Chapter 8

Computational Tools for Guided Discovery and Engineering of Metabolic Pathways

Matthew Moura, Linda Broadbelt, and Keith Tyo

Abstract

With a high demand for increasingly diverse chemicals, as well as sustainable synthesis for many existing chemicals, the chemical industry is increasingly looking to biosynthesis. The majority of biosynthesis examples of useful chemicals are either native metabolites made by an organism or the heterologous expression of known metabolic pathways into a more amenable host. For chemicals that no known biosynthetic route exists, engineers are increasingly relying on automated computational algorithms, as described here, to identify potential metabolic pathways. In this chapter, we review a broad range of approaches to predict novel metabolic pathways. Broadly, these can rely on biochemical databases to assemble known reactions into a new pathway or rely on generalized biochemical rules to predict unobserved enzymatic reactions that are likely feasible. Many programs are freely available and immediately useable by non-computationally experienced scientists.

Key words: Metabolic network, Metabolic pathway design, Heterologous pathways, Enzyme database searching

1. Introduction

Our twenty-first century society has an increasing demand for a range of chemicals, from fuels (1) to polymeric precursors (2, 3) to drug and drug precursors (4–6), to name a few. Not only do we need these compounds (some of which are increasingly complex) in increasing quantities, but we also need to produce these compounds sustainably, minimizing the emission of toxic contaminates, suspected climate-change agents, and reduce the energy associated with production and purification. (7).

Biosynthesis of these compounds is well positioned to meet this need, as biological systems can synthesize complex molecules in high yield in moderate (i.e., aqueous, ambient temperature, and pressure) reaction conditions (8). The capture and redirection of *existing* metabolic pathways toward the production of industrially

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_8,

useful compounds constitutes one of the many options available to us for the development of sustainable biofuel energy and for the reduction in environmental wastes from chemical processing.

However, many of the needed chemicals are not made by any organism. How then do we harness the exquisite capabilities of biology but make compounds that have not been made in nature before? The answer most likely lies in the massive amount of available biochemical information (9, 10). This information is far too complex for manual design of new pathways, as has been the major strategy to date. Rather, computational approaches to design new metabolic pathways have risen (11, 12). Here, we review several recent computational tools, all focusing on this very problem—taking large-scale biochemical data and using it to better inform the design of synthetic metabolic pathways in unicellular organisms. We present ten software programs, describing a wide range of functionalities and approaches to the question of engineering reaction pathways (Table 1). We note that this collection is not a definitive list and other flavors of metabolic pathway design can be found.

1.1. Biochemical Data Sources: The Kyoto Encyclopedia of Genes and Genomes

To design a metabolic pathway, one first needs a source of biochemical data. The Kyoto Encyclopedia of Genes and Genomes (KEGG), an online database of biochemical reactions and their corresponding enzymes and genes, is one of the largest repositories of continuously updated, verified metabolic data available (13, 14). Because it is such a large database, it is a critical resource for scientists and engineers interested in exploiting biochemistry, and from the perspective of computational tools in this chapter, KEGG very often serves as *the* source for all available metabolism to incorporate into organism models or to use in potentially novel pathways. Though most of the programs are capable of linking up to any metabolic database, KEGG is almost always the one used.

1.2. Strategies for Finding Pathways

The main obstacle to pathway discovery is that of complexity, both from the large amount of metabolic reaction data (e.g., KEGG) and from the complex state of the organism. Online databases contain intractable amounts of enzyme/reaction information for a human to determine a pathway, and the search is complicated by the inherent interconnectedness of cellular metabolism.

The two main classes overcome the issue of complexity in a different way. Graph theory-based approaches perform analysis on the reaction databases directly. Biochemical data is broken down into edges and nodes. Most commonly compounds are nodes and reactions are edges, but this can vary depending on the approach. Finding paths is then a matter of following the different routes of the graph and trying to get from the starting compound to the product. Instead of using the data of online databases directly, rule-based methods generalize those reactions into reaction rules and use those rules to map out reaction paths to and from different

Table 1
Summary of computational metabolic pathway design software

Program name	Type of program	Latest publication[a]	Run-time scales	Summarized points	
BNICE	Rule based	Wu et al. (2011) (20)	Minutes to hours	• Biochemical reactions pruned by hand from online databases, reactions generalized into "operator files" based on EC system	• Can predict novel chemistries, suite of "modules" to prune generated networks, incorporates thermodynamic calculations into reactions and MFA
PathPred	Rule based	Tokimatsu et al. (2011) (26)	Minutes to hours	• Online service offered by KEGG, KEGG reactions described by RDM patterns, RDM patterns cluster according to metabolism	• Classes of RDMs/RPAIRs used to simulate reactions, can predict novel chemistry, predictions based on structural similarity rankings, regularly updated
UM-PPS	Rule based	Ellis et al. (2012) (28)	Seconds	• Online service offered by UM-BBD, focused on xenobiotic metabolism, reactions described by generalized database of btrules, fast	• Regularly updated btrules applied to substrates, predict chemistry, can predict novel chemistry, metabolic "logic entries" prune networks, user can set parameters
PPC	Graph theory—probabilistic	Yousofshahi et al. (2011) (35)	Seconds to minutes	• Operates on/necessitates large reaction database, probabilistic selection of edges based on connectivity, incorporates FBA	• Equivalent functionality/results as extensive search, terminates at compounds in organismal model
DESHARKY	Graph theory—probabilistic	Rodrigo et al. (2008) (38)	Seconds	• Operates on/necessitates large reaction database, unweighted probabilistic selection of edges, back-step probability increasing with path length	• Calculates "loads" on cellular system to represent impact of novel pathway, metabolic load = FBA, transcriptional/translational load = kinetic/chassis model, fast

(continued)

Table 1
(continued)

Program name	Type of program	Latest publication[a]	Run-time scales	Summarized points	
ReTrace	Graph theory—atom mapping	Pitkanen et al. (2009) (43)	Minutes to hours	• Creates atom-graph instance of a reaction database, incorporates RPAIR patterns to describe reactions, maximizes atom conservation	• Conservation measured in Z_O metric, optimized *k*-shortest path search approach for fastest computation, successfully finds branching and linear pathways
PathMiner	Graph theory—atom mapping	McShan et al. (2005) (44)	Seconds	• Create "state space" of a reaction database; compounds = states, reactions = state transformations; vectorized description of the database	• Heuristics-based pathway search, use A^* metric of chemical distance as heuristical tool, additional metrics incorporated in edge cost *e*, fast
MetaRoute	Graph theory—atom mapping	Blum et al. (2008) (42)	Seconds to minutes	• Semiautomated creation of reaction rules/sets from digital databases, enzyme clustering score used to inform rule creation	• *k*-lightest path search of weighted atom graph, graph and pathways are weighted toward atom conservation, fast
OptStrain	Linear programming based	Pharkya et al. (2004) (48)	Unknown	• Universal database provides reactions for pathway discovery, multiple databases in one, linear programming techniques find pathways through yield maximization	• Minimizes necessary heterologous expression, uses accessory program (OptKnock) to tie production to biomass growth, pathway optimization based on yield

[a]Most recent publication is the most recent article found from the original authors as of writing.

compounds. In a way, rule-based methods distill the online database information down and use it to create their own, more focused reaction databases focused on the starting/target compounds. By capturing the observed chemistry in these rules, the algorithm can predict new compounds and pathways that are not found in KEGG.

2. Computational Pathway Discovery Tools

2.1. Rule-Based Tools

A significant difference between rule-based methods and the other types discussed in this chapter lies in the ability to predict novel chemistry. While other methods can only reorganize known reactions and compounds, rule-based methods use those same databases to generalize biochemistry in terms of independent reaction rules. These rules are used to generate their own databases of reactions. The rule-based systems are an approach that bridges the often overlapping fields of metabolic engineering and synthetic biology through the inclusion of novel biochemistry into pathway discovery. These chemistries may be indicative of unidentified enzymatic activities or may provide potential targets for protein engineering to alter substrate specificity. An illustrative example of rule creation, as well as the rule's application to generate novel chemistry, is shown in Fig. 1.

Here, we will review the Biochemical Network Integrated Computational Explorer (BNICE), KEGG PathPred system, and the University of Minnesota Biodegradation and Biocatalysis Database's Pathway Prediction System (UM-PPS).

2.1.1. Biochemical Network Integrated Computational Explorer

The Biochemical Network Integrated Computational Explorer (BNICE) is a rule-based system which can carry out both analysis and synthesis: Analytical tools focus on finding all paths among known metabolites, while synthesis tools allow for the identification of novel intermediate compounds and reactions (15).

Inputs and Operation

BNICE consists of four modules: NetGen, thermodynamics, pathway, and thermodynamics-based metabolic flux analysis (TMFA). NetGen predicts enzymatic reactions and products based on generalized reaction rules, and its output serves as the input data for the rest of the program's features. Thermodynamics, pathway, and TMFA are all pruning, or analytical, modules written to take the initial network of NetGen and further analyze it to identify desired reactions and pathways from the initial pool.

Chemical reactions are reproduced by files called operators, which are used to predict enzymatic reactions. These operators have been hand distilled from enzymatic databases, like KEGG and the University of Minnesota Biodegradation/Biocatalysis Database (UM-BBD). Operators are named and generalized

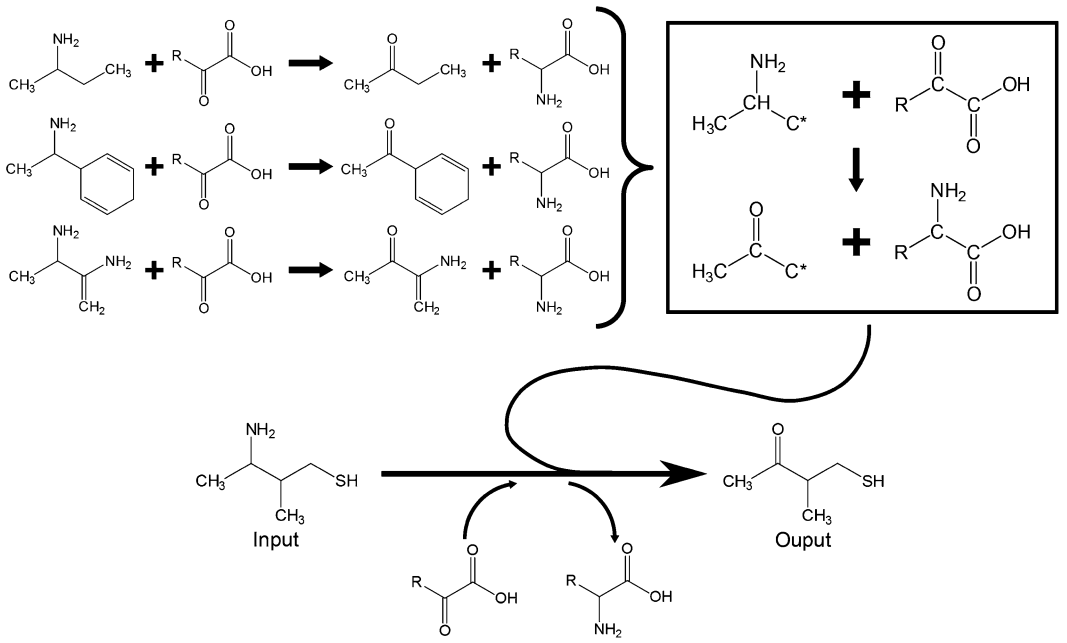

Fig. 1. An example of the rule-distillation process used in rule-based methods. Three reactions with very similar chemistries are compared, and the representative structures are used to describe the chemistry. This can then be applied to a novel substrate to potentially predict biochemistry.

according to their enzyme commission (EC) classification numbers. Rather than operators describing the *chemistry* and the *specific substrate*, the operator creation focuses on generalizing enzymatic reactions that contain the same first three EC digits and thus involve very similar chemistry, but could use different substrates.

Reactions in NetGen are simulated with a bond–electron matrix (BEM). Required reaction sites are defined in the individual operator files in a symmetric $N \times N$ matrix, where N is the number of atoms required in the reaction site. The elements of the matrix represent the bond order between overlapping rows and columns (e.g., a number 2 in an element of row O and column C would describe a double bond between a specific carbon and specific oxygen atom in the molecule). Having used the BEM to describe the required reaction site, the operator files then use an identically sized matrix with positive and negative integers to describe the making/breaking of bonds. The operators are able to cover a large spread of potential chemistries, all based on known biochemical transformations.

To generate its networks, BNICE requires an input of starting compounds, a list of operators to use in the network creation, and the number of generations to run. The pruning modules explained in the next section use output from NetGen.

Search Algorithm

Running BNICE for many generations can easily lead to a reaction network of hundreds of thousands of reactions. For this reason, we have developed the pathway, thermodynamics, and TMFA modules to take the large networks, remove undesired reactions, and provide additional useful information about the simulated biology to provide direction for potential experimental implementation.

Beginning from a starting compound, NetGen scans possible operators that can act on the start compound and generates a new pool of molecules based on the operators' chemistries. This can run for several generations to create many possible reactions. Work can also be done in retrosynthetic analysis, working backward from a product using retrosynthesis operators, which are operators with reversed directions of the initially defined chemistry (reactions that are considered physiologically reversible are handled similarly, but with both directions included in the full operator pool).

Given this initial biochemical network, pathway or thermodynamics can be used to uncover connections between pairs of desired compounds and approximate the thermodynamic changes of the reactions of interest. Pathway performs a basic depth-first search for linear pathways between the user-defined start and end points of the pathway given a maximum path length. Thermodynamics uses a group contribution method (GCM) to approximate ΔG_r across a reaction from changes in substructures (16), which have been assigned individual ΔG_f values. Using these two modules, thermodynamically favorable pathways of reasonable length are culled from the network. Lastly, to help choose from those proposed paths, TMFA can be used with each set of reactions to find those with the highest product yield, highest biomass, or other bioprocessing benchmarks. TMFA performs a flux balance analysis (FBA) of the effects from pathway integration into an organismal model, but with thermodynamic constraints on fluxes to better inform metabolism-scale effects of the pathway (17). FBA analyzes an organismal model and calculates maximum product yields as well as altered biomass rates that result from the introduction of heterologous reactions (see the OptStrain section for more details of FBA). In TMFA, metabolite activities are found, and optimal starting metabolite concentrations are suggested based on the thermodynamics.

Pathway Evaluation and System Validation

BNICE has been successfully applied to specialty chemical production (18), biodegradation of xenobiotics and environmental toxins (19), amino acid synthesis pathways (15), and biofuel production (20). Successes in these projects have independently reproduced known biological pathways and predicted novel biosynthesis routes that were already implemented in industrial settings. BNICE can be applied to a wide range of applications—novel chemistry prediction, native pathway discovery, and alternative pathway discovery. The program is written in C++ and can be run on Windows or Unix systems.

2.1.2. Cho Systems Framework

Cho et al. have implemented the BEM strategy of the BNICE algorithm and have extended it in several useful ways (21). Cho's algorithm is focused on retrosynthesis and has included modules for using chemical similarity, both for entire molecules and for substructures of a molecule, a group contribution thermodynamic analysis, pathway distance, and organism reaction specificity to help improve the selection of potentially useful pathways. Cho's algorithm has successfully predicted pathways for the synthesis of isobutanol, butyryl-CoA, and, like BNICE, 3-hydroxypropanoate.

2.1.3. PathPred

PathPred (22) is a system that utilizes the KEGG RPAIR and RDM databases, an atom-mapping rule-like system that uses KEGG's own data to break down reactions into reaction pairs with smaller rule descriptors for proposing novel metabolic pathways.

Inputs and Operation

The RPAIR database simplifies the reactions of the KEGG database and classifies reactions by reaction rules similar to BNICE, which are termed RDM patterns (for (R)eaction center atoms, (D)ifferent atoms, or (M)atched atoms, described below). The collection of these RDM patterns, and the reactant–product pairs described by the rules, is the KEGG RPAIR database (23). Reactants and products are compared and matched into reaction pairs based on a chemical similarity approach before a manual curation ensures proper pairing.

To create the RDM patterns, paired structures are compared and the overlapping substructures identified. The R atoms are those in the overlap region but on the border, hence where the reaction occurs. The D atoms are those bound to the R atoms but not in the overlap region. The M atoms are those bound to the R atoms in the overlap region. The RDM patterns describe how these three atom types change across a reaction pair and are meant to fully describe the chemistry performed in that pair (24). There are several RPAIR types that distinguish between different reaction pairings: *main*, *cofac*, *trans*, *ligase*, and *leave* pairs. PathPred utilizes only the RDM patterns from *main* pairs, which describe the pairings meant to be the focus of a particular reaction.

When the RPAIR database was first published (22), there were 7,091 reactant pairs described by 2,205 RDM patterns, with the bulk of those RDM patterns (64 %) each describing a single reaction pair.

Search Algorithm

PathPred predicts pathways from an observed clustering of RDM patterns to certain classes of metabolism (25). When running PathPred, the user must choose a type of metabolism—xenobiotic degradation or biosynthesis of secondary plant metabolites. By choosing a specific class, the program will use the associated RDM patterns. This allows for more reasonable computation times and more accurate predictions.

After selecting the type of desired prediction, the user inputs several starting parameters and is able to start a run of PathPred. Depending on the approach, either a starting compound or final compound is required (catabolism and anabolism, respectively), and additional data about chemical similarity thresholds and the number of prediction cycles can be set as well. After loading the compound, the PathPred algorithm analyzes the compound. First, it performs a similarity comparison between the input compound and the full database of KEGG compounds, looking for potential matches within a user-set threshold similarity value. Next, PathPred searches through all of the RDMs of those matched compounds and finds all RDMs that are applicable to the initial input compound. Third, the starting compound is subjected to the RDM transformations for those that matched the structural requirements. These last two steps are repeated until all transformations have been exhausted, at which point the compounds generated will be used in the first step, and the whole process is repeated for however many prediction cycles the user has specified.

Predicted pathways and reactions are ranked according to two different scoring schemes: reaction and pathway scores. The reaction score is a similarity index measure of how structurally close the compound input into the first step is to the compound that the RDM pattern is designed to act on. The pathway score is an average of the reaction scores contained within it. Compounds at the end of pathways with high pathway scores are targeted for the repeated prediction cycles.

Pathway Evaluation and System Validation

In their paper introducing the program, the authors of PathPred used their system to predict one biodegradation (1,2,3,4-tetrachlorobenzene to glycolate) and one biosynthesis (delphinidin to gentiodelphin) process, one each for the two different available sets of RDM patterns (22). The biodegradation exercise matched a documented path from the UM-BBD and found several other paths. The biosynthesis found several paths but not the known biological route, as a necessary RDM pattern was not a *main* pair and thus neglected from the process. More recently, PathPred was also successfully used to predict plant biosynthesis of fraxidin from umbelliferon (26) with several intermediary compounds known to be present in the *Saposhnikovia* root, a known biological source.

PathPred is freely available online through the KEGG database at the following address: http://www.genome.jp/tools/pathpred/. At the time of publication, PathPred was available in version 1.13.

2.1.4. University of Minnesota Pathway Prediction System

The University of Minnesota has developed one of the premier biodegradation databases, the UM-BBD. With this plethora of information, they have also taken steps to create a predictive biodegradation software program, which they have called the

University of Minnesota Pathway Prediction System (UM-PPS) (27). The program has been publicly available since late 2002/early 2003 and has seen continual development since its release, which we detail here.

Inputs and Operation

The actual use of the program is very straightforward and consists of only a few starting steps. First, the user inputs a starting compound through either a MarvinDraw applet or a SMILES string. The user is initially also given the option to limit the search to aerobic reactions. PPS then will generate and display the network in a short directed acyclic graph.

Search Algorithm

One of the original issues with the UM-PPS was that it required informed user intervention for each step. This required some knowledge of microbial biodegradation preferences in order to choose proper steps in generating a pathway. To overcome this, the software developers have implemented five network control features which capture much of the expert knowledge that was previously required, which they call "metabolic logic entries": absolute aerobic likelihood, immediate feature, relative reasoning, super rules, and variable aerobic likelihood. Full details about these features can be found elsewhere (28–30).

The core of UM-PPS is based on the distillation of the chemistries contained within the UM-BBD. The program uses generalized reaction rules, which they have termed biotransformation rules (btrules), that represent a large portion of the chemistries found on the database.

Using the chemistry of the UM-BBD, the authors of UM-PPS currently have 250 btrules for pathway prediction. These btrules are all designed to recognize and react with 50 predefined functional groups that have been distilled from the available reactions and are common across many xenobiotic metabolic reactions. When searching for potential reaction candidates, UM-PPS first performs a selection step, where btrules are matched to potential reactive sites, and then the reactions are carried out in the biotransformation step. Reaction rules are designed to be as generalized as possible, as long as the actual chemistry or known metabolism does not prevent this.

The UM-PPS and btrules are not written to describe any individual bacterium. The reactions contained within the UM-BBD (and the subsequent generalized btrules) are described based on known environmental degradation. The reactions in the database come from a wide variety of organisms and environmental observations. This is justified because of "increasing evidence that [xenobiotic degradation] is often consortial" (27). The UM-PPS is written with the intent of predicting whole-scale breakdown of xenobiotics, not necessarily the breakdowns occurring within a specific microbe. It is important to keep this in mind if using this

tool for metabolic engineering purposes, as btrule steps from different organisms may necessitate engineering beyond the introduction of proteins to a host cell.

Pathway Evaluation and System Validation

UM-PPS was validated in three ways upon publication (27). The program was able to recreate 72 % of the documented reactions of the UM-BBD at the time and gave at least one known pathway for 98 % of all UM-BBD compounds. It also reproduced five out of six biodegradation pathways as predicted by biodegradation experts for non-UM-BBD compounds. As an *in vivo* verification, three compounds predicted to release ammonia were successfully used as nitrogen sources in three cultures of soil-sampled bacteria.

UM-PPS has also been benchmarked against KEGG PathPred (discussed previously in this chapter) for biodegrative prediction (28). Not only did the UM-BBD perform equally or better for all tested compounds when compared to PathPred, but it also correctly predicted 81 % of the biodegradation routes.

The UM-PPS is freely available online for all users at the following link: http://umbbd.msi.umn.edu/predict/. The UM-BBD parent site offers all of the information about the btrules, use of PPS, all of the documented reactions on the site, and which btrules correspond to those reactions.

2.2. Graph-Based Tools: Probabilistic Approaches

Graph-based approaches find optimal or nonnative paths between substrates and products from a preexisting reaction network. There are many potential pools of reactions for these tools. In addition to the already discussed KEGG database, other options are MetaCyc (31), the UM-BBD (32), and organismal models (e.g., the *Escherichia coli iAF1260* model, (33)). Graph theory approaches take these large databases of metabolites and reactions and break them up into nodes and edges, where edges are directed arrows connecting individual nodes as illustrated in Fig. 2. Due to limitations in computational power and available time, the complexity of these massive reaction databases makes it intractable to search through every possible connection. For a breadth or depth-first search, the worst-case computational time is $O(|V| + |E|)$ (34) which, in the case of our biological networks, is a function of the total number of available nodes. For a pathway of length d in a network with an average node connectivity of b edges, the time will then be $O(b^d)$ which can rapidly become overly complex in the large, interconnected databases (e.g., KEGG).

Probabilistic analysis looks at a network and makes a decision about which edge to follow in a network, based on a probability weighting heuristic, which can vary with the individual program's approach. The two presented methods here also utilize biological models for the ranking of pathways. The two programs that we have chosen to describe are Probabilistic Pathway Construction (35) and DESHARKY (36).

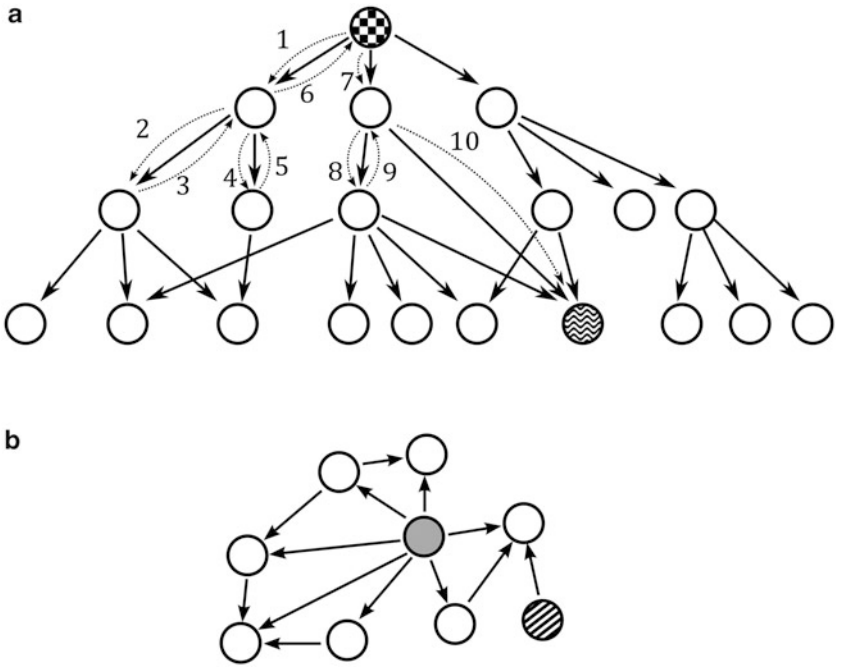

Fig. 2. Illustration of generalized graph-based analysis. (**a**) A depth-first analysis of a graphical tree. The *checkerboard and wave nodes* are the starting and target compounds, respectively. The *arrow numbers* illustrate the search with a maximum depth of 2. (**b**) Another graphical illustration demonstrating connectivity of nodes. The *gray circle* has a high connectivity, while the *striped one* a low connectivity. *Gray* could likely be a currency compound like NADH.

2.2.1. Probabilistic Pathway Construction

One option for pathway analysis of large reaction database networks is Probabilistic Pathway Construction (PPC). PPC utilizes a probabilistic graph-based search method to take large metabolic networks and find the most relevant pathways between two targeted compounds, searching for nonnative biosynthesis pathways. After discovering potential novel metabolic pathways, PPC then uses FBA to calculate the maximum yields of the desired product through all proposed pathways, subject to a biomass production constraint of 80% of that of the wild-type model.

Inputs and Operation

PPC requires three inputs: the biosynthesis product, a multi-organism reaction database to search (e.g., KEGG), and an organismal model for the expression host for FBA. The multi-organism database provides PPC potential production routes, while the organismal model gives PPC a list of native metabolites that can serve as potential pathway starting points.

Search Algorithm

PPC uses a modified depth-first graph search method to find pathway connections. A depth-first network search proceeds in steps going to nodes of deeper generations. When the search hits a stop signal (a node with no further edges or a search depth limit), it backtracks a step and proceeds to the next unexplored node.

When all edge searches from a given node have been exhausted, the program will retreat additional steps until it encounters novel edges to explore (Fig. 2a).

In PPC, molecules are treated as nodes and reactions leading to those nodes as edges. The program first looks to the designated biosynthesis product for all reactions resulting in its synthesis. It then uses a probabilistic selection to choose a reaction step. Having selected a reaction that produces the product of interest, PPC then looks to the starting substrates of that reaction and sets those as the new products, searching for reactions to those molecules with the same probabilistic scoring.

The research group tested probabilistic preferences for high connected nodes, low connected nodes, and for no inherent bias for connectivity (uniform) (Fig. 2b). In their work, the authors found that the best results were obtained with uniform connectivity.

PPC will continue to search in this manner unless it hits a predefined maximum pathway length, it encounters a compound that has already been included into the proposed pathway, or if the pathway encounters a metabolite that is native to the predefined host organism. It treats the first two cases as if it were a node with no further edges. For the third, it records the valid pathway and continues the search.

It is important to note that, being a system based on probabilities, the program must be run through many iterations in order to obtain an accurate representation of potential pathways. In the group's analysis, reliable results were returned between 500 and 1,500 iterations—the maximum yields stabilized at 500, but the average yields increased until 1,500. These results were found across several different types of synthesis products and the multiple scoring mechanisms.

Pathway Evaluation and System Validation

Having found several pathways, PPC is then able to place the paths into the organismal model and approximate theoretical maximum yields in an FBA analysis, which necessitates the selection of an organism model (37).

As validation of the pathways discovered by their program, the group looked to prior literature. While many predicted paths either had no experimental work to be found or reported production values not directly relatable to the maximum yield, there were several successful pathways previously implemented by others. These paths had been predicted independently by PPC and had comparable yield predictions.

The PPC system has a significant advantage over exhaustive searches in that it can save an enormous amount of time on longer path analyses. PPC's run-time scales linearly with respect to the number of reactions within the multi-organism database. An exhaustive search, however, becomes exponentially large with respect to the maximum pathway length. PPC's developers found

that to find a pathway of 23 steps through an exhaustive search would take approx. 400 years, while an identical search with PPC took a mere 6 min. Likewise, the results of a ten-step probabilistic search found pathways with essentially the same maximum theoretical yields as the equivalent exhaustive search. However, in this analysis, the authors have constrained themselves to a single metric of efficacy. Future studies and experimental validation will prove if additional metrics for successful pathway identification are necessary. As will be seen throughout this chapter, there currently exist many ideas about which metrics will accurately predict a pathway's success.

2.2.2. DESHARKY

DESHARKY was developed to find routes of either biosynthesis or biodegradation of compounds that are available in the KEGG database (36). The program relies heavily on the KEGG listings for all potential reactions to use in the pathways and uses a relatively simple pathway search. But where DESHARKY is unique is that it takes this a step further and uses the predicted growth rate of an organism as an indicator of how successful a given pathway will be in a physiological setting. This growth rate is determined under two independent systems, which the authors have classified as the transcriptional–translational load model and the metabolic model.

Inputs and Operation

This program contains a starter list of reactions, compounds, and enzymes from the KEGG database. The user can customize the simulation by updating this list, changing growth media components, or adding metabolites into an organism. If a compound of interest is already present within the host, then the user should input instead a set of desired termination host compounds.

To run, the program requires host organism compound data and the target compound. Biosynthesis will treat the target compound as a product, and biodegradation will use it as a substrate. Other adjustable parameters include number of iterations and maximum path length, among others.

Search Algorithm

DESHARKY takes the target compound and finds potential pathways to/from a host organism's metabolites to the input molecule. The pathway discovery is done in an unweighted probabilistic manner based on graph theory, similar to the approach taken by PPC. However, to avoid long pathways and improve convergence, each step after the first has an additional probability to retreat one step. This reversal probability increases with the number of forward steps that have been taken. Because it is a probabilistic pathway determination method, DESHARKY must be run over many iterations (the default is one million) in order to try and find all potential pathways. The program assumes all reactions of KEGG are reversible to allow both

biosynthesis and biodegradation (36). Metabolic steps with many nonnative compounds to the host organism are also disregarded.

Pathway Evaluation and System Validation

A significant challenge for novel pathway creation is not just finding the pathway but also evaluating the effects of pathway implementation on the physiological state of the organism. The biological consequences of pathways—consuming native metabolites, cytotoxic effects, and others—are often not evident within the pathways themselves. DESHARKY solves this with two independent measures of the altered cellular load resulting from a nonnative pathway: transcription–translation load and metabolic load.

Transcription–translation load estimates the negative cellular effects resulting from the resource drain for nucleic acid and enzyme polymerization, particularly the effects on RNA polymerase and nucleotide availabilities. DESHARKY uses a set of equations to tie growth rate to transcription and translation loads based on experimental measurements and first-order kinetics.

The model uses the available amino acid sequences on KEGG and an empirical mathematical cellular-chassis (organism) model they have developed to estimate reductions in growth rate arising from the heterologous pathway enzymes (38). This type of estimation of physiological effects is unique among pathway prediction systems. Metabolic load is estimated through a standard FBA approach, as used in several other programs. The two load calculators each give an independent assessment of the physiological impact from the novel enzymes introduced into the organism and provide two perspectives on cellular pathway integration, whereas most approaches would analyze only metabolic burden.

DESHARKY takes only a few seconds for each full run. The code is easily amenable to both distributed computing and modified weightings for the probabilistic searching. The program is open source, written in C/C++, and runs in UNIX environments (http://soft.synth-bio.org/desharky.html).

2.3. Graph-Based Tools: Atomistic Mapping

When searching for a pathway, one obstacle faced by graph-based tools is the effect of compounds with high degrees of connectivities (Fig. 2b) which can waste significant computational resources exploring all available edges. These compounds, like ATP or NADH, are often referred to as pool or currency metabolites and are involved in many biological reactions serving canonical roles: providing either oxidoreductive potentials or functional groups. One common solution to the high connectivity metabolite problem is to remove those nodes from the networks, though this prohibits the analysis of any pathways for the synthesis of those currency metabolites or that of any reactions where the compounds perform noncanonical roles.

The atom-tracing approach seeks to sidestep the high connectivity metabolite problem by following how bonds are made and

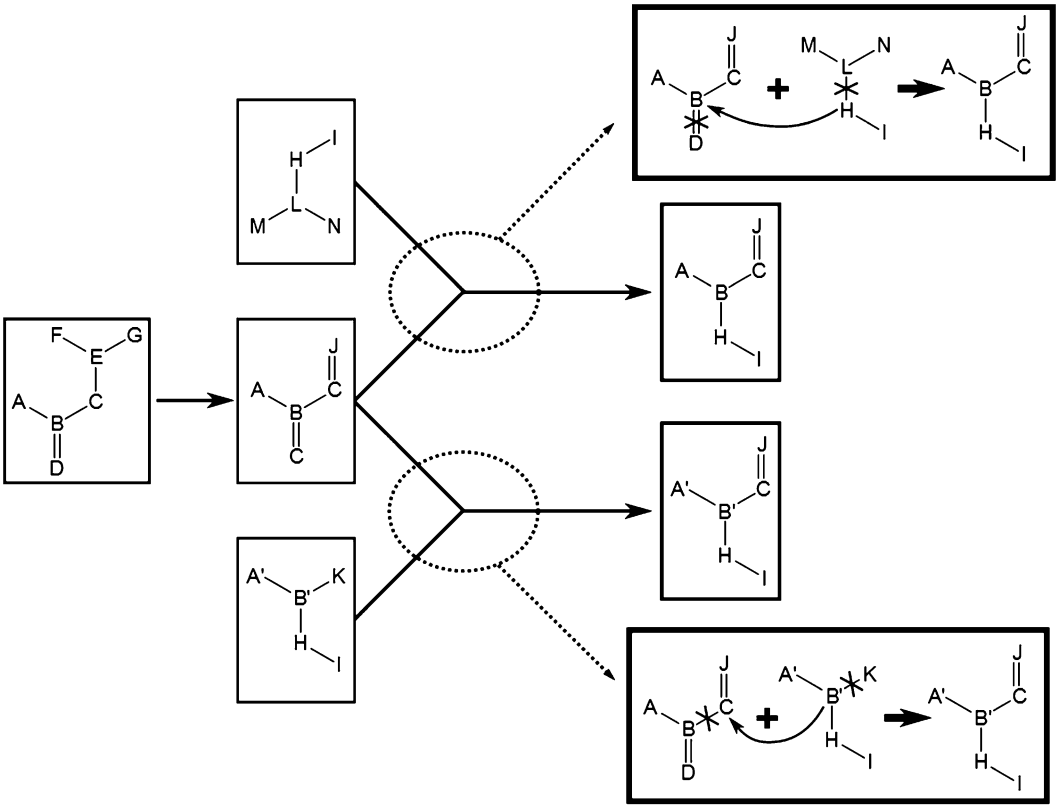

Fig. 3. A simplified graph theoretical analysis with atom mapping incorporated. Note how, although both routes yield the same product, in the *bottom path* the only atom maintained from the starting compound is C, whereas in the *top one* A, B, and C are all conserved.

broken across reactions, similar to that done by the rule-based approaches but in a much more reaction-specific manner. For most cases, currency metabolites donate relatively few atoms across a reaction (e.g., NADH is only involved in the transfer of protons and electrons). Atom-mapping rule sets can detect which substrate/product pairs have the most atoms in common, and pathway searches can focus on edges containing the highest degree of atomistic conservation, as is illustrated in Fig. 3. This strategy was pioneered by Arita (39).

These programs are able to make more informed pathway discoveries than the probabilistic tools we have discussed, with the added information from atom conservation driving pathway discovery. However, this focus may come at the cost of exploring more energetically favorable paths or pathways of significantly shorter length. The programs we have chosen to detail here are ReTrace (40), PathMiner (41), and MetaRoute (42).

2.3.1. ReTrace

ReTrace merges graph theory tracing with the specificity of atom tracing. The goal is for more accurate and concise pathway predictions—overcoming the difficulties faced by many with respect to "irrelevant connections" that often result from the searches focused on whole molecules. ReTrace also identifies branching pathways, a feature other pathway discovery tools often overlook (40).

Inputs and Operation

The program relies on KEGG for potential reaction steps and also utilizes the KEGG RPAIR database (discussed in the PathPred section) for information about where individual atoms move across reactions. This requires that a local version of the KEGG LIGAND data be downloaded from the parent website. The user inputs the source and target metabolites, and there are also several options with preset default values that the user can change for additional control.

ReTrace creates a graph of reactions connecting a starting and ending compound, using atom conservation to prioritize the most viable paths.

Search Algorithm

The overall ReTrace algorithm is made up of three procedures that are run sequentially: ReTrace, FindPath, and FindPathStart. ReTrace creates the atom and reaction graph that is used to find potential pathways. Again, KEGG is the biochemical database of choice. The graph constructed here is different from many others as both the reactions and the individual atoms of the compounds are represented by nodes, with edges describing how the individual atoms change compound locations across reactions. ReTrace constructs this graph from the database, also incorporating the stated target and source compound atoms into the map. The RPAIR reaction pairings are used to describe the core atoms for reactions listed in KEGG and how their bonds change. The program constructs a graph of source and target atom nodes and all reactions/edges of those atoms within a given number of reaction steps.

FindPath uses this graph for pathway construction and optimization, using the heuristic of atomistic conservation to rank them. Optimal predicted pathways should minimize the number of "dangling substrates," or substrates that are used in reactions of the path but enter into the pathway as foreign compounds. This helps minimize unknowns in metabolic reactions (uncertainty on the availability of a given compound), as well as yield much higher metabolic efficiencies, as energy is not wasted on removing and adding back atoms that are necessary.

Following the completion of this atom-traced graph, FindPath first finds the initial pathways within the reaction network. It runs a *k*-shortest path analysis between the source and target compounds, with the procedure FindPathStart being used to match final compound atoms to their starting compound atoms. FindPath looks through the top *k*-shortest paths to find those in which atoms are

and are not fully traced, and pathways where most atoms can be traced to source compounds are stored for further analysis. For those pathways with "unresolved atoms," or atoms untraced to one of the preset starting compounds, the program performs repeated FindPath analyses with those unresolved atoms to add more edges into the network. This is done until the program either runs out of edges to add or the atom conservation score increases beyond a preset minimum value.

Pathway Evaluation and System Validation

The authors have benchmarked their program using 13 metabolites as both source and target molecules (i.e., for each target, there were 12 sources). Computation times ranged from 1,000 s (CO_2) to 16,000 s (CMP-*N*-acetylneuraminate), the number of average discovered pathways to targets ranged from <30 (CO_2) to about 1,700 (CMP-*N*-acetylneuraminate), and the average path lengths ranged from five steps (CO_2) to 33 (*p*-Coumaroyl-CoA). The authors also showed they could find novel pathways from glucose to inosine-5′-monophosphate. Many pathways were found, including a known prokaryotic synthesis route, and routes using several alternative starting compounds were also identified. ReTrace was also used to reconstruct previously unknown pathways for the construction of amino acid carbon backbones in *T. reesei*, with success for many predicted pathways (43).

ReTrace has several unique aspects as compared to other graph-based approaches. By utilizing graph searching (which is fast, but prone to connectivity artifacts) to atomistic mapping (which is slower, but detailed), ReTrace capitalizes on each strategy's strength while minimizing its weakness. The other advantage to ReTrace is its ability to analyze branching pathways. Other graph-based tools make simplifying assumptions to pathway construction in a purely linear manner. By defining reactions and compounds as nodes, ReTrace can analyze all types of reactions—linear or branching—equally. Python code for ReTrace is freely available from http://www.cs.helsinki.fi/group/sysfys/software/retrace/ and requires the KEGG LIGAND database.

2.3.2. PathMiner

PathMiner tweaks the graph-based approach by using vectors of atoms to create "biochemical state spaces" and uses heuristic searching. Though the current implementation relies on KEGG, this approach could easily incorporate novel chemistries or other databases to predict new biochemical reactions (44).

Inputs and Operation

In PathMiner (41), the KEGG reactions and compounds are put into a biochemical state space, where compounds are the different states and the reactions constitute state transitions. This is analogous to a graph-based approach, but using vector notation. Each compound is described by a state vector **x**. Based on biochemistry, the authors have created a set of 145 unique atoms and atom bond

structures (e.g., C, O, S atoms with possible bond structures of C=O, O–S, S–S, among many others). The vector **x** contains 145 elements, and the different numbers within it detail the compound and its chemical structure. An example of this is given as $\mathbf{x}^{CO2} = [(C1)(O2)(C{=}O2)\ldots]$. Each reaction is a vector **t**, which is determined by the vector difference between states of reactant and product. Because individual compounds can perform many different chemistries, any individual **x** will likely have several **t**'s coming from it.

Combining the **t**'s with the **x**'s gives virtually the same result as would be found from a graph theory approach, but with the added benefit of the atomistic state descriptors. These 145 state elements could be easily be expanded to potentially include nonmolecular information about the compounds such as thermodynamic values (41).

Search Algorithm

With a state space defined, the authors take a computer science-based approach and view the discovery of new pathways as a classical state-search problem. An uninformed search of this space would be intractable, opening the door for heuristics to provide a quicker, informed analysis.

Chemical similarity of a molecule's state to the target molecule is used to determine which transitions/reactions are likely to lead toward the product of interest. For chemical similarity, this involves an additive combination of the distance traveled G (i.e., chemical similarity between start compound and present molecule) and the distance not yet traveled H (i.e., chemical similarity between target compound and present molecule) such that $F = G + H$. This is computed from the 145 atomic descriptors in what is essentially a state-space-based similarity index search. As these distances represent the costs of the system, the best paths are those with minimal F values. By expanding the fitness function F to include additional "edge cost" descriptors ($F = G + H + e$) for availability of precursors, heterologous vs. homologous enzymes, and others, much more sophisticated heuristics were incorporated into PathMiner (44).

After the database is fully characterized and assembled, PathMiner only requires a starting and ending compound. Searches are done by exploring all of the state transitions from each step that yields the best fitness values. The program terminates when there are no more states to explore and outputs all of the paths that were found between the two specified compounds.

Pathway Evaluation and System Validation

PathMiner's heuristic approach generally outperformed both uninformed breadth-first and depth-first searches for several different metabolic "themes"(41). In predicting vanillin synthesis (44), PathMiner found the native pathway and was able to suggest alternative host organisms for synthesis: *Brucella melitensis* or *Streptomyces coelicolor* over *E. coli*.

PathMiner uses state-space approaches and chemical distance to efficiently search known metabolic databases. Using the state-space approach, new parameters can be added to bias the network search and could be easily extended to other metabolic databases and even to potentially novel/predictive chemistries.

2.3.3. MetaRoute

MetaRoute (42) combines atom mapping and graph theory analysis with *k*-lightest path analysis to discover metabolic routes from large enzymatic databases. Here, a lightest path search will look for the pathways with the lowest connectivity values. The atom-mapping approach analyzes reactions and tracks where individual atoms go throughout a stated pathway.

Inputs and Operation

MetaRoute uses atom tracing analogous to ReTrace, where pathways with the highest number of atoms that are conserved from starting to final compound are given high scores. However, while ReTrace used RPAIR/RDM for its rules, MetaRoute developed its own rules in an automated fashion by looking for molecular substructures that are the same in reactant and product and then deducing the atoms that take part in the reaction. The KEGG and EcoCyc (45) databases were used to construct the rules (46). However, using this substructure approach can lead to redundancy because of repeated functional groups or substructures across different parts of a molecule. This redundancy in rules is solved by comparing atom-tracing reactions for enzymes clustered together based on the EC numbers. Presumably, similarly clustered enzymes should perform similar chemistries and thus have similar atom-tracing reactions. After multiple atom-tracing reactions are compared within a cluster, the atom-tracing reaction that predicts the most reactions in the cluster is chosen.

Search Algorithm

MetaRoute uses a modified *k*-lightest path search to discover novel pathways between two predefined compounds. The graph used is the reverse of the typical reaction graph, as MetaRoute uses nodes as reactions and edges as compounds. The novelty of the MetaRoute approach lies in the integration of the atom mappings into the *k*-lightest path search in what the authors have termed a "weighted atom-mapping graph." This involves two parts—the weighting of certain nodes and a structural moiety constraint. Each reaction (node) in the network has been pre-analyzed and is weighted by the atom transfer across the reaction. The *k*-lightest search maximizes the atom transfer during the search while subject to a minimum structural moiety constraint. This strategy can fail in the case of a reaction node with no atom mappings. As only 63% of the KEGG database has available reaction rules, the pathway search could potentially stall if no atoms can be traced to any products. When this happens, the program makes a choice based on the

k-lightest path criteria and then restarts the structural moiety constraint with the next compound.

Pathway Evaluation and System Validation

MetaRoute was used in a glycolysis search from D-glucose-6-phosphate to pyruvate. MetaRoute found three pathways: one documented glycolysis route and two novel paths. One of the novel pathways used an enzyme not yet classified into any documented pathways at the time of the paper's publication and had one fewer reaction than typical glycolysis. This is a strong case to use these types of computational tools for not just pathway creation but also for prediction of as yet-unknown biochemical pathways. MetaRoute has an intuitive interface, is easy to use, and is available online for free at http://abi.inf.uni-tuebingen.de/Services/MetaRoute/.

2.4. Linear Programming Tool

One approach that shares some similar approaches to graph-based methods but optimizes paths in a very different manner is the OptStrain program which uses linear programming (LP)/FBA-based tools (47).

2.4.1. OptStrain

OptStrain relies on online reaction databases for novel chemical steps and reactions to construct a pathway. Going beyond just pathway design, it will subsequently make suggestions about which genes from the native organism should be knocked out or added to enhance production.

Inputs and Operation

OptStrain uses a universal database, constructed from KEGG, UM-BBD, MetaCyc, and more, that is updated automatically. Only stoichiometrically balanced reactions from the databases are used in the search. The program also uses an organism model, which serves as the metabolic environment that novel reactions are introduced to. The program will attempt to maximize the production of a target compound given a large set of potential starting substrates.

Search Algorithm

OptStrain approaches the pathway search as an LP problem rather than a node/edge search problem. As with FBA, the universal reaction database is organized as a stoichiometric matrix **S**, with rows representing metabolites and columns representing reactions, where the element (n,m) has the stoichiometric coefficient of the nth metabolite within the mth reaction (48). Given a target product, OptStrain solves the universal matrix as an LP problem to maximize yields from a given substrate set by identifying a set of reactions that will achieve the desired biotransformation.

After identifying a set of possible reactions, OptStrain minimizes the number of new enzymes that would need to be added to the host organism. A mixed-integer linear programming problem, where heterologous enzymes to the host of interest are differentiated from native enzymes, is carried out to maximize the number of heterologous reactions that are removed while maintaining the

maximum product yield. In the last step, OptStrain utilizes another tool developed by the same authors, OptKnock (49), which does not design new pathways but attempts to optimize the host metabolism to maximize yield of the target compound through gene/reaction knockouts.

Pathway Evaluation and System Validation

In initial work, the authors optimized amino acid synthesis and found that seven amino acids could be synthesized by alternative pathways that were more energy efficient than native pathways (50). Subsequently, the authors analyzed hydrogen and vanillin synthesis (49). The first case exemplified OptStrain's ability to look to many different organisms and potential starting substrates, while the second succeeded most in minimizing heterologous genes for successful pathways. The hydrogen work predicted that *E. coli* could produce hydrogen but not in a manner coupled with growth rate. *Clostridium acetobutylicum* was identified as an additional candidate for glucose→hydrogen production that was tightly coupled with growth rate. Vanillin production in *E. coli* was predicted *de novo* with alterations similar to prior work by other researchers; however, OptStrain indicated much higher yields could be achieved with several knockouts not employed by the experimentalists.

3. Concluding Remarks

We have presented here a wide range of different computer programs of potential use in metabolic engineering. As society has a greater demand for sustainable chemicals, designing *de novo* metabolic pathways will become increasingly important. The different approaches and programs have their advantages and disadvantages. Graph-based methods can analyze known biochemistry and bring to light potentially unimaginable combinations of reactions to yield new ways to think about metabolism. This can be used to further pathway creation, as well as to inform scientists on bridging current gaps of metabolism. Several strategies, such as probabilistic searching, atom tracing, and formulating the search as an LP problem, have been successful in identifying pathways out of very large biochemical databases. However, these graph-based approaches rely completely on documented biochemical reactions and cannot predict unobserved but feasible biosynthesis pathways. To design metabolic pathways for compounds not observed to be produced by enzymes, rule-based approaches are essential. By sampling known biochemistry and distilling it into common reaction types, rule-based approaches can predict the potential reactions that could be performed on a given substrate. However, rule-based results are inherently high risk, as predicted chemistry may not be possible.

Regardless of the approach, computational tools for pathway design in metabolic engineering provide powerful methods for realizing novel biochemical processes. The tools described here, and others, demonstrate a diversity of approaches to pathway design. All of the programs, upon their release, presented some form of validation or verification of the program's efficacy for pathway prediction, typically by showing the program's ability to predict already known pathways. While few examples to date have taken a designed pathway and demonstrated that it could work (2), we expect to see a strategy of computer-aided design of metabolic pathways, with implementation, to have an increasing prominence in the design of new synthesis processes for new chemicals.

References

1. Shen CR, Liao JC (2008) Metabolic engineering of *Escherichia coli* for 1-butanol and 1-propanol production via the keto-acid pathways. Metab Eng 10:312–320
2. Yim H, Haselbeck R, Niu W, Pujol-Baxley C, Burgard A, Boldt J, Khandurina J, Trawick JD, Osterhout RE, Stephen R, Estadilla J, Teisan S, Schreyer HB, Andrae S, Yang TH, Lee SY, Burk MJ, Van Dien S (2011) Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol. Nat Chem Biol 7:445–452
3. Kind S, Jeong WK, Schroder H, Wittmann C (2010) Systems-wide metabolic pathway engineering in Corynebacterium glutamicum for bio-based production of diaminopentane. Metab Eng 12:341–351
4. Trantas E, Panopoulos N, Ververidis F (2009) Metabolic engineering of the complete pathway leading to heterologous biosynthesis of various flavonoids and stilbenoids in *Saccharomyces cerevisiae*. Metab Eng 11:355–366
5. Ajikumar PK, Xiao WH, Tyo KE, Wang Y, Simeon F, Leonard E, Mucha O, Phon TH, Pfeifer B, Stephanopoulos G (2010) Isoprenoid pathway optimization for Taxol precursor overproduction in *Escherichia coli*. Science 330:70–74
6. Ro DK, Paradise EM, Ouellet M, Fisher KJ, Newman KL, Ndungu JM, Ho KA, Eachus RA, Ham TS, Kirby J, Chang MC, Withers ST, Shiba Y, Sarpong R, Keasling JD (2006) Production of the antimalarial drug precursor artemisinic acid in engineered yeast. Nature 440:940–943
7. Keasling JD (2012) Synthetic biology and the development of tools for metabolic engineering. Metab Eng 14:189–195
8. Stephanopoulos G, Stafford DE (2002) Metabolic engineering: a new frontier of chemical reaction engineering. Chem Eng Sci 57:2595–2602
9. Pennisi E (2005) How will big pictures emerge from a sea of biological data. Science 309:94
10. Philippi S, Kohler J (2006) Addressing the problems with life-science databases for traditional uses and systems biology. Nat Rev Genet 7:482–488
11. Copeland WB, Bartley BA, Chandran D, Galdzicki M, Kim KH, Sleight SC, Maranas CD, Sauro HM (2012) Computational tools for metabolic engineering. Metab Eng 14:270–280
12. Medema MH, van Raaphorst R, Takano E, Breitling R (2012) Computational tools for the synthetic design of biochemical pathways. Nat Rev Microbiol 10:191–202
13. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M (2012) KEGG for integration and interpretation of large-scale molecular data sets. Nucleic Acids Res 40:D109–D114
14. Kanehisa M, Goto S (2000) KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28:27–30
15. Hatzimanikatis V, Li C, Ionita JA, Henry CS, Jankowski MD, Broadbelt LJ (2005) Exploring the diversity of complex metabolic networks. Bioinformatics 21:1603–1609
16. Jankowski MD, Henry CS, Broadbelt LJ, Hatzimanikatis V (2008) Group contribution method for thermodynamic analysis of complex metabolic networks. Biophys J 95:1487–1499
17. Henry CS, Broadbelt LJ, Hatzimanikatis V (2007) Thermodynamics-based metabolic flux analysis. Biophys J 92:1792–1805
18. Henry CS, Broadbelt LJ, Hatzimanikatis V (2010) Discovery and analysis of novel metabolic pathways for the biosynthesis of industrial

chemicals: 3-hydroxypropanoate. Biotechnol Bioeng 106:462–473

19. Finley SD, Broadbelt LJ, Hatzimanikatis V (2010) In silico feasibility of novel biodegradation pathways for 1,2,4-trichlorobenzene. BMC Syst Biol 4:7
20. Wu D, Wang Q, Assary RS, Broadbelt LJ, Krilov G (2011) A computational approach to design and evaluate enzymatic reaction pathways: application to 1-butanol production from pyruvate. J Chem Inf Model 51:1634–1647
21. Cho A, Yun H, Park JH, Lee SY, Park S (2010) Prediction of novel synthetic pathways for the production of desired chemicals. BMC Syst Biol 4:1–16
22. Moriya Y, Shigemizu D, Hattori M, Tokimatsu T, Kotera M, Goto S, Kanehisa M (2010) PathPred: an enzyme-catalyzed metabolic pathway prediction server. Nucleic Acids Res 38:W138–W143
23. Kotera M, Okuno Y, Hattori M, Goto S, Kanehisa M (2004) Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. J Am Chem Soc 126:16487–16498
24. Oh M, Yamada T, Hattori M, Goto S, Kanehisa M (2007) Systematic analysis of enzyme-catalyzed reaction patterns and prediction of microbial biodegradation pathways. J Chem Inf Model 47:1702–1712
25. Oh M, Yamada T, Hattori M, Goto S, Kanehisa M (2007) Systematic analysis of enzyme-catalyzed reaction patterns and prediction of microbial biodegradation pathways. J Chem Inf Model 47:1702–1712
26. Tokimatsu T, Kotera M, Goto S, Kanehisa M (2011) KEGG and GenomeNet resources for predicting protein function from omics data including KEGG PLANT resource. Protein Function Prediction for Omics Era, 271–288
27. Hou BK, Ellis LBM, Wackett LP (2004) Encoding microbial metabolic logic: predicting biodegradation. J Ind Microbiol Biotechnol 31:261–272
28. Ellis L, Wackett L (2012) Use of the University of Minnesota biocatalysis/biodegradation database for study of microbial degradation. Microb Inform Exp 2:1
29. Fenner K, Gao J, Kramer S, Ellis L, Wackett L (2008) Data-driven extraction of relative reasoning rules to limit combinatorial explosion in biodegradation pathway prediction. Bioinformatics 24:2079–2085
30. Gao JF, Ellis LBM, Wackett LP (2010) The University of Minnesota biocatalysis/biodegradation database: improving public access. Nucleic Acids Res 38:D488–D491
31. Caspi R, Foerster H, Fulcher CA, Hopkinson R, Ingraham J, Kaipa P, Krummenacker M, Paley S, Pick J, Rhee SY, Tissier C, Zhang PF, Karp PD (2006) MetaCyc: a multiorganism database of metabolic pathways and enzymes. Nucleic Acids Res 34:D511–D516
32. Ellis LBM, Hou BK, Kang WJ, Wackett LP (2003) The University of Minnesota biocatalysis/biodegradation database: post-genomic data mining. Nucleic Acids Res 31:262–265
33. Feist AM, Henry CS, Reed JL, Krummenacker M, Joyce AR, Karp PD, Broadbelt LJ, Hatzimanikatis V, Palsson BO (2007) A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol Syst Biol 3:1–18
34. Reif JH (1985) Depth-1st search is inherently sequential. Inform Process Lett 20:229–234
35. Yousofshahi M, Lee K, Hassoun S (2011) Probabilistic pathway construction. Metab Eng 13:435–444
36. Rodrigo G, Carrera J, Prather KJ, Jaramillo A (2008) DESHARKY: automatic design of metabolic pathways for optimal cell growth. Bioinformatics 24:2554–2556
37. Papoutsakis ET (1984) Equations and calculations for fermentations of butyric-acid bacteria. Biotechnol Bioeng 26:174–187
38. Carrera J, Rodrigo G, Singh V, Kirov B, Jaramillo A (2011) Empirical model and in vivo characterization of the bacterial response to synthetic gene expression show that ribosome allocation limits growth rate. Biotechnol J 6:773–783
39. Arita M (2000) Metabolic reconstruction using shortest paths. Simulat Pract Theor 8:109–125
40. Pitkanen E, Jouhten P, Rousu J (2009) Inferring branching pathways in genome-scale metabolic networks. BMC Syst Biol 3:103
41. McShan DC, Rao S, Shah I (2003) PathMiner: predicting metabolic pathways by heuristic search. Bioinformatics 19:1692–1698
42. Blum T, Kohlbacher O (2008) MetaRoute: fast search for relevant metabolic routes for interactive network navigation and visualization. Bioinformatics 24:2108–2109
43. Jouhten P, Pitkanen E, Pakula T, Saloheimo M, Penttila M, Maaheimo H (2009) (13)C-metabolic flux ratio and novel carbon path analyses confirmed that Trichoderma reesei uses primarily the respirative pathway also on the preferred carbon source glucose. BMC Syst Biol 3:1–16

44. McShan D, Shah I (2005) Heuristic search for metabolic engineering: de novo synthesis of vanillin. Comput Chem Eng 29:499–507
45. Keseler IM, Collado-Vides J, Santos-Zavaleta A, Peralta-Gil M, Gama-Castro S, Muniz-Rascado L, Bonavides-Martinez C, Paley S, Krummenacker M, Altman T, Kaipa P, Spaulding A, Pacheco J, Latendresse M, Fulcher C, Sarker M, Shearer AG, Mackie A, Paulsen I, Gunsalus RP, Karp PD (2011) EcoCyc: a comprehensive database of *Escherichia coli* biology. Nucleic Acids Res 39:D583–D590
46. Blum T, Kohlbacher O (2008) Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks. J Comput Biol 15:565–576
47. Pharkya P, Burgard AP, Maranas CD (2004) OptStrain: a computational framework for redesign of microbial production systems. Genome Res 14:2367–2376
48. Fell DA, Small JR (1986) Fat synthesis in adipose-tissue—an examination of stoichiometric constraints. Biochem J 238:781–786
49. Burgard AP, Pharkya P, Maranas CD (2003) OptKnock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. Biotechnol Bioeng 84:647–657
50. Burgard AP, Maranas CD (2001) Probing the performance limits of the *Escherichia coli* metabolic network subject to gene additions or deletions. Biotechnol Bioeng 74:364–375