Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Chemical Engineering 29 (2005) 499–507
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Heurstic search for metabolic engineering: de novo synthesis of vanillin

Daniel McShan, Imran Shah *

Department of Preventive Medicine and Biometrics, School of Medicine, The University of Colorado, 4200 East Ninth Avenue, Box C245, Denver, CO 80262, USA

Received 12 January 2004
Available online 16 March 2005

Abstract

We present the application of heuristic search to in silico metabolic pathway engineering. In particular, we discuss a new computational approach to elucidate complex pathways and to address the practical challenge of combinatorial complexity in pathway inference. We have implemented this approach in a new computational framework, called PathMiner, which is useful for designing metabolic engineering strategies. In this paper, we describe our approach to analyze pathways for the de novo synthesis of vanillin, as well as a transgenic strategy to implement these in a number of hosts. Using PathMiner we are able to automatically elucidate a 19-step pathway for de novo vanillin synthesis from D-glucose, which is in close agreement with the routes reported in the literature. This paper represents a novel integration of artificial intelligence and biochemistry for computational metabolic engineering. As high-throughput biology generates increasing amounts of genomic and metabolic data, automated in silico approaches will become increasingly useful for making biologically useful predictions.

Keywords: Heurstic search; Vanillin; PathMiner

1. Background

Computational approaches to metabolic engineering generally involve quantitative calculations of kinetics, fluxes, control coefficients to optimize a specific network or pathway. Metabolic engineering emphasizes metabolic pathway integration (Stephanopoulos, 1999), but where do these pathways come from? In particular, designing a novel metabolic pathway using heterologous enzymes raises many questions about the choice of enzymes and host organisms. The state of the art in gene addition and knockout is such that it is now possible to engineer almost any pathway in almost any host. Indeed, identifying the metabolic engineering strategy is usually far more complex than implementing it. Selecting a strategy requires not only expertise in chemistry, biocatalysis and molecular biology, but also knowledge of what genes and enzymes are available. With the increasing amounts of genomic data, this is a challenging proposition without the use of intelligent computational aids.

Our goal is to produce a metabolic engineering strategy using only the desired chemical product, the desired carbon source and some qualitative constraints on the choice of the host organism. While the growing databases of metabolic annotations provide a catalog of genes and enzymes, piecing them together into desired pathways is a complex task. Our computational approach, which is based on heuristic search, can sift through the vast amount of information on enzymes and biotransformations to generate combinations of the necessary genes that produce biocatalytic pathways from an input to a desired product. To achieve this, we have developed a heuristic for measuring pathway cost that enables the efficient identification of metabolic routes.

In this paper, we elucidate a pathway for de novo vanillin synthesis. Vanillin, or 4-hydroxy-3-methoxybenzaldehyde, is a very important flavor and aroma molecule. More than 12,000 tons of vanillin are produced each year, but less than 1% of this is natural vanillin from the beans of the *Vanilla*

* Corresponding author. Fax: +1 303 315 7222.
E-mail address: imran.shah@uchsc.edu (I. Shah).

0098-1354/$ – see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.compchemeng.2004.08.038

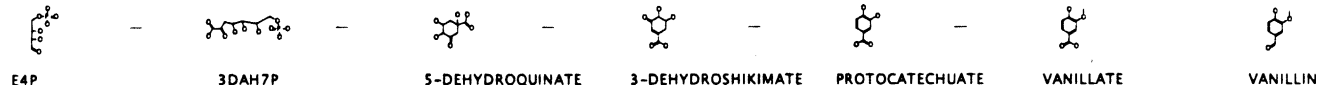


Fig. 1. Vanillin synthetic pathway of Berry (Berry, 1996; Priefert et al., 2001). E4P is produced as a product of pentose phosphate pathway, and is transformed via the shikimic acid pathway into 3-dehydroshikimic acid, which catalyzed by the added 3-dehydroshikimate dehydratase into protocatechuic acid. Protocatechuic acid is then transformed into vanillic acid by the transgenic addition of catechol-*o*-methyltransferase.

orchid (Walton, Mayer, & Narbad, 2003). The natural production of vanillin is complicated by the fact that *Vanilla planifolia* flowers asynchronously, requiring hand pollination of the flowers. Additionally, the vanillin in the green vanilla beans are present exclusively in conjugated form as the β-D-glucoside, which exhibits no vanilla flavor, requiring extensive curing. Indeed, the worldwide increasing demand for natural vanilla flavor far outweighs the available supply with vanilla pods alone (Priefert, Rabenhorst, & Steinbuchel, 2001). As a result, the vast majority of vanillin is synthesized relatively cheaply via chemical processes. Thus, it is important to study alternative approaches for the production of vanillin.

Early work in *V. planifolia* by Zenk (1965) suggested that ferulic acid might be beta-oxidized to form vanilloyl-coa and then vanillin. More recently, though, the literature (Funk & Brodelius, 1994) indicates that the natural biosynthesis of vanillin appears to be a product of phenylalanine via the phenylpropanoid pathway, then to caffeic acid (3,4-dihydroxy-4-methoxycinnamic acid), isoferulic acid (3-hydroxy-4-methoxycinnamic acid), dimethoxycinnamic acid, dimethoxybenzoic acid, and finally to vanillic acid where it is glucosylated, then reduced to vanillin. Vanilla biosynthesis and degradation have also been observed in several microorganisms! (Achterholt, Priefert, & Steinbuchel, 2000; Gasson et al., 1998; Mayer et al., 2001; Narbad & Gasson, 1998), and the relevant enzymes have been isolated from them. Since the genome for *V. planifolia* has not been sequenced, these microbial enzymes can be useful for analyzing potential de novo vanillin biosynthetic pathways.

The biotechnological production of vanillin (for a detailed review please see (Priefert et al., 2001; Walton, Nar bad, Faulds, & Williamson, 2000) has focused on the biotransformation of complex feedstocks like phenolic stilbenes (Dawidar, Ezmiriy, Abdel-Mogib, el Dessouki, & Angawi, 2000; Gill, Bajaj, Chang, Nichols, & McLaughlin, 1987; Murcia & Martinez-Tome, 2001) and eugenol (Brandt, Thewes, Overhage, Priefert, & Steinbuchel, 2001; Chen, Ohmiya, Shimizu, & Kawakami, 1988). On the other hand, the relevant literature for the de novo biosynthesis of vanillin is scarce. We were only able to identify a 1996 review by Berry as elucidated by Priefert et al. (2001) for genetically engineering *Escherichia coli* to produce aromatic compounds. The pathway discussed by Berry is illustrated in Fig. 1 and the transgenic strategy is outlined in Table 1. In this paper, we describe our algorithm to elucidate the de novo synthesis of vanillin using publicly available information on biotransformations, enzymes and genes using PathMiner.

Table 1
Transgenic strategy for implementing pathway by Berry (Berry, 1996; Priefert et al., 2001)

Organism	Genes	Enzyme
E. coli KL7	*+aroE*	Shikimate dehydrogenase
	+aroZ	3-Dehydroshikimate dehydratase
	$+P_{tac}COMT$	Catechol-*o*-methyltransferase
	$aroF^{FBR}$	DAHP synthetase
	sera	Plasmid stabilizer
	aroB	3-Dehydroquinate synthase

This is the only de novo vanillin biosynthesis in the literature.

2. Methods

The foundation of our in silico metabolic engineering approach is a state-space search algorithm. Given an input metabolite, this algorithm finds the optimal biochemical pathway to the target compound. We have described the general algorithm in some depth elsewhere (McShan, Rao, & Shah, 2003). Here, we will focus on the aspects of the algorithm, which are useful for metabolic engineering.

2.1. The search algorithm

A* ("A-star") search is a well-known algorithm for finding an optimal path in a graph. It works by exploring states (or nodes in a graph) in a breadth first manner according to a cost function: $f(n) = g(n) + h(n)$. In the cost function, n is a state (also called a node), $f(n)$ is the estimated cost of the cheapest solution through n, $g(n)$ is the path cost from the start state to state n. The heuristic, $h(n)$, is the estimated cost of the cheapest path from n to the goal. In addition to being both optimal and complete, A* is also optimally efficient. This means that no other optimal algorithm is guaranteed to expand fewer nodes than A* (Dechter & Pearl, 1985). Our implementation of A* is similar to the one given by Russell and Norvig (1995) and our algorithm is described in Algorithm 1. We specialize A* for pathway search by abstracting metabolism as a graph in which the states are compounds and the arcs are biotransformations. Four main constructs enable us to search for metabolic pathways. First, we define biochemical "successors" of a molecule. Second, we develop a heuristic, or an oracle, for the "distance" between any molecule and the final product.

Third, we chose an evaluation function for evaluating the optimality of alternative pathways. Fourth, we define the condition that terminates the search on reaching the final product. Each of these concepts and their implementation are described in greater detail in the following sections.

```
input : x^0, x^L, Ω
output: P^{0,L}
begin
    N ← make-node(x^0)
    while N ≠ () do
        n ← pop(N)
        P^{0,L} ← node-solution(n)
        if goal-test(n) then
            return P^{0,L}
        for (reaction, compound) ∈ successors(n) do
            e ← edge-cost(n reaction compound)
            h ← h-cost(compound)
            g ← g-cost(n) + e
            f ← g + h
            d ← depth(n)
            n' ← make-node(n reaction compound g h f d)
            push(n' N)
        N ← sort(N f-cost)
end
```

Algorithm 1. Best-first search algorithm with heuristic to find pathway, from input compound x^0 to x^L. The search optimizes *f*, where *f* is the total path cost. The heuristic uses an estimate of *f* defined as $f = g + h$ where *g* is the cost of the path so far, and *h* is the estimate to the goal, x^L. Each transformation in the pathway has an edge cost, *e*. At the goal, $f = \sum_{i=0}^{i=L} e_i$. This algorithm will always return the optimal cost path in terms of *f* if *h* always underestimates the actual path-cost from the current node to the goal. The algorithm begins by adding the input node, X^0 to *N*, the queue of nodes to be expanded. The first node on the list is popped off and stored in *n*. The pathway to that node is stored in $P^{0,L}$. If $P^{0,L}$ meets our goal test, then we return the pathway and the algorithm is done. If not, we then generate successors for the current node, returned as pairs of the reaction that performs the biotransformation and the compound to which the current node is transformed. The edge-cost, *e*, of the biotransformation is computed, as is the estimated distance from the successor node to the goal node, *h*. From these values, and the pathway cost so far, we are able to compute the estimated total path length for the continuation of the current pathway, $P^{0,L}$ through the successor node. A new node, n', is created for the successor, and the metrics are stored with the *reaction* and the *compound*. The successor node, n' is then pushed onto the stack of nodes to explore, *N*. Once all the successors are expanded, the list is resorted according to the node's estimated total cost, *f*. Once the list of nodes to expand is exhausted, unless a pathway is found, the algorithm ends, and returns no pathway.

2.2. Compound successors: known and novel biotransformations

The purpose of calculating successors is to generate the biocatalytically possible "next steps", or products, from any given compound. The implementation of this function takes a compound as an input and produces a list of pairs of biotransformations and output compounds. We can compute the successors of a given compound in two ways. First, by using the information about the action of enzymes on known compounds and the products of these reactions from in public databases. Second, by the prediction of novel biotransformations. We use two main sources of metabolic data: KEGG and MetaCyc. In the current work, we used the KEGG database for its breadth of organisms.

A critical issue in pathway search is dealing with the combinatorial explosion due to the large number of potential successors for each compound. If we restrict the successors based on the curated information for the known biotransformations of compounds then there are around ten successors for each compound. Since the known enzyme-catalyzed biochemistry is only a small subset of what exists in nature, there are many more biochemical successors of any given compound. To predict novel products of the action of an enzyme on a given compound we are developing an algorithm to predict completely novel biotransformations based on a chemical graph–theoretical approach. While this approach can identify completely novel pathways, it is computationally intractable due to the order of magnitude increase in the number of successors of each compound. The heuristic search approach we present in this paper is equally powerful for harnessing the computational complexity of de novo pathway prediction.

Another useful supplement for curtailing the computational complexity due to the large number of successors,

Table 2
Filters used in PathMiner to limit the number of successors generated

Filter	Description
KEGG	Only generate transformation from KEGG
KEGG-directional	Only use the transformation directions annotated in KEGG
MeiaCyc	Only generate transformation from MetaCyc (TBD)
Avoid	Specific avoid (compounds, reactions, enzymes, genomes)
Coenzyme	Ignore coenzymes
Trivial	Ignore trivial molecules (H_2O, CO_2, NH_3, etc.)
Inorganic	Ignore successors which do not have carbon
Currency	Ignore "currency" molecules (ATP, ADP, NADPH, etc.)
Organism	Only use enzymes from a particular organism
Annotated	Only use enzymes which are annotated with genes

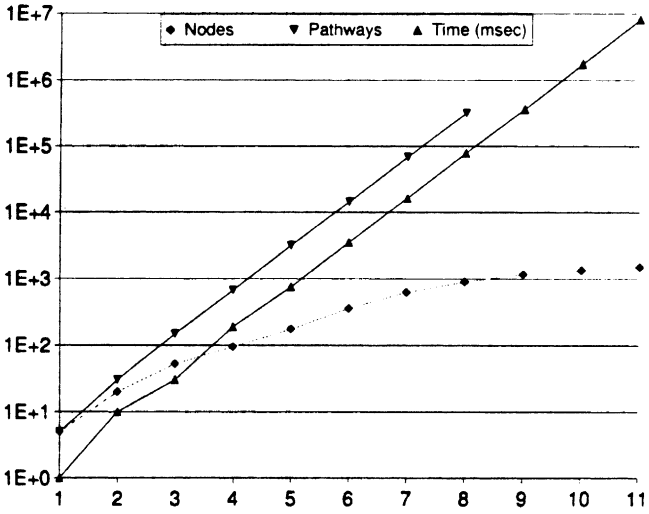
Table 3
Edge costs implemented in PathMiner

Edge cost	Description
Chemical	Cost of adding/removing atoms or bonds from current node, n_{-1}, to n
Step	Cost per step
Unannotated	Cost of using an enzyme not annotated with an organism
Transgenic	Cost of using an enzyme organism not already in pathway
Kingdom	Cost of using an enzyme from a different kingdom
Xenogenice	Cost of using an enzyme not in Ω
Elemental	Cost for change in a particular element (i.e. carbon)
Successors	Cost for # of successors product node has in Ω (flux loss)
Organic reactant	Cost for # of precursors product node has
Organic product	Cost for other organic products
Specific reactant	Cost for using a particular reactant (i.e. ATP)
Specific product	Cost for producing a particular product

which may generated either from a metabolic database or by de novo prediction, is a set of filters shown in Table 2. Despite these filters, the combinatorial growth of the search space is still exponential. As a result, breadth first search is computationally intractable for pathways longer than around ten steps.

2.3. The heuristic evaluation function

In order to control the combinatorial complexity of pathway search we use a biochemical heuristic evaluation function, h. It can be shown that A* is complete, optimal, and optimally efficient if the heuristic is *admissible.* An admissible heuristic is one in which $h^*(n) - h(n) \geq 0$, in which $h^*(n)$ is the actual cost from the state, n, to the goal n_{goal}. The heuristic is admissible if it never overestimates the pathway cost between two states. The condition for sub-exponential growth is $|h(n) - h^*(n)| \leq O(\log h^*(n))$. This means that the deviation of the heuristic from the actual distance must be minimized for efficient search.



Variable	Regression
Nodes	$= 7.87 \times 1.73^x$
Paths	$= 1.23 \times 4.77^x$
Time	$= 0.31 \times 4.74^x$

Fig. 2. Combinatorial complexity. This figure illustrates the expanded states, the pathways, and the worst-case breadth-first search time as a function of pathway length. While the number of states increases linearly with path length, the number of pathways, and hence the search time, increases exponentially. The logarithmic regressions are shown where the abscissa is the path length.

We have developed the notion of biochemical distance (McShan et al., 2003) as a useful heuristic for metabolic search. This distance metric is based on a mapping of all compound states into a multidimensional feature space, C. The dimensions of C are defined by the atoms and bonds of biomolecules. We then use the Manhattan distance between two molecules in C as the heuristic for pathway search. For any two compounds, we define this distance as $d(c1,c2)$. By optimizing this distance, we derive a pathway that effectively reflects the minimum number of chemical changes from the starting compound to the final compound. This is the chemically most parsimonious pathway. In general, we find that the biochemical distance metric is a very efficient heuristic because it consistently outperforms breadth first searching by two orders of magnitude. It is also important to note that we can use the concept of heuristic search to optimize completely different cost models, provided they use an admissible heuristic.

2.4. Preferring biotransformation: edge-cost

The chemical optimality of a pathway is a useful metric for identifying a feasible metabolic route but other practical considerations can be more relevant to engineer a pathway into a host. For example, PathMiner can allow the preferential selection of alternative enzymes for a transformation. To engineer a pathway in *E. coli*, it may be preferable to find a pathway that requires the least heterologous enzymes. We provide this knowledge to the algorithm via the "edge-cost" function. This is called the edge-cost because it is the cost associated with an edge in the state-space search graph. For

Table 4
Multiple pathway table

	Organism	1	2 BME	3 SCO	4 ECO	5 ATH
	D	19	19	20	20	21
	N	1233	1257	1216	1246	1075
	B*	1.4	1.4	1.3	1.3	1.3
	F	524	7526	6532	7526	4532
Abbr/EC	**Compound/Enzyme**					
ADGLU	**ALPHA-D-GLUCOSE**	o	o	o	o	o
5.3.1.5	*XYLOSE_ISOMERASE*	↓	⇓	⇓		
5.1.3.3	*MUTAROTASE*				⇓	⇓
BDGLU	**BETA-D-GLUCOSE**				o	o
2.7.1.1	*HEXOKINASE*				↓	⇓
BDG6P	**BETA-D-GLUCOSE_6-PHOSPHATE**				o	o
5.3.1.9	*OXOISOMERASE*				⇓	⇓
LEV	**LEVULOSE**	o	o	o		
2.7.1.1	*HEXOKINASE*	↓	↓	↓		
BDF6P	**BETA-D-FRUCTOSE_6-PHOSPHATE**	o	o	o	o	o
2.2.1.1	*TRANSKETOLASE*	↓	⇓	⇓	⇓	⇓
ERY4P	**D-ERYTHROSE_4-PHOSPHATE**	o	o	o	o	o
2.5.1.54	*DS-MN*	↓	⇓	⇓	⇓	⇓
3DAH7P	**3-DEOXY-ARABINO-HEPTULONATE_7-PHOSPHATE**	o	o	o	o	o
4.2.3.4	*CYCLIZING*	↓	⇓	⇓	⇓	⇓
5DHQ	**5-DEHYDROQUINATE**	o	o	o	o	o
4.2.1.10	*DHQASE*					
4.2.1.11	*ENOLASE*	↓	⇓	⇓	⇓	⇓
3DHS	**3-DEHYDROSHIKIMATE**	o	o	o	o	o
4.2.1.10	*DHQASE*	↓				
1.1.1.25	*SHIKIMATE_OXIDOREDUCTASE*		⇓	⇓	⇓	
1.1.99.25	*NAD-P-INDEPENDENT_QUINATE_DEHYDROGENASE*					↓
SHI	**SHIKIMATE**		o	o	o	o
2.7.1.71	*SHIKIMATE_KINASE*		⇓	⇓	⇓	⇓
PRO	**PROTOCATECHUATE**	o				
4.1.1.63	*PROTOCATECHUATE_DECARBOXYLASE*	↓				
SHI5P	**SHIKIMATE_5-PHOSPHATE**		o	o	o	o
2.5.1.19	*EPSP_SYNTHASE*		⇓	⇓	⇓	⇓
CAT	**CATECHOL**	o				
1.3.1.55	*2-HYDRO-1-2-DIHYDROXYBENZOATE_DEHYDROGENASE*	↓				
O51C3P	**O5-1-CARBOXYVINYL-3-PHOSPHOSHIKIMATE**		o	o	o	o
4.2.3.5	*CHORISMATE_SYNTHASE*		⇓	⇓	⇓	⇓
CIS12DIHC35D1C	**CIS-1-2-DIHYDROXYCYCLOHEXA-3-5-DIENE-1-CARBOXYLATE**	o				
1.14.12.10	*BENZOIC_HYDROXYLASE*	↓				
CHO	**CHORISMATE**		o	o	o	o
5.4.99.5	*CHORISMATE_MUTASE*		⇓	⇓	⇓	⇓
BEN	**BENZOATE**	o				
1.14.13.12	*BENZOIC_4-HYDROXYLASE*	↓				
PRE	**PREPHENATE**		o	o	o	o
2.6.1.57	*ARAT*		↓		⇓	↓
4.2.1.51	*PREPHENATE_DEHYDRATASE*			⇓		⇓
4HBEN	**4-HYDROXYBENZOATE**	o				
4.1.1.61	*P-HYDROXYBENZOATE_DECARBOXYLASE*	↓				
PHE	**PHENYLPYRUVATE**			o		o
2.6.1.57	*ARAT*			↓		↓
PHE	**PHENOL**	o				
4.1.99.2	*BETA-TYROSINASE*	↓				
LPHE	**L-PHENYLALANINE**			o		o
1.14.16.1	*PHENYLALANINASE*			↓		
4.3.1.5	*PAL*					⇓
PRE	**PRETYROSINE**		o		o	
1.3.1.43	*AROGENIC_DEHYDROGENASE*		⇓		↓	
TRACIN	**TRANS-CINNAMATE**					o
1.14.13.11	*CA4H*					⇓
LTYR	**L-TYROSINE**	o	o	o	o	
4.3.1.5	*PAL*	↓	↓		↓	
1.14.18.1	*CRESOLASE*			⇓	↓	
4COU	**4-COUMARATE**	o	o		o	o
1.14.18.1	*CRESOLASE*	↓	↓		↓	↓
LDOP	**L-DOPA**			o		
4.3.1.11	*DIHYDROXYPHENYLALANINE_AMMONIA-LYASE*			↓		
TRACAF	**TRANS-CAFFEATE**	o	o	o	o	o
2.1.1.68	*CAFFEATE_METHYLTRANSFERASE*	↓	↓	↓	↓	
6.2.1.12	*4CL*					⇓
CAFCOA	**CAFFEOYL-COA**					o
2.1.1.104	*CAFFEOYL-COA_O-METHYLTRANSFERASE*					⇓
FER	**FERULATE**	o	o	o	o	
6.2.1.34	*TRANS-FERULOYL-COA_SYNTHASE*	↓	↓	↓	↓	
FERCOA	**FERULOYL-COA**	o	o	o	o	o
4.2.1.101	*TRANS-FERULOYL-COA_HYDRATASE*	↓	↓	↓	↓	↓
3H34H3MCOA	**3-HYDROXY-3-4-HYDROXY-3-METHOXYPHENYLPROPIONYL-COA**	o	o	o	o	o
4.1.2.41	*VANILLIN_SYNTHASE*	↓	↓	↓	↓	↓
VAN	**VANILLIN**	o	o	o	o	o

Table of compounds and enzymes for Fig. 3. Circles indicate metabolites, arrows indicate enzymes, ⇓ arrows indicate the enzyme is in the genome, ↓ arrows indicate that the enzyme is not in the genome. Pathway 1 is in the "universal network" using enzymes from all organisms in KEGG. Pathway 2 uses *B. melitensis* as a host, pathway 3 uses *S. coelicolor*, pathway 4 uses *E. coli*, and pathway 5 uses *A. thaliana*.

compounds c_1 and c_2, the edge-cost is denoted by $e(c_1, c_2)$. To maintain the admissibility of the heuristic, the actual cost of an edge cannot be negative. That is, a pathway cost cannot be "rewarded"; but it can be "penalized". In above example, to prefer a pathway with enzymes from *E. coli* rather than other hosts, the use of a heterologous enzyme in a transformation is penalized with a positive contribution to h. That is, we use the chemical distance as the baseline cost, and add positive cost penalties to it. This guarantees that the actual cost is always less than the heuristic and satisfies the requirement of admissibility. Consequently, a pathway that uses only *E. coli* enzymes will have a lower h than one requiring heterologous genes, rendering it more optimal.

Since PathMiner has the complete genomes of over 100 organisms, one of its strengths is discovering transgenic pathways. Yet there are many practical issues in introducing and expressing multiple genes from different organism into a single host. If a specific host is desired, then PathMiner can associate a cost with the addition of enzymes that are not endogenous to the host. This is extremely useful when a metabolic engineering project is targeted to a specific host. As discussed below, if a host has not yet been selected, PathMiner can make some suggestions about an appropriate choice.

While the chemical distance is an extremely useful measure of cost, our framework also allows the utilizations of other cost models. In order to retain the guarantee of com-

Table 5
Biochemically optimal pathway for vanillin synthesis

EC	Enzyme	Compound	f	g	h	$\|\Omega\|$
		α-D-Glucose			22	
5.3.1.5	Xylose isomerase	Levulose	22	0	22	26
2.7.1.1	Hexokinase	β-D-Fructose-6-phosphate	42	10	32	69
2.2.1.1	Transketolase	D-Erythrose 4-phosphate	52	29	23	111
2.5.1.5.4	3-Deoxy-7-phosphoheptulonate synthase	3-Dexoy-arabino-heptulonate 7-phosphate	80	49	31	98
4.2.3.4	3-Dehydroquinate synthase	5-Dehydroquinate	84	66	18	97
4.2.1.1.1	Phosphopyruvate hydratase	3-Dehydroshikimate	86	74	12	222
4.2.1.1.0	3-Dehydroquinate dehydratase	Protocatechuate	94	84	10	104
4.1.1.6.3	Protocatechuate decarboxylase	Catechol	104	92	12	0
1.3.1.5.5	*cis*-1,2-Dihydrocyclohexa-3, 5-diene 1-COOH dehydrogenase	*cis*-1,2-Dihydroxycyclo-3, 5-diene-1-COOH	112	104	8	0
1.1.4.1.2.1.0	Benzoate 1,3 dioxygenase	Benzoate	122	114	8	1
1.1.4.1.3.1.2	Benzoate 4 monooxygenase	4-Hydrozybanzoate	126	118	8	0
4.1.1.1.6.1	4-Hydroxybenzoate decarboxylase	Phenol	138	126	12	1
4.1.9.9.2	Tyrosine phenol lyase	1-Tyrosine	160	148	12	3
4.3.1.5	Phenylalanine ammonia lyase	4-Coumarate	164	158	6	2
1.1.4.1.8.1	Monophenol monooxygenase	*trans*-Caffeate	170	162	8	5
2.1.1.6.8	Caffeate-*o*-methyltransferase	Ferulate	180	170	10	0
6.2.1.3.4	*trans*-Feruloyl-CoA synthase	Feruloyl-coA	510	335	175	0
4.2.1.1.0.1	*trans*-Feruloyl-CoA hydratase	3-OH-3,4-OH-3-methoxyphenylpropionyl-CoA	524	343	181	0
4.1.2.4.1	Vanillin synthase	Vanillin	524	524	0	0

The algorithm is attempting to minimize f, the estimated pathway length. $f = g + h$, where g is the exact path length so far, and h is the heuristic estimate of the distance to the goal. Both g and h are computed using distance in chemical space, C. The number of organisms which code for each enzyme is represented by $|\Omega|$, and shown in the last column. Finally, the net equation is provided, including all side compounds.

pleteness and optimality for A*, however, one must maintain the admissibility of h. That is, h must always underestimate the actual cost. Since it can be difficult to find heuristics for other costs, we use the heuristic above, $h(n) = d(n,n_{\text{end}})$, and let $e(n1,n2) = d(n1,n2) + \sum e_i(n1,n2)$ where $e_i(n1,n2)$ are the other metrics we wish to optimize. We have successfully use this approach to minimize the use of heterologous enzymes in predicted pathways (Table 3).

2.5. Terminating the search: goal-test

The pathway search terminates when the final product has been synthesized. This is accomplished by the goal-test function, which evaluates the pathway to determine if a solution has been found. The simplest goal-test is a comparison of the last compound in the pathway with the desired product. We can also specify additional pathway selection criteria like path length, maximum pathway cost, maximum number of heterologous enzyme, and presence or absence of a specific intermediate.

3. Results and discussion

Using heuristic search PathMiner was able to computationally derive a metabolic strategy for the de novo synthesis of vanillin from d-glucose in *E. coli*. We used the KEGG annotation of transformations (essentially searching through the pathway maps). This produces a 19-step pathway as shown in Table 4. Given the magnitude of the search space, automatically identifying this pathway is a very important result. An alternative pathway search engine from KEGG, called PathComp (Ogata, Goto, Fujibuchi, & Kanehisa, 1998), is unable to return a solution for this search. This is probably because it implements a breadth first search algorithm, which suffers from the combinatorial explosion mentioned above. Based on the exponential nature of the time complexity encountered in pathway search observed in Fig. 2($t_{\text{ms}} = 0.31 \times 4.74^{19} = 1 \times 10^{13}$ ms = 67.5 years), it can take up to 70 years to explore all of the pathways with a length of 19 steps using a standard breadth-first search method. PathMiner takes about 18 s to find the optimal solution, which is a speedup of 10^8. As knowledge of biotransformations grows, the time complexity of predicting biochemical routes increases exponentially making intelligent search a necessary tool for in silico pathway engineering (Table 5).

With this 19-step predicted pathway for de novo vanillin synthesis in hand, PathMiner can also aid in designing a metabolic engineering strategy through a number of tools. First, it can identify a suitable host for engineering this pathway. It accomplishes this by sorting the list of all known organisms by the number of genes they have for encoding the enzymes in the desired pathway. For vanillin synthesis this turns out to be a tie between *Brucella melitensis* and *Streptomyces coelicolor* with eight of the genes present in each organism. Though these organisms are not common hosts, they can be worth considering. *E. coli* 0157 has seven of the

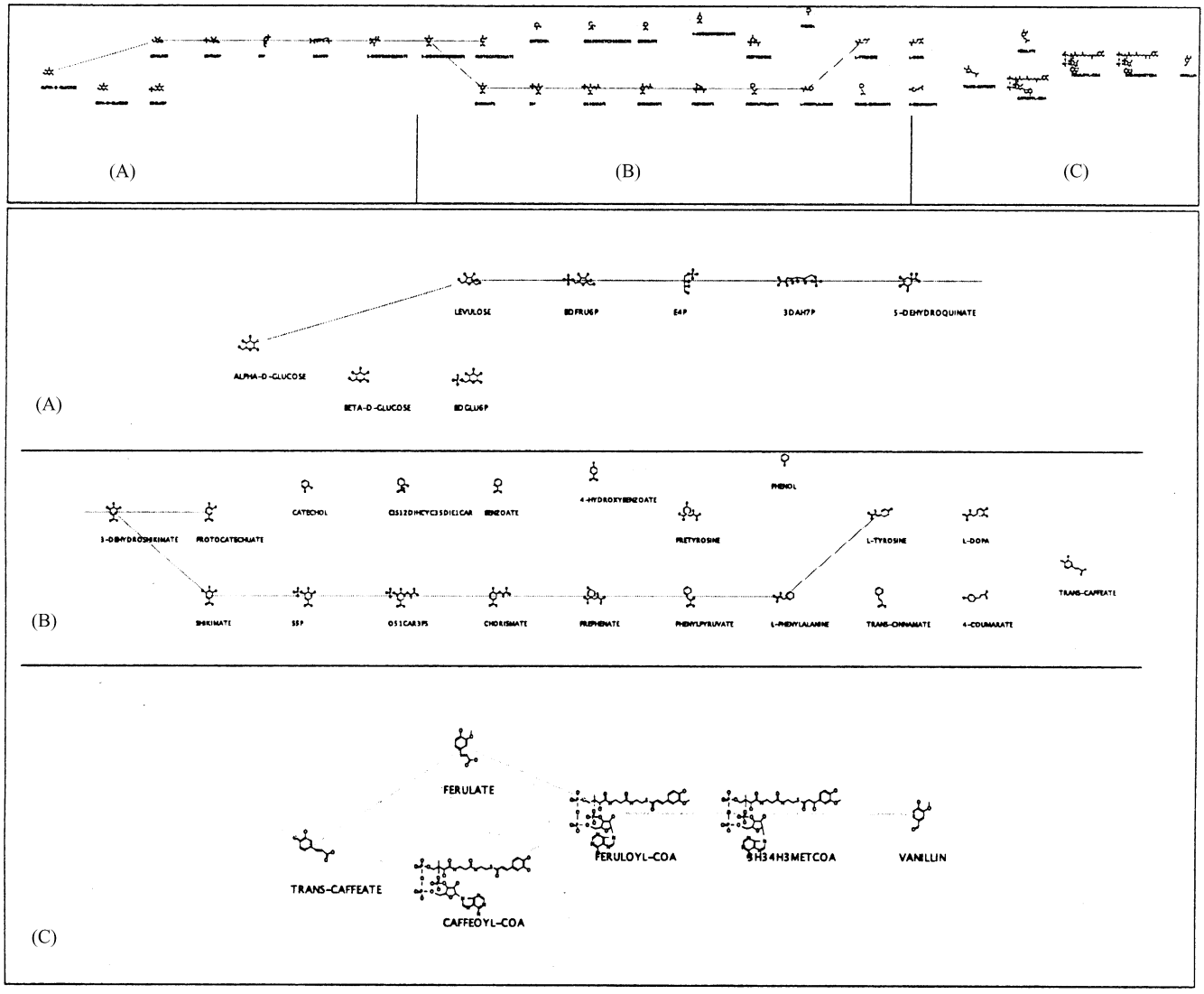


Fig. 3. Multiple pathways from α-D-glucose to vanillin. Universally biochemically optimal pathway. Pathways from *B. melitensis*, *S. coelicolor*, *E. coli* and *A. thaliana* are overlapped. Precise pathways and enzymes are given in Table 4. The pathways are broken into three segments and enlarged for readability.

necessary genes, and the transgenic strategy for this pathway is shown in Table 6. It would take seven heterologous genes to transform L-tyrosine to vanillin via ferulate. This pathway has been discussed at great length in the literature (Priefert et al., 2001; Walton et al., 2003) and *V. planifolia* uses a variant of this pathway. However, the genes encoding the enzymes for the biotransformation of ferulate to vanillin are not annotated in KEGG. This is not surprising since KEGG only contains the annotations for gene from complete genomes. One of the advantages of KEGG and MetaCyc is that they maintain literature references indicating which the organisms that have been observed to code for these enzymes.

The optimal pathways for several other host options are illustrated in Fig. 3 and Table 4.

One of the features of PathMiner is its ability to interactively search for pathways. One such features is the ability to control the directionality of biotransformations. When we relax the constraint on the directionality of transformations from KEGG we find a biochemically optimal pathway from α-D-glucose to vanillin through protocatechuic acid. This is a much shorter route, and is a compelling result because it is the only known example of de novo synthesis of vanillin from glucose in *V. planifolia* (Priefert et al., 2001). PathMiner suggests that *B. japonicum* has the distinction of "best host"

Fig. 4. Lignin degradation to vanillin. The KEGG database does not have the chemical structure for lignin.

Table 6
Transgenic strategy for de novo vanillin synthesis in *E. coli*

E. coli+		
Organism	Genes	Enzyme
M. loti MAFF303099 *S. meliloti 1021*	*+tyrC*	Cyclyhexadienyl dehydrogenase
. . .		
A. thaliana	*+AT2G37040*	Phenylalanine ammonia lyase
H. sapiens, *M. musculus*, *D. melanogaster*, etc.	*+tyr*	Monophenol monooxygenase
*Soybean, spinach	?	Caffeate o methyltransferase
**P. fluorescens* AN103 (Narbad & Gasson, 1998) *S. viridiosporus* (Pometto & Crawford, 1983)	?	*trans*-Feruloyl CoA synthase
**P. fluorescens* AN103 (Narbad & Gasson, 1998) *S. viridiosporus* (Pometto & Crawford, 1983)	?	*trans*-Feruloyl CoA hydratase
**P. fluorescens* AN103 (Narbad & Gasson, 1998) *S. viridiosporus* (Pometto & Crawford, 1983)	?	Vanillin synthase

The last four steps are unannotated with a gene in KEGG. However, the enzyme entries do have a literature reference.

for this pathway. This pathway is nearly identical to the only de novo strategy we were able to identify in the literature as discussed above and illustrated in Fig. 1.

While both pathways can be validated by literature, why do the two transformation methods give very different results? If we use the KEGG annotation of transformations, we find a 19-step pathway through ferolic acid. This is because reactions in KEGG do not include the transformation of protocatechuate to vanillin. The reaction is annotated as bidirectional "⇔", but only the reverse transformation from vanillate to protocatechuate is annotated in KEGG. If we relax the directionality constraint, we are then able to transform protocatechuate ⇒ vanillin via vanillate demethylase. However, the *E. coli* engineering discussed in literature used catechol-*o*-methyltransferase (EC 2.1.1.6) and not the vanillate demethylase (EC 1.2.3.12) described in our pathway solution.

We rationalize relaxing the directionality because there are large amounts of missing annotation in the metabolic databases, and most biotransformations are reversible (though this may involve different enzymes). This turns out to be a valid assumption in this case as vanillate demethylase catalyzes demethylation or monohydroxylation with a variety of different aromatic substrates (Morawski, Segura, & Orgnston, 2000). Ultimately, this is a limitation of the annotation: PathMiner would use the catechol-*o*-methyltransferase if it was annotated. KEGG does have catechol-*o*-demethylase genes in the human, the mouse and the rat genomes but it is not functionally characterized as an enzyme responsible for the biotransformation of protocatechuic acid to vanillic acid.

We are not limited to just a de novo fermentation approach. There is, for example, a reported pathway in white rot fungi for the degradation of the guaiacyl-glycerol-β-coniferyl ether unit of lignin o vanillin (Ishikawa, Schubert, & Nord, 1963). Lignin is one of the most abundant natural sources of aromatic compounds, and it is an obvious source for vanillin biosynthesis. In fact, there is already a process for the chemical oxidation of lignin (Clark, 1990). In KEGG, lignin only occurs in the context of "Flavonoid, Stilbene and Lignin Biosynthesis" as a terminal product (i.e., all transformations lead to lignin but none use it as a precursor). As a result, PathMiner cannot find a pathway from lignin to vanillin. On the other hand, if we relax the directionality, a quick pathway search from lignin to vanillin finds the pathway from lignin through coniferol, shown in Fig. 4, which is converted to coniferyl aldehyde (ferulaldehyde) that joins the above pathways at ferulate. While the white-rot fungi are not genomically annotated, we can find a pathway based on literature curation.

While literature curation is vital for this process, it is not as useful for developing a specific metabolic engineering strategy. As more organisms are sequenced the corresponding genomic annotation will be increase and we will be able to utilize it in pathway elucidation.

4. Conclusion

We have demonstrated how a heuristic search algorithm can elucidate a transgenic strategy for metabolic engineering of vanillin from D-glucose using the KEGG database. The pathway is 19 enzymatic steps in length, and as such it cannot be solve by other computational approaches. The chemical space heuristic with cost penalties allows us to precisely define the criteria that are important for our solution. As more and more genomes are sequenced, and novel enzymes are annotated, PathMiner will become increasingly useful for the metabolic engineering community.

Acknowledgments

The authors acknowledge Weiming Zhang for the visualization software. This work is sponsored by the National Science Foundation (BES-9911447), the Department of En-

ergy (DE-FG03-01ER63111/M003), and the Office of Naval Research (N00014-00-1-0749).

References

Achterholt, S., Priefert, H., & Steinbuchel, A. (2000). Identification of *Amycolatopsis* sp. strain HR167 genes, involved in the bioconversion of ferulic acid to vanillin. *Appl. Microbiol. Biotechnol.*, *54*(6), 799–807.

Berry, A. (1996). Improving production of aromatic compounds in *Escherichia coli* by metabolic engineering. *Trends Biotechnol.*, *14*(7), 250–256.

Brandt, K., Thewes, S., Overhage, J., Priefert, H., & Steinbuchel, A. (2001). Characterization of the eugenol hydroxylase genes (ehyA/ehyB) from the new eugenol-degrading *Pseudomonas* sp. strain OPS1. *Appl. Microbiol. Biotechnol.*, *56*(5–6), 724–730.

Chen, W., Ohmiya, K., Shimizu, S., & Kawakami, H. (1988). Isolation and characterization of an anaerobic dehydrodivaniliin-degrading bacterium. *Appl. Environ. Microbiol.*, *54*(5), 1254–1257.

Clark, G. S. (1990). Vanillin. *Perf. Flavor*.

Dawidar, A. M., Ezmiriy, S. T., Abdel-Mogib, M., el Dessouki, Y., & Angawi, R. F. (2000). New stilbene carboxylic acid from *Convolvulus hystrix*. *Pharmazie*, *55*(11), 848–849.

Dechter, R., & Pearl, J. (1985). Generalized best-first search strategies and the optimality of a*. *JACM*.

Funk, C., & Brodelius, P. E. (1994). Phenylpropanoid metabolism in suspension cultures of *Vanilla planifolia*. *Plant Physiol*.

Gasson, M. J., Kitamura, Y., McLauchlan, W. R., Narbad, A., Parr, A. J., Parsons, E. L., et al. (1998). Metabolism of ferulic acid to vanillin. A bacterial gene of the enoyl-SCoA hydratase/isomerase superfamily encodes an enzyme for the hydration and cleavage of a hydroxycinnamic acid SCoA thioester. *J. Biol. Chem.*, *273*(7), 4163–4170.

Gill, M. T., Bajaj, R., Chang, C. J., Nichols, D. E., & McLaughlin, J. L. (1987). 3,3′,5′-Tri-*O*-methylpiceatannol and 4,3′,5′-tri-*O*-methylpiceatannol: improvements over piceatannol in bioactivity. *J. Nat. Prod.*, *50*(1), 36–40.

Ishikawa, H., Schubert, W. J., & Nord, F. F. (1963). Investigations on lignins and lignification. xxviii. the enzymatic degradation of softwood lignin by white rot fungi. *Arch. Biochem. Biophys*.

Mayer, M. J., Narbad, A., Parr, A. J., Parker, M. L., Walton, N. J., Mellon, F. A., et al. (2001). Rerouting the plant phenylpropanoid pathway by expression of a novel bacterial enoyl-CoA hydratase/lyase enzyme function. *Plant Cell*, *13*(7), 1669–1682.

McShan, D. C., Rao, S., & Shah, I. (2003). Pathminer: predicting metabolic pathways by heuristic search. *Bioinformatics*, *19*(13), 1692–1698.

Morawski, B., Segura, D. A., & Orgnston, L. N. (2000). Substrate range and genetic analysis of acinetobacter vanillate demethylase. *J. Bacteriol*.

Murcia, M. A., & Martinez-Tome, M. (2001). Antioxidant activity of resveratrol compared with common food additives. *J. Food Prot.*, *64*(3), 379–384.

Narbad, A., & Gasson, M. J. (1998). Metabolism of ferulic acid via vanillin using a novel CoA-dependent pathway in a newly-isolated strain of *Pseudomonas fluorescens*. *Microbiology*, *144*(Pt. 5), 1397–1405.

Ogata, H., Goto, S., Fujibuchi, W., & Kanehisa, M. (1998). Computation with the KEGG pathway database. *BioSystems*.

Pometto, A. L., III, & Crawford, D. L. (1983). Whole-cell bioconversion of vanillin to vanillic acid by *Streptomyces viridosporus*. *Appl. Environ. Microbiol.*, *45*(5), 1582–1585.

Priefert, H., Rabenhorst, J., & Steinbuchel, A. (2001). Biotechnological production of vanillin. *Appl. Microbiol. Biotechnol.*, *56*(3–4), 296–314.

Russell, S., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. Prentice Hall.

Stephanopoulos, G. (1999). Metabolic fluxes and metabolic engineering. *Metab. Eng.*, *1*(1), 1–11.

Walton, N. J., Mayer, M. J., & Narbad, A. (2003). Vanillin. *Phytochemistry*, *63*(5), 505–515.

Walton, N. J., Narbad, A., Faulds, C., & Williamson, G. (2000). Novel approaches to the biosynthesis of vanillin. *Curr. Opin. Biotechnol.*, *11*(5), 490–496.

Zenk, M. H. (1965). Biosynthese von vanillin in *Vanilla planifolia*. *Z. Pflanzenphysiol*.