Synthetic Gene Circuits: Design with Directed Evolution

Eric L. Haseltine and Frances H. Arnold

Division of Chemistry and Chemical Engineering 210-41, California Institute of Technology, Pasadena, California 91125; email: frances@cheme.caltech.edu

Key Words

mathematical modeling, gene regulation, synthetic biology

Abstract

Synthetic circuits offer great promise for generating insights into nature's underlying design principles or forward engineering novel biotechnology applications. However, construction of these circuits is not straightforward. Synthetic circuits generally consist of components optimized to function in their natural context, not in the context of the synthetic circuit. Combining mathematical modeling with directed evolution offers one promising means for addressing this problem. Modeling identifies mutational targets and limits the evolutionary search space for directed evolution, which alters circuit performance without the need for detailed biophysical information. This review examines strategies for integrating modeling and directed evolution and discusses the utility and limitations of available methods.

Contents

INTRODUCTION

The synthetic gene circuits discipline can be described succinctly as novel regulation of pre-existing or engineered cellular functions. Consequently, synthetic gene circuits can be viewed as a subset of synthetic biology, the forward engineering of biological systems. The emergence of synthetic gene circuits as an engineering goal has re-emphasized the importance of accounting for the continuous nature of gene regulation. In this manner, it complements views prevalent, for example, in engineering of metabolic pathways where gene regulation is either ignored or treated as all-or-none in an effort to construct genome-wide models that assimilate large numbers of measurements (53). Synthetic gene circuits have also focused attention on the fact that regulation can be used to elicit specific re-

sponses to specific environmental stimuli (i.e., pH, temperature, exogenous signals, or signals from other cells). In this regard, they can coordinate specific functions to internal or external cues and execute precise instructions in a temporal or spatial fashion.

Synthetic gene circuits offer the promise of generating insights into nature's underlying design principles (38, 51), reconstructing and better understanding naturally occurring functions (56), and forward engineering novel biotechnology applications (45). Three exemplary contributions respectively demonstrate these promises:

- Pedraza & van Oudenaarden (49) investigated how noise propagates through a synthetic cascade of transcriptional repressors on the single-cell level. They discovered three sources for noise in gene expression: its own intrinsic fluctuations, transmitted noise from upstream genes, and global noise affecting all genes.

- Süel et al. (59) elucidated an ingenious noise-based mechanism that *Bacillus subtilis* uses to spontaneously and temporarily induce competence. They validated their model of this novel mechanism by constructing a synthetic circuit that permitted induction to, but not exit from, the competent state.

- Farmer & Liao (27) used synthetic circuits to couple expression of rate-controlling enzymes in lycopene synthesis to the presence of a metabolic precursor of lycopene. This regulation increased both lycopene production and the overall growth rate, emphasizing the importance of dynamic control in optimizing gene expression.

The dream of gene circuit designers is that well-characterized components can be easily assembled to achieve particular goals. This dream reflects the field's strong engineering influence. In traditional engineering disciplines, the concepts of standardization, decoupling, and abstraction have streamlined

the process that transforms ideas into physical realities (4, 25). Standardization ensures that individual working components can be used together or exchanged as needed. Decoupling represents a "divide and conquer" strategy in which complex problems are dissected into simpler subsystems. Abstraction pares down information so that attention is focused solely on those details relevant to the problem at hand. Applying these concepts should enable synthetic gene circuits to be conceived, designed, and implemented in a fashion similar to how an electrical engineer might design a computer or how a chemical engineer might construct a process flow sheet and ultimately a chemical plant. As a result, circuits constructed from standardized synthetic components are conceptually superior to naturally occurring circuits encoding the same functions because the latter are context dependent; that is, nature has optimized them in the context of both their natural function and cellular environment. In addition, synthetic gene circuits offer the possibility of implementing new functions and regulation not available to natural circuits.

The reality, however, is that the function of synthetic components and their assemblies also exhibits this context dependence. A gene circuit does not function outside the cellular context and may behave differently when the context changes. In fact, it can behave differently even in the same nominal context because of intrinsic and extrinsic noise. Mutation can routinely and suddenly alter circuit behavior in unexpected ways. In addition, circuits encounter variability owing to cell death, undefined and changing extracellular environments, and interactions with the cellular context. How these phenomena affect circuit function is not well understood and hinders the characterization of even the simplest components (4). Also, most tasks for which circuits are designed require precise, reproducible behavior for an entire population of cells, necessitating coordination at the population level. Simply stated, details matter, and we don't understand the details.

In light of context dependence, what are the strategies one can use to take a synthetic gene circuit from an initial design to an effective physical implementation? Once the initial design is implemented, relatively few strategies exist for altering the behavior of these circuits. Directed evolution guided by mathematical modeling is a powerful method for altering circuit performance. Here, we review the utility and limitations of directed evolution to alter circuit performance and construct novel regulatory components. We then consider how mathematical modeling and systems theory tools (e.g., sensitivity analysis) can suggest specific manipulations, thereby targeting the directed evolution. Finally, we present conclusions and discuss future directions and opportunities for growth in this field.

PROPOSED STRATEGIES FOR MAKING FUNCTIONAL SYNTHETIC CIRCUITS

Approaches for generating a conceptual design of a synthetic circuit include a modular approach, as championed by the engineering community (4, 25, 65), and evolutionary design (29). We define a conceptual design as specifying how individual components (e.g., sensors, regulatory factors, outputs) are connected to accomplish the desired circuit function. The next step is constructing the functional circuit. Surprisingly, there exist relatively few strategies for this process given the ever-increasing number of published synthetic circuits, suggesting that genetic circuit construction is currently more of an art form than a well-established engineering discipline. In this section, we review the "plug and play" strategy, the "design, then mutate" strategy, and a hybrid of the two.

"Plug and Play" Strategy

Using the engineering principle of decoupling, circuits can be divided into simpler subproblems consisting of different components.

One could then envision exchanging these components from a well-characterized library to effectively alter the circuit performance. Consider, for example, the toggle switch design of Gardner et al. (33), a mutually inhibitory network consisting of two repressors. A functional toggle switch requires proper matching of the repressor characteristics to ensure that the region of bistability falls within the operating regime so that transient perturbations can toggle the switch from one operating point to the other. In an ideal world, one could simply choose two such repressors from a well-characterized library (for example, the MIT Parts Registry, **http://parts.mit.edu/**) and construct a functional toggle switch. Most circuit constructs to date have been implemented using this approach. However, the difficulties of this approach are not always apparent from the published literature, which does not include the vast majority of the many failures.

Several practical considerations currently limit the utility of this approach. First, design considerations often restrict the pool from which one can select components. In the toggle switch example, one may desire to maintain the same switching effectors (thermal and chemical) that Gardner et al. (33) chose. Second, the function of each component is context dependent; different components may exhibit different interactions and/or sensitivities to inputs when taken out of their natural contexts. Finally, current libraries of well-characterized components are sparse. For example, as of the submission of this review, the MIT Parts Registry currently reports only around 40 repressors.

"Design, Then Mutate" Strategy

An alternative to "plug and play" is "design, then mutate." Here, one a priori selects the components that comprise the designed circuit. Rather than swapping out different components, directed evolution is used to manipulate the behavior of these components, and the desired phenotype is obtained via screen-

ing or selection. That directed evolution is an appropriate tool for tuning circuits was recognized early in the emergence of the synthetic gene circuit discipline (44), and the utility of this approach has been demonstrated experimentally via random mutagenesis and screening for functional mutants (66). However, implementation of this strategy must be done thoughtfully because the evolutionary search is restricted to the number of mutants that can be screened or selected at each generation, usually 10^3 to 10^8. Genetic circuits are composed of multiple genes. The number of ways to mutate these genes grows rapidly with the size of the system, and the probability that a random mutation will confer the desired functionality is correspondingly small.

Feng et al. (28) proposed to overcome this problem with mathematical modeling of the gene circuit. A flowchart of this strategy is presented in **Figure 1a**. Here, global sensitivity analysis interrogates a model of the circuit to identify targets for mutation. A mutational target is then selected and modified using directed evolution, screening for the desired overall circuit behavior. Identifying a target restricts the evolutionary search space and increases the likelihood that a functional circuit will be screened (provided the right target is identified). However, there are several conceptual problems with this strategy. First, what happens when a single iteration still yields a nonfunctional circuit? One must then decide whether to try the next mutational target or attempt to figure out what went wrong. One possible explanation is that the mathematical model is incorrect and that the identified mutational target does not alter the circuit performance in the desired way. In this case, trusting the model to provide a second mutational target is not justified, and the above procedure has not provided much additional data with which to refine the model. A second possibility is that the directed evolution has not correctly or sufficiently perturbed the mutational target. Here, debugging of the directed evolution may be hindered by the complexity of the genetic circuit. Finally, if
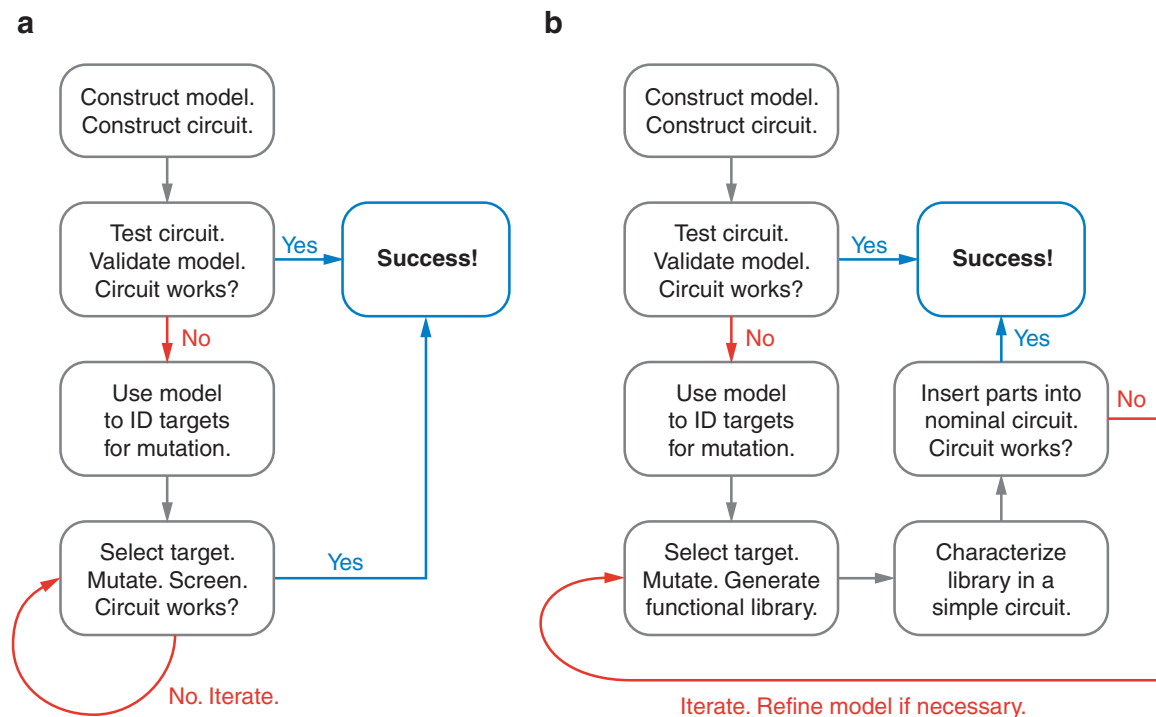
a

```
Construct model.
Construct circuit.
        │
        ▼
Test circuit.        ── Yes ──►   Success!
Validate model.
Circuit works?
        │ No
        ▼
Use model
to ID targets
for mutation.
        │
        ▼
Select target.       ── Yes ──►   (to Success!)
Mutate. Screen.
Circuit works?
        ◄── No. Iterate.
```

b

```
Construct model.
Construct circuit.
        │
        ▼
Test circuit.        ── Yes ──►   Success!
Validate model.                       ▲ Yes
Circuit works?                        │
        │ No                  Insert parts into   ── No ──┐
        ▼                     nominal circuit.            │
Use model               ──►   Circuit works?              │
to ID targets                       ▲                     │
for mutation.                       │                     │
        │                     Characterize               │
        ▼                     library in a               │
Select target.   ──►          simple circuit.            │
Mutate. Generate                                          │
functional library.  ◄────────────────────────────────────┘
        ◄── Iterate. Refine model if necessary.
```

Figure 1

Flowchart for designing and implementing genetic circuits. (*a*) "Design, then mutate" strategy and
(*b*) combining the "plug and play" and "design, then mutate" strategies.

the model indicates that multiple targets must
be perturbed to make the circuit functional,
then the concern again arises that the evo-
lutionary search space may be prohibitively
large, reducing the likelihood that a random
search will produce a functional circuit.

Combining the "Plug and Play" and "Design, Then Mutate" Strategies

Given these problems, screening for function
using only the nominal circuit may not be
the best strategy. An insightful work by Alper
et al. (2) suggests an alternative approach: sys-
tematically perturbing and characterizing in-
dividual components (in effect screening on
the level of a simple circuit), and then us-
ing these perturbed components to gain ad-
ditional insight into the functionality of the
nominal circuit. In this work, the transcrip-
tional strength of the constitutive P_L-λ pro-

moter was perturbed by directed evolution.
Selected promoter mutants were systemati-
cally characterized using two simple circuits.
In the first circuit, green fluorescent protein
was constitutively expressed from the pro-
moter, and the circuits were characterized us-
ing both flow cytometry to measure protein
expression levels and reverse transcriptase-
polymerase chain reaction to measure mRNA
transcription levels. In the second circuit, con-
stitutive expression of an antibiotic resistance
gene permitted measurement of the minimum
inhibitory concentration required for antibi-
otic resistance. Using these results, Alper et al.
(2) selected a set of constitutive promoters
with well-characterized strengths. They sub-
sequently employed these promoter mutants
to study more complex circuits, including op-
timizing expression of enzymes required for
lycopene production. Here, as the promoter
strength of mutants increased, lycopene

production either increased monotonically or exhibited a maximum, depending on the genetic configuration.

The underlying concept is that the nominal circuit should be characterized by systematically perturbing a single component (corresponding to a single model parameter). We briefly outline this approach, which is presented as a flowchart in **Figure 1b**. (Step 1) Implement and experimentally characterize the synthetic circuit. (Step 2) Estimate model parameters by optimally fitting the model to experimental data. Given these estimates, model analysis will prioritize targets that can generate the desired functional behavior with minimal manipulation. (Step 3) Create a library of targeted components with a range of responses via directed evolution and/or rational design. Screening and characterization of the library should be performed in a simple circuit rather than the nominal circuit. (Step 4) Incorporate the members of the resulting library into the nominal circuit and check each individual circuit for functional behavior. If the circuit does not function as desired, refine the model and return to Step 2.

The benefits of this approach are fourfold. First, some circuit properties such as oscillations and population-level responses are not suitable for high-throughput screens or selections, so the directed evolution must be performed in the context of a simple circuit. Second, performing directed evolution in the context of a simple circuit ensures that the resulting library covers a wide range of parameter values and facilitates troubleshooting. Third, the generated library consists of interchangeable components that (*a*) enable "plug and play" alteration of the circuit by simply exchanging one component for another and that (*b*) can be applied readily to other genetic circuits. The underlying assumption is that mutants of a component are locally perturbed and have a greater likelihood of retaining the same circuit and cellular contexts than do components taken from different contexts. Fourth, examining the circuit behavior over a library of components with widely varying parameter values facilitates model validation and refinement. Although one could implement the nominal circuit using only the component predicted to yield functional behavior, such implementation presumes that the model is accurate. Most likely, there will be some discrepancy between the model and the experiment. Checking the nominal circuit for functional behavior with components spanning a range of parameter values is useful for overcoming such discrepancies.

A potential problem of this strategy is that certain circuit properties may not be independent. Changing one property often affects another. For example, changing stability often affects expression level. A possible solution to avoid this problem is to screen or select for multiple properties simultaneously. Recent works (18, 50) have demonstrated successful selection for multiple properties (see below). The success of this method depends on the ability to design a screen or selection that satisfactorily reports on all properties of interest.

Also, the parameter space accessible by DNA mutation is not well understood. For example, Mayo et al. (43) demonstrated how the design of the *lac* operon transforms a uniformly distributed parameter space into a non-uniform phenotype space (i.e., the circuit function). That is, the circuit design limits the set of possible phenotypes, and some of the possible phenotypes are more likely than others. Interestingly, the wild-type *lac* operon and 16 mutants (one to seven nucleotides difference) fall in a sparsely populated region of the phenotype space. These results suggest that the fitness landscape for DNA sequences that encode functions significantly biases the parameter space.

USING DIRECTED EVOLUTION TO PERTURB CIRCUITS

Synthetic gene circuits almost always require fine-tuning for proper function. A brief look at the experimental methods used to construct functional circuits illustrates this point.

In some cases, synthetic gene circuits appear to work by assembling natural (pre-existing) components (9, 10, 40, 41, 67); however, the experimental methods outlined in these works may not reflect all the steps actually performed to constitute the circuits. Other projects have applied various technologies to tune circuit performance, including (*a*) engineering of promoters via construction of novel hybrid promoters (e.g., to include repression) (6, 8, 36) and alteration of existing promoters to adjust transcription rates and transcription factor binding (14, 33, 36, 55); (*b*) alteration of proteins via use of *ssrA* tags to increase protein degradation rates (7, 8, 24, 31), codon optimization (7), and recombination (39); (*c*) manipulation of ribosome-binding sites (RBSs) to alter translation rates (8, 33, 64); (*d*) elimination or rewiring of native metabolic functions (14, 27, 31); and (*e*) random recombination of regulatory and output functions (35).

However, these methods for perturbing circuit function are tedious because they make only a few changes and then check to see if the circuit works. If the desired behavior can be screened in a high-throughput fashion or selected, it is significantly more efficient to make many altered circuits and to screen or select for desired behavior, accumulating mutations over multiple generations, i.e., perform directed evolution. High-throughput screening usually examines between 10^3 and 10^6 clones per generation and (ideally) directly examines the desired phenotype. Genetic selections can examine up to 10^8 clones but require that the desired phenotype be linked to cell survival. Because the goal of a synthetic gene circuit by definition is to regulate a cellular function, selection is often a natural choice for identifying desired phenotypes. In this case, the regulatory function of the circuit is coupled to cell survival, for example, by means of an antibiotic resistance gene, to achieve the evolutionary pressure required for an effective selection.

While screening and selection methods can potentially test large numbers of mutants for functional behavior, these numbers pale in comparison to the astronomically large number of mutants in a given sequence space. Consider, for example, using directed evolution to alter the properties of a relatively small protein consisting of one hundred amino acid residues. The number of proteins possible for this one sequence length is $20^{100} \approx 10^{130}$, a space that could never be explored exhaustively. In addition, the number of sequences expected to encode a solution to a specific problem is an infinitesimal fraction of this space. So how does directed evolution work to optimize circuit performance and generate new functions?

One reason evolution (and directed evolution) works is that the space is highly dimensional: There are many ways to mutate even a single small protein. There are many more ways to mutate a circuit. If some of these pathways enhance the desired performance, then iterative rounds of random mutation of the circuit and screening for the desired properties may provide steady improvements in function. The key is that the process is evolutionary: Small changes accumulated over multiple generations can lead to a good circuit implementation. This only works if the initial design is close to (i.e., within a few mutations of) a better design. Because biological components and systems are the products of exactly such an evolutionary search, they are apparently easily tuned in the same fashion.

Important to this search strategy are the ideas of protein promiscuity and evolutionary plasticity (sometimes called evolvability). Proteins that perform a native function often also perform numerous promiscuous functions, and small numbers of mutations can induce large changes in promiscuous functions (i.e., plasticity) often while having little effect on the native function (1, 62). This theory justifies the standard practice of choosing a protein with a function closely related to the desired function as the starting point for directed evolution. In addition, if residues endowing plasticity are known, then this trait can be exploited by targeting mutagenesis to specific residues (e.g., inserting each amino acid

into the targeted sites) to perturb the protein function.

Directed evolution is a relatively mature field. We refer the interested reader to reviews discussing methods for generating molecular diversity (12) and screening for desired phenotypes (16, 46). Instead, we focus on the role that directed evolution serves in engineering synthetic gene circuits, first in tuning the performance of synthetic circuits and second in engineering novel regulatory functions. We also assess the promise and limitations of directed evolution for circuit engineering.

Altering the Performance of a Circuit

Directed evolution is a powerful method for altering circuit performance; however, it does have some limitations. To investigate these limitations further, we review works that have tuned circuits by employing directed evolution to identify strategies for selecting mutational targets.

Targeting functionally rich segments of the DNA is critical for successfully altering circuit function, since the entire circuit sequence space is too large to explore except in the most cursory fashion. One such rich segment for circuit function is the RBS, which dictates protein expression levels. For example, the codons between positions -7 and -13 in the 5′ to AUG initiation of mRNA appear to cause most of the variation in RBS strength in *Escherichia coli* (54). The size of the resulting library containing all possible variants is $4^7 = 16,384$, a size well within the capability of either screens or selections to cover completely. The design of a genetic circuit typically dictates the function; nonfunctional circuits are often the result of improperly matched expression levels rather than a flaw in the basic design. For example, Anderson et al. (3) used random mutagenesis of the RBS to engineer *E. coli* to invade cancer cells. The bacteria were programmed to express the gene *invasin*, which facilitates adhesion and invasion of mammalian cells expressing β1-integrins, under induced, hypoxic, and density-dependent

conditions. Because the last two environmental conditions indicate that the bacteria have localized in the tumor, the circuit thus targets gene expression only in the tumor. By constructing appropriate RBS libraries to tune *invasin* expression (see **Figure 2a**), Anderson et al. (3) generated circuits that enable bacteria to invade cancer cells under these conditions.

Targeting the RBS for mutation is an effective strategy for tuning circuit behavior for another reason: RBS mutations nominally affect only the expression level of a protein. Protein transcription factors, on the other hand, tend to encompass numerous functions (for example, DNA binding and activation or repression). Point mutations to such proteins may alter these functions in addition to gene expression.

Another feature of mutagenesis is that perturbed circuit components are more likely to be destabilized or detuned than to exhibit improvements in function. For proteins, the fraction of random mutants retaining wild-type function decreases exponentially with increasing numbers of amino acid substitutions (13, 22, 60). An effective strategy for choosing mutational targets is to select targets that yield functional behavior when detuned. Yokobayashi et al. (66) used this strategy to correct a nonfunctional inverter circuit. In this circuit, LacI protein was constitutively expressed and repressed expression of cI. cI in turn repressed expression of the output, a fluorescent protein. As a result, the circuit should naturally fluoresce, and external addition of isopropyl β-D-thiogalactoside, an inducer that interferes with LacI function, should reduce fluorescence. However, the nominal circuit exhibited no fluorescence even in the absence of IPTG. An analysis of a mathematical model describing the inverter indicated that decreasing either the cI repressor operator affinity, RBS strength, or dimerization could generate functional circuits (see **Figure 2b**). Yokobayashi et al. (66) randomly mutagenized the *cI* gene and its RBS. Numerous mutations turned out to yield functional

a

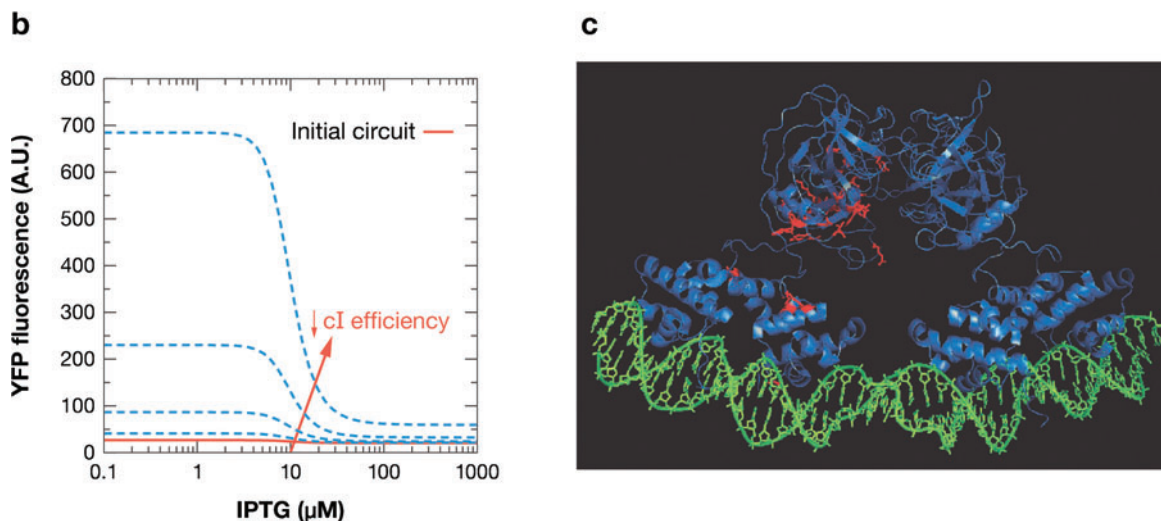|		
Wild type	T T T C A T T T A A A T T a t g a t g g t t	(not inducible)
RBS library	N N N G G A G N N N N N N r t g n t g g t t	
FdhInv	A A A G G A G T A A A A A a t g a t g g t t	(inducible)
AraInv	C C C G G A G C C C C C C a t g c t g g t t	(inducible)

b

c

Figure 2

Strategies for applying directed evolution. (*a*) Targeting ribosome-binding sites (RBSs) and (*b*, *c*) detuning parameters. (*a*) Mutating the wild-type RBS (N = A, T, C, or G; R = A or G) tuned the invasion of cancer cells by *E. coli* for anaerobic induction (FdhInv) and arabinose induction (AraInv). The nominal circuits exhibited undetectable levels of invasion under induced conditions, whereas the tuned circuits demonstrated measurable invasion. (*b*, *c*) The efficacy of detuning parameters to adjust circuit function is demonstrated. The model predicted that decreasing the dimerization constant, the repressor/operator affinity, and the RBS efficiency of cI could generate functional inverter behavior as shown in panel *b*. Panel *c* shows the cI tetramer-DNA model with the side chains of the mutated amino acid residues (*red*) found in the 18 evolved (functional) inverter circuits.

circuits (**Figure 2c**), many of which led to premature protein truncation.

Using a simple circuit to screen for a desired phenotype is another useful strategy because it (*a*) reduces the complexity of the screen and (*b*) is useful in debugging problems in directed evolution. This strategy was implicitly used by Basu et al. (7) to program spatial pattern formation by re-engineering the *lux* quorum-sensing regulatory module, a cell-cell communication system. Here, a pop-

ulation of sender cells generated a spatial gradient of an acyl-homoserine lactone (acyl-HSL), a chemical signal that freely diffuses between bacteria. Receiver cells differentiated to form spatial patterns based on the concentration of signal received via three bandpass constructs, which expressed fluorescent markers only for acyl-HSL concentrations within a certain range. Three different constructs exhibited bandpass behavior for ranges corresponding to low, mid, and high acyl-HSL

concentrations. Wild-type *luxR* and a hypersensitive mutant were used to construct the mid- and low-detect bandpasses required for spatial pattern formation. Collins et al. (17) constructed the hypersensitive mutant by randomly mutagenizing the acyl-HSL signal-binding domain of *luxR* and subsequently screening for gene activation in a simple circuit. Screening in the nominal bandpass circuit would have been significantly more difficult because it would have required screening for behavior at a wide range of acyl-HSL concentrations to confirm bandpass behavior.

Engineering Novel Regulation

Another benefit of directed evolution is the ability to construct novel regulatory functions altogether. For example, recent work has focused on engineering the specificity of transcriptional regulators (32). Two effective strategies for engineering novel regulation include selecting for orthogonal properties and exploiting modularity.

Engineering orthogonal properties. A powerful capability of directed evolution is the ability to select for multiple orthogonal phenotypes. We define orthogonal in this context to mean that the phenotypes are well separated, e.g., one variant exhibits function A but not function B, while another might exhibit function B but not A (see **Figure 3a**). Rackham & Chin (50) evolved orthogonal ribosome-mRNA pairs that selectively translated mRNAs with specific RBSs not recognized by the native *E. coli* ribosomes. The pairs were generated by first selecting RBSs not recognized by the native ribosomes and then selecting for mutant 16S ribosomal RNA that recognized the nonnatural RBSs. Collins et al. (18) evolved orthogonal transcriptional activators by refocusing the specificity of the wild-type *luxR* to a noncognate signal. This specificity was obtained by first selecting mutants that responded to the desired noncognate signal and then selecting from the surviving mutants

those that did not respond to the cognate signal. Other works have re-engineered nuclear hormone receptors (NHRs) to be specific and orthogonal to naturally occurring NHRs [see the recent review by Chockalingam et al. (15)].

Exploiting modularity to engineer novel functions. The modularity of regulatory proteins offers the tantalizing prospect of generating novel input-output relationships by simply recombining domains from different regulators (11). **Figure 3b** presents a simple example of such a swap, illustrating how recombination can generate two novel input-output pairings from two existing pairings. For example, the diverse array of eukaryotic signaling proteins is highly modular in that the protein's input (signal) and output (catalytic function) appear to be physically and functionally separable. Bhattacharyya et al. (11) recently reviewed rewiring of eukaryotic regulation via recombination. The zinc-finger domain presents another possible scaffold for engineering novel DNA-binding proteins in eukaryotes (19). A single zinc finger typically recognizes only three DNA bases, but linking multiple fingers can lead to recognition of longer sequences. Such modularity theoretically permits generating DNA-binding proteins of arbitrary sequence specificities via fusion of premade fingers recognizing any one of the 64 possible DNA triplets. Papworth et al. (47) reviewed recent progress in this field by both rational design and selection strategies.

Modularity is a feature of certain prokaryotic regulatory proteins as well. Native membrane-bound sensors in bacteria are typically modular and consist of a membrane-bound, extracellular sensor and an intracellular response regulator. Levskaya et al. (42) exploited such modularity to create a synthetic phytochrome by fusing a cyanobacterial photoreceptor to an *E. coli* intracellular histidine kinase domain. By importing this novel gene along with genes required to produce the photoreceptor, they engineered
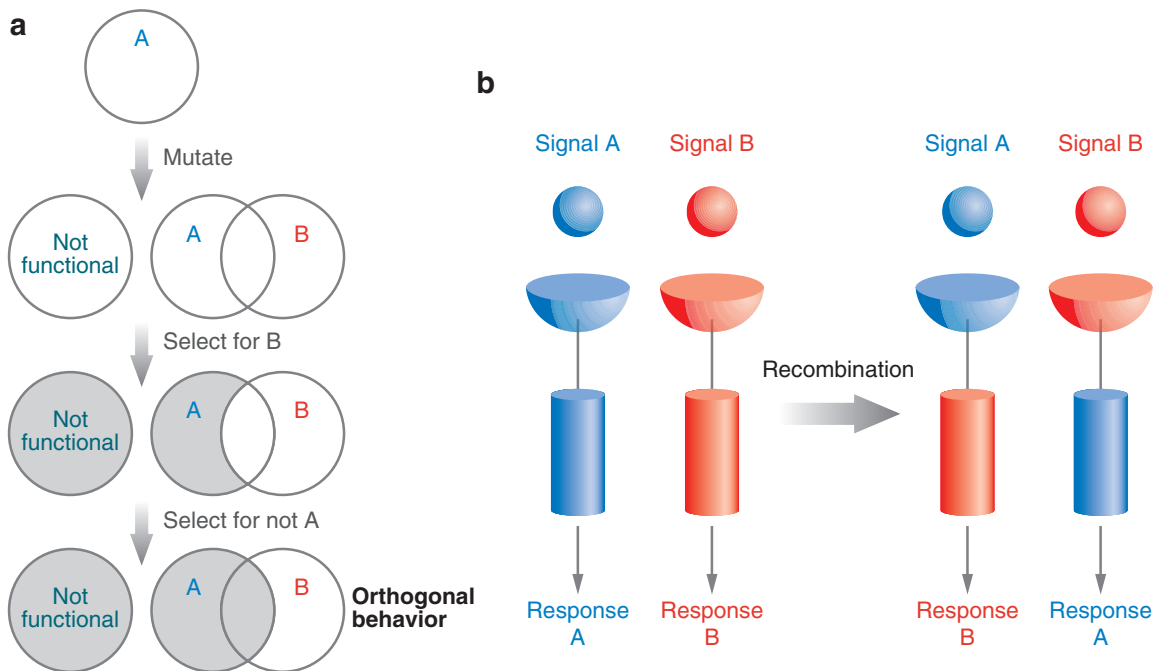
Figure 3

Two effective strategies for engineering novel regulation using directed evolution include (*a*) selecting for orthogonal properties and (*b*) exploiting modularity. Selecting for orthogonal properties (*a*) first involves generating diversity through, for example, random mutagenesis. Thus, a parent starting with function A could yield mutants that are nonfunctional, retain function A, acquire a new function B, or acquire a new function B while retaining function A. To ensure orthogonality of functions, two selections must be performed: first, selection for function B, and then selection for not retaining function A. Extrapolation of this logic to engineering orthogonality of multiple functions is straightforward. Exploiting the modularity of regulatory proteins (*b*) permits synthesis of new proteins with altered input-output behavior. For example, two regulatory proteins, each consisting of two domains, a signaling receptor, and an output effector, could theoretically permit swapping of input-output behavior.

E. coli to photograph light. The modularity of the LuxR protein, whose signal-specificity and DNA-binding functions are encoded in separate domains, has been exploited to create a variant that responds to a new signal by activating at a new promoter (C.A. Tracewell, C.H. Collins & F.H. Arnold, manuscript in preparation). This modularity allows the subfunctions to evolve separately and then recombine to efficiently create novel, well-characterized, signal-dependent transcriptional activators with a wide range of behaviors that can be used interchangeably in other circuits. Guntas et al. (37) used directed evolution to fuse a maltose-binding protein to a β-lactamase, effectively building modu-

larity. This novel protein switch required maltose to hydrolyze β-lactam.

THE ROLE OF MATHEMATICAL MODELING

In the previous section, we have demonstrated the utility and limitations of using directed evolution to perturb gene circuits. In particular, the three primary limitations are that (*a*) the evolutionary search space for a genetic circuit composed of many genes is generally too large to explore efficiently; (*b*) detuning parameters (reducing function) is much easier than improving function; and (*c*) although selecting for independent properties is possible,

it usually requires setting up multiple rounds of screening or selection. Using mathematical models to suggest mutational targets can help overcome each of these limitations. Identifying mutational targets in this way makes the evolutionary search reasonable by exploiting the modularity of the circuit design to divide the original problem into simpler subproblems, thus overcoming the first limitation. Other problems of comparable size (e.g., evolution of a multiprotein assembly) cannot be subdivided in this way unless they are comparably modular and the interactions between modules can be described with some accuracy. Biasing selection of the mutational targets suggested by the model can account for the second and third limitations (i.e., choosing targets that require detuning or independent targets such as RBSs).

We now focus more in depth on how to use mathematical models to identify mutational targets for directed evolution. Mathematical models are analogous to a road map in that they contain a concise abstraction of the local landscape. As with planning any excursion, it is critical to know both the current location and the location of the desired destination. Continuing in this analogy, parameter estimation can determine the current location, bifurcation analysis is useful in identifying the desired destination (robust, stable, or oscillatory states, for example), and sensitivity analysis and optimization serve as a local compass (pointing out which directions are easiest to travel) and as a trip optimizer (the shortest distance to the final destination given trip constraints), respectively. Two types of mathematical models have proven particularly useful in describing synthetic gene circuits: deterministic and stochastic reaction models.

Overview of Deterministic Methods for Identifying Mutational Targets

In a deterministic setting, mass balances are used to determine evolution equations for the quantities of interest. For this type of model, the combination of parameters and initial conditions uniquely determines the system evolution. The most commonly used assumptions are that the system is well mixed and that it is segregated by time, leading to models generally of the form

$$\frac{d(NV)}{dt} = F(N, t; \theta), \qquad 1.$$

in which N is a vector of the reacting component concentrations, V is the system volume, θ is a vector of the model parameters, and $F(N, t; \theta)$ is a vector containing the production rates and accumulation rates due to transport of components into and out of the system.

Models governed by Equation 1 are simulated dynamically or analyzed for steady-state behavior. Dynamic simulation (or time integration) is generally straightforward, and many numerical software packages such as MATLAB and Octave contain tools for solving models governed by Equation 1. Such simulation has been used, for example, to examine how cells might spatially control pulses of gene expression (8) or regulate population density (67). The steady states of Equation 1 are determined by solving for the reacting components when the time derivatives are set to zero, i.e.,

$$0 = F(N, t; \theta). \qquad 2.$$

Continuation methods offer a powerful means for mapping steady-state circuit performance as a function of key parameters. Bifurcation analysis complements continuation methods by examining how changes in the parameter values affect the stability of the steady states; Strogatz (58) provides a nice introductory overview of this topic. Packages such as AUTO (21) and CL_MATCONT (20) combine both of these functionalities. Excellent examples of using bifurcation analysis to analyze synthetic gene circuits include determination of regions of bistability for constructing a genetic toggle switch (33) and calculation of regions generating sustained oscillations (24, 31).

Parameter estimation. Once a model describing the system of interest has been constructed, the next step is to use parameter estimation to identify the exact location in parameter space. This problem can be formulated as an optimization problem,

$$\min_{\theta} \Phi(\boldsymbol{\theta}), \qquad 3.$$

subject to model predictions of the experimental measurements. The goal is to minimize the scalar objective function $\Phi(\boldsymbol{\theta})$ by adjusting the parameters $\boldsymbol{\theta}$, thereby reconciling the experimental data to the model predictions. In the event that the optimal parameters do not satisfactorily predict all the experimental data, the model should be refined and new parameters should be estimated.

Often, the function $\Phi(\boldsymbol{\theta})$ corresponds to a weighted sum-of-squares error, since this formulation arises from a probabilistic setting given appropriate assumptions (57). Of particular importance is the analysis of the Hessian of the objective function at the optimal solution, i.e.,

$$H = \frac{\partial^2 \Phi}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}. \qquad 4.$$

The analysis of the Hessian reveals which combination of parameters can and cannot be estimated from the experimental data. In the latter case, the number of model parameters can often be refined either by fixing parameter values based on previously reported literature estimates or by reducing the model by using, for example, equilibrium or quasi-steady-state approximations. Rawlings & Ekerdt (52) provide a nice introduction to these methods along with several instructive examples.

Sensitivity analysis. Having pinpointed the location of the current circuit behavior in the model parameter space, sensitivity analysis offers a first approximation for determining which parameters are critical for altering circuit behavior (assuming that the desired behavior can be written quantitatively). Both local and global sensitivity analyses are useful in this regard. Local sensitivity analysis seeks to determine how a small perturbation to a given parameter affects the desired circuit response. For example, we might ask how the period of oscillation, τ, changes with respect to a given parameter, k. Then the local sensitivity s is defined as

$$s = \frac{\partial \tau}{\partial k}. \qquad 5.$$

More generically, the period τ could be written as a function of the state N, i.e., $g(N)$, yielding the general definition:

$$s = \frac{\partial g(N)}{\partial \boldsymbol{\theta}}. \qquad 6.$$

Subsequently, one can prioritize which parameter has the greatest effect on the desired response by examining the magnitude of the values in the sensitivity vector s (the greater the magnitude, the greater the effect). As noted above, prioritizing mutational targets should take into account the directionality of the sensitivity.

Alternatively, global sensitivity analysis can be used to identify mutational targets. The basic idea is to determine the effect of parameters on the desired output for ranges of parameter values to elucidate generic or global parameter functions. Feng et al. (28) used the random sampling–high dimensional model representation algorithm to address this task. Here, parameter values are randomly sampled from biologically relevant ranges, and the sensitivities are calculated by averaging the model responses over different parameter sets. When calculating global sensitivities, care must be taken when the model exhibits multiple steady states, periodic behavior without steady states, or chaotic behavior.

The choice of which sensitivity to use may depend on the complexity of the model. Local sensitivity appears better suited for interrogating simple models in which all or most parameters can be uniquely determined from experimental data. In this case, the local sensitivity should accurately reflect the circuit behavior in the experiment. For more complex models in which many combinations of
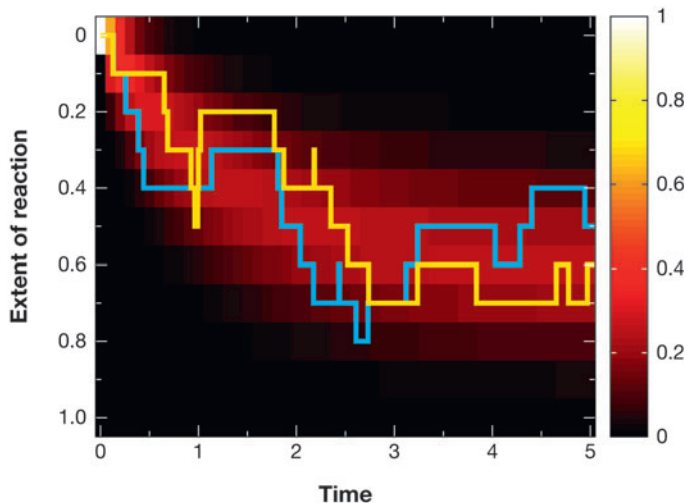
Figure 4

Evolution of both the master equation and two stochastic simulations for the reaction $A + B \leftrightarrow C$. This system can be characterized using a single extent of reaction. The stochastic simulation can explore both regions of high and low probability, and numerous stochastic simulations are required to reconstruct the evolution of the underlying master equation.

parameters cannot be estimated from data, global sensitivity analysis is more appropriate to overcome uncertainty in parameter values.

This use of sensitivity analysis is only appropriate as a local approximation. If the desired operating point is considerably far away in the parameter space, then extrapolating the calculated sensitivities is unlikely to accurately predict circuit behavior at this point. In this case, optimization can be used to determine the minimal parameter changes required to satisfy the design constraints. For example, one could minimize the change in parameters required to meet the design constraints subject to the mathematical model.

Overview of Stochastic Methods for Identifying Mutational Targets

Stochastic models have been used to explain phenomena that deterministic models cannot, such as how noise can partition a homogeneous cell population into distinct phenotypic subpopulations (5) or how quickly a

system passages between multiple attractors (26). These models assume that the probability of being in a particular molecular configuration evolves deterministically in time. One commonly used evolution equation is the discrete chemical master equation,

$$\frac{d P(\boldsymbol{n})}{d t} = \sum_{\beta} w_{\boldsymbol{n},\beta} P(\boldsymbol{\beta}) - w_{\beta,\boldsymbol{n}} P(\boldsymbol{n}), \quad 7.$$

where $\boldsymbol{n}$ is the system state (usually numbers of molecules), $P(\boldsymbol{n})$ is the probability that the system is in state $\boldsymbol{n}$, and $w_{n,\beta}$ is the rate at which the system moves from state $\boldsymbol{n}$ to $\boldsymbol{\beta}$. The solution of Equation 7 is computationally intractable for all but the simplest systems because its size can be enormous. Generally dynamic Monte Carlo simulation (34) is employed to reconstruct the probability distribution and its statistics (usually the mean and variance). Here, simulating the same initial condition given the same parameters can yield different individual trajectories (realizations of the underlying probability distribution), as illustrated in **Figure 4**.

Analyzing stochastic models is less straightforward than analyzing deterministic models. Steady-state analysis is complicated by the fact that these models are solved by dynamic simulation. In this case, either the model can be simulated for long periods, or one can assume a functional form of the solution and employ short bursts of simulation to iteratively identify steady-state attractors (26). Alternatively, one can linearize the master equation around a macroscopic steady state using a small-noise expansion (23, 63), a technique that has proven useful in explaining fluctuations in experimental systems (48, 49). The solution to this linearized equation is a multivariate normal distribution, so its applicability is limited to describing systems in which the steady-state distribution is unimodal.

Parameter estimation and sensitivity analysis are conceptually similar to the deterministic method. Both of these tasks require calculating derivatives of expected quantities generated by simulation and as a result can

be computationally demanding to perform if the simulation is computationally expensive to evaluate. A key concept is to evaluate these derivatives using the same trajectory (practically, the same string of random numbers) to significantly reduce the variance of the sensitivity estimates (30). **Supplemental Figure 1** (follow the Supplemental Material link from the Annual Reviews home page at **http://www.annualreviews.org**) demonstrates the difference between evaluating derivatives using the same trajectory and evaluating derivatives using random trajectories.

CONCLUSIONS AND FUTURE DIRECTIONS

The dream is that well-characterized components can be easily assembled to generate novel genetic regulatory circuits. The reality is that this is hard to accomplish. The components and their assemblies are context dependent: Synthetic circuits do not function outside the cellular context and may behave differently when the context changes. Directed evolution presents a powerful tool for overcoming this problem, but the evolutionary search space can be large. Mathematical modeling can reduce this space to a reasonable size by identifying mutational targets. The modular nature of synthetic gene circuits and the mathematical model permit this reduction.

Screening of the circuit for the desired function is a necessity. However, screening individual components for altered function in simple circuits is a subtly different and powerful complementary approach. By evaluating circuit behavior for components with a well-characterized range of functionality, the nominal circuit can be systematically perturbed, permitting model validation and refinement. In addition, the library of components can be used interchangeably with other circuits. A key requirement of this methodology is the ability to engineer components with independent properties, that is, to per-

turb only a single parameter or combination of parameters while holding the remaining parameters constant. It is unclear whether alteration of biophysical components such as proteins and DNA will actually permit such perturbations. However, recent works have demonstrated the feasibility of selecting for multiple properties, suggesting that such perturbations may indeed be possible at the price of designing more complicated screens or selections.

The reviewed works employing directed evolution to alter circuit performance suggest two additional strategies for selecting mutational targets. First, targets that can be detuned to yield functional circuit behavior are more likely to be generated and identified during directed evolution than are targets that require increases in efficiency. Although many mutations can detune function, mutations that improve a function are usually rare and therefore require more screening effort. Second, mutagenizing RBSs provides a generic and efficient strategy for manipulating some aspects of circuit function.

Directed evolution also has the potential for constructing novel regulatory functions altogether. Here, multiple rounds of selection can be used to construct circuit components that meet multiple, possibly orthogonal, design considerations. Also, exploiting the modularity of proteins shows promise in designing efficiently regulated enzymes, constructing new signaling pathways, and engineering novel regulation in eukaryotic cells.

Identifying mutational targets from mathematical models involves first using parameter estimation to pinpoint the location of the implemented circuit in the parameter space and then sensitivity analysis or optimization to suggest mutational targets. These tasks are straightforward for deterministic models. Because stochastic models are generally solved by simulation, these tasks can be performed efficiently by evaluating required derivatives using the same simulation trajectories.

SUMMARY POINTS

1. Synthetic circuit components and their assemblies are context dependent: They do not function outside the cellular context and may behave differently when the context changes.

2. Context dependence can be overcome by combining directed evolution with mathematical modeling. Using mathematical modeling to identify mutational targets reduces the evolutionary search space for directed evolution, which manipulates circuit performance without requiring mechanistic or detailed structural information.

3. Screening individual components for altered function in simple circuits allows systematic perturbation of the nominal circuit, permitting model validation and refinement while generating a library of components that can be used interchangeably with other circuits.

4. RBSs and targets that detune function are two effective mutational targets that alter circuit performance.

5. Selecting for orthogonal properties and exploiting the modularity of regulatory proteins facilitate construction of novel regulatory functions by directed evolution.

FUTURE ISSUES

1. Is systematic perturbation of circuit components a useful and generic strategy for tuning circuit function? Also, do individual components tuned in one context behave similarly once integrated into the context of the nominal circuit?

2. The parameter space accessible by random mutation of the DNA is not well understood. As a result, it is unclear what circuit functions are accessible by directed evolution.

3. Directed evolution can be used to construct libraries of well-characterized components spanning a wide range of parameter values. Will these libraries enable "plug and play" alteration of future circuits?

4. What are the scalability issues involved in fine-tuning synthetic circuits? Can a few small changes tune circuit behavior when the number of components becomes large?

ACKNOWLEDGMENTS

LITERATURE CITED

1. Aharoni A, Gaidukov L, Khersonsky O, McQ Gould S, Roodveldt C, Tawfik DS. 2005. The 'evolvability' of promiscuous protein functions. *Nat. Genet*. 37:73–76

2. Alper H, Fischer C, Nevoigt E, Stephanopoulos G. 2005. Tuning genetic control through promoter engineering. *Proc. Natl. Acad. Sci. USA* 102:12678–83

3. Anderson JC, Clarke EJ, Arkin AP, Voigt CA. 2006. Environmentally controlled invasion of cancer cells by engineered bacteria. *J. Mol. Biol*. 355:619–27

4. Andrianantoandro E, Basu S, Karig DK, Weiss R. 2006. Synthetic biology: new engineering rules for an emerging discipline. *Mol. Syst. Biol*. 2:2006.0028

5. Arkin A, Ross J, McAdams H. 1998. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli* cells. *Genetics* 149:1633–48

6. Atkinson M, Savageau MA, Myers JT, Ninfa AJ. 2003. Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. *Cell* 113:597–607

7. Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R. 2005. A synthetic multicellular system for programmed pattern formation. *Nature* 434:1130–34

8. Basu S, Mehreja R, Thiberge S, Chen M-T, Weiss R. 2004. Spatiotemporal control of gene expression with pulse-generating networks. *Proc. Natl. Acad. Sci. USA* 101:6355–60

9. Becskei A, Séraphin B, Serrano L. 2001. Positive feedback in eukaryotic gene networks: cell differentiation by graded to binary response conversion. *EMBO J*. 20:2528–35

10. Becskei A, Serrano L. 2000. Engineering stability in gene networks by autoregulation. *Nature* 405:590–93

11. Bhattacharyya RP, Reményi A, Yeh BJ, Lim WA. 2006. Domains, motifs, and scaffolds: the role of modular interactions in the evolution and wiring of cell signaling circuits. *Annu. Rev. Biochem*. 75:655–80

12. Bloom JD, Meyer MM, Meinhold P, Otey CR, MacMillan D, Arnold FH. 2005. Evolving strategies for enzyme engineering. *Curr. Opin. Struct. Biol*. 15:447–52

13. Bloom JD, Silberg JJ, Wilke CO, Drummond DA, Adami C, Arnold FH. 2005. Thermo-dynamic prediction of protein neutrality. *Proc. Natl. Acad. Sci. USA* 102:606–11

14. Bulter T, Lee S, Wong WW, Fung E, Connor MR, Liao JC. 2004. Design of artificial cell-cell communication using gene and metabolic networks. *Proc. Natl. Acad. Sci. USA* 101:2299–304

15. Chockalingam K, Zhao H. 2005. Creating new specific ligand-receptor pairs for transgene regulation. *Trends Biotechnol*. 23:333–35

16. Cohen N, Abramov S, Dror Y, Freeman A. 2001. In vitro enzyme evolution: the screening challenge of isolating the one in a million. *Trends Biotechnol*. 19:507–10

17. Collins CH, Arnold FH, Leadbetter JR. 2005. Directed evolution of *Vibrio fischeri* LuxR for increased sensitivity to a broad spectrum of acyl-homoserine lactones. *Mol. Microbiol*. 55:712–23

18. Collins CH, Leadbetter JR, Arnold FH. 2006. Dual selection enhances the signaling specificity of a variant of the quorum-sensing transcriptional activator LuxR. *Nat. Biotechnol*. 24:708–12

19. Collins CH, Yokobayashi Y, Umeno D, Arnold FH. 2003. Engineering proteins that bind, move, make and break DNA. *Curr. Opin. Biotechnol*. 14:371–78

20. Dhooge A, Govaerts W, Kuznetsov YA, Mestrom W, Riet AM. 2004. CL_MATCONT: a continuation toolbox in Matlab. **http://users.ugent.be/∼ajdhooge/research.html#new**

21. Doedel EJ, Paffenroth RC, Champneys AR, Fairgrieve TF, Kuznetsov YA, et al. 2002. AUTO 2000: continuation and bifurcation software for ordinary differential equations (with HomCont). *Tech. Rep*., Concordia Univ., Montreal, Can.

22. Drummond DA, Iverson BL, Georgiou G, Arnold FH. 2005. Why high-error-rate random mutagenesis libraries are enriched in functional and improved proteins. *J. Mol. Biol*. 350:806–16

23. Elf J, Ehrenberg M. 2003. Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome Res*. 13:2475–84

24. Elowitz MB, Leibler S. 2000. A synthetic oscillatory network of transcriptional regulators. *Nature* 403:335–38

25. Endy D. 2005. Foundations for engineering biology. *Nature* 438:449–53

26. Erban R, Kevrekidis IG, Adalsteinsson D, Elston TC. 2006. Gene regulatory networks: a coarse-grained, equation-free approach to multiscale computation. *J. Chem. Phys*. 124:084106

27. Farmer WR, Liao JC. 2000. Improving lycopene production in *Escherichia coli* by engineering metabolic control. *Nat. Biotechnol*. 5:533–37

28. Feng X, Hooshangi S, Chen D, Li G, Weiss R, Rabitz H. 2004. Optimizing genetic circuits by global sensitivity analysis. *Biophys. J*. 87:2195–202

29. François P, Hakim V. 2004. Design of genetic networks with specified functions by evolution in silico. *Proc. Natl. Acad. Sci. USA* 101:580–85

30. Fu M, Hu JQ. 1997. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. Boston: Kluwer

31. Fung E, Wong WW, Suen JK, Bulter T, Lee S, Liao JC. 2005. A synthetic gene-metabolic oscillator. *Nature* 435:118–22

32. Galvão TC, de Lorenzo V. 2006. Transcriptional regulators a la carte: engineering new effector specificities in bacterial regulatory proteins. *Curr. Opin. Biotechnol*. 17:34–42

33. Gardner TS, Cantor CR, Collins JJ. 2000. Construction of a genetic toggle switch in *Escherichia coli*. *Nature* 403:339–42

34. Gillespie DT. 1976. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys*. 22:403–34

35. Guet CC, Elowitz MB, Hsing W, Leibler S. 2002. Combinatorial synthesis of genetic networks. *Science* 296:1466–70

36. Guido NJ, Wang X, Adalsteinsson D, McMillen D, Hasty J, et al. 2006. A bottom-up approach to gene regulation. *Nature* 439:856–60

37. Guntas G, Mansell TJ, Kim JR, Ostermeier M. 2005. Directed evolution of protein switches and their application to the creation of ligand-binding proteins. *Proc. Natl. Acad. Sci. USA* 102:11224–29

38. Hasty J, McMillen D, Collins JJ. 2002. Engineered gene circuits. *Nature* 420:224–30

39. Hyun Park S, Zarrinpar A, Lim WA. 2003. Rewiring MAP kinase pathways using alternative scaffold assembly mechanisms. *Science* 299:1061–64

40. Kobayashi H, Kaern M, Araki M, Chung K, Gardner TS, et al. 2004. Programmable cells: interfacing natural and engineered gene networks. *Proc. Natl. Acad. Sci. USA* 22:8414–19

41. Kramer BP, Viretta AU, Daoud-El-Baba M, Aubel D, Weber W, Fussenegger M. 2004. An engineered epigenetic transgene switch in mammalian cells. *Nat. Biotechnol*. 22:867–70

42. Levskaya A, Chevalier AA, Tabor JJ, Simpson ZB, Lavery LA, et al. 2005. Synthetic biology: engineering *Escherichia coli* to see light. *Nature* 438:441–42

43. Mayo AE, Setty Y, Shavit S, Zaslaver A, Alon U. 2006. Plasticity of the *cis*-regulatory input function of a gene. *PLoS Biol*. 4:e45

44. McAdams HH, Arkin A. 2000. Gene regulation: towards a circuit engineering discipline. *Curr. Biol*. 10:R318–20

45. McDaniel R, Weiss R. 2005. Advances in synthetic biology: on the path from prototypes to applications. *Curr. Opin. Biotechnol*. 16:476–83

46. Olsen M, Iverson B, Georgiou G. 2000. High-throughput screening of enzyme libraries. *Curr. Opin. Biotechnol*. 11:331–37

47. Papworth M, Kolasinska P, Minczuk M. 2006. Designer zinc-finger proteins and their applications. *Gene* 366:27–38

48. Paulsson J. 2004. Summing up the noise in gene networks. *Nature* 427:415–18

49. Pedraza JM, van Oudenaarden A. 2005. Noise propagation in gene networks. *Science* 307:1965–69

50. Rackham O, Chin JW. 2005. A network of orthogonal ribosome x mRNA pairs. *Nat. Chem. Biol.* 1:159–66

51. Rao CV, Arkin AP. 2001. Control motifs for intracellular regulatory networks. *Annu. Rev. Biomed. Eng.* 3:391–419

52. Rawlings JB, Ekerdt JG. 2002. *Chemical Reactor Analysis and Design Fundamentals*. Madison, WI: Nob Hill Publ.

53. Reed JL, Palsson BO. 2003. Thirteen years of building constraint-based in silico models of *Escherichia coli*. *J. Bacteriol.* 185:2692–99

54. Shultzaberger RK, Bucheimer RE, Rudd KE, Schneider TD. 2001. Anatomy of *Escherichia coli* ribosome binding sites. *J. Mol. Biol.* 313:215–28

55. Solem C, Jensen PR. 2002. Modulation of gene expression made easy. *Appl. Environ. Microbiol.* 68:2397–403

56. Sprinzak D, Elowitz MB. 2005. Reconstruction of genetic circuits. *Nature* 438:443–48

57. Stewart WE, Caracotsios M, Sørensen JP. 1992. Parameter estimation from multiresponse data. *AIChE J.* 38:641–50

58. Strogatz SH. 1994. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Cambridge, MA: Westview

59. Süel GM, Garcia-Ojalvo J, Liberman LM, Elowitz MB. 2006. An excitable gene regulatory circuit induces transient cellular differentiation. *Nature* 440:545–50

60. Suzuki M, Christians FC, Kim B, Skandalis A, Black ME, Loeb LA. 1996. Tolerance of different proteins for amino acid diversity. *Mol. Divers.* 2:111–18

61. Deleted in proof

62. Umeno D, Tobias AV, Arnold FH. 2005. Diversifying carotenoid biosynthetic pathways by directed evolution. *Microbiol. Mol. Biol. Rev.* 69:51–78

63. van Kampen NG. 1992. *Stochastic Processes in Physics and Chemistry*. Amsterdam, The Neth.: Elsevier. 2nd ed.

64. Weiss R, Basu S. 2002. *The device physics of cellular logic gates*. Presented at 1st Workshop Non-Silicon Comput., Cambridge, MA

65. Weiss R, Homsy GE, Knight TF Jr. 1999. Toward in vivo digital circuits. In *Dimacs Workshop on Evolution as Computation*, ed. L Landweber, E Winfree, R Lipton, S Freeland, pp. 275–95. Princeton, NJ: Springer

66. Yokobayashi Y, Weiss R, Arnold FH. 2002. Directed evolution of a genetic circuit. *Proc. Natl. Acad. Sci. USA* 9:16587–91

67. You L, Cox RS III, Weiss R, Arnold FH. 2004. Programmed population control by cell-cell communication and regulated killing. *Nature* 428:868–71

Contents

INDEX

ERRATA

An online log of corrections to *Annual Review of Biophysics and Biomolecular Structure*
chapters (if any, 1997 to the present) may be found at
http://biophys.annualreviews.org/errata.shtml