Appeared in 15th Annual International Conference on Research in Computational Molecular Biology (RECOMB), Volume 6577/2011, p.70-84 (2011)
DOI:10.1007/978-3-642-20036-6_9

Identifying Branched Metabolic Pathways by Merging Linear Metabolic Pathways

Allison P. Heath[1], George N. Bennett[2], and 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
{aheath,gbennett,kavraki}@rice.edu

Abstract. This paper presents a graph-based algorithm for identifying complex metabolic pathways in multi-genome scale metabolic data. These complex pathways are called *branched pathways* because they can arrive at a target compound through combinations of pathways that split compounds into smaller ones, work in parallel with many compounds, and join compounds into larger ones. While most previous work has focused on identifying linear metabolic pathways, branched metabolic pathways predominate in metabolic networks. Automatic identification of branched pathways has a number of important applications in areas that require deeper understanding of metabolism, such as metabolic engineering and drug target identification. Our algorithm utilizes explicit atom tracking to identify linear metabolic pathways and then merges them together into branched metabolic pathways. We provide results on two well-characterized metabolic pathways that demonstrate that this new merging approach can efficiently find biologically relevant branched metabolic pathways with complex structures.

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) [17] and MetaCyc [8]. Gaining understanding from these vast quantities 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, potentially spanning multiple species, that could not have been identified by manual means. Importantly, the ability to find metabolic pathways in multi-genome scale data has applications in fields such as metabolic engineering, which focuses on discovering and implementing new metabolic schemes.

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

V. Bafna and S.C. Sahinalp (Eds.): RECOMB 2011, LNBI 6577, pp. 70–84, 2011.

the start compound. Previous work in this area has primarily focused on finding linear sequences of reactions between start and target compounds [25]. However, more complex metabolic pathways, termed *branched pathways*, are dominant in metabolic networks and provide a more complete picture of metabolic processes [22,24]. 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. The identification of branched pathways enables the analysis of metabolic processes with a more comprehensive perspective as compared to the limited picture provided by linear pathways.

The main contribution of this paper is a novel 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 [24,16]. The results demonstrate that the new algorithm is able to efficiently find different network topologies in multi-genome scale data obtained from KEGG. The rest of the paper proceeds as follows: Section 2 describes the relevant previous work in the area of graph-based metabolic pathfinding; Section 3 describes how our new algorithm merges linear pathways to find branched pathways; Section 4 contains the results of our algorithms, validated on two 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 [11]. Graph-based metabolic path finding algorithms complement stoichiometric approaches, as they focus on different aspects of modeling and understanding metabolism [25,12,13]. Stoichiometric models are typically utilized for modeling specific organisms or metabolic systems [15]. Most stoichiometric models are based on the steady-state assumption and therefore require explicit labeling of internal and external compounds [19]. 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 [25]. 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 [31,14,27], adding weights based on the degree of the nodes [9,12] or using measures of structural similarity between compounds [21,26]. However, an approach more closely related to the underlying biochemistry is to use *atom mapping data* [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 [18,20].

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 [12,23] or only returning pathways that conserve at least one atom, typically carbon [2,3,4,5]. 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 [6,24]. 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 [16].

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 [16].

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 [16]. This problem has been shown to be PSPACE-complete and NP-complete when a compound can only be used once in a pathway [6]. Previously unnamed, we will call the linear path finding algorithm from [16] LPAT, for Linear Pathfinding with Atom Tracking. Our results demonstrated that LPAT is able to search across the thousands of reactions and compounds from multiple species contained in KEGG and return realistic metabolic pathways in a few minutes. Based on LPAT, we also developed and validated a novel graph-based algorithm for identifying branched metabolic pathways. Also previously unnamed, we will call the branched path finding algorithm from [16] 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 lin-

ear 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 efficiently find and return branched pathways that correspond to known branched pathways [16]. To the best of our knowledge, the only other algorithm with similar abilities is the recently developed ReTrace algorithm. ReTrace takes a similar approach as BPAT-S, but is based on pathways that only conserve one atom [24]. 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 and extends the topologies of pathways that can be identified automatically.

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 removes the division between seed and branch pathways found in BPAT-S, thus enabling BPAT-M to find pathway topologies that BPAT-S cannot. 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 3.1. Section 3.2 then describes Algorithm 1, 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 chemical constraints. On the right of Figure

1 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 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.

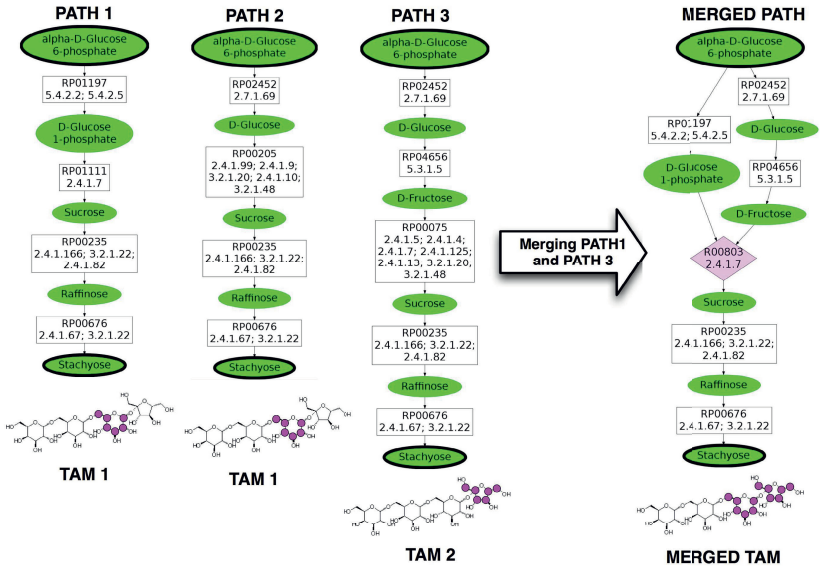

Fig. 1. Three linear pathways from α-D-glucose 6-phosphate to stachyose and their associated carbon TAMs, as indicated by the magenta circles. The two potentially mergeable pairs of paths are PATH 1 with PATH 3 and PATH 2 with PATH 3. The result of merging PATH 1 and PATH 3 is displayed on the right.

The ability to use the TAMs to quickly determine if pathways are not mergeable motivates 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 1, 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 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 mergability, 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 is identified. In Figure 1, 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 common reaction that can be used to merge the two pathways. If there is no common reaction, then the pathways are not mergable. In the case of PATH 1 and PATH 3 in Figure 1, both RP01111 and RP00075 are found in the reaction R00803 (EC Number 2.4.1.7) in KEGG. The right side of Figure 1 depicts PATH 1 and PATH 3 merged by R00803. This information about the mergability of PATH 1 and PATH 3 in Figure 1 would then be stored as $M[PATH1, PATH3] = (R00803, 2)$. In the construction of M, only the mapping nodes that differ closest to the target compound are considered as potential merge points 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 interact closer to the start compound, but this is currently not considered by the algorithm because it does not impact the TAM.

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 1 along with n number of branched pathways to return and a fixed beam width w, which can be used to bound the search. Algorithm 1 then returns the final result of BPAT-M, the top n branched pathways it finds 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 by using biochemical constraints, 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

Algorithm 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

```
1: P ← {}
2: for each c in C do
3:    if P contains more than n pathways and the nth pathway conserves more atoms
      than the size of c then
4:       Truncate P to n pathways
5:       Break
6:    T ← {} //for storing the IBPs, sorted by the same criteria as P
7:    for each pair of linear pathways (p_i, p_j) in (Q[c[1]] × Q[c[2]]) do
8:       if M(p_i, p_j) exists then
9:          Add IBP containing p_i, p_j, M(p_i, p_j) to T
10:   for k = 3 to size of c do
11:      N ← T
12:      T ← {}
13:      for each IBP P in N do
14:         for each linear pathway p_q in Q[c[k]] do
15:            for each linear pathway p_l in P do
16:               if M(p_q, p_l) exists and is a valid merge point in P then
17:                  Add new IBP containing P merged with p_q and M(p_q, p_l) to T
18:      if T contains more than w pathways then
19:         Truncate T to w pathways
20:   Add all pathways in T to P
21: Return P
```

finding the optimal combination. However, the results demonstrate that the search performs well in practice.

Algorithm 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 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_l containing a and p_q containing b, there are two possibilities for a third pathway p_3, that is potentially mergable 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, multiple pathways being merged through the same reaction are handled in a general way and only limited by the substrates used in the reaction.

In Algorithm 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 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 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. At this point, it's recommended to perform a larger search by setting the parameter values high to utilize the computational and storage resources available. Our results demonstrate that even with the heuristic limits, BPAT-M performs well in practice.

4 Results

We present two representative, biologically interesting, test cases for branched metabolic pathways. Both begin from α-D-Glucose 6-Phosphate (G6P), a common form of intracellular glucose. The target compound of the first pathway is lycopene and the target compound of the second pathway is cephalosporin C.

4.1 Experimental Setup

All KEGG data used in the following experiments was downloaded on February 10, 2010. After processing the KEGG RPAIR to obtain a universal index for each atom in each compound, G_{am} was constructed using 11,892 RPAIR entries involving 6,002 compounds and corresponding to 7,510 reactions from more than 1,200 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

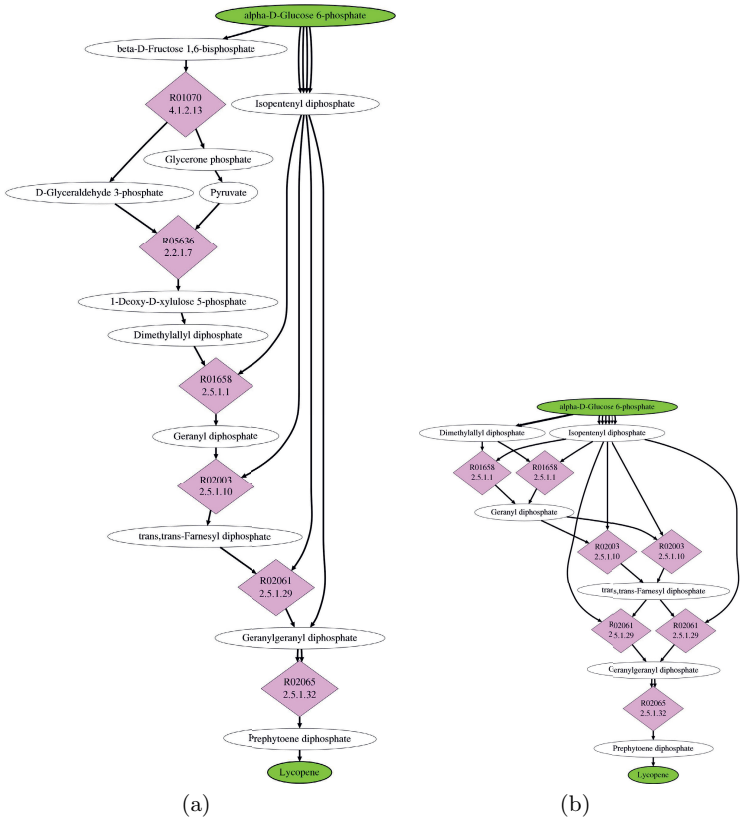

Fig. 2. Top ranked branched metabolic pathways for G6P to lycopene, with the linear pathways conserving 2 carbons, as found by (a) BPAT-S and in (b) BPAT-M. In the interest of space, in (a) 34 mapping nodes and 28 compound nodes and (b) 14 mapping nodes and 11 compound nodes along the linear paths between merge points are hidden from view. The full figures can be viewed in online supplementary material (URL at the end of the paper).

pathway maps, distributed in the 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 6,386 RPAIR entries being considered irreversible.

The implementation was done in Java using the Chemical Development Kit [29] 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. The parameters for BPAT-S and BPAT-M were generally chosen to perform the most exhaustive search, given the resources available. In the branched pathway figures the ellipses are compounds, with the start and target compounds highlighted in green, the pink diamonds contain the KEGG ID and EC numbers for the reactions that occur at the branching points of the pathway. Each edge corresponds to one molecule of each compound. In the interest of space and clarity, the intermediate mapping and compound nodes of linear pathways composing the branched pathway have been removed from the figures, leaving only the reactions that occur at the branching points and their immediate input and output compounds, as well as the start and target compounds. The figures depicting the full branched pathways can be found in the online supplementary material (URL at the end of the paper).

4.2 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 [1,32]. The known biosynthesis pathway of lycopene is relatively well understood and has 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 [28]. BPAT-S and BPAT-M were given as the start compound G6P, the target compound lycopene and three as the number of carbons to conserve. For BPAT-M, k for LPAT was set to 1,000,000, resulting in

36,405 linear pathways without cycles, which had 16 mutually exclusive target atom markings and generated 6,301 combinations. The number of pathways in each cluster was limited to 2,500 and w was set to 500, due to memory limitations. For BPAT-S, due to run time limitations, k for LPAT was set to 500,000, resulting in 22,064 linear pathways without cycles. In both cases, LPAT required about one minute to find the linear pathways.

The top ranked results from BPAT-S and BPAT-M are depicted in Figures 2(a) and 2(b), respectively. BPAT-M performs better than BPAT-S, in that it can find find the known "woven" topology starting with IPP and DMAPP. BPAT-S cannot identify the known topology because it only allows branches off of an initial seed pathway. Additionally, BPAT-M completed in 74.1 minutes, while BPAT-S required 463.2 minutes. Due to space constraints, the intermediates to IPP and DMAPP are not displayed in Figure 2. The full figures found in the online supplementary material (URL at the end of the paper) show that the BPAT-M result correctly finds the lycopene pathway that utilizes the MEP pathway to synthesize IPP and DMAPP; the BPAT-S result utilizes both the MEP/DOXP pathway for DMAPP and MVA pathway for IPP thus revealing the variety available. The MEP/DOXP and MVA pathways demonstrate how search tools for metabolic pathways can illuminate different alternative pathways that may be found in different organisms, which can have applications in areas such as metabolic engineering. At the same time, the ability to find alternative or novel pathways makes it more difficult to judge the performance of different algorithms. Both the BPAT-M and BPAT-S results 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.

4.3 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 [10]. However, it is an important precursor for a number of related antibiotics and has been a target for increased production using metabolic engineering approaches [30]. 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 [30]. BPAT-M and BPAT-S 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. For both BPAT-S and BPAT-M, k for LPAT was set to 1,000,000, resulting in 31,280 linear pathways without cycles found in less than one minute. For BPAT-M the number of pathways in each cluster was limited to 5,000 and w was set to 1,000. BPAT-M found 5 mutually exclusive target atom markings resulting in 18 combinations. BPAT-M took 1.9 minutes to complete the search and BPAT-S took 366.4 minutes.

The top ranked pathways for BPAT-S and BPAT-M are depicted in Figures 3(a) and 3(b), respectively. The simplified figures demonstrate that both BPAT-M and

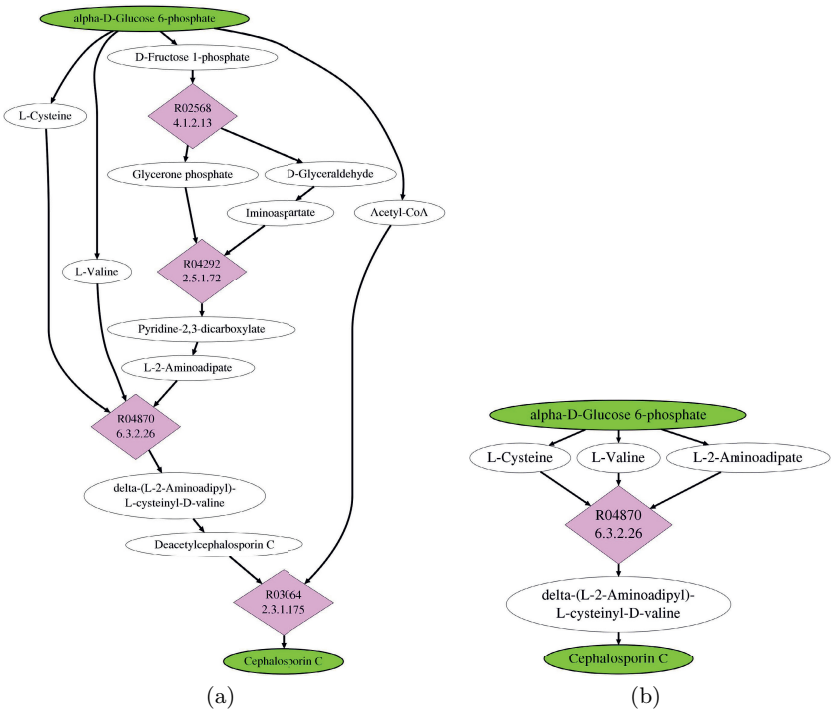

Fig. 3. Top ranked branched metabolic pathways for G6P to cephalosporin C, with the linear pathways conserving 3 carbons, as found by (a) BPAT-S and (b) BPAT-M. In the interest of space, in (a) 33 mapping nodes and 25 compound nodes and in (b) 27 mapping nodes and 21 compound nodes along the linear paths between merge points are hidden from view. The full figures can be viewed in the online supplementary material (URL at the end of the paper).

BPAT-S find the correct overall metabolic scheme, containing the crucial reaction catalyzed by ACV synthetase, which requires three different substrate compounds and produces ACV as the product [7]. Similar to the lycopene pathway, BPAT-M is able to identify this scheme much quicker than BPAT-S. BPAT-S is able to identify the branch through acetyl-CoA which contributes its acetyl group, containing two carbons, to deacetylcephalosporin C to make cephalosporin C. BPAT-M does not identify the pathway through acetyl-CoA since it is not in the original set of linear pathways that conserved three carbons. The full pathway figures, found in the online supplementary material (URL at the end of the paper), reveal that BPAT-M is able to find shorter and pathways more similar to the known pathways through L-valine and L-2-aminoadipate. While BPAT-S finds a similar branched pathway to L-2-aminoadipate, it returns a long and unlikely pathway to L-valine. The unlikely pathway to L-valine is due the approach

taken by BPAT-S of attaching branches that maximize the number of atoms to a single seed pathway.

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 weakness of the approach taken by BPAT-M. These results reveal that the algorithm's performance may depend on the underlying structure of the pathway they seek to find. 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. 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 "woven" topology that is returned as the top result by BPAT-M, but BPAT-S is unable to find. In both cases, BPAT-M took significantly less time than BPAT-S.

Despite the promising results, it is clear that the approach would benefit from a number of improvements. Future work will allow multiple start and target compounds to be used as input, utilize other sources of metabolic data and examine ways to mine the resulting pathways to help the user understand them. The results also highlight that comparing the resulting branched metabolic pathways is nontrivial, especially if the goal is to find alternative or novel pathways, and further work is required to develop meaningful evaluation methods. Results from ReTrace, as described in [24], are not presented because ReTrace has several parameters that affect the search. We observed that the results and runtime varied widely depending on the parameters given to ReTrace. We found that ReTrace, given the same KEGG data, efficiently finds branched pathways similar to those presented in Section 4, but do not contain as many branches. This may be due to a number of factors and performing a valid and exhaustive comparison will be the subject of future work. While it can be 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 an 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.

Online Supplementary Material. The full pathways of Figure 2 and Figure 3 can be found at: http://www.kavrakilab.org/metapath/recomb-2011-supp.

References

1. Alper, H., Miyaoku, K., Stephanopoulos, G.: Construction of lycopene-overproducing E. coli strains by combining systematic and combinatorial gene knockout targets. Nature Biotechnology 23(5), 612–616 (2005)
2. Arita, M.: In silico atomic tracing by substrate-product relationships in Escherichia coli intermediary metabolism. Genome Research 13(11), 2455–2466 (2003)
3. Arita, M.: 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. Blum, T., Kohlbacher, O.: MetaRoute: fast search for relevant metabolic routes for interactive network navigation and visualization. Bioinformatics 24(18), 2108–2109 (2008)
5. Blum, T., Kohlbacher, O.: Using Atom Mapping Rules for an Improved Detection of Relevant Routes in Weighted Metabolic Networks. Journal of Computational Biology 15(6), 565–576 (2008)
6. Boyer, F., Viari, A.: Ab initio reconstruction of metabolic pathways. Bioinformatics 19(90002), 26ii–34ii (2003)
7. Byford, M.F., Baldwin, J.E., Shiau, C.Y., Schofield, C.J.: The Mechanism of ACV Synthetase.. Chemical Reviews 97(7), 2631–2650 (1997)
8. Caspi, R., Altman, T., Dale, J.M., Dreher, K., Fulcher, C.A., Gilham, F., Kaipa, P., Karthikeyan, A.S., Kothari, A., Krummenacker, M., Latendresse, M., Mueller, L.A., Paley, S., Popescu, L., Pujar, A., Shearer, A.G., Zhang, P., Karp, P.D.: The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Research 38(Database issue), D473–D479 (2010)
9. Croes, D., Couche, F., Wodak, S.J., van Helden, J.: Inferring meaningful pathways in weighted metabolic networks. Journal of Molecular Biology 356(1), 222–236 (2006)
10. Demain, A.L., Elander, R.P.: The β-lactam antibiotics: past, present, and future. Antonie van Leeuwenhoek 75(1), 5–19 (1999)
11. Deville, Y., Gilbert, D., van Helden, J., Wodak, S.J.: An overview of data models for the analysis of biochemical pathways. Briefings in Bioinformatics 4(3), 246–259 (2003)
12. Faust, K., Croes, D., van Helden, J.: Metabolic pathfinding using RPAIR annotation. Journal of Molecular Biology 388(2), 390–414 (2009)
13. de Figueiredo, L.F., Schuster, S., Kaleta, C., Fell, D.A.: Response to comment on 'Can sugars be produced from fatty acids? A test case for pathway analysis tools. Bioinformatics 25(24), 3330–3331 (2009)
14. Gerlee, P., Lizana, L., Sneppen, K.: Pathway identification by network pruning in the metabolic network of Escherichia coli. Bioinformatics 25(24), 3282–3288 (2009)
15. Gombert, A.K., Nielsen, J.: Mathematical modelling of metabolism. Current Opinion in Biotechnology 11(2), 180–186 (2000)
16. Heath, A.P., Bennett, G.N., Kavraki, L.E.: Finding Metabolic Pathways Using Atom Tracking. Bioinformatics 26(12), 1548–1555 (2010)
17. Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., Yamanishi, Y.: KEGG for linking genomes to life and the environment. Nucleic Acids Research 36(Database issue), D480–D484 (2008)

18. Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K.F., Itoh, M., Kawashima, S., Katayama, T., Araki, M., Hirakawa, M.: From genomics to chemical genomics: new developments in KEGG. Nucleic Acids Research 34(Database issue), D354–D357 (2006)
19. Kauffman, K.J., Prakash, P., Edwards, J.S.: Advances in flux balance analysis. Current Opinion in Biotechnology 14(5), 491–496 (2003)
20. Kotera, M., Hattori, M., Oh, M., Yamamoto, R., Komeno, T., Goto, S., Yabuzaki, J., Kanehisa, M.: RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions. Genome Informatics 15, P062 (2004)
21. McShan, D.C., Rao, S., Shah, I.: PathMiner: predicting metabolic pathways by heuristic search. Bioinformatics 19(13), 1692–1698 (2003)
22. Michal, G.: Biochemical Pathways: An Atlas of Biochemistry and Molecular Biology. John Wiley & Sons, Inc., New York (1999)
23. Mithani, A., Preston, G.M., Hein, J.: Rahnuma: hypergraph-based tool for metabolic pathway prediction and network comparison. Bioinformatics 25(14), 1831–1832 (2009)
24. Pitkänen, E., Jouhten, P., Rousu, J.: Inferring branching pathways in genome-scale metabolic networks. BMC Systems Biology 3(1), 103 (2009)
25. Planes, F.J., Beasley, J.E.: A critical examination of stoichiometric and path-finding approaches to metabolic pathways. Briefings in Bioinformatics 9(5), 422–436 (2008)
26. Rahman, S.A., Advani, P., Schunk, R., Schrader, R., Schomburg, D.: Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). Bioinformatics 21(7), 1189–1193 (2005)
27. Ranganathan, S., Maranas, C.D.: Microbial 1-butanol production: Identification of non-native production routes and in silico engineering interventions. Biotechnology Journal 5(7), 716–725 (2010)
28. Sandmann, G.: Carotenoid biosynthesis and biotechnological application. Archives of Biochemistry and Biophysics 385(1), 4–12 (2001)
29. Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R., 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), 2111–2120 (2006)
30. Thykaer, J.: Metabolic engineering of β-lactam production. Metabolic Engineering 5(1), 56–69 (2003)
31. Wagner, A., Fell, D.A.: The small world inside large metabolic networks. Proceedings of the Royal Society B: Biological Sciences 268(1478), 1803–1810 (2001)
32. 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., 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), 131–139 (2007)