A case study of evolutionary computation of biochemical adaptation

2008 Phys. Biol. 5 026009

(http://iopscience.iop.org/1478-3975/5/2/026009)

Phys. Biol. **5** (2008) 026009 (12pp) doi:10.1088/1478-3975/5/2/026009

A case study of evolutionary computation of biochemical adaptation

Paul François and Eric D Siggia

Center for Studies in Physics and Biology, The Rockefeller University, 1230 York Avenue, 10065 New York, NY, USA

Received 11 April 2008
Accepted for publication 2 June 2008
Published 24 June 2008
Online at stacks.iop.org/PhysBio/5/026009

Abstract

Simulations of evolution have a long history, but their relation to biology is questioned because of the perceived contingency of evolution. Here we provide an example of a biological process, adaptation, where simulations are argued to approach closer to biology. Adaptation is a common feature of sensory systems, and a plausible component of other biochemical networks because it rescales upstream signals to facilitate downstream processing. We create random gene networks numerically, by linking genes with interactions that model transcription, phosphorylation and protein–protein association. We define a fitness function for adaptation in terms of two functional metrics, and show that any reasonable combination of them will yield the same adaptive networks after repeated rounds of mutation and selection. Convergence to these networks is driven by positive selection and thus fast. There is always a path in parameter space of continuously improving fitness that leads to perfect adaptation, implying that the actual mutation rates we use in the simulation do not bias the results. Our results imply a kinetic view of evolution, i.e., it favors gene networks that can be learned quickly from the random examples supplied by mutation. This formulation allows for deductive predictions of the networks realized in nature.

1. Introduction

Evolution is a retrospective theory, it explains the similarities among genes or organisms as a consequence of common ancestry. However, if one wants to predict the sequence of a protein that has a desired activity, or a gene network that realizes some function, Darwinian evolution does not provide an answer. The impediments to true prediction in biology are (i) lack of a mathematical expression for the fitness, the quantity to be optimized, (ii) ignorance of mutation rates and (iii) inability to infer phenotype from genotype. In this paper we will show how these difficulties can be overcome for the problem of adaptation.

The choice of fitness (more accurately termed a fitness function since it assigns a number to an arbitrary genetic network) is analogous to setting up a genetic screen. It defines the problem in quantitative terms. Our ignorance of the mutation rates is circumvented by showing for the chosen fitness function that there is a smooth monotone path leading to at least a good local optimum in the fitness. So like a rock tumbling down a valley, the bottom is unique even though the path leading to it may have arbitrary jigs and jags.

1.1. Defining a proper fitness function

Genetic algorithms have become a standard component in the toolbox of techniques for nonlinear optimization of a cost function. They mimic biology by following a 'population' of potential solutions that are 'mutated' by changing each one into a related solution in a random way. The most 'fit' solutions, as measured by the cost function, are replicated and replace the less fit ones. The topography of the cost function and the way mutations sample it control the convergence rate to an optimum. If the topography is a funnel leading to a unique minimum, convergence for any reasonable mutation process is assured. For a 'golf course' with multiple local minima and no cues about proximity to any one of them, there will be long periods of neutral evolution and occasional trapping.

Both the fitness function and the mutation process are hard to quantify in biology. For simulations, we need a precise fitness function that can be evaluated for any system even if far from optimum. The stock answer of reproductive success is not helpful since it depends on too many unknown factors, when in fact all we can hope to evolve is some small part of an organism such as a signaling pathway. Our

definition of fitness should be guided by a salient fact of organic evolution that pathways do not evolve in isolation, but are shared among many species and reused in many contexts. The fitness function should select for the generic features of the system, and ideally provide cues that will direct an arbitrary genetic network along a path of continuous increasing fitness. This both makes for efficient numerical search but also has implications for how biological evolution may operate.

In some cases there is a natural fitness function, such as visual acuity in the case of the eye [1], from which the authors demonstrated a monotonic adaptive path from a photosensitive patch to a compound eye. Another plausible fitness function was segment number for somitogenesis [2]. Interesting clock and bistable systems were obtained in [3] and particular networks for adaptation in [4], but in both cases the choice of fitness function was ad hoc.

More typical is the situation where multiple criterion impinge on fitness and the tradeoffs among them are not apparent and likely organism or context dependent. It is not a matter of imposing additional numerical criterion on a geometric quantity such as segment number (e.g., demanding segments of a particular spacing or size). If biological clocks evolved to anticipate light–dark cycles, how should one weigh the persistence of the anticipation, the strength of the response, its sensitivity to the phase of the light source and not its amplitude, etc. To only reward the perfect circadian oscillator would reduce the topography of the fitness landscape to a 'golf course', and render a computational study either infeasible or irrelevant to biology. Organisms certainly benefit from a less than perfect clock.

Evolutionary computation has to be generalized to optimize an entire class of fitness functions. For adaptive circuits we can solve this generalized problem and show there exists a path in the parameter space of our networks that continuously improves both components of the fitness function.

1.2. Adaptation in biology

It is convenient to describe adaptation as a feature of gene network that converts an *input*, e.g., a ligand, into an *output*, e.g., a transcriptional response. Adaptation occurs when a network registers a transient response to a rapid jump of input between two nearly constant states. Perfect adaptation ensues when the output returns to precisely the same value whenever the input is constant, no matter what its value.

A classic example of adaptation in sensory transduction is the Ca-mediated desensitization of the photoreceptors in vertebrates [5, 6] and allows us to perceive gradients over a wide range of ambient illuminations. This is not perfect adaptation in that we still sense the average light level. The MAPK pathway when stimulated by a step in the EGF concentration will generate a pulse in active ERK and then return to near-background levels [7]. Both processes are driven by negative feedback [8]. Bacterial chemotaxis exhibits perfect adaptation, in that the tumble frequency is independent of the mean chemo-attractant level [9]. Adaptation is useful to the organism in that it extends the range of the sensory system. A quantitative analysis of biologically plausible adaptive circuits was presented in [10] along with a refined characterization of their dynamics. An evolution simulation asks the converse question, what networks are preferred if certain criterion are imposed.

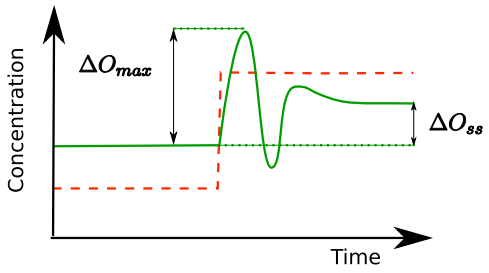

Figure 1. Illustration of input and output relationships for a general network. The (external) input of the network is displayed in dashed red while the output is in green.

1.3. Mathematical definition of adaptation

We simulate networks of interacting genes and proteins as a system of differential equations. To evolve an adaptive circuit within this system, we designate a prescribed time-dependent 'input' concentration, and a second 'output' concentration from which the fitness is derived. The first requirement for perfect adaptation is that for constant input I, the output should tend to a constant $O(I)$; so any fitness function should penalize persistent time dependence.

Next, fix the concentration of the input to I_1, far in the past so O assumes a steady state $O(I_1)$, and at $t = 0$ change I to I_2, and follow O till it reaches a new steady state $O(I_2)$. For adaptive networks, the deviation

$$\Delta O_{ss} = |O(I_2) - O(I_1)| \tag{1}$$

should be small and ideally zero.

Although $\Delta O_{ss} = 0$ is necessary for perfect adaptation, it is not sufficient, since the trivial network where O is independent of the input satisfies the condition. So we also have to impose that O varies if the input changes. We quantify the change in O after the input I switches from I_1 to I_2 at $t = 0$, in terms of

$$\Delta O_{\max} = \max_{t>0}|O(t) - O(I_1)|. \tag{2}$$

This quantity defines the response of the network to a change of the input. We do not care about the sign of the response or whether it is monotonic. Intuitively, a bigger response should correspond to a better fitness. More generally, when our model input consists of a series of plateau, (1) is the difference in O between the first and last time on each plateau, and the maximum in (2) is taken over one plateau at a time. Both functions should be averaged separately over all plateau.

To sum up, a network is perfectly adapting if and only if

- it responds to the input: $\Delta O_{\max} > 0$;
- its steady states concentration does not depend on the input, i.e. $\Delta O_{ss} = 0$.

Thus, evolving adaptation requires the optimization of these two quantities. Since fitness is one function we have to combine these two quantities, e.g.,

$$f_a = \Delta O_{ss} - \Delta O_{\max}, \tag{3a}$$

$$f_b = \Delta O_{ss} + \epsilon/\Delta O_{\max}, \tag{3b}$$

$$f_c = (\Delta O_{ss} + \epsilon)/\Delta O_{\max}, \tag{3c}$$

$$f_d = (\Delta O_{ss} + \epsilon O_{\mathrm{av}})/\Delta O_{\max}, \tag{3d}$$

where ϵ is a fixed number, small in comparison with typical values of O, and O_{av} is the average value of O over the time of simulation. Note by analogy with statistical physics we have defined the *fitness* as a cost function to be *minimized*.

Any reasonable fitness function such as those in (3) favors ΔO_{ss} small and $\Delta O_{\max}$ large but assumes different tradeoffs between them. For example, an evolutionary path that is monotone decreasing for one fitness function may not be monotone for another, even if the optima are the same.

Still, some fitness functions are more meaningful than others. For instance, (3*a*) and (3*b*) are clearly dependent on the absolute value of the output. In the limit $\epsilon \to 0$, fitness (3*c*) is dimensionless and therefore is less arbitrary. The ϵ term has been added to prevent an ambiguous ratio of 0/0. Once $\Delta O_{ss} = 0$, the ϵ term pushes $\Delta O_{\max}$ to higher values. Fitness (3*d*) is also dimensionless, but tends to select for big responses relative to the average concentration of output O_{av}. For random networks relaxing exponentially toward a steady state, $\Delta O_{\max} \simeq \Delta O_{ss}$, so that a ratio fitness, such as (3*c*) and (3*d*), correctly assigns a (maximum) value 1 to all such networks. Later in evolution, any increase of $\Delta O_{\max}$ relative to ΔO_{ss} will be selected.

2. Results

Our evolutionary algorithm was described previously [2]. It models transcription as a Hill function with three parameters: a maximum rate, a binding affinity between transcription factor and promoter and an exponent defining the nonlinearity. The steps between transcription and the appearance of new protein are modeled as a time delay [11, 12]. Proteins can form complexes as defined by forward and backward rates. We define a phosphorylation interaction between kinase and substrate by a Hill–Michaelis–Menten forward rate (with an exponent to model cooperativity) and a constant dephosphorylation rate. Explicit formulae are given in the appendix.

Mutations can create or remove genes, change the topology of the interaction network by adding or removing interactions or change parameters of existing genes and interactions. A 'population' consists of typically 50 different networks. After one round of mutation, the most-fit half of the population is copied and replaces the remainder, and the process repeats.

The input as a function of time is a series of level plateau, whose values are log-normally distributed and whose average duration imposes an upper bound on the timescale on the network. The metrics ΔO_{ss} and $\Delta O_{\max}$ are averaged over all the plateau and used in equations (3). There is an overall timescale in our simulations that will adjust to fit the response within the duration of the steps in the input time course. This feature of the environment is directly mapped into the evolved parameters by any of our fitness metrics.

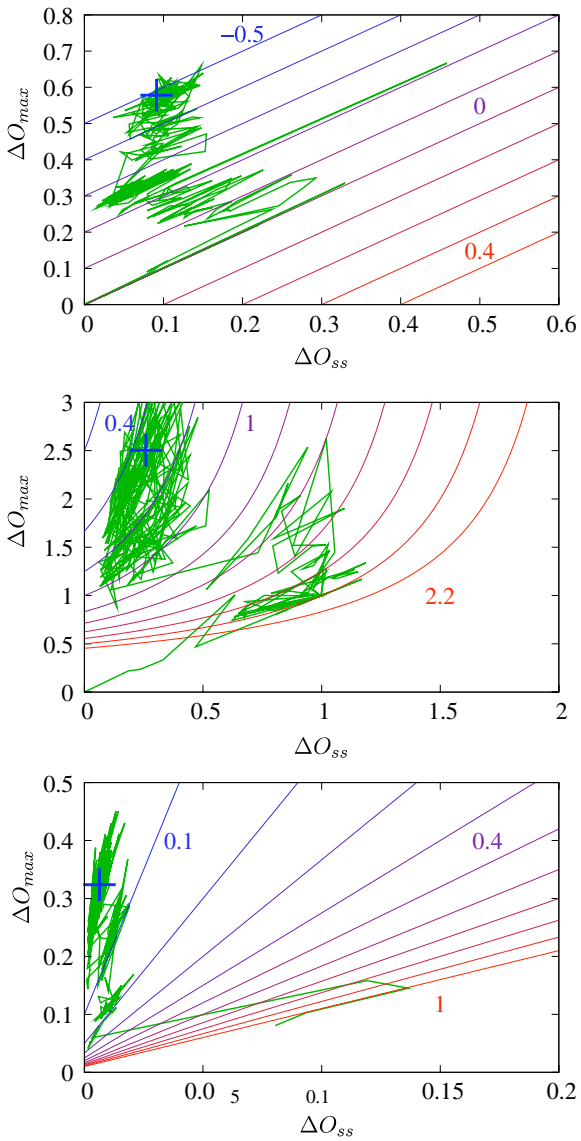

Figure 2. Typical evolutionary trajectories for fitness functions (3*a*), (3*b*) and (3*d*) from top to bottom. The fitness contour lines are smooth and equally spaced with red high and blue low as labeled. A typical evolutionary trajectory is shown in green evolving toward low values and ending at the blue cross. Almost perfect adaptive networks are evolved in 400 generations with similar topologies for all the three functions.

Similar results were obtained for all fitness functions in (3) as illustrated in figure 2. We next discuss the two classes of networks obtained and then show that they do not depend on the fitness function, mutation rates or the statistics of the inputs.

2.1. Type I: buffered output type

Example: figure 3

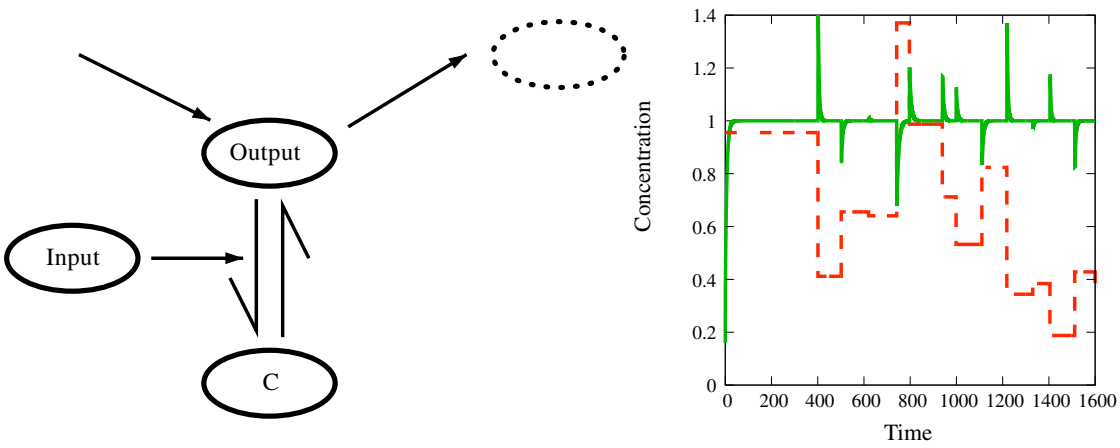

Figure 3. Network topology of type I. Sketch of the topology of the network with the typical behavior of the output (green) under the control of a random input (dashed red) as a function of time (parameters for this simulation are $\rho = 1, \epsilon = 2, \delta_O = 1, d = 0.5, \delta_C = 0$ and $f(O) = O$). Only the output undergoes constitutive degradation in the limit of perfect adaptation.

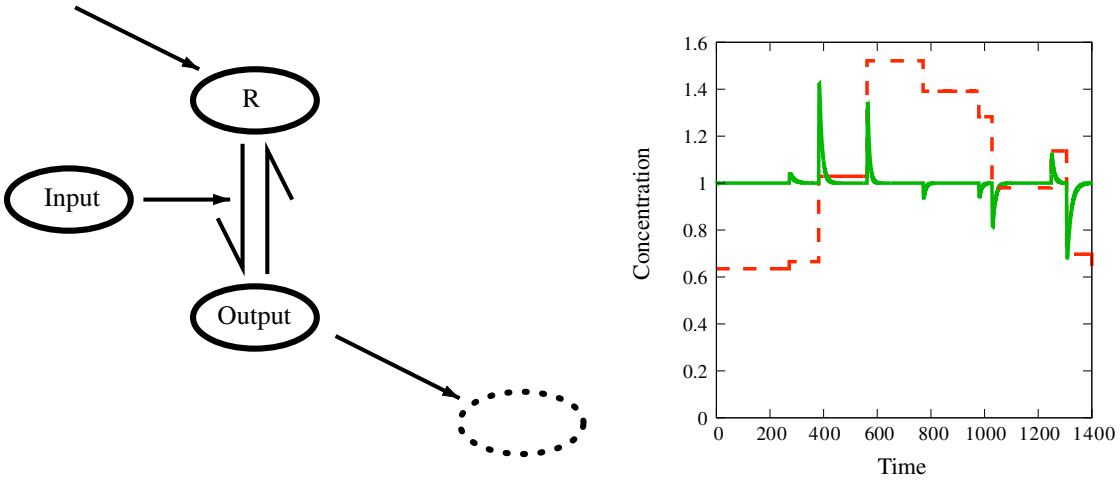

Figure 4. Network topology of type II. Sketch of the topology of the network with the typical behavior of the output (green) under the control of a random input (dashed red) as a function of time (parameters for this simulation are $\rho = 1, \epsilon = 0.1, \delta_O = 1, d = 0, \delta_R = 0$ and $f(R) = R$).

$$\dot{O} = \rho - \epsilon f(O) I + \mathrm{d}C - \delta_O O, \tag{4}$$

$$\dot{C} = \epsilon f(O) I - \mathrm{d}C - \delta_C C, \tag{5}$$

where f is an arbitrary function describing the coupling of the input to the output ($f(O) = O$ for protein–protein interactions or $f(O) = O^n / \left(O_0^n + O^n\right)$ for catalytic interactions; both kind of networks were obtained) and ϵ is our generic symbol for the strength of interaction. Selection forces δ_C to zero.

2.2. Type II: ligand–receptor type

Example: figure 4

$$\dot{R} = \rho - \epsilon f(R) I + \mathrm{d}O - \delta_R R, \tag{6}$$

$$\dot{O} = \epsilon f(R) I - \mathrm{d}O - \delta_O O. \tag{7}$$

Computational evolution sets δ_R to zero.

3. Why are these topologies selected?

We need to show both that the final network is adaptive and explain why the evolutionary algorithm is able to find it independently of the mutation parameters. Let us consider a sufficiently general equation of the type I networks:

$$\dot{O} = \rho - \epsilon O I + \mathrm{d}C - \delta_O O, \tag{8}$$

$$\dot{C} = \epsilon O I - \mathrm{d}C - \delta_C C. \tag{9}$$

Now, in this problem, there is a clear correspondence between the physical parameters in the equation and the two metrics comprising the fitness functions:

- $\Delta O_{\max} > 0$ iff $\epsilon > 0$: there must be an interaction between O and I to make the output depend on the input;
- at steady state, $O_{ss} = \frac{\rho}{\delta_O + \frac{\delta_C \epsilon I}{d + \delta_C}}$, so that $\Delta O_{ss} = 0$ for all I iff $\delta_C = 0$ or $\epsilon = 0$.

Any plausible fitness function for this problem should increase $\Delta O_{\max}$: it is therefore obvious that evolution should

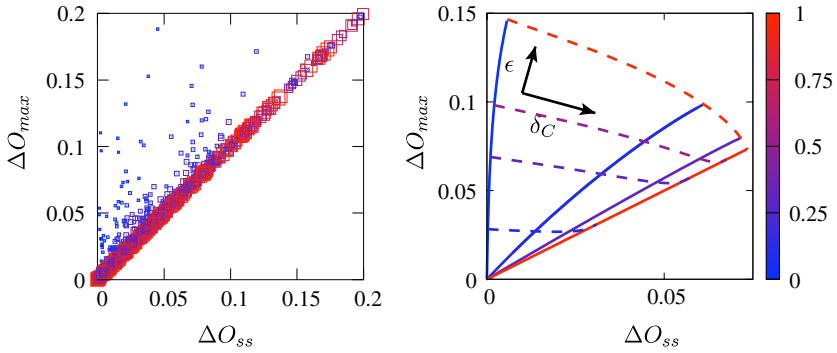

Figure 5. Variation of $\Delta O_{\max}$ and ΔO_{ss} with parameters for the type I networks. The left panel shows the parameter sampling used in the simulations with color indicating values of δ_C as seen in scale bar. Smaller sizes correspond to smaller values of δ_C. The right panel shows typical trajectories while varying only one parameter, δ_C, at fixed epsilon (dashed, $\epsilon = 0.1, 0.3, 0.5, 1$) and ϵ at fixed δ_C (solid, $\delta_C = 0, 0.1, 0.3, 1$), the other parameters being held constant. Arrows indicate the directions of increasing δ_C and ϵ.

choose $\epsilon > 0$. In other words, this interaction and thus the network topology will be selected as the first step of evolution.

It can also be seen that the fitness should decrease monotonically as δ_C decreases to 0 from the expression for O_{ss} when expanded for small δ_C:

$$\Delta O_{ss} \simeq \frac{\rho}{\delta_O^2}\frac{\delta_C}{d}\epsilon|\Delta I|.$$

In the opposite limit of large δ_C, $\dot{C}$ will be close to 0, so a quasi-equilibrium value of C can be substituted into the O equation:

$$\dot{O} = \rho - \epsilon O I - \delta_O O. \tag{10}$$

It is clear from this equation that O simply relaxes exponentially toward its steady state, so we have

$$\Delta O_{\max} = \Delta O_{ss} \simeq \frac{\rho\epsilon|\Delta I|}{(\delta_O + \epsilon I)^2}. \tag{11}$$

Since by definition $\Delta O_{\max} \geqslant \Delta O_{ss}$, the equality in (11) clearly corresponds to a (bad) limiting case; any move that breaks the equality of $\Delta O_{\max}$ and ΔO_{ss} should be evolutionary favorable. Thus, selection must decrease δ_C to decrease ΔO_{ss} (and increase the relative value of $\Delta O_{\max}$). This is also seen on the left panel of figure 5 when all parameters are randomized, adaptive networks are obtained only for low values of δ_C.

We can also compute an estimate of the optimum $\Delta O_{\max}$. Let us suppose that $\delta_C = 0$. The network will have a maximum response if O relaxes quickly ($\delta_O + \epsilon \gg 1$), while C relaxes very slowly ($d \ll 1$) toward its steady state value. Shortly after the jump from I_1 to I_2, when O reaches its maximum $O_{\max}$, C has not moved appreciably from its pre-jump value, C_1. Using $O_{ss} = \rho/\delta_O$ the stationary value of O, one has

$$\begin{aligned} 0 &= \rho - \epsilon O_{\max} I_2 + \mathrm{d}C_1 - \delta_O O_{\max} \\ &= \delta_O O_{ss} - \epsilon O_{\max} I_2 + \epsilon O_{ss} I_1 - \delta_O O_{\max}, \end{aligned}$$

and one gets

$$O_{\max} = O_{ss}\frac{\delta_O + \epsilon I_1}{\delta_O + \epsilon I_2}, \tag{12}$$

$$\Delta O_{\max} \simeq -O_{ss}\frac{\epsilon}{\delta_O + \epsilon I_2}\Delta I. \tag{13}$$

Note that $\Delta O_{\max}$ is an increasing function of ϵ under our assumptions. However, the typical timescale of response will be controlled by d, which cannot be too slow if we want a quick response (i.e., within the duration of the plateau I_2). The statistics of the plateau duration enter here since a larger $\Delta O_{\max}$ may be realized by tolerating an incomplete response for very short plateau so that d can be smaller and increase the response from more typical plateau.

From the right panel of figure 5, we can see that for typical parameters there is no tradeoff between increasing $\Delta O_{\max}$ and decreasing ΔO_{ss}: increase of $\Delta O_{\max}$ is not only relative to ΔO_{ss}, but also absolute all other parameters being fixed, i.e. by decreasing δ_C, one moves toward the upper-left corner of the plane displayed in figure 5.[1] This is the reason why any fitness function tending to maximize $\Delta O_{\max}$ while minimizing ΔO_{ss} should be able to select for this network.

From all these considerations, one can predict that (figure 5)

- evolution will select the topology first since $\epsilon > 0$, $\Delta O_{\max} > 0$, and thus create the other parameters on which selection can act,
- once the topology is selected, evolution should decrease δ_C, since $\Delta O_{ss} = 0$ iff $\delta_C = 0$ and since decreasing δ_C both increases $\Delta O_{\max}$ and decreases ΔO_{ss},
- evolution should increase ϵ and adjust the various rates to maximize $\Delta O_{\max}$, in response to features of the input such as its typical timescale.

Similar arguments can be given for the other network topology (figure 4) that evolved for which a sufficiently general system of equations is

$$\dot{R} = \rho - \epsilon R I + \mathrm{d}O - \delta_R R, \tag{14}$$

$$\dot{O} = \epsilon R I - \mathrm{d}O - \delta_O O. \tag{15}$$

Now, in a similar way

- O is defined as the reaction between R and I so trivially $\Delta O_{\max} > 0$ iff $\epsilon > 0$, and there will be a response of O to I;

[1] The increase in $\Delta O_{\max}$ when δ_C decreases can made plausible by taking the ratio between equations (11) and (13), to yield $1 + \epsilon I/\delta_O$ which is always bigger than 1.

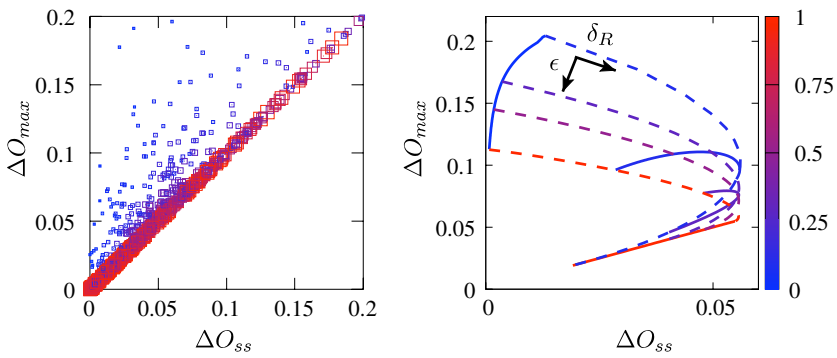

Figure 6. Variation of $\Delta O_{\max}$ and ΔO_{ss} with parameters for the type II networks. The left panel shows the parameter sampling used in the simulations for different values of δ_R. The panels follow the same conventions as in figure 5.

- at steady state, $O_{ss} = \frac{\rho}{\delta_O + \frac{\delta_R(d+\delta_O)}{\epsilon I}}$, so that $\Delta O_{ss} = 0$ for all I iff $\delta_R = 0$.

Given the interaction, it is very easy to evolve parameters to get an adaptive network: as before, it is clear that ΔO_{ss} is an increasing function of δ_R, so that one has to minimize δ_R to minimize the fitness. When δ_R is large the R equation is at quasi-equilibrium, so R is a linear function of O and the O equation describes exponential decay to its steady state. So reducing δ_R also increase $\Delta O_{\max}$ with respect to ΔO_{ss}.

Now, if $\delta_R = 0$, again one can estimate $\Delta O_{\max}$. Assuming that O responds more rapidly than R, then R will still be approximately R_1 when O attains its maximum $O_{\max}$ and $\dot{O} = 0$. So we can solve for $O_{\max}$:

$$0 = \epsilon I_2 R_1 - \mathrm{d} O_{\max} - \delta_O O_{\max}$$
$$= I_2/I_1(d + \delta_O) O_{ss} - (d + \delta_O) O_{\max},$$

i.e.

$$O_{\max} = (I_2/I_1) O_{ss}$$

or $\Delta O_{\max} \simeq O_{ss} \Delta I/I_1$.

This represents a clear physical limit on the maximum value O can reach for any set of parameters. Thus evolution will favor rapid O and slow R responses, which can be realized by several different parameter combination. Again, there is no competition between minimizing ΔO_{ss} and increasing $\Delta O_{\max}$ in figure 6.[2]

The shapes of the ϵ and δ_R contour lines in figure 6 are partially due to the breakdown in the quasi-static assumptions made above. For $\delta_R \sim 0$, increasing ϵ decreases the response time of R and thus allows ΔO_{ss} more time to reach its asymptotic value of 0. But a more rapid response of R limits the maximum excursion of O and thus $\Delta O_{\max}$. For large δ_R, the curve parameterized by ϵ is double valued over ΔO_{ss}. The upper branch has the same rational as for small δ_R. On the lower branch for small ϵ, the system behaves as if O is a function of I and thus $\Delta O_{\max}$ and ΔO_{ss} are proportional.

[2] In the limit of big δ_R, it is clear from equations (14) and (15) that R, O and therefore $\Delta O_{\max}$ are zero so one has to decrease δ_R to have nonzero $\Delta O_{\max}$.

4. Networks with constraints on the types of inputs

The previous examples did not place any restrictions on the types of input, and the evolution selected an input that could either participate in a protein complex or act catalytically, the input was never a transcription factor. What happens if we constrain the input to act transcriptionally but let evolution define all other parameters?

Again we found adaptation, with one predominant topology shown in figure 7 along with the fitness as a function of time.

The general equations of these network are

$$\dot{O} = f(I)(t - \tau_O) - \epsilon R O + \mathrm{d} C - \delta_O O, \tag{16}$$
$$\dot{R} = g(I)(t - \tau_R) - \epsilon R O + \mathrm{d} C - \delta_R R, \tag{17}$$
$$\dot{C} = \epsilon R O - \mathrm{d} C - \delta_C C, \tag{18}$$

where we make explicit the temporal lags τ_O, τ_R between the protein concentration at the promoter and the appearance of new protein.

4.1. Why is this topology selected?

Since three interactions are needed to build this network versus only one for the type I and II networks, it was necessary to constrain the type of input to eliminate the latter two networks. It is less obvious why there is a downward path in fitness favoring it. As before, since evolution favors $\Delta O_{\max} > 0$, the first mutation will be for the regulation of I by O. Then a neutral mutation can create another protein R under the control of I. We then have to show why a protein–protein interaction between O and R (i.e. $\epsilon > 0$) is favorable.

At steady state, one clearly has the following relationships:

$$0 = f(I) - \frac{O}{O + \lambda} g(I) - \delta_O O, \tag{19}$$

with $\lambda = \frac{\delta_R(d+\delta_C)}{\delta_C \epsilon}$. For small ϵ (large λ), one therefore has at lowest order

$$O_{ss} \simeq \frac{f(I)}{\delta_O + g(I)/\lambda} \simeq \frac{f(I)}{\delta_O} \left(1 - \frac{g(I)}{\delta_O \lambda}\right),$$

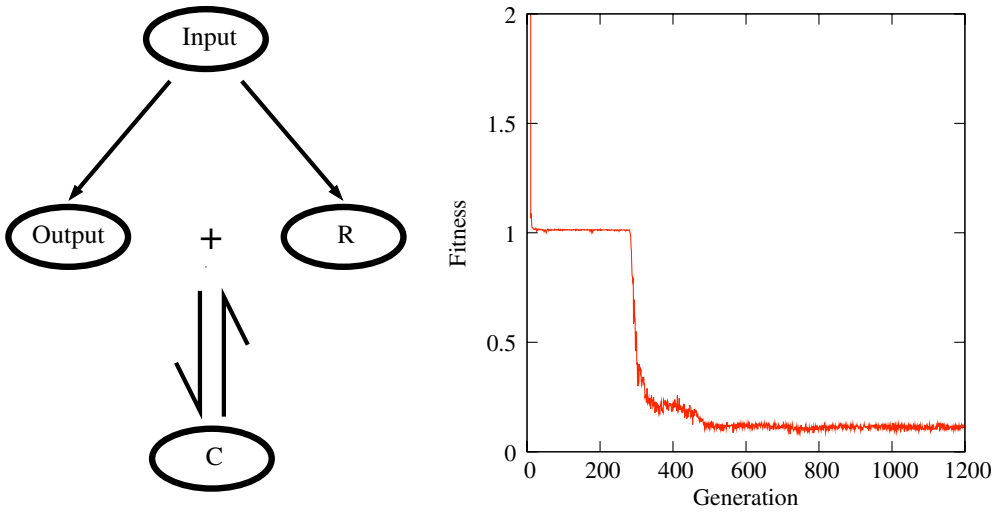

Figure 7. Topology of evolved networks when the input is a transcription factor. Right panel: typical evolution of fitness (3*d*) for this simulation. The first plateau occurs when activation of the output by the input is selected. When the protein–protein interaction appears, the fitness drops further and parameters are optimized to reach perfect adaptation.

so that

$$\Delta O_{ss} \simeq \left| \left(\frac{f'(I)}{\delta_O} - \frac{g(I)f'(I) + f(I)g'(I)}{\delta_O^2 \lambda} \right) \Delta I \right|.$$

From this expression, one clearly sees that ΔO_{ss} is reduced for $\epsilon > 0$ if f and g are both increasing or decreasing functions of I. This is assumed by our algorithm (and accords with biological norms) since most transcription factors (together with cofactors, and absent any covalent modifications) act either as activators or repressors.

To evaluate $O_{\max}$, we evaluate (16) and (17) in the quasi-static limit and assume for the lag times $\tau_O < \tau_R$, so that O responds before R does and returns O to the steady state (when the opposite inequality between the lag times applies, it can be shown that $\Delta O_{\max} = \Delta O_{ss}$):

$$\begin{aligned} O_{\max} &= \frac{f(I_2)}{\delta_O}\left(1 - \frac{g(I_1)}{\delta_O \lambda}\right), \\ \Delta O_{\max} &\simeq \frac{f'(I)}{\delta_O}\left(1 - \frac{g(I)}{\delta_O \lambda}\right)\Delta I \end{aligned} \tag{20}$$

or

$$\frac{\Delta O_{\max}}{\Delta O_{ss}} \simeq \frac{1 - \frac{g(I)}{\delta_O \lambda}}{1 - \frac{g(I)}{\delta_O \lambda} - \frac{g'(I) f(I)}{\delta_O \lambda f'(I)}}. \tag{21}$$

Because $g'(I)$ and $f'(I)$ have the same sign, $\Delta O_{\max} > \Delta O_{ss}$. So this evolutionary move increases $\Delta O_{\max}$ relative to ΔO_{ss}. However, there is an implicit tradeoff in that $\Delta O_{\max}$ decreases from its value with $\epsilon = 0$. Both fitness functions (3*c*) and (3*d*) avoid this problem since they score just the ratio of $\Delta O_{\max}$ and ΔO_{ss}.

Once this topology was selected, we verified many times numerically that perfect adaptation follows on a decreasing fitness path (see figure 7 for a typical example). Again this requires one species to be stable, i.e. $\delta_O = 0$, and the parameters in $g(I)$ and $f(I)$ changed so that $g(I) = \mu f(I)$. Then $O_{ss} = \lambda/(\mu - 1)$. The response is much stronger if the time lag in the O equation $\tau_O = 0$, while the lag in the R equation is large. When combined with $g(I) = \mu f(I)$ the input to the O equation becomes a finite time difference (cf equation (19)). However, if the input does not vary over too wide a range, it is possible to have approximate adaptation if δ_O is nonzero for some specific values of f and g (see figure 8 for examples of both situations). Plausibly a larger region of parameter space is compatible with approximate adaptation than the exact variety.

5. Extensions to more general networks

Subnetworks with the topology of types I and II can evidently be found in more complex circuits, but there is rarely specific evidence that their parameters are in an adaptive regime. We have obtained elaborations of the basic types which can appreciably amplify the response. Figure 9 generalized the type I network and obeys the equations

$$\dot{O} = \rho - f(O)I + d_C C - \delta_O O, \tag{22}$$

$$\dot{C} = f(O)I - d_C C - I g(C) + d_D D, \tag{23}$$

$$\dot{D} = I g(C) - d_D D. \tag{24}$$

A generalization of type II networks, figure 4, replaces the output by a transcription factor which drives the new output species (figure 10). If the parameters in the transcriptional interaction are adjusted properly, the output can be made to respond to only upward jumps in the input.

5.1. General adaptive networks

Reference [10] classifies adaptive networks loosely as feed forward where one input activates two parallel pathways which combine to generate an output, or feedback where the output negatively regulates itself. The feedback can either promote the decay of the output or block its activation by the input or both. Activation of a phosphatase is a common

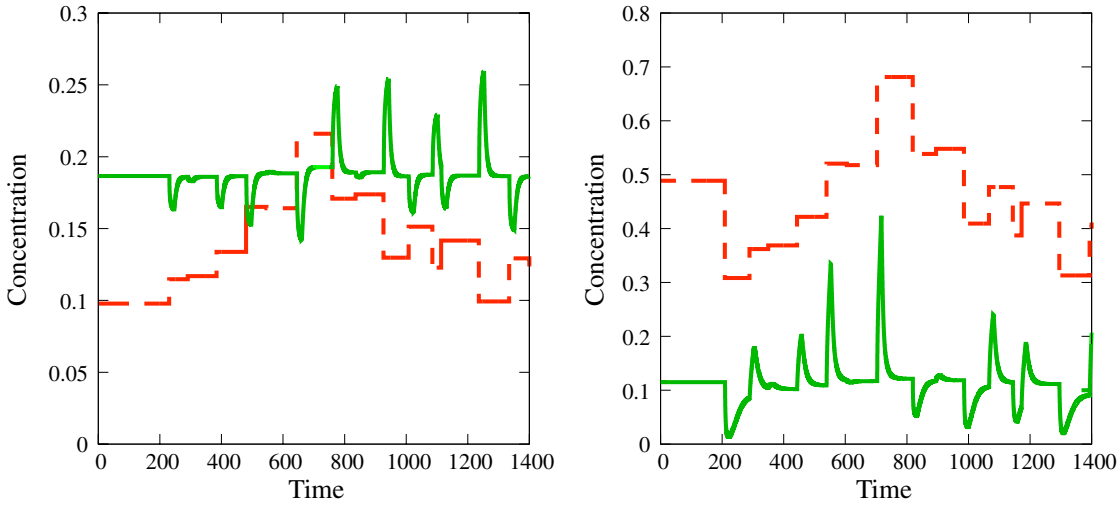

Figure 8. Behaviors of evolved networks from (16), (17) and (18). Left panel: dynamics for an example where non-linearities are different in f and g. Parameters are $f(I) = 0.6/(1 + I/0.54)$, $g(I) = 0.96/(1 + (I/0.7)^{1.4})$, $\delta_0 = 0.18$, $\delta_R = 0.14$, $\epsilon = 0.9$, $d = 0.3$, $\delta_C = 0.8$, $\tau_O = 0.8$ and $\tau_R = 16$. Right panel: example where nonlinearities on f and g have evolved independently to be very similar. Parameters are $f(I) = 0.4I^{4.4}/(0.67^{4.4}/ + I^{4.4})$, $g(I) = 0.98I^{4.2}/(0.68^{4.2}/ + I^{4.2})$, $\delta_0 = 0.01$, $\delta_R = 0.09$, $\epsilon = 0.8$, $d = 0.4$, $\delta_C = 0.8$, $\tau_O = 0$ and $\tau_R = 13.5$.

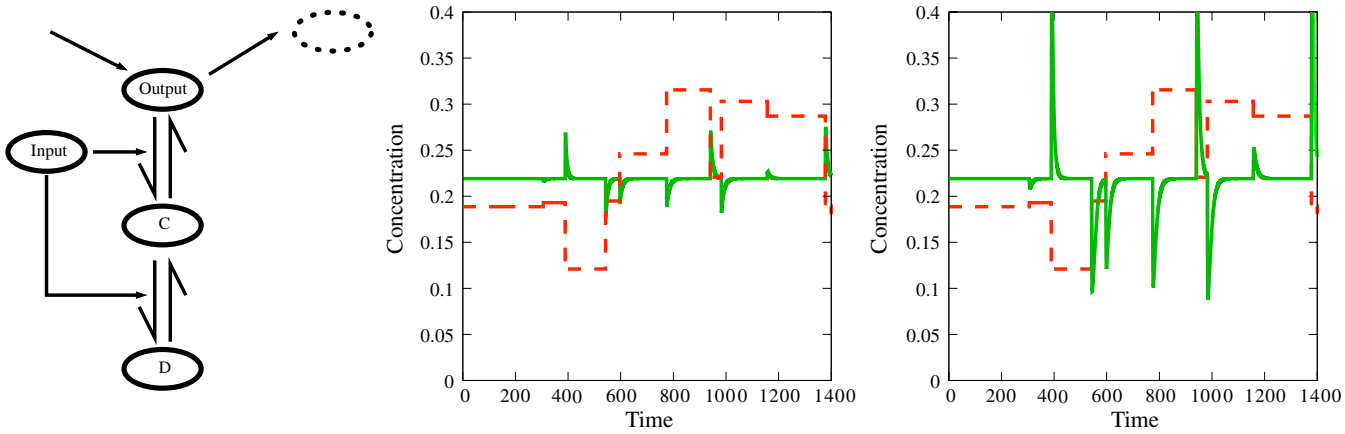

Figure 9. Amplification of the output from the type I network. Left panel: sketch of the network. Middle: dynamics for the network without the interaction creating D. Right panel: dynamics of the complete network. (Input has been rescaled for display on both panels.) Parameters are $\rho = 0.098$, $f(O) = 0.83O$, $d_C = 0.85$, $\delta_O = 0.44$, $g(C) = 0.76C^{1.75}/(0.05^{1.75} + C^{1.75})$ and $d_D = 0.75$.

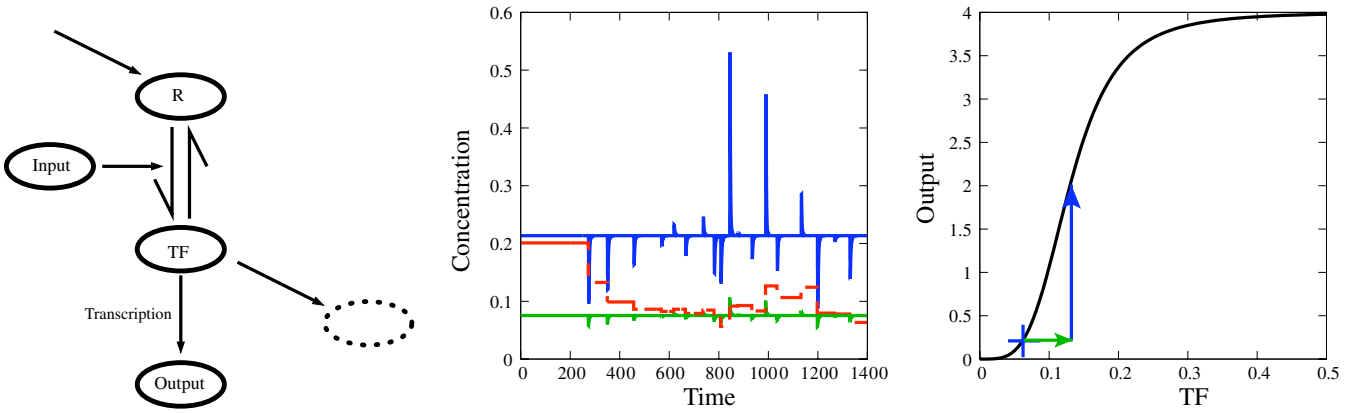

Figure 10. Amplification of the output of a type II by an additional transcriptional step, TF. Left: topology used. Middle: typical behavior of such a network (input in dashed red, TF in green, output in blue). Right: illustration of the mechanism of amplification. The steady state concentration of the output is displayed as a function of the TF level. The steady state of the system is low (blue cross), so that excursion of the TF into the nonlinearity (green arrow) triggers massive response of the output (blue arrow) relative to its steady state concentration.

way to deactivate pathways that signal by phosphorylation and receptor deactivation blocks further input after pathway activation.

Our network in figure 7 is loosely feed forward, the input activates the output and a second species, and the protein–protein interaction compares them. It actually implements a finite time difference of the input, so constant inputs produce no response. Most of the networks in [10] are based on phosphorylation–dephosphorylation, so the total concentration of any protein is constrained which makes it more difficult to achieve perfect adaptation.

Our type I and II networks are best understood as buffered, there is a continual production and decay of protein, pinning the output to a fixed value irrespective of the input, provided it is constant in time. A change in input transiently affects the production or decay rates of the output and therefore elicits a response. Output buffering is accomplished by the other variable in the network which records the absolute level of the input. In the type I networks $C \sim I$, while in type II $R \sim 1/I$. These networks parse the input into its derivative and mean amplitude.

A potentially biologically relevant distinction between the type I and II networks is how they respond to large jumps in input. The response of the former is $\sim I_2 - I_1$ (in the limit of large δ_O), while the latter respond as $\sim I_2/I_1$, i.e. linear versus logarithmic.

A generalization of networks of types I and II considers a set of species O_i that are inter-converted at rates governed by I, and which when summed satisfy

$$\sum \dot{O}_i = \rho - \sum \delta_i O_i. \tag{25}$$

At steady state, we have $\sum \delta_i O_i = \rho$, so this specific weighted sum of O_i is adaptive. To capture this sum into a single variable would require implementing the decay processes by some common interaction (e.g. protein–protein complex formation). The concentration of the common reactant would then itself adapt. Note that the concentration of any O_i with a nonzero decay constant in equation (25) is uniformly bounded for all values of I. In the type I and II networks, the designated adaptive output was one of the O_i in the sum, so the decay rate of the remaining variable had to be zero for adaptation. Thus nothing constrains the non-adaptive variable to be uniformly bounded irrespective of input level, and as already noted $C \sim I$ in type I, while $R \sim 1/I$ in type II.

6. Discussion

There is a large literature on 'digital life', compact programs that increase in copy number, mutate and complete for resources in a biologically inspired fashion [13, 14]. Evolutionary simulations also purport to illuminate more general issues of complexity, robustness and evolvability in biology [15–17].

Our focus is much more on engineering, namely specific differential equation models that perform specific tasks. For the results to be biologically credible requires a demonstration that the fitness function is realistic and the mutation rate parameters do not matter. The mathematical expression of the fitness function should be as general as possible and omit numerical constants, other than those that define scales of time and concentration.

To understand the space of plausible fitness functions, it is very helpful to parameterize it with multiple metrics. For vision, both acuity and brightness play a role, but their relative weight depends on the animals lifestyle [1]. For adaptation we choose conditions (1) and (2). Additional less important criterion might include a rapid response, which we imposed implicitly by randomizing the duration of the constant plateau in the input function, and limits on the total amount of protein used in the network, which we ignored. We assume then that for a particular network, organism and environment the true fitness is some function of these metrics. Mutations that improve all metrics should clearly be accepted. When tradeoffs are necessary, we need to pause and enumerate the limiting networks for which no further improvements in all metrics are possible. For adaptation this last step was not necessary, both metrics could be satisfied simultaneously.

A single fitness function that may seem natural for adaptation just forces O to behave as $\dot{I} + cst$, by means of a dimensionless linear correlation

$$1 - r(O, \dot{I}) = 1 - \left| \frac{\langle O\dot{I}\rangle - \langle O\rangle\langle \dot{I}\rangle}{\sqrt{(\langle O^2\rangle - \langle O\rangle^2)(\langle \dot{I}^2\rangle - \langle \dot{I}\rangle^2)}} \right|. \tag{26}$$

Its deficiencies from our perspective are that it favors an overly specific linear correlation between the instantaneous derivative of I with O at the same time. With a more continuous input time course, we have seen that (26) gives rise to networks of types I and II, but the amplitude of the response was free, not optimized, and in many cases very small.

Once a fitness space is defined in terms of several metrics, we can argue for the irrelevance of mutation rates if there is a monotone decreasing path to the desired state. If there are bifurcations in the path, and the fitness improves along both branches, then both arms have to be followed, since in reality the branch chosen will depend on the relative rates, which are unknown. On a serial path, the mutation rates only matter for the overall time, which is of no interest. When positive selection directs the parameter choice in a model, 'parameter tuning' is no longer a pejorative, it is how nature works.

Incremental evolution and arguments about the efficacy of positive selection suppose small mutational changes in the gene network. To check our analytic arguments we have run our simulations in this mode. But all adaptive networks were also obtained by taking random parameters (within fixed bounds) for each mutation. We can then argue that adaptive networks occupy a nonzero volume of parameter space. Of course a parameter cannot be set exactly to zero by random sampling, but the network can get close enough to give a good fitness. Small genetic changes can have many variable effects on parameters, e.g. there are dominant negative point mutations.

We evolved small adaptive networks, e.g. (3), (4) that can be inserted into larger gene circuits in any context, but it is often not clear if they occur with parameters that render them adaptive. One case where rates are known are G-protein

receptor kinases [18, 19]. They are very specific for activated receptors, and lead to the binding of arrestin and subsequent endocytosis. Thus they fall into our type II class with $\delta_R = 0$. If the cell receives stimuli with a repeat time much shorter than the recycling time of the receptors, then there will be an average steady state and the recycling process would satisfy our assumption of constitutive production (e.g., the ρ term in (14)). However, one instance of G-protein signaling that has been quantified does not exhibit adaptation [20].

Prior work on adaptive networks includes a large literature on bacterial chemotaxis [21–25], which is much more complex than anything we have evolved. Bacteria can chemotax over a huge, 10^5 range in ambient ligand levels and do so in part by receptor clustering [26] which is not included in our algorithm (though we can evolve receptor dimerization prior to activation). In addition, there is probably considerable signal amplification happening along the pathway from receptor to motor, and molecular noise could well be an important factor in defining the network. We have only evolved deterministic equations here, but could easily generalize to stochastic ones. Other specific features of the *Escherichia coli* chemotactic network, such as the four-receptor methylation states, are not necessary for adaptation; two will suffice. It remains an interesting question whether if we took the known network topology and evolved just the parameters from a random starting point, adaptation would emerge. There are many constrained parameters in all existing models: for instance methylation and demethylation rates are assumed independent of the methylation state, fully methylated receptor are always active, fully demethylated receptors are inactive, etc.

Soyer *et al* [4] evolved parameters with a fitness function correlating the derivative of the input with the output (and therefore similar in principle to (26)) for a phosphorylation network with fixed topology. Iglesias and co-workers [27, 28] wrote down several quadratically nonlinear adaptive systems in two variables that encompass our type I and II models in certain limits. They did not consider their evolution.

7. Outlook

More refined simulations of adaptive networks are possible by imposing more criteria on the selection. One can model stochasticity due to small numbers of molecules and then impose a cost based on the amount of protein used [29]. In stochastic simulations, selection for amplification can be combined with adaptation, with obvious relevance to signal transduction. One can also ask for adaption that minimizes the rates of protein production and decay to which we did not assign a cost. The questions and approach remain the same: are there obligatory tradeoffs between the multiple criterion, and can an adequate solution to the problem be achieved by stepwise positive selection.

Evolution driven by selection is fast, as Darwin understood intuitively, and [1] made explicit for the eye. The alternative metaphor of multiple fitness peaks created by epistasis makes transitions much more contingent on unknowable parameters implicit in moving through fitness valleys.

With the exception of [2], all efforts we are aware of to evolve simple genetic networks do not control for biases implicit in the fitness function and the mutation rates. Adaptation has provided a clear illustration of how to treat a family of fitness functions, and an example where mutation rates demonstrably do not matter for the final outcome. If pathways in developmental biology can be recovered in general terms by our methods, it implies that evolution selects for structures that can be learned rapidly from the random changes supplied by mutation. Better structures that cannot be reached incrementally are invisible. Evolution recast as a search for networks that can be found by incremental selection may become a deductive theory.

Acknowledgments

This work was supported by an NSF grant DMR-0129848 to ES, and Lavoisier Fellowship to PF. We thank J Skotheim for comments on the manuscript, and John Guckenheimer and William Bialek for discussions.

Appendix A. List of interactions with equations

The following table summarizes all possible interactions that our algorithm can choose, with corresponding equations and ranges of variation for parameters:

Example of interaction	Equations	Authorized range of parameters variation
Degradation of protein A	$\frac{dA}{dt} = -\delta_A A$	$\delta_A \in [0, 1]$
Unregulated protein B production	$\frac{dB}{dt} = \rho_B$	$\rho_B \in [0, 1]$
Transcriptional regulation of protein B by protein A	$\frac{dB}{dt} = \rho_B \frac{1}{1+(A(t-\tau_B)/A_0)^n}$	$A_0 \in [0, 1]$ $n \in [-5, 0]$ for activation $n \in [0, 5]$ for repression $\tau_B \in [0, 20]$ $\rho_B \in [0, 1]$
Dimerization: A and B form a dimer C	$\frac{dA}{dt} = -\epsilon AB + dC$ $\frac{dB}{dt} = -\epsilon AB + dC$ $\frac{dC}{dt} = \epsilon AB - dC$	$\epsilon \in [0, 1], d \in [0, 1]$
Phosphorylation: K phosphorylates protein A into protein A^*	$\frac{dA}{dt} = -\epsilon K \frac{(A/A_0)^n}{1+(A/A_0)^n} + dA^*$ $\frac{dA^*}{dt} = \epsilon K \frac{(A/A_0)^n}{1+(A/A_0)^n} - dA^*$	$\epsilon \in [0, 1]$ $d \in [0, 1]$ $A_0 \in [0, 1]$ $n \in [0, 5]$

When multiple transcriptional activators are present, their combined output is defined by the maximum of their activities. Repressors are combined multiplicatively. This is analogous to using an OR between activators and an AND between repressors in discrete systems.

All these interactions are combined to create the final equations. Suppose for instance that P is repressed by R, activated by A_1 and A_2 and phosphorylated by K, the P reaction rate would typically look like

$$\frac{\mathrm{d}P}{\mathrm{d}t} = \left(\max\left[\frac{A_1^{n_1}}{A_1^{n_1} + (A_{1P})^{n_1}}, \frac{A_2^{n_2}}{A_2^{n_2} + (A_{2P})^{n_2}} \right] \times \frac{\rho_P}{1 + (R/R_P)^{n_3}} \right)(t - \tau_P) - \epsilon K \frac{(P/P_0)_4^n}{1 + (P/P_0)_4^n} + \mathrm{d}P^* - \delta_P P. \quad \text{(A.1)}$$

Appendix B. Algorithm overview

The strategy is to evolve genetic networks by repeated rounds of selection, growth and mutation, and is very similar to our previous works [2, 3].

Typically 50 different networks are followed in parallel. At each step of the algorithm, differential equations corresponding to each network are integrated with random inputs as described in the main text. Then, the fitness function is computed from the final state of each network.

After the fitness function is computed, the 25 best networks are retained (selection), copied (growth) and mutated (mutation). Mutations are chosen randomly from among the following types:

- creation or removal of an existing gene,
- creation or removal of any regulatory linkages (transcriptional regulations, protein–protein interactions, phosphorylations),
- change of any kinetic parameter,
- change of the reporter proteins on which the fitness function is computed (i.e. change of the output).

Kinetic parameters are changed by resampling uniformly from a predefined range from 0 to a maximum of $\mathcal{O}(1)$ times arbitrary constants defining the units of time and concentration. Mutations that change topology (add or remove a gene or interaction) occur at one-tenth the rate of parameter changing moves. As explained in the main text and described earlier [2], results are insensitive to mutation rates. The degradation rates in particular never converge to their upper limits.

After the mutation step, the entire process is iterated. A 'generation' is one iteration of this selection/growth/mutation process. Since the creation of genes and interactions are separate mutations, a gene may be created which has no effect on the output and thus a neutral change. This gene may disappear in subsequent generations, or may be linked by an interaction to favorably affect the output. It will then be favored by selection and may ultimately fix in the population.

We used a simple Euler algorithm for time integration, to simply take into account transcriptional delays. The integration time step was typically 10% of the minimum timescale set by the kinetic parameters (i.e. for most cases since the range for degradation constants and most kinetic parameters is [0, 1], the time step used is 0.1).

Specific tools have been developed in PYTHON to define, encode and modify genetic network structures and evolution parameters. A PYTHON translator has been developed to automatically write small C programs encoding the dynamics of each network, compile and run it to integrate equations and compute the fitness function. This computational strategy was used to minimize the overall computational time for evolution.

References

[1] Nilsson D E and Pelger S 1994 A pessimistic estimate of the time required for an eye to evolve *Proc. Biol. Sci.* **256** 53–8

[2] François P, Hakim V and Siggia E D 2007 Deriving structure from evolution: metazoan segmentation *Mol. Syst. Biol.* **3** 154

[3] François P and Hakim V 2004 Design of genetic networks with specified functions by evolution in silico *Proc. Natl Acad. Sci. USA* **101** 580–5

[4] Soyer O S, Pfeiffer T and Bonhoeffer S 2006 Simulating the evolution of signal transduction pathways *J. Theor. Biol.* **241** 223–32

[5] Matthews H R, Murphy R L, Fain G L and Lamb T D 1988 Photoreceptor light adaptation is mediated by cytoplasmic calcium concentration *Nature* **334** 67–9

[6] Nakatani K and Yau K W 1988 Calcium and light adaptation in retinal rods and cones *Nature* **334** 69–71

[7] Marshall C J 1995 Specificity of receptor tyrosine kinase signaling: transient versus sustained extracellular signal-regulated kinase activation *Cell* **80** 179–85

[8] Kholodenko B N 2006 Cell-signalling dynamics in time and space *Nat. Rev. Mol. Cell Biol.* **7** 165–76

[9] Block S M, Segall J E and Berg H C 1983 Adaptation kinetics in bacterial chemotaxis *J. Bacteriol.* **154** 312–23

[10] Behar M, Hao N, Dohlman H G and Elston T C 2007 Mathematical and computational analysis of adaptation via feedback inhibition in signal transduction pathways *Biophys. J.* **93** 806–21

[11] Lewis J 2003 Autoinhibition with transcriptional delay: a simple mechanism for the zebrafish somitogenesis oscillator *Curr. Biol.* **13** 1398–408

[12] Monk N A M 2003 Oscillatory expression of hes1, p53, and nf-kappab driven by transcriptional time delays *Curr. Biol.* **13** 1409–13

[13] Lenski R E, Ofria C, Collier T C and Adami C 1999 Genome complexity, robustness and genetic interactions in digital organisms *Nature* **400** 661–4

[14] Adami C 2006 Digital genetics: unravelling the genetic basis of evolution *Nat. Rev. Genet.* **7** 109–18

[15] Neyfakh A A, Baranova N N and Mizrokhi L J 2006 A system for studying evolution of life-like virtual organisms *Biol. Direct.* **1** 23

[16] Bergman A and Siegal M L 2003 Evolutionary capacitance as a general feature of complex gene networks *Nature* **424** 549–52

[17] Wagner A 2005 *Robustness and Evolvability in Living Systems* (Princeton Studies in Complexity) (Princeton, NJ: Princeton University Press)

[18] Ferguson S S and Caron M G 1998 G protein-coupled receptor adaptation mechanisms *Semin. Cell Dev. Biol.* **9** 119–27

[19] Premont R T and Gainetdinov R R 2007 Physiological roles of g protein-coupled receptor kinases and arrestins *Annu. Rev. Physiol.* **69** 511–34

[20] Violin J D, DiPilato L M, Yildirim N, Elston T C, Zhang J and Lefkowitz R J 2008 Beta2-adrenergic receptor signaling and desensitization elucidated by quantitative modeling of real time camp dynamics *J. Biol. Chem.* **283** 2949–61

[21] Barkai N and Leibler S 1997 Robustness in simple biochemical networks *Nature* **387** 913–7

[22] Yi T M, Huang Y, Simon M I and Doyle J 2000 Robust perfect adaptation in bacterial chemotaxis through integral feedback control *Proc. Natl Acad. Sci. USA* **97** 4649–53

[23] Sourjik V and Berg H C 2002 Receptor sensitivity in bacterial chemotaxis *Proc. Natl Acad. Sci. USA* **99** 123–7

[24] Rao C V, Kirby J R and Arkin A P 2004 Design and diversity in bacterial chemotaxis: a comparative study in escherichia coli and bacillus subtilis *PLoS Biol.* **2** E49

[25] Emonet T and Cluzel P 2008 Relationship between cellular response and behavioral variability in bacterial chemotaxis *Proc. Natl Acad. Sci. USA* **105** 3304–9

[26] Bray D, Levin M D and Morton-Firth C J 1998 Receptor clustering as a cellular mechanism to control sensitivity *Nature* **393** 85–8

[27] Levchenko A and Iglesias P A 2002 Models of eukaryotic gradient sensing: application to chemotaxis of amoebae and neutrophils *Biophys. J.* **82** 50–63

[28] Yang L and Iglesias P A 2006 Positive feedback may cause the biphasic response observed in the chemoattractant-induced response of dictyostelium cells *Syst. Control Lett.* **55** 329–37

[29] Ziv E, Nemenman I and Wiggins C H 2007 Optimal signal processing in small stochastic biochemical networks *PLoS ONE* **2** e1077