A synthetic biology challenge: making cells compute

Cheemeng Tan,[a] Hao Song,[a] Jarad Niemi[b] and Lingchong You*[ac]

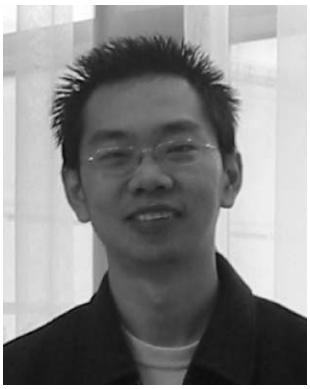
Received 18th December 2006, Accepted 8th February 2007
First published as an Advance Article on the web 27th February 2007
DOI: 10.1039/b618473c

Advances in biology and engineering have enabled the reprogramming of cells with well-defined functions, leading to the emergence of synthetic biology. Early successes in this nascent field suggest its potential to impact diverse areas. Here, we examine the feasibility of engineering circuits for cell-based computation. We illustrate the basic concepts by describing the mapping of several computational problems to engineered gene circuits. Revolving around these examples and past studies, we discuss technologies and computational methods available to design, test, and optimize gene circuits. We conclude with discussion of challenges involved in a typical design cycle, as well as those specific to cellular computation.

[a]*Department of Biomedical Engineering, Duke University, Durham, NC 27708, USA*
[b]*Institute of Statistics & Decision Sciences, Duke University, Durham, NC 27708, USA*
[c]*Institute for Genome Sciences and Policy, Duke University Medical Center CIEMAS 2345, 101 Science Drive, Durham, NC 27708, USA. E-mail: you@duke.edu*

Introduction

The last several years have witnessed the emergence of synthetic biology, a new area of biological research that aims to engineer gene circuits with desired functions for broad applications.[1–6] This line of research has led to the development of



Cheemeng Tan

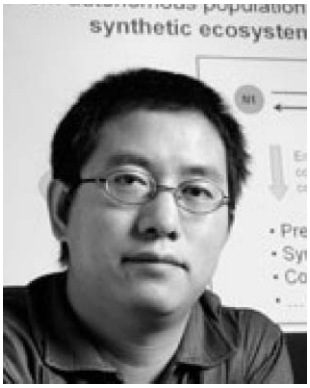
Cheemeng Tan is a graduate student in the Department of Biomedical Engineering at Duke University. He received his Master Degree in High Performance Computing from the Singapore-MIT alliance in 2002. His research focuses on using simulation and optimization methods to design synthetic bacteria.

Hao Song

Hao Song received his PhD in Chemical Engineering (2004) for works on polymerization reaction and process engineering at the University of Houston under the guidance of Prof. Dan Luss. He is currently a research associate in Prof. Lingchong You's laboratory at the Biomedical Engineering Department of Duke University. His research interests include synthetic and systems biology, focusing on engineering genetic circuits and their applications.

Jarad Niemi

Jarad Niemi is a graduate student in the Institute of Statistics and Decision sciences at Duke University. His research interest is on modeling and inference in non-linear multivariate processes representing stochastic dynamics of cellular protein networks at the single cell level. Jarad earned his Bachelor degree in chemical engineering (1999) and his Master's in biostatistics (2005), both at the University of Minnesota.



Lingchong You

Lingchong You is an Assistant Professor in the Department of Biomedical Engineering and the Institute for Genome Sciences and Policy at Duke University. He obtained his PhD in Chemical Engineering from the University of Wisconsin-Madison in 2002. He then did postdoctoral research at the California Institute of Technology before moving to Duke University in 2004. Using a combination of mathematical modeling and experiments, his laboratory explores design features of natural biological networks and builds synthetic gene circuits for applications in computation and medicine. He received a David and Lucile Packard Fellowship in 2006.

diverse systems, such as oscillators,[7–9] communication-based circuits,[10,11] band detectors,[11] bistable switches,[12–15] riboswitches,[16–19] logic gates,[20,21] and drug-producing cells.[22,23] The broad spectrum of advances highlights the potential of the nascent field to impact diverse areas, including medicine,[24] bioremediation,[25] and computation.[26]

Current computation is largely performed by digital computers with binary information encoding. Information is transmitted and processed by electronic circuits, which can be arranged to eliminate crosstalk between signals and circuit components so as to minimize errors in information processing. It has been recognized that there are limitations to current digital computing technology: electronic circuits are often rigid and inflexible by design. They cannot evolve and adapt once implemented. Furthermore, computing is energy inefficient—even a desktop PC can consume significant amount of energy and generate waste heat. Furthermore, their architecture, which is largely based on sequential computation of complex tasks, may not be best suited for handling certain classes of computation.[27–29] Finally, digital computers are believed to approach their computing speed limits due to constraints in miniaturization of transistors.[30] These challenges have led to the development of new computing modes that build on alternative technologies, such as quantum computing[31] and DNA computing.[32]

The advent of synthetic biology offers yet another opportunity to address limitations of digital computation, which may ultimately lead to ''cellular computation''.[33] At the outset, cells appear to possess some distinct advantages over digital computers. Cells can thrive in extreme environments, such as geysers with high temperatures,[34] deep oceans with extreme pressures,[35] and even nuclear plants filled with toxic chemicals.[36] They can be readily engineered to sense and respond to diverse environmental signals. They can communicate with one another to exchange information about their environment,[37] which enables coordinated population dynamics.[38] In this scenario, a cell population can be considered as a vast collection of miniature computers working in parallel, where each consumes an extremely minute amount of energy (at least in comparison to digital computers). Importantly, cells can adapt and evolve when they are challenged with new information that arises due to changes in nutrient abundance or temperature.[39] This feature has been exploited to optimize cellular parts or gene circuits.[40,41]

These apparent advantages suggest the appealing potential and opportunity of using cells as programmable devices for computation. Neurobiology has a long history of considering neurons as computing units and extensive research is being carried out to decode mechanisms of such computing.[42–44] These studies have led to attempts to design simple chemical networks to implement basic computing tasks, such as a Turing machine[45] and a frequency band-pass filter.[46] Along another line of thinking, biological development has inspired the concept of ''amorphous computing'' that aims to generate coherent behavior from a group of unreliable and unknown parts, such as cells.[47] Given the impressive advances in synthetic biology, it seems a natural next step to consider implementing computing tasks from bottom-up using well-defined synthetic gene circuits. Before engineering cells to perform human-defined computation, however, we must address several fundamental questions. For example, what kind of computational problems can we pose to cells in such a way to fully exploit their advantages? What are the fundamental challenges involved in cellular information processing and computation? Exploration of these issues will benefit from deeper understanding of how cells process information in nature, which may in turn offer inspiration for cellular computation.

Cellular information processing

Cells process a plethora of environmental information relevant to their functions, such as variations in nutrients, growth factors, and temperature. This processing can be regarded as the flow of information from the extracellular environment, across the cell membrane and, through a series of signaling networks in the cytoplasm, finally initiating gene expression from the chromosome (Fig. 1).[48] Extracellular information is received by cell membrane receptors and transformed into intracellular signals. For instance, binding of a specific ligand to a receptor can often trigger fast time-scale responses, such as methylation or phosphorylation of the intracellular domains of the receptor. These responses in turn can modulate downstream signaling cascades, leading to longer-term regulation of a diverse array of cellular components, such as kinases, phosphatases, secondary messengers, and transcription factors. Some proteins (*e.g.*, kinases) can directly regulate cellular metabolic activities in the cytoplasm. Others (*e.g.*, transcription factors) may translocate to the nucleus to activate gene expression. However, natural information processing may bypass some steps in the generic architecture illustrated here. An extracellular molecule may diffuse across the cell membrane, activate a transcription regulator, and then initiate gene

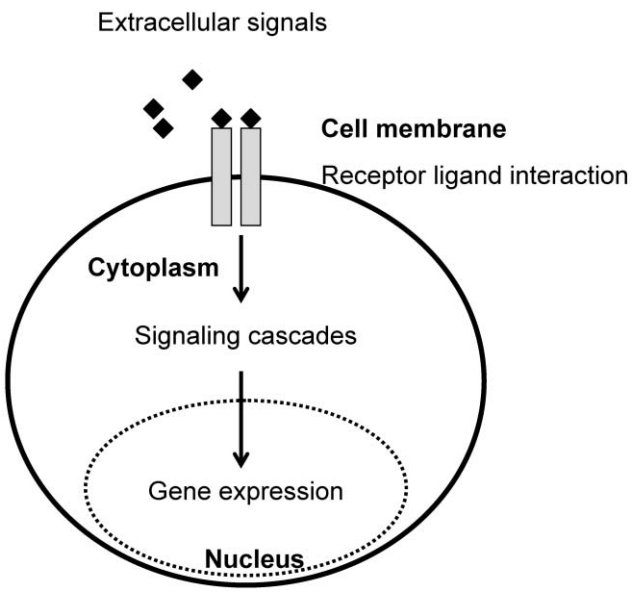


Fig. 1 Information flow in a cell. Upon binding with extracellular signals, cell membrane receptors activate downstream signaling molecules. These signaling molecules (*e.g.*, kinase, phosphatase, *etc.*) are further processed by signaling cascades in the cytoplasm. These components may subsequently activate transcription factors, which in turn activate gene expression.

expression. This is a common mechanism of quorum sensing for many Gram-negative bacteria. In this case, information processing bypasses many intermediate steps, for example the receptor–ligand binding reactions.

Natural cellular information processing exhibits several interesting features. In particular, cells use various mechanisms to regulate information processing. This regulation may be conceptualized as being carried out by network motifs. A motif, such as feedback or feedforward regulation, is commonly defined as a small set of interactions among cellular components with a well-defined function.[49–53] For example, positive feedback serves as the basis for ultrasensitive or bistable switches.[54,55] Negative feedback may reduce noise and introduce stability,[56] but may also generate oscillations if a long time delay is involved.[57] A wide variety of feedforward loops are also common in cellular networks and have diverse yet well-defined dynamic properties depending on their specific architecture.[58] Network motifs can be integrated to carry out more sophisticated cellular functions.[50,59] These may include cell cycle regulation, apoptosis, and differentiation.

Another feature of cellular information processing is parallelism, where a large number of cells can carry out a certain function simultaneously. The parallelism is also evident when cells process complex information from diverse sources in parallel.[60,61] Such parallel information processing can be illustrated using the *Escherichia coli* chemotaxis pathway, by which *E. coli* cells sense extracellular stimuli and then migrate accordingly. Bacteria live in a dynamic environment with multiple conflicting stimuli, including chemoattractants and chemorepellents. These stimuli simultaneously bind to respective membrane receptors and initiate signaling cascades leading to phosphorylation or dephosphorylation of a protein CheY. The extent of CheY phosphorylation modulates the frequency and direction of flagellar motor rotation, which then controls *E. coli* migration. Equipped with signaling cascades and gene regulatory networks, *E. coli* can respond flexibly to different streams of information in a complex and dynamic environment.

Mapping computation to circuit engineering

The versatility of cells in information processing seems to suggest a great potential for cellular computation mediated by engineered gene circuits. One immediate set of questions then relates to the nature of computational problems for which cellular computation might be advantageous. Evidently, some initial examples would be various logic gates that have been proposed or implemented over the last few years.[20,26] Combinations of these logic gates can in principle serve as the basis for universal computation.[62] To move towards more complex computation problems, however, we will need to develop much deeper understanding of how cells process information in natural contexts.

Here we attempt to map three commonly encountered computational problems (integration, optimization and signal-processing) to specific designs of gene circuits to illustrate the concept of cellular computation. For simplicity, we assume all these circuits are being implemented in *E. coli*, a major workhorse for synthetic biology research to date. These circuits (Fig. 2–4) are hypothetical and their actual implementation may require fine-tuning or replacement of individual genetic components. However, we believe that these designs are within our reach given recent achievements in the field. Their implementation, if demonstrated to be functional, may open doors to exciting opportunities for cellular computation. Importantly, analysis and implementation of these circuits may offer insight into underlying mechanisms of natural cellular information processing.

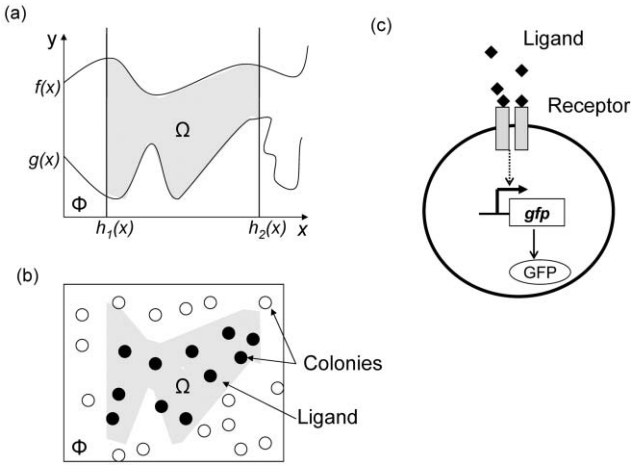


Fig. 2 An integrator. (a) The integration is defined as the limit sum of $f(x)$ minus $g(x)$, bounded by $h_1(x)$ and $h_2(x)$. (b) An area Ω (a subset of an area Φ) will be covered by uniform concentration of a specific ligand. (c) Cells can be engineered with a simple circuit that enables them to respond to the ligand by fluorescing. To "integrate", these cells can be randomly plated onto the surface. The integral can be calculated as fraction of the fluorescent colonies multiplied by Φ.

Integration

The aim is to compute the integral of a function $f(x)$ in a limit, defined by $h_1(x)$, $h_2(x)$ and $g(x)$ (eqn (1) and Fig. 2a).

$$\text{Integral} = \int_{h_1(x)}^{h_2(x)} (f(x) - g(x))\mathrm{d}x \quad (1)$$

When $f(x)$ and $g(x)$ are simple functions, the integral is calculated easily with simple numerical methods such as the Simpson rule or the Trapezoidal rule. For more complex functions, such as ones encountered in quantum mechanics,[63] the integral can often be approximated numerically using one of many Monte Carlo integration methods.[64] The integral is essentially the area (Ω) as defined in Fig. 2a. We now define an area Φ with well-defined dimensions that encloses Ω. Such algorithms evaluate multiple points in Φ and the points inside Ω will be registered as "hits". In the simplest Monte Carlo method, the integral can then be calculated as the ratio of hits to the total number of points multiplied by Φ (eqn (2)). With this method, the approximation error decreases proportionally with increasing total number of points evaluated in Φ.

$$\text{Integral} = \frac{\#\text{hits}}{\#(\text{all points})}\Phi = \frac{\#(\text{fluorecent colonies})}{\#(\text{all colonies})}\Phi \quad (2)$$

Such a method can be readily mapped into cellular computation. For simplicity, we consider a bounded area in an agar plate, covered by a uniform concentration of an immobilized ligand (Fig. 2b). To ''integrate'', we can engineer bacteria to carry a simple circuit that enables them to respond to the ligand by fluorescing (Fig. 2c). By randomly plating these bacteria onto the agar surface, one can perform ''integration'' by simple colony counting: the integral can be estimated by the fraction of fluorescent colonies multiplied by the total area of the agar plate (eqn (2)). This simple system can be considered as an ''integrator'' and it can be implemented experimentally by using well-characterized inducible promoters. For illustration of the concept, colony counting on a surface will be convenient. However, the integration can also be carried out in three dimensions to integrate over irregular volumes. In this case, one may first collect the bacteria and use flow cytometry to count the number of fluorescent bacteria.

We note that the non-trivial computational method can be mapped to an apparently trivial biological circuit (an inducible promoter). This suggests that using cells to do what we normally consider as ''computation'' may be quite straightforward. By itself, a simple integrator will unlikely have any advantage over current computing methods. However, if cells can be engineered so that the integration result is connected to a physiological function, then cells can carry out real-time calculation and make decisions accordingly. For instance, one can imagine using a ligand to control a quorum sensing module to enable cell–cell communication. This configuration may allow cells to ''calculate'' the volume of an environment and act accordingly. In a sense, natural quorum sensing bacteria are constantly using such Monte Carlo type integration to probe their environments.

Optimization

The notion of coupling environmental sensing and communication can lead to a gene circuit working as an ''optimizer''. In an optimization problem as formulated in eqn (3), the aim is to find the (x, y) value in a bounded search space that maximizes an objective function $F(x, y)$

$$\textit{Objective}: \max F(x, y) \tag{3}$$

For a simple $F(x,y)$ with a distinct global maximum in the space, the problem can be readily solved by using numerical methods such as interior point methods or simplex methods.[65] These methods often fail for complex $F(x, y)$ with multiple local maxima. In this case, the problem can be solved by using stochastic search algorithms such as simulated annealing,[66] genetic algorithms,[67] and swarm algorithms.[68] These stochastic search algorithms are also Monte Carlo methods that depend on statistical sampling of a search space. Briefly, $F(x, y)$ is evaluated at multiple points in the search space. The evaluations in each iteration are used to guide the search directions in the next iteration. The processes of evaluation and searching are repeated until the algorithm finds solutions that maximize $F(x, y)$ With limited computational resources, the success in locating the optimal solution will depend on the efficiency of ''search'' strategies. This efficiency can be improved by incorporating ''communication'' in the search process. That is, the algorithm integrates information from evaluations to choose appropriate search directions for the next iteration.[68] In more mainstream statistical analysis, stochastic search methods are also coupled with Monte Carlo integration and ''swarm-like'' variants have proven most effective in even very challenging applications.[69,70]

Many such algorithms have been inspired from biological processes (*e.g.* swarm algorithms and genetic algorithms) and they can be readily mapped back to cellular computation. $F(x, y)$ can be represented by a distribution of ligand molecules on a surface (Fig. 3a). The molecules are immobilized and are not metabolized by cells. Bacteria can be engineered to respond to these molecules and adjust their motility (Fig. 3b). This can be achieved by using the bacterial chemotaxis pathway that senses the ligand's concentration gradient. In addition, they need to be engineered to communicate with one another, for example, by using a quorum sensing module. The communication will coordinate cells' migration and increase the convergence rate of the population towards the optimal solution. In this design, we need to require that communication and ligand sensing be integrated at an AND gate. The AND gate can then be used to control chemotaxis by modulating the activity of key regulatory proteins, such as CheZ. This design aims to direct a cell population towards the optimal solution when both concentration gradient and cell density is high. We anticipate that cells regulated by this circuit will cluster significantly more than wild type cells at the globally optimal solution. As for any other complex gene circuit, successful implementation of the optimizer will require fine-tuning of circuit components to balance the amplitude and the signaling speed of both the ligand sensing module and the cell–cell communication module.

Some existing synthetic or natural circuits already possess partial characteristics of an optimizer described above. For instance, Anderson *et al.* engineered bacteria to invade cancer cells by using either a hypoxia-sensing module or a quorum sensing module.[24] As such, the engineered bacteria will only invade cancer cells either under an anaerobic condition or when their density is high. Modification of the gene circuit by integrating hypoxia-sensing and quorum-sensing at an AND gate may improve specificity of bacteria invasion of cancer cells. This modified circuit can be considered as an optimizer in which the bacteria locate the optimal solutions (the cancer cells) by detecting multiple input signals. Budrene and Berg have shown that bacteria aggregate at a local space by using both chemotaxis and self-secretion of a chemo-attractant.[71] In this case, bacteria become the source of attractant molecules to other cells, which further promotes bacteria chemotaxis towards a certain direction. Extension of this natural circuit by using a cell–cell communication module to control cell motility will also allow implementation of the optimizer. If we take lessons from performance of stochastic search algorithms, the integration of communication with motility behavior will likely improve ability of these cells to locate optimal solutions. However, this notion will require further investigation by modeling and experimentation.

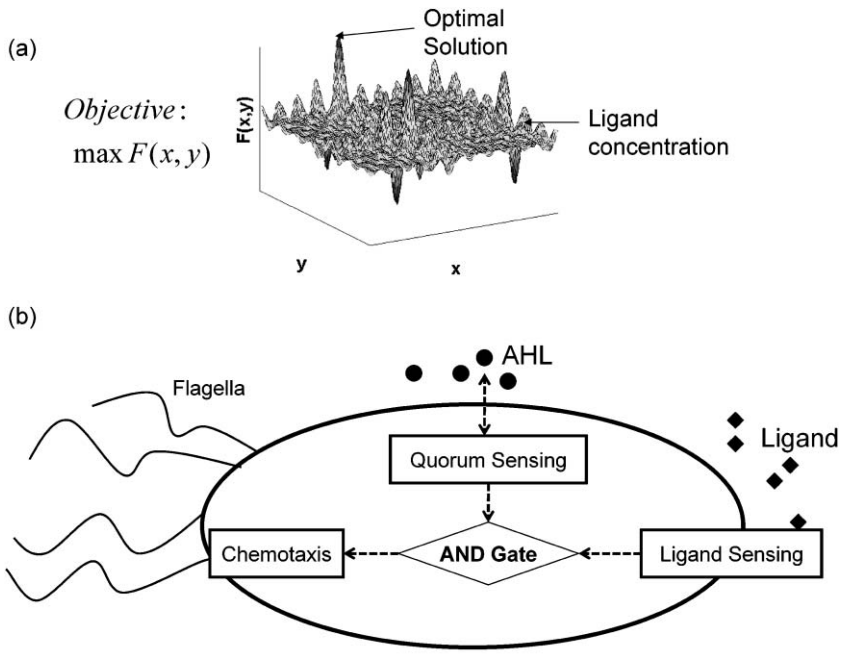


Fig. 3 An optimizer. (a) We aim to find an optimal solution (x, y) that maximizes a function $F(x, y)$. The function $F(x, y)$ can be represented by distribution of a specific ligand on a surface. (b) Cells can be engineered to sense the ligand concentration gradient and communicate with each other, and adjust their motility accordingly. Signals from ligand sensing and communication can be integrated at an AND gate, which modulates the activity of key chemosensory components, such as CheZ. The modulation aims to direct a cell migration towards a space with higher ligand concentration gradient and higher cell density. As a consequence of ligand sensing and communication, cells are expected to cluster around the position with the highest ligand concentration, thus realizing the optimization task.

Fourier transform

The first two examples will end up producing "steady-state" solutions of the computation problems. Natural information processing can also be carried out in the temporal domain. For example, consider an input signal represented by temporal fluctuations of a specific entity, such as electric pulses, concentrations of a molecule, or temperature. In many biological systems, an input signal can oscillate and its oscillation frequency may be critical. For example, during inflammatory response, expression of IL-2 and IL-8 cytokines peaks at an approximate $[Ca^{2+}]$ oscillation frequency of 0.01 s^{-1}.[72] In this case, one would wonder whether cells are processing frequency information, how they do it, and how we can achieve a similar task using synthetic circuits.

Computationally, frequency contents of an oscillatory signal $x(t)$ can be analyzed using the Fast Fourier Transform.[73] The essence of this method is to compare similarity between the signal ($x(t)$) and a basis function ($b(t)$) (eqn (4)). Common basis functions include sine and cosine functions. A coefficient λ can be calculated to indicate similarity between $x(t)$ and $b(t)$. In this analysis, the signal-processing capacity is directly proportional to the number of basis functions: higher number of basis functions increases the temporal sensitivity of the signal analysis.

$$\lambda = \int x(t)b(t)\mathrm{d}t \tag{4}$$

To map this computational method into cellular computation, we can define a signal $x(t)$ by changing the concentration of a signaling molecule (*e.g.*, AHL) (Fig. 4). We can then use a tunable natural or synthetic oscillator, such as the repressilator[7] to generate oscillations in a transcription regulator LuxR (R) with different frequencies, which "implements" the basis functions $b(t)$. In particular, we shall require AHL to activate LuxR (R*). If $x(t)$ has the same frequency as $b(t)$, we will expect the maximum concentration of active LuxR *via* a resonance effect. As a consequence, this can lead to maximum production of a fluorescent protein (GFP), whose concentration thus quantifies the similarity between $x(t)$ and $b(t)$. Based on the GFP concentration, we can identify $b(t)$ with high similarity with $x(t)$ and thus determine the frequency content of $x(t)$. We shall refer to this circuit as a "signal processor". We note that these concepts have been proposed in the context of chemical networks.[46] Their experimental implementation with engineered gene circuits will pose new and exciting challenges to synthetic biologists.

Again, aspects of this computation process can be identified in numerous natural biological processes, where cells process temporally changing information, for example in circadian clocks,[74] segmentation clocks,[75] and Ca^{2+} signaling pathways.[76] In these examples, several fundamental questions are yet to be elucidated. In particular, what is the speed limit of information processing in each system? What is the underlying mechanism of cells' response to an oscillatory signal? That is, how do cells decode frequency signals? Active pursuit of these questions will not only enable us to better understand how natural signaling networks are evolved to accurately handle diverse information contents, but also benefit engineering of cell-based computation.

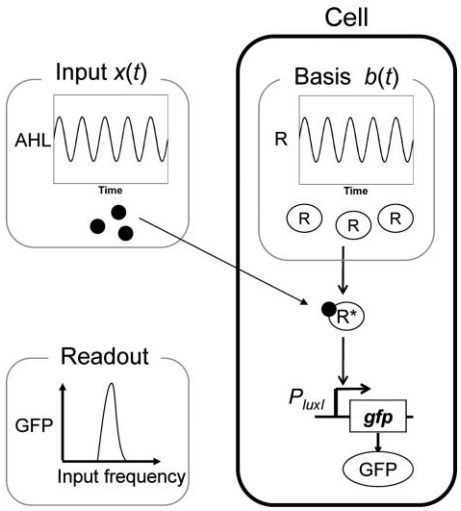


Fig. 4 A signal processor. We can define an input signal $x(t)$ by concentration changes of an AHL molecule. A natural or synthetic oscillator can be used to generate oscillations in LuxR with different frequencies. Each oscillation with a specific frequency serves as a basis function ($b(t)$). If AHL has the same frequency as LuxR, we will expect maximum amount of activated LuxR (R*) molecules. This will lead to maximum expression of a green fluorescent protein (GFP). The fluorescent intensity can then represent the similarity between the $x(t)$ and a particular $b(t)$. Since we know the frequency content of $b(t)$, we will be able to extract the frequency power spectrum of the $x(t)$. For clarity, we have shown one basis function. With a tunable oscillator, multiple basis functions with different frequencies can be generated.

Design cycle of synthetic gene circuits

Each of the three examples detailed above requires engineering of an appropriate gene circuit. Circuit engineering typically involves multiple rounds of a design cycle (see detailed review by Marguet *et al.*[6]). Given a conceptual design, mathematical modeling is often used to explore its dynamics, which may offer guidance for subsequent implementation, test, and revision. For circuits with complex dynamics, the role of modeling becomes even more critical. In addition to suggesting choices of appropriate circuit components to initiate implementation, modeling may also identify critical components for further optimization, by rational design or directed evolution. Conversely, data collected from experimental tests of an initial implementation can provide valuable information for evaluating and revising mathematical models and circuit parameters. This may benefit from established statistical methods.

Modeling-guided design

A wide spectrum of modeling methods, encompassing both deterministic and stochastic approaches, is available for circuit analysis. Interactions among circuit components can often be conveniently described by ordinary differential equations (ODE). An ODE model can be represented in a general form shown by eqn (5):

$$\frac{\mathrm{d}\tilde{C}}{\mathrm{d}t} = V\tilde{f}(\tilde{C}) \tag{5}$$

where $\tilde{C}$ is a vector containing concentrations of each molecular species, t is time, V represents the stoichiometry matrix of reaction networks, and $\tilde{f}(\tilde{C})$ is a vector of expressions defining the rate of change for the concentration of each species.

When spatial transport of cellular components (*e.g.* by diffusion) plays a significant role in a circuit function, partial differential equations (PDE) can be used. This method has been frequently used to analyze pattern formations in organism development.[77,78] Here, it would be most suited for analyzing the ''optimizer'', where cells migration and quorum sensing signals diffusion are integral part of the circuit dynamics. Typically, this can be modeled by a diffusion-reaction model, as represented in a generic form in eqn (6).

$$\frac{\mathrm{d}\tilde{C}}{\mathrm{d}t} = D\nabla^2\tilde{C} + V\tilde{f}(\tilde{C}) \tag{6}$$

where D is a matrix containing diffusivity of each component, and $\nabla^2\tilde{C}$ is the Laplacian of $\tilde{C}$ and it can be represented by $\frac{\partial^2\tilde{C}}{\partial x^2} + \frac{\partial^2\tilde{C}}{\partial y^2}$ in a two-dimensional Cartesian coordinate system (x, y).

Both ODE and PDE models are deterministic in the sense that, given the same initial conditions for each model, repeated simulations will produce the same results. Both models can be analyzed by using two popular types of mathematical analysis: parametric sensitivity analysis and bifurcation analysis. Parametric sensitivity analysis can be carried out to characterize quantitative changes of circuit dynamics in response to perturbations of circuit parameters.[79] This analysis is most helpful for identifying critical circuit components. Bifurcation analysis is used to determine how qualitative properties of a circuit depend on its parameters. Specifically, it aims to find the steady-state solutions of a system and their stability.[80,81] Bifurcation is said to occur when there is a change either in the number of steady state solutions or the stability of one or multiple solutions. For example, at a Hopf bifurcation point, a steady-state solution loses stability and sustained oscillation arises. Bifurcation analysis can be used to analyze the oscillator module in the signal-processor mentioned above, by predicting effects of circuit parameters on the oscillation amplitude and frequency.

The differential equations (DE) framework may be inadequate when molecular number of a species is low, which will generate ''noise'', or stochastic fluctuations in cellular processes. When cellular noise drastically affects a circuit function, it has to be accounted for in the mathematical analysis. This can be carried out by chemical Langevin equations (CLE) or chemical master equations (CME).[82] In CLE, a noise term is appended explicitly to an ODE. A typical form of CLE is shown in eqn (7).[82,83]

$$\frac{\mathrm{d}\tilde{C}}{\mathrm{d}t} = V\tilde{f}(\tilde{C}) + \sqrt{V\tilde{f}(\tilde{C})}\tilde{\xi}(t) \tag{7}$$

where the 2nd term captures intrinsic noise and $\tilde{\xi}(t)$ is the white noise with unit amplitude.

A CLE model can be solved numerically by algorithms derived for standard ODE models.[84] In fact, it approximates a CME model, which captures the ''true'' stochastic dynamics of

a system consisting of elementary reactions in homogenous environment. A CME model can be solved by using the Gillespie algorithm.[85] Following a Monte Carlo scheme, this algorithm tracks firings of individual reactions and time intervals between these firings.

Similar to spatial extension of ODE to PDE, spatial effects can also be incorporated into stochastic models. A spatial-stochastic model is particularly useful when noise affects reaction-diffusion dynamics of a molecule or a cell. This model has been applied to analyze intracellular signaling of MAPK cascades,[86] segmentation dynamics of *E. coli*,[87] and bacterial chemotaxis pathways.[88] The model can be used to analyze the optimizer by predicting effects of noise on cells' migration in a search space.

Implementation

Many circuits may be constructed by combinations of existing genetic parts.[5,89] Currently, there are concerted efforts to collect, index, and standardize commonly used genetic parts as Biobricks, which include RBS sequences, promoter sequences, and protein coding sequences (http://parts.mit.edu/). Well-characterized Biobricks will drastically facilitate implementation of diverse gene circuits by not only serving as the elementary building materials, but also by providing critical kinetic parameters for initial modeling. At the next level, well-characterized devices and systems with a matching interface can be integrated to generate even more complex systems.[1] For instance, implementation of the optimizer will require a quorum sensing module and an AND gate, as well as an interface to connect with the chemotaxis pathways.

Experimental test

Once a circuit is assembled, its performance can be evaluated by standard biological techniques, including assays for gene expression and cell growth. In particular, circuit engineering will benefit from recent development of an array of new detection technologies, including flow cytometry,[90] light microscopy,[91] and microfluidics.[92] In a flow cytometer, each cell is passed through an excitation light source. The light scattering and emission data from the cells will be collected. This technology has been used widely to collect population statistics such as fluorescence distribution[12,93] and cellular morphology.[94,95] Light microscopy is becoming an important tool for characterizing synthetic gene circuits. A distinct advantage of microscopy is that it can be used to track real-time cellular dynamics by recording time-lapse movies of engineered cells.[7] This technique has been used to study noise in gene expression of a single cell[96,97] and behavioral variations of bacteria chemotaxis activity.[98]

In some cases, engineered cells have to be characterized under a variety of specific conditions for an extended period. Such tasks will benefit from recent development in microfluidics. Based on advances in microfabrication technologies, microfluidics is the design of miniaturized devices that handle samples at nanoliter volumes.[92] One major impact of microfluidics on synthetic biology focuses on miniaturization of cell growth environment, in order to facilitate single cell measurements. For instance, Balagadde *et al.* used a microchemostat to characterize a population control circuit for hundreds of hours.[99] Groisman *et al.* used a microfluidic device to maintain chemostatic conditions for bacteria and yeast growing in microchambers.[100] Cookson *et al.* used a microchemostat to track gene expression of single yeast cells over multiple cell cycles.[101] Oloffson *et al.* created a chemical waveform synthesizer that can generate spatial chemical gradient on a microfluidic chip.[102] These devices can be particularly useful for characterizing circuits involving sophisticated manipulation of experimental conditions, for instance, the signal processor. In this case, a chemical waveform synthesizer can be used to generate an oscillatory signal and the microchemostat can be used to maintain the signal processor, in order to process a signal for a long period of time.

Model refinement guided by statistical methods

In the initial phase of mathematical modeling, reaction mechanisms and kinetic parameters are often derived from the literature or from the preliminary measurement of circuit components. Due to lack of quantitative information on cellular components (even standard parts), a model may not generate the ''true'' cellular dynamics as a result of parameter and model misspecification. In this case, and assuming adequacy of the overall functional form of the model, the analysis can be fine-tuned using statistical methods to fit the model to experimental measurements and refine understanding of relevant regions of parameter space. In the computational and technically attractive approach *via* Bayesian statistical methods, the steps involved are:

1) develop a statistical model representing errors of measurement, biological variability and parameter uncertainty,

2) summarize and represent existing biochemical information in terms of prior distributions, or more commonly classes of prior distributions, that reflect the substantive information and uncertainty about model parameters;

3) develop and understand the implied posterior distributions representing how the prior information is updated with experimental data to revise estimations and uncertainty measures about parameters—typically this is carried out using Markov chain Monte Carlo (MCMC) techniques to simulate and summarize the statistical analysis.[103,104]

4) Explore aspects of model fit, both within models fit to training data and in using fitted models to predict test data, for evaluation and criticism of assumed model forms that may lead to model refinements.

A statistical model, $P(Y|\theta)$, describes the relationship between the data Y to be observed and a set of unknown parameters θ, in terms of a population sampling distribution of the data based on the mathematically specified model combined with statistical error components. For examples, suppose data are fluorescence levels for the integrator, spatial cell densities for the optimizer, or fluorescence levels through time for the signal-processor. One statistical model for the integrator is shown in eqn (8).

$$P(Y_c|\theta) = N(\alpha+\beta x_c, \sigma^2) \quad (8)$$

where Y_c is the background adjusted fluorescence level of cell c,

x_c is 1 if cell c is bound to the ligand and 0 if not,

$N(\mu,\sigma^2)$ indicates a Gaussian distribution with mean μ and standard deviation σ,

α is the mean fluorescence level of unbound cells, and

$\alpha + \beta$ is the mean fluorescence level of ligand-bound cells.

If the cells are assumed to be independent, the full statistical model is as shown in eqn (9).

$$P(Y|\theta) = \prod_c P(Y_c|\theta) \tag{9}$$

After observing data, this is called the likelihood and is denoted $L(\theta)$, indicating that it is a function of the parameters having been defined on the basis of this specific, now known and fixed data outcome.

Bayesian parameter estimation requires encoding initial scientific information about parameters in a prior distribution that combines with the likelihood function to provide summary inferences. Repeat analysis with different models and priors is part of the natural, iterative model building process. Assuming one assigned *prior* denoted $P(\theta)$, analysis can proceed and then be repeated under variations of the prior and model. One typical approach is to choose priors under which component parameters are independent, *viz*. $P(\theta) = P(\alpha)P(\beta)P(\sigma)$. At this point, we can assign probability distributions that restrict the range of our parameters. For example, we may expect the fluorescence levels of proteins bound to the ligand to be higher than those that are not. Therefore a probability distribution should restrict the range of β to be positive. An example is to choose $P(\beta) = N(\beta_0,{\sigma_\beta}^2)\delta(\beta > 0)$ where β_0 is a prior point-estimate for β, ${\sigma_\beta}^2$ represents prior uncertainty about that estimate, and $\delta(\beta > 0)$ is 1 whenever $\beta > 0$ and 0 otherwise which restricts the range of β to the positive real numbers. Information from previous experimentation can be formally incorporated in this analysis through the prior. If we do not have any information about parameters, then we can use flat, non-informative priors.

After deriving the likelihood and assigning appropriate prior probability distributions to our parameters, the statistical model is fully specified. The next step, although not trivial, involves only algebra and calculus to derive the *posterior distribution*, $P(\theta|X)$, of the parameters given the data. This is defined by Bayes' theorem as in eqn (10).

$$P(\theta|X) \propto P(X|\theta)\ P(\theta) \tag{10}$$

The posterior can only be derived and understood mathematically in the simplest models. Nonetheless, MCMC and other numerical simulation methods can draw simulated values of the parameters from this posterior to obtain very large posterior samples. Such sampled parameter sets can be directly summarized—in terms of histograms, sample means, variances, quantiles and so forth—to provide descriptions of what has been learned from the experimental data relative to the specific prior distributions chosen.

As mentioned, such formal statistical methods can suggest model refinements in terms of suggesting different ranges of parameter values. For example, in the integrator, if the point-estimate for β is smaller then that for σ our signal-to-noise ratio will be very small. This indicates that the system requires cells that have a greater difference in fluorescence protein production between the ligand-bound and unbound states.

A variety of statistical models and data collection schemes can be developed to answer different questions. For example, in the optimizer, we can collect cell density measurements at a single time point or we can collect time-course data, and develop rich classes of statistical models for biochemical parameter estimation and evaluation of specific mathematical model forms.[105]

Circuit optimization and revision

Preliminary design and testing of an engineered circuit will allow us to pinpoint and modify specific circuit components, such as ribosomal binding sites, promoters, or genes. This can be carried out by structural based rational design, guided by molecular dynamic simulations. This strategy has been recently applied to engineer a bacterial membrane receptor to bind new ligands such as trinitrotoluene, l-lactate, and serotonin.[106]

However, when detailed knowledge about a component's structure is unavailable, directed evolution can be used to modify its function. Directed evolution optimizes a component according to a specific design objective. The optimization starts with generation of a component library by randomly changing its sequence, for example, by using the error-prone PCR method.[107] The library is screened to identify component variants that can satisfy the design objective. For instance, Collins *et al.* used directed evolution to increase specificity of a LuxR protein to a specific target molecule, C8HSL, while maintaining its specificity to a wild type target, 3OC6HSL.[40] The optimal LuxR proteins were selected based on their sensitivity to different dosage of C8HSL and 3OC6HSL. Similar approach has also been taken to optimize binding properties of receptor–ligand pairs,[108] nucleic acid enzymes,[109] and a genetic circuit.[41]

Challenges

While appealing, turning cells into computers will be tremendously challenging and requires deeper understanding of natural biological systems as well as new design strategies relevant to gene circuit engineering. In particular, what are the limits of cellular computation in terms of processing speed and accuracy? How do we control, modulate or exploit noise in cellular information processing? How can we increase robustness of a gene circuit, so that it will function properly under different conditions?

Computation speed is a critical factor in any computation. Cells running simple logic gates will not have advantages over digital computers—they would be too slow. Future cellular computation must be so designed to take advantage of the vast parallelism of cells. The parallelism can be exploited in algorithms that require a huge amount of simple calculations, such as the Monte Carlo type algorithms. This strategy is exemplified by the integrator and the optimizer. Still, the overall computation speed will benefit from optimization of individual components to increase their processing speed. This can be

improved by increasing the production rate or the decay rate of a critical component in a circuit.[110] It may also be modulated by incorporating network motifs such as negative feedback[111] or feed forward regulation.[58] Different cellular components will also have different intrinsic capabilities in terms of their processing speed. Signaling networks, which involve covalent modification of proteins (*e.g.* methylation and phosphorylation), in general have much faster dynamics than gene expression.

Importantly, computation results will be affected by the presence of cellular noise. As alluded earlier, cellular noise exists in the form of fluctuations in molecular concentration, which further leads to diverse cellular phenotypes.[82,112–117] Noise can be classified into two categories: the intrinsic noise, which arises from interactions between small numbers of interacting molecules; the extrinsic noise, which arises from fluctuations in the cellular environment.[96] Noise can be both advantageous and disadvantageous in cellular computation. On one hand, noise is undesirable because it may reduce computation accuracy by causing randomness in gene expression. For instance, in the repressilator, noise affects the circuit's function, resulting in only 40% of oscillating cells.[7] In this case, noise is a serious concern in engineering of a reliable and tunable oscillator. On the other hand, noise might be beneficial to cellular computation by introducing behavioral variability. For instance, Blake *et al.* have shown that variability in gene expression increases the survival advantage of yeast cells under acute environment stress.[118] In the context of cellular computing, a heterogeneous optimizer population with different migration rates and directions may allow cells to explore a bigger search space.[98] This way, the optimizer will have a higher probability of finding the optimal solutions.

The impact of noise and other perturbations from the environment can be probed by sensitivity analysis, bifurcation analysis, or stochastic modeling, as described above. In most cases, we would want an engineered circuit to be robust, or insensitive, to such perturbations. To this end, noise can be modulated by different control strategies.[82] For instance, it was demonstrated that negative feedback can reduce variability in gene expression.[56] Cell–cell communication has been suggested as an effective strategy to couple cellular oscillators to realize synchronous oscillations.[119,120] Other mechanisms, such as feedforward regulation[121] and gene dosage[122] may also enable further tuning of noise levels.

Conclusion

Drawing inspiration from biology, computer science, mathematics, and engineering, we may one day engineer cells to solve computation problems that are difficult to solve using conventional methods. The engineering process may also help us to understand how cells perform computation in nature. In the long term, we envision the design of cellular computers, consisting of cells with specific computation roles such as integrator, optimizer, and signal-processor.

Acknowledgements

We thank Mike West, John Pearson, and anonymous reviewers for critical reading of the manuscript and helpful discussions. This work was partially supported by the David and Lucile Packard Foundation, the National Institute of Health, and the National Science Foundation.

References

1 D. Endy, *Nature*, 2005, **438**, 449–453.
2 S. A. Benner and A. M. Sismour, *Nat. Rev. Genet.*, 2005, **6**, 533–543.
3 G. M. Church, *Mol. Syst. Biol.*, 2005, **1**, 0032.
4 R. McDaniel and R. Weiss, *Curr. Opin. Biotechnol.*, 2005, **16**, 476–483.
5 C. A. Voigt, *Curr. Opin. Biotechnol.*, 2006, **17**, 548–557.
6 P. Marguet, F. Balagadde, C. Tan and L. You, *J. R. Soc. Interface*, 2007, **10**, DOI: 10.1098/rsif.2006.0206.
7 M. B. Elowitz and S. Leibler, *Nature*, 2000, **403**, 335–338.
8 E. Fung, W. W. Wong, J. K. Suen, T. Bulter, S. G. Lee and J. C. Liao, *Nature*, 2005, **435**, 118–122.
9 M. R. Atkinson, M. A. Savageau, J. T. Myers and A. J. Ninfa, *Cell*, 2003, **113**, 597–607.
10 L. You, R. S. Cox III, R. Weiss and F. H. Arnold, *Nature*, 2004, **428**, 868–871.
11 S. Basu, Y. Gerchman, C. H. Collins, F. H. Arnold and R. Weiss, *Nature*, 2005, **434**, 1130–1134.
12 T. S. Gardner, C. R. Cantor and J. J. Collins, *Nature*, 2000, **403**, 339–342.
13 A. Becskei, B. Seraphin and L. Serrano, *EMBO J.*, 2001, **20**, 2528–2535.
14 B. P. Kramer and M. Fussenegger, *Proc. Natl. Acad. Sci. U. S. A.*, 2005, **102**, 9517–9522.
15 F. J. Isaacs, J. Hasty, C. R. Cantor and J. J. Collins, *Proc. Natl. Acad. Sci. U. S. A.*, 2003, **100**, 7714–7719.
16 C. I. An, V. B. Trinh and Y. Yokobayashi, *RNA*, 2006, **12**, 710–716.
17 F. J. Isaacs, D. J. Dwyer, C. Ding, D. D. Pervouchine, C. R. Cantor and J. J. Collins, *Nat. Biotechnol.*, 2004, **22**, 841–847.
18 T. S. Bayer and C. D. Smolke, *Nat. Biotechnol.*, 2005, **23**, 337–343.
19 S. K. Desai and J. P. Gallivan, *J. Am. Chem. Soc.*, 2004, **126**, 13247–13254.
20 C. C. Guet, M. B. Elowitz, W. Hsing and S. Leibler, *Science*, 2002, **296**, 1466–1470.
21 R. Weiss, Ph.D. Thesis, Massachusetts Institute of Technology, 2001.
22 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. Chang, S. T. Withers, Y. Shiba, R. Sarpong and J. D. Keasling, *Nature*, 2006, **440**, 940–943.
23 E. S. Gilbert, A. W. Walker and J. D. Keasling, *Appl. Microbiol. Biotechnol.*, 2003, **61**, 77–81.
24 J. C. Anderson, E. J. Clarke, A. P. Arkin and C. A. Voigt, *J. Mol. Biol.*, 2006, **355**, 619–627.
25 M. Urgun-Demirtas, B. Stark and K. Pagilla, *Crit. Rev. Biotechnol.*, 2006, **26**, 145–164.
26 R. Weiss, S. Basu, S. Hooshangi, A. Kalmbach, D. Karig, R. Mehreja and I. Netravali, *Natural Comput.*, 2003, 47–84.
27 J. Backus, *Commun. ACM*, 1978, **21**, 613–641.
28 M. Sipper, *Computer*, 1999, **32**, 18–26.
29 P. Wegner and D. Goldin, *Commun. ACM*, 2003, **46**, 100–102.
30 N. Forbes and M. Foster, *Comput. Sci. Eng.*, 2003, **5**, 18–19.
31 M. Hirvensalo, *Quantum Computing*, 1st edn, Springer-Verlag, New York, USA, 2001.
32 A. J. Ruben and L. F. Landweber, *Nat. Rev. Mol. Cell Biol.*, 2000, **1**, 69–72.
33 M. L. Simpson, G. S. Sayler, J. T. Fleming and B. Applegate, *Trends Biotechnol.*, 2001, **19**, 317–323.
34 R. Lieph, F. A. Veloso and D. S. Holmes, *Trends Microbiol.*, 2006, **14**, 423–426.
35 K. Horikoshi, *Curr. Opin. Microbiol.*, 1998, **1**, 291–295.
36 D. Appukuttan, A. S. Rao and S. K. Apte, *Appl. Environ. Microbiol.*, 2006.
37 B. L. Bassler, *Cell*, 2002, **109**, 421–424.
38 A. M. Lazdunski, I. Ventre and J. N. Sturgis, *Nat. Rev. Microbiol.*, 2004, **2**, 581–592.

39 H. H. McAdams, B. Srinivasan and A. P. Arkin, *Nat. Rev. Genet.*, 2004, **5**, 169–178.
40 C. H. Collins, F. H. Arnold and J. R. Leadbetter, *Mol. Microbiol.*, 2005, **55**, 712–723.
41 Y. Yokobayashi, R. Weiss and F. H. Arnold, *Proc. Natl. Acad. Sci. U. S. A.*, 2002, **99**, 16587–16591.
42 J. V. Neumann, *The Computer and the Brain*, 2nd edn, Yale University Press, New Haven, Connecticut, USA, 1958.
43 L. F. Abbott and W. G. Regehr, *Nature*, 2004, **431**, 796–803.
44 M. London and M. Hausser, *Annu. Rev. Neurosci.*, 2005, **28**, 503–532.
45 A. Hjelmfelt, E. D. Weinberger and J. Ross, *Proc. Natl. Acad. Sci. U. S. A.*, 1992, **89**, 383–387.
46 M. Samoilov, A. Arkin and J. Ross, *J. Phys. Chem. A*, 2002, **106**, 10205–10221.
47 H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. Knight, R. Nagpal, E. Rauch, G. J. Sussman and R. Weiss, *Commun. ACM*, 2001, 43.
48 J. Downward, *Nature*, 2001, **411**, 759–762.
49 C. V. Rao and A. P. Arkin, *Annu. Rev. Biomed. Eng.*, 2001, **3**, 391–419.
50 D. M. Wolf and A. P. Arkin, *Curr. Opin. Microbiol.*, 2003, **6**, 125–134.
51 H. M. Sauro and B. N. Kholodenko, *Prog. Biophys. Mol. Biol.*, 2004, **86**, 5–43.
52 A. Ma'ayan, R. D. Blitzer and R. Iyengar, *Annu. Rev. Biophys. Biomol. Struct.*, 2005, **34**, 319–349.
53 J. J. Tyson, K. C. Chen and B. Novak, *Curr. Opin. Cell Biol.*, 2003, **15**, 221–231.
54 J. E. Ferrell, Jr., *Curr. Opin. Cell Biol.*, 2002, **14**, 140–148.
55 U. S. Bhalla and R. Iyengar, *Science*, 1999, **283**, 381–387.
56 A. Becskei and L. Serrano, *Nature*, 2000, **405**, 590–593.
57 A. Goldbeter, *Biochemical oscillations and cellular rhythms*, Cambridge University Press, Cambridge, UK, 1996.
58 S. Mangan and U. Alon, *Proc. Natl. Acad. Sci. U. S. A.*, 2003, **100**, 11980–11985.
59 L. H. Hartwell, J. J. Hopfield, S. Leibler and A. W. Murray, *Nature*, 1999, **402**, C47–52.
60 R. B. Bourret and A. M. Stock, *J. Biol. Chem.*, 2002, **277**, 9625–9628.
61 J. D. Jordan, E. M. Landau and R. Iyengar, *Cell*, 2000, **103**, 193–200.
62 N. E. Buchler, U. Gerland and T. Hwa, *Proc. Natl. Acad. Sci. U. S. A.*, 2003, **100**, 5136–5141.
63 N. A. Benedek, I. K. Snook, M. D. Towler and R. J. Needs, *J. Chem. Phys.*, 2006, **125**, 104302.
64 C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd edn, Springer, New York, 2005.
65 D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Nashua, New Hampshire, USA, 1997.
66 S. Kirkpatrick, C. Gelatt and M. Vecchi, *Science*, 1983.
67 D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, 1989.
68 J. Kennedy and R. Eberhart, *Proc. IEEE Int. Conf. Neural Networks*, 1995, 1942–1948.
69 B. Jones, A. Dobra, C. Carvalho, C. Hans, C. Carter and M. West, *Stat. Sci.*, 2005, **20**, 388–400.
70 C. Hans, A. Dobra and M. West, *J. Am. Stat. Assoc., in press*, 2007.
71 E. O. Budrene and H. C. Berg, *Nature*, 1991, **349**, 630–633.
72 R. E. Dolmetsch, K. Xu and R. S. Lewis, *Nature*, 1998, **392**, 933–936.
73 A. V. Oppenheim, R. W. Schafer and J. R. Buck, *Discrete-Time Signal Processing*, 2nd edn, Prentice Hall, Upper Saddle River, New Jersey, USA, 1999.
74 S. L. Harmer, S. Panda and S. A. Kay, *Annu. Rev. Cell Dev. Biol.*, 2001, **17**, 215–253.
75 O. Pourquie, *Science*, 2003, **301**, 328–330.
76 M. J. Berridge, M. D. Bootman and H. L. Roderick, *Nat. Rev. Mol. Cell Biol.*, 2003, **4**, 517–529.
77 H. Meinhardt, *Models of biological pattern formation*, Academic Press, London, New York, 1982.
78 G. T. Reeves, C. B. Muratov, T. Schupbach and S. Y. Shvartsman, *Dev. Cell*, 2006, **11**, 289–300.
79 A. Varma, M. Morbidelli and H. Wu, *Parametric sensitivity in chemical systems*, Cambridge University Press, Cambridge and New York, 1999.
80 J. Guckenheimer and P. Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Corr. 7th print. edn, Springer, New York, 2002.
81 Y. A. Kuznetsov, *Elements of applied bifurcation theory*, 3rd edn, Springer-Verlag, New York, 2004.
82 C. V. Rao, D. M. Wolf and A. P. Arkin, *Nature*, 2002, **420**, 231–237.
83 D. T. Gillespie, *J. Chem. Phys.*, 2000, **113**, 297–306.
84 P. E. Kloeden, E. Platen and H. Schurz, *Numerical solution of SDE through computer experiments*, Springer-Verlag, Berlin, New York, 1994.
85 D. T. Gillespie, *J. Phys. Chem.*, 1977, **81**, 2340–2361.
86 K.-H. Chiam, C. M. Tan, V. Bhargava and G. Rajagopal, *Phys. Rev. E: Stat. Phys., Plasmas, Fluids. Relat. Interdiscip. Top.*, 2006, 74.
87 D. Fange and J. Elf, *PLoS Comput. Biol.*, 2006, **2**, e80.
88 T. S. Shimizu, S. V. Aksenov and D. Bray, *J. Mol. Biol.*, 2003, **329**, 291–309.
89 N. J. Guido, X. Wang, D. Adalsteinsson, D. McMillen, J. Hasty, C. R. Cantor, T. C. Elston and J. J. Collins, *Nature*, 2006, **439**, 856–860.
90 H. M. Davey and D. B. Kell, *Microbiol. Rev.*, 1996, **60**, 641–696.
91 D. J. Stephens and V. J. Allan, *Science*, 2003, **300**, 82–86.
92 D. N. Breslauer, P. J. Lee and L. P. Lee, *Mol. BioSyst.*, 2006, **2**, 97–112.
93 H. Kobayashi, M. Kaern, M. Araki, K. Chung, T. S. Gardner, C. R. Cantor and J. J. Collins, *Proc. Natl. Acad. Sci. U. S. A.*, 2004, **101**, 8414–8419.
94 P. O. Krutzik and G. P. Nolan, *Cytometry A*, 2003, **55**, 61–70.
95 D. A. Veal, D. Deere, B. Ferrari, J. Piper and P. V. Attfield, *J. Immunol. Methods*, 2000, **243**, 191–210.
96 M. B. Elowitz, A. J. Levine, E. D. Siggia and P. S. Swain, *Science*, 2002, **297**, 1183–1186.
97 N. Rosenfeld, J. W. Young, U. Alon, P. S. Swain and M. B. Elowitz, *Science*, 2005, **307**, 1962–1965.
98 E. Korobkova, T. Emonet, J. M. Vilar, T. S. Shimizu and P. Cluzel, *Nature*, 2004, **428**, 574–578.
99 F. K. Balagadde, L. You, C. L. Hansen, F. H. Arnold and S. R. Quake, *Science*, 2005, **309**, 137–140.
100 A. Groisman, C. Lobo, H. Cho, J. K. Campbell, Y. S. Dufour, A. M. Stevens and A. Levchenko, *Nat. Methods*, 2005, **2**, 685–689.
101 S. Cookson, N. Ostroff, W. L. Pang, D. Volfson and J. Hasty, *Mol. Syst. Biol.*, 2005, **1**, 0024.
102 J. Olofsson, H. Bridle, J. Sinclair, D. Granfeldt, E. Sahlin and O. Orwar, *Proc. Natl. Acad. Sci. U. S. A.*, 2005, **102**, 8097–8102.
103 A. Gelman, J. B. Carlin, H. S. Stern and D. B. Rubin, *Bayesian Data Analysis*, 2nd edn, Chapman & Hall, New York, 2004.
104 Y. J. Jiang, B. L. Aerne, L. Smithers, C. Haddon, D. Ish-Horowicz and J. Lewis, *Nature*, 2000, **408**, 475–479.
105 D. J. Wilkinson, *Stochastic Modelling for Systems Biology*, 1st edn, Chapman & Hall/CRC, Boca Raton, 2006.
106 L. L. Looger, M. A. Dwyer, J. J. Smith and H. W. Hellinga, *Nature*, 2003, **423**, 185–190.
107 R. C. Cadwell and G. F. Joyce, *PCR Methods Appl.*, 1994, **3**, S136–140.
108 K. Chockalingam, Z. Chen, J. A. Katzenellenbogen and H. Zhao, *Proc. Natl. Acad. Sci. U. S. A.*, 2005, **102**, 5691–5696.
109 G. F. Joyce, *Annu. Rev. Biochem.*, 2004, **73**, 791–836.
110 R. T. Schimke, *Curr. Top. Cell. Regul.*, 1969, **1**, 77–124.
111 N. Rosenfeld, M. B. Elowitz and U. Alon, *J. Mol. Biol.*, 2002, **323**, 785–793.
112 H. H. McAdams and A. Arkin, *Proc. Natl. Acad. Sci. U. S. A.*, 1997, **94**, 814–819.
113 I. R. Booth, *Int. J. Food Microbiol.*, 2002, **78**, 19–30.
114 A. Aertsen and C. W. Michiels, *Crit. Rev. Microbiol.*, 2004, **30**, 263–273.
115 H. Maughan and W. L. Nicholson, *J. Bacteriol.*, 2004, **186**, 2212–2214.

116 E. M. Ozbudak, M. Thattai, I. Kurtser, A. D. Grossman and A. van Oudenaarden, *Nat. Genet.*, 2002, **31**, 69–73.
117 W. J. Blake, K. A. M, C. R. Cantor and J. J. Collins, *Nature*, 2003, **422**, 633–637.
118 W. J. Blake, G. Balazsi, M. A. Kohanski, F. J. Isaacs, K. F. Murphy, Y. Kuang, C. R. Cantor, D. R. Walt and J. J. Collins, *Mol. Cell*, 2006, **24**, 853–865.
119 D. McMillen, N. Kopell, J. Hasty and J. J. Collins, *Proc. Natl. Acad. Sci. U. S. A.*, 2002, **99**, 679–684.
120 J. Garcia-Ojalvo, M. B. Elowitz and S. H. Strogatz, *Proc. Natl. Acad. Sci. U. S. A.*, 2004, **101**, 10955–10960.
121 B. Ghosh, R. Karmakar and I. Bose, *Phys. Biol.*, 2005, **2**, 36–45.
122 D. L. Cook, A. N. Gerber and S. J. Tapscott, *Proc. Natl. Acad. Sci. U. S. A.*, 1998, **95**, 15641–15646.