Journal of Theoretical Biology 258 (2009) 502–512

Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

Reliability of regulatory networks and its evolution

Stefan Braunewell, Stefan Bornholdt *

Institute for Theoretical Physics, University of Bremen, D-28359 Bremen, Germany

ARTICLE INFO

Article history:
Received 27 May 2008
Received in revised form
18 February 2009
Accepted 20 February 2009
Available online 28 February 2009

Keywords:
Gene regulatory networks
Evolution models
Boolean networks
Computer simulation
Robustness

ABSTRACT

The problem of reliability of the dynamics in biological regulatory networks is studied in the framework of a generalized Boolean network model with continuous timing and noise. Using well-known artificial genetic networks such as the repressilator, we discuss concepts of reliability of rhythmic attractors. In a simple evolution process we investigate how overall network structure affects the reliability of the dynamics. In the course of the evolution, networks are selected for reliable dynamics. We find that most networks can be easily evolved towards reliable functioning while preserving the original function.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Biological systems are composed of molecular components and the interactions between these components are of an intrinsically stochastic nature. At the same time, living cells perform their tasks reliably, which leads to the question how reliability of a regulatory system can be ensured despite the omnipresent molecular fluctuations in its biochemical interactions.

Previously, this question has been investigated mainly on the single gene or molecule species level. In particular, different mechanisms of noise attenuation and control have been explored, such as the relation of gene activity changes, transcription and translation efficiency or gene redundancy (Ozbudak et al., 2002; Raser and O'Shea, 2005; McAdams and Arkin, 1999). Apart from these mechanisms acting on the level of the individual biochemical reactions, also features of the circuitry of the reaction networks can be identified which aid robust functioning (Barkai and Leibler, 1997; Alon et al., 1999; von Dassow et al., 2000). A prime example of such a qualitative feature that leads to an increased stability of a gene's expression level despite fluctuations of the reactants is negative autoregulation (Becskei and Serrano, 2000). At higher levels of organization, the specific linking pattern of the larger biochemical regulatory networks can further contribute to the overall robustness. In comparative computational studies of several different organisms, it has been shown that among those topologies that produce the desired functional behavior only a small number also displays high robustness against parameter variations. Indeed, the experimentally observed networks rank high among these robust topologies (Kollmann et al., 2005; Wagner, 2005a; Ma et al., 2006).

However, most current models are based on the deterministic dynamics of differential equations. Modeling of the intrinsic noise associated with the various processes in the network requires an inherently stochastic modeling framework, such as stochastic differential equations or a master equation approach (Thattai and van Oudenaarden, 2001; Kepler and Elston, 2001; Ozbudak et al., 2002; Rao et al., 2002). These complex modeling schemes need a large number of parameters such as binding constants and reaction rates and can only be conducted for well-known systems or simple engineered circuits. For generic investigations of such systems, coarse-grained modeling schemes have been devised that focus on network features instead of the specifics of the reactions involved (Bornholdt, 2005).

To incorporate the effects of molecular fluctuations into discrete models, a commonly used approach is to allow random flips of the node states. Several biological networks have been investigated in this framework and a robust functioning of the core topologies has been identified (Albert and Othmer, 2003; Li et al., 2004; Davidich and Bornholdt, 2008). However, for biological systems, the perturbation by node state flips appears to be an unrealistic type of noise: in real organisms, concentrations and timings fluctuate, while the qualitative state of a gene is often quite stable. A more realistic form of fluctuations than macroscopic (state flip) noise should allow for microscopic fluctuations. This can be implemented in terms of fluctuating timing of switching events (Klemm and Bornholdt, 2005b; Chaves et al., 2005; Braunewell and Bornholdt, 2007). The principle idea is to allow for fluctuations of event times and test whether the

* Corresponding author. Tel.: +49 421 218 8198; fax: +49 421 218 9104.
E-mail address: bornholdt@itp.uni-bremen.de (S. Bornholdt).

0022-5193/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2009.02.013

dynamical behavior of a given network stays time ordered despite these fluctuations.

In this work we want to focus on the reliability criterion that has been used to show the robustness of the yeast cell-cycle dynamics against timing perturbations (Braunewell and Bornholdt, 2007) and investigate the interplay of topological structure and dynamical robustness. Using small genetic circuits we explore the concept of reliability and discuss design principles of reliable networks.

However, biological networks have not been engineered with these principles in mind, but instead have emerged from evolutionary procedures. We want to investigate whether an evolutionary procedure can account for reliability of network dynamics. A number of studies has focused on the question of evolution towards robustness (Wagner, 1996; Bornholdt and Sneppen, 2000; Ciliberti et al., 2007; Szejka and Drossel, 2007; Aldana et al., 2007). However, the evolution of reliability against timing fluctuations has not been investigated. First indications that network architecture can be evolved to display reliable dynamics despite fluctuating transmission times has been obtained in a first study in Braunewell and Bornholdt (2008). Using a deterministic criterion for reliable functioning, introduced in Klemm and Bornholdt (2005a), it was found that small networks can be rapidly evolved towards fully reliable attractor landscapes. Also, if a given (unreliable) attractor is chosen as the "correct" system behavior, it was shown that with a high probability a simple network evolution is able to find a network that reproduces this attractor reliably, i.e. in the presence of noise.

Here, we use a more biologically plausible definition of timing noise to investigate whether a network evolution procedure can generate robust networks. We focus on the question whether a predefined network behavior can be implemented in a reliable way, just utilizing mutations of the network structure. We use a simple dynamical rule to obtain the genes' activity states, such that the dynamical behavior of the system is completely determined by the wiring of the network.

2. Model description

2.1. Boolean dynamics

A standard approach to computer simulations of molecular biological systems starts from chemical master equations and their explicit stochastic modeling, e.g. via Monte Carlo algorithms (Gillespie, 1977). However, such methods need a large number of parameters and detailed knowledge about the system in order to completely describe the system dynamics. As an alternative, for gaining first, qualitative insights into the dynamics of genetic regulatory systems it has proven useful to apply strongly coarse-grained models (Bornholdt, 2005).

Boolean networks, first introduced by Kauffman (1969) as anecdotal models of gene regulation based on random networks, have emerged as a successful tool for qualitative dynamical modeling and have been successfully employed in models of regulatory circuits in various organisms such as *Drosophila melanogaster* (Albert and Othmer, 2003), *Saccharomyces cerevisiae* (Li et al., 2004), *Arabidopsis thaliana* (Espinosa-Soto et al., 2004), and *Schizosaccharomyces pombe* (Davidich and Bornholdt, 2008). In this class of dynamical models, genes, proteins, and mRNA are modeled as discrete switches which assume one of only two possible states. Here, the active state represents a gene being transcribed or molecular concentrations (of mRNA or proteins) above a certain threshold level. Thus, at this level, a regulatory network is modeled as a simple network of switches.

Time is modeled in discrete steps and the state of all nodes is updated at the same time depending only on the state of all nodes at the previous time step according to the wiring of the network and the given Boolean function at each node.

When such a system is initialized with some given set of node states, it will in general follow a series of state changes until it reaches a configuration that has been visited before (finite number of states). Because of the deterministic nature of the dynamics, the system has then entered a limit cycle and repeats the same sequence of states indefinitely (or keeps the same state, then called a fixed point attractor).

2.2. Stochastic dynamics

In the original Boolean model there are two assumptions that are clearly non-biological and are thus often criticized: (1) The synchronized iteration of the Boolean network in discrete time steps implies total synchrony of all components. (2) The binary (ON/OFF) node states which prohibit intermediate levels and gradual effects.

There have been various attempts at loosening these assumptions while keeping the simplicity of the Boolean models. It is a clear advantage of Boolean models that they operate on a finite state space. The synchronous timing, however, does not hold a similar advantage apart from computational simplicity. Models that overcome this synchronous updating scheme have been suggested in a variety of forms. In Chaves et al. (2006) different asynchronous schemes are used in the model of the fruit fly. The simplest asynchronous model keeps the discrete notion of time but lets events happen sequentially instead of simultaneously. A continuous-time generalization of Boolean models that is inspired by differential equation models has been suggested in Klemm and Bornholdt (2005b). Here, the discreteness of the node states is kept but the dynamics take place in a continuous time. In Klemm and Bornholdt (2005a) and Braunewell and Bornholdt (2008) the limit of infinitesimally small disturbances from synchronous behavior is investigated.

This concept of allowing variations from the synchronous behavior will also be used in this work. The principle idea is to use a continuous time description and identify the state of the nodes at certain times with the discrete time steps of the synchronous description (Glass, 1975). Further, an internal continuous variable is introduced for every node and the binary value of the node is obtained from this continuous variable using a threshold function. Now a differential equation can be formulated for the continuous variable.

This is pictured in Fig. 1. Here the internal dynamics and the resulting activity state of a node with just one input are shown for a given input pattern. The activator A of the node B is switched on (through a signal from another node, for example) at time $t = 1$ and stays on until it is switched off at time $t = 2$. In the Boolean description we would say node A assumes state $S_A = 1$ at time step 1 and at time step 2 switches to state $S_A = 0$. Node B would react by switching to state $S_B = 1$ at step 2 and to $S_B = 0$ at step 3. In the continuous version, we implement this by a delay time and a "charging" behavior of the concentration value of node B, driven by the input variable S_A. As soon as c_B crosses the threshold of $\frac{1}{2}$, the activity state of B switches to $S_B = 1$.

Let us formulate the time evolution of a system of such model genes by the set of delay differential equations as

$$\tau \frac{dc_i(t)}{dt} = f_i(t, t_d) - c_i(t). \quad (1)$$

Here, $f_i(t, t_d)$ denotes the transmission function of node i and describes the effect of all inputs of node i at the current time. The parameter τ sets the time scale of the production or decay process.

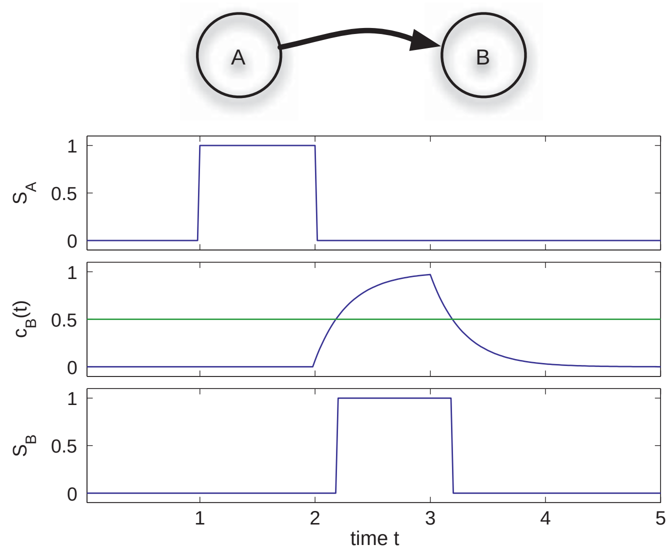

Fig. 1. Concentration buildup and decay of a protein given a specific input signal S_A and the corresponding activity state S_B ($t_d = 1, \tau = 0.3$).

In general, any Boolean functions can be used as transmission function f_i. For simplicity, we choose threshold functions, which have proven useful for the modeling of real regulatory networks (Li et al., 2004; Davidich and Bornholdt, 2008).

Let us use the following transmission function:

$$f_i(t, t_d) = \begin{cases} 1, & \sum_j a_{ij} S_j(t - t_d) \geqslant T_i, \\ 0, & \sum_j a_{ij} S_j(t - t_d) < T_i, \end{cases} \tag{2}$$

where t_d is the transmission delay time that comprises the time taken by processes such as translation or diffusion that cause the concentration buildup of one protein to not immediately affect other proteins. The interaction weight a_{ij} determines the effect that protein j has on protein i. An activating interaction is described by $a_{ij} = 1$, inhibition by $a_{ij} = -1$. If the presence of protein j does not affect expression of protein i, $a_{ij} = 0$. The discrete state variable S_i is determined by the continuous concentration variable c_i via a Heaviside function $S_i(t) = \Theta[c(t) - \frac{1}{2}]$. The threshold value T_i is given by $T_i = \sum_j a_{ij}/2$ (this choice is equivalent to the commonly used threshold value of 0 if the activity states are given by $S_i = \pm 1$ instead of the Boolean values used here).

For the simple transmission function given above, Eq. (1) can be easily solved piecewise (for every time span of constant transmission function), leading to the following buildup or decay behavior of the concentration levels:

$$c_i(t > t_0) = \begin{cases} 1 - (1 - c(t_0)) \exp(-(t - t_0)/\tau), & f_i \geqslant 0, \\ c(t_0) \exp(-(t - t_0)/\tau), & f_i < 0. \end{cases} \tag{3}$$

This has the effect of a low-pass filter, i.e. a signal has to sustain for a while in order to be able to affect the discrete activity state. A signal spike, on the other hand, will be filtered out.

Up to now we have only introduced a continuous, but still deterministic generalization of the synchronous Boolean network model. Let us now allow noise on the timing delay, such that the model becomes stochastic and asynchronous. We implement this stochastic timing into the model by means of a signaling mechanism. As soon as one node flips its discrete state at time $t = t_0$, it sends a signal to each node it regulates. This signal affects the input of a regulated node at a later time $t = t_0 + t_d + \chi$ where χ is a uniformly distributed random number between 0 and $\chi_{\max}$. The random number χ is chosen for each signal and each link independently, which means that a switching node will affect two regulated nodes at slightly different times.

Due to the timing perturbations, the network states at exact integer times do not hold a particular significance any more. To overcome this problem, we define a new macro time step by requiring that all discrete node states (not the concentration levels) are constant for a time interval of at least $t_d/2 + \tau$. Each such macro time step is the equivalent of one discrete time step in the synchronous model. Only the system states at these macro time steps of extended rest are used in the comparison with the synchronous behavior.

This way, small fluctuations of the signaling and switching events are tolerated, but extended times of inactivity of the system must exist and the state of the network at these times must correspond to the respective states under synchronous dynamics. We call network dynamics “reliable” if, despite the stochastic effects on the signal transmission times, the network follows the same state sequence as in the synchronous version of the model. Although fluctuations in the exact timing are omnipresent, ordered behavior of the sequence of states can still be achieved. An exact definition of the algorithm can be found in Appendix A.

In Klemm and Bornholdt (2005a) a similar model, but with infinitesimal timing perturbations, was used to identify those attractors in random Boolean networks that are reliable. In that study it was found that most attractors in fact are unreliable, but are irrelevant for the system because of very small basins of attraction. Further, it was shown that the number of reliable attractors scales sub-linearly with system size, which reconciles the scaling of random Boolean networks with the numbers of observed cell types as was originally proposed in Kauffman (1993). A similar result was also obtained in a sequential updating scheme (Greil and Drossel, 2005).

3. Reliable and unreliable network dynamics

3.1. Dynamical sequence does not uniquely determine reliability

In order to illustrate the differences between reliable and unreliable attractors let us discuss an example of two small circuits that exhibit identical dynamics under synchronous dynamics, while they differ in dynamics once noise is added (resulting in asynchronous dynamics of the nodes). The first circuit is called reliable in our definition, while the second is called unreliable.[1]

Let us consider the well-known example of two mutually activating genes and model the system according to Eqs. (1)–(3). Apart from the trivial fixed points (both on or both off) this system displays an unreliable attractor as shown in the left panel of Fig. 2. In the upper part, the synchronous Boolean attractor is depicted in a simple pictorial form (black means active, white inactive). Below that the continuous variable of both nodes is plotted over time in an example run and it can be seen that because of desynchronization the system can exit the synchronous state sequence.

Changing just one link and thus creating an inhibiting self-interaction at the first gene (see right panel of Fig. 2), the dynamics is now driven by this one node loop. The synchronous sequence of the attractor is still the same, but now the fixed points of the old network are no longer fixed points but transient states

[1] The parameters for the time delay, t_d, the production time constant, τ and the noise level $\chi_{\max}$ are chosen for optimal readability of the figures. We stress that all our conclusions also hold for the parameter choice from the results part or for any variation of these values within reasonable bounds.

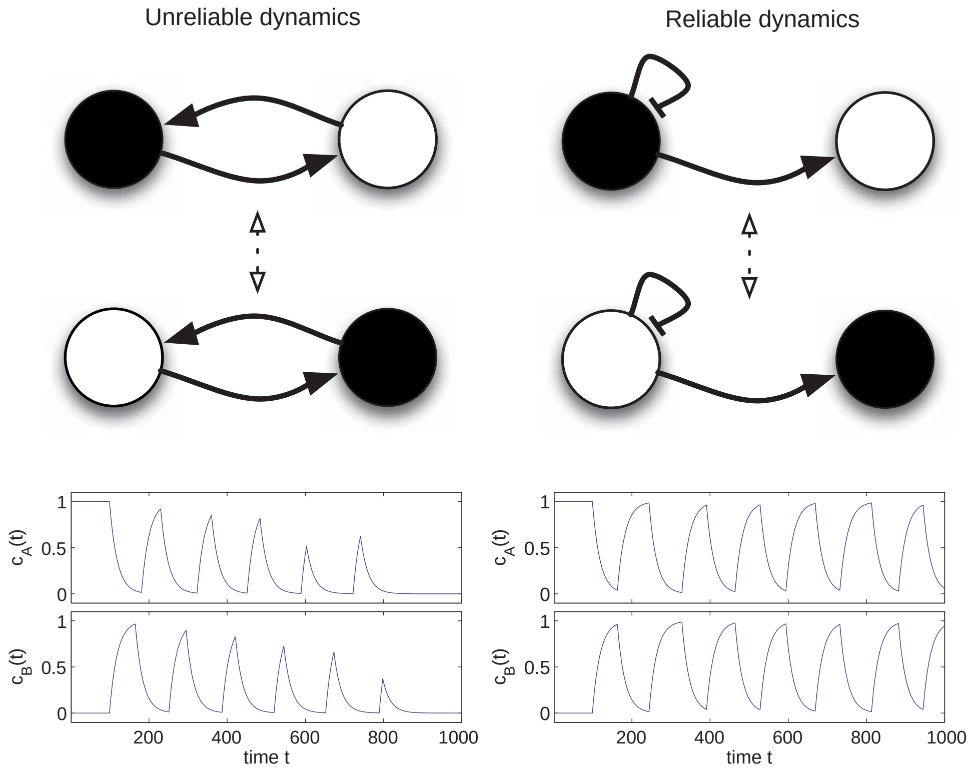

Fig. 2. Comparison of two networks that have a common synchronous attractor. The mutually activating network is unreliable when subject to noise on the signal delay times. In contrast, a negatively autoregulated gene that activates the second exhibits reliable dynamics.

to this attractor. The asynchronous dynamics, as shown in the lower part of Fig. 2 now display an ordered behavior that would continue indefinitely.

Note that an essential feature that causes these stable oscillations is the time delay involved. Without a time delay, the system would not exhibit stable oscillations in either case but would assume intermediate levels for both nodes. Thus, a direct comparison of these dynamics with a stability analysis of ordinary differential equations without delay is not adequate.

Next, let us test the reliability of examples of circuits that can be created artificially within living cells. The so-called repressilator is a simple artificially generated genetic circuit implemented in *Escherichia coli* (Elowitz and Leibler, 2000). Consisting of three genes inhibiting each other in a ring topology (see upper part of Fig. 3), this system displays stable oscillations.

Describing this system using differential equations it has been found that the unique steady state is unstable for certain parameter values and that numerical integration of the differential equations displays oscillatory behavior. Also in a stochastic modeling scheme, sustained but irregular oscillations can be observed, which show some resemblance of the experimental time series (Elowitz and Leibler, 2000).

To discuss this model system in our framework, the synchronous Boolean description has to be analyzed first. Here, the three-gene repressilator exhibits two attractors which comprise all eight network states—the "all-active–all-inactive" (two states) and the "signal-is-running-around" pattern (six states). In the asynchronous scheme, independent of the initial conditions, the system reaches the second attractor. Once the attractor is reached, the system stays in it forever (i.e. is reliable in our definition)—see Fig. 3. This is due to the fact that only a single switching happens at a given time. This is depicted in the lower part of Fig. 3 by the arrows which are successively active, no two events happening at the same time.

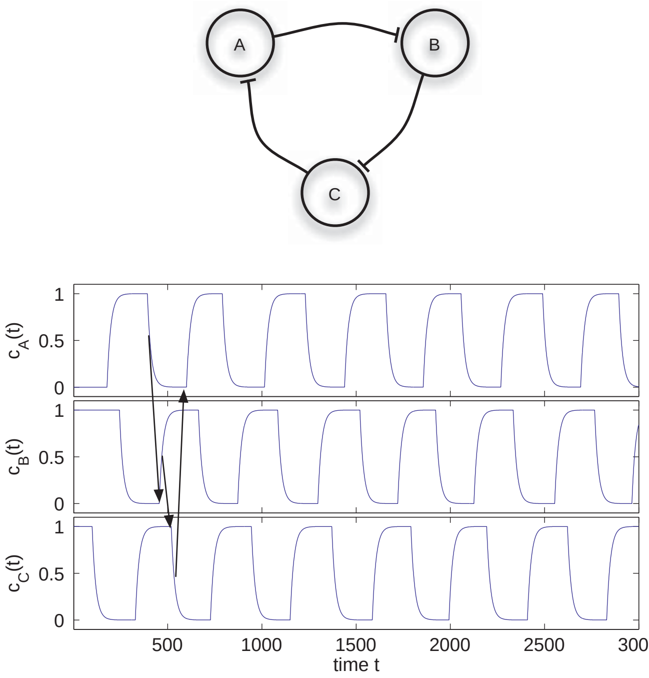

Fig. 3. Wiring diagram (top) and example time evolution of concentrations of all three internal variables of the three-gene repressilator. The dynamics is governed by a single event running around the circle—here depicted by arrows which denote the flow of signals.

This picture changes in the case of a hypothetical four-genes repressilator. The attractor structure is now much more involved in the synchronously updated model. It consists of the two fixed points (1 0 1 0) and (0 1 0 1) and three attractors with four states each. Using the stochastic Boolean model as before, we find that only the fixed points emerge as reliable attractors of the system. If any state of one of the four cycles is prepared as initial condition, the system thus always ends up in one of the two fixed points. In Fig. 4, an example run is shown which is initialized with the state (1 1 0 0).

Without any noise, the system would follow a four-state sequence consisting of all states where the two active nodes are adjacent and the two inactive nodes are as well. However, if a small perturbation is allowed, the system can exit this attractor as shown in the lower panel of Fig. 4. Here, we have drawn two arrows showing two causal events happening at the same time. In fact, there are two independent causal chains in the system dynamics. If these two chains fluctuate in phase relative to each other, they can extinguish each other and drive the system into a fixed point (Klemm and Bornholdt, 2005b).

This dramatic difference in attractors between the traditional (noiseless and synchronously updated) Boolean network and our generalized Boolean network model (with autonomously updated nodes plus timing noise) is not just limited to such simple toy models. Our interpretation w.r.t. biological systems is that attractors that are not robust against noise in the sense demonstrated here cannot occur in biological regulatory networks where noise is omnipresent. This view is supported by our earlier finding that the dynamics of the budding yeast cell cycle network is found to be a reliable attractor of the corresponding network circuitry when simulated in the presence of noise (Braunewell and Bornholdt, 2007).

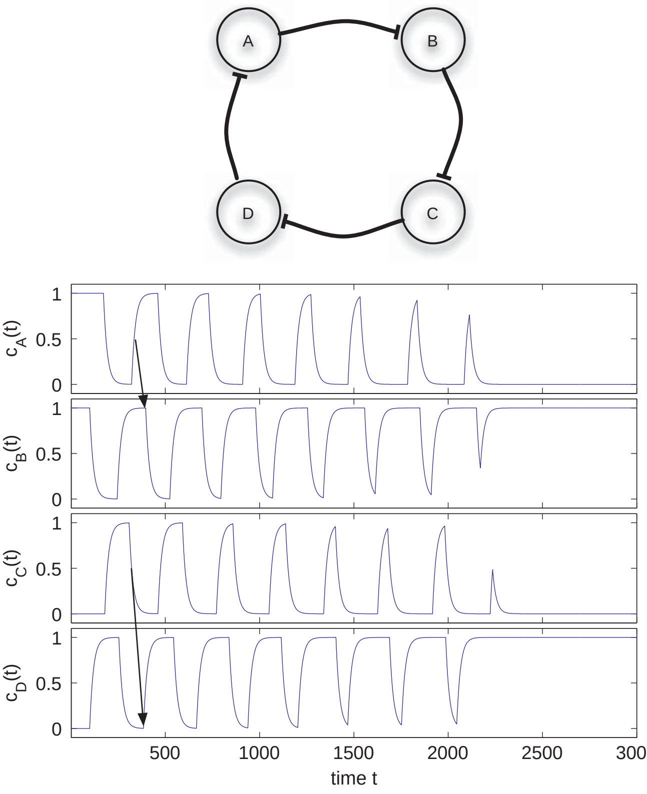

Fig. 4. Wiring diagram (top) and example time evolution of concentrations of all four internal variables of the four-gene repressilator. As two events happen independently at the same time (shown by the arrows depicting the signal events), the attractor can be left when the timing of the two event chains desynchronizes.

3.2. Stable and unstable dynamics

Apart from a system accumulating a phase lag through small perturbations in a random walk-like fashion and eventually ending up in a different attractor, there are also examples of systems in which any small perturbation drives the system away from the current attractor.

We show this behavior in Fig. 5. Here again, the four node repressilator is shown, but with the initial state configuration 0 0 0 0. Without noise, this state belongs to the “all-active–all-inactive” attractor and four independent events are happening at each time step. The small stochastic asynchrony in the beginning is amplified and leads to a quick loss of the attractor. The system then enters an intermediate attractor where the neutral perturbation behavior is predominant, because the concentration levels have more time to approach their saturation value.

The opposite behavior is also possible, that a system itself prevents divergence of the phases. This can happen if the intermediate system state creates a signal spike (i.e. a short-term status change of a node) that itself feeds back to the causal chain. Even though the causal chains are independent in synchronous mode, they can be connected through such intermediate states.

We want to stress that in the criterion employed in Braunewell and Bornholdt (2008) it cannot be identified whether an attractor is marginally stable or exhibits such a self-catching behavior. This is a limitation of the deterministic criterion that is overcome by the explicit modeling used in the following.

In this work, we consider all attractors as “unreliable” that can desynchronize so strongly that the system does not maintain a “rest phase” in which no switching events occur for an extended period of time. This includes all marginally stable as well as all unstable attractors. We do not distinguish between these in our results as both do not seem suitable for the reliability of a biological system.

4. Network evolution, simulation details

Now that we have introduced the main concepts and ideas surrounding our definition of reliability, we want to turn to the question, whether such a simple model of regulatory networks can be evolved towards realizations displaying reliable dynamics. For this question, we define the notion of a “functional attractor”. As we are dealing with random networks, we need a measure of what the system is supposed to do. Thus, we choose one attractor of the starting network as the prototype dynamics that define the desirable dynamical sequence. The functional attractor is determined by running the synchronous model with a randomly chosen initial state until an attractor is found. During the subsequent evolution process we demand each network to reproduce this attractor.

This prescription introduces a bias towards attractors with large basins of attraction. However, as the basin of attraction is commonly understood as a measure of the significance of an attractor, this appears to be a natural choice. Only unreliable attractors are used as functional attractors, because when using a reliable attractor as target, the evolution goal would be achieved before the start of the evolution.

The evolution procedure is chosen as a simple algorithm consisting of a mutation and a selection step. We start by creating a directed random (Erdös–Renyi) network with a given number of links M (self-links are allowed) and determine the functional attractor. Mutation of the current network is defined as a single rewiring of a link, that is, removal of one link and simultaneous addition of a random link between two nodes that are not yet

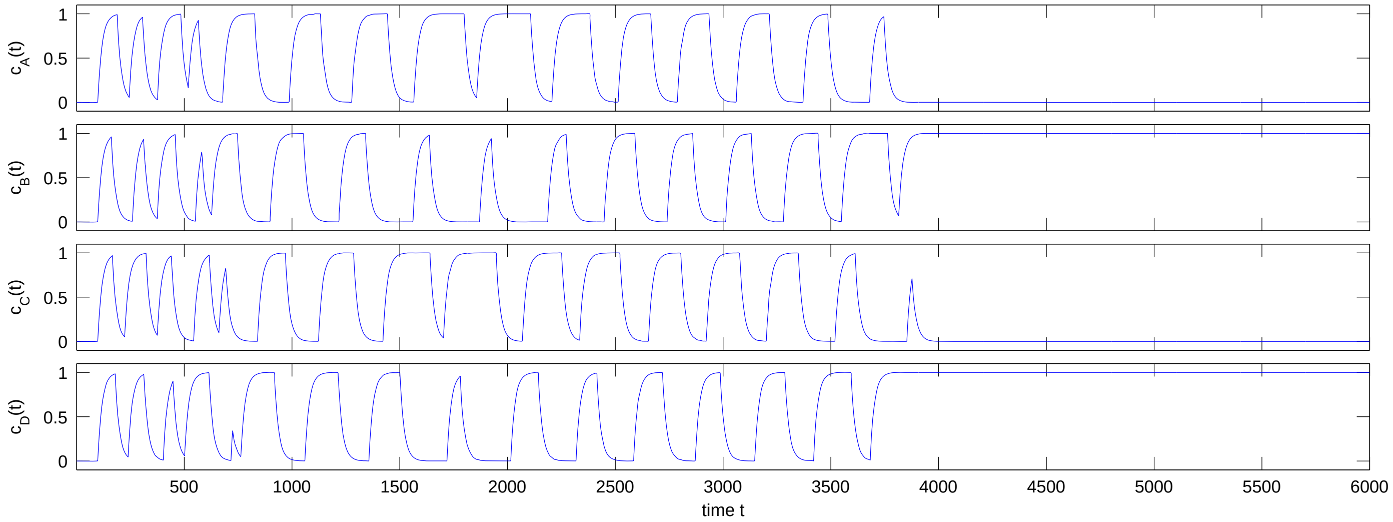

Fig. 5. Example of an unstable, a marginally stable and a fixed point attractor in the four-gene repressilator.

connected. These two simultaneous operations on the graph ensure that the connectivity $\langle k\rangle = M/N$ is kept fixed.

The fitness of a given network is assessed by comparing the attractor obtained under asynchronous dynamics with the functional attractor (obtained by synchronous update). The initial network state is set to one randomly chosen state of the synchronous attractor. The concentration levels are initialized to the same value (either 0.0 or 1.0). Now the system dynamics is explicitly simulated using the stochastic algorithm introduced above (details given in Appendix A). We follow the dynamics for a maximum of 10^6 macro time steps (i.e. number of rest phases). The fitness score is then obtained by dividing the number of steps with identical network states by the maximum step number.

The algorithm used in Braunewell and Bornholdt (2008) is an abstraction of the principle that a small fluctuation on its own cannot drive the system out of its attractor, but only if successively adding up. Thus, in a systematic way, for any possible retardation of signal events it is checked whether it persists in the system for a full progression around the attractor. If so, the attractor is marginally stable and can in principle lose its synchrony. While this has the advantage of being a deterministic criterion and thus leads to a noiseless fitness function, there are situations in which this criterion is sufficient but not necessary for reliable behavior. By explicitly modeling the time course, as done in this work, only the truly unstable attractors are marked as such.

We do not take into account the transient behavior of the system and define only the limit cycle as the functional attractor. As our reliability definition would be trivially fulfilled in the case of fixed points, we always start with a network exhibiting a limit cycle attractor.

In the selection step, the fitness of a mutant network is compared to the fitness of the mother network. A network is selected only if it scores higher than any other network found before during the evolution. As the dynamics is inherently stochastic, the fitness criterion is noisy, too. Thus, networks which are not more reliable than the mother network might still be selected in the evolution due to variability in the fitness score.

If a network follows the attractor up to a maximum step number, it is said to be "reliable". If during the network process a given number of mutation tries is exceeded, the evolution process is aborted. Here, the maximum number of mutation tries in the evolution is $a_{\text{step}} = 20\,000$ at each step and $a_{\text{tot}} = 10^6$ during the full course of evolution. We later discuss the implications of these fixed parameter settings. We have used the following further parameters in the results part. The delay time t_d is set to unity, the buildup time τ is 0.1. Maximal noise $\chi_{\max}$ is 0.02. This means, that the impact of any individual perturbation is low and cannot itself cause a failure in the fitness test. Only if several perturbations consecutively drive the system away from synchronization, the requirement of an extended static period can be missed.

In Figs. 6 and 7 we show an example of a typical evolution process for a small network of $N = 12$ nodes. During three steps, the network is evolved towards a reliable architecture. The initial network (upper-left in Fig. 6) displays three synchronous attractors (top panel in Fig. 7) of which the first is chosen as the functional attractor. The structural changes are depicted in Fig. 6 by a gray arrow for the removed link and a plus sign for the newly added link. As is typical for these evolution processes (Braunewell and Bornholdt, 2008), the attractor landscape is affected dramatically during the evolution. In this example, only the functional attractor survives the evolution procedure.

5. Results of the network evolution

We have performed the described network evolution for a variety of different network sizes as well as connectivities. For system sizes of $N = 16$, 32, and 50 and connectivities between 0.5 and 6 the ratio of networks that were stabilized by evolution is shown in Fig. 8. Whenever we plot the ratio of stabilized networks, we have calculated the sample errors by a Poissonian error estimate, $\Delta x = \sqrt{x(1-x)/n}$, where x is the obtained ratio from n sample runs.

One can see that for intermediate connectivities between 2.5 and 4.5, the ratio of stabilized networks is above 80% for all system sizes under investigation. This means, that starting from any random network, in four out of five cases a simple network evolution is able to find a network that displays the same dynamical attractor, but performs it reliably. This result matches a very similar dependence on the connectivity that was found for 16 nodes in the infinitesimal scheme in Braunewell and Bornholdt (2008).

It is interesting to note that for lower connectivities, the ratio of stabilized networks decreases significantly. For all system sizes considered, there is a sizable decrease in the stabilization ratio for connectivities below 2. This is especially apparent for the large system size $N = 50$. This is due to the "essentiality" of the structure on the dynamics. Changing a link without destroying the

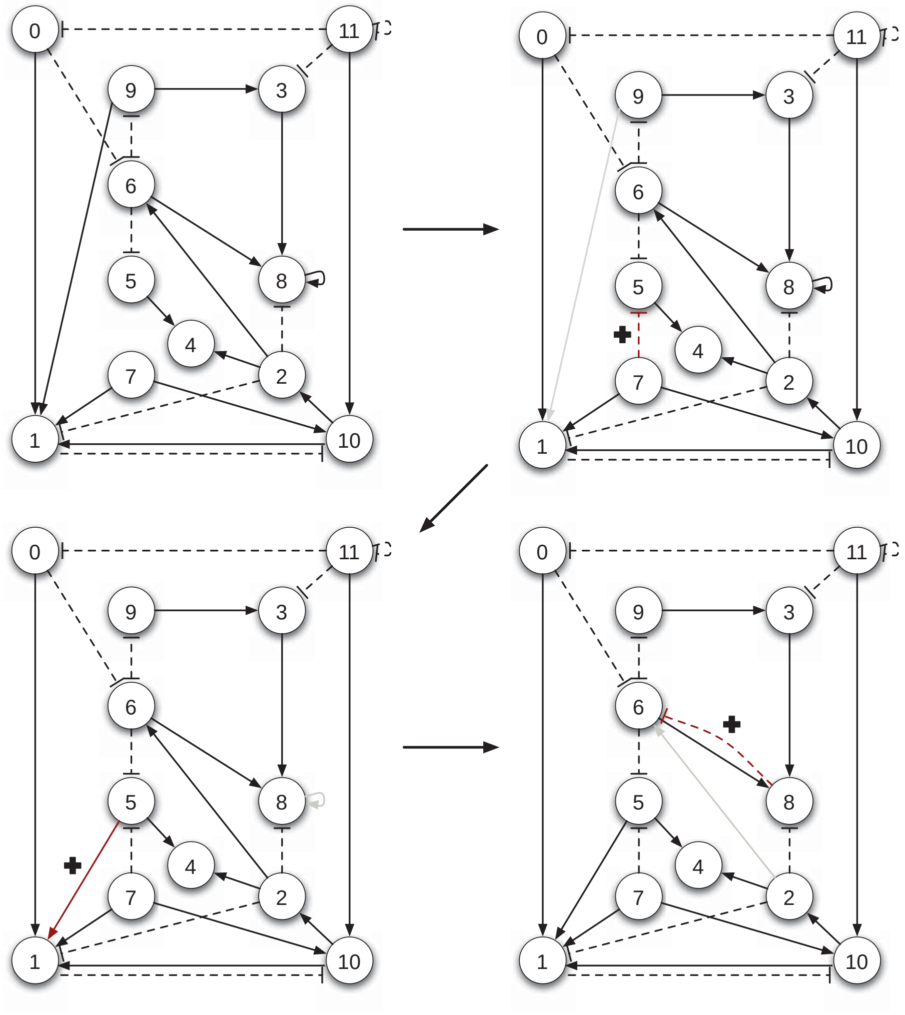

Fig. 6. A typical example of an evolution process for a network of size $N = 12$. In this example, three steps suffice for stabilization. The structure of each network during the evolution is shown, with the arrows denoting the subsequent step in the evolution. In every step, one link is lost (shown in gray color) and a new link is added (denoted by the plus sign). The change of the state space of the network is given in Fig. 7.

dynamical attractor is less likely for lower connectivities. At higher connectivities the larger number of non-essential links in the system aids evolvability towards reliable dynamics via phenotypically neutral mutations.

However, considering large connectivities and large system sizes, the ratio of stabilized networks drops again, along with the increase in attractor lengths with system size that impairs reproducibility of dynamics. Thus, we find an area of connectivity between 1.5 and 4.5 for which the ratio of stabilized networks is similar for all system sizes considered.

The plot in Fig. 9 shows the average number of rewiring steps necessary until a stable network realization is found for networks of 32 nodes. For all connectivities, this number is remarkably low, as the evolution procedure basically implements a biased random walk through structure space. This is due to the large variation of the fitness score of a single network. Despite the rather small evolutionary pressure, the evolution procedure quickly finds a realization exhibiting reliable dynamics. Interestingly, the number of evolution steps does not monotonically grow with the connectivity, but instead drops for connectivities larger than 2.

This again is an indication that networks with higher connectivities are easier to evolve towards reliability. The ratio of links rewired in the evolution to the total number of links is even monotonically decreasing (not shown).

We want to further investigate the dependence on network size by repeating the evolution procedure with system sizes up to $N = 400$. This is shown in Fig. 10 in a log-linear plot of the ratio of stabilized networks vs. system size. We find that the ability of the process to stabilize a given network decreases with system size. The line in the figure represents a fit of the function $f(N) = a - b\log(x)$ with $a = 1.416$, $b = 0.198$ thus a relatively slow decay with system size. One also has to keep in mind that the fixed set of parameters for the number of attempted mutations per evolution step and the total number of attempted mutations during the evolution reduces the success rate for larger networks. For small networks of $N = 16$, 20 000 attempted mutations per evolution step suffices for a good estimate of the space of all one-link mutations, but as the number of possible mutations scales with the system size N as N^3, it quickly becomes impossible to check all possibilities. Thus, the results in Fig. 10 underestimate the probability to find a stable instance.

We have checked the dependence of the results on the selection parameters (attempted mutations per evolution step a_{step} and total number of attempted mutations during evolution a_{total}) for selected network sizes and connectivities. In Fig. 11 we again show the ratio of stabilized networks vs. system size, this time for two different parameter values—the original parameter set with $a_{\text{step}} = 2 \times 10^4$ (denoted by "+") and for an increased value of $a_{\text{step}} = 10^5$ (denoted by "×"). For small networks, the

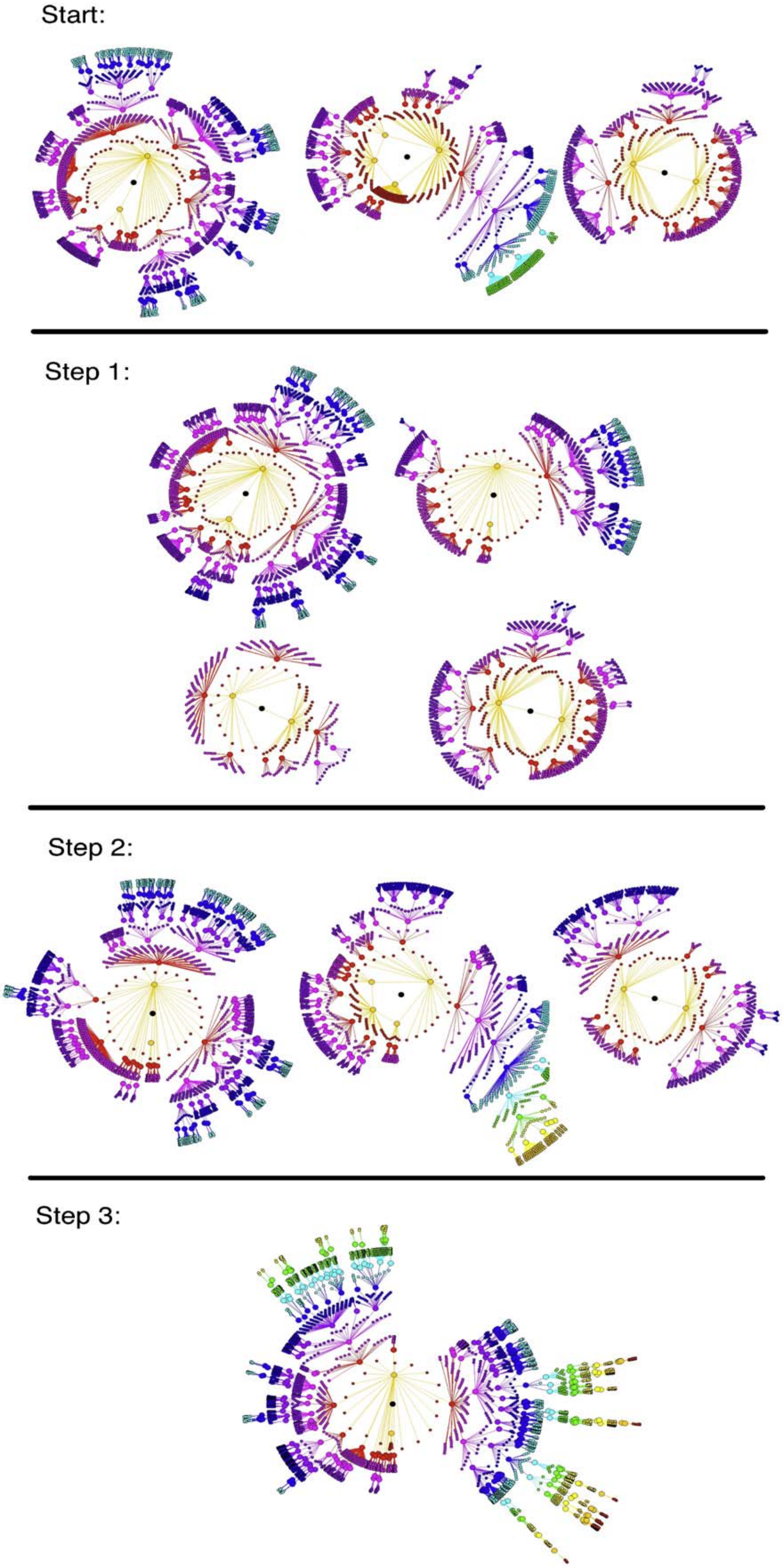

Fig. 7. Change of the (synchronous) attractor landscape during evolution—corresponding to the network structures shown in Fig. 6. For every step, the full attractor landscape is shown. Every dot denotes a state, the subsequent state is connected via a line. The limit cycle is shown in the center of each attractor basin. The functional attractor is shown as the upper leftmost attractor in all steps.

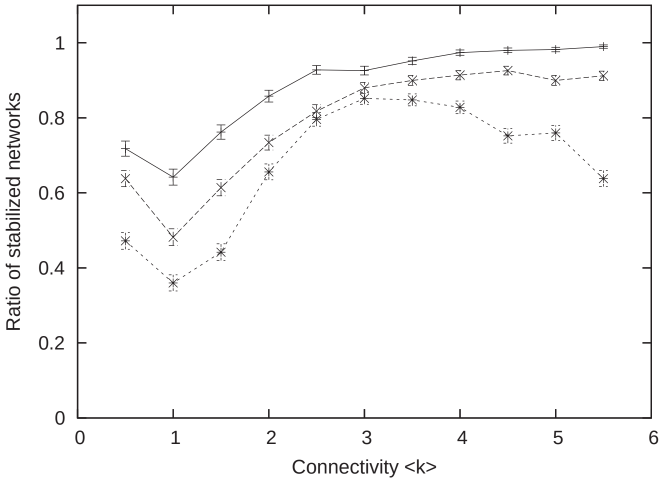

Fig. 8. Ratio of networks that were stabilized during the evolution plotted against the average connectivity of the networks for network sizes of $N = 16$ (straight line), $N = 32$ (long dashes) and $N = 50$ (short dashes).

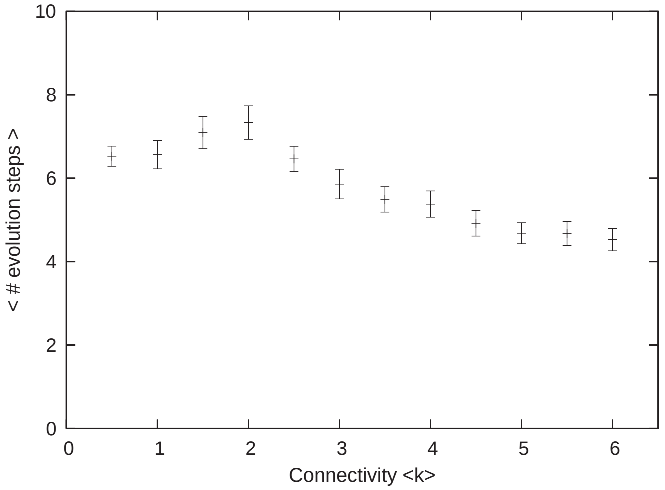

Fig. 9. Average number of evolution steps until stable realization is reached ($N = 32$).

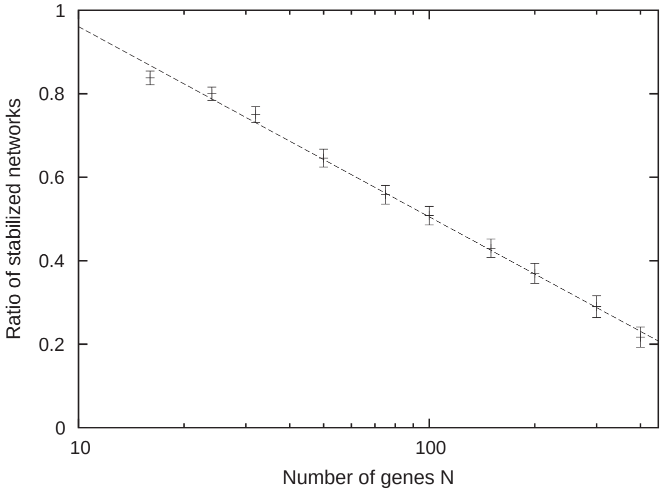

Fig. 10. Ratio of networks that were stabilized during the course of evolution plotted against the number of nodes in the networks for an average connectivity of $\langle k \rangle = 2$. The dashed line is given by a logarithmic fit of the data.

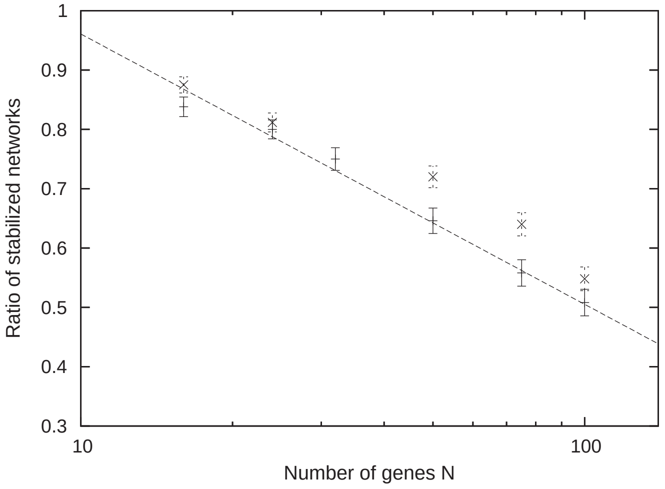

Fig. 11. Comparison of parameter values. Ratio of networks that were stabilized versus the number of nodes in the networks for an average connectivity of $\langle k \rangle = 2$. Original set of parameters marked with "+", points obtained with increased step number marked with "×".

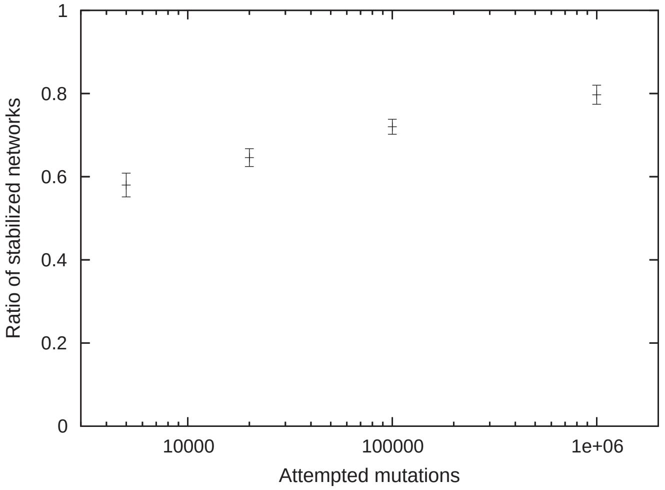

Fig. 12. Effect of parameter a_{step} on the results for $N = 50$. The ratio of stabilized networks is plotted against the value of the parameter a_{step}, giving the maximal number of attempted mutations per evolution step.

value of this parameter does not significantly affect the results, for $N > 50$, however, differences can be clearly seen. For $N = 50$ the ratio rises from 0.65 ± 0.02 at $a_{\text{step}} = 2 \times 10^5$ to 0.72 ± 0.02 at $a_{\text{step}} = 10^6$. Interestingly, for larger system sizes this effect does not seem to be amplified: for $N = 100$ the ratio rises from 0.51 ± 0.02 to 0.55 ± 0.02.

For $N = 50$ we plot the dependence of the ratio of stabilized networks on the parameter a_{step} in Fig. 12. The largest parameter value used, $a_{\text{step}} = 10^6$, is about twice the total number of possible rewirings and should thus suffice.

One can see that the decrease in the ratio of successfully evolved networks can be significantly reduced when attempting more mutations per evolution step. This is due to the fact that an enormous number of mutations is possible of which only a small fraction retains the requested dynamical sequence.

Still, one can deduce from these results that it is harder to stabilize large networks than smaller ones: even though there might be a path to a stable network instance, it may not be practically realizable as the chance to find exactly the right mutations may be too small.

However, real world systems display a large amount of modularity that leads to smaller cores of strongly interacting components. We have not taken this into account in our random network approach, nor did we consider specific connectivity patterns, for example scale-free network types, which may lead to simpler attractor structures, as well. We view our study as a model for small networks of key generators as were described in recent Boolean models of biological systems (Li et al., 2004; Davidich and Bornholdt, 2008). The resulting dynamics of the full networks are then influenced by this core without strong feedback. This allows for rather simple expression patterns of the full network without constraints on the network size.

6. Summary and conclusions

We have discussed a simple reliability criterion for biological networks and have applied it to network design features that produce reliable dynamics. We showed that small changes in the network topology can dramatically affect the dynamical behavior of a system and can lead to reliable network dynamics.

To investigate how reliability can emerge in real-world systems that have been shaped by evolution, we studied an evolutionary algorithm that selects networks with a prescribed dynamical behavior if they function more reliably than a given mother network.

We found that a high ratio of random networks can evolve towards instances displaying reliable dynamics. In accordance with other recent work (Szejka and Drossel, 2007; Ciliberti et al., 2007) it was shown that the evolution of network structures can lead to reliable dynamics, both with a high probability and within short evolutionary time scales.

Surprisingly, small connectivities are detrimental to this evolvability. This is counter-intuitive as sparsely connected networks show rather simple dynamics with short attractor lengths. However, at the same time they are difficult to evolve because they have a small structural "buffer" of links that can be neutrally rewired without changing the dynamics.

This is related to the concepts of "degeneracy" and "distributed robustness" where additional elements are present in a system that are not strictly necessary for the system's function but have a positive effect on robustness (Tononi et al., 1999; Wagner, 2005b). Here, these additional elements are links that are not strictly necessary to perform a specific function. Thus, rewiring of these links is possible and allows for a higher probability to find a network with reliable dynamics. We thus find in our framework that high connectivity, although leading to increasing complexity of the dynamics, can be beneficial for the evolution of networks.

For larger system sizes the evolvability towards reliable dynamics decreases. This is due to the increasing dynamical complexity of such networks (longer attractor cycles, more non-frozen nodes). Our strict criterion requests the reliable reproduction of the exact state sequence for every node, which leads to a more difficult selection process for large system sizes.

In summary, our results suggest that reliability is an evolvable trait of regulatory networks. In the present simple model, reliability can be achieved by topological changes alone and without fine-tuning of parameters. This means that through mutations of the reaction networks, biological systems may have the ability to rapidly acquire the property of reliable functioning in the presence of biochemical stochasticity.

Acknowledgments

The authors would like to thank Maria Davidich for discussions and helpful comments on the manuscript and Fabian Zöhrer for help with the state-space visualization. This work was supported by Deutsche Forschungsgemeinschaft Grants BO1242/5-1 and BO1242/5-2.

Appendix A. Algorithm

The asynchronous algorithm is implemented such that no discretized clock is needed. Only those times will be investigated when changes in the system happen.

For this, internal variables are needed to keep track of the dynamics. Every node i has the following state variables:

- $t_{0,i}$: time of the last change of buildup/decay behavior,
- $c_i(t_{0,i})$: concentration level at that time,
- b_i: flag for current behavior—either buildup (1) or decay (0),
- $s_{i,\text{current}}$: current discrete state of node i,
- $s_{i,\text{aim}}$: discrete state of node i that would result from the current states of all nodes: $s_{i,\text{aim}} = \Theta(\sum_{j=1}^{n} a_{ij} s_{j,\text{current}} - \frac{1}{2})$.

In addition, a global event queue Q is maintained which keeps track of future changes in buildup/decay behavior.

The system is initialized by setting all values of discrete states $s_{i,\text{current}}$ equal to the state given by the discrete initial conditions. The concentration levels are set to the same values (0.0 or 1.0). The times of the last behavior changes $t_{0,i}$ are set to 0.

Before the simulation is started, for every node i it is checked whether the aspired state $s_{i,\text{aim}}$ differs from the current state $s_{i,\text{current}}$. If so, an event is added to the queue Q (sorted by time) for time $t_d + \chi$, where χ is a uniformly distributed random number between 0 and $\chi_{\max}$.

When the simulation is run, it is checked which of the two following possible events takes place next:

1. Crossing of the concentration level of a node with the threshold value 0.5.
2. The next event in queue Q.

A simple analytical expression can be given for the times when the concentration levels are crossed (case 1). If $b_i = s_{i,\text{current}}$, the node will not switch its state because the concentration is moving away from the threshold. Otherwise, one can calculate the time of the next concentration level to cross the threshold by solving Eq. (3) for t with $c_i = 0.5$:

$$\min_i [t_{0,i} + \tau \log(1 + |1 - 2c_i(t_{0,i})|)]. \tag{4}$$

If an event of type 1 happens next, the discrete state of the respective node i, $s_{i,\text{current}}$, is updated and the effect on other nodes is calculated. For definiteness, let us assume this crossing takes place at time t. If this switch causes the aspired state of another node j to switch, an event is sorted into the queue Q at $t + t_d + \chi$. When in the queue events for the same node are scheduled to happen at later times, they will be removed. They are thought to have been "caught" by the newly added event.

In the second case, the concentration level of the node at time t is calculated according to Eq. (3) and saved as $c_i(t_{i,0})$ with the new time $t_{i,0}$. The behavior flag b_i is switched to reflect that the node has changed from buildup to decay or vice versa.

If the time between any two successive node state changes in the network (not necessarily of the same node) is larger than $t_d/2 + \tau$, the node states are recorded and set as a new step to be compared to the synchronous attractor.

References

Albert, R., Othmer, H.G., 2003. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster.* J. Theor. Biol. 223 (1), 1–18.

Aldana, M., Balleza, E., Kauffman, S., Resendiz, O., 2007. Robustness and evolvability in genetic regulatory networks. J. Theor. Biol. 245, 433–448.

Alon, U., Surette, M.G., Barkai, N., Leibler, S., 1999. Robustness in bacterial chemotaxis. Nature 397, 168–171.

Barkai, N., Leibler, S., 1997. Robustness in simple biochemical networks. Nature 387 (6636), 913–917.

Becskei, A., Serrano, L., 2000. Engineering stability in gene networks by autoregulation. Nature 405 (6786), 590–593.

Bornholdt, S., 2005. Systems biology: less is more in modeling large genetic networks. Science 310 (5747), 449–451.

Bornholdt, S., Sneppen, K., 2000. Robustness as an evolutionary principle. Proc. R. Soc. London B 267, 2281.

Braunewell, S., Bornholdt, S., 2007. Superstability of the yeast cell-cycle dynamics: ensuring causality in the presence of biochemical stochasticity. J. Theor. Biol. 245 (4), 638–643.

Braunewell, S., Bornholdt, S., 2008. Reliability of genetic networks is evolvable. Phys. Rev. E 77 (6), 060902(R).

Chaves, M., Albert, R., Sontag, E.D., 2005. Robustness and fragility of Boolean models for genetic regulatory networks. J. Theor. Biol. 235, 431–449.

Chaves, M., Sontag, E.D., Albert, R., 2006. Methods of robustness analysis for Boolean models of gene control networks. IEE Proc. Syst. Biol. 153 (4), 154–167.

Ciliberti, S., Martin, O.C., Wagner, A., 2007. Robustness can evolve gradually in complex regulatory gene networks with varying topology. PLoS Comput. Biol. 3 (2).

Davidich, M.I., Bornholdt, S., 2008. Boolean network model predicts cell cycle sequence of fission yeast. PLoS ONE 3, el672, doi:10.1371/journal.pone.0001672.

Elowitz, M.B., Leibler, S., 2000. A synthetic oscillatory network of transcriptional regulators. Nature 403, 335–338.

Espinosa-Soto, C., Padilla-Longoria, P., Alvarez-Buylla, E.R., 2004. A gene regulatory network model for cell-fate determination during *Arabidopsis thaliana* flower development that is robust and recovers experimental gene expression profiles. Plant Cell 16 (11), 2923–2939.

Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81 (25), 2340–2361.

Glass, L., 1975. Combinatorial and topological methods in nonlinear chemical kinetics. J. Phys. Chem. 63 (4), 1325–1335.

Greil, F., Drossel, B., 2005. Dynamics of critical Kauffman networks under asynchronous stochastic update. Phys. Rev. Lett. 95 (4), 048701.

Kauffman, S.A., 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. J. Theor. Biol. 22, 437–467.

Kauffman, S.A., 1993. The Origins of Order. Oxford University Press, New York.

Kepler, T.B., Elston, T.C., 2001. Stochasticity in transcriptional regulation: origins, consequences, and mathematical representations. Biophys. J. 81 (6), 3116–3136.

Klemm, K., Bornholdt, S., 2005a. Stable and unstable attractors in Boolean networks. Phys. Rev. E 72 (5), 055101(R).

Klemm, K., Bornholdt, S., 2005b. Topology of biological networks and reliability of information processing. Proc. Natl. Acad. Sci. USA 102 (51), 18414–18419.

Kollmann, M., Lovdok, L., Bartholome, K., Timmer, J., Sourjik, V., 2005. Design principles of a bacterial signalling network. Nature 438 (7067), 504–507.

Li, F., Long, T., Lu, Y., Ouyang, Q., Tang, C., 2004. The yeast cell-cycle network is robustly designed. Proc. Natl. Acad. Sci. USA 101 (14), 4781–4786.

Ma, W., Lai, L., Ouyang, Q., Tang, C., 2006. Robustness and modular design of the drosophila segment polarity network. Mol. Syst. Biol. 2, 70.

McAdams, H.H., Arkin, A., 1999. It's a noisy business! Genetic regulation at the nanomolar scale. Trends Genet. 15 (2), 65–69.

Ozbudak, E.M., Thattai, M., Kurtser, I., Grossman, A.D., van Oudenaarden, A., 2002. Regulation of noise in the expression of a single gene. Nat. Genet. 31 (1), 69–73.

Rao, C.V., Wolf, D.M., Arkin, A.P., 2002. Control, exploitation and tolerance of intracellular noise. Nature 420, 231–237.

Raser, J.M., O'Shea, E.K., 2005. Noise in gene expression: origins, consequences, and control. Science 309 (5743), 2010–2013.

Szejka, A., Drossel, B., 2007. Evolution of canalizing Boolean networks. Eur. Phys. J. B 56, 373–380.

Thattai, M., van Oudenaarden, A., 2001. Intrinsic noise in gene regulatory networks. Proc. Natl. Acad. Sci. USA 98 (15), 8614–8619.

Tononi, G., Sporns, O., Edelman, G.M., 1999. Measures of degeneracy and redundancy in biological networks. Proc. Natl. Acad. Sci. USA 96 (6), 3257–3262.

von Dassow, G., Meir, E., Munro, E.M., Odell, G.M., 2000. The segment polarity network is a robust developmental module. Nature 406, 188–192.

Wagner, A., 1996. Does evolutionary plasticity evolve? Evolution 50, 1008–1023.

Wagner, A., 2005a. Circuit topology and the evolution of robustness in two-gene circadian oscillators. Proc. Natl. Acad. Sci. USA 102 (33), 11775–11780.

Wagner, A., 2005b. Distributed robustness versus redundancy as causes of mutational robustness. Bioessays 27 (2), 176–188.