

In silico evolution of functional modules in biochemical networks

S.R. Paladugu, V. Chickarmane, A. Deckard, J.P. Frumkin, M. McCormack and H.M. Sauro

Abstract: Understanding the large reaction networks found in biological systems is a daunting task. One approach is to divide a network into more manageable smaller modules, thus simplifying the problem. This is a common strategy used in engineering. However, the process of identifying biological modules is still in its infancy and very little is understood about the range and capabilities of motif structures found in biological modules. In order to delineate these modules, a library of functional motifs has been generated via *in silico* evolution techniques. On the basis of their functional forms, networks were evolved from four broad areas: oscillators, bistable switches, homeostatic systems and frequency filters. Some of these motifs were constructed from simple mass action kinetics, others were based on Michaelis–Menten kinetics as found in protein/protein networks and the remainder were based on Hill equations as found in gene/protein interaction networks. The purpose of the study is to explore the capabilities of different network architectures and the rich variety of functional forms that can be generated. Ultimately, the library may be used to delineate functional motifs in real biological networks.

1 Introduction

The apparent complexity of biological networks is abundantly clear when we view the many network graphs now available, as posters or diagrams in textbooks. However, we perceive the network as being complex, because we choose to view the entire network in all its intricate details and are inevitably overwhelmed by the number of components and connections. Engineers who design the complex networks of electrical components found in modern microchips, take a different approach. Rather than trying to view the entire network simultaneously (which would be impossible for modern chip designs), engineers divide networks into modules, such modules are further divided as necessary, resulting in a hierarchy of functional descriptions. Other disciplines, such as computer programming, take a similar approach. It is this separation of behavioural levels that allows engineers to comprehend and build highly complex information processing systems.

In biology, we can use a similar strategy by breaking biological networks down into simpler and more manageable modules. The process of understanding complex biological networks then involves describing and locating functional modules in a larger network.

The difficult question then arises, what is a functional module? There have been numerous discussions of this issue in the literature [1–3] and a number of common themes have emerged. A key idea is replacement, where a module can be replaced without disturbing the rest of the

system behaviour. With replacement comes the notion of an interface, in which a module has a defined interface, which is the point of contact between the module and the rest of the system. Finally, the number of contact points at a module interface will often be smaller than the number of interactions internal to the module. This latter aspect is of interest because it has been used to uncover modules in complex networks. In particular, a common metric [4] used to uncover topological modularity in networks is based on this very idea.

Most of the work in understanding modularity in networks has focused on identifying topological patterns. In this paper, we are interested in functional modularity, that is, modules that can perform specific dynamic functions.

This paper describes an evolutionary approach used to generate functional modules, with the goal of building a library of modules categorised according to function [5, 6]. The method is simple, a target function is assigned and an evolutionary algorithm is used to evolve networks that satisfy the assigned function. We have used this method in the past to successfully evolve simple mathematical functions [5]. In this paper, we describe a much more extensive list of networks that have been evolved using a similar approach.

1.1 Mathematical description of biochemical systems

A multitude of approaches are available for modelling biochemical networks [7, 8]. Here, we shall limit our discussion to continuous models based on differential equations. The mathematical description of biochemical systems comprises a system of ordinary differential equations derived by considering the net balance between inflows and outflows of a particular species. The rates depend on species concentrations and kinetic parameter values. In general, we can depict a biochemical model in

the following form

$$\frac{dS_i}{dt} = f_i(S_j, P_k) = \sum v_i^{\text{in}} - \sum v_i^{\text{out}} \quad (i, j = 1, \dots, m), \quad (k = 1, \dots, p) \quad (1)$$

where S_i represents the concentration of the species i , P_k represents the kinetic parameters, m is the number of species, p is the number of parameters, v_i^{in} is the inflow rate, v_i^{out} is the outflow rate and f_i is the net production (if it is positive) or consumption (if it is negative) rate of the species S_i . Alternatively, we can represent the above set of differential equations in a more compact form (state-space representation)

$$\frac{d\mathbf{S}}{dt} = \mathbf{N}\mathbf{v}(\mathbf{S}(t), \mathbf{P}) \quad (2)$$

where \mathbf{N} is the $m \times n$ stoichiometry matrix and \mathbf{v} is the n -dimensional rate vector. Stoichiometric analysis of biochemical networks provides useful insights into network properties. The stoichiometry matrix is an important attribute of a network model and gives information on constraints among the fluxes and concentrations. Of interest to modellers are the conservation constraints that can exist in a biochemical model. Such constraints will often arise from subgroups of molecules, called moieties, whose concentration remains unchanged throughout the evolution of the system. Thus, the following constraint applies to conservation moieties

$$\mathbf{\Gamma}\mathbf{S} = \mathbf{T}$$

where \mathbf{T} represents the conservation vector and $\mathbf{\Gamma}$ represents the conservation matrix. The above constraint allows us to subdivide the species into dependent (\mathbf{S}_d) and independent (\mathbf{S}_i) species. The dependent species have the property that their concentrations can be calculated from the concentrations of independent species. This property allows us to reduce the total number of simultaneous differential equations to be solved by the number of conservation relations present in the system. More importantly, it permits us to compute a non-singular Jacobian, a necessary requirement for many of our evolution experiments. Further details on computing the conservation laws and Jacobian can be found in the work of Sauro and Ingalls [9].

1.2 Investigation of the dynamics of the biochemical systems

The inherent nonlinearity in the differential equations (2) will often preclude the analytical investigation of their dynamics. But, we can gain an insight into the behaviour of the system by linearising it around some fixed point, usually the steady-state. A steady-state is characterised by constant fluxes between the species and hence constant species concentrations. The steady-state concentrations of the species can be obtained by applying the steady-state constraint to (1)

$$\frac{dS_i}{dt} = 0 = f_i(S_1^0, \dots, S_m^0, P_k)$$

where S_1^0, \dots, S_m^0 are the steady-state concentrations of the species. As the above equations are nonlinear, solving the above system of equations may lead to more than one steady-state. What is of interest to us, however, are the steady-state solutions that satisfy $S_i \geq 0$ (as we cannot allow negative species concentrations) and the stability of

the solutions. Stability analysis of reaction networks is a well-established area and many excellent reviews exist [8, 10]. We can linearise (1) using a Taylor series around the steady-state. By ignoring the higher-order terms, we arrive at

$$\frac{d}{dt}(\delta S_i) = \sum_{j=1}^m \left. \frac{\partial f_i}{\partial S_j} \right|_{S_i^0} \delta S_j \quad (3)$$

where δS_i are small perturbations from the steady-state concentrations of the species S_i^0 . Thus, the linearised state-space representation of (2) would look like

$$\frac{d}{dt} \delta \mathbf{S} = \mathbf{A} \cdot \delta \mathbf{S}$$

where \mathbf{A} is called the Jacobian of the system with elements

$$a_{ij} = \left. \frac{\partial f_i}{\partial S_j} \right|_{S_i^0, P_k^0}$$

The general solution for the above linear differential equation is of the form

$$c_1 e^{\lambda_1 t} \mathbf{v}^1 + \dots + c_n e^{\lambda_n t} \mathbf{v}^n$$

where $\lambda_1, \dots, \lambda_n$ are n distinct eigenvalues of \mathbf{A} , and $\mathbf{v}^1, \dots, \mathbf{v}^n$ are the corresponding eigenvectors. For local stability about the fixed point, all the eigenvalues of \mathbf{A} must have negative real parts.

1.3 Control theoretic approach to analysis of biochemical networks

In the previous section, we described the response of the system to small perturbations in steady-state species concentrations. We can extend this idea further and look at the steady-state response of the system to small perturbations in species concentrations as well as the parameters. The linearised system then takes the following form

$$\frac{d}{dt}(\delta \mathbf{S}) = \mathbf{A} \cdot \delta \mathbf{S} + \mathbf{B} \cdot \delta \mathbf{P} \quad (4)$$

where the elements of matrix \mathbf{A} are $a_{ij} = \partial f_i / \partial S_j |_{S_i^0, P_k^0}$ and the elements of matrix \mathbf{B} are $b_{ik} = \partial f_i / \partial P_k |_{S_i^0, P_k^0}$. Although standard sensitivity analysis looks at the response of the system to small constant perturbations, it is also possible to look at time varying perturbations [11]. For this, we need to obtain the frequency response $\mathbf{H}(j\omega)$ of the system, which can be derived through the Laplace transform of the system. The frequency response for the system (4) is given by

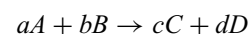
$$\mathbf{H}(j\omega) = (j\omega \mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \quad (5)$$

where ω is the frequency and \mathbf{I} is the identity matrix.

2 Network architectures

2.1 Networks based on mass action kinetics

The law of mass action states that the rate of a reaction is proportional to the concentration of its reactants. This is based on the assumption that the reaction mixture is well stirred and hence the frequency of a fruitful encounter between reactants depends on the number of reactant molecules. Thus, for a simple reaction given below



we can depict the rate v as

$$v = k[A]^a[B]^b$$

where $[A]$ and $[B]$ are reactant concentrations and $[C]$ and $[D]$ are the product concentrations, k is the rate constant and a , b , c and d are the stoichiometric coefficients. In our *in silico* evolutionary experiments, we have limited the reaction schema to uni–uni (involves one substrate and one product molecule), uni–bi (one substrate molecule dissociates into two product molecules), bi–uni (formation of one product molecule from two substrate molecules) and bi–bi (involves two substrate and two product molecules) reactions.

2.2 Networks based on Michaelis–Menten kinetics

When evolving protein networks, we used simple irreversible Michaelis–Menten kinetics. To simulate inhibition, we used a standard model for competitive inhibition. For activation, we treated the enzyme factor in the Michaelis equation as the activator. These choices follow previous models of protein signalling networks [12]. Also, we have used cycles as building blocks to generate protein networks to reflect the cascading behaviour seen in signal transduction networks. Clearly, more sophisticated kinetic laws could be envisaged, however, the current work is limited to those listed in Table 1.

2.3 Networks based on Hill kinetics

A problem with Michaelis–Menten formalism is that it often cannot explain the activity pattern of enzymes with multiple active sites or multiple subunits (viz. allosteric enzymes). In the case of allosteric enzymes, the binding of substrate to one site increases/decreases the activity of its neighbouring site(s). One of the simple extensions of Michaelis–Menten kinetics is the Hill equation that accounts for the co-operative nature of binding sites. The most common form of this equation is given by

$$v = \frac{V_{\max}[S]^h}{K_H + [S]^h}$$

where V_{\max} is the maximal velocity, h is the Hill coefficient. The value of h depends on substrate co-operativity, in which values of h greater than one indicate positive co-operativity and values of h less than one indicate negative co-operativity. A list of Hill type kinetics laws that were used in our gene–protein networks is given in Table 2

Table 1: Rate laws used to build protein–protein networks

Michaelis–Menten rate law	$V_{\max} \frac{[S]}{K_M + [S]}$
Competitive inhibition rate law	$V_{\max} \frac{[S]}{K_M(1 + ([I]/K_I)) + [S]}$
Activator kinetics	$[A]k_{\text{cat}} \frac{[S]}{K_M + [S]}$
Mixed activation/inhibition kinetics	$[A]k_{\text{cat}} \frac{[S]}{K_M(1 + ([I]/K_I)) + [S]}$

A , activator; I , inhibitor

Table 2: Rate laws used to build gene–protein networks

Repressor kinetics	$b + \frac{k_1}{k_2 + [R]^h}$
Activator kinetics	$b + \frac{k_1[A]^h}{k_2 + [A]^h}$

A , activator; R , repressor; b , basal expression rate

and are similar to previous models employing repressor kinetics [13].

In the gene–protein evolution experiments involving Hill kinetics, if more than one protein affects a single gene expression reaction (i.e. multiple transcription factors on a single promoter), the Hill equations were multiplied. Clearly, the kinetics we have used to describe the gene–protein networks are simplistic and a better approach would be to use the Shea and Ackers' formalism [14] for describing the effects of transcription factors on gene expression. For the initial studies on the evolution of gene–protein networks reported here, we feel that the simple Hill-like kinetics are sufficient. At a later time, we will investigate whether the use of Shea and Ackers' formalism would yield different results.

3 Materials and methods

The current implementation uses a type of evolutionary algorithm to evolve the network motifs [5, 6, 15, 16]. We begin the iterative process by creating a random population of individuals. An individual in this case is a biochemical network containing a collection of nodes (or species) and connections (or reactions) between them. The number of nodes as well as the number and type of connections between the nodes were chosen randomly. This results in reaction networks with different topological frameworks. The next step is to evaluate the fitness of each of these individuals on the basis of an objective function that varies depending on the functional motifs to be evolved, and rank them according to their fitness values. A selection process is then used to remove networks with poor fitness scores. The next generation of individuals is then created by randomly adding or deleting connections between nodes and mutating the rate constants. The cycle of creating and selecting networks is repeated until the fitness value reaches a desired threshold level (Fig. 1).

3.1 Creation of random population of networks

As we were using three different network architectures, we used three different sets of rules for creating the initial population of networks. In the case of networks with mass action kinetics, a fixed set of nodes were created and each of the nodes was connected randomly with uni–uni, uni–bi, bi–uni and bi–bi connections. In the case of protein–protein networks governed by Michaelis–Menten kinetics, a fixed number of cycles were created and each of these cycles were connected randomly with activation/inhibition feedback and feed-forward loops (see Fig. 15 for example). In the case of gene–protein networks with Hill kinetics, a fixed set of gene–protein pairs was created and then out of these pairs, a set of proteins was randomly chosen to serve as transcription factors for the genes. In all three cases, nodes with connections coming in and going out were characterised as floating nodes. If a node had

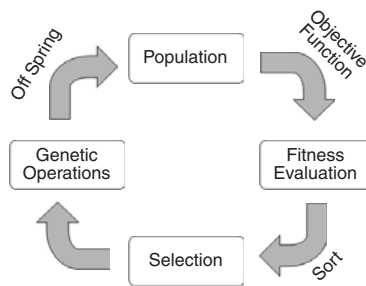


Fig. 1 *In silico evolution: the cycle starts with the creation of a random population of networks*

Each of the networks is assigned a fitness value depending on the closeness of network's behaviour to the desired response. Networks are then sorted in ascending order based on their fitness values and moved on to the selection phase. In the selection phase, networks with poor fitness scores are removed from the genetic pool. Finally, a new generation of individuals is created by cloning the surviving networks and then mutating the clones.

connections going out but not coming in, then it was characterised as an in-boundary node. If a node had a connection coming in but not going out, then it was characterised as an out-boundary node. Fig. 2 depicts the colouring schema we used to represent different kinds of nodes.

3.2 Fitness evaluation

The fitness value was computed by randomly choosing one of the floating nodes as the output node and one of the in-boundary nodes as the input node. Thus, in the case of protein-protein networks, any node can serve as an output node, whereas in the case of gene-protein networks only protein nodes can serve as output nodes. In all three cases, the steady states were computed by simulating the networks for sufficiently long time periods such that $\sum_{i=1}^m (dS_i/dt)^2$ is less than a predefined tolerance level. The simulations were done using a simple fixed step size fourth-order Runge-Kutta integration method [17]. The objective function to be minimised in the computation of fitness value of a network depends on the exact functional motif to be evolved. Further details on the fitness computation are given under the corresponding sections.

3.3 Selection

For the evolution of networks with mass-action kinetics and Michaelis-Menten kinetics, we used truncation selection [16]. We have implemented truncation selection by first sorting the networks according to their fitness scores, and then selecting a proportion (top 30%) of the ordered networks to serve as a candidate pool for the next generation. In the case of gene-protein networks governed by Hill kinetics, we used elitism along with tournament selection. Elitism guarantees that the best solutions (as determined from their fitness evaluations) are not lost from one generation to the next. We have implemented elitism by sorting

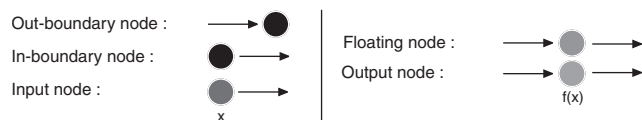


Fig. 2 *Colour codes: any one of the in-boundary nodes in the network can be chosen to act as an input node*

Similarly, any one of the floating nodes in the network can be chosen to act as an output node.

the networks on the basis of their fitness scores and carrying over up to 10% of the best individuals to the next generation. Tournament selection is performed by randomly choosing a pair of networks and selecting the network with a better fitness score; this network becomes one of the parents for the next generation. The tournament process is continued by sampling (with replacement) from the current generation until a full complement of parents has been chosen. Tournament selection allows us to adjust the selection pressure by changing the tournament size.

3.4 Genetic operations

To maintain an appropriately high level of genetic variability in our population, we generated the offsprings by asexual reproduction and mutation. We used two different types of mutational operators: (1) mutation of the network configuration, in which we randomly add or delete nodes and connections. Addition and deletion of connections could potentially convert a boundary node to a floating node and vice versa and (2) mutation of rate constants, in which we randomly selected a kinetic parameter and decreased or increased its value by a random percentage. Configuration mutations are generally responsible for drastic changes in fitness scores, whereas rate constant mutations bring about finer changes in fitness scores. Fig. 3 illustrates the changes in network dynamics brought about by the application of various genetic operators at different stages of evolutionary process.

Finally, the evolved networks were trimmed down manually by removing one connection at a time using Jarnac [18] until the desired response is disrupted. All the orphan-nodes (nodes without any in-coming and out-going connections) that resulted from the trimming process were deleted from the network. This helps us to identify the core network architecture responsible for the dynamic behaviour that is exhibited by its more complex precursor.

A typical run usually included only 100 individuals. Run times usually were of the order of 5 min on a standard Windows-based PC. The number of generations that was required to evolve a particular motif varied according to the sophistication of the objective function. For example, the evolution of homeostatic networks, using mass action kinetics took an average of 70 generation runs, whereas the evolution of oscillatory networks, using mass action kinetics took between 150 and 200 generations. In general, networks that were governed by mass action kinetics took much longer to evolve functionality than those governed by Michaelis-Menten and Hill kinetics. Fig. 4 illustrates typical fitness profiles obtained during the evolution of an oscillatory network.

3.5 Software tools

Software was developed in a variety of computer languages on Windows PCs running Windows XP. Protein-protein network evolution software was developed in C/C++, gene-protein networks software was developed using Delphi (www.borland.com) and the mass-action-based network software was developed using .NET (www.microsoft.com). Bifurcation plots were generated using Emery Conrad's Oscill8 (<http://oscill8.sourceforge.net/>). Frequency plots and post-evolution simulations were all carried out using the Systems Biology Workbench (SBW) [19]. Eigenvalue determination and conservation analysis were done using the LAPACK library [20] in conjunction with SBW. All the models (Jarnac and SBML files) listed in this paper and the source code for the software can be

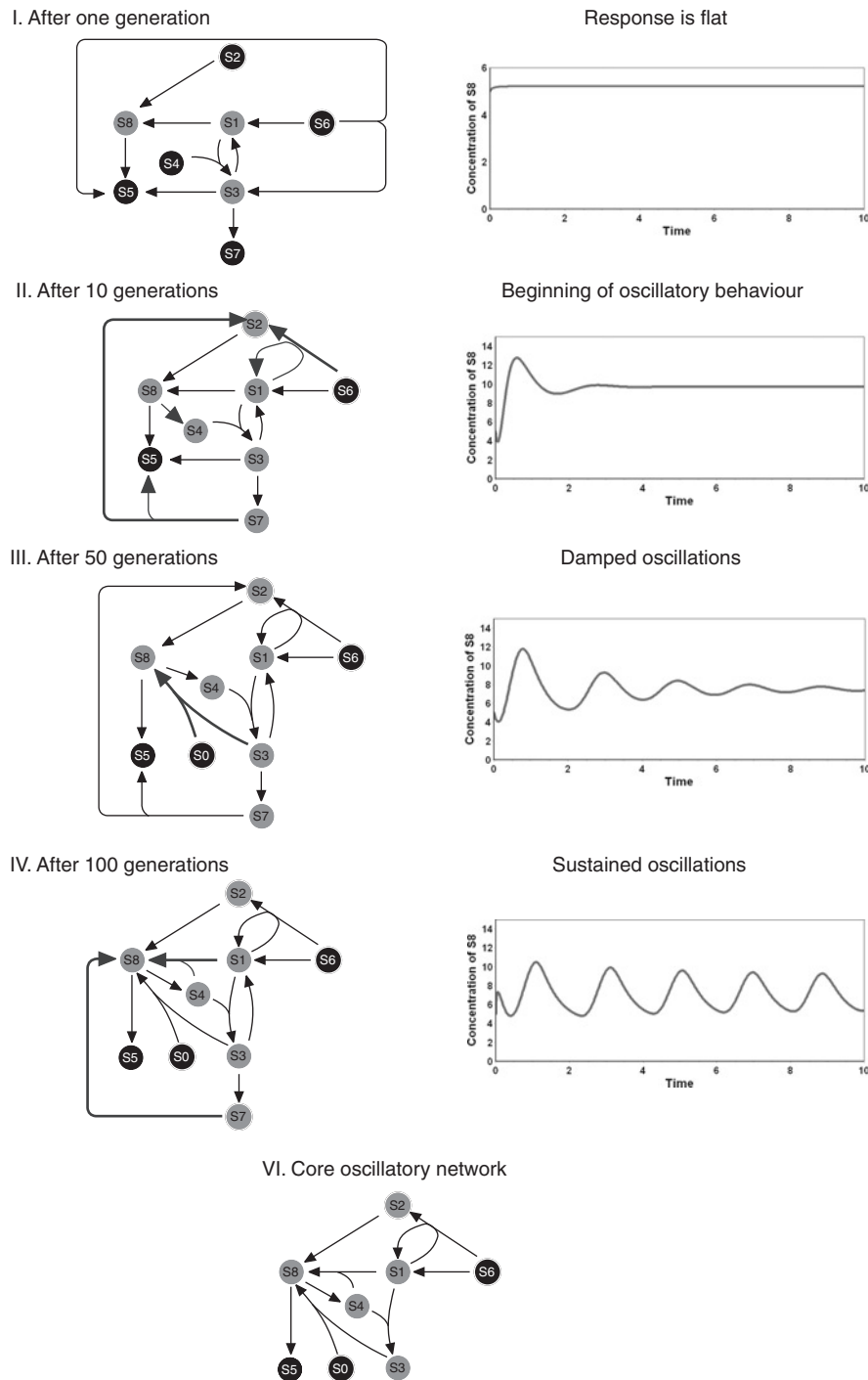


Fig. 3 Dynamics of four different networks at different stages in the evolution of an oscillatory network

Left column shows the network configuration (nodes/species and connections between them) and the right column shows the concentration of species S8 as a function of time

The diagram only illustrates the changes in the topological structure of the network

Connections that were affected at each stage are shown in bold

Not shown are the numerous changes in the rate constants that also occur during the evolution process

The core network shown in the last row is obtained by manually trimming the evolved network until the oscillatory behaviour is disrupted

downloaded from www.sys-bio.org, under the Research and Evolutionary Computing Section.

4 Bistable switch

Bistable switches are one of the most extensively studied functional motifs in biochemical networks [13, 21]. Much theoretical/experimental work has been done to elucidate the design and the mechanism of action of bistable switches in biochemical networks. A bistable system is like a toggle

switch, which can reside in only one of two stable steady states. One of the characteristic features of bistable systems is that they exhibit hysteresis, that is, it is harder to flip the system from one stable state to another than to maintain the system in its current state. One can observe the hysteresis behaviour of bistable systems by plotting the stimulus response curve. Stimulus response curves (or bifurcation diagrams) are plots that display the qualitative changes in the behaviour of the fixed points as a system parameter is smoothly varied. Readers are referred to the work of Strogatz [22] for an excellent introduction to this topic.

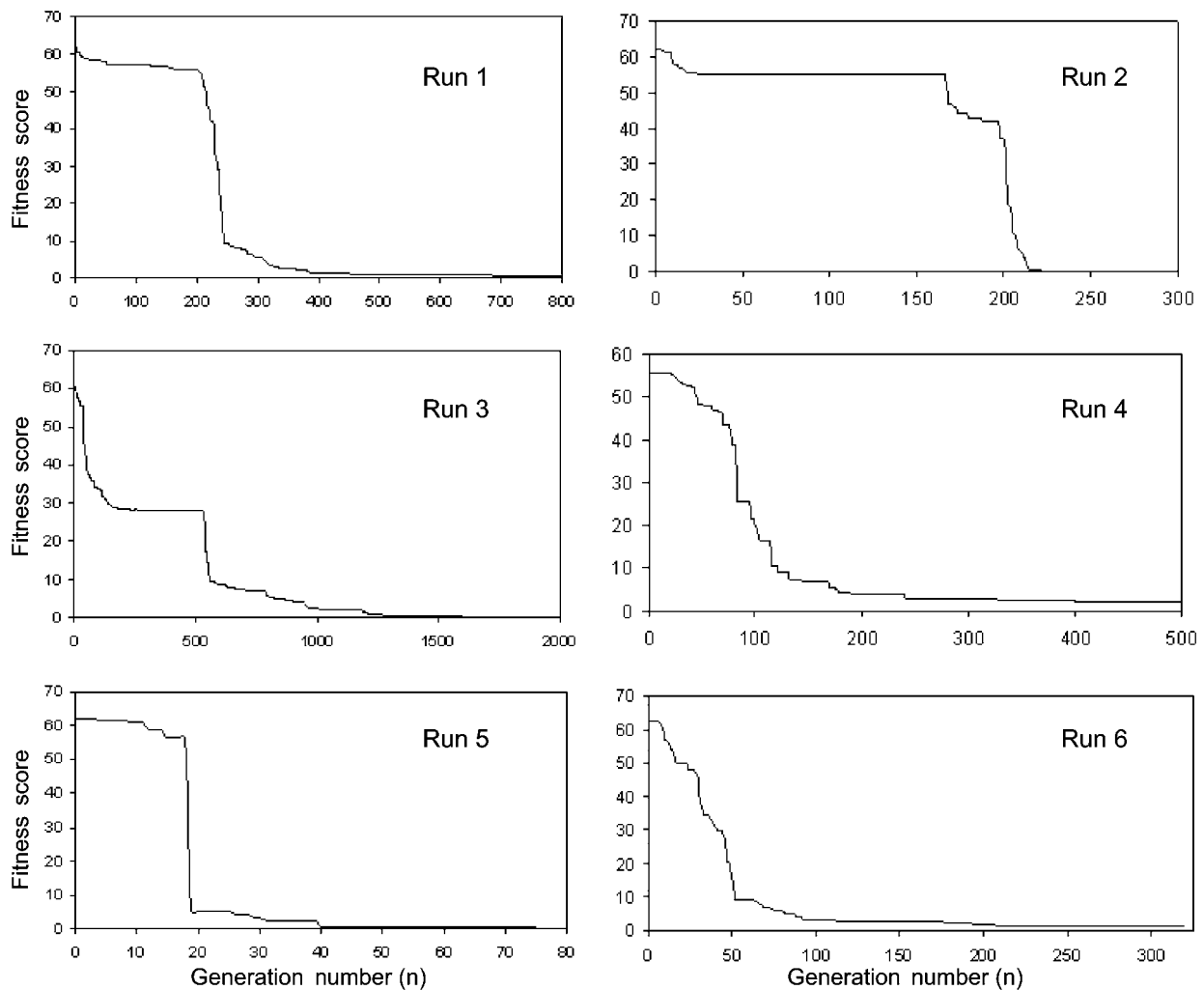


Fig. 4 Collection of typical fitness profiles showing the progress of evolution of oscillatory networks
Vertical axis indicates the fitness score and the horizontal axis the generation number

These bifurcation diagrams (or stimulus response curves) are very useful in detecting the turning points, that is, the critical values of parameters at which the system switches from one steady-state to another [23, 24].

4.1 Evolution approach

We have tried two different approaches to evolve bistable switches. In the first approach, we selected for networks that exhibited two stable states similar to the method employed by Francois and Hakim [6]. The fitness function was evaluated in two stages. In the first stage, each floating node was selected in turn to serve as an output node. The network was simulated to steady state (time T_1) and a deviation score (D_1) was computed by summing and squaring the differences between the output node's concentration at different time points of simulation and a predefined stable state value (off state). In the second stage, the output node was pulsed to a higher concentration and the network was simulated to steady state starting from time T_1 to time T_2 . A second deviation score (D_2) was then computed by summing and squaring the differences between the output node's concentration at different simulation time points and the pulsed level (on state). The fitness score of the network was then computed by simply adding the deviation scores D_1 and D_2 . The floating node with the best fitness score was selected to represent the fitness of the network.

Our second approach involved minimisation of the mathematical function

$$\epsilon = \frac{\prod \lambda_i}{1 - 0.99 * \exp(-|\prod \lambda_{\min}|)}$$

where $\prod \lambda_i$ refers to the product of all the eigenvalues of the Jacobian and $\prod \lambda_{\min}$ refers to the product of all the eigenvalues except the minimum eigenvalue. This is based on the fact that the turning point bifurcation arises when one of the eigenvalues of the reduced system approaches zero [25]. We computed the eigenvalues from the reduced Jacobian of the system rather than the full Jacobian because linear dependencies in the system of equations can also lead to zero eigenvalues. It is important to note that this approach does not guarantee the evolution of a bistable switch, it only selects for networks that can exhibit turning point behaviour. Although we were able to evolve networks with bistable behaviour, most of the networks we obtained were memoryless/ultrasensitive switches (which elicit a graded response with respect to the parameter changes).

Each of the evolved networks were grouped under two distinct categories.

4.1.1 Irreversible bistable switch: It has been hypothesised that cellular systems use bistability as a means to achieve irreversibility of biochemical processes [21, 26].

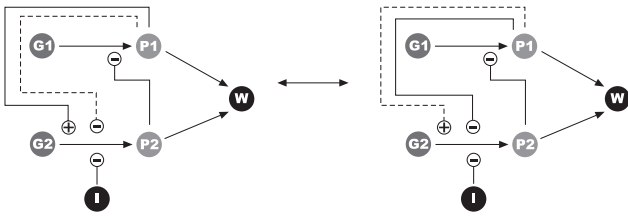


Fig. 5 *Irreversible bistable switch based on Hill kinetics*
 At low levels of concentration, P1 activates the production of P2 and at high levels it inhibits the production of P2
 At low values P1 inhibits its own production thereby locking itself in the low state
 At high values P1 activates its own production and thereby remains in the high state
 The change in modifier status of P1 is a result of P1 acting as both a repressor and activator simultaneously
 The change from activator to repressor is a result of the changing balance between the two effects as P1 changes concentration

Irreversibility is normally observed when a near zero stimulus is able to push the system from one stable state to another. One can recognise irreversibility in bistable systems by looking at the stimulus response curves, in which one of the two turning points appear in the negative parameter space indicating that it is impossible to switch back to the alternate steady-state. The network shown in Fig. 5 is one of the irreversible bistable switches that was evolved using Hill kinetics. Fig. 6 depicts the bifurcation plot for the network in Fig. 5.

4.1.2 Reversible bistable switch: In contrast to irreversible bistable switches, both the saddle nodes of reversible bistable switches appear in the positive region of parameter space and thus allows one to switch back and forth between the two stable states merely by changing the parameter values. Figs. 7 and 8 illustrate some of the reversible bistable switches that were evolved.

Fig. 9 represents the network diagram for one of the memoryless switches that was evolved.

5 Oscillators

Oscillations can be witnessed at several levels in living systems, be it at the cellular level (intracellular calcium

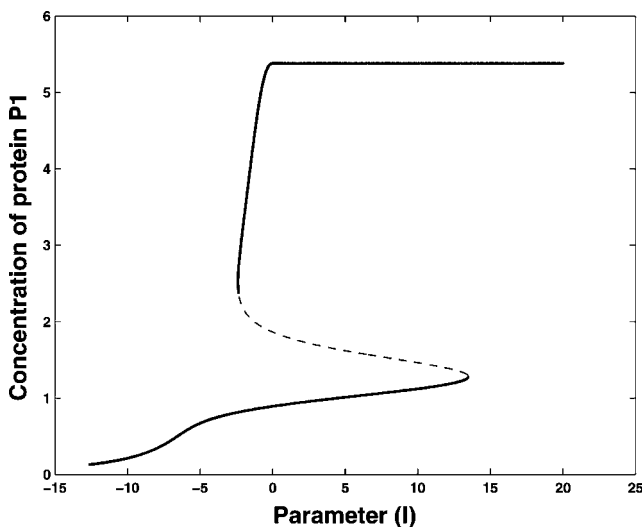


Fig. 6 *Bifurcation plot for the network in Fig. 5*
 One of the two turning points appears in the negative parameter space that indicates irreversibility

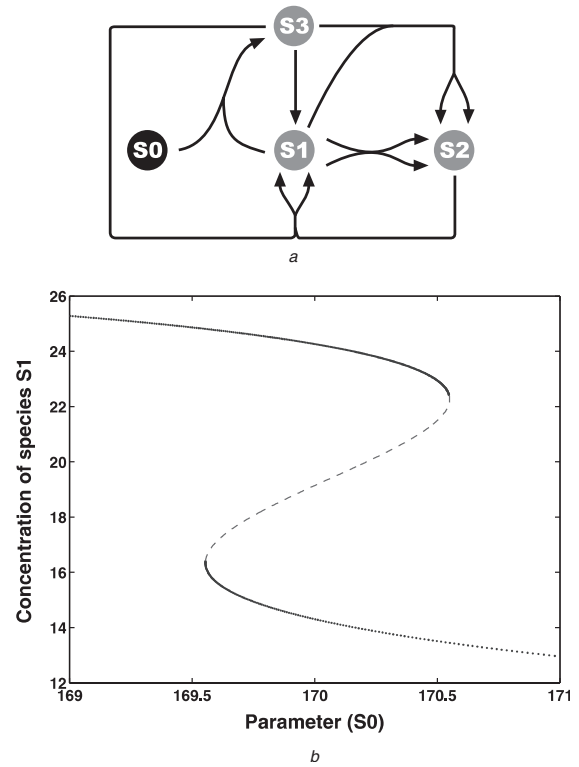


Fig. 7 *Bistable switch based on mass action kinetics: bistability is achieved by mutual activation of S1 and S2*
 Dotted line indicates the unstable branch
 a Network configuration
 b Bifurcation plot

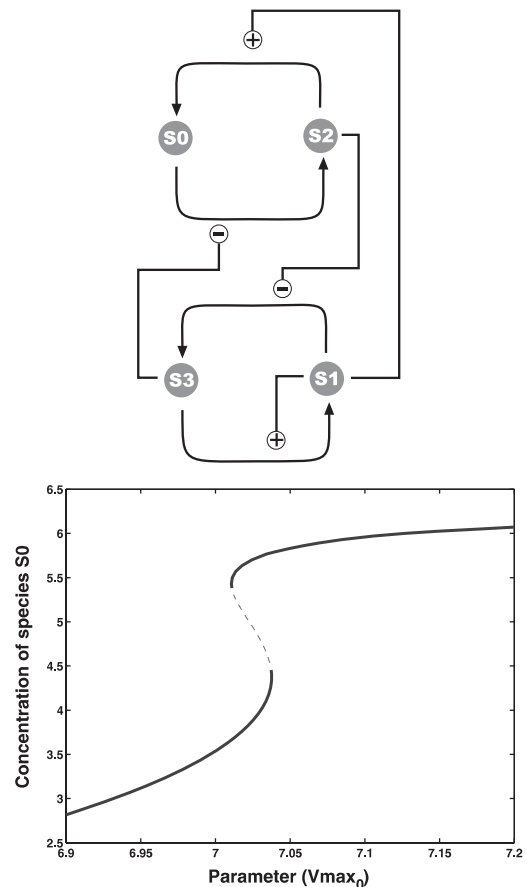


Fig. 8 *Bistable switch based on Michaelis-Menten kinetics: S2 inhibits the production of S3 and S3 in turn inhibits the production of S2*
 Dotted line indicates the unstable branch

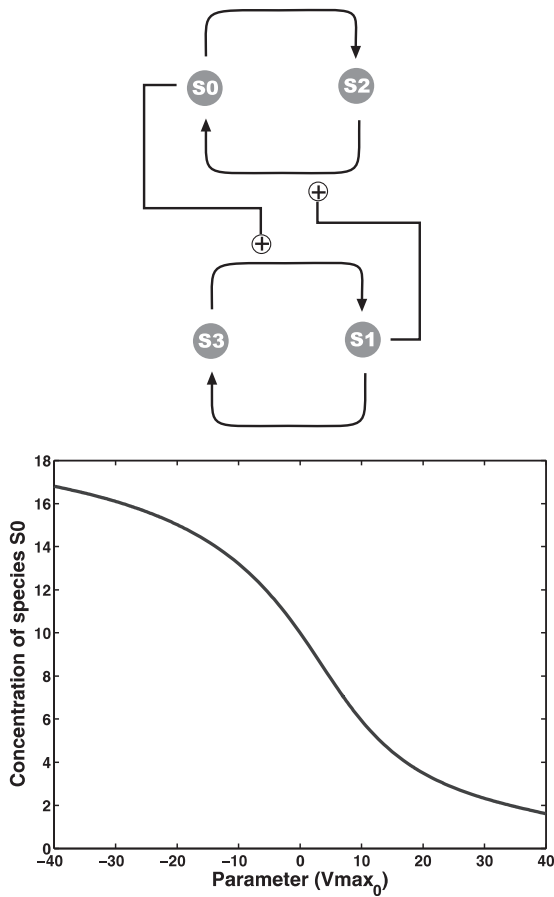


Fig. 9 Memoryless switch based on Michaelis–Menten kinetics: elicits a graded response

oscillations) or at the organism level (circadian rhythm) [27]. In fact, rhythmic phenomena are not limited to the biological world; its ubiquitous presence has triggered much interest in various scientific disciplines [28, 29]. Over the past century, extensive research has been carried out to gain a mechanistic understanding of the design principles that underlie this interesting phenomenon. One of the remarkable features of oscillators is that they can encode information in both the frequency and amplitude of their oscillations [30].

5.1 Evolution approach

We used two different strategies to evolve oscillators. In the first and simplest approach, the objective function involved minimising the error between the time course simulation data of the *in silico* network and a series of points corresponding to alternating peaks and troughs. In the second case, we attempted to minimise the mathematical function

$$\epsilon = \frac{\prod \lambda_i^R}{\prod (1 - 0.99 * \exp(-|\lambda_i^I|))}$$

where λ_i^R and λ_i^I represent the real and imaginary parts of the eigenvalues (λ_i) of the Jacobian. This approach relies on the fact that the Hopf bifurcation occurs when a pair of complex eigenvalues crosses the imaginary axis [25].

Using the above two approaches, we were able to evolve a variety of oscillator motifs, each of which can be grouped under two distinct categories.

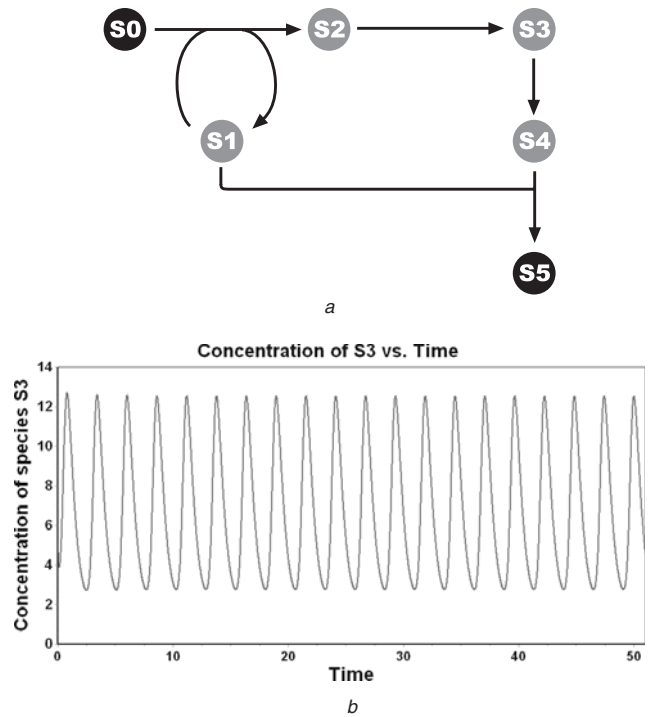


Fig. 10 Feedback oscillator based on mass action kinetics
Oscillations are brought about by negative feedback of S4 on S1 coupled with a delay
a Network configuration
b Timecourse simulation

5.1.1 Feedback oscillator: Feedback loops can be used to drive a system towards oscillatory behaviour because of phase differences arising out of delays in the system [31]. Figs. 10–12 represent some of the evolved oscillators based on negative feedback loops.

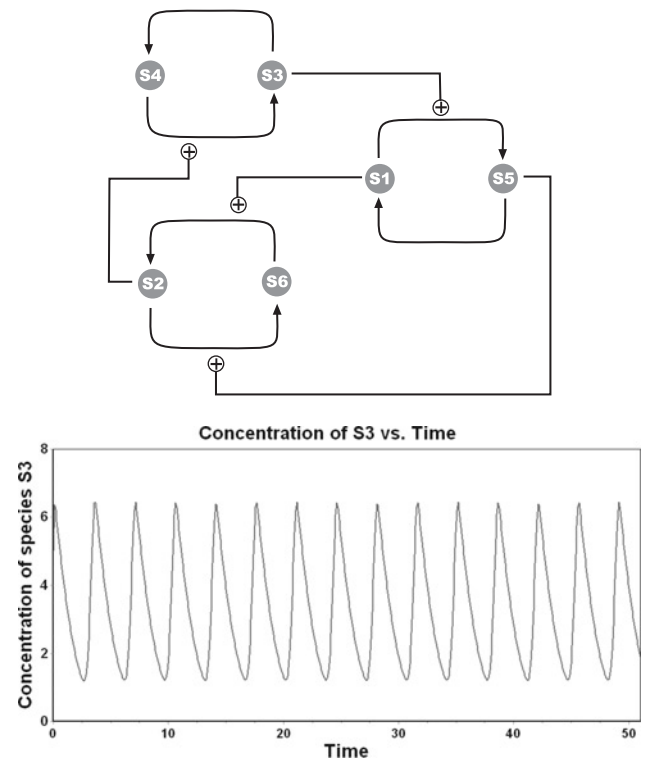


Fig. 11 Ring feedback oscillator based on Michaelis–Menten kinetics

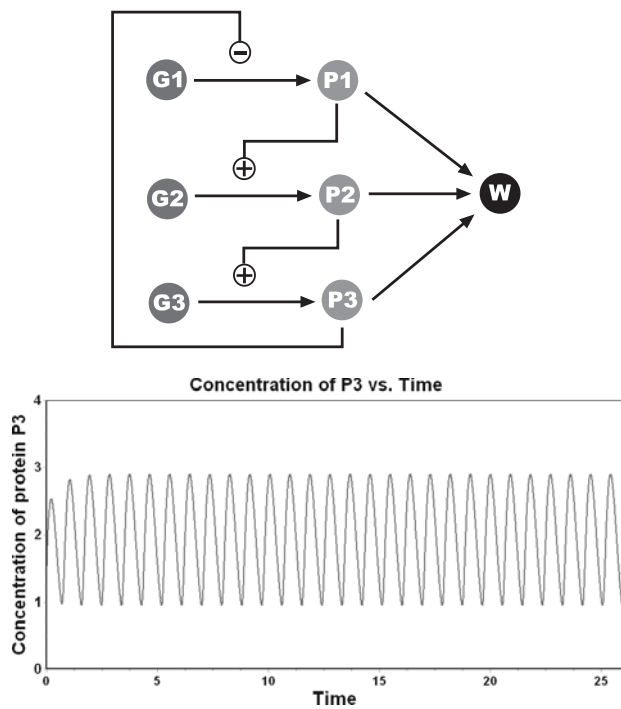


Fig. 12 Feedback oscillator based on Hill kinetics

5.1.2 Relaxation oscillator: In electrical circuits, a relaxation oscillator usually includes a resistor, a capacitor and a threshold device. The capacitor is charged through the resistor, which causes an exponential rise in its voltage. The threshold device, which is connected in parallel with the capacitor, discharges the capacitor quickly when the voltage across it reaches some upper threshold level and stops conducting when the voltage across it falls below some lower threshold level. This cycle is repeated in a periodic manner, which results in oscillatory behaviour.

Biochemical systems employ autocatalysis as a means to implement relaxation oscillators. In fact, autocatalysis is the primary requirement for Hopf-bifurcation in a network with only two independent species [32]. One can detect autocatalysis by looking for positive elements in the main diagonal of the Jacobian matrix. Autocatalysis is only a necessary but not a sufficient condition for a Hopf bifurcation. For a two-component network to oscillate, it must also contain a negative feedback along with an autocatalytic species [33]. Figs. 13 and 14 depict two of the evolved relaxation oscillators. Note the spiking pattern in the oscillations, which is a characteristic of relaxation oscillators.

A bistable switch can be used as a building block to generate a relaxation oscillator (Fig. 14). A more elaborate example of this can be found in Fig. 15.

6 Homeostatic network

Homeostasis is one of the most remarkable properties of living systems. A homeostatic system is capable of maintaining its internal environment within set limits by virtue of specific active regulatory mechanisms. Homeostatic systems react to random disturbances in the environment through a series of modifications that are equal in size and opposite in direction to those that created the disturbance. The ability of biological networks to respond and adapt to changes in the environment is an important topic of current research. The chemotaxis circuit in *E. coli* provides us with a well-known example of a biological network that exhibits perfect homeostasis when subjected to disturbances.

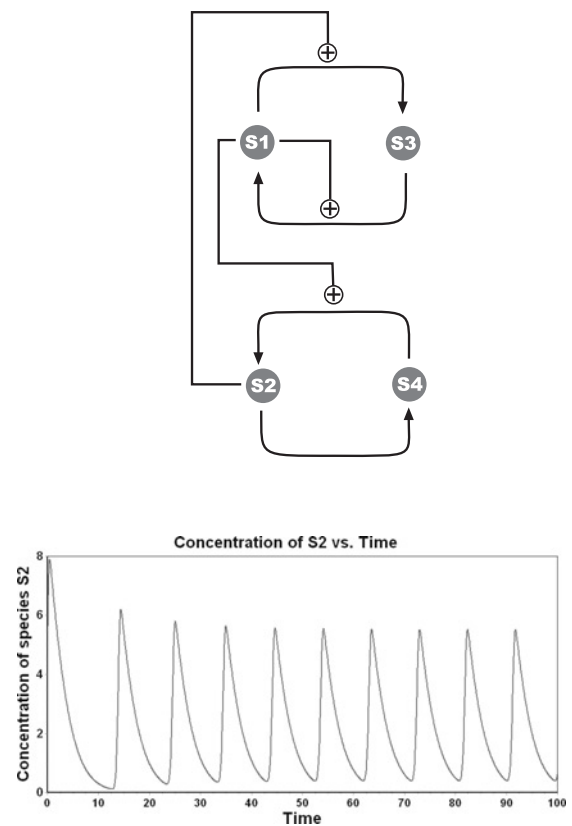


Fig. 13 Relaxation oscillator based on Michaelis–Menten action kinetics

6.1 Evolution approach

As in the case of oscillators and bistable switches, we have used two different strategies to evolve homeostatic networks. In the first approach, the objective function involved minimisation of the difference in the steady-state concentrations of a state variable (a floating species) of the network before and after an external disturbance, which was introduced by perturbing a boundary node. The

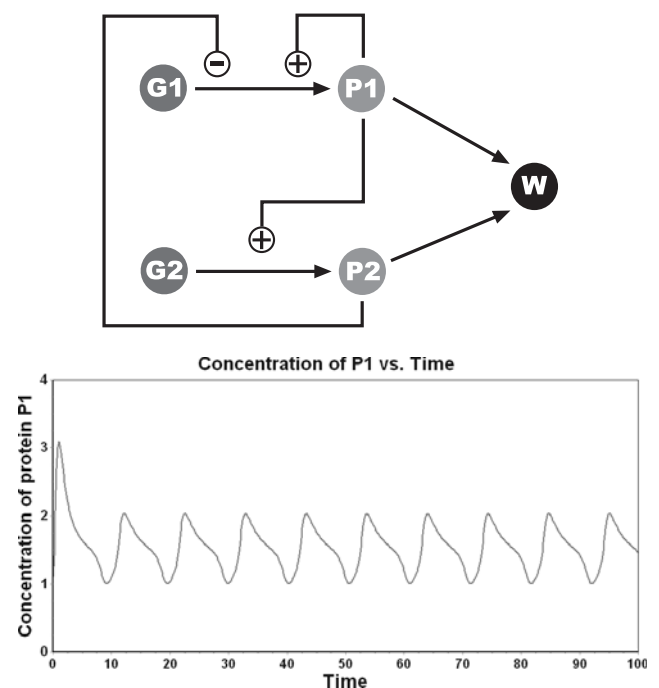


Fig. 14 Relaxation oscillator based on Hill kinetics

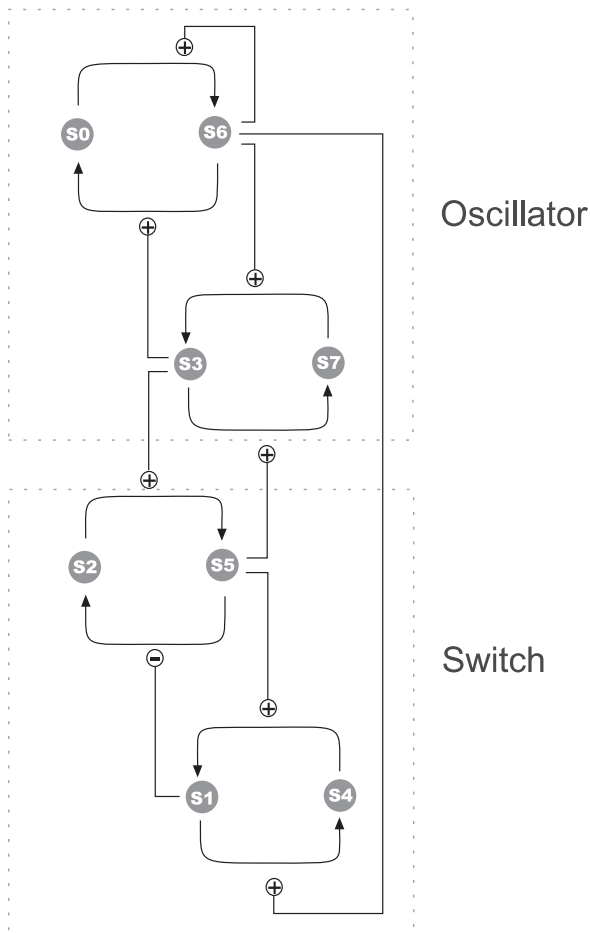


Fig. 15 Hybrid of a switch and an oscillator based on Michaelis–Menten kinetics

Relaxation oscillator constantly drives the switch between the on and off states, thus generating oscillations

second approach is based on the fact that homeostatic networks can employ integral-like control mechanisms [34]. The integral action has a characteristic frequency response very much like a band-pass filter. We therefore decided to evolve reaction networks where the objective function was a frequency response exhibiting band-pass-filter-like characteristics.

The networks that were evolved using the above two objective functions achieve homeostasis by increasing the production and consumption rates of the floating species by the same amount (feed-forward mechanism). This mechanism is a common strategy used by metabolic systems to increase flux through a pathway.

Some of the evolved homeostatic networks are shown in Figs. 16 and 17.

7 Frequency filters

Frequency filters are often used in man-made devices to manipulate and process signals. Examples include filtering a signal from a noisy source, identifying specific frequencies in a complex signal and performing various other operations such as phase shifting and inversion. It is likely that biological processing networks perform similar operations [35], particularly for eliminating unwanted noise in the system [36–38].

The ability to pass certain frequencies and cutoff others is the characteristic feature of frequency-selective filters. Here, we will present three basic kinds of filters that are commonly encountered in man-made systems.

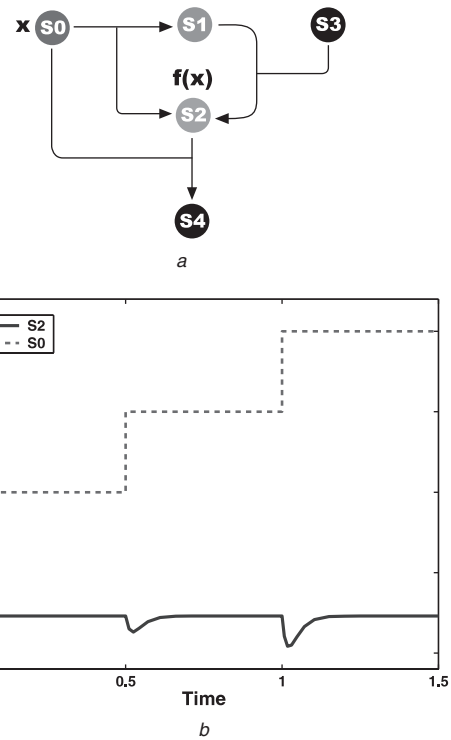


Fig. 16 Homeostatic network based on mass action kinetics

Concentration of species S2 remains unchanged before and after perturbation of boundary species S0

a Network configuration
b Timecourse simulation

7.1 Bode plots

The most useful technique for plotting the frequency response function was developed by H.W. Bode at Bell Laboratories between 1932 and 1942 and hence the name Bode plot [31]. Notice that the frequency response that we presented in Section 1.3 is a complex function of frequency ω . The frequency response is typically plotted as

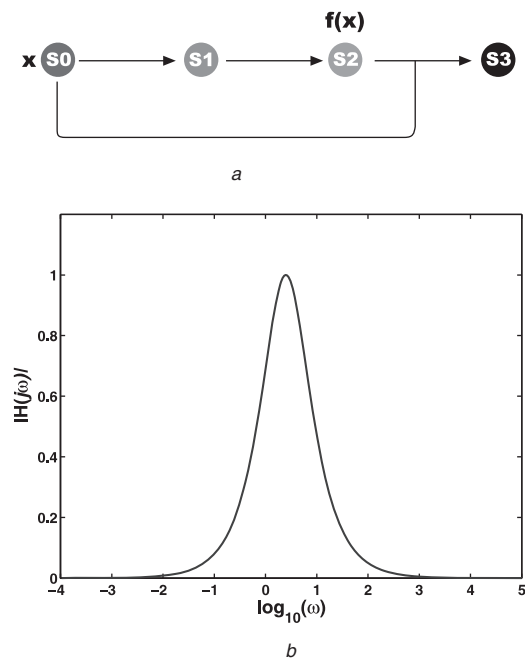


Fig. 17 Homeostatic network based on mass action kinetics

a Network configuration
b Amplitude response in frequency domain

two curves: magnitude denoted by $|H(j\omega)|$ against $\log \omega$, and the phase denoted by $\angle H(j\omega)$. Using a log scale for frequency (ω) and linear scale for magnitude ($|H(j\omega)|$) allows us to span a much wider range of frequencies on a single plot. In the following work, we only show the magnitude plots.

7.1.1 Low-pass filter: A low-pass filter selects for frequencies near zero and rejects or attenuates high frequencies. An ideal low-pass filter should have a magnitude close to one near zero frequency and zero at higher frequencies. Most real-world systems act as low-pass filters.

7.1.2 High-pass filter: A high-pass filter passes all high frequency signals and rejects or attenuates signals with low frequency. An ideal high-pass filter should have a magnitude close to zero at low frequencies and close to one at high frequencies. One could argue that it is impossible to design an ideal high-pass filter as it would mean that the transmission band should extend to infinite frequency. For all practical purposes, one can use a band-pass filter with a very long frequency range as a high-pass filter.

7.1.3 Band-pass filter: A band-pass filter rejects frequencies that are too high and too low and selects for frequencies within a defined range. An appropriate combination of the properties of low-pass and high-pass filters can be used to construct a band-pass filter. We have discussed the application of the band-pass filter in a biochemical framework in Section 6.1, wherein we tried to evolve a homeostatic network whose frequency response is similar to that of a band-pass filter.

7.2 Evolution approach

The approach used to evolve each of the three filters is quite similar. The objective function involved minimisation of the difference between the magnitude of the frequency response ($|H(j\omega)|$) of the evolved network and the magnitude of the desired frequency response at various values of ω . The magnitudes of the desired frequency response function $D(\omega)$ for each of the three filters are as follows

$$\text{Low-pass filter: } |D(\omega)| = \frac{1}{k_1 + \omega^n}$$

$$\text{High-pass filter: } |D(\omega)| = \frac{\omega^n}{k_2 + \omega^n k_1}$$

$$\text{Band-pass filter: } |D(\omega)| = \frac{1}{k_0} - \frac{1}{k_1 + \omega^n} - \frac{\omega^n}{k_2 + \omega^n k_1}$$

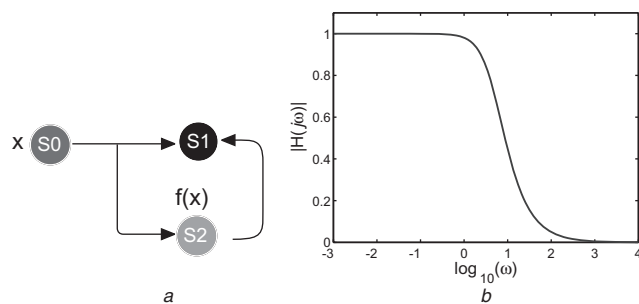


Fig. 18 Low-pass filter based on mass action kinetics
a Network configuration
b Amplitude response in frequency domain

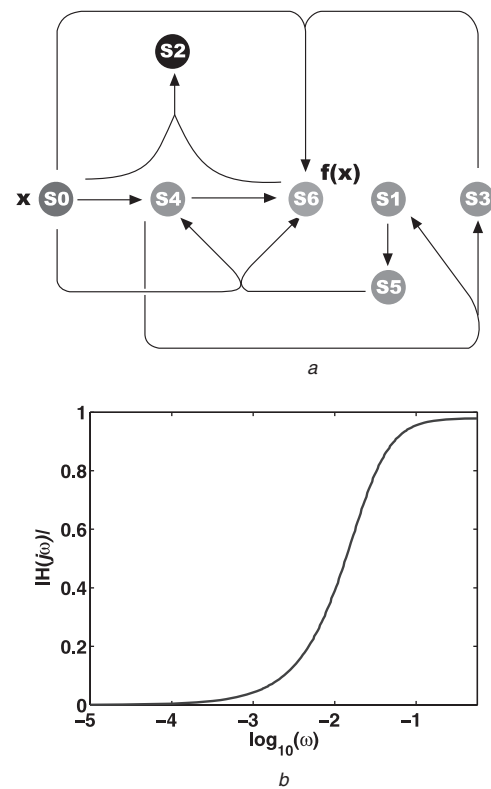


Fig. 19 High-pass filter based on mass action kinetics
a Network configuration
b Amplitude response in frequency domain

where k_0 , k_1 and k_2 are constants chosen such that $D(\omega)$ has the desired filter characteristics. The frequency response function $H(j\omega)$ of the evolved networks was computed using (5). The elements in matrix \mathbf{B} were computed by perturbing an input node chosen at random.

Examples of each of the three different kinds of filters that were evolved are shown in Figs. 18–20.

In the current studies, we only attempted to evolve filtering networks, using mass action kinetics. Of more interest is the possibility of evolving such networks based on the gene–protein architecture. This work will be left for future studies.

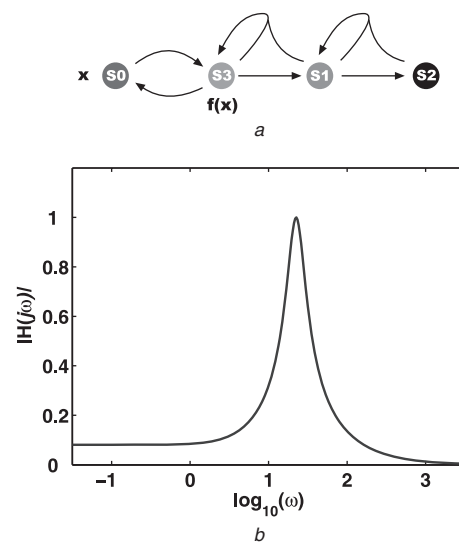


Fig. 20 Band-pass filter based on mass action kinetics
a Network configuration
b Amplitude response in frequency domain

8 Conclusions

In this paper, we have described a novel evolutionary approach to generate reaction networks with specific functional characteristics. In a previous paper [5], we described the *in silico* evolution of simple mathematical functions. We extend the work by exploring the evolution of more biologically relevant functions, such as oscillators, switches, homeostatic systems and frequency filters. In all cases, we were successful. In addition, we made a tentative investigation into how different network architectures influence the ease by which specific functionalities can be evolved. For example, evolving functional networks using mass action kinetics proved to be difficult compared with that using protein–protein or gene–protein networks. This difference can probably be explained by the size of the search space in each network architecture. Mass action networks are more generic and their search space is very large; hence it proved more difficult for the search algorithm to locate a solution. A summary of run statistics from various evolution experiments is shown in Table 3.

Another problem we encountered was related to the nature of the objective function. For those objective functions that were able to describe the objective in a continuous fashion, the rate of evolution was rapid. Examples include oscillators and frequency filters. In these cases, one can generate an almost continuous range of behaviour for these classes (Fig. 3). The most problematic objective function was undoubtedly the bistable switch. Although previous work [6] indicated that the evolution of bistability to be relatively straightforward, we found it to be extremely difficult, particularly for mass action networks. Out of hundreds of evolution runs, we only encountered two mass action networks that showed bistability (see Table 3). One of these is shown in Fig. 7. For protein–protein and gene–protein networks, generating bistable systems was straightforward. To understand the reason for this difference, we made a detailed examination of the phylogenetic tree for the evolution of the network shown in Fig. 7. What became clear was that the emergence of bistability was the result of a series of unrelated single point mutations, which together generated bistable behaviour. The problem with bistability is that there is no intermediate bistable behaviour, the objective function for bistability is discontinuous, a network is either bistable or not and there are no intermediate forms. The evolution of bistable behaviour in mass action networks therefore depends on a serendipitous combination of mutations. Once a bistable system emerges (no matter

how weak), the normal evolutionary process takes over and selects and refines the network. The number of mutations necessary to generate a bistable network in a mass action network is very large. At present, we do not know the size of the search space, but we intend to investigate the actual size in future work. For protein–protein and gene–protein networks, the number of mutations required to generate a bistable system was much smaller, hence it was easier to generate bistability in these systems. To explain the discrepancy between our results and the work by Francois and Hakim [6], we note that these authors used a different approach to evolve bistability. They created very large networks through continual growth by the addition of new nodes and connections. Although the authors do not indicate the size of the final network, we imagine they must have been quite large, possibly hundreds of nodes in each network of the population. As a result, they were able to sample a much larger space and hence were more likely to encounter a network that had bistable behaviour.

One of the interesting, though not surprising observations, was the range of unexpected solutions that we observed. For example, we anticipated that in the evolution of homeostatic networks some form of integral control might evolve. Instead the evolutionary process took a simpler route and employed a mechanism on the basis of the simultaneous stimulation of reactions to achieve homeostasis. Another example, where we found unexpected solutions was in the evolution of bistability. The objective function for bistability was simple: we would give high scores to networks that could maintain two different concentrations at a given set of parameters. However, there are alternative ways to achieve this objective than by evolving a genuine bistable network. The simplest solution turned out to employ conserved cycles in the network, so that perturbations in species levels resulted in fixed changes to the total mass in the cycles, thus mimicking bistability. These observations clearly indicate the importance of selecting an appropriate objective function.

There are many questions that we have not addressed in this work and the results we present in this paper represent only an initial exploration of the potential for using evolutionary techniques to generate functional networks. One thing is certain, we have shown that it is straightforward to evolve networks with a wide variety of functional properties.

In future, we wish to explore a number of areas. We do not know how the different mutational operators influence the rate of evolution; at present, we have network

Table 3: Summary of results from different evolution experiments

Evolved motif	Kinetics	Average number of generations	Approximate percentage of successful runs, %
Oscillator	Mass action	105	57
	Michaelis–Menten	48	23
	Hill kinetics	55	50
Bistable switch	Mass action	317	2
	Michaelis–Menten	75	11
	Hill kinetics	7	90
Homeostatic networks	Mass action	70	21
Low-pass filter	Mass action	10	close to 100
Band-pass filter	Mass action	54	42
High-pass filter	Mass action	140	5

configuration changes and kinetic parameter changes. In addition, we do not know how the mutation rate affects the emergence of solutions. We did explore the effect of different selection regimes on the rate of evolution, including the use of elitism and tournament selection. As expected, elitism generated less variation and fewer successful runs. The best combination we found was tournament selection combined with elitism.

The most obvious element missing from our work is the use of crossover in the evolution of reaction networks. In the current work, we have not employed any form of crossover. Early studies showed that crossover tended to be disruptive and we therefore avoided it. We feel that the detrimental effects of crossover were caused by crossing over non-homologous networks. In future work, we hope to revisit the use of crossover and to apply crossover only to homologous networks.

9 Acknowledgments

This work was supported by grants from the National Science Foundation (0432190 and FIBR 0527023 to H.M.S.). V.C. and H.M.S. are grateful for generous support from the DARPA/IPTO BioComp program, contract number MIPR 03-M296-01. S.R.P. was supported by a grant from the DOE GTL program.

10 References

- Hartwell, L.H., Hopfield, J.J., Leibler, S., and Murray, A.W.: 'From molecular to modular cell biology', *Nature*, 1999, **402**, pp. 47–52
- Kholodenko, B.N., Klyatkin, A., Bruggeman, F.J., Sontag, E., Westerhoff, H.V., and Hoek, J.B.: 'Untangling the wires: a strategy to trace functional interactions in signaling and gene networks', *PNAS*, 2002, **99**, pp. 12841–12846
- Wolf, D.M., and Arkin, A.P.: 'Motifs, modules and games in bacteria', *Curr. Opin. Microbiol.*, 2003, **6**, pp. 125–134
- Newman, M.E.J., and Girvan, M.: 'Finding and evaluating community structure in networks', *Phys. Rev. E*, 2004, **69**, p. 026113
- Deckard, A., and Sauro, H.M.: 'Preliminary studies on the *in silico* evolution of biochemical networks', *Chem. BioChem.*, 2004, **5**, (10), pp. 1423–1431
- Francois, P., and Hakim, V.: 'Design of genetic networks with specified functions by evolution in silico', *PNAS*, 2004, **101**, (2), pp. 580–585
- Bower, J.M., and Bolouri, H.: 'Computational modeling of genetic and biochemical networks' (MIT Press, 2001)
- Heinrich, R., Rapoport, S.M., and Rapoport, T.A.: 'Metabolic regulation and mathematical models', *Prog. Biophys. Mol. Biol.*, 1977, **32**, pp. 1–82
- Sauro, H.M., and Ingalls, B.: 'Conservation analysis in biochemical networks: computational issues for software writers', *Biophys. Chem.*, 2004, **109**, (1), pp. 1–15
- Stucki, J.W.: 'Stability analysis of biochemical systems – a practical guide', *Prog. Biophys. Mol. Biol.*, 1978, **33**, (2), pp. 99–187
- Ingalls, B.P.: 'A frequency domain approach to sensitivity analysis of biochemical systems', *J. Phys. Chem.*, 2004, **B108**, pp. 1143–1152
- Kholodenko, B.N.: 'Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades', *Eur. J. Biochem.*, 2000, **267**, (6), pp. 1583–1588
- Gardner, T.S., Cantor, C.R., and Collins, J.J.: 'Construction of a genetic toggle switch in *Escherichia coli*', *Nature*, 2000, **403**, pp. 339–342
- Shea, M., and Ackers, G.: 'The OR control system of bacteriophage lambda. A physical–chemical model for gene-regulation', *J. Mol. Biol.*, 1995, **181**, pp. 211–230
- Blake, W.J., and Isaacs, F.J.: 'Synthetic biology evolves', *Trends Biotechnol.*, 2004, **22**, (7), pp. 321–324
- Goldberg, D.E.: 'Genetic algorithms in search, optimization and machine learning' (Addison-Wesley, 1989)
- Abramowitz, M., and Stegun, I.A.: 'Handbook of mathematical functions', (Applied Mathematics Series) (Washington: National Bureau of Standards, 1964, reprinted 1968 by Dover Publications, New York), vol. 55
- Sauro, H.M.: 'Jarnac: a system for interactive metabolic analysis' in Hofmeyr, J.-H.S., Rohwer, J.M., and Snoep, J.L. (Eds.): 'Animating the Cellular Map 9th International BioThermoKinetics Meeting' (Stellenbosch University Press, 2000, ISBN 0-7972-0776-7), pp. 221–228
- Sauro, H.M., Hucka, M., Finney, A., Wellock, C., Bolouri, C., Doyle, J., and Kitano, H.: 'Next generation simulation tools: the systems biology workbench and biospice integration', *OMICS*, 2003, **7**, (4), pp. 355–372
- Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D.: 'LAPACK users guide' (Society for Industrial and Applied Mathematics, Philadelphia, 1999)
- Ferrell, J.E.: 'Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability', *Curr. Opin. Chem. Biol.*, 2002, **6**, pp. 140–148
- Strogatz, S.H.: 'Nonlinear dynamics and chaos' (Westview Press, 1994)
- Kuznetsov, Y.: 'Elements of applied bifurcation theory' (Springer-Verlag, 1998)
- Doedel, E.J., Keller, H.B., and Kernevez, J.P.: 'Numerical analysis and control of bifurcation problems, Part I', *Int. J. Bifurcation Chaos*, 1991, **1**, (3), pp. 493–520
- Chickarmane, V., Paladugu, S.R., Bergmann, F., and Sauro, H.M.: 'Bifurcation discovery tool', *Bioinformatics*, 2005, **21**, (18), pp. 3688–3690
- Hervagault, J.F., and Canu, S.: 'Bistability and irreversible transitions in a simple substrate cycle', *J. Theor. Biol.*, 1987, **127**, pp. 439–449
- Goldbeter, A.: 'Biochemical oscillations and cellular rhythms' (Cambridge University Press, Cambridge, 1996)
- Sagues, F., and Epstein, I.R.: 'Nonlinear chemical dynamics', *Dalton Trans.*, 2003, **7**, pp. 1201–1217
- Goldbeter, A., Gonze, D., Houart, G., Leloup, J.C., Halloy, J., and Dupont, G.: 'From simple to complex oscillatory behavior in metabolic and genetic control networks', *Chaos*, 2001, **11**, pp. 247–260
- Goldbeter, A.: 'Computational approaches to cellular rhythms', *Nature*, 2002, **420**, pp. 238–244
- Franklin, G.F., Powell, J.D., and Emami-Naeini, A.: 'Feedback control of dynamic systems' (Prentice-Hall, 2002)
- Wilhelm, T., and Heinrich, R.: 'Smallest chemical reaction system with Hopf bifurcation', *J. Math. Chem.*, 1995, **17**, pp. 1–14
- Tyson, J.J., Chen, K.C., and Novak, B.: 'Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell', *Curr. Opin. Cell. Biol.*, 2003, **15**, (2), pp. 221–231
- Yi, T.M., Huang, Y., Simon, M.I., and Doyle, J.: 'Robust perfect adaptation in bacterial chemotaxis through integral feedback control', *PNAS*, 2000, **97**, pp. 4649–4653
- Rao, C.V., Wolf, D.M., and Arkin, A.P.: 'Control, exploitation and tolerance of intracellular noise', *Nature*, 2002, **420**, pp. 231–237
- Blake, W.J., Kaern, M., Cantor, C.R., and Collins, J.J.: 'Noise in eukaryotic gene expression', *Nature*, 2003, **422**, pp. 633–637
- Korobkova, E., Emonet, T., Vilar, J.M.G., Shimizu, T.S., and Cluzel, P.: 'From molecular noise to behavioural variability in a single bacterium', *Nature*, 2004, **428**, pp. 574–578
- McAdams, H.H., and Arkin, A.: 'It's a noisy business! Genetic regulation at the nanomolar scale', *Trends Genet.*, 1999, **15**, pp. 65–69