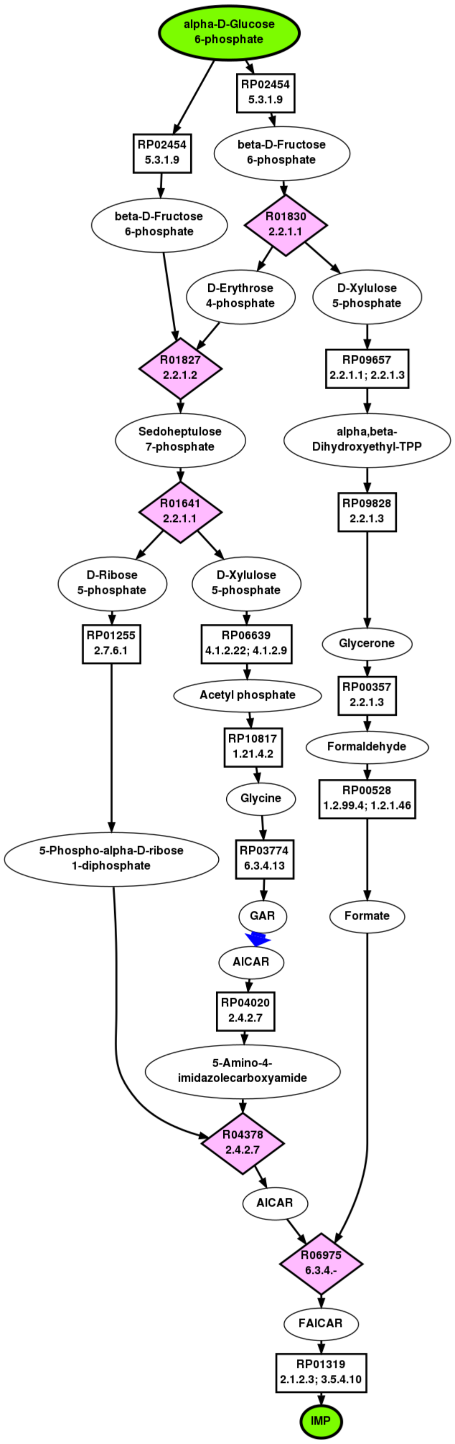

Figure 9: Top ranked pathway from G6P to IMP as found by BPAT-S. To fit on the page, the BPAT-S figure has been simplified by placing a larger blue arrow between GAR to AICAR to represent the path between them, which is described in the text and correctly found by BPAT-S. The full BPAT-S pathway, along with the top ranked pathways for all five test cases, can be viewed in the online supplementary material (URL at the end of the paper).

uses an unfavorable set of reactions that first remove the ribose component AICAR to make 5-amino-4-imidazolecarboxyamide. This is then combined with PRPP from the seed pathway to make FAICAR and continue to IMP. This unfavorable pathway found in the top ranking result of BPAT-S makes judging the quality of the path difficult. While the pathway has a high level of overlap with the known compounds and reactions composing the IMP biosynthesis pathway, it contains an unlikely branch point in a critical part of the pathway.

In a several other cases, BPAT-S is able to find branches that BPAT-M is unable to find because they do not exist in the original set of linear pathways. For example, in the erythromycin pathway, BPAT-S identifies a branch that goes through S-adenosyl-L-methionine which donates a methyl group. This branch does not appear in the BPAT-M results because it only provides one carbon to the target compound. However, BPAT-S required over 10 hours while BPAT-M takes only 4.5 minutes to find the core pathway for erythromycin. Additionally, BPAT-S is currently constrained to only find branches that start and end on the original seed pathway. ReTrace does not have this constraint and compares more favorably with BPAT-M.

4.3.3 Comparison to ReTrace

Overall, the pathways found by ReTrace identify similar key components of the branched pathways as described in Section 4.2. They also contain substantial differences from the pathways found by both BPAT-M and BPAT-S. These observations are demonstrated by the top ranked stachyose pathway returned by ReTrace, illustrated in Figure 10. The stachyose pathway found by ReTrace contains the key branch points of the formation of sucrose and the sequential additions of galactose from α-D-galactosyl-(1→3)-1D-myo-inositol, similar to the BPAT-M pathway depicted in Figure 3. However, the pathway also contains a number of additional reactions, often creating cycles between compounds in the pathway.

A similar comparison can be made of the erythromycin pathway returned by ReTrace, illustrated in Figure 11, with the pathway returned by BPAT-M. The erythromycin pathway

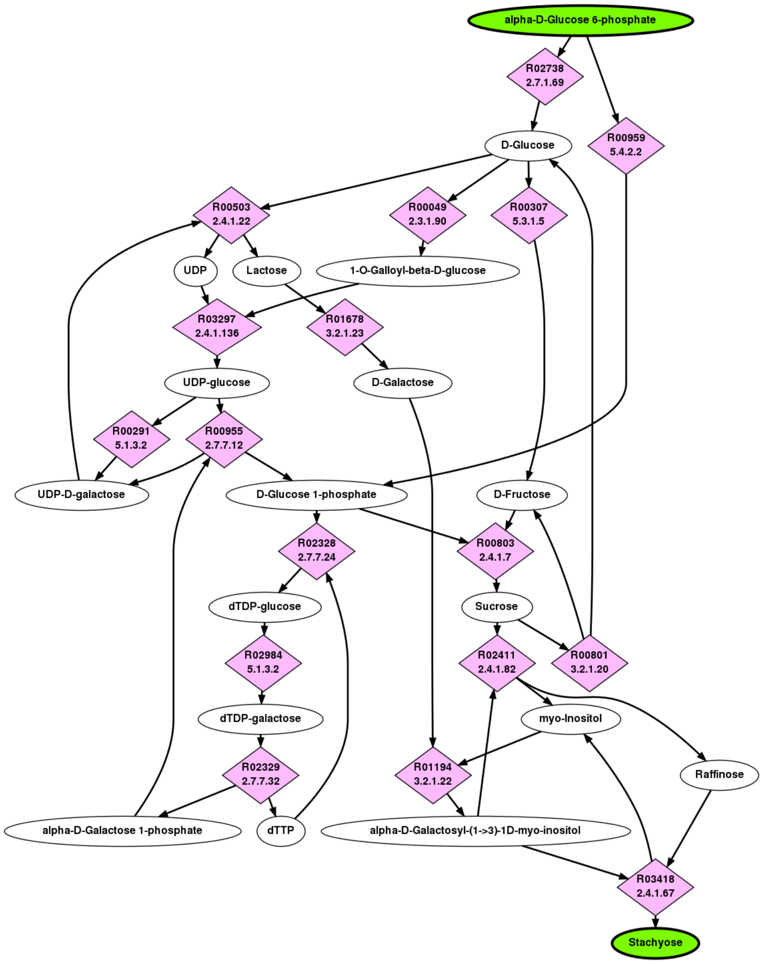

Figure 10: Top ranked pathway from G6P to Stachyose as found by ReTrace. For clarity, compounds other than the start and target compound that do not act as both substrates and products have been removed from the figure. The figure with all compounds included can be viewed in the online supplementary material (URL at the end of the paper).

from ReTrace finds the reactions that construct 6DB and attach the two sugars, L-mycarose and D-desosamine, but misses that six molecules of methylmalonyl-CoA are required. Similar to BPAT-S, the ReTrace pathway for erythromycin contains the branch through S-adenosyl-L-methionine. However, as with the BPAT-S pathway discussed in Section 4.3.2, ReTrace requires a much longer time to find the core part of the erythromycin pathway than BPAT-M.

The top ranking result for celphalosporin C from ReTrace, which can be found in the online supplementary material (URL at the end of the paper), provides a third example of the same comparative theme; the pathway contains the crucial reaction resulting in ACV while also containing a number of additional reactions and cycles not found in the BPAT-M

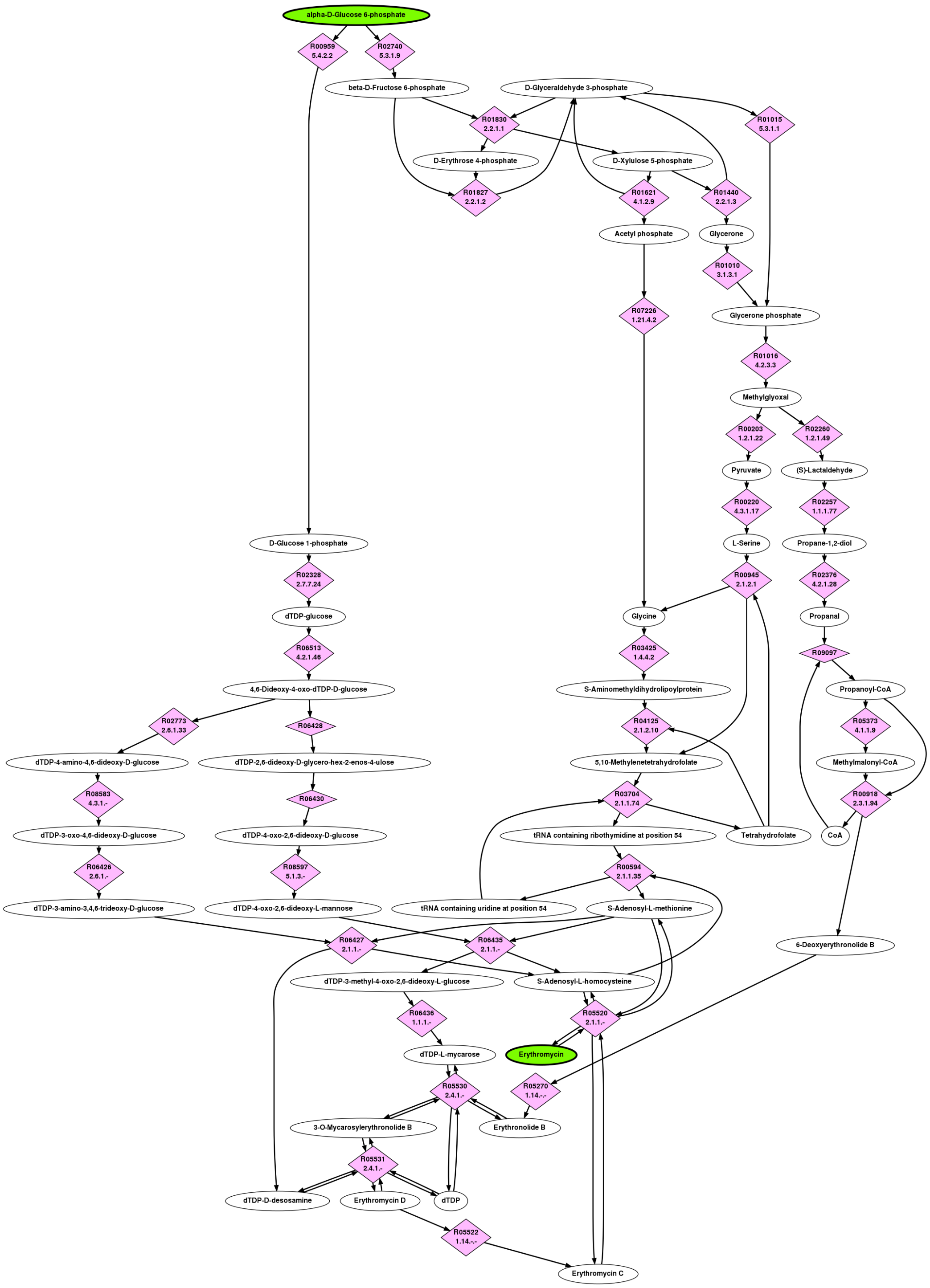

Figure 11: Top ranked pathway from G6P to Erythromycin as found by ReTrace. For clarity, compounds other than the start and target compound that do not act as both substrates and products have been removed from the figure. Due to the size of the pathway, the node labels may not be legible and the reader is referred to the online supplementary material (URL at the end of the paper) where the full pathway can be viewed in greater detail.

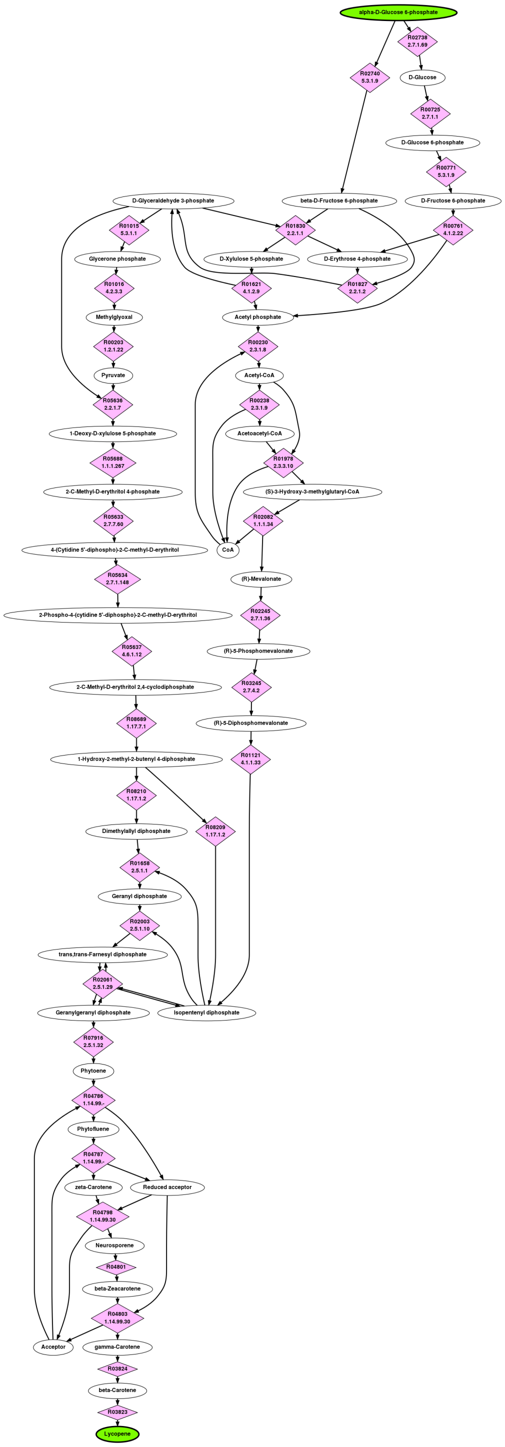

Figure 12: Top ranked pathway from G6P to Lycopene as found by ReTrace. For clarity, compounds other than the start and target compound that do not act as both substrates and products have been removed from the figure. Due to the size of the pathway, the node labels may not be legible and the reader is referred to the online supplementary material (URL at the end of the paper) where the full pathway can be viewed in greater detail.

pathway. The stachyose, erythromycin and celphalosporin C pathway results underscore the difficulty of comparing the quality of found branched metabolic pathways. One perspective is that the additional reactions and cycles are spurious should not be included in the pathway, making BPAT-M have the "better" result. Another perspective is that these reactions may have important implications for the core part of the stachyose pathway and should be included in the pathway, making ReTrace have the "better" result.

The top ranking pathway for lycopene found by ReTrace, depicted in Figure 12, contains both the MEP/DOXP pathway for DMAPP and MVA pathway for IPP, thus revealing the variety available. In comparison, the BPAT-M pathway in Figure 7 only uses the MEP pathway to synthesize both IPP and DMAPP. The MEP/DOXP and MVA pathways demonstrate how search tools for metabolic pathways can illuminate alternative pathways that may be found in different organisms. The ability to automatically find different pathways that can be used to produce the same compound is of great use in areas such as metabolic engineering or identifying and evaluating drug targets. At the same time, the ability to find alternative or novel pathways makes it more difficult to judge the performance of different algorithms. The resulting pathways from ReTrace and BPAT-M are biochemically correct in the usage of MEP/DOXP and MVA pathways, but one may be preferred over the other based on the specific application in mind.

5 Discussion

We have described and tested a new algorithm, BPAT-M, for identifying branched metabolic pathways that utilizes atom tracking information to efficiently merge together biochemically interacting linear pathways. The experimental results highlight both the strengths and weaknesses of the approach taken by BPAT-M, as compared to earlier approaches, namely BPAT-S and ReTrace. These results reveal that an algorithm's performance may depend on the underlying structure of the pathway. The merging approach used by BPAT-M will likely perform better if all of the branches conserve at least the given number of atoms and are of similar length. For example, the erythromycin pathway follows this scheme and BPAT-M is able to find pathway in minutes instead of the hours taken by ReTrace and BPAT-S. In contrast, BPAT-M does not perform well on the IMP pathway because it contains pathways of many different lengths that provide different numbers of atoms. This type of pathway may be better addressed using the approach taken by ReTrace and BPAT-S, where new branches are found as required. Since the merging approach of BPAT-M is complementary to the approach taken by ReTrace and BPAT-S, one future direction would be to combine the algorithms to take advantage of their different strengths. Other areas of future work include allowing multiple start and target compounds to be used as input, utilizing other sources of metabolic data and developing ways to mine the resulting pathways to help the user understand them.

Further work is also required to develop meaningful evaluation methods for branched metabolic pathways. The experimental results highlight that comparing branched metabolic pathways is nontrivial, especially if the goal is to find alternative or novel pathways. The evaluation process also depends on the particular application for the resulting pathways. In applications such as metabolic network reconstruction, it is typically desirable to find pathways that are similar to metabolic pathways known to exist in organisms. However, in applications such as metabolic engineering and synthetic biology, it can be desirable to find alternative pathways or pathways that do not exist in one single organism. At the same time, the algorithms must be validated so that there is confidence that the pathways are not meaningless. A further complication in evaluating the algorithms is that path finding algorithms use a number of heuristic parameters, due to the difficulty of the problem. In the case of the algorithms examined in this paper, the parameters adjust the trade-off between the solutions found and the running time of the search. While we strove to choose reasonable parameters for the different algorithms in this study, there remain many questions about the impact of parameter selection. Continuing to improve metabolic path finding will require the development of standard evaluation techniques and gaining better understanding of parameter choices. While it is currently difficult to identify *a priori* which method will perform better, it is reasonable to try different algorithms and analyze the results to gain better understanding of metabolic pathways.

Acknowledgments

APH and LEK were supported in part by NSF ABI-0960612, NHARP 01907, a fund from the John & Ann Doerr Fund for Computational Biomedicine at Rice University, and a Sloan Fellowship. APH is a fellow with the Shell Center for Sustainability at Rice University and is partially supported by a SCS grant. 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.

Author Disclosure Statement

No competing financial interests exist.

Online Supplementary Material

The full result pathways can be found at: http://www.kavrakilab.org/data/bpatm-jcb-2011-supp. Additionally, a web server for some of our metabolic path finding methods is located at: http://metabolicpaths.kavrakilab.org.

References

[1] H. Alper, K. Miyaoku, and G. Stephanopoulos. Construction of lycopene-overproducing E. coli strains by combining systematic and combinatorial gene knockout targets. *Nature Biotechnology*, 23(5):612–616, 2005.

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

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

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

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

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

[7] F. Boyer and A. Viari. Ab initio reconstruction of metabolic pathways. *Bioinformatics*, 19(Suppl. 2):ii26–ii34, 2003.

[8] R. Caspi, T. Altman, J. M. Dale, K. Dreher, C. A. Fulcher, F. Gilham, P. Kaipa, A. S. Karthikeyan, A. Kothari, M. Krummenacker, M. Latendresse, L. A. Mueller, S. Paley, L. Popescu, A. Pujar, A. G. Shearer, P. Zhang, and P. D. Karp. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research*, 38(Database issue):D473–9, 2010.

[9] Y. Chen, W. Deng, J. Wu, J. Qian, J. Chu, Y. Zhuang, S. Zhang, and W. Liu. 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):1820–8, 2008.

[10] D. Croes, F. Couche, S. J. Wodak, and J. van Helden. Inferring meaningful pathways in weighted metabolic networks. *Journal of Molecular Biology*, 356(1):222–236, 2006.

[11] L. F. de Figueiredo, S. Schuster, C. Kaleta, and D. A. Fell. Response to comment on 'Can sugars be produced from fatty acids? A test case for pathway analysis tools'. *Bioinformatics*, 25(24):3330–1, 2009.

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

[13] Y. Deville, D. Gilbert, J. van Helden, and S. J. Wodak. An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics*, 4(3):246–59, 2003.

[14] K. Faust, D. Croes, and J. van Helden. Metabolic pathfinding using RPAIR annotation. *Journal of Molecular Biology*, 388(2):390–414, 2009.

[15] A. M. Feist, M. J. Herrgå rd, I. Thiele, J. L. Reed, and B. O. Palsson. Reconstruction of biochemical networks in microorganisms. *Nature Reviews Microbiology*, 7(2):129–43, 2009.

[16] P. Gerlee, L. Lizana, and K. Sneppen. Pathway identification by network pruning in the metabolic network of Escherichia coli. *Bioinformatics*, 25(24):3282–8, 2009.

[17] A. K. Gombert and J. Nielsen. Mathematical modelling of metabolism. *Current Opinion in Biotechnology*, 11(2):180–6, 2000.

[18] A. P. Heath, G. N. Bennett, and L. E. Kavraki. Finding Metabolic Pathways Using Atom Tracking. *Bioinformatics*, 26(12):1548–1555, 2010.

[19] A. P. Heath, G. N. Bennett, and L. E. Kavraki. Identifying branched metabolic pathways by merging linear metabolic pathways. *15th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, In Press, 2011.

[20] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, 36(Database issue):D480–4, 2008.

[21] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(Database issue):D354–7, 2006.

[22] K. J. Kauffman, P. Prakash, and J. S. Edwards. Advances in flux balance analysis. *Current Opinion in Biotechnology*, 14(5):491–6, 2003.

[23] C. Khosla, Y. Tang, A. Y. Chen, N. A. Schnarr, and D. E. Cane. Structure and mechanism of the 6-deoxyerythronolide B synthase. *Annual Review of Biochemistry*, 76:195–221, 2007.

[24] M. Kotera, M. Hattori, M. Oh, R. Yamamoto, T. Komeno, S. Goto, J. Yabuzaki, and M. Kanehisa. RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions. *Genome Informatics*, 15:P062, 2004.

[25] D. C. McShan, S. Rao, and I. Shah. PathMiner: predicting metabolic pathways by heuristic search. *Bioinformatics*, 19(13):1692–1698, 2003.

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

[27] A. Mithani, G. M. Preston, and J. Hein. Rahnuma: hypergraph-based tool for metabolic pathway prediction and network comparison. *Bioinformatics*, 25(14):1831–2, 2009.

[28] A. W. Murray. The biological significance of purine salvage. *Annual Review of Biochemistry*, 40:811–26, 1971.

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

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

[31] T. Peterbauer and A. Richter. Biochemistry and physiology of raffinose family oligosaccharides and galactosyl cyclitols in seeds. *Seed Science Research*, 11(03):185–197, Sept. 2001.

[32] B. Pfeifer, Z. Hu, P. Licari, and C. Khosla. Process and metabolic strategies for improved production of Escherichia coli-derived 6-deoxyerythronolide B. *Applied and Environmental Microbiology*, 68(7):3287–92, 2002.

[33] E. Pitkänen, P. Jouhten, and J. Rousu. Inferring branching pathways in genome-scale metabolic networks. *BMC Systems Biology*, 3(1):103, 2009.

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

[35] S. A. Rahman, P. Advani, R. Schunk, R. Schrader, and D. Schomburg. Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). *Bioinformatics*, 21(7):1189–1193, 2005.

[36] S. Ranganathan and C. D. Maranas. Microbial 1-butanol production: Identification of non-native production routes and in silico engineering interventions. *Biotechnology Journal*, 5(7):716–25, 2010.

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

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

[39] N. Sprenger and F. Keller. 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):249–258, 2000.

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

[41] R. G. Summers, S. Donadio, M. J. Staver, E. Wendt-Pienkowski, C. R. Hutchinson, and L. Katz. 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 1:3251–62, 1997.

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

[43] P. J. Turnbaugh and J. I. Gordon. An invitation to the marriage of metagenomics and metabolomics. *Cell*, 134(5):708–13, 2008.

[44] R. V. Vadali, Y. Fu, G. N. Bennett, and K. Y. San. Enhanced lycopene productivity by manipulation of carbon flow to isopentenyl diphosphate in Escherichia coli. *Biotechnology Progress*, 21(5):1558–1561, 2005.

[45] A. Wagner and D. A. Fell. The small world inside large metabolic networks. *Proceedings of the Royal Society B: Biological Sciences*, 268(1478):1803–10, 2001.

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

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

[48] J. M. Weber, J. O. Leung, G. T. Maine, R. H. Potenz, T. J. Paulus, and J. P. DeWitt. Organization of a cluster of erythromycin genes in Saccharopolyspora erythraea. *Journal of Bacteriology*, 172(5):2372–83, 1990.

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

[50] S.-H. Yoon, J.-E. Kim, S.-H. Lee, H.-M. Park, M.-S. Choi, J.-Y. Kim, S.-H. Lee, Y.-C. Shin, J. D. Keasling, and S.-W. Kim. 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):131–9, 2007.

[51] Y. Zhang, M. Morar, and S. E. Ealick. Structural biology of the purine biosynthetic pathway. *Cellular and Molecular Life Sciences*, 65(23):3699–724, 2008.

[52] R. Zrenner, M. Stitt, U. Sonnewald, and R. Boldt. Pyrimidine and purine biosynthesis and degradation in plants. *Annual Review of Plant Biology*, 57:805–36, 2006.