Design Automation for Synthetic Biological Systems

Douglas Densmore
Boston University

Soha Hassoun
Tufts University

Editors' notes:
Through principled engineering methods, synthetic biology aims to build specialized biological components that can be modularly composed to create complex systems. This article outlines bio-design automation using two complementary design approaches, bottom-up modular construction from biological primitives and pathway-based approaches. The article also highlights future challenges for both.

—Douglas Densmore, Boston University, and Soha Hassoun, Tufts University

Engineering biology

Synthetic biology encompasses the synthesis or enhancement of complex biological systems to elicit behaviors that do not exist in nature. Synthetic biology promises to introduce new biotherapeutic, bioremediation, biosensing, bioenergy, and biomaterials based solutions to a diverse set of grand challenges. Progress in designing novel biological systems has been hindered primarily by the complexity of biology. Living systems perform a variety of functions including self-replication, cell-to-cell communication, cell division and differentiation into a more specialized collections of cells. In contrast to human-engineered systems, much of the underlying science of biology is still largely a mystery. Every organism is unique and studied under very specific environmental conditions. Extraordinary efforts spanning multiple hierarchical levels are needed to completely characterize and understand every component and reaction in the context of the whole. Despite knowledge gaps, experimentalist utilize their instincts and experiences to engineer biological systems, often through trial and error, and more recently with some assistance from computational tools. Recent achievements include engineered bacteria to treat malaria [1], to invade cancer cells [2], to remove toxins such as herbicides from the environment [3], to produce biofuels such as ethanol and butanol [4], and to develop highly tuned biological sensors [5].

Computational methods and tools to (re-)engineer and synthesize biological systems, referred biodesign automation, are poised to play a critical role in the development of novel biological systems similarly to how electronic design automation (EDA) transformed designing VLSI circuits since the advent of silicon transistors in the 1950s. BDA tools will conceptually span specification, modeling, analysis, design, simulation, synthesis, verification, and assembly. Similarly to how Moore's law has shaped the EDA industry, biological discoveries, reduced DNA synthesis costs and technical innovations will drive BDA tools. Biology-specific metrics (e.g., evolutionary stability and reliability) and application-specific metrics (e.g., yield of desired compounds) will be used to evaluate design quality.

Digital Object Identifier 10.1109/MDT.2012.2193370
Date of publication: 5 April 2012; date of current version: 31 August 2012.

One synthetic biology design approach aims for systematic construction of larger systems from biological primitives. DNA-encoded "Parts" are designed and then assembled to create modular "Devices" that can be integrated into a host organism or assembled into a larger "System." Such hierarchy paves way to, familiar, and proven engineering concepts such as abstraction, modularity, standardization and composition. Devices such as toggle switches and oscillators have been experimentally built (see sidebar 1). To specify context and assembly chemistries, Parts are characterized and catalogued in libraries (see sidebar 2, iGEM). The focus on design methodologies and supporting tools are now emerging (see sidebar 2, IWBDA).

A second complementary approach manipulates existing biological pathways or adds novel pathways to an existing cell. This approach has been long advocated by metabolic engineering, the discipline concerned with optimizing genetic and regulatory processes within cells to increase production of particular substances. Pathway engineering of microbially produced artemisnic acid as a viable source of antimalarial drugs [1] resulted in decreasing the production cost from $2.40 per dose to $0.25 per dose, enabling cheaper treatment for marlaria that threatens 300–500 million people and annually kills more than one million people. While design methodologies utilizing this approach are often *ad hoc* and domain specific (therapeutics versus biofuels), they share point computational tools that aid the design cycle.

This article reviews basic concepts and BDA tool advances for these two approaches. We first provide a short review and shed some light on how the two approaches evolved. We then describe computational design tools available for each approach, drawing parallels between BDA and EDA when appropriate. We believe BDA has the potential to usher a design era that can radically transform living systems.

Biology primer

DNA, discovered in 1953 and consisting of two long entwined strands of repeating units called nucleotides, encodes genetic instructions that are executed during the development and function of all known living organisms. This encoding/decoding process is known as the "central dogma," and is illustrated in the top box in Figure 1. Intricate biological machinery executes the code, performing the following transformations: DNA becomes mRNA (transcription via the RNA polymerase machinery);

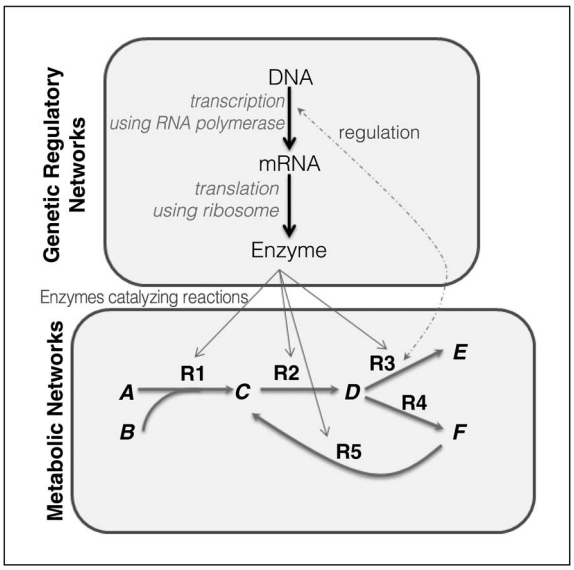


Figure 1. Overview of a biological system at two levels: GRNs and metabolism. The transcription/translation machinery (top box) produce enzymes, which in turn control the rate of reactions within metabolic pathways (bottom box). Feedback loops within the metabolic networks and across levels allow the system to self regulate (adapted from [7]).

mRNA then becomes a protein (translation via the ribosome machinery). Proteins that influence biochemical reactions are referred to as "enzymes." Transcription and translation processes specify the production rate, conditions and concentrations of produced proteins. Example circuitry that performs these tasks can be referred to as a "genetic regulatory network," GRN. Here genes both regulate their own expression as well as the expression of other genes.

Realized proteins in turn catalyze (accelerate) biochemical reactions, as illustrated in the bottom box of Figure 1. Reactions consume and produce metabolites and signify chemical activities in living cells. Each reaction is associated with a flux, the molecule turnover rate. Reactions are organized in pathways, and conceptually as functional modules. Several organizational networks have been identified within cells such as cell signaling (communication) and metabolic networks. Metabolism sustains life within cells. The cell's metabolism is regulated using positive or negative (control) feedback loops at multiple levels. Regulation also occurs through allosteric regulation.

Conceptually and admittedly simplistically, GRNs provide control over the data flow in a system's pathways. GRN engineering has focused on designing primitives that can be assembled into meaningful control circuitry. The pathway approach has focused on either adding new data paths, removing paths, or modifying control of existing paths. While tools for the hierarchical, part-based assembly approaches are relatively new, top-down modeling and analysis tools are more established. These approaches have their roots in metabolic engineering and in systems biology, a field that calls for an integrative approach for studying and analyzing biological systems. These two particular approaches were chosen over others for this article because they clearly benefit from structured design automation tools and flows. For example, a promising technique is whole-genome engineering [6]; however, the technique is recent and computational tools and design flows have not been established.

While genetic and metabolic networks are currently treated as separate networks with vastly different operational time scales, understanding feedback and influences is necessary to engineer biology. The availability of novel high-throughput experimental methods allows various (-omics) measurements, which in turn will enable correlating activities and models at various levels. Genomics refers to understanding the DNA composition through sequencing and annotation of whole genomes. Transcriptomics measure mRNA and signify gene expression levels. Proteomics measure protein abundance. Metabolomics measure the concentration of metabolites. Fluxomics measure fluxes through the metabolic network. In the future, detailed coordinated models spanning multi-scale levels will become available. Moreover, manipulating biology at multiple scales will enhance the capabilities of engineered complex biological systems. For example, a GRN may provide some monitoring capability of a particular metabolite within the system. Once exceeding a particular value, the GRN may change enzyme values to enable suppressing the production of a particular metabolite. Appropriately dispensing medicine is one application example. Just like in electronics, the design technology and the tools will evolve over time to enable creating complex synthetic systems.

Bottom up: DNA to parts to devices

A genetic circuit is a collection of biological components organized to detect biological signals via a series of transcriptional and translational steps and to produce other signals which ultimately define the behavior (output) of the circuit. Signals are biological and typically small molecules and proteins. Small molecules often are externally introduced in the system or present in the surrounding environment. Proteins are collections of amino acids produced in the cell during translation of mRNA. The modular construction approach abstracts biological functionality into "Parts" and then utilizes the Parts to create "Devices." Parts are specific DNA sequences categorized by their role in the central dogma. Attempts are made to characterize the performance of these Parts and standardize how they are composed into larger Devices. "Systems" can then be created by composing Devices. For example, one might encapsulate a green florescent protein (GFP) as a Part by isolating the specific gene that encodes that protein. In addition to the DNA sequence for the gene, additional DNA sequences (called "restriction sites") will be added at both ends of the DNA. These sequences are selected such that other Parts using similar sequences might be more easily joined with the introduction of specific enzymes which cut the DNA at these sites leaving single stranded overhangs which can be matched and ligated together. A heavy metal sensor Device could be created by joining a lead sensitive Part with a GFP Part (e.g., glow green when lead is detected). These Devices then would be put into organisms [e.g., Escherichia Coli (*E.Coli*)] and flow cytometer data will report on the florescence levels achieved by these Devices in the presence or absence of a number of control and experimental companion Parts. This data will be used to drive the creation of models which predict the florescence levels in more complex designs. This Device now will be added to the library of Devices and can be used in future designs.

Example: Transcription-based combinational logic

Figure 2 provides a genetic regulatory network (GRN) of a 2-input (*tf*1 and *tf*2) single-output (*pro*1) "NOR gate" [8]. It should be pointed out that the following is a prokaryotic (e.g., bacterial) system. The translational/transcriptional mechanisms described will be different in other organisms. The general concepts will be similar in eukaryotes but items such as ribosome binding site structure, translational/transcriptional coupling, and the

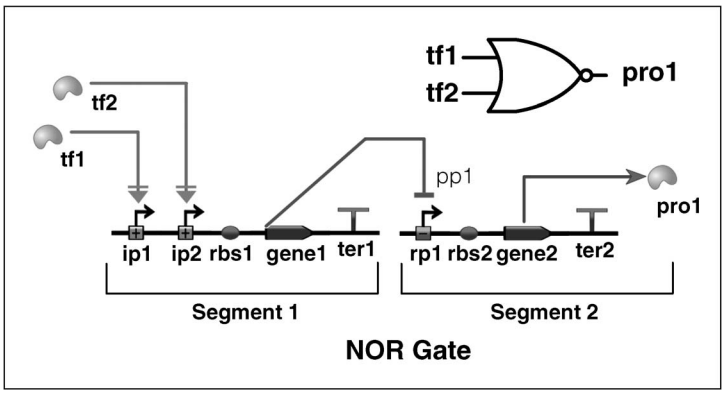


Figure 2. Synthetic biological genetic regulatory networks have been described using traditional digital logic terminology. Example circuits include a two-input (*tf*1 and *tf*2) single-output (*prot*1) NOR gate (described in the text). The presence or absence of small molecules (ultimately indicating transcription) dictates the production or absence of an output protein.

presence of additional organelles will make the details decidedly different. The design consists of the following elements:

1) The system input consists of small molecule *transcription factors* (*tf*1 and *tf*2), which may not exist naturally in the cellular context and can be added from an external source.
2) *Inducible Promoters* (*ip*1 and *ip*2). Promoters as a biological primitive broadly can be considered where transcription begins on the DNA. In the absence of transcription, DNA cannot encode for a protein (it never will have become mRNA). Only the genes "downstream" (to the right by convention) of a promoter have the ability to be transcribed and ultimately expressed. Inducible promoters allow for RNA polymerase to bind and begin transcription only when specific transcriptions factors come into physical contact with the operator site of the promoter. In this case, ip1 and ip2 require tf1 and tf2, respectively.
3) *Ribosome binding sites* (*rbs*1 and *rbs*2) indicate where a ribosome will attach to the mRNA transcript to begin translation. Once bound, the ribosome will examine the mRNA in three base pair chunks (codons) for the purpose of translation. These codons correspond to amino acids as defined by the genetic code.
4) *Gene*1 and *Gene*2 are specific regions of the DNA which encode for specific proteins. These segments of DNA are flanked by start and stop codons (specific three base pair DNA sequences) which signal the ribosome to begin creating the amino acids which are chained together by peptide bonds to create the protein expressed by the gene. The amino acid chain begins at the start codon and finishes at the stop codon.
5) *Terminators* (*ter*1 and *ter*2) indicate where the RNA polymerase will end the process of transcription. This is where the RNA polymerase will leave the DNA and end the mRNA transcript.
6) *Repressible promoter* (*rp*1). This primitive is similar to an inducible promoter. However, it is turned off in the presence of its transcription factors to prevent transcription from occurring at the transcriptional start site.
7) *Protein* (*pro*1) is expressed by Gene2. This is a collection of amino acids created during translation of Gene2 by a ribosome. This is the system output.

For our purposes the reader can consider the machinery of the central dogma occurring from left to right. The GRN NOR gate acts as follows.

1) In the presence of externally introduced small molecules (*tf*1 or *tf*2) either *ip*1 or *ip*2 (or both) will be induced. This process will allow RNA polymerase to bind upstream (to the left) of *rbs*1.
2) RNA polymerase will produce an mRNA transcript containing sequences for *rbs*1 and *gene*1. Nothing further downstream will be transcribed because of terminator *ter*1.
3) A ribosome will then bind to this transcript at the *rbs*1 site and translate *gene*1 into a protein.
4) *gene1*'s protein will now act as a repressor of *rp*1. By doing so, it will prevent transcription of the second DNA segment which in turn ultimately prevents the production of *pro*1.
5) In true NOR fashion, if either or both of the inputs (*tf*1 or tf2) are present, then the output *pro*1 is not present. In the absence of *tf*1 and *tf*2, nothing represses *rp*1 and hence *pro*1 will be expressed. The time for transcription and translation in this system is in the order of tens of minutes from first input to output signal.

It should be pointed out that the DNA for this style of NOR gate has two distinct segments (labeled

on Figure 2). While each segment requires contiguous DNA, the segments themselves need not be. In fact, their order in a single DNA could be changed, they could be on opposite strands, or they could be on different DNA molecules in the same cell. This spacial computation aspect highlights a key difference between circuits in silicon and those in DNA.

Building genetic regulatory networks

The process of physically building a GRN requires the following steps.

1) Obtain the DNA segments for the primitives of interest. These can include promoters, rbs, genes, and terminators. These can be isolated from natural sources or created via a chemical process called "DNA synthesis." This begins as a request for a specific DNA sequence (e.g., ACTTTAG) and ends with a physical DNA sample stored in a tube in a laboratory freezer. Companies like DNA2.0, GeneArt, and Blue Heron provide these services, priced per base pair (~$1/bp). This process is more accurate and expensive compared to standard assembly (step 3).
2) PCR amplify the DNA. Polymerase Chain Reaction (PCR) creates several orders of magnitude more DNA than the initial starting sample. This process will provide enough DNA primitives to ensure successful composite assembly going forward given a certain concentration of DNA is needed for assembly.
3) Assemble DNA primitives into a composite DNA Device. There are a variety of assembly chemistries for this process but they all involve making the DNA primitives compatible with their neighboring primitives, exposing a single strand of the double stranded DNA primitives, and ligating the complementary single strands together. Methods include BioBricks [9] and Gibson [10]. This manual approach is relatively inexpensive but potentially prone to error. Note that if one wished to bypass this step they could synthesize the entire Device in step 1).
4) Insert the DNA into a host organism. Depending on the organism the process can differ dramatically but this ultimately results in introducing the DNA into a cell. In prokaryotes (e.g., bacteria) this is done via a process called transformation where the DNA is made circular (also called a DNA plasmid) and taken up into the cell via a process called "heat shock" where the competent cells (cells developed specifically for this process) are heated and cooled causing the outer membrane to become porous enough for the DNA to enter.
5) Growing cultures of the host organism. The cells with the DNA of interest are allowed to go through the cell division cycle to produce colonies when plated on growth media. The DNA introduced to the cells has an "origin of replication" associated with it so that upon division the newly introduced DNA is also in the daughter cells.
6) Harnessing the DNA back out for future use from the host organisms. The cells can be harvested and the DNA extracted through a process called "plasmid preparation." In this way you go from one small set of DNA constructs to many more. These are saved for another round of processing in the future.

More general information on Part-based design can be found at `partsregistry.org`.

The design process

Bio-design automation for Part-based systems can encompass specification, design, assembly, and data management workflows. Figure 3 illustrates that a formal biological specification can be created. Here the biological behavior and constraints on this behavior can be described. For example, under which biological inputs the system should respond to and its eventual actuation requirements (e.g., glow green, produce chemical X, etc.) can be formally captured. Constraints such as the desired reaction concentrations, permitted or desired primitives to be used, or general Device topologies can be specified. Languages such as Proto [11], Eugene [12] and GEC [13] exist for this stage. The Design stage then takes abstract genetic regulatory networks (collections of transcriptional promoters and genes created in the previous stage) and represents them as bi-partite graphs of promoters and transcription factors. Using graph isomorphism algorithms Parts are assigned to these elements based on available library primitives and their experimentally characterized performance. The goal is to cover this network with elements which when joined and put into a specific cellular context, will carry out the desired behavior. Work has been done to provide robustness

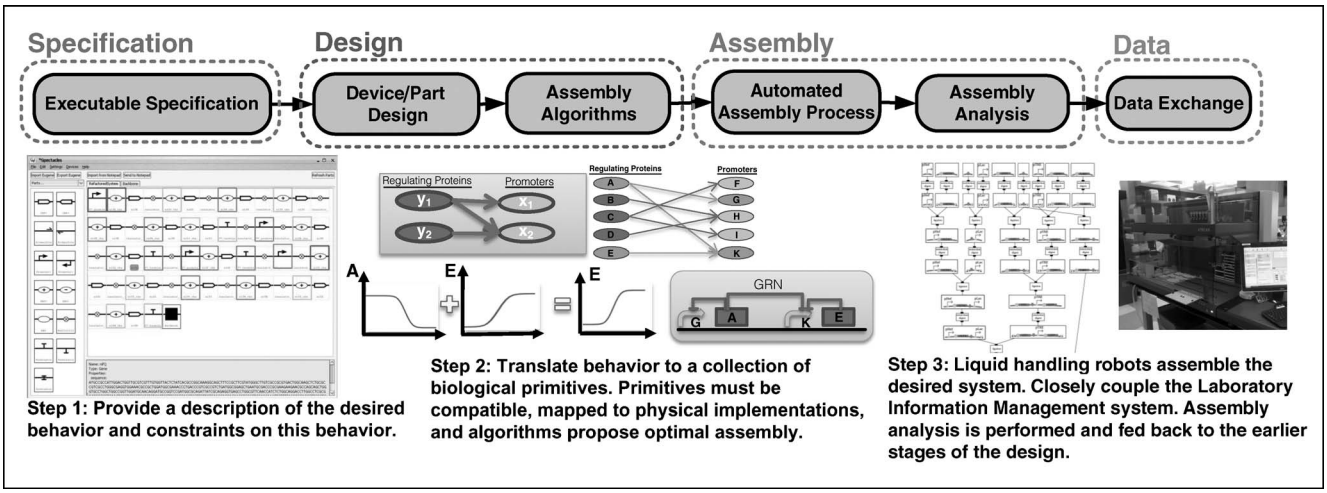


Figure 3. Bio-design automation for Part-based systems can be decomposed into specification, design, assembly, and data management stages. Tools are developed for each stage to satisfy specific optimization and constraint requirements. Workflows can be developed around these four areas to create synthetic biological systems starting from abstract specifications and ending with physically realized DNA constructs.

and reliability in these networks by introducing "retroactivity" [14] and biological network control feedback [15]. Once elements have been selected in the design stage, they can be retrieved from laboratory stock and physically assembled in the lab. Physical chemistry steps can be converted into liquid handling commands for robotics and optimized assembly strategies can minimize the time and cost of the assemblies [16]. Finally, the newly created Parts can be added back into the data management software along with characterization data on their performance once experiments have been carried out.

Metrics

Like electronics, there are an emerging number of metrics for design evaluation. Complexity is captured by the number of promoters, design length in base pairs, number of stages in the genetic circuit, fanout/fanin of transcriptional signals, and the number of individual DNA segments assembled in a single assembly step. Performance can be specified using Polymerase per second (PoPs) [17], which indicates the rate of mRNA transcription and florescence levels can be correlated to protein production. Tolerance to environmental factors (e.g., temperature, PH) characterize a Part's variability. Evolutionary resiliency against genetic mutations (e.g., point mutations, small DNA insertions and deletions) determine the circuit's reliability. Many other metrics exist and there is a movement toward "datasheets" for synthetic biological Parts [18].

Current tools and future challenges

Currently there are only a handful of BDA software tools that enable the design flow outlined in this paper for part-based synthetic biological systems. Figure 4 illustrates this space [19]–[25]. Data management tools enable locating specific Parts and examine relationships between Parts (e.g., regulatory relationships, physical sample tracking). Simulation tools validate functional system requirements. Design and assembly tools refine and constrain designs to enable their physical realization.

Engineering GRNs imposes unique challenges compared to designing electronic circuits. A wide variety of small molecules and proteins can be used to induce or repress transcription. These lead to a strong requirement of orthogonality to ensure correct genetic circuit operation. The concept of "crosstalk" is quite prevalent in synthetic biology, and spans multiple levels. For designs to function correctly, the impact of small molecules introduced and proteins produced by the system must be thoroughly understood.

The physical DNA which makes up the GRN can be acted on at any given location. For example in Figure 2, tf1 and tf2 can act anywhere on the DNA

concurrently. There is no linear requirement that ip1 and ip2 will be solely activated upon. The same is true of proteins. While the process of transcription and translation in Figure 2 was depicted left to right, GRNs operate massively in parallel and biological agents act on DNA both upstream and downstream.

Finally, proteins and small molecules degrade over time. Protein concentrations which at one stage were strong enough to repress or activate a promoter will fluctuate. Cell division also occurs. Biological designs must be resilient against their own biological processes.

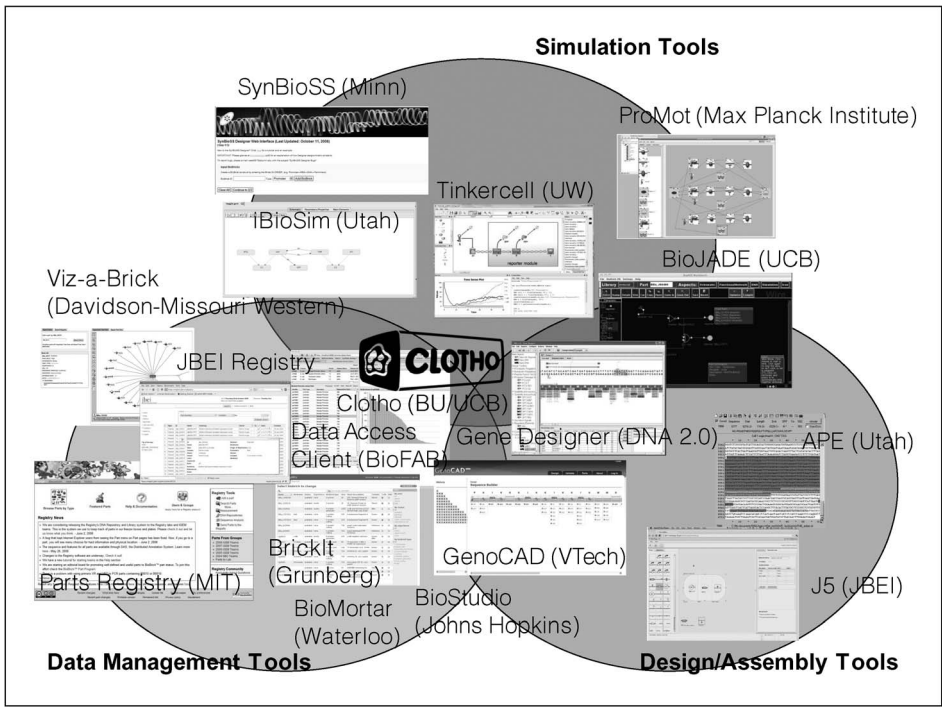


Figure 4. BDA tools for part-based synthetic biological systems can be classified into tools for data management, simulation, and design and assembly activities. Some approaches (e.g., Clotho) provide "App" based environments in which users can develop different tools which can span all of these areas. Other tools (e.g., Tinkercell) provide simulation frameworks where externally created biological process models can be imported. See [19]–[25].

Towards synthetic synthesis pathways

Unlike in human-made electronic systems where each module is designed to perform a distinct function, biological modules evolved over billions of years and exhibit high degrees of robustness and redundancy. To engineer cells to produce compounds that are non-native to the host cell, similar to adding functionality to an existing chip, or to enhance the production of a compound already produced within the cell, three distinct experimental approaches, independently or synergistically, are currently used. All three focus on engineering synthesis pathways.[1] A pathway refers to a series of enzyme-catalyzed chemical reactions that map substrate(s) to a product metabolite(s). It is implied that the pathway is stoichiometrically balanced, with equal number of atoms consumed and produced along the internal nodes of the pathway.

In the first approach, non-native pathways are embedded into a host organism. For example, pathways from Clostridium were embedded into *E. Coli* for the production of butanol [4]. In the second approach, one or more competing pathways are removed from a micro-organism to maximize production of a desired compound. Maximal ethanol production was achieved by removing undesired reactions using gene deletions [26]. In the third approach, existing pathways are modified by changing gene expression levels which in turn modify enzyme concentrations. A strain of *E. coli* was modified to produce fatty esters (biodiesel), fatty alcohols, and waxes directly from simple sugars [27]. The three approaches share common underlying concepts: a) individual pathways do not operate in isolation but in the context of other system components, and b) treating pathways, rather than individual reactions, as modular, functional units of cellular biosynthesis.

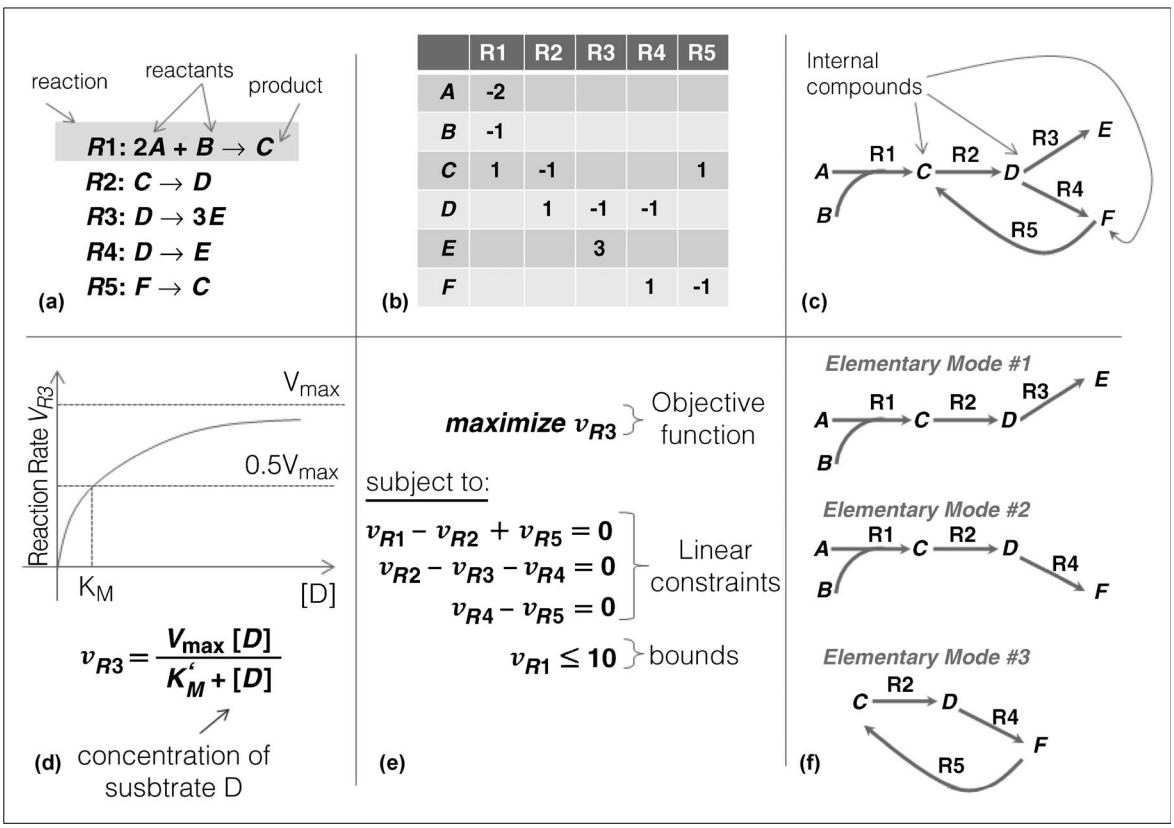
Pathway-based design methodologies are ad hoc, driven by intuition and domain expertise. The key conceptual steps however can be summarized as follows [28]. Once a particular compound is identified as a target, a suitable host is identified. If the compound is native to the host, then host

[1]A synthesis pathway produces compounds that are non-native to the host cell. The pathway may be native to another organism, or completely or partially synthetic.

modifications, such as knocking competing pathways or enhancing the activity of another, are pursued to optimize yield. If non-native, then a suitable synthesis pathway must be selected and evaluated in the context of the host cell. Pathway synthesis, evaluation, and host enhancement, however, are interdependent, and an iterative design cycle ensues. The relevant current computational tools presented here focus on system (host) analysis, pathway analysis, and pathway synthesis.

System analysis

A biochemical network represents a cellular process consisting of a set of reactions and compounds (Figure 5a). Reaction stoichiometry, invariant to the cell's operating conditions, specifies the relative number of atoms consumed or produced due to the chemical reaction. A biochemical network with m compounds and n reactions is represented using a $m \times n$ stoichiometric matrix N (Figure 5b). Each column describes a reaction. A column entry represents the stoichiometric coefficient of a compound participating in the relevant reaction. A column entry is zero if the compound does not participate in the reaction, positive if the compound is produced and negative if consumed. Reactions in a network can be classified as internal or exchange reactions linking a biochemical network to its external environment, as defined by the user and providing either uptake or production of external metabolites. Each row summarizes how a compound participates in various reactions. When utilizing the N matrix during analysis, typically only rows corresponding to internal compounds are included. The matrix can be viewed as a graph (see Figure 5c). Reactions maybe be reversible, and are sometimes split into



	R1	R2	R3	R4	R5
A	-2				
B	-1				
C	1	-1			1
D		1	-1	-1	
E			3		
F				1	-1

Figure 5. System analysis fundamentals. (a) The systems are modeled as a set of biochemical reactions. (b) Reaction stoichiometry is captured using a stoichiometric matrix. Zero entries are removed from the matrix for simplicity. (c) The network of reactions can be represented using a graph. (d) Example equation and graph for reaction rate as a function of substrate concentration. (e) Flux Balance Analysis example to maximize the flux of reaction R3. (f) Elementary modes for the network in Figure (c).

forward and reverse reactions during analysis. Only steady-state analysis, similar to DC analysis in circuits, is possible using a stoichiometric matrix.

Kinetic models of biochemical networks capture dynamic behaviors such as how fast a reaction occurs as a function of the relevant concentration. Kinetic models resemble RLC circuit models as they enable SPICE-like "transient analysis." Biochemical reactions, distinct from purely chemical reactions, experience saturation when catalysed by an enzyme. An example reaction rate as a function of substrate concentration is shown in Figure 5d, along with the simplest equation, in Michaelis–Menten form [29], used to describe enzyme kinetics. $V_{\max}$ is the maximum rate for a particular reaction that occurs at saturating substrate concentrations. The Michaelis constant K'_m is determined experimentally and represents the substrate concentration at which the reaction rate is half of $V_{\max}$. The system thus can be described by coupled ordinary differential equations (ODEs). However, parameters are often unknown, and the equation forms are best fits. Despite computational advances in parameter estimation, the size and complexity of biochemical networks reconstructed from genome databases have greatly increased over the years, rendering the estimation of kinetic or regulatory parameters, or fitting against *in vitro* experimental data, either impractical or outright infeasible. Often, steady-state analysis is the only means to analyze a biochemical system.

An interesting feature of biochemical networks is that they exhibit a large number of possible functional states, resulting in a great variety of phenotypes. At best, system biologists today can utilize known constraints, such as conservation of mass, energy and momentum, to *limit* possible functional states. The quintessential use of constraints occurs when using a technique called flux balance analysis (FBA) [30] to analyze flux distributions at steady state, when the net production and consumption rates are equal. Flux, the turnover rate of molecules associated with a reaction or pathway, resembles the flow of current in an electrical circuit. Flux for a particular reaction i, is typically denoted by v_i. Equivalent to Kirchhoff's current law, mass conservation at steady state declares that the rate of consumption and production of internal compounds must be equal for a particular metabolite (see Figure 5e). Specifying mass balance constraints for all internal compounds results in a set of linear equations. An objective function can be defined to correspond to maximizing the flux through a reaction leading to a desired target metabolite. For example, as shown in Figure 5e, specifying the uptake rate of $R1$ to be 10 and maximizing v_{R3}, results in v_{R3} equal to 10. There are, however, several flux distributions that maximize v_{R3} as the set of linear equations describing the system is underdetermined (fewer equations than unknowns). One possible distribution vector is $[10\ 15\ 10\ 5\ 5]^T$, which the entries corresponding to the flux in reactions 1 through 5. Another is $[10\ 10\ 10\ 0\ 0]^T$. In each case, the equations in Figure 5e are satisfied. Only lab measurements of flux values can verify the fluxes within the cell. This situation does not arise in dc analysis in circuits as systems are completely specified and each voltage and current value is uniquely determined. Constrained-based analysis have been used to analyze flux variability, flux coupling, and to identify optimal gene (reaction) knockout strategies. See [31] for a review.

Pathway analysis

Elementary flux mode (EFM) analysis is a pathway analysis technique that decomposes a biochemical network into an independent set of stoichiometrically balanced pathways called elementary flux modes (EFMs) [32]. When applied to the example in Figure 5c, the resulting three elementary modes are as illustrated in Figure 5g, and correspond to vectors $[1\ 1\ 1\ 0\ 0]^T$ $[1\ 1\ 0\ 1\ 0]^T$; $[0\ 1\ 0\ 1\ 1]^T$. A feasible flux distribution, such as $[10\ 15\ 10\ 5\ 5]^T$, can be expressed as a linear combination of the EFMs. Using weights 10, 0, and 5 for elementary modes 1, 2, and 3, respectively, we can write the distribution as the linear sum of $10 \times [1\ 1\ 1\ 0\ 0]^T + 0 \times [1\ 1\ 0\ 1\ 0]^T + 5 \times [0\ 1\ 0\ 1\ 1]^T$. EFM analysis exhaustively enumerates all stoichiometrically balanced pathways and cycles. Once all EFM are identified, they can be analyzed individually or within EFM families and used to make engineering decisions. Yield improvements can be obtained by enhancing enzyme activities along a particular pathway and eliminating competing pathways through gene knockouts, where a reaction is effectively eliminated from the network by suppressing the production of the catalyzing enzyme. For example, in Figure 5c, when maintaining an uptake rate for R1, suppressing the enzyme that catalyzes R4 will allow all D molecules to convert to

E molecules and not to F, thus enhancing the production of E. While increasingly sophisticated algorithms have been developed to generate EFMs (see [33] for a summary), including the canonical basis approach, the nullspace approach, and bit-pattern trees, the analysis remains computationally intractable for larger models, as the run-time scales exponentially with the complexity of the network.

From a microbe redesign perspective, not all pathways or elementary modes are of interest. Identifying a pathway of interest without exhaustive enumeration provides an excellent and familiar alternative, similar to shortest and longest delay analysis in timing analysis. The Dominant Edge algorithm [34] identifies a pathway containing the best thermodynamic bottleneck reaction, from a source metabolite to a destination metabolite using Gibbs free energy change as edge weights. Results for several tests cases indicated that thermodynamically feasible paths are either identical, a proper subset, or overlaps with EFMs. The Dominant-Edge algorithm can be utilized with flux values as edge weights to identify a path that contains the flux-limiting reaction, or to find the pathway with the least flux variability.

Pathway synthesis

Pathway synthesis is the process of identifying a series of reactions to form a pathway to produce a particular metabolite in a host organism. In some cases, the choice for a synthesis pathway is obvious. For example, there is only one known pathway for biosynthesis of 1,3-propanediol (a building block for synthetic polymers such as laminates and adhesives) from glycerol. This pathway consists of two reactions, each catalyzed by a singular enzyme. More generally, the number of alternative pathways for a given target may be too large for computational and experimental exploration, especially if the goal is to exploit the diversity of metabolic enzymes across many different organisms. A database such as the Kyoto Encyclopedia of Genes and Genomes (KEGG), which currently lists over 8000 reactions, must be searched to produce the final product molecule from one or more reactant metabolites in the host organism. The search process needs to take into account not only the main reactants, but also cofactors.

Because of the combinatorial nature of the problem, an exhaustive search for candidate pathways is impractical. Over the past several years, a number of heuristic approaches have been developed for particular applications (e.g., predicting novel pathways for degradation of xenobiotics or biosynthesis of native and nonnative compounds). One example approach is PathPred, a method to construct plausible reaction pathways based on the chemical structure transformation patterns of small molecules [35]. PathPred specifically exploits the KEGG RPAIR database, which contains biochemical structure transformation patterns for substrate product pairs (reactant pairs) of known enzymatic reactions. Another example approach is OptStrain, which uses mixed integer programming to identify stoichiometrically balanced pathways by adding or deleting reactions to selected host metabolic reaction networks [36]. A key advantage of this approach is to couple the selection of reactions with the ranking of the synthesis pathways in terms of theoretical yields. Success of the optimization however critically depends on thoroughly preprocessing the database, which remains a non-trivial task. Another method for constructing synthesis pathways utilizes a graph-based probabilistic-search approach and ranking the pathways using FBA [37]. This approach is promising as when compared to an exhaustive search enumerating all possible reaction routes consisting of 10 reaction steps, the search returned nearly identical distributions of maximal yields, while requiring far less computing time.

In the likely event that a large number of candidate pathways have been identified, the computational analysis needs to evaluate these pathway based on a performance metric such as maximal predicted yield once placed in the host system. The evaluation needs to also assess whether the introduction of the synthesis pathway will negatively impact the host organism's capacity for balanced growth. There currently is a lack of data and consensus on the best synthesis pathway scoring methods. The number of pathway steps does not necessarily correlate with yield or the implementation practicality. Another metric for ranking the non-native pathway is metabolic burden which computes the reduction in the growth rate as a result of added reactions. Thermodynamic feasibility which tries to compute the change in the Gibbs free energy of the reaction along the pathways is another possible ranking metric. Tighter integration between synthesis and evaluation, or precharacterizing the host could improve finding the optimal pathway.

Current tools and future challenges

While several point analysis and synthesis tools are available as described above, the analysis and synthesis at the system and pathway levels can benefit greatly from algorithmic improvements in terms of efficiency and prediction accuracy. Importantly, predictive models that capture complex biological behaviors will elucidate underlying biological principles and advance synthesis and reengineering practices. Building dynamic predictive models are of essence as steady state analysis has limited predictive capabilities. Within EDA, we clearly understand the value and limits of DC analysis and abstracted event driven simulations, and utilize detailed transient SPICE simulations as needed. One possible direction to build dynamic models is to exploit hierarchical modularity, an inherent organizational principle of biochemical networks, where larger, less cohesive clusters of network components comprise functionally distinct sub-clusters [38]. While there is general agreement that a biochemical module should represent a group of connected network components, and that the arrangement of modules in the network is hierarchical, there is less consensus on the criteria that should be used to systematically extract biologically meaningful modules. Uncovering the modularity of a biochemical network will allow system partitioning into minimally interdependent parts and will enable coarse-grained yet predictive models. The parameter estimation problem becomes simpler by substituting detailed reaction kinetics with less detailed module kinetics.

Pathway analysis using EFMs is computationally intractable. Computational methods based on statistical sampling, graph-based approaches, or more compact basis to represent the EFM solution space are possible. Efficient representation of EFMs in a BDD-like structure could improve average runtimes. Another profitable approach is to focus on the *enumeration objective* in lieu of enumeration to obtain results more efficiently. Integrating synthesis pathways (as well as synthetic GRN circuitry) within a system poses a metabolic burden and may compromise the cell's growth and evolutionary stability. Developing multiscale models and multiscale simulation methodologies that integrate regulatory and metabolic interactions will become necessary. Efficient impact prediction due to an added module will enhance pathway and GRN synthesis, and can be validated against more detailed models.

Conclusion

BDA today is a reminder of EDA in the 1960s, prior to Intel's first processor with only 2300 transistors. Synthetic biology's principled design methodology encompassing modularity, composition, standardization, and abstraction holds great promise to streamline engineering biology. Progress certainly hinges on further understanding biology. This article highlighted the state of BDA tools and design flows for designing synthetic biological circuits and pathways, and outlined computational challenges that span specifying desired biological behaviors to understanding biological systems. While several analogies can be drawn between BDA and EDA, challenges in BDA will require unique algorithmic solutions. Success will not be counted in number gene/promoter interactions or produced metabolites. The societal impact will be the metric. Will Design Automation work this time around?

Sidebar 1: Classic synthetic biology circuits

To provide some historical perspective, two "classic" genetic circuits are presented (see Figure 6). These circuits fundamentally changed the way in which engineers approached the design of genetic regulatory networks. Both were introduced in the year 2000 and ushered in a new era of genetic engineering.

The genetic toggle switch [39] is composed of two repressors and two promoters. Each promoter is inhibited by the repressor that is transcribed by the opposing promoter. A specific configuration of the toggle switch responds to the introduction of isopropyl-β-D-thiogalactopyranoside (IPTG) or a pulse of anhydrotetracycline (aTc). These small molecules are considered anti-repressors in the system (their presence enables the constitutive transcription of a promoter by disabling its repression). This device produces two stable "genetic states" and can be thought of as a primitive memory element. In the absence of either small molecule either stable state is possible (analogous to powering up electronic system state elements). In the presence of both small molecules, the behavior is undefined and subject to a number of competing biological factors (analogous to an SR latch). Memory elements are an important type of genetic device investigated by synthetic biologists.

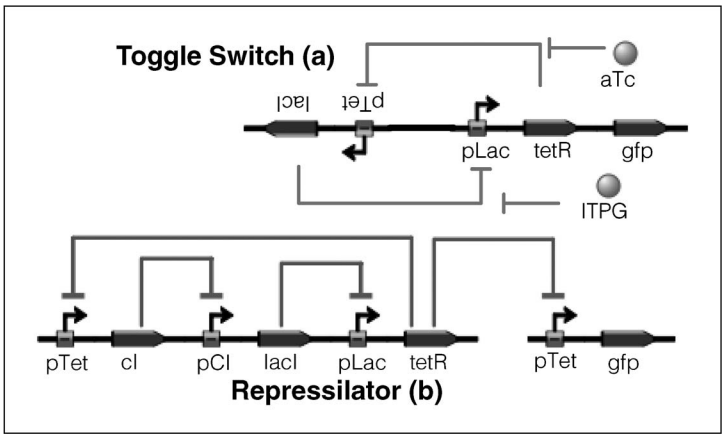


Figure 6. The genetic toggle switch (a) and the repressilator (b) represent two classic examples of genetic circuits. These circuits showed that engineering-based methodologies using primitive components could be applied to produce larger circuits with predictive behaviors.

Another seminal circuit is the repressilator [40]. Here a cascade of three genes each repressing each other produces oscillatory behavior. A green fluorescent protein acts as a periodic readout of the state in individual cells. The resulting oscillations, with typical periods of hours, are slower than the cell-division cycle, so the state of the oscillator has to be transmitted from generation to generation. Genetic oscillators can be used to recreate many of the rhythmic patterns found in nature or act as clocks to synchronize genetic systems.

Sidebar 2: IGEM and IWBDA

The International Genetically Engineered Machine Competition (iGEM) is the premier synthetic biology competition for undergraduate researchers. Teams of students are provided with access to biological parts (`partsregistry.org`) at the start of the summer. They are tasked with building novel biological systems to present at regional jamborees (Americas, Asia, and Europe). The best teams then compete at the world jamboree at MIT in late fall. Teams are awarded bronze, silver, and gold medals for completing predesignated requirements. Additionally, they compete for prizes for best wiki, presentation, engineered Part, natural Part, software tool, model, and human practices. A key aspect of the competition is not only its global nature (over 160 teams from all over the world) but also its requirement that teams contribute the designs they create back to the part registry at the conclusion of the competition. In this way the number of biological designs available to the community (and subsequent competitions) continues to grow. Winning teams have created colored pigment producing biosensors, bioenergy solutions, and heavy metal bioremediation. For more information see `igem.org`.

The International Workshop on Bio-Design Automation (IWBDA), founded in 2009 by Soha Hassoun, Douglas Densmore, and Marc Riedel, brings together researchers from the synthetic biology, systems biology, and design automation communities. The focus is on concepts, methodologies and software tools for the computational analysis and synthesis of biological systems. IWBDA has brought together over 430 researchers, 60 presentations, 55 posters, 12 keynote presentations, and three tutorial sessions since its introduction. In addition, it has hosted three Synthetic Biology Open Language (`sbolstandard.org`) meetings, and supported 40 sponsored students. For more information, see biodesignautomation.org.

Acknowledgment

The authors would like to thank Chris Voigt and Roza Ghamari for discussions on NOR-gate-based genetic regulatory networks. Genetic regulatory networks were drawn using Tinkercell [19]. In addition, this work reflects numerous discussions with Ron Weiss, Jonathan Babb, Jacob Beal, Aaron Adler, Fusun Yaman, Swapnil Bhatia, and Traci Haddock, and with Kyongbum Lee, Gautham Sridharan, Ehsan Ullah, and Mona Yousofshahi. Soha Hassoun gratefully acknowledges support by the National Science Foundation (under Grant 0829899).

References

[1] D.-K. Ro, E. M. Paradise, M. Ouellet, K. J. Fisher, K. L. Newman, J. M. Ndungu, K. A. Ho, R. A. Eachus, T. S. Ham, J. Kirby, M. C. Y. Chang, S. T. Withers, Y. Shiba, R. Sarpong, and J. D. Keasling. (2012, Apr.). Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature.* [Online]. *440(7086)*, pp. 940–943. Available: http://dx.doi.org/10.1038/nature04640

[2] J. C. Anderson, E. J. Clarke, A. P. Arkin, and C. A. Voigt. (2012, Apr.). Environmentally controlled invasion of cancer cells by engineered bacteria. *J. Mol.*

Biol. [Online]. *355(4)*, pp. 619–627. Available: http://dx.doi.org/10.1016/j.jmb.2005.10.076

[3] J. R. Kirby. (2012, Apr.). Synthetic biology: Designer bacteria degrades toxin. *Nature Chemical Biology.* [Online]. *6(6)*, pp. 398–399. Available: http://dx.doi.org/10.1038/nchembio.378

[4] S. Atsumi, T. Hanai, and J. C. Liao, ''Non-fermentative pathways for synthesis of branched-chain higher alcohols as biofuels,'' *Nature*, vol. 451, pp. 86–89, Jan. 2008.

[5] H. Salis, A. Tamsir, and C. Voigt, ''Engineering bacterial signals and sensors,'' *Contrib. Microbiol.* vol. 16, pp. 194–225, 2009.

[6] H. H. Wang, F. J. Isaacs, P. A. Carr, Z. Z. Sun, G. Xu, C. R. Forest, and G. M. Church. (2012, Apr.). Programming cells by multiplex genome engineering and accelerated evolution. *Nature.* [Online]. *460(7257)*, pp. 894–898. Available: http://dx.doi.org/10.1038/nature08187

[7] B. Palsson, *Systems Biology: Properties of Reconstructed Networks*. New York: Cambridge University Press, 2006.

[8] A. Tamsir, J. J. Tabor, and C. A. Voigt. (2012, Apr.). Robust multicellular computing using genetically encoded NOR gates and chemical/'wires/'. *Nature.* [Online]. *469(7329)*, pp. 212–215. Available: http://dx.doi.org/10.1038/nature09565

[9] J. C. Anderson, J. E. Dueber, M. Leguia, G. C. Wu, J. A. Goler, A. P. Arkin, and J. D. Keasling. (2012, Apr.). BglBricks: A flexible standard for biological part assembly. *J. Biol. Eng.* [Online]. *4(1)*, pp. 1+. Available: http://dx.doi.org/10.1186/1754-1611-4-1

[10] D. G. Gibson, L. Young, R.-Y. Y. Chuang, J. C. Venter, C. A. Hutchison, and H. O. Smith. (2012, Apr.). Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nature Meth.* [Online]. *6(5)*, pp. 343–345. Available: http://dx.doi.org/10.1038/nmeth.1318

[11] J. Beal, T. Lu, and R. Weiss. (2012, Apr.). Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS ONE.* [Online]. *6(8)*, pp. e22490+. Available: http://dx.doi.org/10.1371/journal.pone.0022490

[12] L. Bilitchenko, A. Liu, S. Cheung, E. Weeding, B. Xia, M. Leguia, J. C. Anderson, and D. Densmore. (2012, Apr.). Eugene A domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS ONE.* [Online]. *6(4)*, pp. e18882+. Available: http://dx.doi.org/10.1371/journal.pone.0018882

[13] M. Pedersen and A. Phillips. (2012, Apr.). Towards programming languages for genetic engineering of living cells. *J. Roy. Soc. Interface.* [Online]. *6(Suppl 4)*, pp. S437–S450. Available: http://dx.doi.org/10.1098/rsif.2008.0516.focus

[14] S. Jayanthi and D. D. Vecchio, ''Retroactivity attenuation in bio-molecular systems based on timescale separation,'' *IEEE Trans. Automat. Control*, vol. 56, no. 4, pp. 748–761, 2011.

[15] E. Franco and R. Murray, ''Design and performance of in vitro transcription rate regulatory circuits,'' in *Proc. 47th IEEE Conf. Dec. Contr., 2008. CDC 2008.*, Dec. 2008, pp. 161–166.

[16] D. Densmore, T. H. C. Hsiau, J. T. Kittleson, W. DeLoache, C. Batten, and J. C. Anderson. (2012, Apr.). Algorithms for automated DNA assembly. *Nucleic Acids Res.* [Online]. *38(8)*, pp. 2607–2616. Available: http://dx.doi.org/10.1093/nar/gkq165

[17] P. A. Varadarajan and D. Del Vecchio, ''Design and characterization of a three-terminal transcriptional device through polymerase per second,'' *IEEE Trans. Nanobiosci.*, vol. 8, pp. 281–289, Sep. 2009.

[18] B. Canton, A. Labno, and D. Endy. (2012, Apr.). Refinement and standardization of synthetic biological parts and devices. *Nature Biotechnol.* [Online]. *26(7)*, pp. 787–793. Available: http://dx.doi.org/10.1038/nbt1413

[19] D. Chandran, F. Bergmann, and H. Sauro. (2012, Apr.). TinkerCell: Modular CAD tool for synthetic biology. *J. Biol. Eng.* [Online]. *3(1)*, pp. 19+. Available: http://dx.doi.org/10.1186/1754-1611-3-19

[20] Y. N. Kaznessis. (2012, Apr.). SynBioSS-aided design of synthetic biological constructs. *Methods Enzymol.* [Online]. *498*, pp. 137–152. Available: http://dx.doi.org/10.1016/B978-0-12-385120-8.00006-1

[21] M. J. Czar, Y. Cai, and J. Peccoud. (2012, Apr.). Writing DNA with GenoCADTM. *Nucl. Acids Res.* [Online]. *37(suppl 2)*, pp. W40–W47. Available: http://dx.doi.org/10.1093/nar/gkp361

[22] B. Xia, S. Bhatia, B. Bubenheim, M. Dadgar, D. Densmore, and J. Anderson, ''Developer's and user's guide to clotho v2.0 a software platform for the creation of synthetic biological systems,'' *Methods Enzymol.*, vol. 498, 2011.

[23] C. J. Myers, N. Barker, K. Jones, H. Kuwahara, C. Madsen, and N.-P. P. Nguyen. (2012, Apr.). iBioSim: A tool for the analysis and design of genetic circuits. *Bioinformatics.* [Online]. *25(21)*, pp. 2848–2849. Available: http://dx.doi.org/10.1093/bioinformatics/btp457

[24] N. J. Hillson, R. D. Rosengarten, and J. D. Keasling. (2012, Apr.). j5 dna assembly design automation software. *ACS Synthetic Biol.* [Online]. *1(1)*, pp. 14–21. Available: http://pubs.acs.org/doi/abs/10.1021/sb2000116

[25] S. Mirschel, K. Steinmetz, M. Rempel, M. Ginkel, and E. D. Gilles. (2012, Apr.). Promot: Modular modeling for systems biology. *Bioinformatics.* [Online]. *25(5)*, pp. 687–689. Available: http://bioinformatics.oxfordjournals.org/content/25/5/687.abstract

[26] C. T. Trinh, P. Unrean, and F. Srienc, ''Minimal Escherichia coli cell for the most efficient production of ethanol from hexoses and pentoses,'' *Appl. Environ. Microbiol.*, vol. 74, pp. 3634–3643, Jun. 2008.

[27] E. J. Steen, Y. Kang, G. Bokinsky, Z. Hu, A. Schirmer, A. McClure, S. B. Del Cardayre, and J. D. Keasling, ''Microbial production of fatty-acid-derived fuels and chemicals from plant biomass,'' *Nature*, vol. 463, pp. 559–562, Jan. 2010.

[28] J. W. Lee, T. Y. Kim, Y. S. Jang, S. Choi, and S. Y. Lee, ''Systems metabolic engineering for chemicals and materials,'' *Trends Biotechnol.*, vol. 29, pp. 370–378, Aug. 2011.

[29] L. Michaelis, M. L. Menten, K. A. Johnson, and R. S. Goody. (2012, Apr.). The original Michaelis constant: Translation of the 1913 Michaelis-Menten paper. *Biochemistry.* [Online]. *50(39)*, pp. 8264–8269. Available: http://www.ncbi.nlm.nih.gov/pubmed/21888353

[30] D. A. Fell and J. Small, ''Fat synthesis in adipose tissue. An examination of stoichiometric constraints,'' *Biochem. J.*, vol. 238, no. 3, pp. 781–786, Sep. 1986.

[31] N. Price, J. Reed, and B. Palsson, ''Genome-scale models of microbial cells: Evaluating the consequences of constraints,'' *Nature Rev. Microbiol.*, vol. 2, no. 11, pp. 886–897, 2004.

[32] S. Schuster and C. Hlgetag. (2012, Apr.). On elementary flux modes in biochemical reaction systems at steady state. *J. Biol. Syst.* [Online]. *2(2)*, pp. 165–182. Available: http://wwwback.jacobs-university.de/imperia/md/content/groups/schools/ses/chilgetag/schusterhilgetagjbiolsyst1994.pdf

[33] M. Terzer, ''Large Scale Methods to Enumerate Extreme Rays and Elementary Modes,'' Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich, 2009.

[34] E. Ullah, K. Lee, and S. Hassoun, ''An algorithm for identifying dominant-edge metabolic pathways,'' in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Dig. Tech. Papers*, 2009, pp. 144–150.

[35] Y. Moriya, D. Shigemizu, M. Hattori, T. Tokimatsu, M. Kotera, S. Goto, and M. Kanehisa, ''PathPred: An enzyme-catalyzed metabolic pathway prediction server,'' *Nucleic Acids Res.*, vol. 38, pp. 138–143, Jul. 2010.

[36] P. Pharkya, A. P. Burgard, and C. D. Maranas. (2012, Apr.). Optstrain: A computational framework for redesign of microbial production systems. *Genome Research.* [Online]. *14(11)*, pp. 2367–2376. Available: http://genome.cshlp.org/content/14/11/2367.abstract

[37] M. Yousofshahi, K. Lee, and S. Hassoun, ''Probabilistic pathway construction,'' *Metab. Eng.*, vol. 13, no. 4, pp. 435–44, Jul. 2011.

[38] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabasi, ''Hierarchical organization of modularity in metabolic networks,'' *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[39] T. S. Gardner, C. R. Cantor, and J. J. Collins, ''Construction of a genetic toggle switch in Escherichia coli,'' *Nature*, vol. 403, no. 6767, pp. 339–42, Jan. 2000.

[40] M. B. Elowitz and S. Leibler. (2012, Apr.). A synthetic oscillatory network of transcriptional regulators. *Nature.* [Online]. *403(6767)*, pp. 335–338. Available: http://dx.doi.org/10.1038/35002125

Douglas Densmore is the Richard and Minda Reidy Family Career Development Assistant Professor in the Department of Electrical and Computer Engineering at Boston University. He received a PhD from the University of California, Berkeley. His research focuses on tools and automation for biological systems using techniques from electronic design automation. He is a member of IEEE.

Soha Hassoun is an Associate Professor at Tufts University in the Department of Computer Science and Electrical and Computer Engineering. She has a PhD from the University of Washington, Seattle. Her research spans both electronic design automation (EDA), and systems biology. She is interested in pathway analysis, modularity, pathway synthesis, and predictive modeling of biochemical networks. She is a senior member of the IEEE.

■ Direct questions and comments about this article to Douglas Densmore, Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215; dougd@bu.edu.