Integrative Biology

Dynamic Article Links

Cite this: *Integr. Biol.*, 2011, **3**, 97–108

www.rsc.org/ibiology

PERSPECTIVE

Computational design approaches and tools for synthetic biology†

James T. MacDonald,[ab] Chris Barnes,[bc] Richard I. Kitney,[ad] Paul S. Freemont*[ab] and Guy-Bart V. Stan*[ad]

Received 5th August 2010, Accepted 14th December 2010
DOI: 10.1039/c0ib00077a

A proliferation of new computational methods and software tools for synthetic biology design has emerged in recent years but the field has not yet reached the stage where the design and construction of novel synthetic biology systems has become routine. To a large degree this is due to the inherent complexity of biological systems. However, advances in biotechnology and our scientific understanding have already enabled a number of significant achievements in this area. A key concept in engineering is the ability to assemble simpler standardised modules into systems of increasing complexity but it has yet to be adequately addressed how this approach can be applied to biological systems. In particular, the use of computer aided design tools is common in other engineering disciplines and it should eventually become centrally important to the field of synthetic biology if the challenge of dealing with the stochasticity and complexity of biological systems can be overcome.

Introduction

Synthetic biology involves the application of engineering principles to the science of biology. In the first instance, being able to design and build biological systems is a good test of our current understanding of how these systems work but ultimately the aim is to engineer biological systems to carry out economically valuable tasks. These tasks could include the engineering of bacteria to invade and kill cancer tumours,[1] the cheap synthesis of drugs by metabolic engineering,[2] the production of biofuels,[3] the production of commodity chemicals,[4,5] bioremediation,[6,7] the engineering of biosensors,[8–10] and the rational design of enzymes that catalyse novel reactions.[11,12] Synthetic biology aims to not just tinker with naturally occurring biological systems but to rationally construct complex systems from well understood components in the way that, for example, an electronic circuit may be designed. Given this aim there is a clear need for computer aided design (CAD) tools together with a set of standardised parts and composition rules.

CAD for electronic engineering is a mature field but this is not yet the case for synthetic biology. While the analogy with electronic engineering is useful there are important differences. For example, unlike electronic components, biological components are generally not physically separated from each other making the reuse of modules in the same system more difficult. There is also a lack of a standard modelling framework based on the simple composition of parts. This is due to the difficulty of unambiguously defining "signal/information" exchanges between biological parts. In particular, the following questions still need to be appropriately

Insight, innovation, integration

Our technical ability to physically engineer biological systems is progressing rapidly but our ability to rationally design these systems has not kept pace. Engineers have long had to deal with the types of challenges synthetic biology designers are now confronted with. The use of computer aided design (CAD) tools and modelling is widespread in other engineering disciplines and has enabled the design and manufacture of complex systems with a large number of interacting parts. This review examines the role engineering concepts and techniques can play in synthetic biology and the tools that have already been developed.

[a] *Centre for Synthetic Biology and Innovation, Imperial College London, London, SW7 2AZ, UK*
[b] *Division of Molecular Biosciences, Imperial College London, London, SW7 2AZ, UK. E-mail: p.freemont@imperial.ac.uk; Tel: (+44) 0207 5945327*
[c] *Institute of Mathematical Sciences, Imperial College London, London, SW7 2AZ, UK*
[d] *Department of Bioengineering, Imperial College London, London, SW7 2AZ, UK. E-mail: g.stan@imperial.ac.uk; Tel: (+44) 0207 5946375*

† Published as part of an iBiology themed issue on Synthetic Biology: Guest Editor Professor John McCarthy.

addressed: What are the signals that allow biological parts to be connected?[13] How will the behaviour of individual parts change upon connection? In which context (*e.g.*, chassis) are the interconnected parts going to work?

Another difference with other engineering disciplines is the lack of a catalogue of quantitatively characterised biological components although this is beginning to be addressed.[14] It has been proposed that standardised and comprehensive datasheets be produced to provide quantitative descriptions of biological parts as is commonly used in other engineering disciplines.

The building of complex systems from the interconnection of parts or devices[15] can be significantly facilitated by using a forward-engineering approach relying on the separation of the design from the actual implementation. In this approach, various designs are first optimised and tested *in silico* and their properties are assessed using mathematical analysis and model-based computer simulations.

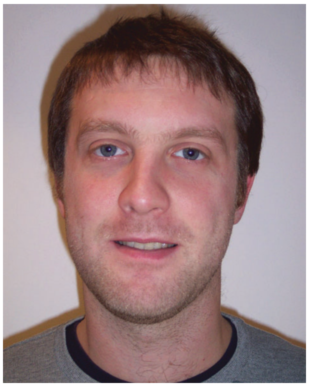
Using a model-based approach, the design of synthetic systems can be made more efficient through the use of CAD tools allowing *in silico* optimisation and testing of the design before the actual wet-lab implementation.

Model-based design of synthetic biology systems

As the requirements of synthetic biological systems have become more complex the need for new modelling methods and software design tools has become more acute. This can refer to both the specification of the structure of the system, *i.e.* constituent parts and their connections, and to the set of

James T. MacDonald

Dr James MacDonald was born in the UK, in 1979. He received his PhD from Birkbeck College, University of London in 2006 in Protein Structure having studied at the School of Crystallography. In 2006 he joined the MRC National Institute for Medical Research to work on computational protein design as a Career Development Fellow as part of the DARPA Protein Design Processes program. Since 2010 he has been a Research Associate at the EPSRC Centre for Synthetic Biology and Innovation at Imperial College London. His current research interests involve developing computational tools and methods for synthetic biology and protein design.



Chris Barnes

Dr Chris Barnes was born in the UK, in 1979. He received his PhD from Imperial College London in 2005 in High Energy Physics while working at Fermilab near Chicago. He moved to the Wellcome Trust Sanger Institute in 2006 to work in statistical genetics and Copy Number Variation (CNV) where he developed tools to perform robust association studies using Copy Number Variants. In 2009 he moved to the Theoretical Systems Biology Group at Imperial and his current research interests cover design, modelling and inference in synthetic and systems biology.

Richard I. Kitney

Professor Richard Kitney was born in the UK, in 1948. He received his PhD in Biomedical Engineering from Imperial College and holds the Chair of Biomedical Systems Engineering at Imperial College. Kitney was Founding Head of the Department of Bioengineering; is Chairman of the Institute of Systems and Synthetic Biology and Co-director of the new EPSRC Centre for Synthetic Biology and Innovation. His research interests over the last 25 years have focussed on modelling biological systems, biomedical information systems and, more recently, synthetic biology. He is a Fellow of The Royal Academy of Engineering; an Academician of the International Academy of Biomedical Engineering; a Fellow of the American Academy of Biomedical Engineering and an Honorary Fellow of both The Royal College of Physicians and The Royal College of Surgeons (UK).

Paul S. Freemont

Professor Paul Freemont was born in the UK, in 1959. He received his PhD in Biochemistry from the University of Aberdeen in 1984. In 1989 he joined the Imperial Cancer Research Fund where he was a Principal Scientist. Since 2001 he has been at Imperial College London where he holds the Chair of Protein Crystallography at Imperial College London and is currently the Head of the Division of Molecular Biosciences and Co-director of the new EPSRC Centre for Synthetic Biology and Innovation. His research interests over the last twenty years have focused on understanding the molecular basis and mechanisms of a number of human diseases including pathogenic infection. He is currently co-leading an initiative in the emerging new field of Synthetic Biology.

parameters describing its kinetics (production and decay rates for example). It has been noted that the complexity of synthetic biological systems (as measured by the number of promoters in the system) being published over the past 10 years seems to have reached a plateau.[16] In part this could be due to limitations in our current ability to mathematically model biological systems accurately.

A mathematical model is a representation of the essential aspects of an existing system (or a system to be constructed) that presents knowledge of that system in a usable form. Mathematical modelling plays a crucial role in the efficient and rational design of complex synthetic biology systems as it serves as a formal mathematical link between the conception and physical realisation of a biological circuit.

It is important to understand that there is no such a thing as "the model." A model can only be defined based on the type of questions that one seeks to answer. These questions determine the level of abstraction or granularity and type of model that should be built. Therefore, building "good" models takes practice, experience and iteration. The goal of a "good" model is to appropriately capture the fundamental aspects of the system while leaving out the details that are irrelevant to the questions that are asked. With this goal in mind, the modelling process needs to take into account the appropriate time and spatial scales that need to be considered, the type of data available, and also the types of simulation and analysis tools to be applied.

The modelling process is considered successful when the obtained model possesses the following characteristics:

- Accurateness: the model should attempt to accurately describe current observations.
- Predictability: the model should allow the prediction of the behaviour of the system in situations not already observed.
- Reusability: the model should be reusable in other, similar cases.
- Parsimony: the model should be as simple as possible. That is, given competing and equally good models, the simplest model should be used.

Guy-Bart V. Stan

Dr Guy-Bart Stan was born in Belgium, in 1977. He received his PhD degree in Analysis and Control of Nonlinear Dynamical Systems from the University of Liège, Belgium in 2005. In 2006, he joined the Control Group of the Department of Engineering at the University of Cambridge, UK, as Research Associate. Since 2009 he is a University Lecturer in Engineering Design for Synthetic Biology Systems in the Department of Bioengineering and the EPSRC-funded Centre for Synthetic Biology and Innovation at Imperial College London. His current research interests are in synthetic biology, systems biology, and more specifically, mathematical modelling, analysis and control of complex biological systems/networks.

When appropriately developed, 'good' mathematical models allow design decisions to be taken regarding how to interconnect subsystems, choose parameter values and design regulatory elements. Model analysis and model design can be seen as two facets of the same coin. Mathematical analysis when realised at a high enough level of generality can provide the modeller with important information about the fundamental limits of a particular class of models and therefore inform the design of the types of model structures that need to be considered if a particular behaviour is sought after.

Types of models (ODEs, PDEs, SDEs, MJPs)

As in other disciplines, synthetic biology systems can be modelled in a variety of ways and at many different levels of resolution and time scales (Fig. 1). For example, we can attempt to model the molecular dynamics (MD) of components of the cell, in which case we attempt to model the individual proteins and other species and their interactions *via* molecular-scale forces and motions. At this scale, the individual interactions between proteins, nucleic acids, and other biomolecules are resolved, resulting in a highly detailed model of the dynamics of these subunits.

Most of the time, however, such a level of resolution is both computationally intractable and too quantitatively inaccurate to answer the questions that one is interested in during the design of synthetic biology systems. Therefore, more coarse-grained models using Ordinary Differential Equations (ODEs), Partial Differential Equations (PDEs), Stochastic Differential Equations (SDEs), or Markov Jump Processes (MJPs) are typically used to model simple synthetic biology circuits (Fig. 2). These coarse-grained models can be used as simplifications as long as their corresponding assumptions are satisfied.

One typical assumption, for example, is homogeneity (either within the cell or at the population level). Under the assumption of a spatial homogeneity, ODE models are most commonly used. In ODEs, each variable (*e.g.*, biochemical species concentration) can only depend on time but not on space (*e.g.*, Fig. 2A, where the variables $p(t)$ and $m(t)$ are only function of time). If spatial variations or inhomogeneities need to be explicitly taken into account, then the modelling may

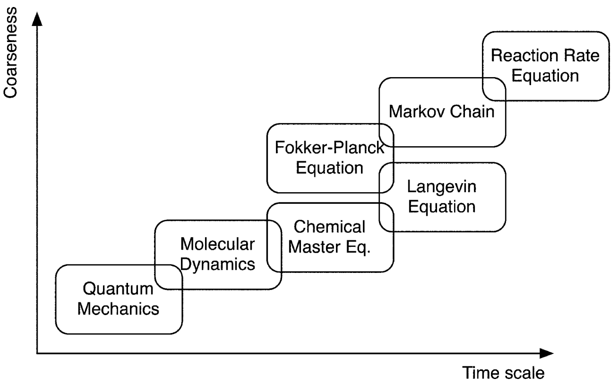


Fig. 1 Levels of abstraction typically used in the modelling process (inspired by Del Vecchio and Murray http://www.cds.caltech.edu/~murray/amwiki/index.php/Supplement:_Biomolecular_Feedback_Systems).

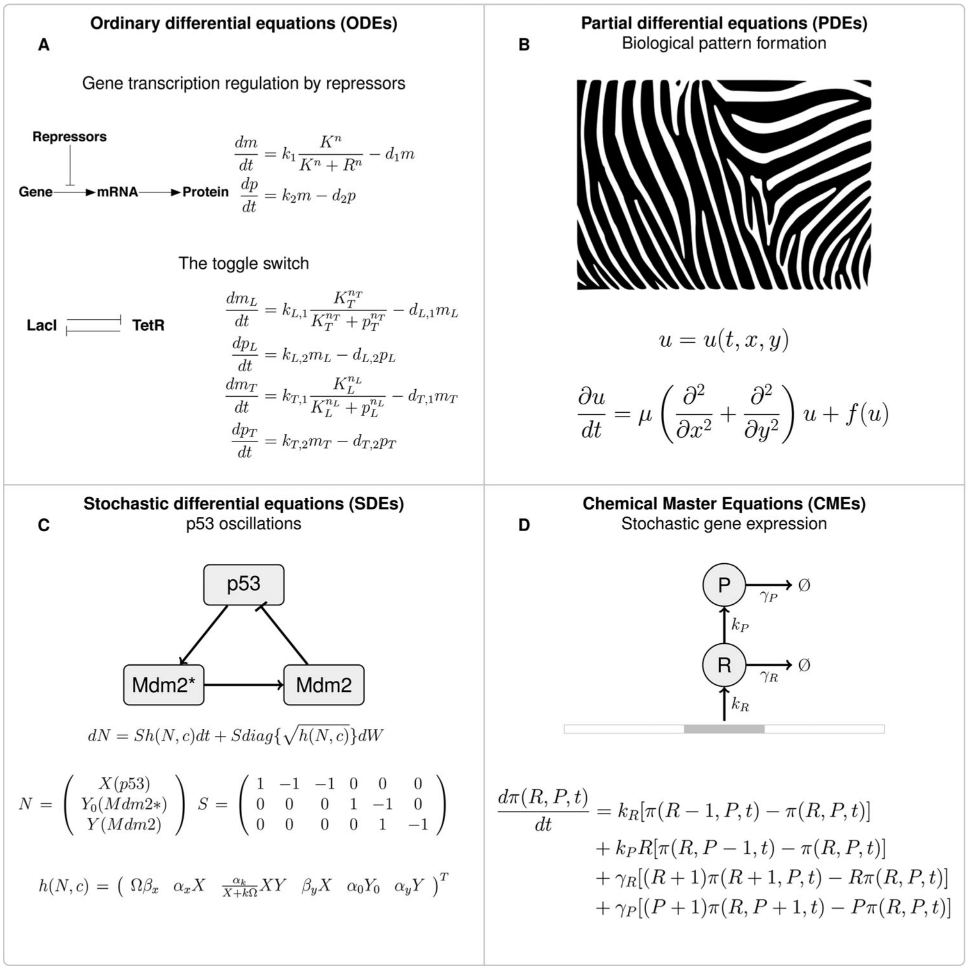

Ordinary differential equations (ODEs)
A
Gene transcription regulation by repressors
Repressors
Gene
mRNA
Protein
$\frac{dm}{dt} = k_1 \frac{K^n}{K^n + R^n} - d_1 m$
$\frac{dp}{dt} = k_2 m - d_2 p$
The toggle switch
LacI
TetR
$\frac{dm_L}{dt} = k_{L,1} \frac{K_T^{n_T}}{K_T^{n_T} + p_T^{n_T}} - d_{L,1} m_L$
$\frac{dp_L}{dt} = k_{L,2} m_L - d_{L,2} p_L$
$\frac{dm_T}{dt} = k_{T,1} \frac{K_L^{n_L}}{K_L^{n_L} + p_L^{n_L}} - d_{T,1} m_T$
$\frac{dp_T}{dt} = k_{T,2} m_T - d_{T,2} p_T$
Partial differential equations (PDEs)
B
Biological pattern formation
$u = u(t, x, y)$
$\frac{\partial u}{dt} = \mu \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) u + f(u)$
Stochastic differential equations (SDEs)
C
p53 oscillations
p53
Mdm2*
Mdm2
$dN = Sh(N,c)dt + Sdiag\{\sqrt{h(N,c)}\}dW$
$N = \begin{pmatrix} X(p53) \\ Y_0(Mdm2*) \\ Y(Mdm2) \end{pmatrix}$ $S = \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$
$h(N,c) = \begin{pmatrix} \Omega\beta_x & \alpha_x X & \frac{\alpha_k}{X+k\Omega} XY & \beta_y X & \alpha_0 Y_0 & \alpha_y Y \end{pmatrix}^T$
Chemical Master Equations (CMEs)
D
Stochastic gene expression
P
R
$\emptyset$
γ_P
k_P
$\emptyset$
γ_R
k_R
$\frac{d\pi(R,P,t)}{dt} = k_R[\pi(R-1,P,t) - \pi(R,P,t)]$
$+ k_P R[\pi(R,P-1,t) - \pi(R,P,t)]$
$+ \gamma_R[(R+1)\pi(R+1,P,t) - R\pi(R,P,t)]$
$+ \gamma_P[(P+1)\pi(R,P+1,t) - P\pi(R,P,t)]$

Fig. 2 The different types of models typically used in systems and synthetic biology. (A) Using ordinary differential equations to model gene transcription regulation by repressors. The model possesses two variables, $m(t)$ and $p(t)$. $m(t)$ represents the concentration of mRNA obtained through transcription of the considered gene and $p(t)$ represents the concentration of protein obtained through translation of the mRNA, at time t. The proposed model contains several parameters: the maximal transcription rate k_1, the repression coefficient K, the Hill coefficient n, the mRNA degradation rate d_1, the protein translation rate k_2, and the protein degradation rate d_2. In this model, R is considered as an external input representing the concentration of transcriptional repressor. Based on this simple model for gene transcription regulation by repressors a more complex model of a toggle switch (composed of two mutually repressing genes) was designed and experimentally built in E.coli.[115] (B) Using partial differential equations to model biological pattern formation using the reaction-diffusion equation. The model considered here possesses a single variable $u(t,x,y)$ which depends on time t and on the planar spatial coordinates x and y. The change in local concentration of each chemical species over time is a sum of a term proportional to the Laplacian of the local concentration (to account for diffusion) and a function of the local concentration of the chemical species (to account for chemical reactions).[116–118] The parameter μ is the diffusion coefficient (a measure of how fast molecules diffuse from regions of high concentration to regions of low concentration) while the function $f(u)$ is defined based on the specific details of how the molecules in the system react with each other. (C) Stochastic differential equations for the p53 oscillation model,[119] where dW is the increment of the Wiener process (also known as Brownian motion – the stochastic element of the equation) and X, Y_0 and Y represent the numbers of p53, Mdm2 precursor and Mdm2 respectively. Ω represents the volume of the system and α, β represent the production and degradation rates. As with the previous types of models described above, the variables X, Y_0 and Y vary continuously (*i.e.* they cannot take discrete values). (D) Chemical master equation for the simple stochastic gene expression model[120] where R and P represent the number of RNA and protein molecules respectively, and $\pi(R, P, t)$ is the probability of observing R RNA and P protein molecules at time t. In contrast to the previous types of equations, R and P are discrete integer values and change stochastically in discrete jumps over time according to the parameters of the model. The parameters k_R and k_P are the production rates of mRNA and protein, while γ_R and γ_P are degradation rates of mRNA and protein respectively.

require the use of PDEs where the variables can depend on time and on space (see Fig. 2B where the variable $u(t,x,y)$ is a function of time and of the spatial coordinates x and y (in this model we only consider two spatial coordinates, *i.e.*, u is assumed to be evolving on a plane)).

Another distinction occurs based on the type of modelling framework used, *i.e.*, deterministic *versus* stochastic. The deterministic framework is appropriate to describe the mean behaviour (*i.e.*, averaged across a large number of molecules) of a biochemical system. Deterministic models implicitly assume that the underlying quantities, *i.e.*, concentrations or molecule numbers, vary in a deterministic and continuous fashion. On the other hand, the stochastic framework takes into account the random interactions of biochemical species.[17,18] More specifically, stochastic models are used to mathematically capture stochastic variations and noise inherent in biological systems. They are typically used when the number of species involved is small and stochastic effects can no longer be "averaged out", such as is the case for transcription factors, which, in certain circumstances, can be expressed at low levels (*i.e.* a few tens of molecules), or for DNA, for which a single copy may exist in the cell. The analysis of such stochastic models can occasionally be realised by mathematically deriving the most important statistical moments (*e.g.*, mean and variance) though this is rarely possible when the corresponding complexity (non linearity, high dimensionality) of the derived models is high as is typically found in realistic biological modelling.

Two main types of stochastic models (MJPs and SDEs) are typically used in the literature to represent stochastic systems. MJPs and SDEs differ in how the number of molecules is treated. MJPs typically use a discrete state space and, in the context of biochemical reactions, are referred to as Chemical Master Equations (CMEs) (Fig. 2D). Exact numerical simulations of CMEs can be achieved using stochastic simulation algorithms (SSA) such as the Gillespie algorithm.[19] An alternative approach approximates CMEs with a continuous state space for the species number. Under this approximation, CMEs can be transformed into SDEs (Fig. 2C) or equivalently the Fokker–Planck equation. The time evolution of these stochastic models can be simulated using numerical methods such as the Euler–Maruyama algorithm.[20] Typically simulation of MJPs is very computationally expensive, especially if the number of different biochemical species in the model is large or the reaction rates are high. Numerical simulations from SDEs are less computationally expensive but SDEs do not accurately represent chemical processes at low numbers of molecules. For this reason there has been an effort to develop hybrid models that combine ODE, SDE and MJP modelling into a single framework where each species is modelled most appropriately while minimising computational time. One approach to obtain such hybrid models is to dynamically partition the system into "fast" and "slow" subsets. The "fast" subsystem models reactions that occur frequently using continuous Markov processes while the "slow" subsystem models reactions that occur infrequently using discrete Markov processes.[21]

Choosing the appropriate level of resolution in answering the modelling question must also take into account the type of experimental data that are or can be made available. Indeed, a model with a very high resolution might not be very useful if the amount and quality of data available to estimate the model parameters is low. This also raises the question of identifiability of models from data, *i.e.*, what is the minimal amount of information necessary in order to be able to estimate the parameters of a given model?[22] This question is not only important for the appropriate design of models but also for the design of experimental protocols according to the minimal information required to answer the identifiability question.[23–26]

Parameter inference and model selection

Often models contain unknown parameters such as production and degradation rates, binding affinities and other rate constants that are difficult to directly measure experimentally. Additionally there may be a number of competing models for a given mechanism. Given experimental data, which usually consist of time course measurements of the model species under some conditions *in vivo* or *in vitro*, both these aspects can be addressed. The first question refers to parameter inference (also known as parameter estimation or model calibration) and the second is known as model selection.

Parameter inference can be posed as an optimisation problem where the objective function is some measure of how well model simulations match the experimental data. In statistics the maximum likelihood approach can be used with the likelihood function as the objective.[27] In physics and engineering often a weighted sum of squares statistic is used, which is equivalent to assuming normally distributed errors. The parameter estimates are those that maximise the objective function. Usually an exhaustive enumeration of parameter combinations is unfeasible and thus optimisation algorithms try to move towards the maximum in an informed way. Deterministic methods such as gradient ascent (or gradient descent depending on whether one wishes to minimise or maximise the objective function) use the local neighbourhood of parameter space to determine the direction of the next optimisation move.[28] These methods are efficient but can produce suboptimal solutions if the objective function is multimodal and has local maxima. Stochastic optimisation methods such as simulated annealing try to avoid this problem by allowing the possibility of downhill moves in parameter space thus providing the ability to move out of local maxima. Other examples of optimisation techniques include genetic and evolutionary algorithms.[29]

Model selection can also be performed in the maximum likelihood framework. However, models with more parameters will always fit the data better than models with fewer parameters and this must be accounted for. In nested model comparisons (where one of the two models being compared can be transformed into the other model by imposing a set of constraints on the parameters, thereby reducing the number of free parameters) the Likelihood ratio test (LRT) can be used. For non-nested models the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC)[30] can be used which penalise the log likelihood by the number of parameters.

Bayesian methods combine the likelihood function with a prior distribution and give a probability distribution over the estimated parameters known as the posterior distribution.[31] As such they are an alternative to optimisation and while being computationally more expensive can provide more information on the correlation structure of the parameters and whether there are multiple (sub) optimal regions of parameter space. This is important when many combinations of parameters can explain the data equally well and point estimates can be misleading.[32] One additional advantage of Bayesian methods is that parameter estimation and model selection are handled in the same framework which means competing models can be assigned posterior probabilities.

Mathematical analysis of models

Once models have been created and all their parameters determined, they can then be numerically simulated and mathematically analysed in order to characterise their behaviour with respect to specific criteria of interest, *e.g.* which parameters have the "largest impact" on the model behaviour? (this is typically done through a parameter sensitivity analysis as we will explain in more detail below); what types of behaviours can be expected from the considered model and how can this behaviour change when the parameter values are modified? (bifurcation analysis); how robust is the behaviour/performance predicted by the model to structural and dynamic perturbations? (*e.g.* is the behaviour robust to nodes or edges removals in a designed genetic or metabolic network?; is the behaviour robust to unmodelled dynamic disturbances acting as inputs to some of the nodes of this network?)

The mathematical analysis of models provides the designer with important information regarding the possible types of behaviours that can be expected from the implementation of a particular model, and most importantly, the performance limits that constrain the considered design.[33,34]

In the following sections, we briefly describe the main model analysis techniques commonly used.

Bifurcation analysis. The goal of bifurcation analysis is to characterise regions in the parameter space in which the steady-state behaviour of the system is qualitatively the same. At the boundaries of such regions, a qualitative change in the steady-state behaviour occurs, *e.g.*, increasing the value of a single parameter while keeping the others constant, leads to a transition from a unique asymptotically stable equilibrium point to an asymptotically stable periodic oscillation (*e.g.* through a Hopf bifurcation). The characterisation of these regions provides crucial information for the design as it allows the identification of the parameter regions within which a desired behaviour can be obtained (*e.g.* for the design of an oscillator[35]).

Numerical continuation analysis tools such as XPPAUT[36] or MATCONT[37] can be used to perform a bifurcation analysis of a given model once an initial guess for its associated equilibrium points is known.

Parameter sensitivity analysis. The goal of a parameter sensitivity analysis is to determine the influence of the parameters on the performance of the model.[38–40] Loosely speaking, a parameter sensitivity analysis consists of estimating the variation of performance induced by a fixed parameter variation. Local sensitivity analysis is used to analyse the effect of small parameter perturbations and is usually carried out by computing partial derivatives of the output or performance function with respect to the model parameters. On the other hand global sensitivity analysis is used to analyse the change in performance over the whole admissible range of parameter variations. This is of particular importance in biological models for which parameters can vary within large intervals depending on their meaning. Examples of sensitivity analysis software tools developed specifically for biological systems include BioSens (http://www.chemengr.ucsb.edu/~ceweb/faculty/doyle/biosens/BioSens.htm), SensSB[41] and SBML-SAT.[42]

Robustness analysis. Although several designs can lead to the same behaviour, their ability to withstand structural

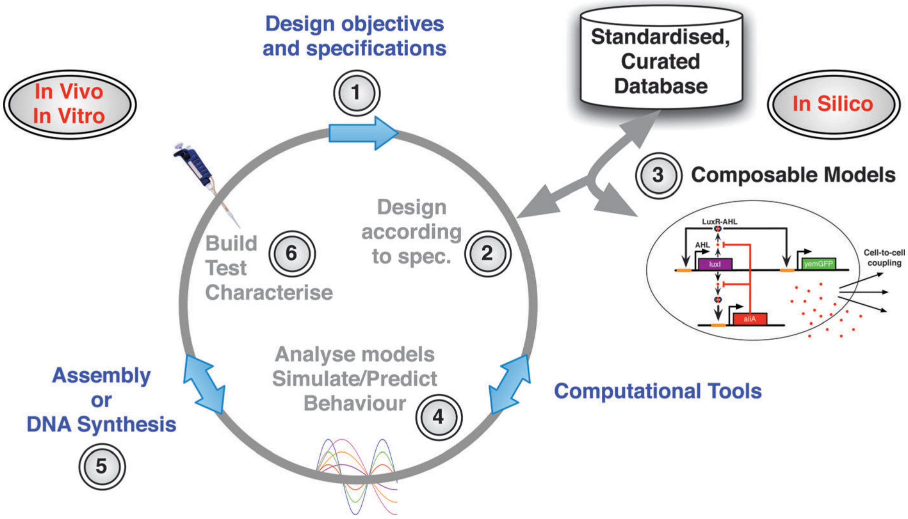


Fig. 3 The engineering design cycle (inspired by Chandran *et al.*).[121] Design starts *in silico* and proceeds iteratively along the cycle with the following workflow: (1) The design begins by the definition of the design objectives, *i.e.*, what dynamical phenotypic behaviour is sought after? with what properties (*e.g.*, robustness, yield, time response)? under which constraints (*e.g.*, upper limits on output variability, chassis and environment specifics)? (2) Based on these design specifications, different possible designs are envisioned.[43,122] These designs differ by the choice of components or parts used, and the way these parts are interconnected. (3) For each design a set of models is constructed, ideally by using a library of composable models for the parts and considering the interconnection rules imposed by the considered design. (4) Model-based analysis and simulations are performed to assess *in silico* the performance of each design with respect to the initial design specifications. At this stage, *in silico* analysis and optimisation allows for the search for parameter values leading to the desired behaviour. Furthermore, using robustness analysis, different models can be assessed with respect to their ability to withstand structural and dynamic perturbations. This typically leads to *in silico* iterations whose goal is to eventually select a subset of design candidates for the wet-lab implementation. (5) The candidate *in silico* designs are "translated/compiled" into DNA sequences for *in vitro* or *in vivo* wet-lab implementation. (6) The *in vitro* or *in vivo* implementation is tested and characterised to yield biological data that are then fed back into the model, thereby closing the engineering design loop.

(*e.g.* removal of nodes or edges in a network) or dynamic perturbations (*e.g.* unmodelled dynamics resulting from circuit-chassis interactions or from cross-talk) may be very different.[43,44] Provided enough information is known about the perturbation (*e.g.*, at least the knowledge of an upper bound on the energy of the perturbation), robust analysis of the models[45] allows the assessment of their relative immunity to such perturbation. In particular, this step can prove very informative for the robust design of models, and established multivariable robust control techniques can be used to that effect in the design process.[45] A widely used tool for this type of analysis is the Robust Control Toolbox (http://www.mathworks.com/products/robust/) for Matlab. The SBML-SAT[42] tool mentioned above also implements algorithms for robustness analysis, as does the BIOCHAM[46] modelling environment.

A forward-engineering approach to the design of synthetic biology systems

To ensure that models can be efficiently reused in the design of increasingly complex systems, the models must be composable and the behaviour resulting from their interconnection must be predictable from the behaviour of the components and the way they are interconnected. This last step is the goal of systems and control theory and has led to many developments in these fields.[47–50] The definition of composable design models and of their appropriate interconnection is a key step in enabling the design of complex systems from the interconnection of several parts. This step is at the core of the forward-engineering of synthetic biology systems in which, aided by computer-aided-design tools, the designer uses the workflow described in Fig. 3.

The use of CAD tools makes the iterations between the different steps represented on Fig. 3 easier and more efficient. CAD tools can be supplemented by Graphical User Interfaces (GUIs), that allow for the construction of devices and systems by graphically interconnecting components on a canvas and building the corresponding models in the background. Ultimately, these tools could also directly predict the DNA sequence that is required for the implementation of the designed model *in vivo* or *in vitro*.

The engineering design cycle (Fig. 3) enables the efficient design of complex synthetic biology systems using a forward-engineering approach very similar to the one successfully used in many other engineering disciplines. In particular, in this approach, system design and system fabrication are separated (Fig. 4). The advantage of such a separation is that it saves time, money and effort since the main burden of the design can be done *in silico*.

Automated system design tools

A number of computational methods have already been developed to facilitate system design. To date these have concentrated mainly on the design of small transcriptional circuits. Some have been packaged into downloadable software tools. Common to most of the methods developed so far is the use of ordinary differential equations (ODEs) to model the dynamics of the system (though some can also handle stochastic dynamics). These computational methods differ in the way the networks are parameterised, how the dynamics are approximated and how the optimisation is formulated.

One approach is to use an evolutionary algorithm[29] and this has been achieved for a fixed network topology[51] and for simultaneous optimisation of topology and kinetic parameters.[52] In the study by Francois and Hakim[52] seven possible types of reaction were considered including translation, activation and repression and post-translational modifications. The growth phase of the evolutionary algorithm either adds a gene, a new reaction or modifies a dynamical parameter. The selection phase calculates fitness scores for the population of networks and removes the low ranking ones. Using this method the authors investigate the possible topologies that give rise to a bistable switch and to an oscillator.

Genetdes[53] attempts to find an optimal network topology and kinetic parameters given specified target dynamics. Genes have an associated promoter that allows repression and activation by two transcription factors with the interaction properties defined by a library designed to give rise to common logic operations. The optimisation is performed using simulated annealing with moves in model space including addition and deletion of both genes and regulatory interactions and the change of kinetic parameters. The optimal model can be output in SBML format[54] for analysis in other programs. A C program and many logic gate examples with varying complexity are available online (http://www.enseignement.polytechnique.fr/profs/biochimie/Alfonso.Jaramillo/genetdes.html).

RoVerGeNe[55] takes an existing network topology as input and desired dynamic behaviour is expressed through a set of constraints. Regulation terms in the dynamical equations are approximated using piecewise linear functions, which makes analysis more efficient in comparison to the full nonlinear ODE model. Desired network behaviour is expressed through Linear Temporal Logic (LTL) and, given a parameter region, RoVerGeNe uses abstraction and model checking[56] to assess whether the region is able to satisfy the constraints and therefore the desired dynamical behaviour. If the region is valid for the given constraints then it finds sets of parameters that give rise to the specified dynamics. The method was used to tune a synthetic transcriptional cascade and MATLAB code is available online (http://iasi.bu.edu/∼batt/rovergene/rovergene.htm).

The OptCircuit framework[57] uses a predefined list of promoters, protein molecules and inducers (small molecules that can interact with transcription factors to alter gene expression). Using this framework, systems are built to maximise an objective function derived from the desired dynamics. Here the full dynamics are approximated under the assumption that fast reactions (rate constants on the order of seconds) are in equilibrium.[58] The optimisation is formulated as a mixed integer dynamic optimisation problem[59] and can be applied to both system topology and kinetic parameters. The authors demonstrate the method on the design of a toggle switch, a genetic decoder and a concentration band detector.

An entirely different approach is to develop languages that can then be compiled into sequences of standard biological parts.[60,61] In the simplest example, GenoCAD,[60] this consists of a context-free grammar that enforces a set of production

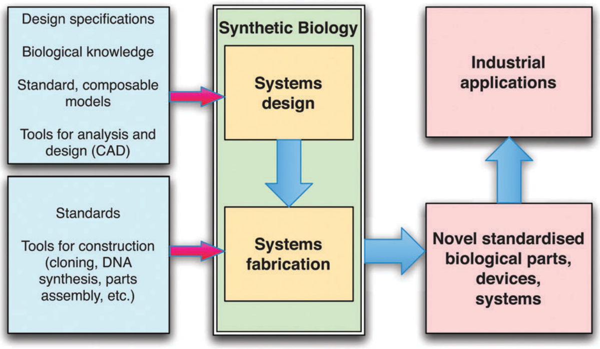

Design specifications
Biological knowledge
Standard, composable models
Tools for analysis and design (CAD)
Standards
Tools for construction (cloning, DNA synthesis, parts assembly, etc.)
Synthetic Biology
Systems design
Systems fabrication
Industrial applications
Novel standardised biological parts, devices, systems

Fig. 4 Using the engineering design cycle in Fig. 3, system design can be separated from system fabrication. The use of CAD tools allows for the *in silico* design, analysis and optimisation prior to wet-lab implementation (inspired by Heinemann and Panke).[123]

rules that ensures that the user will produce a biologically valid construct. A more elaborate approach is to allow the expression of logical interactions between (possibly undetermined) biological parts using a programming language specifically developed for that purpose (the only example of this currently is the GEC language[61]). A compiler can then take the program and a database of standard biological parts with associated properties, and produce a set of designs that satisfy the requirements defined in the program. The resulting set of solutions can then be simulated for further analysis and selection.

Despite the achievements of the methods described above there is still need for further development in the design of systems other than genetic circuits. Sophisticated applications will require systems combining genetic, signalling and metabolic circuits and the design and optimisation tools that can handle this complexity.

Graphical user interface design tools

In parallel to the development of automated design methods, a number of tools have been developed to enable users to design genetic circuits in a graphical user interface (GUI) by, for example, dragging and dropping components on a canvas to compose a device by interconnecting selected parts. In this context, the main burden of the design process falls on the user's experience and intuition. However, many of these tools have either integrated simulation modules or can call external programs in order to help the user verify that their graphically designed circuits are predicted to perform as expected. Some of the tools are available as online web applications and some as downloadable programs that run on the user's personal computer. The available tools also vary in that some are designed to allow the user to directly design DNA constructs while others merely allow users to design circuits by defining interactions between constituent parts without specifying how this higher-level abstraction would be implemented at the DNA sequence level.

One of the first drag and drop graphical design tools to be made available was BioJADE[62] which provided the user with the ability to both access databases of parts and run simulations of the designed circuits using TABASCO.[63]

GenoCAD, implemented as an online web application,[64] enables users to graphically design a DNA construct from constituent parts (either user uploaded or from a pre-loaded library) using a formal context-free grammar. The resulting DNA sequence can then be automatically generated and downloaded by the user. Another web application for graphically designing DNA constructs is SynBioSS Designer[65] which is part of the broader downloadable software suite[66] called SynBioSS. SynBioSS Designer can export designed models in SBML format[54] for analysis. The downloadable portion of the software, SynBioSS DS (Desktop Simulator) allows users to then load the models and run hybrid stochastic simulations.

Given that computational design for synthetic biology is still in its infancy and does not yet possess well-established standard methods, some tools have incorporated a modular approach to their software architecture to allow for future extension. The developers of TinkerCell[67] (the successor to a software package called Athena by the same authors) have created a modular plug-in architecture to permit third-party developers access to its rich Application Programming Interface (API), GUI and Python scripting features. ProMoT[68] is another example of a flexible, modular GUI tool for CAD with third-party add-ons.[13]

Finally, ClothoCAD (http://www.clothocad.org/) is an ambitious large-scale project to develop a modular integrated software platform for synthetic biology systems providing graphical sequence editing, data management, algorithm management tools and a plug-in system. This software includes the ability to interface with liquid handling robots for automated production.

Computational biomolecular design

It is likely that there already exists in Nature a large proportion of the parts a bio-designer would require in order to build their desired system,[69] however it may sometimes be necessary to design new parts at the nano-scale level. These new parts could be proteins with modified or novel functions,[10–12,70–76] synthetic ribosome binding sites with a specific translation initiation rate,[77] synthetic promoter sequences with specific transcription rates,[78,79] or three-dimensional biomolecular structural scaffolds that can be finely controlled at the atomic-level.[80–82]

Computational protein design is generally split into two coupled problems.[83] Firstly, one needs to find or generate a backbone scaffold with a high degree of 'designability.' This can be generated artificially or derived from experimentally solved protein structures. The second problem, often referred to as the 'inverse folding problem', is to find amino acid sequences that are able to fold into the required backbone structure, *i.e.* one needs to minimise the free energy of folding ($\Delta G_{\mathrm{folding}}$). In order to do this one would need to evaluate the potential energy of a given sequence over all backbone and sidechain conformations in order to calculate the partition function which is, of course, computationally intractable unless one knows what the relevant alternative states are. Therefore most popular methods take a 'positive design' approach where the aim is to find sidechain rotamers that minimise the potential energy of a given backbone structure with some simple approximate heuristic to account for alternative states. This problem can either be solved by using a stochastic method such as simulated annealing[84,85] or by using a deterministic method such as Dead-end elimination[86–88] but both methods tend to give similar results.

Using computational protein design methods it has now been shown to be possible to rationally engineer enzymes with novel catalytic functions that do not exist in Nature,[11,12,89] albeit with much lower activities than for naturally occurring enzymes and for relatively simple reactions. A key requirement of computational enzyme design is the ability to place the functional groups of sidechains with atomic-level accuracy.[90,91]

According to Transition State Theory, the activated complex (*i.e.* the transition state) is in quasi-equilibrium with its reactants and the rate of the reaction is proportional to the concentration of the transition state. Therefore the aim of an enzyme designer is to engineer a protein that can specifically bind the substrates and catalyse the desired reaction by lowering the free energy of the transition state.

One approach to the design of novel enzymes is to design the enzyme from the inside out. In this approach one must determine the geometric constellation of chemical functional groups that are likely to stabilise the transition state, then work outwards to find compatible sidechain conformations (*e.g.* by an inverse rotamer method), and finally find compatible backbone scaffolds.[11,12,90] To determine the possible molecular interactions that are capable of stabilising the transition state one could either carry out quantum chemistry calculations[92,93] (as would be necessary for a novel reaction not found in Nature) using programs such as GAUSSIAN (http://www.gaussian.com/), or by studying the structures of known enzymes. In the case of designing a novel ''theozyme'' (theoretical enzyme), the transition state of the reaction must first be determined (*e.g.* using the Hartree–Fock method, density functional theory or semi-empirical methods) and then the geometry of stabilising functional groups placed around this transition state can be optimised.[93]

DNA assembly

Once the synthetic biology designer has produced a design *in silico* it is then necessary to translate the *in silico* design into something that can be tested experimentally. In the vast majority of cases this will involve some form of assembly of smaller sections of DNA to form a larger section of DNA that implements the design. This can range in scale from assembling a gene optimised to efficiently express a protein with a given amino acid sequence ,[94–96] to assembling a set of genes to form a pathway, and up to assembling an entire bacterial genome.[97] For PCR-based assembly methods computational tools can help design reliable thermodynamically-balanced primers with minimised hairpin formation.[94,95] Furthermore, computational algorithms can help eliminate human error as well as minimise the time and cost of experiments with the ultimate aim of high-throughput fully automated assembly by liquid handling robots.[98]

At the pathway assembly level the BioBrick standard provides a set of simple rules to allow physical composition of parts by a restriction enzyme-based mechanism and is accompanied by a registry of user submitted parts (http://partsregistry.org/Main_Page). This is highly used due to its simplicity and due to the requirements of the popular International Genetically Engineered Machine student competition (iGEM) for competing teams to submit parts to the registry (http://www.igem.org). The disadvantages of this method include its serial step-wise assembly of parts making it more time-consuming to assemble large constructs than parallel one-pot methods, its non-combinatorial nature and the fact that it leaves an 8 base pair ''scar'' region between the assembled parts. For these reasons, new and improved DNA assembly methods (such as the isothermal assembly (Gibson)[99] and Golden-gate[100] methods) are vitally important for synthetic biology, and constitute a rapidly developing field (see review by Ellis *et al.*, also in this issue[101]).

Future developments

Synthetic biology holds the promise of cleaner, cheaper and less energy intensive technology. Its potential applications include energy production, bioreactors, biosensors, medical devices, smart materials, and perhaps even terraforming planets to make hostile environments habitable by humans.

In order to fulfil this potential a number of key enabling technologies need to be improved and developed.[102] One of the most important is the creation of a registry of professionally characterised biological parts, which is one of the main goals of BIOFAB (http://www.biofab.org/; an allusion to the ''chip fab''—the technology for the engineering of semiconductor chips). In contrast, the BioBrick parts deposited in the Registry of Standard Biological Parts (http://partsregistry.org/) are of varying quality and often uncharacterised. Without a catalogue of standard parts with well characterised behaviours in the context of known chassis the dream of being able to rationally engineer biological systems will be seriously impeded.[14] Advancements in other key enabling technologies are also required such as DNA assembly techniques, automated fabrication technology, microfluidics, more sophisticated modelling methods, simplified synthetic chassis,[103] standards for data exchange and better ways to measure the behaviours of biological systems. As these technologies improve and mature, synthetic biologists will be able to spend more time thinking about engineering whole systems rather than worrying about low-level implementation details.

Modelling biological systems is especially difficult given the degree of unpredictable crosstalk between components, the stochasticity, and the context dependence of parameters. Future automated design methods will have to find ways of dealing with these issues, perhaps by designing heterogeneous redundancy into the system and developing design methodologies where stochastic variations and crosstalk are features rather than undesired behaviours. Crosstalk and other unexpected interactions could be minimised by the development of libraries of orthogonal parts and perhaps by the use of a ''minimal cell'' (a cell containing only those functions necessary for basic life) as the chassis.[104] For example, the design and use of orthogonal ribosomes[105] could help reduce the risk of unintended crosstalk by allowing the creation of gene expression pathways that are unreadable by the host's

Table 1 A non-exhaustive list of software for synthetic biology design

Software name	Website
Automated circuit design	
Genetdes[53]	http://soft.synth-bio.org/genetdes.html
RoVerGeNe[55]	http://iasi.bu.edu/~batt/rovergene/rovergene.htm
OptCircuit[57]	http://maranas.che.psu.edu/
GEC[61]	http://research.microsoft.com/en-us/projects/gec/
GUI circuit design	
BioJade[62]	http://web.mit.edu/jagoler/www/biojade/
GenoCAD[64]	http://www.genocad.org/
ProMoT[68]	http://www.mpi-magdeburg.mpg.de/projects/promot/
SynBioSS[66]	http://www.synbioss.org/
ClothoCAD	http://www.clothocad.org/
TinkerCell[67]	http://www.tinkercell.com/
CellDesigner[124]	http://www.celldesigner.org/
Biomolecular design	
Rosetta[125]	http://www.rosettacommons.org/
ORBIT[71]	http://www.mayo.caltech.edu/index.html
Dezymer[70]	http://www.biochem.duke.edu/modules/biochem_hellinga_lab/index.php?id = 1
RBS Calculator[77]	http://voigtlab.ucsf.edu/software/
caDNAno[82]	http://cadnano.org/
PRODART[81]	http://mathbio.nimr.mrc.ac.uk/wiki/Software#De_Novo_Backbone_Scaffolds
DNA assembly	
GeneDesigner	https://www.dna20.com/genedesigner2/
GeneDesign	http://baderlab.bme.jhu.edu/gd/

native cellular machinery. This creates the possibility of booting up a number of orthogonal "parallel operating systems" within the cell that are freed from the constraint of also having to carry out the cell's essential functions.[106]

While a large number of computational design tools have been developed (Table 1), most existing automated design methods have concentrated on the design of synthetic transcriptional networks. However potential applications are much broader than this.[107] For example, one may wish to re-create, in an automated way, a biosynthetic pathway that exists in a slow growing organism in a faster growing organism with a higher yield. Such a method would have the ability to automatically determine the reaction steps required to synthesise a given small molecule, retrieve suitable enzymes from online databases and assemble the genes into a *de novo* biosynthetic pathway.[108]

New data exchange standards for synthetic biology will greatly assist the task of automated design and is an area of active development. SBOL (Synthetic Biology Open Language; http://www.sbolstandard.org) is a collaborative effort to develop standards for the exchange of biological parts data including standard visual representations of the parts. The aim is to facilitate the integration of different software tools and allow the easy transfer of information between synthetic biologists.

To accompany methods of forward-engineering systems there will also be a need for debugging tools such as methods for network reconstruction from data[23,26] to test if the (dynamical) structure of the wet-lab implemented synthetic biology circuit corresponds to the one originally intended by its designer.

Finally, given all the uncertainties in modelling biological systems it may be necessary to optimise *in silico* designs with directed evolution methods. Such an approach would be useful both for optimising individual designed components, such as designed enzymes,[109,110] and for optimising pathways.[111–113] Developments in microfluidics could help increase throughput dramatically[114] and allow the development of the lab-on-a-chip technology.

Despite the significant progress in synthetic biology in the past few years it is clear there is still a long way to go before we can routinely engineer biological systems. It is also clear that developments in our ability to design biological systems *in silico* will play an increasingly central role in this process.

Acknowledgements

J.T.M is funded by the EPSRC, UK. C.B. is funded by BBSRC, UK. G.-B.V.S is funded by EPSRC Centre for Synthetic Biology and Innovation of which R.I.K. and P.S.F. are Co-Directors. Michael Stumpf is thanked for useful discussions.

References

1 J. C. Anderson, E. J. Clarke, A. P. Arkin and C. A. Voigt, *J. Mol. Biol.*, 2006, **355**, 619–627.
2 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.
3 S. K. Lee, H. Chou, T. S. Ham, T. S. Lee and J. D. Keasling, *Curr. Opin. Biotechnol.*, 2008, **19**, 556–563.
4 A. L. Marshall and P. J. Alaimo, *Chem.–Eur. J.*, 2010, **16**, 4970–4980.
5 T. S. Bayer, D. M. Widmaier, K. Temme, E. A. Mirsky, D. V. Santi and C. A. Voigt, *J. Am. Chem. Soc.*, 2009, **131**, 6508–6515.
6 D. R. Lovley, *Nat. Rev. Microbiol.*, 2003, **1**, 35–44.
7 V. de Lorenzo, *Curr. Opin. Biotechnol.*, 2008, **19**, 579–589.
8 D. E. Nivens, T. E. McKnight, S. A. Moser, S. J. Osbourn, M. L. Simpson and G. S. Sayler, *J. Appl. Microbiol.*, 2004, **96**, 33–46.
9 D. M. Rawson, A. J. Willmer and A. P. Turner, *Biosensors*, 1989, **4**, 299–311.
10 L. L. Looger, M. A. Dwyer, J. J. Smith and H. W. Hellinga, *Nature*, 2003, **423**, 185–190.

11 L. Jiang, E. A. Althoff, F. R. Clemente, L. Doyle, D. Rothlisberger, A. Zanghellini, J. L. Gallaher, J. L. Betker, F. Tanaka, C. F. Barbas, 3rd, D. Hilvert, K. N. Houk, B. L. Stoddard and D. Baker, *Science*, 2008, **319**, 1387–1391.

12 D. Rothlisberger, O. Khersonsky, A. M. Wollacott, L. Jiang, J. DeChancie, J. Betker, J. L. Gallaher, E. A. Althoff, A. Zanghellini, O. Dym, S. Albeck, K. N. Houk, D. S. Tawfik and D. Baker, *Nature*, 2008, **453**, 190–195.

13 M. A. Marchisio and J. Stelling, *Bioinformatics*, 2008, **24**, 1903–1910.

14 B. Canton, A. Labno and D. Endy, *Nat. Biotechnol.*, 2008, **26**, 787–793.

15 D. Endy, *Nature*, 2005, **438**, 449–453.

16 P. E. M. Purnick and R. Weiss, *Nat. Rev. Mol. Cell Biol.*, 2009, **10**, 410–422.

17 N. G. v. Kampen, *Stochastic Processes in Physics and Chemistry*, Elsevier, Amsterdam; Boston; London, 3rd edn, 2007.

18 D. J. Wilkinson, *Nat. Rev. Genet.*, 2009, **10**, 122–133.

19 D. Gillespie, *J. Comput. Phys.*, 1976, **22**, 403–434.

20 P. E. Kloeden and R. A. Pearson, *The ANZIAM Journal*, 1977, **20**, 8–12.

21 Y. N. Kaznessis, *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, 2009, **2009**, 6408–6411.

22 L. Ljung, *System Identification: Theory for the User*, Prentice Hall PTR, 2nd edn, 1999.

23 J. Gonçalves and S. Warnick, *IEEE Trans. Autom. Control*, 2008, **53**, 1670–1674.

24 E. D. Sontag, *Essays Biochem.*, 2008, **45**, 161–176.

25 E. Sontag, A. Kiyatkin and B. N. Kholodenko, *Bioinformatics*, 2004, **20**, 1877–1886.

26 Y. Yuan, G.-B. Stan, S. Warnick and J. Goncalves, 49th IEEE Conference on Decision and Control, Atlanta, Georgia, USA, 2010.

27 D. R. Cox, *Principles of Statistical Inference*, Cambridge University Press, Cambridge; New York, 2006.

28 J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

29 A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*, Springer, 2008.

30 K. P. Burnham and D. R. Anderson, *Technometrics*, 2008, **45**, 181–181.

31 A. Gelman, *Bayesian Data Analysis*, Chapman & Hall/CRC, Boca Raton, Fla, 2nd edn, 2004.

32 R. N. Gutenkunst, J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers and J. P. Sethna, *PLoS Comput. Biol.*, 2007, **3**, 1871–1878.

33 I. Lestas, J. Paulsson, N. E. Ross and G. Vinnicombe, *IEEE Trans. Circuits Syst. I. Regul. Pap.*, 2008, 189–200.

34 I. Lestas, G. Vinnicombe and J. Paulsson, *Nature*, 2010, **467**, 174–178.

35 J. Stricker, S. Cookson, M. R. Bennett, W. H. Mather, L. S. Tsimring and J. Hasty, *Nature*, 2008, **456**, 516–519.

36 B. Ermentrout, Author, A. Mahajan and Reviewer, *Appl. Mech. Rev.*, 2003, **56**, B53–B53.

37 A. Dhooge, W. Govaerts and Y. A. Kuznetsov, *Assoc. Comput. Mach., Trans. Math. Software*, 2003, **29**, 141–164.

38 A. Saltelli, *Global Sensitivity Analysis: The Primer*, John Wiley, Chichester, England; Hoboken, NJ, 2008.

39 A. Saltelli, M. Ratto, S. Tarantola and F. Campolongo, *Chem. Rev.*, 2005, **105**, 2811–2827.

40 A. Kiparissides, S. S. Kucherenko, A. Mantalaris and E. N. Pistikopoulos, *Ind. Eng. Chem. Res.*, 2009, **48**, 7168–7180.

41 M. Rodriguez-Fernandez and J. R. Banga, *Bioinformatics*, 2010, **26**, 1675–1676.

42 Z. Zi, Y. Zheng, A. E. Rundell and E. Klipp, *BMC Bioinformatics*, 2008, **9**, 342.

43 O. Purcell, N. Savery, C. Grierson and M. di Bernardo, *Journal of The Royal Society Interface*, 2010, DOI: 10.1098/rsif.2010.0183.

44 H. Kitano, *Nat. Rev. Genet.*, 2004, **5**, 826–837.

45 K. Zhou, J. Doyle, K. Glover and J. Doyle, *Robust and Optimal Control*, Prentice Hall, 1995.

46 A. Rizk, G. Batt, F. Fages and S. Soliman, *Bioinformatics*, 2009, **25**, i169–178.

47 H. Kitano, *Foundations of Systems Biology*, MIT Press, Cambridge, Mass., 2001.

48 U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*, Chapman & Hall/CRC, Boca Raton, FL, 2007.

49 P. A. Iglesias and B. P. Ingalls, *Control Theory and Systems Biology*, MIT Press, Cambridge, Mass., 2010.

50 P. Wellstead, E. Bullinger, D. Kalamatianos, O. Mason and M. Verwoerd, *Annu. Rev. Control*, 2008, **32**, 33–47.

51 D. Bray and S. Lay, *Biophys. J.*, 1994, **66**, 972–977.

52 P. Francois and V. Hakim, *Proc. Natl. Acad. Sci. U. S. A.*, 2004, **101**, 580–585.

53 G. Rodrigo, J. Carrera and A. Jaramillo, *Bioinformatics*, 2007, **23**, 1857–1858.

54 M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, Goryanin, II, W. J. Hedley, T. C. Hodgman, J. H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novere, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner and J. Wang, *Bioinformatics*, 2003, **19**, 524–531.

55 G. Batt, B. Yordanov, R. Weiss and C. Belta, *Bioinformatics*, 2007, **23**, 2415–2422.

56 E. M. Clarke, O. Grumberg and D. Peled, *Model Checking*, MIT Press, Cambridge, Mass., 1999.

57 M. S. Dasika and C. D. Maranas, *BMC Syst. Biol.*, 2008, **2**, 24.

58 J. Hasty, F. Isaacs, M. Dolnik, D. McMillen and J. J. Collins, *Chaos*, 2001, **11**, 207–220.

59 M. J. Mohideen, J. D. Perkins and E. N. Pistikopoulos, *Comput. Chem. Eng.*, 1997, **21**, S457–S462.

60 Y. Cai, B. Hartnett, C. Gustafsson and J. Peccoud, *Bioinformatics*, 2007, **23**, 2760–2767.

61 M. Pedersen and A. Phillips, *J. R. Soc. Interface*, 2009, **6**, S437–S450.

62 J. A. Goler, *BioJADE: A Design and Simulation Tool for Synthetic Biological Systems*, MIT Computer Sciece and Artificial Intelligence Laboratory, Cambridge, Mass., 2004.

63 S. Kosuri, J. R. Kelly and D. Endy, *BMC Bioinformatics*, 2007, **8**, 480.

64 M. J. Czar, Y. Z. Cai and J. Peccoud, *Nucleic Acids Res.*, 2009, **37**, W40–W47.

65 E. Weeding, J. Houle and Y. N. Kaznessis, *Briefings Bioinf.*, 2010, **11**, 394–402.

66 A. D. Hill, J. R. Tomshine, E. M. Weeding, V. Sotiropoulos and Y. N. Kaznessis, *Bioinformatics*, 2008, **24**, 2551–2553.

67 D. Chandran, F. T. Bergmann and H. M. Sauro, *J. Biol. Eng.*, 2009, **3**, 19.

68 S. Mirschel, K. Steinmetz, M. Rempel, M. Ginkel and E. D. Gilles, *Bioinformatics*, 2009, **25**, 687–689.

69 K. E. Drexler, *Proc. Natl. Acad. Sci. U. S. A.*, 1981, **78**, 5275–5278.

70 H. W. Hellinga and F. M. Richards, *J. Mol. Biol.*, 1991, **222**, 763–785.

71 D. N. Bolon and S. L. Mayo, *Proc. Natl. Acad. Sci. U. S. A.*, 2001, **98**, 14274–14279.

72 J. M. Shifman and S. L. Mayo, *J. Mol. Biol.*, 2002, **323**, 417–423.

73 P. S. Huang, J. J. Love and S. L. Mayo, *Protein Sci.*, 2007, **16**, 2770–2774.

74 T. P. Treynor, C. L. Vizcarra, D. Nedelcu and S. L. Mayo, *Proc. Natl. Acad. Sci. U. S. A.*, 2007, **104**, 48–53.

75 A. L. Pinto, H. W. Hellinga and J. P. Caradonna, *Proc. Natl. Acad. Sci. U. S. A.*, 1997, **94**, 5562–5567.

76 H. Liu, J. J. Schmidt, G. D. Bachand, S. S. Rizk, L. L. Looger, H. W. Hellinga and C. D. Montemagno, *Nat. Mater.*, 2002, **1**, 173–177.

77 H. M. Salis, E. A. Mirsky and C. A. Voigt, *Nat. Biotechnol.*, 2009, **27**, 946–950.

78 M. De Mey, J. Maertens, G. J. Lequeux, W. K. Soetaert and E. J. Vandamme, *BMC Biotechnol.*, 2007, **7**, 34.

79 V. A. Rhodius and V. K. Mutalik, *Proc. Natl. Acad. Sci. U. S. A.*, 2010, **107**, 2854–2859.

80 J. E. Dueber, B. J. Yeh, K. Chak and W. A. Lim, *Science*, 2003, **301**, 1904–1908.

81 J. T. MacDonald, K. Maksimiak, M. I. Sadowski and W. R. Taylor, *Proteins: Struct., Funct., Bioinf.*, 2010, **78**, 1311–1325.

82 S. M. Douglas, H. Dietz, T. Liedl, B. Hogberg, F. Graf and W. M. Shih, *Nature*, 2009, **459**, 414–418.

83 G. L. Butterfoss and B. Kuhlman, *Annu. Rev. Biophys. Biomol. Struct.*, 2006, **35**, 49–65.

84 C. A. Rohl, C. E. Strauss, K. M. Misura and D. Baker, *Methods Enzymol.*, 2004, **383**, 66–93.

85 B. Kuhlman and D. Baker, *Proc. Natl. Acad. Sci. U. S. A.*, 2000, **97**, 10383–10388.

86 B. I. Dahiyat and S. L. Mayo, *Protein Sci.*, 1996, **5**, 895–903.

87 C. A. Voigt, D. B. Gordon and S. L. Mayo, *J. Mol. Biol.*, 2000, **299**, 789–803.

88 L. L. Looger and H. W. Hellinga, *J. Mol. Biol.*, 2001, **307**, 429–445.

89 J. B. Siegel, A. Zanghellini, H. M. Lovick, G. Kiss, A. R. Lambert, J. L. St Clair, J. L. Gallaher, D. Hilvert, M. H. Gelb, B. L. Stoddard, K. N. Houk, F. E. Michael and D. Baker, *Science*, 2010, **329**, 309–313.

90 A. Zanghellini, L. Jiang, A. M. Wollacott, G. Cheng, J. Meiler, E. A. Althoff, D. Rothlisberger and D. Baker, *Protein Sci.*, 2006, **15**, 2785–2794.

91 B. Kuhlman, G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard and D. Baker, *Science*, 2003, **302**, 1364–1368.

92 J. Na, K. N. Houk and D. Hilvert, *J. Am. Chem. Soc.*, 1996, **118**, 6462–6471.

93 D. J. Tantillo, J. Chen and K. N. Houk, *Curr. Opin. Chem. Biol.*, 1998, **2**, 743–750.

94 X. Gao, P. Yo, A. Keith, T. J. Ragan and T. K. Harris, *Nucleic Acids Res.*, 2003, **31**, 143e.

95 J. C. Cox, J. Lape, M. A. Sayed and H. W. Hellinga, *Protein Sci.*, 2007, **16**, 379–390.

96 M. Welch, S. Govindarajan, J. E. Ness, A. Villalobos, A. Gurney, J. Minshull and C. Gustafsson, *PLoS One*, 2009, **4**, e7002.

97 D. G. Gibson, J. I. Glass, C. Lartigue, V. N. Noskov, R. Y. Chuang, M. A. Algire, G. A. Benders, M. G. Montague, L. Ma, M. M. Moodie, C. Merryman, S. Vashee, R. Krishnakumar, N. Assad-Garcia, C. Andrews-Pfannkoch, E. A. Denisova, L. Young, Z. Q. Qi, T. H. Segall-Shapiro, C. H. Calvey, P. P. Parmar, C. A. Hutchison, 3rd, H. O. Smith and J. C. Venter, *Science*, 2010, **329**, 52–56.

98 D. Densmore, T. H. Hsiau, J. T. Kittleson, W. DeLoache, C. Batten and J. C. Anderson, *Nucleic Acids Res.*, 2010, **38**, 2607–2616.

99 D. G. Gibson, L. Young, R. Y. Chuang, J. C. Venter, C. A. Hutchison, 3rd and H. O. Smith, *Nat. Methods*, 2009, **6**, 343–345.

100 C. Engler, R. Gruetzner, R. Kandzia and S. Marillonnet, *PLoS One*, 2009, **4**, e5553.

101 T. Ellis, T. Adie and G. S. Baldwin, *Integr. Biol.*, 2010, DOI: 10.1039/c0ib00070a.

102 F. A. B. G. Bio, D. Baker, G. Church, J. Collins, D. Endy, J. Jacobson, J. Keasling, P. Modrich, C. Smolke and R. Weiss, *Sci. Am.*, 2006, **294**, 44–51.

103 A. C. Forster and G. M. Church, *Mol. Syst. Biol.*, 2006, **2**, 45.

104 M. C. Jewett and A. C. Forster, *Curr. Opin. Biotechnol.*, 2010, **21**, 697–703.

105 O. Rackham and J. W. Chin, *Nat. Chem. Biol.*, 2005, **1**, 159–166.

106 W. An and J. W. Chin, *Proc. Natl. Acad. Sci. U. S. A.*, 2009, **106**, 8477–8482.

107 W. J. Holtz and J. D. Keasling, *Cell*, 2010, **140**, 19–23.

108 K. L. Prather and C. H. Martin, *Curr. Opin. Biotechnol.*, 2008, **19**, 468–474.

109 C. Jackel, P. Kast and D. Hilvert, *Annu. Rev. Biophys.*, 2008, **37**, 153–173.

110 O. Khersonsky, D. Rothlisberger, O. Dym, S. Albeck, C. J. Jackson, D. Baker and D. S. Tawfik, *J. Mol. Biol.*, 2010, **396**, 1025–1042.

111 D. Umeno, A. V. Tobias and F. H. Arnold, *Microbiol. Mol. Biol. Rev.*, 2005, **69**, 51–78.

112 C. Richaud, D. Mengin-Lecreulx, S. Pochet, E. J. Johnson, G. N. Cohen and P. Marliere, *J. Biol. Chem.*, 1993, **268**, 26827–26835.

113 D. Bikard, S. Julie-Galau, G. Cambray and D. Mazel, *Nucleic Acids Res.*, 2010, **38**, e153.

114 A. D. Griffiths and D. S. Tawfik, *Trends Biotechnol.*, 2006, **24**, 395–402.

115 T. S. Gardner, C. R. Cantor and J. J. Collins, *Nature*, 2000, **403**, 339–342.

116 G. Turk, *Siggraph 91 Conference Proceedings*, 1991, **25**, 289–298.

117 H. Meinhardt and P. A. de Boer, *Proc. Natl. Acad. Sci. U. S. A.*, 2001, **98**, 14202–14207.

118 T. Danino, O. Mondragon-Palomino, L. Tsimring and J. Hasty, *Nature*, 2010, **463**, 326–330.

119 N. Geva-Zatorsky, N. Rosenfeld, S. Itzkovitz, R. Milo, A. Sigal, E. Dekel, T. Yarnitzky, Y. Liron, P. Polak, G. Lahav and U. Alon, *Mol. Syst. Biol.*, 2006, **2**2006 0033.

120 M. Thattai and A. van Oudenaarden, *Proc. Natl. Acad. Sci. U. S. A.*, 2001, **98**, 8614–8619.

121 D. Chandran, W. B. Copeland, S. C. Sleight and H. M. Sauro, *Drug Discovery Today: Dis. Models*, 2008, **5**, 299–309.

122 W. A. Lim, *Nat. Rev. Mol. Cell Biol.*, 2010, **11**, 393–403.

123 M. Heinemann and S. Panke, *Bioinformatics*, 2006, **22**, 2790–2799.

124 A. Funahashi, Y. Matsuoka, A. Jouraku, M. Morohashi, N. Kikuchi and H. Kitano, *Proc. IEEE*, 2008, **96**, 1254–1265.

125 R. Das and D. Baker, *Annu. Rev. Biochem.*, 2008, **77**, 363–382.