Stefan Braunewell

Reliability and Evolvability of Genetic Regulatory Networks

Reliability and Evolvability of Genetic Regulatory Networks

Vom Fachbereich für Physik und Elektrotechnik
der Universität Bremen
zur Erlangung des akademischen Grades eines
Doktor der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation

von
Stefan Braunewell

1. Gutachter: Prof. Dr. rer. nat. S. Bornholdt
2. Gutachter: Prof. Dr. rer. nat. K. Pawelzik

Eingereicht am: 11.3.2008
Tag des Promotionskolloquiums: 23.4.2008

Abstract:

Reliability and Evolvability of Genetic Regulatory Networks

Living organisms are remarkably robust despite fluctuating concentrations of functional molecules in the cell and changing environmental conditions. In the biological literature, the question how organisms cope with this stochasticity has been investigated in theory and experiment in specific organisms. To identify and understand general mechanisms that facilitate reliable dynamical behavior, computer modelling can be useful to investigate specific effects in isolation.

In this thesis, the effect of the topological structure of transcriptional regulation networks on the reliability of the resulting dynamics is investigated in simple dynamical models. The activity of genes and proteins is modeled by discrete values. An extension of this discrete dynamical model to continuous time is used and molecular fluctuations are implemented by random delays of signals. Reliability of the dynamics is defined as ordered dynamical behavior despite these fluctuations.

Using this criterion, simple systems of interacting genes as well as the model organism budding yeast *S. cerevisiae* are investigated. The reliability of the cell-cycle regulation is assessed and simple mechanisms of the regulational organization are identified which lead to the robust dynamical behavior.

Further, the recently discovered feature of biological networks to display a non-random distribution of interaction patterns among triads of nodes, the "motif distribution," is investigated. In a simple evolutionary model using a suitable selection criterion, dynamically robust networks are produced. However, these networks do not display the expected motif distributions.

This points to an ability of the evolution model to create reliable dynamics without significant changes of the network structure. To further explore this, it is investigated how easily reliable networks emerge in an evolution process using different models: First, the reliability of all dynamical attractors of networks shall be accomplished. An astonishing evolvability towards reliable dynamics is observed. Second, a specific "functional" attractor is defined that has to be reproduced reliably to reach the goal of the evolution. Most such evolution processes successfully finish in this criterion. These results indicate that dynamical reliability is an evolvable property of regulatory systems.

Kurzfassung:

Zuverlässigkeit und Evolvierbarkeit genetischer Regulationsnetze

Lebende Organismen sind erstaunlich robust trotz fluktuierender Konzentrationen der funktionalen Moleküle in der Zelle und variierender Umgebungsbedingungen. In der biologischen Forschung wurde die Fragestellung, wie Regulationssysteme trotz dieser Stochastizität zuverlässig funktionieren, in theoretischen und experimentellen Untersuchungen spezieller Organismen durchgeführt. Um generelle Mechanismen, die zuverlässiges dynamisches Verhalten begünstigen, zu identifizieren und zu verstehen, eignet sich besonders die Modellierung im Computer, mit der spezielle Effekte isoliert betrachtet werden können.

In dieser Dissertation soll der Effekt der topologischen Struktur von Regulationsnetzen auf die Zuverlässigkeit der resultierenden Dynamik in einfachen dynamischen Modellen untersucht werden. Die Aktivitäten von Genen und Proteinen werden dabei als diskrete Werte modelliert. Eine Erweiterung dieser diskreten Dynamik auf kontinuierliche Zeit wird verwendet und molekulare Fluktuationen werden durch zufällige Verzögerungen der Signale zwischen den Genen implementiert. Zuverlässigkeit der Dynamik wird unter dem Einfluß dieser Fluktuationen durch geordnetes dynamisches Verhalten definiert.

In diesem Kriterium werden einfache Systeme interagierender Gene sowie der Modellorganismus der Bierhefe *S. cerevisiae* untersucht. Die Zuverlässigkeit der Zellzyklus-Regulation wird festgestellt und einfache Organisationsmechanismen des Regulationssystems, die zu einem robusten dynamischen Verhalten führen, werden identifiziert.

Weiterhin wird eine kürzlich entdeckte Eigenschaft biologischer Netze, eine spezielle Verteilung der Interaktionsmuster innerhalb von Dreiergruppen von Komponenten, die sogenannte "Motivverteilung", untersucht. In einem einfachen Evolutionsprozeß mit einem geeigneten Selektionskriterium werden dynamisch robuste Netze erzeugt. Diese Netze weisen jedoch nicht die erwarteten Motivverteilungen auf.

Dies deutet darauf hin, daß das Evolutionsmodell zuverlässige Dynamik erzeugen kann, ohne große Veränderungen an der Struktur der Systeme zu verursachen. Um dies weiter zu erforschen, wird untersucht, wie zuverlässige Dynamik in einem Evolutionsprozeß entsteht. Dazu werden verschiedene Modelle benutzt: Zum einen soll die Zuverlässigkeit aller dynamischen Attraktoren eine Netzes erreicht werden. Es wird eine erstaunliche Evolvierbarkeit zu zuverlässiger Dynamik beobachtet. Zum anderen wird ein spezieller "funktionaler" Attraktor definiert, der im Verlaufe der Evolution zuverlässig reproduziert werden muss. Die meisten Evolutionsprozesse können in diesem Kriterium erfolgreich abgeschlossen werden. Diese Ergebnisse weisen darauf hin, daß dynamische Zuverlässigkeit eine evolvierbare Eigenschaft von Regulationssystemen ist.

Contents

1 Introduction

The existence and tenacity of life in all its forms is one of the most stunning phenomena of our world. Controlled by regulatory systems of molecular components which are inherently stochastic in their dynamics, and facing ever-changing environmental conditions, living organisms truly display a remarkable capability for robust functioning. To identify sources and mechanisms of this robustness is one of the main goals of systems biology as it does not suffice to tackle this question by investigating the properties of the individual components alone [1].

This endeavor has to be approached at all different levels of cellular organization – even the oldest known forms of life display an amazing complexity of internal composition at vastly different scales. Investigations from the cellular all the way down to the molecular level are necessary to understand the internal workings of living organisms. In recent years, advances in experimental techniques have allowed more and more detailed and precise descriptions of the processes taking place in the cell. This has opened up the possibility to investigate, model and understand these enormously complex systems of regulatory control.

All of the regulatory processes in cells are fundamentally driven by biochemical reactions [2]. As they take place in a warm environment and often require very specific reactants that potentially occur in low copy numbers, they are affected by stochastic fluctuations. To attenuate this noise, specific mechanisms of noise control are necessary and they are implemented at different levels of the regulatory system. In recent years, insight into these noise control mechanisms has been assembled both from experimental observations as well as from theoretical models and simulations.

At the lowest level of organization, the process rates of the individual chemical reactions can influence the effect of fluctuating reactant concentrations [3]. At the next level of organization, properties of the individual genes come into play: an increase in the copy number of a gene (through gene duplication or polyploidy) leads to a reduction in the intrinsic noise [2, 3]. Yet advancing in complexity, another well-characterized noise-reduction mechanism is negative autoregulation [4, 5], in which a gene's product inhibits its own production.

Although all of these mechanisms already involve complex cascades of biochemical processes, they all take place at the level of a single gene or protein species. The regulatory processes in cells, however, are much more complicated, involving the interplay of several different molecular species that can promote, inhibit, functionally modify or inactivate each other. Thus, the question was raised how the architecture of a reaction system can influence its robustness. In [6], a robust adaptation mechanism in the chemotactic system of the bacteria *E. Coli* was suggested that leads to correct chemotactic response over a wide range of concentration levels.

In the investigation of biological robustness, different notions of the term have been used,

ranging from robustness against genetic mutations or changes in the model parameters (rate constants) over modified initial conditions to stochastic dynamical processes [1]. Using experimental and modeling techniques, robustness properties of such different biological systems as bacterial chemotaxis [7], circadian clocks [8], embryonic patterning in fruit fly [9, 10, 11, 12], the cell cycle of the budding yeast [13] and flower development [14] have been explored. In recent comparative computational studies, 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, with the experimentally established network ranking high among the robust topologies [15, 16, 17].

While all of the previously mentioned investigations demonstrate that individual organisms perform their tasks reliably, it is interesting to ask whether general properties of the network architecture can be identified that aid in robust functioning. To investigate questions of the network topology, it has been suggested to neglect details of the reaction machinery and use simplified, abstract rules as dynamical models [18]. This allows to focus on the topological features of the reaction networks.

In the case of transcriptional regulation, this means a reduction to genes and their interactions. Many more components (RNA, proteins, complex molecular machines) are active in these processes and their influence is subsumed under one simple interaction type between two genes. A particularly simple model that captures main features of transcriptional regulatory networks, "Random Boolean Networks", was introduced as early as 1969 [19]. In this model, genes can implement Boolean functions of their inputs, i.e. the states of those genes whose transcript serve as transcription factors. It is widely accepted that discrete dynamical systems inspired by these Boolean networks serve as useful models of transcriptional regulation [11, 13, 14]

To incorporate the effects of molecular fluctuations into discrete models, one commonly used approach is to allow random flips of the node states [11, 13, 20, 21]. However, the perturbation by node state flips seems to be a very harsh form of noise: in real organisms, concentrations and timings fluctuate but the qualitative states of genes are often quite stable. A more subtle form of fluctuations can be implemented in the timing of switching events [22]. The principle idea is to allow fluctuations of event times and test whether dynamical behavior stays ordered despite these fluctuations. It was suggested that biologically relevant dynamical behavior should not depend on an exact synchrony. In this thesis this viewpoint will be adopted and reliability of dynamics will be defined as the ability to function orderly despite fluctuations in the timing of signals.

With sufficiently simple dynamical models, the effect of the network topology on dynamical features can be studied. Measures such as degree distribution, clustering coefficient and modularity have been used to characterize the structural properties of biological networks [23, 24]. In the framework of Boolean dynamics it has been shown that the average number of links in random networks is crucial for the dynamical properties [25]. Also the distribution of the number of links plays a role: networks with a scale free degree distribution display more ordered dynamics than random networks (with a Poissonian degree distribution) of the same mean degree [26, 27, 28].

Also the local wiring structures among triads of nodes were investigated and it was found

that real-world signaling networks show specific patterns called "motifs" which distinguish them from random networks [29, 30, 31]. This finding sparked great interest and a series of works explored the characteristics and potential causes of the motifs [32, 33, 34, 35, 36, 37]. In particular, for isolated triads, the robustness against parameter changes in a differential equation model [34] and the reliability against perturbations of the signal transmission times [33] have been shown to correlate with the abundance of the respective triads in real-world networks.

However, biological networks have not been engineered with design principles in mind, but instead have emerged from evolutionary procedures. Using computer simulations, it is possible to evolve artificial networks by selecting those that perform well under a given criterion. In this framework one can investigate whether an evolutionary procedure can account for reliability of network dynamics.

The concept of robustness as the sole constraint of evolutionary selection was proposed in a model that selects a network if it reproduces the behavior of a mother network with randomly chosen initial conditions [38, 39]. Surprisingly, even this minimal selection criterion leads to network behavior of the evolved networks that is markedly different from the dynamical properties of random networks.

Furthermore, in an investigation on mutations of network structure, it was found that a population of networks can evolve to a state in which the effect of mutations on expression patterns is significantly reduced [40]. A very recent study has shown that networks can simultaneously be robust against structural changes (attractors are conserved) as well as evolvable (new attractors emerge) [41]. Finally, network evolution towards robustness against gene state errors in the initial configuration turns out to be an easy and rapid process [42, 43].

However, all of these previous works consider robustness against perturbations that take place before the regulatory system conducts its dynamical behavior. In contrast, in this thesis the perturbations are caused by the intrinsic fluctuations during the dynamical time-course and it will be investigated how reliable dynamical functioning can be achieved despite these fluctuations. On one hand, features of the dynamical behavior that determine reliability will be identified in models of simple artificial circuits as well as of the regulatory network of budding yeast. On the other hand, as biological systems emerge from evolution, the main focus will be on whether and how reliable dynamics can emerge from network evolution processes.

It will be investigated whether selecting networks for reliable dynamics causes modifications of the motif distribution as expected from the reliability assessments of the isolated triads. Furthermore, it will be explored to what extent regulatory networks possess the ability to evolve towards reliable dynamics. The structural and dynamical effects of such network evolutions will be investigated.

To tackle these problems, a new model of genetic networks is introduced that incorporates the main features of regulatory dynamics, yet is simple, needs a minimal set of parameters and can be efficiently simulated. The main ingredients are a low-pass filter in the time course of protein concentration levels and signal transmission delay (i.e. a delay between the concentration change of a gene's promoter and the effect on this gene's product) that

can randomly fluctuate. This model will be introduced in chapter 5 and will be used to test the reliability of simple circuits. Features of the dynamical behavior and isolated structural properties that aid robust functioning will be studied.

As an application of the dynamical model to a real-world organism, budding yeast *Saccharomyces Cerevisiae* and one of its most fundamental processes, the cell cycle, is analyzed. The cell cycle of budding yeast is one of the best studied biological systems and the regulatory processes are known in exceptional detail [44, 45]. In [13] a small network of eleven core regulators was identified from the data and it was shown in a simple Boolean threshold model, that the experimentally observed sequence of protein activities can be successfully reproduced in a simple discrete model almost devoid of tunable parameters. Because of the simplicity of the model, it provides a perfect testing ground for the investigation of reliability in a concrete living system. By applying the methods developed in this thesis, the reliability of the cell cycle sequence is investigated and it is shown that specific features of the network dynamics facilitate robustness against timing noise (chapter 6).

After these investigations of dynamical reliability in fixed networks, the question of evolvability of reliable dynamics will be examined. In chapter 8, an evolutionary process selecting for reliable network dynamics will be devised and it will be studied whether the motif structure of real-world networks can be explained by the constraint of reliable dynamics. Using the asynchronous dynamics introduced in chapter 5 and employing a suitable selection criterion, networks are evolved towards reliability of dynamics and the resulting motif distributions are investigated. As only Boolean threshold functions are used as rulesets of the nodes, the dynamical outcome and reliability properties can only be influenced by the network's wiring structure and the type of the link (activating or inhibiting). Thus, in this framework, one can isolate the effect of network structure on the reliability.

In chapter 9, the evolvability of networks towards reliable dynamics shall be explicitly quantified. For small networks and in the limit case of small perturbations, a different model can be utilized that was introduced in [46]. Systematically checking whether small perturbations persist or die out, one can assess the reliability of a network without explicitly simulating the asynchronous dynamics and obtains a deterministic reliability measure. With this criterion, two different evolution processes will be investigated. First, the reliability of the complete attractor landscape is evolved to quantify how the reliability constraint affects the network structure. Second, it is investigated whether it is possible to evolve networks to reproduce a given dynamical sequence, termed "functional attractor," and to do this with reliable dynamics.

After this implementation of a more abstract dynamical model, also the model from chapter 5 is used to investigate the question of evolution to a reliable functional attractor. While the deterministic criterion is well suited for quantifying the effects of the evolution, explicitly modeling the time course of the dynamics can be more easily communicated as a realistic model of regulatory systems. It also allows larger system sizes and therefore complements the results from chapter 9.

2 Biological principles of gene regulation

Living organisms are controlled by incredibly complex systems of biochemical molecules, that intricately interact with each other to regulate the processes required for survival and reproduction. Even inside individual cells, the organization of the biochemical reactions displays an astonishing complexity. In this chapter, the biological principles underlying regulatory processes will be introduced. This is not an attempt to comprehensively survey what happens in biological cells (for an in-depth introduction, see e.g. [47]). In order to keep these sections short, only an overview will be given that applies to archaea, prokaryotes and eukaryotes alike and neglects many of the details and differences of the individual domains.

The foundations that are needed to understand the results of this thesis as well as results from recent biological research with immediate implications on this work will be covered. First, a very brief overview of the biochemical processes in living cells will be given in section 2.1. Then, the general principles underlying transcriptional regulation and regulatory networks will be covered in section 2.2 along with a short review of experimental methods. Finally in section 2.3, recent results on the origins and the control of fluctuations in cells will be reviewed.

2.1 Gene expression

Many processes in cells are controlled and conducted by proteins: they work as enzymes to catalyze chemical reactions, for example to participate in the metabolism and DNA replication, repair and RNA synthesis; they also take vital roles in cell signaling and signal transduction and act as structural components to give fluid elements rigidity and stiffness. In fact, the essential role proteins play for all living organisms is already expressed in the term "protein" itself (from Greek "$\pi\rho\omega\tau\alpha$" = "of prime importance").

Proteins are three-dimensional structures composed of amino acids. They are coded for in genes and the process of protein synthesis is regulated via a cascade of complex molecular processes – the final outcome of the production of proteins from DNA strands is called "gene expression". The simplified description that is used as a general framework of protein synthesis and regulation is known as the "central dogma of molecular biology". Although nowadays it is known that in real cells the interaction of the many components is much more complicated, it serves as a good starting point for the explanation of the different components.

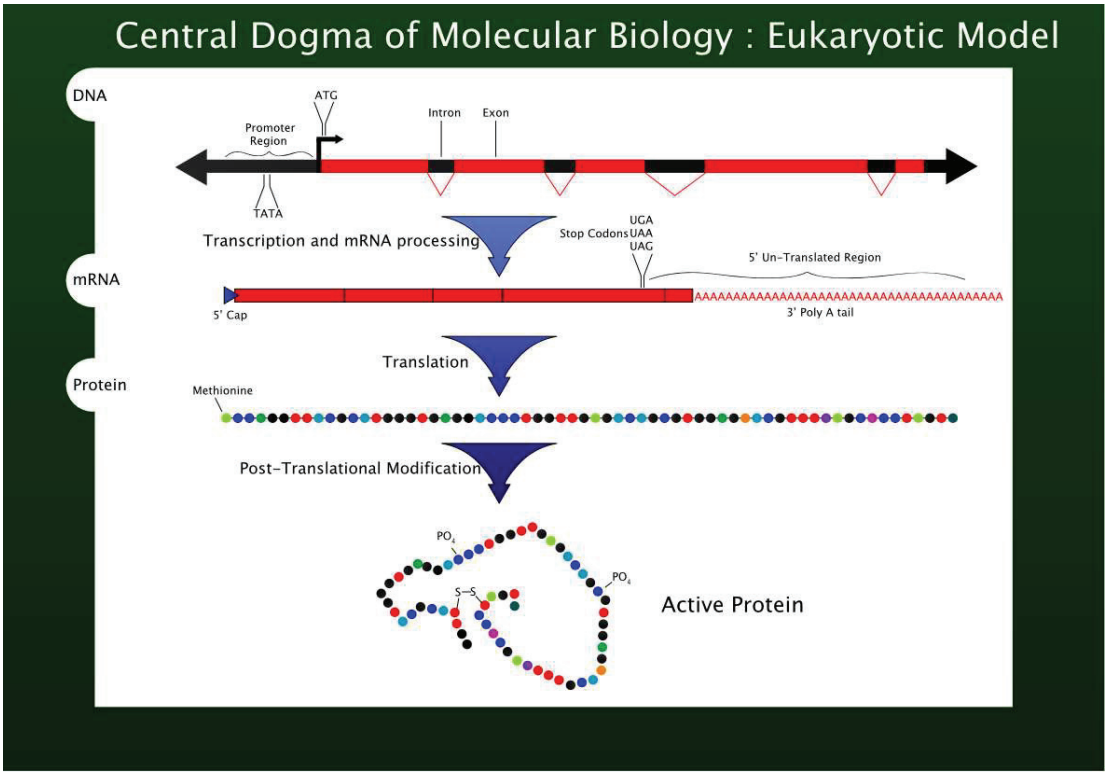

Figure 2.1: Overview of the central dogma of molecular biology. Image by Mike Jones.

In figure 2.1 an overview of the typical process cascade is given. At the top, a strand of DNA is shown. The DNA (Deoxyribonucleic acid) contains all the genetic information necessary for the development and functioning of an organism. Chemically, it is a "nucleic acid", i.e. a chain of units called "nucleotides" with a backbone of sugar and phosphate groups and attached bases (one per nucleotide) of one of four types: guanine (G), adenine (A), thymine (T) and cytosine (C). In living organisms, DNA usually occurs in the form of a double helix, in which two strands are bound to each other due to base pairing: each of the bases can pair with its complementary base, G pairs with C, A pairs with T, thus forming a connection between two complementary strands of DNA.

The DNA strand depicted in the figure codes for a protein. The promoter region controls the transcription by allowing a specific enzyme, the RNA polymerase to bind to the strand to start the transcription (this binding is mediated by so-called "transcription factors" in eukaryotes). In the process of transcription, the RNA polymerase traverses the strand, creating another nucleic acid called RNA (Ribonucleic acid) by base pairing, therefore creating a copy of the other DNA strand (called "coding strand"). Ribonucleic acid is composed of ribose (instead of deoxyribose in DNA) and usually occurs as a single-strand. In contrast to DNA, thymine is usually replaced by uracil, which is energetically favored but less stable.

In the special case of mRNA transcription, the resulting mRNA after additional processing can be translated by an organelle, the ribosome, which in turn is a complex machinery consisting of RNA and proteins. The ribosome uses amino acids and assembles them according to the code of the mRNA. Three bases of RNA (collectively called a "codon") code for one amino acid. This translation process terminates at specific stop codons. The result of this process is then a string of amino acids, a protein.

This string of amino acids then folds into an active protein. In fact, this process starts as soon as a part of the chain emerges out of the ribosome. Immediately, the posttranslational modifications of the chain set in, with proteins and RNA modifying the structure by adding or removing amino acids or other functional groups (phosphates, carbohydrates, etc.), cutting and by making structural changes.

In reality all these processes are much more complicated and all steps of the central dogma are actually regulated and influenced by other molecules in the cell. For example, there exist non-coding genes, that do not produce proteins by the described process, but instead give rise to functional RNA molecules, such as miRNA (micro-RNA) that can prevent mRNA molecules from being translated [48]. For the assemblage of the amino acids into a protein, another sort of RNA, the transfer RNA (tRNA), is necessary which binds to amino acids and provides them to the ribosome, according to the codon prescription on the mRNA strand.

Also, the DNA is not just freely floating, but instead is wound around structural proteins called "histones," which can prevent gene expression by simply disallowing the RNA polymerase to access the respective binding sites. The histone organization is itself regulated by cellular processes and can be changed through specific enzymes called "remodeling factors."

Another complication specific to eukaryotic organisms is due to the spatial organization of the cellular compartments. The DNA is located in the nucleus of the cell and thus the transcription and mRNA production takes place inside the nucleus. However, ribosomes exist both in the plasma as well as bound to the cell membrane. The transcribed mRNA molecules have to diffuse through the nuclear envelope to be translated by the ribosomes. Similar diffusion and transport processes are necessary for all parts of cellular control.

For the foundations of this work it is sufficient to know that genes code for proteins that fulfill particular functions in the cell. As we look at the cellular processes from a system's perspective, the details of the synthesizing process will not concern us here. Instead, the whole cascade of processes which ultimately produce a protein will be abstracted as one, where the genes act as "factories" for the proteins. The rest of the cellular machinery is assumed to be present and active.

2.2 Gene regulatory networks

One important role of proteins is to regulate the process of protein synthesis itself. These proteins are called "transcription factors" and they can bind at a gene's binding site in the promoter region by special DNA binding domains. Being bound to the DNA, they can

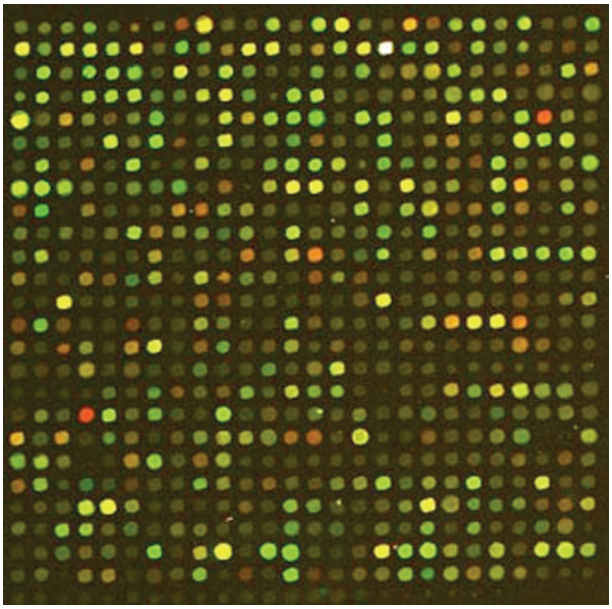

Figure 2.2: Example of microarray hybridization. Taken from Reinke, V. *Germline genomics*, http://www.wormbook.org [49]

allow, enhance, reduce or block the binding of the RNA polymerase to promote or suppress the transcription of the specific gene's DNA sequence. In this way, genes and their protein transcripts form a dynamical network.

The simplest example of a regulatory network is one consisting of a single gene, whose transcript is its own transcription factor or inhibitor, thus forming a feedback system. In general, any complex interaction among genes is possible and real-world networks have been characterized consisting of hundreds of genes interacting with each other in this way [44, 45]. To experimentally determine these interaction networks, several techniques have been developed, and here only a very brief collection of them shall be given.

In the so-called "microarray analysis" [50], parts of single-stranded DNA, approximately corresponding to single genes, are attached to a glass surface. In mRNA or gene expression profiling, mRNA is reversely transcribed to DNA which is then fluorescently labeled and applied to the microarray, where it binds to those spots to which its sequence matches. Through fluorescence measurements, one can thus obtain a rather noisy estimate of expression levels of the individual genes – an example of a microarray experiment of the worm *C. elegans* is shown in figure 2.2. The differently colored spots visualize the expression levels of individual genes.

Whereas this technique allows a snapshot of the expression of genes, a more direct measurement of the binding of a transcription factor to the regulatory regions of genes is also possible. In chromatin-immunoprecipitation ("ChIp") experiments [51], *in vivo* DNA-protein bindings are fixated and the DNA strand is broken into small pieces. Using

antibodies that bind to the specific protein under consideration, those pieces of the DNA with the protein attached are isolated and can be detected on a microarray chip ("ChIp-on-chip"). One large-scale application of this method was used to map transcriptional regulation interactions in *S. Cerevisiae* [45].

While all these methods require a large number of molecules and one typically averages over a cell population, recently, measurement techniques have been developed to investigate concentration levels in single cells. By incorporating fluorescent reporter proteins, the level of abundance of a protein in individual cells can be measured under the microscope [52, 53, 54, 55].

Using fluorescence in-situ hybridization ("FISH") [56] even single mRNA molecules can be detected. In this method, fluorescent DNA strands (complementary to the mRNA) bind to the mRNA molecules and can be detected by fluorescence microscopy methods. To achieve single molecule resolution, several such probes can be used that bind to different regions of the same mRNA molecule [57].

So far, only natural organisms have been discussed. It is also possible to assemble artificial gene regulatory networks to study specific features in isolation. In [58], a network of mutually inhibiting genes was constructed by the insertion of strands of DNA into the promoter regions of genes in the bacteria *E. coli*. This system displays the characteristics of a bistable switch. In [59], a closed feedback system consisting of three elements was constructed. It was shown that this system exhibits stable oscillatory behavior. In another study, the effect of autoregulation on a gene was experimentally studied in such a synthetic network [4].

2.3 Noise in gene expression

The individual processes that make up the networks of gene regulation are subject to sizable fluctuations [2, 3, 60, 61]. All steps of the procedure of protein production are governed by chemical reactions directly or indirectly involving several different species of molecules whose concentration in turn is dependent on multiple factors.

One can discriminate two main causes of noise [52, 62]: extrinsic noise caused by fluctuations outside of the domain of the system involved and intrinsic noise due to the inherent stochasticity of the regulatory processes.

Extrinsic noise can be characterized by having the same effect on two reporters of the same gene in a single cell but causing differences between reporters in different cells in a culture or on different genes in the same cell [52]. Extrinsic noise can be caused by differences in local environment or in the concentration or activity of components (such as RNA polymerase or ribosomes) affecting regulatory processes. They can be further distinguished into global noise which directly affects the processes involved at all genes or gene-specific noise caused by differences in the concentrations of certain transcription factors. Examples of causes for extrinsic noise are concentration differences of morphogens (signaling molecules) or cells being at different stages of the cell cycle in unsynchronized cell cultures.

Intrinsic noise is caused by the stochasticity inherent to the chemical processes itself. Chemical reactions are of an essentially random nature and due to the low copy numbers of molecular species involved, the fluctuations in reaction events are not averaged out as in well-mixed large chemical reactions. One main cause of noise in expression levels is the nature of the transcription and translation processes itself, which are believed to happen in bursts at random time intervals [2, 63, 64].

There are features in the transcriptional machinery that suppress fluctuations to ensure a more deterministic regulation procedure [3]. It has been shown that efficient transcription followed by inefficient translation is favorable in terms of reducing the fluctuation in the produced protein levels [5, 64, 65, 66, 67]. Similarly, frequent promoter transitions followed by inefficient transcription leads to reduced mRNA fluctuations [68, 69, 70]. In both cases, the effect of noise reduction is due to the fact that variations in the first process are not amplified by the second. However, this comes at the cost of higher energy demand.

Noise-regulating features at higher organizational levels have already been discussed in the introduction. Here, we just want to add that organisms also have explicit control features which can prevent mistakes due to noise from potentially destroying the organism. In the eukaryotic cell cycle, such regulatory checkpoints verify whether the required processes have been accurately completed before the cell is allowed to continue in the cell cycle process [2, 71].

On a side note, one has to mention that noise can also be a necessary property, for example when phenotypic variation from a genotypically equivalent population is desired [52, 57]. This is a topic distinctly different from the focus of this work and shall not be covered here. Throughout this work, noise will be regarded as detrimental to the ordered behavior of the organism under inspection.

3 Modeling of regulatory networks

The advent of new and more precise experimental techniques in the realm of molecular biology has created an urgent need for quantitative theoretical models to interpret and understand experimental results. The traditional approach to this problem is to dissect such systems into the most fundamental, independently acting components and to describe their behavior and dynamical features in the most detailed way accessible to current experimental methods. Although very successful in describing processes at the fundamental level, such as the molecular processes happening during transcription and translation, this approach is limited by the complexity of large scale systems.

Both the interpretation of results and the computational limits in numerical simulations call for simplified descriptions that neglect many explicit details of the underlying molecular reactions but capture the essential features of the processes involved. From a physics perspective, this is the natural way to approach a complex problem as it may lead to insights that apply to a potentially broad range of systems.

However, in molecular biology, this task is extremely difficult: the processes take place at vastly different scales – both in time as well as space – and it is unclear which level of detail has to be chosen to correctly describe biological phenomena. Thus, a number of different approaches and modeling techniques has been developed and we want to give a very brief overview of some of these techniques in this chapter.

First, we give a short introduction into the young field of network science, which is concerned with the topological features of systems of interacting components. Then, common modeling techniques of biological systems are discussed.

3.1 Complex networks

The recent interest in complex networks was mainly triggered by two studies that discovered that real world complex systems, when abstracted to the level of networks, display features that are markedly different from random graphs. First, it was noted that many natural systems show a structure that incorporate features of both regular lattices as well as of random graphs, a property which has been termed "small world" networks [72]. "Complex networks" are often thought of as systems that lie somewhere in between order and randomness. In [73] another typical feature of many real-world systems was observed. As opposed to random graphs, which display a Poissonian distribution of the number of connections, it was found that in many real-world system, this distribution is rather fat-tailed – which has become famous under the term "scale free" networks.

These structural investigations of complex systems sparked a wealth of research in physics

and many systems, ranging from social over technological to biological systems have since been investigated with regards to the topology of their interactions. For extensive coverage of these topics, several reviews can be recommended [74, 75, 76]. As the terminology of the network literature is used extensively in this work, the following overview of network features also serves as a definition of the key concepts.

A network is an abstraction of a system composed of nodes and links. The "nodes" (or "vertices") are the objects of the system, for example people in social networks (there often termed "agents"), cities, airports, crossings in infrastructure networks, or web pages in the world wide web. "Links" (or "edges") denote the interactions of these nodes, for example the acquaintance of people, tracks between two train stations or a hyperlink leading from one web page to another. Links in networks can either be directed ("from a to b") or undirected, which amounts to directed links in both directions. In this work, the word "link" always refers to a directed link as undirected links can be seen as a special configuration of directed ones.

In real-world systems, it is a matter of choice, which network exactly is constructed. In this respect, networks are always idealized pictures of subsystems. Whether a particular network is a faithful representation of a real system, can only be judged in the respective quantities that one is interested in. Even a seemingly closed picture of a network as in the example of pages in the world wide web and hyperlinks between these pages is incomplete as it neglects for example the users that might make connections between pages by personal communication. Thus, it has to be kept in mind that networks are always coarse-grained models and the validity of this approach has to be tested against real-world data.

When a particular choice is made about the nature and properties of the nodes and links of the system, the topology of networks can be investigated rather easily and a lot of systems have been analyzed and categorized according to structural properties.

A prime example of a structural feature is the so-called "degree" of a node, which is defined as the number of connections this node has. One can distinguish the in-degree and the out-degree of a node, counting only links pointing to or away from the node, respectively. The "connectivity" of a network is the arithmetic mean over all node degrees. The degree distribution is the distribution of the node degrees.

As was mentioned above, the degree distribution was one of the first features that was identified as distinguishing real-world networks from random ones. Real-world networks often exhibit a much broader distribution of node degrees, so called "fat-tailed" degree distributions, a class of functions in which the "scale-free" or "power-law" distributions are of special popularity. These networks are characterized by a small number of nodes called "hubs" that are very strongly connected and therefore play a special role in the structure and the dynamical features of the network. For example, it was shown for scale-free networks that the average distance between any two nodes is hardly affected by random node removal, while it is very sensitive to removal of the hubs [77].

In the applications on biological networks, the investigation of the degree distribution has proven especially useful in the study of protein-protein interaction networks [78]. It was shown that there is a positive correlation between the degree of a protein and the lethality when this protein is knocked-out [79]. In a topological study of the transcriptional network

of yeast, it was shown that while the in-degrees of the nodes are exponentially distributed, the out-degrees are in fact distributed as a power-law [80]. Also correlations of node degree have been shown to play a role and in protein networks, hubs are unlikely to be connected to other hubs, which increases network robustness by localizing perturbations caused by deletions [81].

Other local structural features are the cluster coefficient (ratio of triangular linked triples of nodes to all triples of nodes), which gives a measure of the locality property of the links ("friends of my friends are also my friends"). Recently, generalizations of this concept have been suggested ("motifs") and again it was found that real-world networks show significant differences to ensembles of random networks [30, 31]. In chapter 8 the concept of network motifs will be extensively discussed.

On a larger scale, properties concerning the topology of the network come into play. For biological systems, especially the concept of modularity has proven to be fruitful [82]. Modularity or "community structure" is defined as heterogeneity of linking patterns within the network. A group of densely interconnected nodes is defined as a "module" or "community" if it has significantly more links that lie within the group than links coming in or going out of the group. A number of different definitions of modularity and algorithms for the identification of community structures have been developed [83]. Cellular networks typically display a high degree of modularity [84] and it has been shown that they can emerge from simple evolution procedures without selective pressure [85].

Other network properties include the distribution of loops and, on the largest scale, the diameter of a network, defined as the maximum of the shortest distance between any two nodes in the network [75]. The famous "small world" phenomenon, which was already mentioned in the beginning of this section, describes a logarithmic scaling of the diameter with system size, a property that can be found in a wide class of random networks.

Although all of these features can be used to distinguish real-world from random networks and to categorize them, structural properties describe networks in terms of their static configuration. What usually is of more interest to a physicist is the dynamics that can take place in a particular system.

3.2 Network dynamics and biology

The field of network dynamics has been quickly growing in recent years and a vast array of literature has emerged [75, 76]. This section comprises only a brief overview of network dynamics that have been employed as models for regulatory systems. Moving from very detailed and precise modeling approaches to extremely simplified dynamics, a range of different scales will be covered.

Even the most detailed of the commonly used models are strongly coarse-grained as the detailed physical and chemical simulation of all molecules as a many-body system is simply intractable. Therefore, averaging molecular behavior over a certain process-relevant space is necessary and the individual state variables of the molecular components are reduced to an absolute minimum [86].

In the class of models using reaction kinetics, the number of molecules of each species present in the system is modeled under the assumptions of homogeneous mixing of all molecules and instantaneous reactions. These again are simplifications, considering the compartmentalization of cells and the finite process times involved, for example, in transcription or diffusion. Taking these simplifications into account, a Master equation for the probabilities of n_i molecules of each species i being present at some time t can be written down.

$$\frac{dP(\{n_i\};t)}{dt} = \sum_{\{\tilde{n}_i\}} T_{\{\tilde{n}_i\}\to\{n_i\}} P(\{\tilde{n}_i\};t) - T_{\{n_i\}\to\{\tilde{n}_i\}} P(\{n_i\};t) \tag{3.1}$$

where $\{n_i\} = \{n_1, n_2, \ldots, n_k\}$ denotes a set comprising the numbers of molecules of each species and $P(\{n_i\};t)$ gives the probability of the system to be in this particular configuration of occupation numbers. The sum runs over all possible configurations and $T_{\{\tilde{n}_i\}\to\{n_i\}}$ is the rate of transition from one set $\{n_i\}$ to a different set $\{\tilde{n}_i\}$.

For each set of occupation numbers, one such differential equation is written down, which leads to a tremendous number of coupled differential equations even for small systems. This set of equations can thus be practically intractable, but obtaining the steady state distribution is possible as the equations then reduce to coupled linear equations.

If one is interested in the transient behavior of the system, one can (numerically) integrate the Master equation to obtain trajectories of the probability functions. In practice, however, often Monte Carlo methods are used to create statistically exact sample trajectories of the system states [86]. Given a system state $\{n_i\}$ at time t, the next transition to a different state is drawn from all possible transitions (according to the transition rates) and the time for this event to happen is drawn from an exponential distribution. To reach statistically relevant results, many of these sample trajectories have to be simulated, which amounts to high demands on computational power. To address this, optimization methods have been suggested [87], but the Monte Carlo approach remains suitable only for small systems containing a few different species. Successful examples of this stochastic modeling technique for gene regulatory networks are the lysis-lysogeny switch in phage λ [88] and the lactose uptake control network in *E. Coli* [89].

At the next level of coarse-graining, the molecule numbers are approximated by continuous values and the evolution of these functions is governed by differential equations. This modeling scheme is inherently deterministic but has the advantage of being somewhat more tractable, at least numerically, such that many transcriptional regulatory systems have been modeled using this method. If the number of molecules involved is large enough and stochastic deviations do not play a significant role, this modeling scheme leads to good results for ensemble-averaged behavior, like protein expression levels in a group of cells. Organisms that were modeled using differential equations include budding yeast [90, 91, 92] and fission yeast [93], but a plethora of other systems has been studied as well [94, 95]. To overcome the deterministic nature, differential equations can also be augmented with noise terms to describe the stochastic fluctuations in concentration levels [96, 97]. Also methods of bifurcation theory have been extended to stochastic systems [66].

At an even more coarse-grained level, discrete dynamical models have been devised to describe the qualitative behavior. As early as 1969, Kauffman proposed to use Boolean dynamics to describe gene activity [19]. Roughly at the same time, Thomas introduced a similar scheme [98]. Originally these models were introduced as a numerically tractable dynamical tool that incorporated most of the features of complex regulatory networks. However, there is evidence of switch-like behavior in regulatory system that also gives some justification for discrete modeling from the data [99].

In fact, although these qualitative models neglect many of the details of regulatory systems, successful descriptions of several biological organisms have been established, including the cell cycle of budding yeast [13] as well as fission yeast [21] and of mammals [100], the segment polarity network in embryogenesis of the fruit fly [11, 101], a development pathway in budding yeast [102], carcinogenesis [103], flower development [14, 104], and the regulation of the metabolism in bacteria [105]. All of these studies have used Boolean or multi-state discrete models. As we adopt this modeling technique in this thesis, we will give a more detailed introduction into the foundations and previous results on Boolean networks in section 4.1.

In order to combine the advantages of the differential equation approach with the discrete models, also hybrid models have been used [106, 107, 108, 109]. To end this roundup of modeling techniques (reviews can be found in [110, 111], we also have to mention Petri nets, a class of abstract modeling tools that can for example incorporate Boolean models but can also be extended to include more quantitative effects (for a recent review of Petri nets in biological modeling see [112]). At the most abstract level, even percolation dynamics can yield insight into the signal propagation in large-scale networks [113].

4 Boolean networks

In the results part of this thesis, Boolean dynamics will be used extensively. To base this on a firm footing, an introduction and review of this class of dynamical models will be given in this chapter. The general class of Boolean network models will be introduced and previous results on the dynamical features as well as on stochastic extensions will be described.

4.1 Definitions

As all of the results in this work are from the domain of discrete dynamical models, the first model of this kind should be given special attention. In a seminal work, Kauffman postulated that different cell types might relate to the different dynamical attractors of a particular kind of Boolean networks, as he numerically found a scaling of the number of attractors consistent with the number of different cell types in real-world organisms [19]. Although the original result later turned out be flawed, the lack of detailed knowledge about real gene regulatory networks and the simplicity and computational ease of this class of networks sparked considerable research interest.

A random Boolean network is defined as a set of N nodes that can assume only two different values (0 or 1, corresponding to "off" and "on", respectively). Each of these nodes has k_i inputs from other nodes and a rule table that gives the resulting value of the node corresponding to the input values. Assigning an initial condition to each of the nodes, the dynamics of this system can then be characterized by the following map:

$$s_i(t+1) = f_i(I_1(t), ..., I_{k_i}(t)), \tag{4.1}$$

where f_i stands for the particular Boolean function that is implemented by node i, I_j is the state s of the j-th input of node i and t is an integer number representing time. For now, a synchronous (or parallel) updating procedure will be assumed, which means that the next state of all nodes is determined before the time is advanced by one unit.

This defines a deterministic system and the set of node states $\mathbf{S} = \{s_1, \ldots, s_N\}$ (referred to as a "network state" in the following) can assume one of 2^N different configurations. As the dynamics progresses a network state eventually has to be revisited by the system, followed by an infinite repetition of all states between the two occurrences. This situation is called "attractor" in the language of dynamical systems, as this sequence of states cannot be left by the system. If a state leads directly to itself, it is called a "fixed point", otherwise the attractor performs a "limit cycle". The length of the limit cycle is simply given by the number of distinct states visited in the attractor.

All states that do not belong to an attractor are called "transient states" and for a particular initial condition, the "transient" is defined simply as all states visited by the system before the attractor is reached. For every attractor, an "attractor basin" can be defined, consisting of the set all states, which, if chosen as initial conditions of the system, lead the dynamics to this attractor. With the term "attractor basin", also the number of states in this set is meant. The configuration of states into basins of attraction is collectively called "attractor landscape."

In the original case introduced by Kauffman, each node has exactly two inputs from other nodes, each possible Boolean function (out of $2^{2^2} = 24$) is equally probable and the states of all nodes are updated in parallel.

4.2 Specific Boolean functions

The prescription to generate any one of the complete set of possible Boolean rules naturally also includes some obscure examples that do not seem suitable for the description of biological systems. As the logical rules in the promoter region of genes have to be implemented by exclusion or attraction among molecules, rules like the logical XOR are unlikely to appear in real-world genetic networks.

Therefore, alternative Boolean rules have been proposed for the modeling of transcriptional regulation. In this class of models, only a limited subset of all Boolean functions is used, whereas a lot other, some of which are simply too obscure to be reasonably implemented by molecular processes, are forbidden. One popular rule set that still allows a wide variety of Boolean functions is the class of "canalizing" functions [114, 115]. These rule are characterized by the occurrence of at least one "canalizing" input with a "canalizing" value. This means that if this specific input carries the specified value, the outcome of the node is fixed, irrespective of the value of any of the other possible inputs of the node.

A different sub-class of Boolean functions are the so-called "threshold" functions, originally studied by statistical physicists in the context of asymmetric diluted spin glasses and neural networks [25, 116, 117]. They have recently also been identified as a potential candidate for the modeling of gene regulatory networks [13, 21]. Threshold rules can be written in the following way:

$$s_i(t+1) = \Theta\left[\sum_j a_{ij} s_j(t) - \theta_i\right], \tag{4.2}$$

where a_{ij} denotes the connection matrix, with the value 0 if node j does not influence node i, and a weight of the connection otherwise. The value of θ_i is a threshold for node i and Θ represents the Heaviside function. Usually, the weights a_{ij} are set to $+1$ or -1 to represent activating or inhibiting interactions, but can also take any real value.

In this thesis, threshold Boolean functions will be extensively used as they provide a particularly simple choice of functions. They are well suited for the investigation of network

evolution, as the dynamical behavior of a network of Boolean threshold nodes only depends on features of the links.

4.3 Characterization of the dynamics

The class of all Boolean networks of a given size N and a fixed number of connections between the nodes is a very diverse set as can be illustrated with some simple examples: If every node implements a constant Boolean function independent of its inputs, every initial condition leads to the same fixed point. If every node has its own state as an input and the rule to copy its own state, every network state is a fixed point of the system. Apart from these pathological examples, a wealth of different structures of attractor landscapes is possible.

This wealth of dynamical behaviors makes analytical treatment quite difficult. However, for large random networks ($N \to \infty$) some characteristics of the ensemble of random networks can be calculated analytically.

One such property is the behavior under small perturbations of the network state, which can be studied by comparing the dynamics of some initial conditions with those obtained by the same network state except for a single node state change. If both dynamical sequences quickly converge and the perturbation thus dies out, the system is said to be in the "ordered" regime. If however the perturbation spreads across the system and, on average, half of all nodes behave differently in the two systems, the dynamics is called "chaotic". In the bordering case of a marginal spread of perturbation, the system is said to sit on the "edge of chaos" or on the "critical line".

To analytically determine this property, the so-called "annealed approximation" can be used, which neglects any correlations in the connections of the nodes [25]. It amounts to a situation where the inputs of all nodes are randomly shuffled at each time step and is exact in the limit of large systems up to times of order $\log N$ [118, 119].

For the original Kauffman network (in which each node has exactly the same number of inputs), the critical line can be obtained by a simple consideration: with probability 1/2 a differing state at one input affects the Boolean function in a node in such a way that the output value will be changed and on average such a perturbation will be spread to 2 other nodes. Thus, at a fixed connectivity of 2, a perturbation will just marginally stay in the system. For more general random networks in which the degree distribution can be characterized by the average connectivity $\langle k \rangle$, the condition for the critical line can be shown to be $\langle k \rangle = 2$ [120].

To characterize the role of different nodes in Boolean networks, one can define classes of dynamical behavior. In [121] nodes are called "stable" if they reach the same fixed state independently of the initial conditions. "Relevant" nodes as defined by [122] are those nodes that determine the dynamical behavior of the network. Nodes are irrelevant, if they are stable or do not influence any other nodes. One can then define a network decimation procedure that removes all those nodes that are irrelevant to reach the dynamical core of a network, i.e. a subset of nodes that has the same asymptotical behavior as the original

network.

The analytical treatment of the distributions of the attractor characteristics has proven to be quite difficult. In the annealed approximation, the distributions of transient and attractor lengths and basin sizes was determined by [122] but there it was also noticed by numerical computations that this approximation is only reliable if the dynamics is chaotic enough, thus resembling a random map.

The determination of the number of different attractors and attractor lengths on the critical line turned out to be a difficult problem and in contrast to the original assumption of a square-root scaling with the system size, it was only recently analytically determined for the special case of critical $\langle k \rangle = 1$ networks that both the number and the average length of attractors scales superpolynomially with system size [123, 124, 125].

Although seemingly much simpler than the original definition of Boolean networks, the sub-class of random threshold networks also displays a wealth of interesting dynamical behavior with similar features and an order/disorder phase transition [116, 117]. In contrast to networks with any random Boolean functions, in the case of threshold functions the number of inputs of a node affects the probability for a perturbation to be passed through. Using the annealed approximation, the critical average connectivity in the limit of large system size is found to be below 2, numerically estimated to be 1.849 ± 0.001 [126].

Real-world gene regulatory networks obviously are not infinitely large, so the question how finite system sizes affect the dynamical behavior is important to study. Recently the question of damage spreading and criticality has been addressed using finite size scaling methods. There, it was found that the critical connectivities for random Boolean networks as well as for random threshold networks are close to but slightly lower than the expectation from the annealed approximation (even in the limit of large system sizes) [127].

4.4 Asynchronous and stochastic extensions

The network dynamics described in the previous sections can generally be characterized by discrete (and finite) state variables and a synchronous updating scheme. Together with the deterministic update functions, this leads to deterministic dynamics. As we have discussed in section 2.3, biological dynamics are inherently noisy and thus call for a stochastic modeling approach. Several extensions of the simple Boolean model have been introduced, and here an overview of stochastic and asynchronous models shall be given.

The easiest concept, to allow random flips of the node states, has already been mentioned in the previous section. In modeling a concrete real-world network, oftentimes the full attractor landscape is obtained [13, 21] which already gives all information about the flow of network states. The size of the individual attractor basins and the probability to fall back into the same attractor after a node state perturbations are common measurements of the robustness of the dynamics. A related concept, Probabilistic Boolean networks, has been introduced for network inference [128]. Instead of fixing the Boolean rules of all nodes, uncertainty in the connections and the dynamics is implemented by defining multiple different networks and randomly choosing one to determine a requested node

update.

Even more than the deterministic nature of the node update rules, the synchronous timing seems an unbiological assumption given that no central clock ensures perfect synchrony of the individual components. It is well known that the details of the updating procedure in discrete systems can tremendously affect the dynamical behavior [129]. Also for the specific case of random Boolean networks and random threshold networks synchronous updating can lead to artifacts that do not stem from the orginal system under study but instead only from the algorithm used [130].

To overcome these problems, a number of asynchronous extensions of the Boolean model have been suggested. The most simple models keep the concept of discrete time steps and just use different updating schemes and are commonly known as "asynchronous random Boolean networks". In partial asynchronous updating, the order of the individual node updates is drawn randomly, but each node has to be updated once before the updating step is complete; in totally asynchronous updating, the next node to be updated is simply drawn randomly without further constraints [12]. In contrast to the synchronous systems, these asynchronous models are inherently stochastic which leads to problems with the definition of cyclic attractors - depending on a specific update order, the system may follow different paths. Thus, the term "loose attractor" was introduced [130] to denote a set of states that can never be left, independently of the order of updates. However, even in asynchronous dynamics, "pseudo-periodic" behavior can be observed which only requests a recurrence of a state after approximately a specified time frame [131].

A different route to asynchronous updating has been taken by [22]. There, small perturbations of the synchronous timing were investigated. In a similar model, it was shown that the number of reliable attractors, which are those synchronous attractors that are faithfully reproduced even under small perturbations of the signal times, scales sublinearly with system size, thus reconciling the number of attractors again with the original claim by Kauffman [46]. A similar result was also obtained in [132] from an analytical study of loose attractors in asynchronous random Boolean networks, where a power-law upper bound on the number of attractors was derived.

Another possibility of asynchronous modeling is to use different time scales for different processes happening in a network. A comparison of several such algorithms in the fruit fly segmentation network was conducted in [133]. One of these models, the so-called "Glass-type networks", originally introduced in [134], will be used in this thesis and explained in detail in section 5.2.

5 Reliability of network dynamics

In this chapter, the principle methods that are used in this thesis will be explained. Exemplified with the well-known phenomenon of oscillation of Hes1 protein in embryonic segmentation, a simplified description of transcriptional regulation dynamics will be motivated.

In section 5.2 the dynamical model that will be used in chapters 6, 8 and 10 will be described in detail. With this, the concept of reliable dynamics will be defined and small genetic circuits will be studied using this dynamical model. Simple conclusions on the interplay of topological and reliability features will be drawn.

As we will also utilize other algorithms that will be explained in detail in the respective chapters, we will give a brief overview and comparison of the different methods of reliability assessment used in this work at the end of this chapter.

5.1 From differential equations to discrete dynamics

As described in section 3.2, a number of modeling schemes has been developed that describe regulatory dynamics at different levels of detail. As this thesis is concerned with generic features of biological networks, an abstract description is chosen. In this section, using the example of the oscillation of Hes1 protein, this dynamics will be introduced and the applicability on a biological system will be shown.

In [135] the time evolution of hes1 mRNA and Hes1 protein, stimulated by serum treatment, was recorded and a two hour oscillation was found. The original plots of this work are shown on the left hand side figure 5.1. From the experimental results, the authors conclude that an indirect negative feedback of Hes1 protein on the transcription of hes1 mRNA exists. In [136] a simple set of differential equations was shown to reproduce this oscillatory behavior with a direct feedback using an explicit time delay. The differential equations for the concentration of mRNA and Hes1 protein with time delay t_d read:

$$\frac{d\mathrm{mRNA}}{dt} = \alpha \frac{k^h}{k^h + \mathrm{Hes1}(t-t_d)^h} - \frac{\mathrm{mRNA}(t)}{\tau_{\mathrm{rna}}} \tag{5.1}$$

$$\frac{d\mathrm{Hes1}}{dt} = \beta \mathrm{mRNA}(t) - \frac{\mathrm{Hes1}}{\tau_{\mathrm{hes1}}} \tag{5.2}$$

Here, τ is the life time of the mRNA and protein species, k is a typical concentration at the binding site, and h is the so-called "Hill"-factor which determines the shape of the response function. The explicit delay time t_d is the essential reason for the sustained

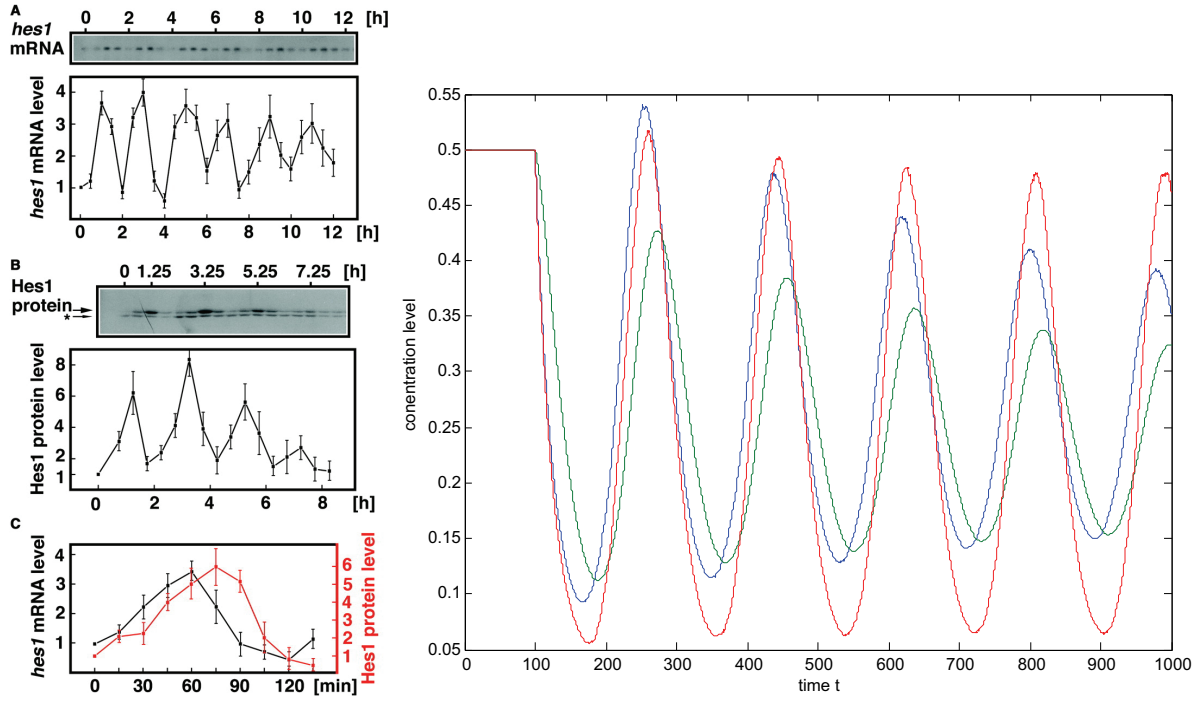

Figure 5.1: Left panel: Oscillation of hes1 mRNA and Hes1 protein level in cultured cells [135]. Right panel: Differential equation model of oscillations of hes1 mRNA (blue curve) and Hes1 protein (green curve) concentration levels. Red curve: simplified model of combined mRNA/protein description.

oscillations. In [136] it is shown that a realistic parameter set can lead to oscillations of the order of two hours, which is the experimentally observed value.

On the right hand side of figure 5.1 a reproduction of the results is shown. The numerical integration of this system of differential equations was performed using MATLAB and the forward Euler method with appropriate parameters (which do not coincide with the original parameter set reported in [136]. Hes1 protein concentration is given by the green line, mRNA level by the blue line.

Further simplification of this system can be applied here. By removing the second equation and replacing mRNA concentration by protein concentration in the first, the protein concentration dynamics can be tuned to closely resemble the original curve – this is shown by the red curve in figure 5.1. Thus, the minimal model that reproduces oscillatory behavior is actually a single component system with time delayed feedback. In this thesis, we thus use the conception of a combined mRNA/protein production cycle. The concentration level of Hes1 protein would then be determined by the following delay differential equation:

$$\frac{dc(t)}{dt} = \alpha \frac{k^h}{k^h + c(t - t_d)^h} - \frac{c(t)}{\tau_c}. \tag{5.3}$$

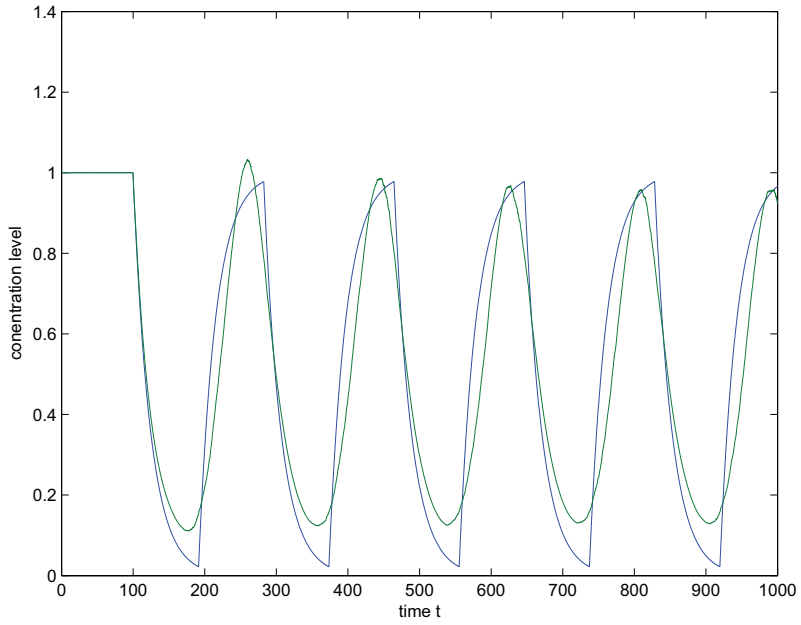

Figure 5.2: Comparison of differential-equation dynamics with Hill-type (green curve) and threshold (blue curve) input of the concentration level.

For networks of many interacting components, we further simplify the differential equations to allow for faster computation, by replacing the Hill-type dependence on the input protein by a simple step function. To do this, we simply use a Heaviside function to define a thresholded value of the concentration level,

$$s(t) = \Theta(c(t) - c_{\text{crit}}), \tag{5.4}$$

which will be called "activity state", as it gives a qualitative measure of a node's activity by comparison with a given threshold value. The value c_{crit} denotes the critical concentration (corresponding to k in the original function). Thus, in our example of Hes1 oscillation, the first term on the right hand side of equation 5.3 would be replaced by $\alpha(1 - s(t))$. Setting α to $1/\tau$ normalizes the concentration levels ($0 \leq c(t) \leq 1$).

We show a comparison of the Hill-type dynamics and the simplified dynamics from the linear differential equations in figure 5.2. One can see that the simplified description still leads to very similar oscillatory behavior. Although the details of the time evolution of the concentration levels have changed, this does not concern us here – the original data from [135] is too coarse to be used as an experimental test of the exact dynamical behavior.

5.2 Model description

Now that we have defined a dynamical rule that depends on discrete activity states, we can turn to the question how to incorporate multiple regulating nodes. In general, any Boolean rules could be used to combine several inputs. Instead, we limit the choice to threshold functions, which, despite their simplicity, have proven suitable as a description of regulatory systems [13, 21]. For our later purposes of running evolution procedures on networks, threshold functions are especially useful because of their inherent simplicity: no tinkering with specific Boolean functions is possible and the network architecture (along with the weights of the links) fully determines the dynamical behavior.

We have already mentioned threshold functions in section 4.2 and have given the explicit definition in equation (4.2). However, for our needs a slightly more general version is useful. We will define the transmission function as the result of the combination of all inputs at a specific node. One can write the transmission function for threshold dynamics in the following form:

$$f_i(t, t_d) = \Theta\left(\sum_{j=1}^{n} a_{ij} s_j(t - t_d) - \theta_i\right), \tag{5.5}$$

where n denotes the number of genes and a_{ij} is the adjacency matrix of the network of genes which for a particular connection (i,j) can take values of 0 (gene j does not influence gene i), 1 (excitatory) or -1 (inhibitory). The value of θ_i sets the threshold at which the summation of the inputs in the end leads to inhibition or activation of node i. The differential equation of the concentration level of a particular protein i now reads:

$$\frac{dc_i(t)}{dt} = \alpha_i f_i(t) - \frac{c_i(t)}{\tau_i}. \tag{5.6}$$

Plugging in the simple transmission function (5.5) (and again setting $\alpha_i = \tau_i$), we can easily solve this differential equation and get the following behavior for build-up and decay, respectively, of the concentration of protein i:

$$c_i(t > t_0) = \begin{cases} 1 - (1 - c_i(t_0)) \exp(-(t - t_0)/\tau_i) & f_i(t) \geq 0, \\ c_i(t_0) \exp(-(t - t_0)/\tau_i) & f_i(t) < 0. \end{cases} \tag{5.7}$$

Here, t_0 is the last time the input of gene i (represented by the transmission function f_i) switched, i.e. the start or end time of the production process of the protein.

We have simplified the equations only so far, as to be able to efficiently model a full network of interacting genes. We keep the characteristics of gradual concentration build-up and degradation and a transmission delay which leads to the observed oscillations. At the same time by being able to analytically solve this equation, we construct the concentration levels as a function of time by piecewise assemblage of buildup and degradation parts. This way, in our modeling we can work with continuous time and prevent the problem of having to integrate numerically. This dramatically improves the performance of the simulation and allows us to consider larger and more densely connected networks.

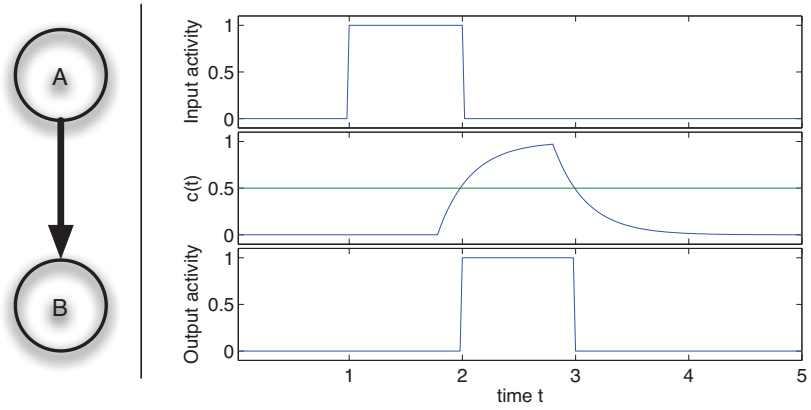

Figure 5.3: Concentration buildup and decay of a protein B given a specific input signal from a protein A and the corresponding activity state ($t_d = 0.8$, $\tau = 0.3$). In this simple example, protein B is activated by A and thus simply copies the state.

This dynamical model can also be seen as a simple continuous-time generalization of Boolean dynamics, if one identifies the thresholded concentration values that enter the input of a node with the discrete states of the Boolean system. The principle idea is pictured in figure 5.3. Here the internal dynamics and the resulting discrete 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 (say, externally) 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 1 at time step 1 and at time step 2 switches to state 0. Node B would react by switching to state 1 at step 2 and to 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. As soon as c_B crosses the threshold of 1/2, the activity state of B switches to 1. If one now identifies the Boolean state sequence with the sequence of activity states at integer times, one clearly finds that in the case of a delay time of unity and an infinitesimal charging time, both descriptions are exactly equivalent. If the parameters t_d and τ are set to the same values for all nodes, an analogous extension of Boolean network dynamics to continuous time is achieved.

Up to now we have only introduced a continuous, but still deterministic generalization of the synchronous Boolean model. If one now allows noise on the timing delay, the model becomes stochastic and asynchronous. The way we model this stochastic timing is via a signal mechanism. As soon as one node crosses its activity threshold, say, at a 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.

This defines aynchronous dynamics on the network of genes. In the noiseless case $\chi \to 0$ with infinitely fast concentration build-up and decay ($\tau \to 0$), we again exactly reproduce the case of a synchronously updated threshold network. Thus, by tuning the range in which the noise χ lies, we allow for a small effect that only slowly drives the system away from the ordered dynamics of synchronous updates.

With noise $\chi_{\max}$ and the filter τ set to a finite value, the switching times at the different genes are now allowed to differ (and diverge over time), so the network states at exactly integer time do not hold a special significance any more. To overcome this problem, we define a new "macro step" whenever all discrete node states (not necessarily the concentration levels) are constant for at least a time span of $t_d/2 + \tau$, which amounts to one discrete time step in the synchronous model. This determines a sequence of states, which can then be compared with the synchronous behavior.

This way, small fluctuations of the signal 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 synchronous state. We call network dynamics reliable if, despite the stochastic effect on the signal transmission times, the network follows the state sequence in a quasi-deterministic fashion. Although fluctuations in the exact timing is omnipresent, orderly behavior of the sequence of states can still be achieved. In the following sections we want to explore this asynchronous model by assessing the reliability of simple circuits.

5.3 Simple examples

In this section we want to show that the same dynamical sequence can be achieved both in a reliable or in an unreliable fashion, depending on the underlying network driving the dynamics. We start with the well-known example of two mutually activating genes. Apart from the trivial fixed points (both on or both off) this system displays an unreliable attractor as shown in the left panel of figure 5.4. In the upper part, the synchronous Boolean attractor is shown in a simple pictorial form (black stands for "on", white for "off" state). 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 figure 5.4), 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 obviously no longer fixed points but transient states to this attractor. As an example of the asynchronous dynamics we show in the lower part a run in which the ordered dynamical sequence can be seen.

With our continuous model, we now want to test the reliability of examples of circuits that can in principle be created artificially, namely the repressilator involving three and four genes. To more easily explain the main concepts we wish to introduce in this section, we

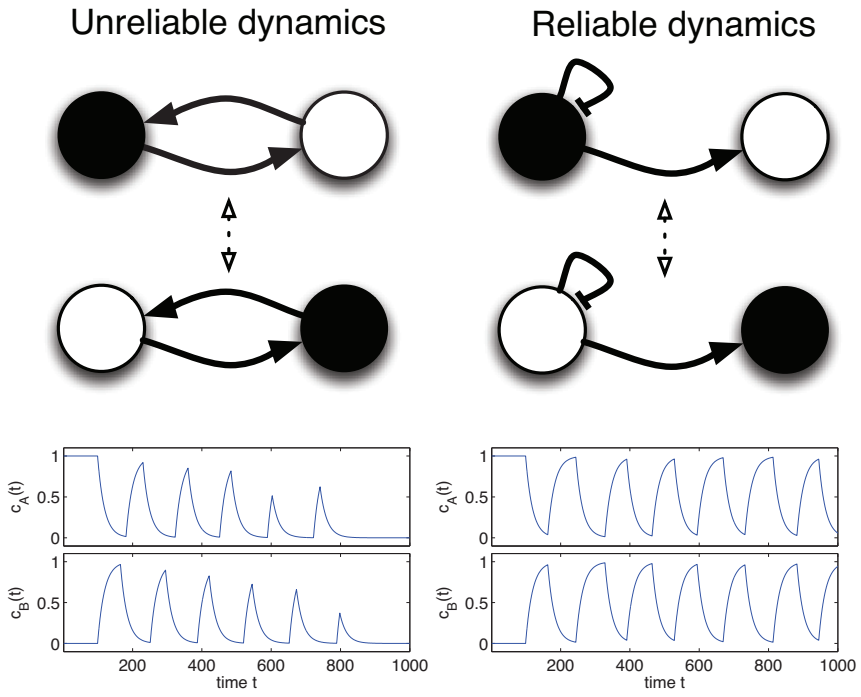

Figure 5.4: Comparison of two networks that have a common synchronous attractor. The mutually activating network is unreliable when subject to noise on the signal delay times. The self-inhibitor that activates a second node in contrast shows completely reliable dynamics.

choose parameters that improve readability of the figures. We stress that all our conclusions also hold for the parameters that we choose for the main simulation results in the results part of this work.

In the synchronous Boolean description, the three-gene repressilator exhibits two attractors which comprise all eight network states – the all-on-all-off (two states) and the signal-is-running-around pattern (six states). In the asynchronous scheme, independently 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 figure 5.5. This is due to the fact that only a single event is happening at the same time. This is depicted on the right hand side of figure 5.5 by the arrows which are successively active and show that no two events happen at the same time.

This picture changes dramatically in the case of the four-gene repressilator. The attractor structure is now much more involved, consisting of two fixed points (1010 and 0101), again the two-cycle all-on-all-off and three attractors with four states each. Using the stochastic scheme as before, we find that only the fixed points emerge as reliable attractors of the system. If any state of one of the 4-cycles is prepared as initial condition, the system thus

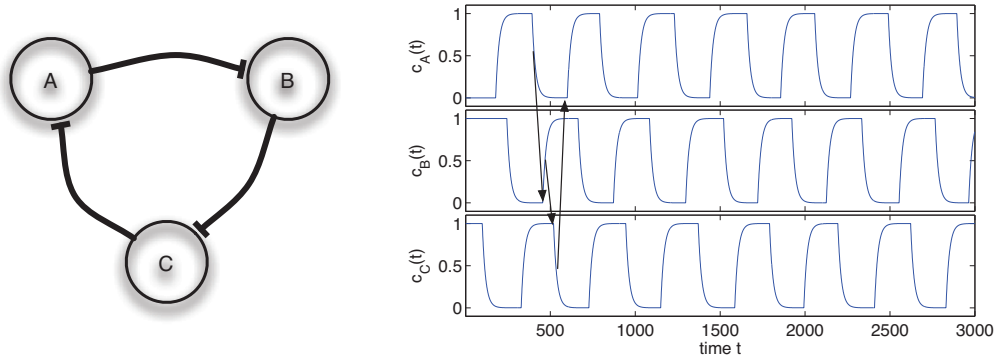

Figure 5.5: Wiring diagram and example time evolution of concentrations of all three internal variables of the three-gene repressilator. The dynamics is governed by a single event running around the circle – here depicted by arrows which denote the flow of signals.

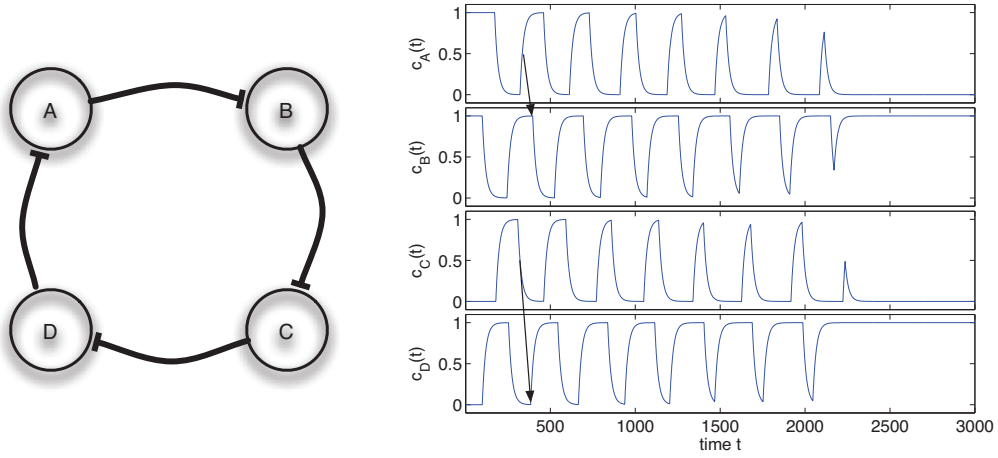

Figure 5.6: Wiring diagram and example time evolution of concentrations of all four internal variables of the four-gene repressilator. As two events happen independently at the same time (shown by the arrows depicting the signal events), the attractor can be left when the timing of the two event chains desynchronizes.

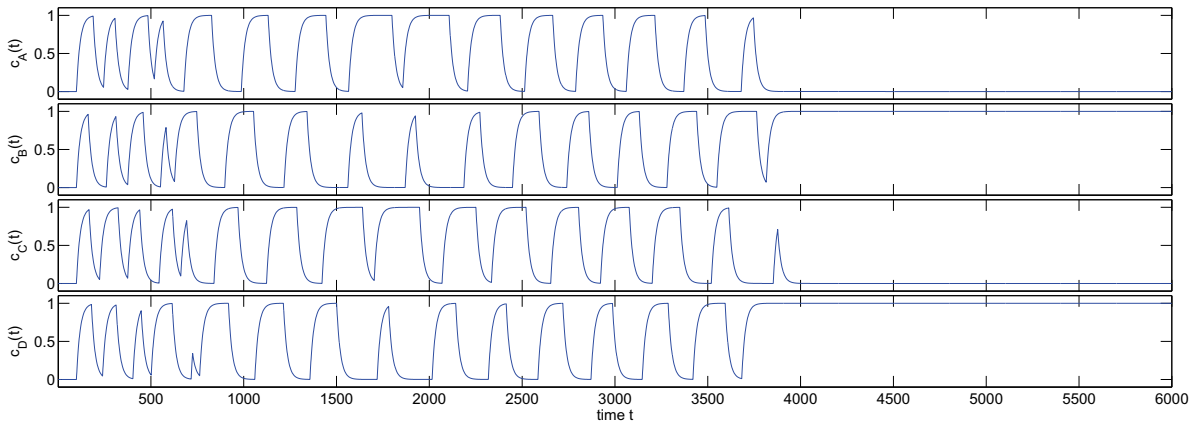

Figure 5.7: Example of an unstable, a marginally stable and a fixed point attractor in the four-gene repressilator.

always ends up in either one of the fixed points. In figure 5.6, an example run is shown which is initialized with the state 1100. 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 5.6. 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 move in time relative to each other, they can extinguish each other and drive the system into a fixed point. For a thorough discussion of the concept of multiple causal chains, also see [33].

5.4 Reliable and unreliable dynamics

Apart from the concept explained in the last section, when a system can accumulate a phase lag through small perturbations in a random walk-like fashion and eventually ends up in a different attractor, there are also examples of systems, in which any small perturbation directly causes the loss of the current attractor. This is the case if the perturbation itself leads to further desynchronization of the signals which can happen if the concentration variables do not reach their maximal value before their input nodes switch their state. This causes the following node to switch even earlier as compared to the unperturbed timing (in the limit $\tau \to \infty$, this behavior is impossible and each perturbation that is not removed from the system is merely "neutral").

We show this behavior in figure 5.7. Here again, the four repressilator is shown, but with the initial state configuration 0000. Without noise, this state belongs to the "all-on-all-off" attractor and four independent events are happening at each time step. The small stochastic asynchrony in the beginning is amplified due to the hysteresis effect and leads

to a very quick loss of the attractor. The system then enters an intermediate attractor where the "neutral" perturbation behavior is predominant, because the charging curves have more time to approach their maximal value.

The opposite behavior is also possible, that the 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.

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 switches occur for at least half the transmission delay time. This includes all marginally stable as well as unstable attractors. We do not distinguish between these in our results as both do not seem suitable for the reliability of a biological system.

5.5 Other models used in this thesis

The previous sections have given general insight into the concept of reliability that will be used throughout this thesis. In this work, different algorithms are used, that correspond to different levels of noise in the system. The main algorithm that will be used in chapters 8, 10, 6 was described in detail in this chapter. A different algorithm will be used in chapter 9 and will be explained in detail there.

If one considers only very small timing perturbations, one can implement a deterministic criterion, which was developed in [46]. The principle idea is that at small noise levels, the effect of a single signal time perturbation has to persist in the system over the full length of an attractor cycle of the synchronous system to actually be able to desynchronize the system gradually.

If one now starts at one particular step of a synchronous attractor and perturbs the signal event times of some of the switching nodes, two states at slightly different times are created. Then, using the synchronous updating prescription, the next time step at unperturbed and perturbed time can be computed. If the perturbation stays in the system, the dynamics is potentially unstable. If, however, the system regains synchrony and all switches happen at the same time later, this perturbation cannot significantly affect the system dynamics. The details of this algorithm are given in chapter 9.

6 Reliability of the budding yeast cell cycle

In this chapter, we want to investigate whether reliable dynamics is implemented in real regulatory networks. To do this, we apply our model of stochastic discrete dynamics to the well-characterized cell-cycle control network of the budding yeast *Saccharomyces cerevisiae*.

6.1 Background

The cell cycle is the process of cell growth and duplication. In eukaryotes, four main phases are distinguished: G_1 (growth and commitment to replication), S (DNA replication), G_2 ("gap" phase consisting of DNA repair and preparation for division) and M (mitosis or cell division). Progression from one stage to the next is controlled by a network of proteins that control gene expression. Thus, robust control of the cell cycle is required to maintain normal cellular function. As the molecular components of the eukaryotic cell cycle are highly conserved among different organisms [137], detailed studies of organisms such as budding yeast provide a relevant model for cell cycle regulation in multicellular organisms.

Approximately 800 genes (15% of all genes) in the budding yeast are regulated in response to the cell cycle [44, 45]. Fortunately, a much small number of regulators control the overall process. Several detailed differential equation models have been used to describe the dynamical behavior [90, 92], and an approximate agreement with experimental observation [91, 138] has been noted. For a recent review of cell cycle models, refer to [139].

In [13], the yeast cell cycle was modeled in the framework of a discrete threshold network. From the experimental data in [44] eleven proteins that play a key role in the cell cycle process were identified along with their known (direct or indirect) interactions. The activity of a certain protein was modeled as a two-state system, with values 1 (active) or 0 (inactive). Using synchronous, deterministic dynamics, the biological sequence of activity states in the process is exactly reproduced.

This is an astonishing result as the large number of network states and the corresponding vast number of combinatorial possibilities for attractors rule out coincidental agreement. On the other hand, it is clear that the discrete, synchronous framework is a rather crude approximation of the cell cycle dynamics. Processes governing gene regulatory networks take place on a molecular level and only small numbers of molecules are typically present in the cell, which calls for a stochastic description.

However, quantitative models such as the differential equation models mentioned above or, even more detailed, reaction kinetics (see section 3.2) have in common that a very

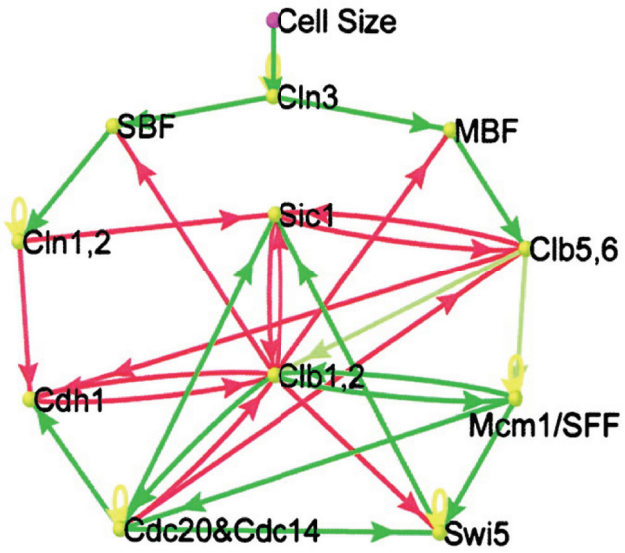

Figure 6.1: The simplified network of the core regulators in the yeast cell cycle from [13]. Yellow circles depict proteins, green arrows stand for activating, red arrows for inhibiting links. Small yellow self-connections denote that the respective node is modeled as a self-degrader. The Cell Size signal is an external signal to start the dynamical sequence.

detailed knowledge of the system in question is needed to determine all parameters of the equations. When considering whole networks of interacting genes and proteins and looking for system-wide properties and effects, models that focus on the network properties and neglect details of the individual components have proven useful [110, 18], such as the Boolean networks that we have used in this thesis.

The synchronous updating used in [13] for the yeast cell-cycle model is an artificial assumption which strongly affects the network dynamics. In particular, as was shown in [46], attractors under synchronous dynamics can be unstable if stochasticity is imposed on the transmission times. In this chapter we use the generalization of Boolean dynamics introduced in section 5.2 that allows for stochastic effects and investigate why the seemingly over-simplified framework of synchronous discrete dynamics is able to reproduce the basic activity pattern of the cell-cycle control network. We identify the mechanisms that ensure the robustness of the dynamics and thereby support the applicability of the simplified modeling.

6.2 Model description

Following [13], a network of eleven nodes is used to describe the cell cycle process and is depicted in figure 6.1. Table 6.1 reproduces the full synchronous sequence of activity

Time	Cln3	MBF	SBF	Cln1,2	Cdh1	Swi5	Cdc20/ Cdc14	Clb5,6	Sic1	Clb1,2	Mcm1/ SFF	Phase
1	**1**	0	0	0	1	0	0	0	1	0	0	Start
2	0	**1**	**1**	0	1	0	0	0	1	0	0	G_1
3	0	1	1	**1**	1	0	0	0	1	0	0	G_1
4	0	1	1	1	**0**	0	0	0	**0**	0	0	G_1
5	0	1	1	1	0	0	0	**1**	0	0	0	S
6	0	1	1	1	0	0	0	1	0	**1**	**1**	G_2
7	0	**0**	**0**	1	0	0	**1**	1	0	1	1	M
8	0	0	0	**0**	0	**1**	1	**0**	0	1	1	M
9	0	0	0	0	0	1	1	0	**1**	1	1	M
10	0	0	0	0	0	1	1	0	1	**0**	1	M
11	0	0	0	0	**1**	1	1	0	1	0	**0**	M
12	0	0	0	0	1	1	**0**	0	1	0	0	G_1
13	0	0	0	0	1	**0**	0	0	1	0	0	G_1

Table 6.1: The synchronous sequence of states as recorded in [13]. We color time steps in which only one switch occurs blue, and those with more than one switch red.

states that was obtained in [13]. Using the technique introduced in section 5.2 we extend that model to include fluctuating transmission delays and to allow for real numbers for protein concentrations levels ($0 \leq c_i(t) \leq 1$ for anyprotein i). We keep the characteristics of the description of [13], that is the effect of protein j on the transcription of protein i is determined by a discrete activity state ("active" or "inactive") of protein j. In our continuous description, we set the activity state S_i of a protein to 1 if the concentration is above a certain threshold ($c_i(t) > 0.5$), otherwise it is 0.

The transmission function that determines the build-up or degradation of the concentration of a protein i is given by

$$f_i(t, t_d) = \begin{cases} 1, & \sum_j a_{ij} s_j(t - t_d) > 0, \\ 0, & \sum_j a_{ij} s_j(t - t_d) < 0. \end{cases} \tag{6.1}$$

where t_d is the transmission delay time that comprises the time taken by processes such as translation or diffusion that cause an activity change of one protein to not immediately affect other proteins. The numbers a_{ij} determine 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, $a_{ij} = 0$.

If $\sum_j a_{ij} S_j(t - t_d) = 0$, the value of f_i depends on whether the node is modeled as a self-degrader. Self-degraders are those nodes that are down-regulated by external processes (Cln3, Cln1,2, Swi5, Cdc20/Cdc14, Mcm1/SFF). Self-degrader nodes will take a value $f_i(t, t_d) = 0$ whereas the transmission function of non-self-degraders is left unchanged, i.e. the last time $\tilde{t}$ when $f_i(\tilde{t}, t_d) \neq 0$ determines the state at time t.

We now describe the time evolution of the system of genes by delay differential equations

as given by equation (5.6). Again, in appendix A a technical description of the algorithm is given.

We now ask the following question: Is the original discrete sequence reliable under stochastic timing noise (stochastically varying signal delay times) or can the noise cause the system to assume different states? As the sequence from [13] runs into the stationary G_1 fixed point and an external signal is needed to trigger the starting state again, we create a repeating cycle of states by explicitly adding the rule that Cln3 production is triggered as soon as the final state in the synchronous sequence is reached. We will investigate whether this limit cycle is inherently reliable or whether it needs the perfect synchronization of the artificial synchronous update.

In this context it is important to note that the reliability of the complete cell-cycle system also depends on the behavior of all other proteins involved. However, the reliability of the core genes is most important, as they regulate the other proteins. Only if the regulators perform reliably, the system as a whole can be robust.

As described in section 5.2, we use a uniform distribution of noise values between 0 and $\chi_{\max}$ that perturb the delay times. For the investigation of this chapter this is important in order to be able to identify different regimes in the noise level. The uniform distribution has the advantage of being bounded, thus allowing to assess the maximal perturbation, but at the same time having a large variance. At large noise levels, the value of t_d can be neglected against $\chi_{\max}$ and the noise leads to a random order of events. As long as the distribution is bounded by the maximal noise parameter, the qualitative results do not depend on the distribution - the exact numbers given in the results section, however, would be different for a different choice of a distribution.

We emphasize that our model is not able to reproduce realistic time courses of protein concentrations. More elaborate models taking into account the explicit reaction kinetics would be needed for this (for example the Gillespie tau-leap algorithm [87]). However, we do not try to model the explicit time evolution, but are interested in the reliability of the state sequence of the synchronous model against timing fluctuations. We investigate this question on a purely theoretical basis using a simple and generic model.

Our model captures two principles of real world gene regulatory networks: Interactions occur with a characteristic time delay (denoted by t_d); and we use continuous concentration levels and implement low pass filter behavior due to protein concentration buildup with a characteristic time τ [135]. The time delay resembles the time steps of the synchronous model and does not have a significance in terms of the system size. Choosing $t_d = 1$ sets the simulation time scale.

6.3 Results

First, we check if the system reproduces the synchronous sequence under small perturbations of the delay time. Thus we stay in the regime where $\chi_{\max}$ is significantly smaller than the characteristic protein decay or buildup time τ. In the main simulation runs we set $t_d = 1$, $\tau = 0.3$ and $\chi_{\max} = 0.1$, but any numbers that fulfill $\chi_{\max} \ll t_d$ give the same

results.

We find that the synchronous sequence of states is reliably reproduced by this stochastic dynamics. Even long simulation runs of $t > 10^7$ cannot push the system out of the original attractor. This means that the biological sequence is absolutely reliable against small perturbations.

To understand this, we look at the synchronous sequence of states in table 6.1. In steps $2 \to 3$, $4 \to 5$, $8 \to 9$, $9 \to 10$, $11 \to 12$, $12 \to 13$ (marked blue in the table) only a single protein changes its activity state. If all steps were of this kind, fluctuations of the event times would not be able to destroy the attractor at all. States marked in red denote events where multiple switches happen at the same time.

To illustrate this point, let's assume two nodes switch their states at times t_1 and t_2 (we call this a "phase lag"). The system thus assumes an intermediate state in the time span between t_1 and t_2. Approximately at time $t_1 + t_d$ the next switches occur and due to the intermediate state it is possible that proteins switch their states which would normally be constant in this step. Because of the charging behavior of the concentration levels, these "spikes" will be filtered out. The only way to destroy the attractor is thus when the phase lag accumulates in a series of steps. This cannot happen in the yeast cycle, however, due to the states marked in blue color in the table. When only one protein changes its state in a time step, all divergence of signal times will be reset and the synchrony is restored. We therefore call these steps "catcher states" as they remove phase lags from the system.

Now that we know that small perturbations cannot drive the system out of the synchronous attractor, we want to investigate reliability under stronger noise. To address this question, we have to loosen our definition of reliability. Up to now, we have requested the system to follow the exact sequence of states of the synchronous dynamics. It is clear that this strict reliability cannot be obtained if we increase the noise to be more than half of the transmission delay itself, because two nodes switching at the same synchronous time step can receive switching times that differ by more than $t_d/2$. The intermediate step taken when only one node has switched obviously violates the reliability criterion.

To assess the reliability of the system under strong noise, we employ a different criterion. We let the system run with the sole constraint that the stationary G_1 state will be assumed regularly for a time span of at least t_d. Any fluctuations occurring inbetween two G_1 incidences will be tolerated, as long as the system finds its way to the G_1 state of the cell cycle in which growth occurs until the cell size signal is triggered. Although this might seem too loose a criterion for robust biological functioning, one has to remember that the cell-cycle process is also backed up by a system of checkpoints that can catch faulty system states. We investigate here the inherent reliability of the system disregarding these checkpoints but at the same time allowing more variability in the sequence.

Remarkably, with noise of the order of the delay time and largely independent of the filter used, the system reliably stays in the biological attractor. An example run with $t_d = 1$, $\tau = 0.3$ and $\chi = 0.9$ ran for a time of 10^7 following the biological attractor sequence (in the wider sense mentioned above). A typical time span of this run is shown in figure 6.2. This is a surprising result, because in general one expects a system to be able to leave its attractor sequence under such strong noise if a series of multi-switch events (steps 5 to

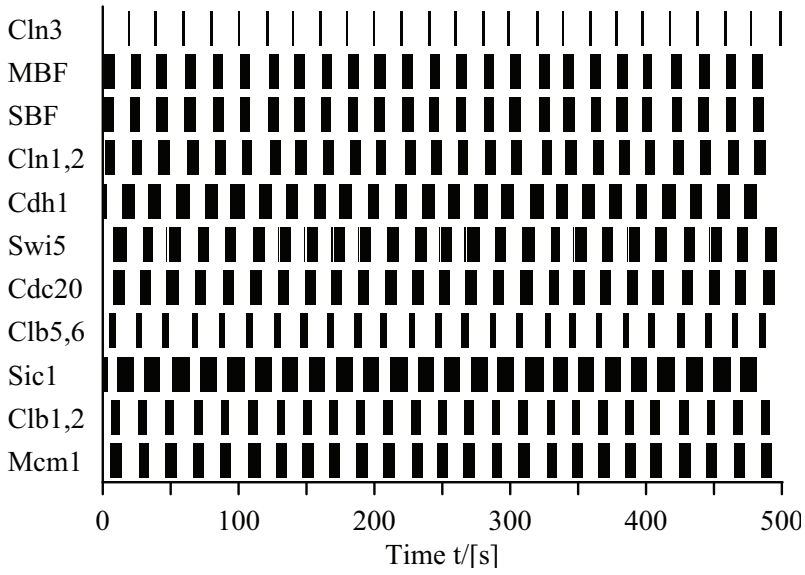

Figure 6.2: Time course of a run with noise of the order of the delay time $\chi_{\max} = 0.9\ t_d$. Black boxes denote active states, white means inactive. On a micro-time level the effect of fluctuations is visible, but on a larger time scale the dynamics is very stable.

8) is involved anywhere during the sequence.

Our proposed criterion is not trivially fulfilled: by changes in the sequence of switching events or by delaying one of several events that occur at the same synchronous time step, a new sequence could be triggered. This could force the system to jump into one of the other six fixed points identified in [13] without the possibility to return to the biological sequence. In figure 6.3 we show an example of a simulation run with extremely strong noise $\chi_{\max} = 3\ t_d$ that shows that the system can jump out of the attractor. However, it is also apparent that even under such strong fluctuations the system runs quite regularly until it finally loses its attractor sequence.

We now quantify the reliability of the biological pathway under such strong noise. How likely is it for the system to lose its biological sequence and to run into a different fixed point? To address this question, we initialize the system at the Start state again and check whether it completes the cycle. Again, we use the lose criterion described above, which means we only request the system to reach the Start state again. In figure 6.4 we show the ratio of erroneous runs of the biological pathway plotted against the noise level $\chi_{\max}$. It can be clearly seen that for reasonable noise levels the ratio of sequence runs not ending in a biological fixed point is very small. In fact, even with unrealistically high noise levels of $\chi_{\max} = 20$ or more (which amounts to arbitrary update times), only in a quarter of the runs the system jumps out of the biological state sequence. At such high noise levels, the homogeneity of the delay times is practically irrelevant. If we allowed a different time

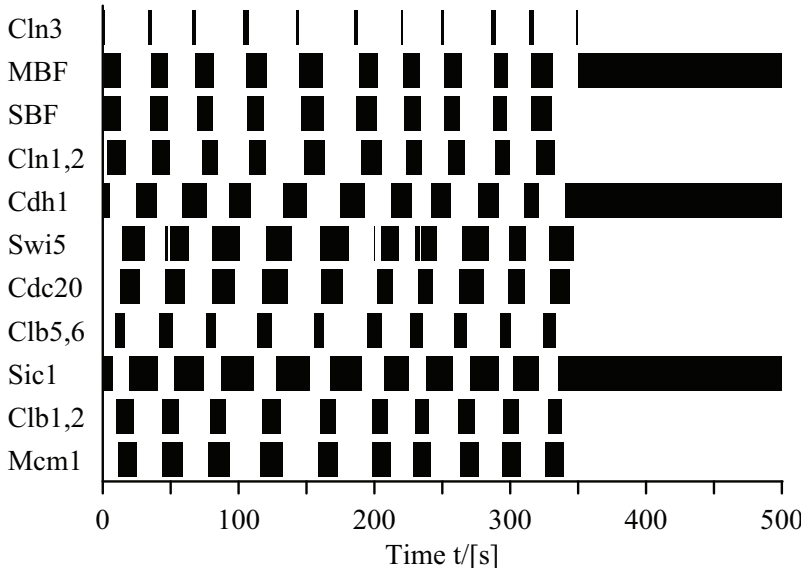

Figure 6.3: Time course of an example run with strong noise $\chi_{\max} = 3\ t_d$. After some repetitions of the biological state sequence the attractor cycle is lost and a fixed point is assumed.

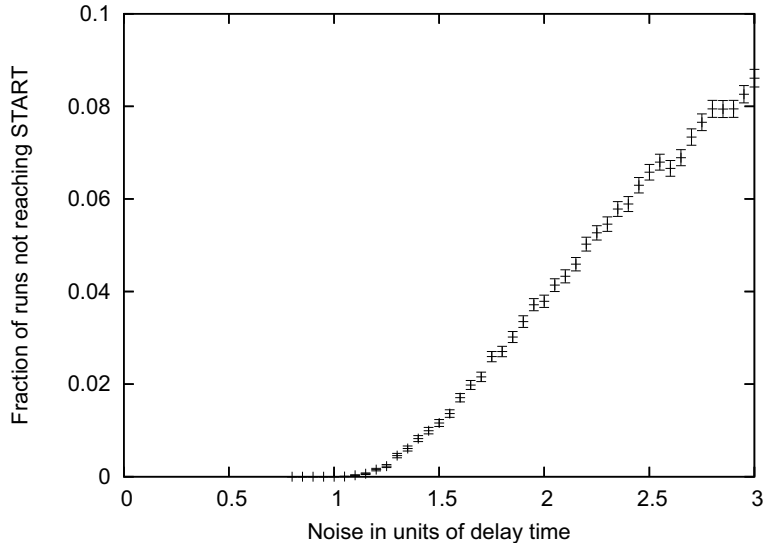

Figure 6.4: The ratio of runs escaping the biological limit cycle plotted against the maximal noise level $\chi_{\max}$. Even for strong noise, the fraction of erroneous runs is very small.

delay value for each node, the results of strong noise would be unchanged.

The by far dominating cause for this (very small) instability is the first step (cf. table 6.1) where both SBF and MBF are activated by Cln3. If the Cln3 concentration is degraded before activating the transcription of either SBF or MBF, the system loses the biological sequence. If we explicitly force Cln3 activity to sustain long enough to make sure that both SBF and MBF are produced, even this small instability vanishes and the system assumes practically complete reliability for all reasonable noise levels (0.1% erroneous runs at $\chi_{\max} = 3t_d$). This superstability is due to the fact that all proteins keep their activity states for an extended time. Extremely strong noise is therefore needed to delay a single activity switch long enough to significantly perturb the system.

We have tested all results with a wide variety of parameters. With a fixed number for the delay time t_d, only the noise level $\chi_{\max}$ and the characteristic protein buildup time τ can be adjusted. Our results are completely robust against changes of τ, even removing the filter completely or setting it an order of magnitude larger than the delay time does not affect the robustness properties described above.

6.4 Discussion

As we have shown in the previous section, the sequence of states as recorded in [13] is astonishingly reliable against fluctuations of the protein activation and degradation times. We have used a very simple model and have neglected details of the system such as different time scales for the different processes involved. Thus, it is not clear that our results translate directly to the biological system. However, we can rephrase our results in the following way: the robustness of the state sequence means that the control network completes its cycle even though the fluctuations destroy the synchrony of the model in [13]. This means that the sequence of activation/deactivation reactions is such that no coincidental synchronization of different processes is necessary for reliable dynamics. This holds strictly for the case of small noise and still surprisingly well for very large noise. This result is solely based on the sequence of states and one can expect to find similar results in more elaborate dynamical models.

The network and the resulting dynamics exhibit a number of features that cause this reliability: As was already discussed in [13], the basin of attraction is very large, making it unlikely that an intermediary state belongs to one of the other fixed point basins. A second remarkable property is that all node states are sustained for at least three (synchronous) steps, making the system less dependent on the specifics of the concentration buildup procedure. Third and most important for the observed superstability under noisy transmission times, is the presence of the catcher states which prevent the system from gradually running out of synchrony.

Thus, we have seen that without even taking into account the biological checkpoint mechanisms that give additional stability and error-correction features, the system shows a strong inherent robustness against intrinsic fluctuations. In this example of the yeast cell-cycle dynamics, potential mechanisms that provide robustness under biological noise can

be observed. A system without an external clock (or any other external control) can still run reliably if it has intrinsic features that enforce robustness: catcher states, persistence of states and an attractor landscapes that minimizes the possibilities to escape the biological sequence.

To conclude, we have investigated the reliability of the cell-cycle network by extending the model of Li et al. to allow asynchronous updating of the activity states of the genes. We find that the system exhibits robust behavior under noisy transmission times. Even without taking into account the checkpoint mechanisms that give additional stability and fallback features, the system shows a strong inherent robustness that aids in maintaining reliable functioning.

Recently, a different stochastic model of the yeast cell cycle has been investigated in [20]. There, the synchrony of the original model of [13] is retained, but nodes can erroneously switch with a probability that depends on the value of the transmission function and on a global parameter analagous to the temperature in physical systems. This defines ergodic dynamics and the steady state distribution of the probabilities of being in a specific network configuration can be computed. The biological pathway is shown to be significantly overrepresented. Also, in this work, a transition-like behavior depending on the "temperature" parameter is observed, where the order parameter is the probability to be in the biological G_1 state.

Another recent stochastic model investigates the efficiency of the cell-cycle checkpoints in a model using probabilistic Boolean networks [140]. In this work S and M phase checkpoints are added to the network of [13] and systematic perturbations of the cell cycle architecture are used to identify the fragile links of the network.

7 Evolution of networks in the computer

In the previous chapters we have investigated the reliability properties of small artificial circuits as well as of the control network of the yeast cell-cycle regulation. While simple properties can be identified that lead to reliable functioning, the question arises how real biological systems can actually acquire these properties through the process of evolution.

In the following chapters, we will investigate this question by employing computer models of evolution processes with selection criteria tailored for the search for reliable dynamical behavior. Evolution models in the computer have a long tradition that will be briefly surveyed here to put the results of this thesis in the relevant context.

The first attempt of an evolutionary procedures in the computer dates back as far as 1959 and considered mutation and selection of programming code [141], which however turned out to randomize the behavior and rendered an adaptive process impossible. While a large array of literature has since been devoted to evolutionary principles in computer science, this shall not be the subject here and the interested reader is referred to the excellent account of evolution in the computer across disciplines in [142].

Here, the focus lies on evolutionary models concerned with biological regulatory networks. The first such investigation of dynamical networks in [143] attempted to evolve Boolean networks towards a specific target expression pattern and a value is approached that is below the achievable optimum.

However, if the target function requires specific expression states only of a subset of all nodes, evolution procedures can be successful in reaching the exact required dynamical behavior. In several works, the investigation how such evolution is affected by noise from external sources [144], modular goals [145], and specific degree distributions [146] was conducted. In a recent work the relation between isogenic phenotypic variation and variation due to genetic differences in a population was investigated [147]

In contrast to these models, the following chapters will not be concerned with evolution towards a predefined target behavior but instead ask whether networks can be evolved towards robust functioning.

Different notions of robustness have been used in the context of network dynamics that have to be briefly surveyed in order to avoid confusion: mutational robustness denotes the ability of a system to function correctly despite changes of the structural properties; robustness against variations of the initial states denotes the feature that a specific behavior is obtained despite changes of the node states with which the systems dynamics is started.

Another robustness property was discussed in [38, 39]. There, a mutated network in the evolution replaces the mother network if it follows along the exact same sequence of states

until an attractor is reached. Thus, this criterion selects for the continuity in expression patterns under randomly changing environmental conditions.

Mutational robustness was investigated in a threshold network model in [40]. In a population of networks that are mutated and selected for the reproduction of a predefined behavior, it was found that after many such evolution steps the effects of structural mutations are significantly reduced.

In a recent investigation of a model incorporating gene duplication and subsequent divergence (by rewiring of links and change of Boolean rules) it was found that networks can be robust against structural changes (attractors are left unchanged) and at the same time possess the ability to acquire new attractors [41]. In two very recent papers, also the question of robustness against variations of the initial states was investigated and it was found that in evolution processes networks can rapidly acquire this property [42, 43].

In the investigations of network evolution, quite different mutation prescriptions can be employed. The conceptually most simple version is to allow modifications of the network structure by removal, addition or rewiring of a link. In transcriptional regulation, this amounts to the creation or loss of a binding site at a gene's promoter region or the modification of the sequence or the structure of a protein that causes it to influence a different gene.

It is well known that genetic networks can also evolve by gene duplications and subsequent divergence of the two gene copies [148, 149]. This kind of mutation prescription was used for example in [41].

A more elaborate simulation technique of evolving networks in the computer is that of the artificial genome [150, 151, 35]. In this class of models, random bit strings (or symbols out of a small set) represent the genome and specific bit sequences are defined as markers for promoter regions and coding regions of genes. Then a dynamical model can be constructed using rules connecting the different genes.

In the evolution processes of this thesis, the simple version of link rewiring will be used exclusively as a model of the mutations in genes. It has the main advantages that the number of genes in the model and the corresponding space of gene activity states is fixed and that the topology of networks resulting from such mutations is indistinguishable from random networks. Thus, non-random structural features can be fully attributed to the employed selection process.

8 Motif distributions resulting from evolution

As we have mentioned in section 3.1, with the increasing availability of large-scale data, the investigation of the structures of interactions in biological systems has received growing interest in the last few years. One of the fields of investigation is the topological structure of genetic regulatory networks.

In [29, 30, 31] the topology of different classes of networks was investigated by looking at the local property of the so-called "motifs". In [31] it was found that real-world networks can be categorized into "superfamilies" according to their signature in the distribution of specific linking patterns among connected triads of nodes. One of these superfamilies consists of signal-transduction networks including transcription networks.

Some of the triads can be identified to play significant roles in these systems. For example, the common feed-forward loops have been shown to conduct persistence detection and sign-dependent delay [29, 152], which can aid in the attenuation of molecular fluctuations.

Several studies have tried to explain the motif structures of real-world networks by examining robustness properties of the isolated triad structures. In [34] a correlation between the structural robustness of the motifs (robustness of the steady state in a differential equation model against parameter variations) and their abundance in real-world networks was observed. Also, synchronization properties have been investigated and a similar correlation of the stability of the synchronous state with some real-world network motifs was observed [37]. In [33] the dynamical stability of the individual triads was investigated. It was shown that those triads which are enhanced in natural networks, perform well in an appropriate robustness test.

From a different perspective one can ask whether evolutionary dynamics can reproduce realistic motif structures if the essential selection mechanisms are implemented in the model. In [36] an evolution model with simple flow dynamics was investigated. By selecting for a desired flow output and robustness against structural perturbations of the networks, it was shown that, depending on the selection criterion, the resulting networks can be assigned to one of the superfamilies. On the other hand, there are also approaches to explain motif distribution from structural effects of the evolutionary process of duplication-divergence [35] or of the large-scale topological organization of a network [32].

Here, we ask whether the requirement of reliable dynamics can lead to the observed motif distributions of biological regulatory networks. Unreliable triads should be suppressed in an evolutionary procedure as they do not seem to be suitable building blocks for reliable network dynamics. Thus, we develop a simple evolutionary model where networks are selected for reliable functioning. We use the continuous-time generalization of Boolean

dynamics to test the reliability of the network dynamics against small fluctuations of the signal transmission times. We ask whether this fitness criterion leads to the observed motif distributions in nature.

8.1 Model description

The dynamical model that will be used in this chapter has been described in section 5.2. We implement the dynamics defined by equation (5.6) with stochastically varying transmission times. An explicit description of the algorithm can be found in appendix A.

The transmission function is given by the threshold rule (5.5), which only leaves the freedom to set the threshold value. Here, we will implement a simple majority vote with positive and negative coupling, $\theta_i = \sum_j a_{ij}/2$, where a_{ij} again denotes the adjacency matrix of the network.

Now, we define a fitness score which allows us to choose whether an evolutionary step will be accepted or discarded. The "synchronous dynamics" is defined as the sequence of states the network visits without noise on the transmission delays and with vanishing build-up and decay rate ($\tau \to 0$).

If noise and filter characteristics are now turned on, the asynchronous dynamics can be explicitly modeled. Then the fitness score is simply given by the number of macro steps the synchronous sequence was correctly followed by the asynchronous dynamics, divided by the maximal number of macro steps, which is which is set to 500 (we will motivate this explicit choice later on).

In the case of fixed point attractors, we need to define a fitness value, as there are no switching events as soon as the fixed point is reached. We could either assign them a value of 1 if they reach the fixed points also in the noisy dynamics or abort the dynamics and just count the steps until the fixed points is reached. We have tried both possibilities, but unless explicitly stated otherwise, we have used the latter possibility and counted fixed points as unstable attractors. This prevents networks from building up a structure devoid of loops.

Until now we have defined the fitness measure for one network and one particular set of initial conditions (the exact choice of which obviously has a strong impact on the dynamics). However, real-world regulatory networks have to respond to external stimuli and we represent this in our model as different initial conditions. At the beginning of each evolutionary procedure, we therefore create 100 sets of initial states that stand for typical environmental conditions in which the network has to function correctly. To obtain the fitness value of a network during the evolution, we average over the 100 fitness values obtained from each run corresponding to one set of initial states. As a model variation, we will also examine varying initial conditions, that is creating new sets of states at each mutation round.

With our fitness measure defined, we can finally come to the description of the evolutionary process taking place on the space of possible network configurations. We start with a random (Poissonian) network of a given mean degree. Both in the creation process of the

random network, as well as in the mutations afterwards, we disallow any self-links. This is to allow for better comparison with the data from [31].

Every evolution step now consists of the following procedure:

1. Rewire the network by randomly removing one link, picking two nodes that are not already connected and putting a new link (with weight +1 or -1) between them.

2. Let the dynamics run on both networks (once for each initial configuration) and calculate the network's fitness by averaging over the fitness values of the individual runs.

3. If the mutation is fitter, keep it and advance to the next evolutionary step, otherwise repeat the steps with the original network.

We have chosen a simple rewiring technique in order to be least biased away from random network structure. It has been shown in [35] that network motifs may also result from a duplication-divergence process without functional constraints. As we want to test whether a functional constraint can lead to non-random motif distributions, we specifically choose a rewiring procedure that is biased towards random networks.

The evolution continues until a fixed number of evolutionary steps is performed or a fitness larger than a defined critical fitness value is reached. In the results below, we set the maximal number of evolution steps to 1000 and discard an evolution run if the evolution is not successfully finished until this number is reached (however, this practically never happens). The critical fitness value that determines a successful run is set to 0.97. As we check the network's stability only for a comparatively small subset of all possible input states, it is not necessary to demand robustness for every of the selected initial states. Setting the required fitness score to a high number as we have done, already ensures that the resulting network is robust under most (but usually not under all) initial states. We have tested using a necessary score 1.0, which tremendously increases the run time but does not significantly alter the results.

The structure analysis is based on the method presented in [30]. The network to be analyzed is rewired in a way that preserves the number of incoming and outgoing links at each node. Two links connecting nodes A and B as well as C and D are rewired to connect A and D, C and B. We use 100 rewiring steps to create a randomized network and create #ensemble = 1000 such networks (starting the randomization process with the last randomized network).

The Z-score is then given by the deviation of the analyzed network's triad numbers from the average of the randomized networks, divided by the respective standard deviation in the ensemble.

$$Z_i = \frac{N_i - < N_{i,\mathrm{rand}} >}{\sigma_{N_{i,\mathrm{rand}}}} \tag{8.1}$$

If the standard deviation of the ensemble is 0 (which can easily happen for example in the case of the fully connected triad), we set it to $1/\sqrt{\#\mathrm{ensemble}}$. Contrary to the original method, we have not normalized the Z-score. In the original work the vector of Z-scores

Triad number	symmetry factor S_i	# of links k_i in triad	Scaling of expectation value with system size	Expectation value for $N = 16, \langle k \rangle = 3$
 1	3	2	$\mathcal{O}(N)$	27.8
 2	3	2	$\mathcal{O}(N)$	27.8
 3	6	2	$\mathcal{O}(N)$	55.6
 4	6	3	$\mathcal{O}(1)$	13.5
 5	6	3	$\mathcal{O}(1)$	13.5
 6	3	4	$\mathcal{O}(1/N)$	1.6
 7	6	3	$\mathcal{O}(1)$	13.5
 8	2	3	$\mathcal{O}(1)$	4.5
 9	3	4	$\mathcal{O}(1/N)$	1.6
 10	3	4	$\mathcal{O}(1/N)$	1.6
 11	6	4	$\mathcal{O}(1/N)$	3.2
 12	6	5	$\mathcal{O}(1/N^2)$	0.74
 13	1	6	$\mathcal{O}(1/N^3)$	0.03

Table 8.1: Overview over numerical characteristics of each triad. For detailed explanations of the columns see text.

is normalized to length 1 which leads to the fact that the overall significance of deviation cannot be seen in the resulting numbers. This was done to take care of effects of different system sizes of various networks. If left unnormalized, the Z-score gives the significance of the difference from the expectation in terms of multiplicities of the ensemble's standard deviation. In all results below, the absolute significance of the deviations are shown to allow for a better comparison with random structures.

On the other hand, it is also possible to calculate the expectation value of each of the triads in a random graph. One simply has to determine the multiplicity (i.e. symmetry factor) of each triad and the number of links in the triad.

For the multiplicity, one has to determine whether cyclic or pairwise relabeling of the nodes leads to a different realization of the same motif. Starting with a symmetry factor of unity, one has to multiply with three if a cyclic relabeling leads to a different realization of the same motif, and with two for pairwise relabeling. The resulting symmetry factor for each of the triads is given in table 8.1.

The expectation value for each triad is given by the following combinatorial expression:

$$\langle \#\text{triads}_i \rangle = S_i \binom{N}{3} \frac{K!}{(K-k_i)!} \frac{(M-K)!}{(M-K-6+k_i)!} \frac{(M-6)!}{M!}, \tag{8.2}$$

where N is the number of nodes in the network, K is the number of links in the network, k_i is the number of links in the triad and $M = N(N-1)/2$ is the number of possible links. To estimate the dependence of the motifs on the system size and the connectivity $\langle k \rangle = K/N$, one can write down a simplified expression under the assumptions $k_i, \langle k \rangle \ll N$:

$$\langle \#\text{triads}_i \rangle \sim S_i \frac{N^3}{6} \frac{K^{k_i}(M-K)^{6-k_i}}{M^6} \sim \frac{S_i}{6} N^{3-k_i} \langle k \rangle^{k_i}. \tag{8.3}$$

Thus, one can see that the occurrence number of all triads increases with $\langle k \rangle$ (this holds as long as $\langle k \rangle \ll N$). On the other hand, the dependence on system size crucially depends on the number of links involved in the triad, k_i. For the two-link triads, the occurrence number rises with system size, for three-link triads its constant, whereas for higher number triads, it is suppressed with system size. The scaling behavior with system size is also given in table 8.1 along with example values for the expectation values of the triads in a random network of 16 nodes with connectivity $\langle k \rangle = 3$. One can see that the densely connected triads occur very seldomly in sparsely connected random networks.

8.2 Results

As we have altered the set of Boolean functions that we use for the dynamics of the nodes, we first determine the reliability scores of the individual triads as was done for canalizing Boolean functions in [33]. Therefore, we also use the same criteria for reliability: the triad reliability score is defined as the probability that for a given triad a random selection of link weights and a random assignment of initial states leads to reliable network dynamics. The strict triad reliability score is defined as the probability that a random selection of link weights leads to reliable dynamics for all possible initial states. These criteria are slightly different from the criterion that we will use later on. In the reliability assessment of the isolated triads the number of runs that correctly reproduce the dynamical sequence up to the maximal macro step count is obtained. In the criterion explained in the previous section, which will also be used in the evolution process, also unreliable runs contribute to the fitness score according to the number of steps they ran orderly. In the limit of infinite macro steps, both criteria are identical.

We show the triad reliability and strict triad reliability scores in figure 8.1, together with the Z scores of [31]. Apart from minor differences for the motifs 12 and 13, the reliability scores closely resemble those found in [33]. Thus, also for threshold dynamics the original claim holds: significant correlations between the reliability of isolated triads and their abundance in real-world networks can be seen.

We also used the assessment of the isolated triads to fix the parameter value of the maximal number of macro steps to be used in the evolution procedure. By applying the

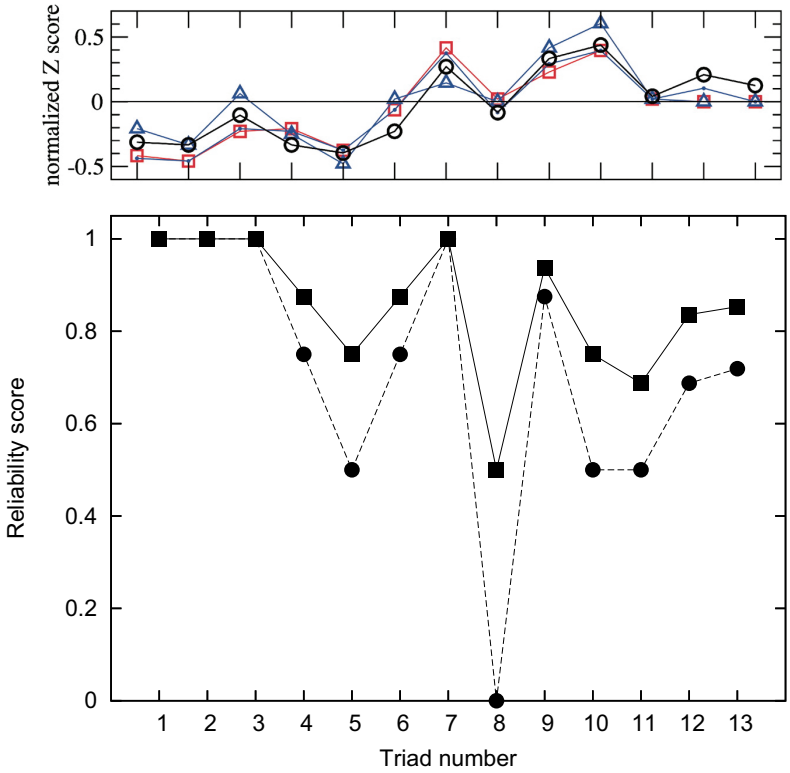

Figure 8.1: Comparison of the triad significance profiles of real-world signal transduction networks with the reliability scores of isolated triads. Upper panel: Normalized Z score of four signal transduction networks as obtained in [31]. Lower panel: triad reliability score (squares) and strict triad reliability score (circles) in the threshold model. See text for the definitions of the reliability scores.

fitness criterion described in the previous section for different values of the parameter, we determined that 500 steps suffice to obtain very similar results to the fitness scores obtained in figure 8.1. Higher values do not significantly alter this score. We therefore use this value in the evolution process. We have also tested to increase this value to 2000 steps in the evolution runs and did not find significant effects.

Next, we want to gain some insight into how reliable random networks perform in our criterion. To do this, we have randomly created 4000 networks with 32 nodes and a connectivity of 2 and determine the reliability score of each by averaging over 100 random initial conditions. The average fitness obtained is 0.344 ± 0.006, illustrating that random networks are typically rather unreliable in their performance. We also plot the cumulative distribution of reliability scores in figure 8.2. One can see that a large fraction of about one-half of all networks is very unreliable with scores of less than 0.1. The other half of all networks is practically uniformly distributed between a reliability score of 0.1 and 0.99 (which is depicted by a straight line in the cumulative plot). Another one-tenth of

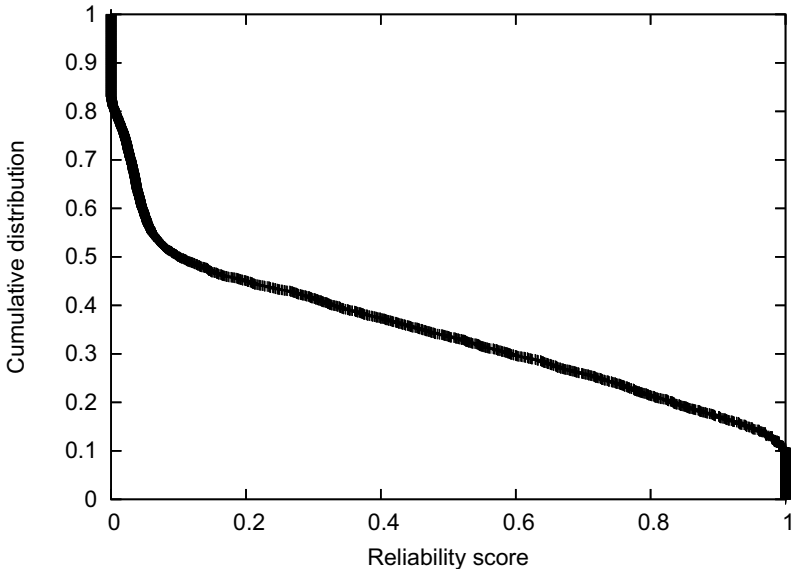

Figure 8.2: Cumulative distribution of reliability scores of random networks. 32 nodes with connectivity 2.

all random networks is perfectly reliable. Thus, one can see that the reliability score of random networks allows for a significant improvement through an evolution process.

First we used our evolution procedure on single networks to see if it forces them into a motif structure resembling the experimental data. To address this, we checked various different fitness scores (counting fixed points as stable or not, using a fixed set of initial states or randomly choosing initial states at every evolution step) and tried to minimize the statistical errors in the motif distribution.

At a fixed connectivity $\langle k \rangle = K/N = pN$, the total expected number of those triads on which we expect our fitness criterion to be selective (all triads with numbers higher than four) decreases (or at best is constant) with system size. Thus, increasing the number of nodes does not lead to more triads of number four or above. Instead one would have to increase the connectivity. Unfortunately, this measure has a strongly enhancing effect on the complexity of the dynamics (number of non-frozen nodes, length of transients and attractor cycles) and therefore leads to a largely increased runtime of the evolutionary process. Also, the real-world networks investigated in [31] displayed connectivities between two and three.

For these reasons, we will investigate only small networks of 16 or 32 nodes with connectivities of two or three. From equation (8.2) we obtain that apart from the first three, all triads are expected to occur in quantities of only a few per network or even less. This means that large fluctuations in triad numbers are expected. However, one can still look for a systematic suppression or enhancement of certain triads.

Of all the different model variants, we exemplify the results on one specific case: a fixed

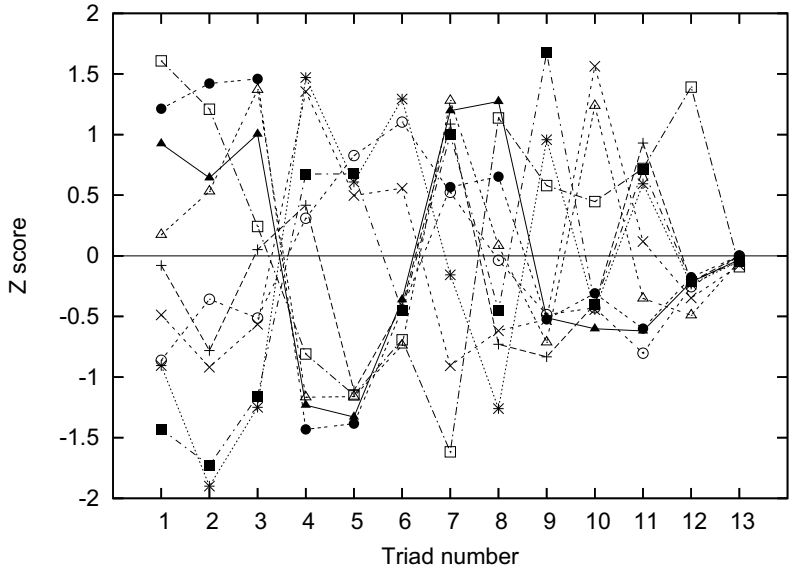

Figure 8.3: Abundance of triads in evolved networks as compared to a randomized ensemble. On the x-axis, the 13 different triads are listed, the y-axis shows the Z-scores of the individual networks for nine independent evolution runs. Fixed points are counted as unstable attractors. The networks consist of 16 nodes with connectivity 2.

set of initial states is used during the evolution procedure and fixed points are regarded as unstable attractors, which means that the networks have to evolve towards reliable limit cycles. We show the Z-score for nine different networks resulting from the evolution procedure in figure 8.3. The networks consist of $N = 16$ nodes with an average connectivity of $\langle k \rangle = 2$. The individual networks differ largely in their triad significance profiles and clearly one cannot state that a typical motif distribution emerges from the evolution. We thus conclude that an effect of the robustness features under noisy dynamics cannot be seen in the resulting motif structure of the individual networks. In addition to the results shown here, we have tested a large number of model variations (increasing the number of initial states, changing the initial states at every evolution steps, varying the parameter values). In no case a motif distribution resembling that found in biological signaling networks.

To quantify the general effect of the selection pressure towards reliable dynamics on the motif statistics of genetic networks, we next turned to the investigation of ensembles of evolved networks. To do this, we run the evolution procedure multiple times and take the average of the motif structures of the networks resulting from each evolution run. The following results thus do not give structure statistics of single networks but instead show an average over the ensemble of evolved networks. The error bars in all plots are determined by the standard deviation of the mean of the ensemble of Z-score values.

In figure 8.4 we present the results of the evolution process of networks with 32 nodes

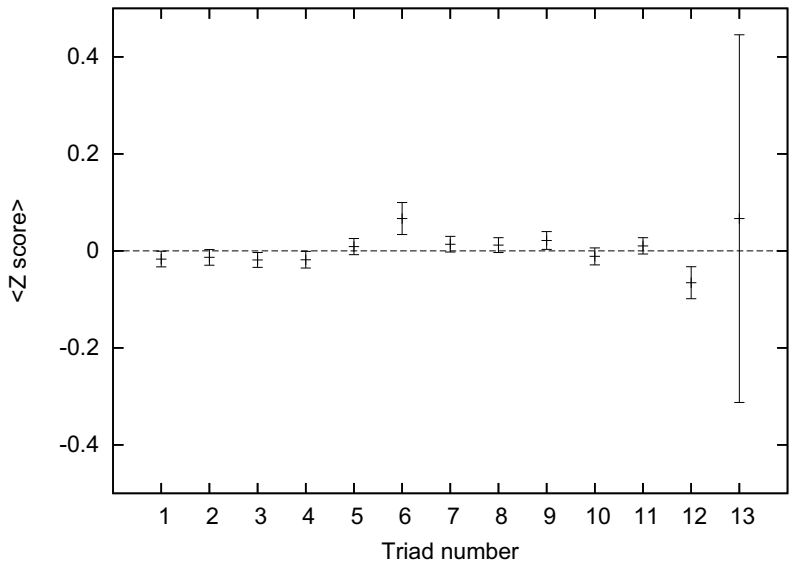

Figure 8.4: Abundance of triads in evolved networks as compared to a randomized ensemble. On the x-axis, the 13 different triads are listed, the y-axis shows the average over the Z-scores of the individual networks. Fixed points are counted as unstable. The networks consist of 32 nodes with connectivity 2. Average over 4000 evolution runs.

and a connectivity of 2. The averages were taken over 4000 individual evolution runs. As can be seen from the figure, no significant deviations from the null-line are observed – the "outliers" triads 6 and 12 are statistically expected. Very similar plots were also obtained for other values of the system size and of the connectivities. The large error bar on triad 13 stems from the fact that only few networks have enough mutual links to allow the possibility of creating this triad during the link reshuffling that is the basis of the Z-score determination. The average number of evolution steps in this simulation was 11.1 ± 0.2.

Next, we want to investigate whether non-random structures have emerged in the evolution that cannot be determined by the Z-score because of the randomization procedure that fixes the in- and out-degrees of all nodes. To do this, we use the expectation values of the triads as obtained from equation (8.2). The occurrence numbers of the triads in the evolved networks are simply divided by the theoretical expectation for random networks. The average of these quantities over the full set of evolved networks is shown in figure 8.5. The results are slightly different when compared to random networks than what was obtained in figure 8.4 for the randomized ensemble of fixed degree distribution. However, again no significant deviation compared to random networks can be seen.

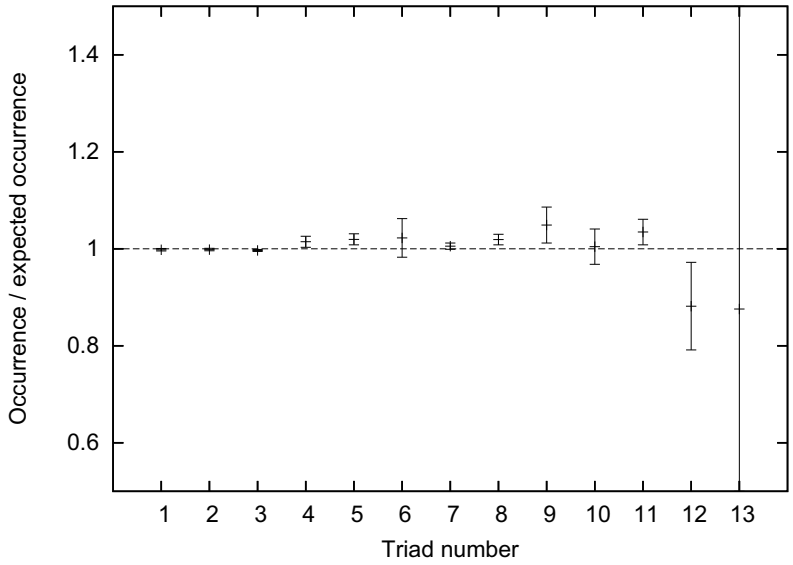

Figure 8.5: Abundance of triads in evolved networks divided by the expectation value for the occurrence of the triads in random networks. Networks are the same as in figure 8.4.

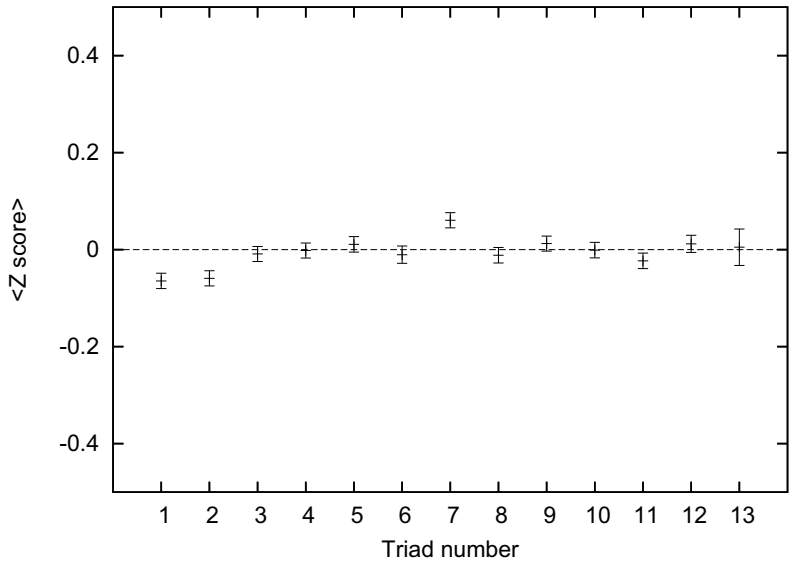

Figure 8.6: Abundance of triads in evolved networks as compared to a randomized ensemble. Fixed points are counted as unstable. The networks consist of 16 nodes with connectivity 3. Average over 4000 evolution runs. A new set of initial states is randomly created at every evolution step.

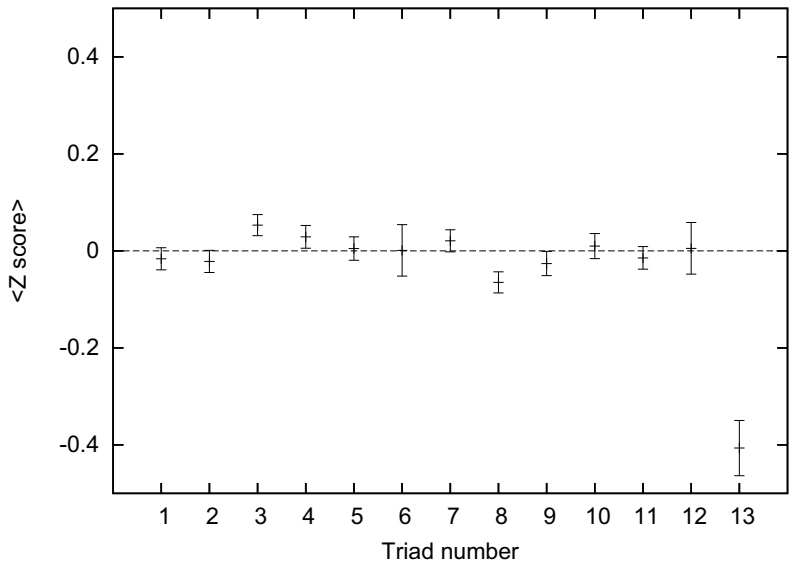

Figure 8.7: Abundance of triads in evolved networks as compared to a randomized ensemble. Fixed points are counted as stable. The networks consist of 32 nodes with connectivity 2. Results from 4000 independent evolution runs.

8.2.1 Model variants

We have tested several variations of the model to assess the robustness of our results. First, we changed the way different initial conditions are tested. Instead of using a fixed set of initial conditions for every network during a given evolution run, we now randomly create initial states at every evolutionary step. This prevents the evolution procedure from creating networks that are specialized to run reliably for a specific set of initial states. However, it also means an increase of the noise of the fitness score of a given network, as the score is now dependent on the chosen initial states. The results of this modified evolution process are shown in figure 8.6. Here, a network of 16 nodes with connectivity 3 is shown, but a similar result was obtained for a 32-node, connectivity 2 network. Again, the results closely resemble the expectation of random networks, with small outliers that are within statistically expected deviations.

Next, we investigated the effect of the fixed points on the evolution procedure. As mentioned before, it has to be explicitly defined whether fixed points are regarded as stable or unstable attractors. So far, all evolution runs uesd the defintion that fixed points are unstable, in order to prevent networks from running into landscapes consisting only of fixed points. However, in the reliability assessment of the isolated triads in figure 8.1, fixed points were regarded as stable attractors. Now, we set the fitness score of a fixed point attractor to 1 if the asynchronous dynamics reaches this fixed point to check whether allowing fixed points attractors significantly changes the results. In fig. 8.7 one can see that

also in this case no significant deviations from the random network ensemble are present.

Due to restrictions in computational complexity we had to limit the extent of our simulations which leads to a number of problems that we now want to discuss.

First, only a small number of initial states was used, which leads to the possibility that the network adapts to the specific set of initial states. The approximate run-time of a full enumeration of initial states even at a network size of only 16 nodes is of the order a minute. This renders an evolution process using full enumeration intractable. To estimate the performance of the networks under all initial conditions, we have checked the fitness value of the evolved networks with 1000 different random initial conditions (10 times more than were involved in the evolution process itself). We have found good agreement with the results from the evolution runs. As we have stated earlier, networks that have emerged from the evolution are stable under most (but not all) possible initial conditions, simply because we test many different attractors even if we only start at a limited set of initial states.

Second, there is an intrinsic noise of the fitness definition. Unreliable networks may by chance perform well in the evolution. However, this effect is suppressed by the repetition of the dynamics for each set of initial states. Increasing the maximal number of steps the asynchronous dynamics has to follow the synchronous version, does not have any measurable impact on the results.

8.3 Discussion

Our evolutionary scenario does not reliably (or even typically) reproduce the motif distribution of the real-world data set. Networks with completely different motif distributions can also be very stable under our noisy dynamics and networks with a similar motif distribution than observed in the data can be unstable. We have run a series of tests with different fitness definitions, varying network size and connectivity. In the end, we were unable to reproduce the experimentally observed motif distribution in an evolutionary process selecting for reliable dynamics.

A possible explanation for the result is that even those triads which are potentially unreliable, can also be reliable for certain configurations of link weights. For example, the feedback loop, which has a rather poor reliability score when one averages over all possible configurations of inhibiting or activating links, can be reliable for most initial states if an odd number of links carries a negative weight [33]. Thus, networks might be able to adapt towards reliability by changing link weights which does not affect the motif structures. Further network characteristics which are not captured in the triad significance profile are non-local effects such as loops of length larger than 3 and the potential suppression of instabilities by additional inputs on an otherwise unstable node (for example the synchronization between stable and individually unstable regions of the network). This may have such a strong impact on the reliability, that the effect of the triad robustness is highly suppressed.

Our results are in disagreement to some conjectures of [33] where it was proposed that

stability of triads could influence the overall stability of the network in which they are embedded and could thus explain the motif distribution found in real-world networks [31]. In our evolution, no such effect can be seen. One can understand this result in the way that the features of isolated triads do not translate into respective characteristics when embedded in a network context. This observation has also been made in a comparative study of fungi, in which it was shown that the triad linking patterns did not play a dominant role in the determination of the regulatory function [153].

As a potential extension of this work, one could consider networks with modular organization as the triads might then have a stronger impact on overall system features. In [145] it was shown that modularly varying goals can lead to spontaneous evolution of modularity and enhancement of network motifs. It is conceivable that a modular robustness test leads to more isolated triads and might potentially enhance the effect of the individual triads on the reliability of the full network topology.

9 Networks can rapidly evolve to reliability

In the preceding chapter we have investigated whether the requirement of robust dynamics can explain the motif distribution found in real signaling networks. Using a suitable evolution procedure and selecting for robust dynamics, we showed that although dramatic changes in the reliability properties were achieved, the real-world motif distributions could not be reproduced. This leads to the question of how easily evolvable the property of reliable dynamics really is. In this chapter we want to investigate this problem in a model that allows more rigorous treatment of the evolution process.

The previous investigations were hampered by two main problems. First, due to the enormous state space of the networks involved (consisting of 2^N configurations), a complete enumeration of all states is not suitable in evolutionary processes, in which a large number of different networks has to be treated. Thus, we will constrain ourselves to small networks in this chapter. Second, the fitness criterion employed suffered from an intrinsic noise due to the small number of repetitions of the dynamical procedure. Considering only small networks, we can solve this problem by employing a deterministic check for reliability first introduced in [46]. It requires complete enumeration of the state space, which is tractable for networks of size $N < 20$. It is strictly valid only in the limit of infinitely small perturbations and makes use of the fact that any perturbation that dies out during the course of an attractor cycle cannot further influence and destabilize the system. Thus, only those attractors are called unstable in which small perturbations can persist.

Using this reliability criterion, we want to quantify the evolvability of small networks towards reliable dynamics. In the first part of this chapter, section 9.2.1, we will evolve networks towards reliability of the complete attractor landscape. That means for any given initial condition the resulting attractor needs to be stable against timing perturbations. With this criterion we will investigate questions such as the number of evolution steps necessary to find a completely stable network instance and the effect this evolution procedure has on the dynamical properties of the system.

In the second part of the results part, section 9.2.2, we will drop the demand that all attractors of a network need to be stable and instead concentrate on a single attractor of the original network. This attractor is chosen as the desired function of the network and serves as a functional prototype. Any mutant network during the evolution has to reproduce this functional attractor. The constraint of reliable dynamics is then only applied to this functional attractor and it is investigated how many networks can be stabilized under this criterion.

9.1 Model description

To assess the stability of a network against fluctuations of the signal times, we use the stability measure of [46] which gives a deterministic result for a network under investigation. It requires two principle assumptions: the nodes implement a low-pass filter that removes the effect of activity states that are maintained only over short time spans; and the signal time fluctuations are small compared to the time scales of the processes and that of the filter.

The first assumption is again justified by the buildup and decay processes of protein concentrations [135]. Gene activity states that persist only for a short time do not significantly affect other proteins as the gene's transcript can only be produced in small numbers. The second assumption means that we are investigating systems with low noise. A single signal fluctuation does not significantly perturb the system, but only the addition of many similar perturbations over time can drive the system away from the synchronous behavior.

As the criterion uses small perturbations from the synchronous behavior, the process is started by determining the full phase space of the synchronous dynamics. The result is a mapping of every state to its sequential state and a list of attractors (fixed points and limit cycles). One can now determine the stability of a chosen attractor. For this, any state that is one of the limit cycle states can be used as a starting condition.

Now we determine all nodes that switch their state from the previous step to the chosen step. We call this set of switching nodes M. Next, we choose a proper, non-empty subset $S \subset M$ and change the switching times from $t = 0$ to $t = \epsilon$, i.e. we retard the switching times for all nodes in S by an infinitesimal number. Thus, a new intermediate state from time $t = 0$ to $t = \epsilon$ is created, where some nodes have already switched, whereas other nodes still exhibit the state of the previous (synchronous) time step. We then follow the dynamics with two times for every synchronous time step:

1. Determine the states at times $t = i$, $i = 1, 2, \ldots$ and $t' = i + \epsilon$ from the states at $t = i - 1$ and $t' = i - 1 + \epsilon$, respectively, according to the synchronous prescription.

2. Apply the filter rule: if a node switches both at integer and perturbed time, the node state persists only for a time span of ϵ. Because of the gradual effects of activities in the system, we assume that such a short-term activity does not further affect the system. Thus, we can just flip the state of the node at integer time and thereby explicitly remove the short-term activity fluctuation from the dynamical sequence.

3. If all nodes switch at either integer or perturbed time, the system has regained synchrony and the attractor is stable against the perturbation of the signal times of this particular subset of nodes. If however, the system reaches a new attractor in the combined state space of both times, the system is unstable as the perturbation can in general persist in the system and might diverge, thus leading to a different atttractor or to a "chaotic" regime of incessant switches.

This systematic test is repeated for all possible subsets S. We call an attractor "stable" if it is stable against all subset perturbations, otherwise we call it "unstable". It suffices to start from one randomly chosen state within the attractor cycle and investigate the perturbation of all switching events that occur at this step. This is due to the filter

characteristics. To prove this point, we assume the opposite: if there was a situation of perturbed signal times (caused by a perturbation at a different step of the cycle) that included a node not switching at the chosen step in the synchronous sequence, it can only be a short-term signal (i.e. the node would have to switch at both integer and perturbed time) and consequently would be filtered out. Thus, only those perturbations can persist that include only nodes switching at the respective time steps and it suffices to investigate all possible subsets of switching nodes at one particular step.

The reliability measure explicitly uses the concept of switches during a limit cycle. As no switches occur when the system has reached a fixed point, any fixed point attractor is trivially stable in this definition. However, we will again test our results also with the definition that fixed points are regarded as unstable attractors to investigate the effects of this definition.

Again, we use an evolution process to drive the networks to stability. The procedure is initialized by random creation of a network with a given number of nodes and connectivity. In every step of the evolution, the network is copied and mutated and its stability assessment is compared with the mother network. In contrast to the previous chapter, self-coupling will be allowed here unless otherwise stated. As different selection criteria are used, the definition of the fitness score is given in the respective part of the results section. If the fitness of the mutant is higher than that of the mother network, the mutant is kept and replaces the original, otherwise a new mutant is tested. This is repeated until the requested criterion is maximally fulfilled.

Mutation is again performed through a single link rewiring, which means that at the same time a connection between two nodes is removed and a new connection between two nodes is added. This procedure amounts to two elementary manipulations of the network structure but it has the advantage that the average connectivity of the network is unchanged by the mutation. This allows for better comparison between the random and the evolved networks. As our method requires full enumeration of the space of 2^N states where N is the number of nodes, we can only perform this analysis for small networks. We show the results for $N = 16$ nodes, but have checked that the conclusions also hold for networks with $N = 12$ and $N = 20$ nodes.[1]

9.2 Results

9.2.1 Reliable attractor landscapes

In the first part we evolve networks towards reliability regarding the complete attractor landscape. We define the evolution process in the following way: given a network, we accept a mutation of it, if the mutant has a higher number of initial states leading to a stable attractor. If so, the network is replaced by the mutant and the next evolution step is taken, otherwise a new mutation is tested. This procedure stops as soon as all initial states lead to stable attractors.

[1] In chapter 10 we will investigate larger system sizes in a related model.

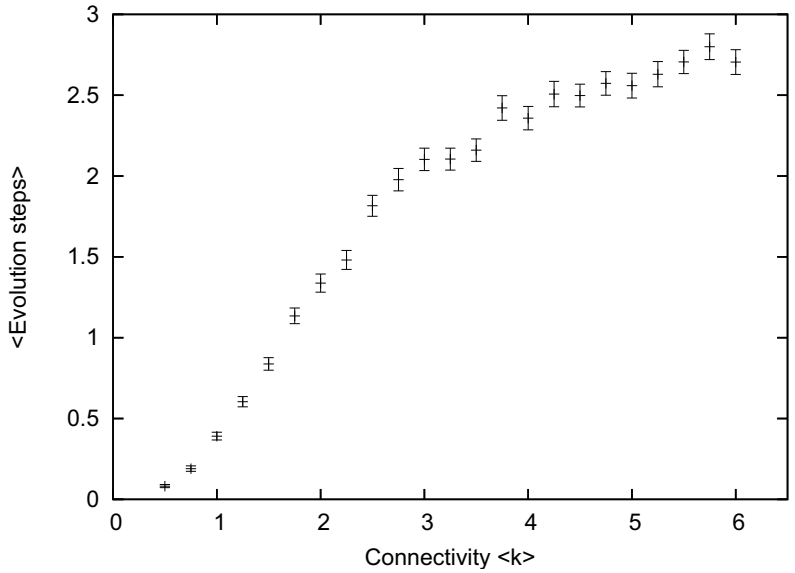

Figure 9.1: Network evolution rapidly leads to stable attractor landscapes: average number of evolution steps vs. average connectivity of the networks. The networks consist of 16 nodes, each data point corresponds to the average of 1000 runs starting with random networks of the respective connectivity.

Interestingly, all such evolution runs ended successfully, such that a completely reliable network could be found starting from any random network. In figure 9.1 we show the average number of evolution steps necessary to reach full stability of the attractor landscape, plotted against the average connectivity, defined by the total number of links divided by the number of nodes. Networks consist of 16 nodes and 1000 repetitions were run for every data point. One can see that for all connectivities a very small number of mutations already suffices to completely stabilize a network.

Inevitably, some of the networks that are being evolved exhibit reliable dynamics already at the start of the evolution process. This is especially true for small connectivities, where a large number of attractors are fixed points. To show that we do not simply observe the effect of networks not being evolved at all, for figure 9.2 only those runs are used in which the initial network was not completely reliable. The average number of evolution steps is again plotted against the average connectivity of the network. Although one can see that the average number of evolution steps is raised by about 1 across the whole range of connectivities, it is still a small number. Thus, one can see that short evolutionary paths suffice to stabilize networks.

Next, it is interesting to look at network properties and how they change during the course of the evolution process. In table 9.1 we compare random networks with networks that have undergone the evolution process for an average connectivity of $\langle k \rangle = 3$ (but the qualitative results are typical for any value of $\langle k \rangle$). One can see that the average number

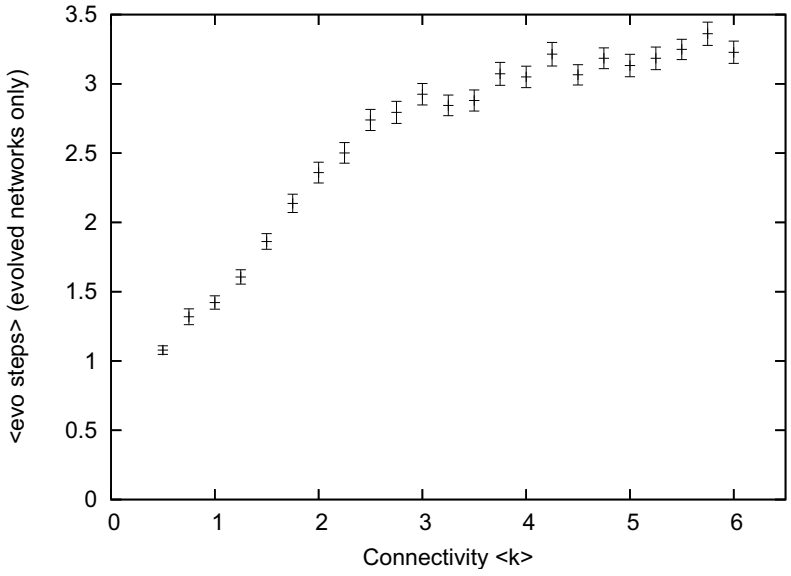

Figure 9.2: Number of evolution steps vs. average connectivity. Here, only those evolution processes are taken into account in which the starting network was unreliable.

	random networks	evolved networks
number of attractors	3.98 ± 0.02	2.12 ± 0.01
average attractor length	3.47 ± 0.03	4.10 ± 0.04
largest basin size	47800 ± 100	57100 ± 100
IS to stable atts.	40300 ± 200	65536
number of evolution steps	–	2.07 ± 0.02

Table 9.1: Comparison of attractor basin characteristics of random and evolved networks for N=16, $\langle k \rangle = 3$. Averages over 20000 runs.

of attractors has decreased and that the size of the largest basin has increased at the same time. Again, these significant effects take place within very few evolution steps. Thus, we find that the dynamical landscape of a threshold network can be significantly altered by only a few mutations of the network topology. Stability of the attractor landscape can be achieved without significant changes of the overall network structure, because at the same time, dramatic alterations of the dynamical characteristics can occur.

The changes in network properties can be explained by rather simple arguments. If the basin of attraction of a cycle is large, there is a high probability that an intermediate state belongs to the same basin of attraction and will thus reenter the same cycle. If it enters at the step before or after the current synchronous step, the stability criterion is fulfilled. Longer attractor cycles can also aid stability: As it is necessary to have more than one causal chain through the full limit cycle [33], the probability for unstable attractors decreases with the length of the attractor cycles.

To ensure that we do not simply observe the effects of networks evolving towards fixed points (which are defined as being stable), we have checked all results also with the rule that a fixed point is counted as an unstable attractor. These results are presented in section 9.3.1, but the conclusions drawn above hold also in this case. The only qualitative difference is that the pronounced drop of the average number of evolution steps at low connectivities (fig. 9.1) is not present if fixed points are regarded as unstable attractors. Thus, the two different slopes for $\langle k \rangle < 3$ and $\langle k \rangle > 3$ are due to the abundance of fixed points in networks with low connectivities.

These results clearly show that in our simple dynamical model, random networks exhibit an astonishing evolvability regarding their attractor landscapes. The landscape can be easily shaped and robustness in dynamical functioning can be achieved after very few evolution steps.

In figure 9.1 we have already shown that very short paths through network structure space suffice to find fully reliable dynamics. This is particularly surprising as we have not actually searched for the shortest such path. The selection criterion used was very loose: any mutant with a higher fitness than the mother network would be accepted. If one is interested in the minimum number of steps necessary to reach a reliable instance of a network, an alternative selection process can be utilized. At every evolution step, a set of 2000 mutants is tested. The mutant that achieves the highest fitness score is then selected to replace the original network. The results of this modified evolution procedure (denoted by '$\times$') are given in figure 9.3 along with the results of the original procedure (denoted by '+'). Here, we again show the average number of evolution steps in those runs where the original network was unreliable. Quite surprisingly, the average number of rewirings necessary to reach a reliable network is very close to 1 across the full range of connectivities tested – thus, almost every random network can be mutated into a reliable network within a single rewiring.

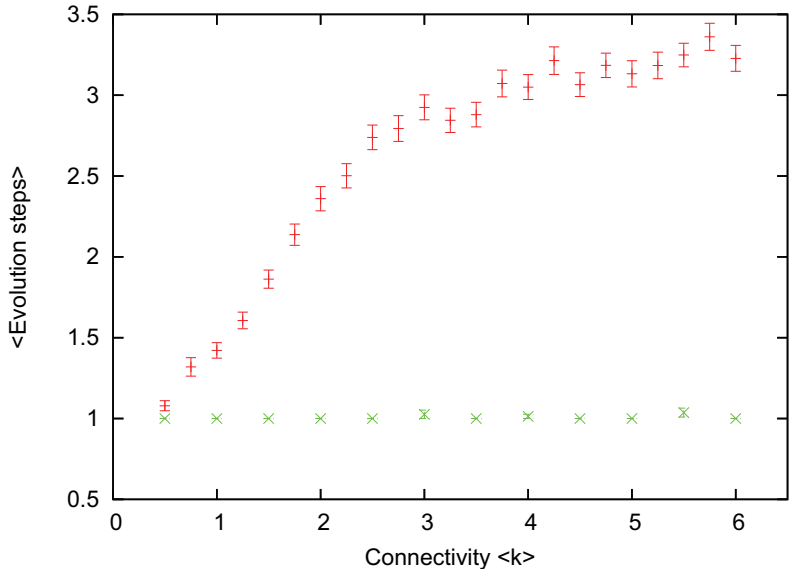

Figure 9.3: Number of evolution steps necessary to reach completely reliable attractor landscapes in two different selection schemes. Original criterion as shown in figure 9.1, denoted by '+', and alternative criterion, where the best out of 2000 tested mutants is chosen to replace the mother network, denoted by '×'.

9.2.2 Functional attractor stabilization

So far we have demonstrated that random networks can be quickly evolved towards a completely stable attractor landscape. However, we have not constrained the dynamics in any way, so the evolved networks might show completely different dynamical behavior than the original networks. If we think of attractors as a function being performed by a genetic network, we should restrict evolution to those networks that are able to reproduce the original attractor dynamics.

This leads to a modified selection criterion. We choose the largest attractor of the original network as the "functional attractor" and require stabilization of this attractor. If it is a fixed point or a stable limit cycle, there is trivially nothing to do in the evolution, so we just discard these networks and create a new one until we find a network with an unstable largest attractor. During evolution, every mutant has to reproduce this attractor. This means that, starting at one step of the attractor cycle, the dynamics of the original network and of the mutant have to be exactly the same. If the mutant does not reproduce the attractor, it is immediately discarded. We do not request the networks to reproduce the transient states as this constraint is too strict and disallows practically every mutation.

The fitness score is given by the multiplication of the stability value (0 if unstable, 1 if stable) with the basin size of the functional attractor. This amounts to the selection of any network that reliably reproduces the functional attractor, followed by an evolution

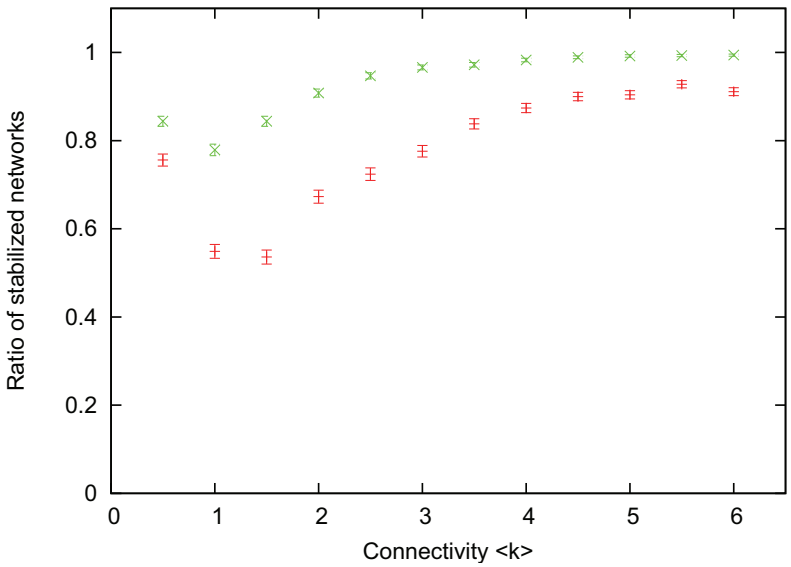

Figure 9.4: Functional attractor stabilization. The ratio of networks successfully stabilized in the evolution is plotted against average connectivity. Network size $N = 16$, every data point amounts to the average of 1000 runs at the respective connectivity. Upper points ($\times$): neutral mutations (see text). Lower points ($+$): single link rewirings. Many networks can be stabilized, with neutral mutations practically all networks with connectivities larger than 2.

towards large basins of attraction. We employ two different selection criteria: strict or neutral selection. In the strict selection scheme, a network is only accepted, if it increases the fitness score, whereas in the neutral selection a larger or equal fitness score suffices. This means that in the strict scheme, the stabilization has to occur within a single rewiring, whereas the neutral criterion allows for a random walk through the space of networks that exhibit the functional attractor. The evolution process is complete as soon as the functional attractor is stable with a basin size of half the total state space, which makes the functional attractor the dominant dynamical expression pattern.[2]

In figure 9.4 we show the results of the evolution processes using the functional attractor criterion for a network size of $N = 16$ and 1000 attempted evolution runs. The ratio of networks that can be stabilized in both selection schemes is plotted against the average connectivity of the networks. For each evolution step, we have attempted 20000 mutations before marking a network as not evolvable towards stability (this arbitrary value for the parameter does not influence the results as long as the number of attempts is sufficiently large). In the neutral selection a stable network also has to be found within 10^6 mutation

[2] After the stabilization of the attractor, the evolution process continues until the requested basin size is reached. This, however, does not affect the following results. We have also conducted evolution runs with the sole constraint that the attractor be stable and obtained the same results.

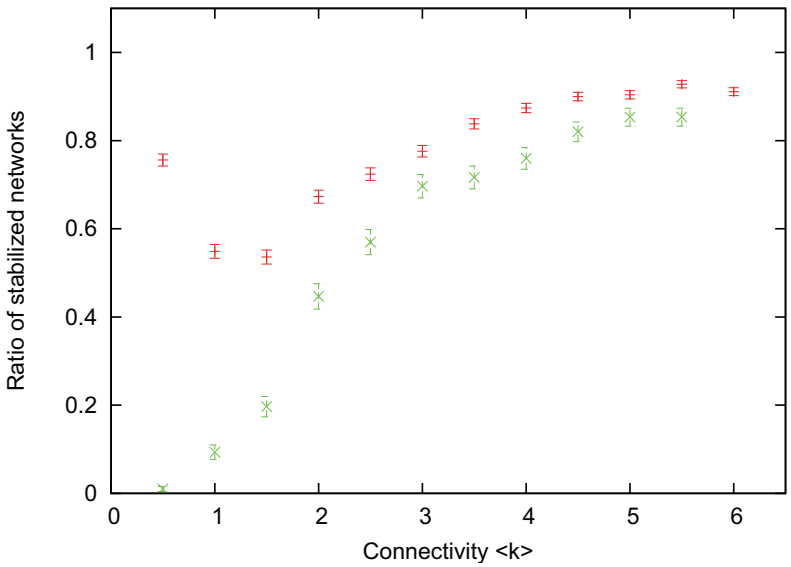

Figure 9.5: Model variation: no self-coupling allowed. Again, the ratio of networks successfully stabilized in the evolution is plotted against average connectivity, denoted by '×'. For comparison. the results for allowed self-couplings are also shown (denoted by '+').

attempts during the full evolution run.

First, one can see that even in the single-step evolution (marked by '+'), more than half of all networks can be stabilized. For very low connectivity as well as connectivities above 3, more than 3/4 of all networks fulfill the criterion. In the case of neutral selection (points marked by '×'), this ratio is even higher. Especially for networks of connectivities around 1.5, the probability of evolving towards a stable realization is significantly increased. For connectivities above 2, practically every network can be stabilized using this evolution process.

There is a dip at connectivities around 1 in the single-step evolution and around 1.5 in the neutral selection. The reason for this is that at small connectivities, attractor cycles are caused by short loops in the network topology – at very low connectivities unstable dynamics caused by two nodes influencing each other can often be stabilized by the formation of a self-loop of one of these nodes (as illustrated in figure 5.4). At higher connectivities close to the critical value, larger dynamical cores emerge, which are, however, still dependent on a small number of links and cannot easily be stabilized. At even larger connectivities, these dynamical cores are often formed by dense connections of many links, some of which are dispensable and can be rewired without destroying the attractor.

To demonstrate this point, we show the effect of excluding self-links in the network structure. In figure 9.5 the comparison with the original model is shown. One can clearly see that for small connectivities the ratio decreases significantly, while at higher connectivities

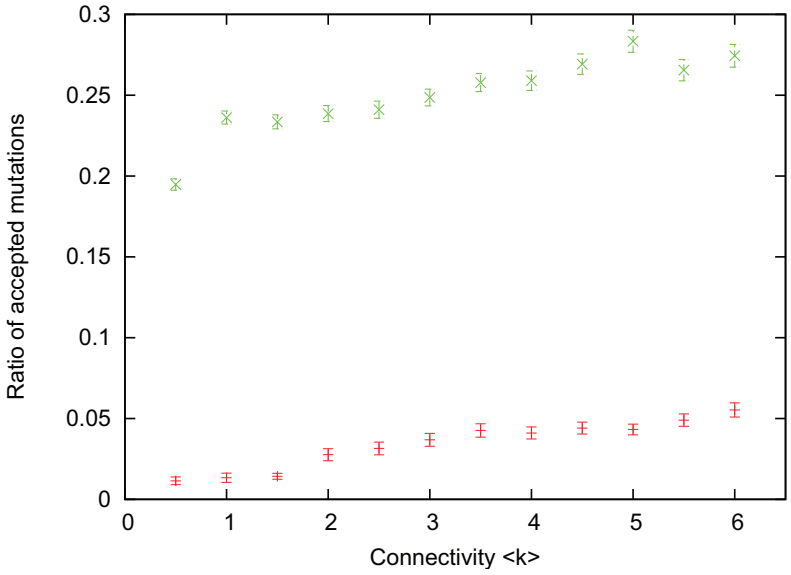

Figure 9.6: Ratio of accepted mutations during the course of evolution. Neutral (×) and single-step (+) evolution.

the effect is much less pronounced.

As we have seen, the space of threshold networks is of such kind that, starting from any network with an unreliable largest attractor, one can usually (especially for large connectivities) find a path through structure space that leads to a realization of the same dynamics, but one which is reliable against timing perturbations. However, it is not clear, whether this path is actually realistically accessible to a biological evolution process.

In figure 9.6 we therefore plot the ratio of accepted mutations, that is the number of mutant networks that were accepted during the course of evolution, divided by the total number of attempted mutations. As expected, the ratio of accepted mutations is much higher for neutral than for the single-step evolution. But even in the single-step case, the evolution process does seem feasible as for all connectivities more than 1% of the mutations are accepted. For high connectivities, this ratio rises up to 5%. In the neutral evolution scenario, the ratio of accepted mutations is between 20 and 30%.

However, in the neutral scenario, it is conceivable that long evolutionary paths are necessary until a stabilized network is found. Thus, we show in 9.7 the average number of steps required to complete the evolution process. The results are presented in a log-linear plot and there are quite dramatic differences for the different connectivities. Surprisingly, larger connectivities display much shorter evolution paths, with a more than twenty-fold increase from $\langle k \rangle = 5.5$ to $\langle k \rangle = 1.0$.

The distribution of evolution steps seems to be heavy-tailed. As an example we show the cumulative distribution of evolution steps for a connectivity of 4.5 as obtained from 1000 simulation runs in figure 9.8. The dashed line is a plot of the function x^{-1} and serves as a

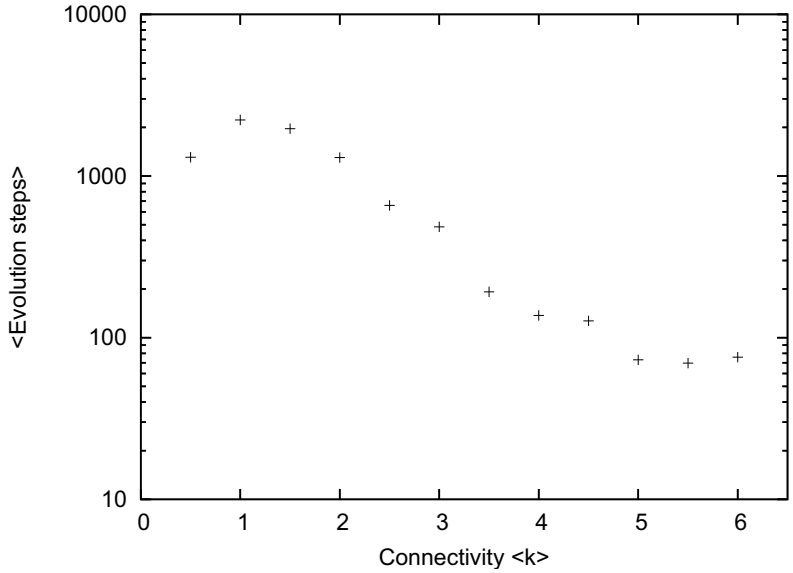

Figure 9.7: Number of evolution steps in the neutral selection evolution. Log-linear scale. The second moment of the distributions for a given connectivity might be diverging (see figure 9.8), which is why now error bars are shown.

guide to the eye. It corresponds to a scaling behavior $\sim x^{-2}$ which has a diverging mean. As the distribution is below this line, the mean converges and can be used to characterize the distribution. However, the variance might be diverging which is why no error bars are plotted in figure 9.7.

In figure 9.9 we show a typical example of an evolutionary process for a network with 12 nodes and $\langle k \rangle = 2$ (created with DDLAB [154]). The network structure as well as the full synchronous attractor landscape are shown, before (top) and after a single mutation (bottom). On the left side of each panel, the network's structure is given, where solid lines depict activating and dashed lines depict inhibiting links. The thick lines mark the mutated links: in the upper panel the thick line shows the connection that was removed by the mutation; in the lower panel the newly added link is marked in this way.

In the attractor landscape figure, each state is represented by a dot that is connected by a line to the concurrent state. The central shape denotes the limit cycle (or fixed point). All four attractors of the original network are unstable (obviously, this cannot be seen in a graph of the synchronous state space). After mutation only two attractors remain. The functional attractor is now stable. One can see how the single mutation dramatically affects the attractor landscape of the network.

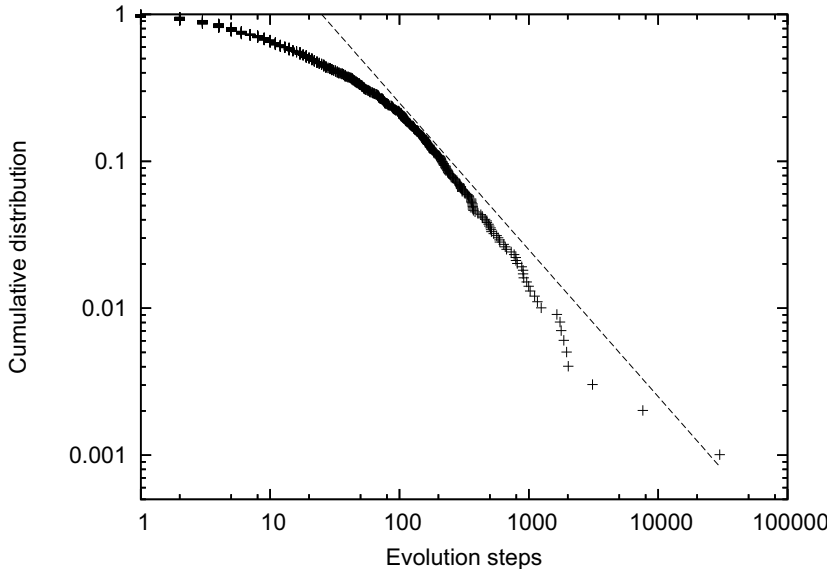

Figure 9.8: Plot of the cumulative distribution of the number of evolution steps for a network connectivity of 4.5. The straight line serves as a guide to the eye and is a plot of the function x^{-1}.

9.3 Model robustness

We have investigated several variations of this model to check the robustness of our results. In the following sections we present these results that further support our claims of the main results section.

9.3.1 Fixed points as unstable attractors

In the previous chapter, we have considered fixed points as stable attractors, simply because no switches are happening at all, thus no desynchronization can occur. Here, we want to investigate, whether this arbitrary choice has an important effect on our results. Thus, we regard fixed points as unstable attractors in this section. It obvious that this increases the effort to find a stable network, particularly for low connectivity networks where many fixed points are present in the attractor landscape.

In figure 9.10, we show the same evolution procedure as in figure 9.1, but with fixed points regarded as unstable attractors. Thus, the evolution needs to create attractor landscapes consisting entirely of stable limit cycles. It is immediately apparent that the evolution procedures take longer than before to complete. However, as can be seen from figure 9.10, over the whole range of connectivities, the average number of evolution steps is still quite small. The highest value for the average number of rewiring steps needed to complete the evolution is less than four. Thus, our conclusion that networks can rapidly

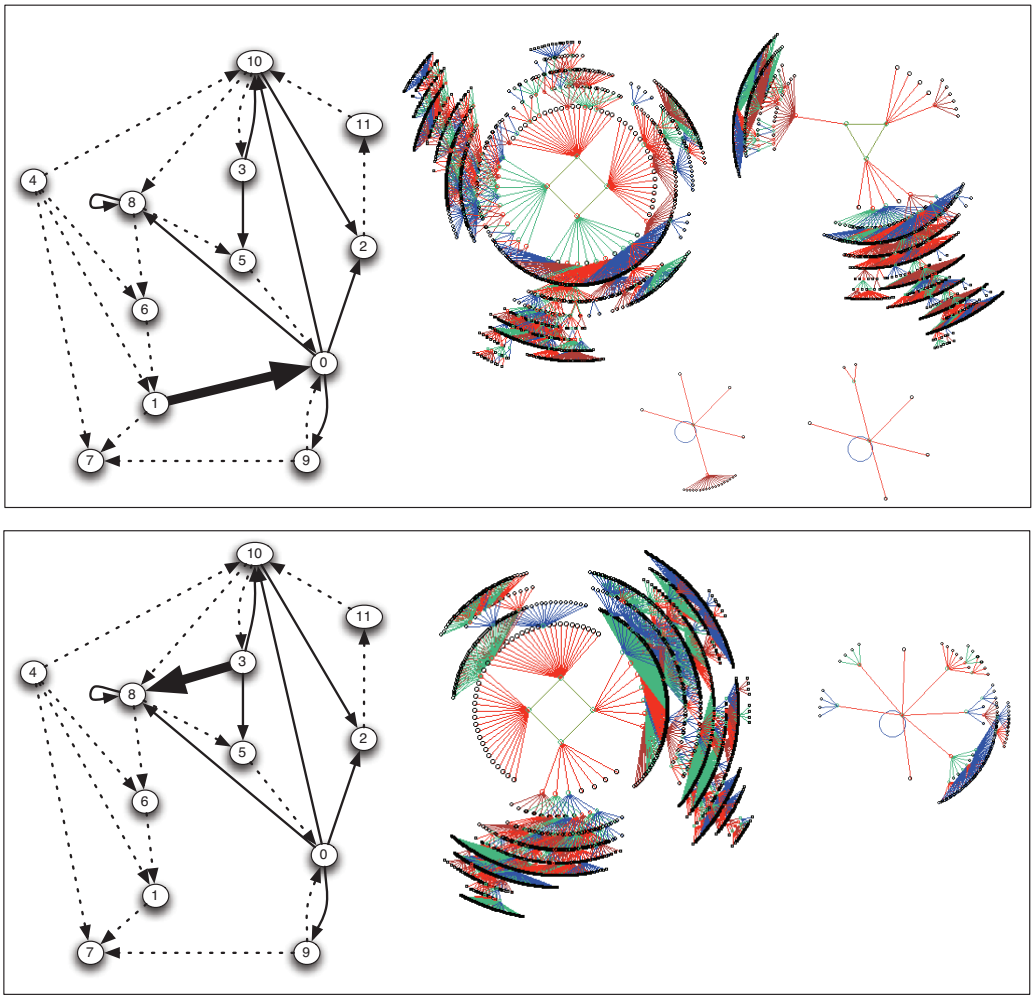

Figure 9.9: A single rewiring can dramatically affect the attractor landscape. The network structure and attractor landscape (state space visualization) of a network before (top) and after (bottom) a single rewiring. Central shape in the attractor pictures show limit cycle, transient states are arranged on arcs.

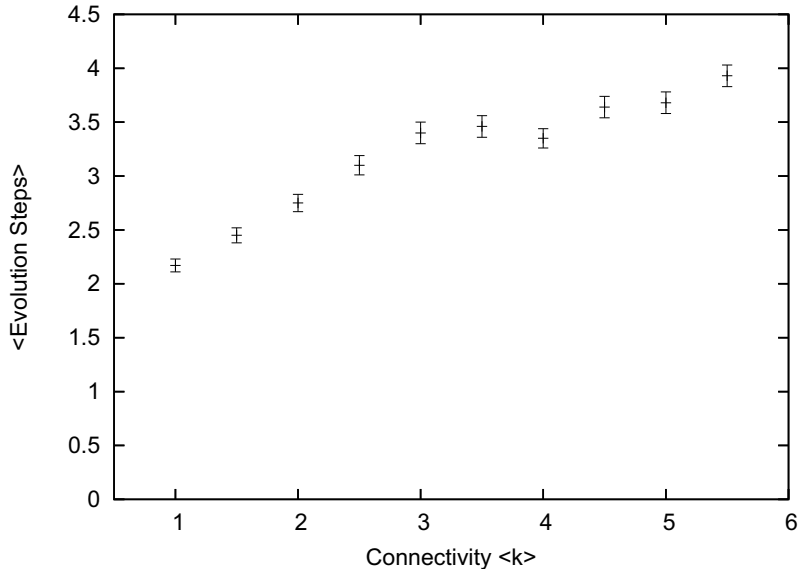

Figure 9.10: Model variation: Fixed points are regarded as unstable attractors. Average number of evolution steps vs. average connectivity of the networks.

evolve towards reliable attractor landscapes still holds true even if regarding fixed points as unstable attractors.

In the case of the functional attractor criterion, it does not make sense to investigate fixed points, as their reliability is fixed by definition and thus cannot be affected through evolution.

	random networks	evolved networks
number of attractors	3.22 ± 0.02	1.66 ± 0.01
average attractor length	3.96 ± 0.04	5.02 ± 0.05
largest basin size	50900 ± 400	61000 ± 400
IS to stable atts.	42800 ± 400	65536
number of evolution steps	0	1.83 ± 0.02

Table 9.2: Model variation: No self-coupling allowed. Comparison of attractor basin characteristics of random and evolved networks for N=16, average connectivity k=3. Averages over 20000 runs.

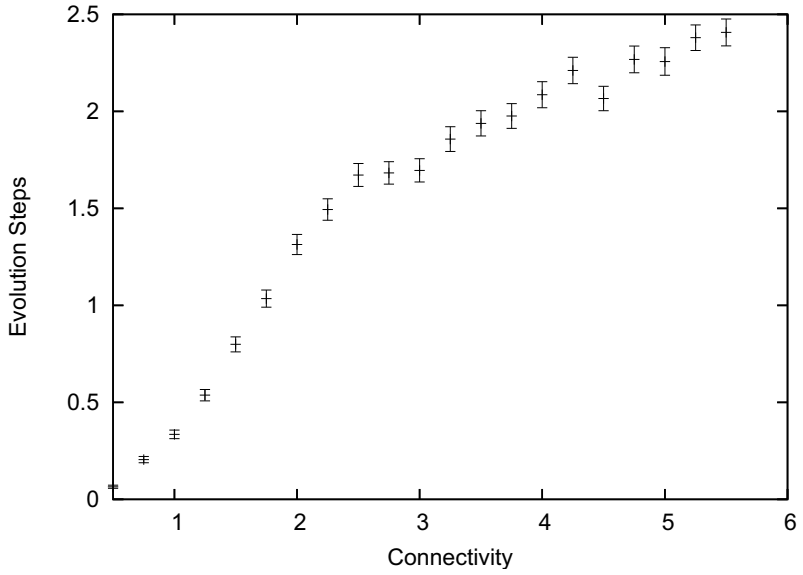

Figure 9.11: Model variation: No self-coupling allowed. Average number of evolution steps vs. average connectivity of the networks.

9.3.2 No self-coupling

Next, we investigate the effect of excluding self-links in the landscape stabilization scenario. First, we give a characterization of the effects of this change on the dynamical properties. In table 9.2 one can see that the number of attractors is smaller and the average attractor length is larger than in the case of allowed self-links (given in table 9.1), suggesting that this seemingly small model change actually strongly affects the dynamics. This is true for both random as well as evolved networks. Again, we find the same effects of the evolutionary process: the number of attractors decreases and the average attractor lenghts increase. The number of evolution steps necessary to perform these changes, is even smaller than in the original model.

In figure 9.11 we again show the plot of the average number of evolution step against the connectivity. The number of evolution steps is somewhat smaller than in the original model over the tested range of connectivities but the overall functional form is very similar. The main result is again valid here: short evolutionary paths suffice to find stable realizations for random networks.

9.3.3 Finite size scaling of attractor lengths with sample size

Attractor lengths are approximately power-law distributed [46]. To make sure that we do not obtain results that depend on the sample size, we show the scaling of the average attractor length with sample size in figure 9.12 – the x-axis is logarithmically scaled. The straight lines gives the final value for a sample size of 10^6 as a guide to the eye. The curve

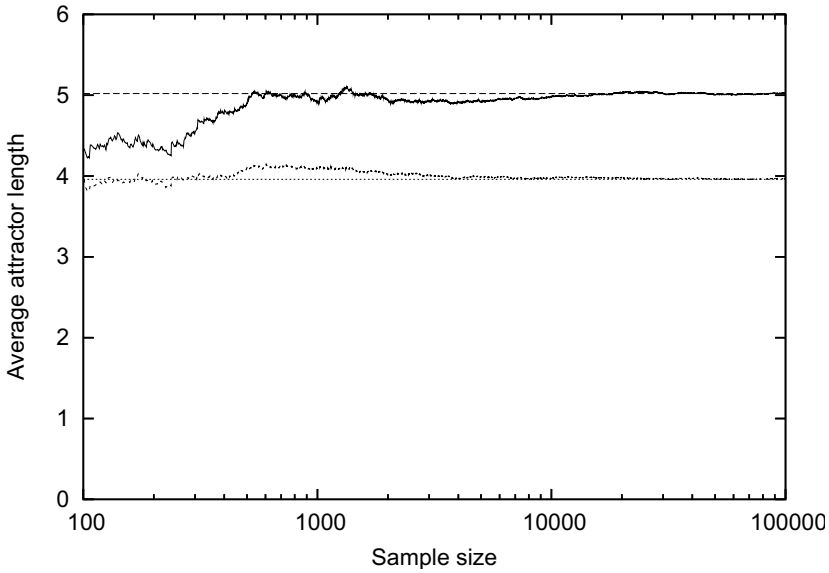

Figure 9.12: Finite size scaling of attractor length. With increasing sample size, the average value for the attractor length converges – logarithmic scaling of the x-axis. Upper curve: evolved networks, lower curve: random networks.

quickly converges to the final result, which gives confidence to the results shown in tables 9.1 and 9.2.

9.3.4 Different system sizes

In the functional attractor criterion, we have investigated the dependence of the ratio of stabilized networks on the system size. Because of the deterministic criterion used, we cannot perform this investigation for arbitrary system sizes, but we can investigate to which extend the exact number of nodes influences the results. In figure 9.13 we repeat the plot from figure 9.4 for three different system sizes, $N = 12, 16, 20$. As can be seen, the differences between the three system sizes is negligible for the results. The investigation of larger system sizes is conducted in a modified model in chapter 10, but for small system sizes one can conclude that our results presented in this chapter apparently do not depend on the exact system size considered.

9.3.5 Improved fitness criterion

In section 9.2.2 we have performed the evolution process using the functional attractor criterion employing two different selection schemes: single-step and neutral selection. In this section, we want to check how a more elaborate selection scheme affects the results. Instead of a binary choice of either stable or unstable attractors, we now quantify the

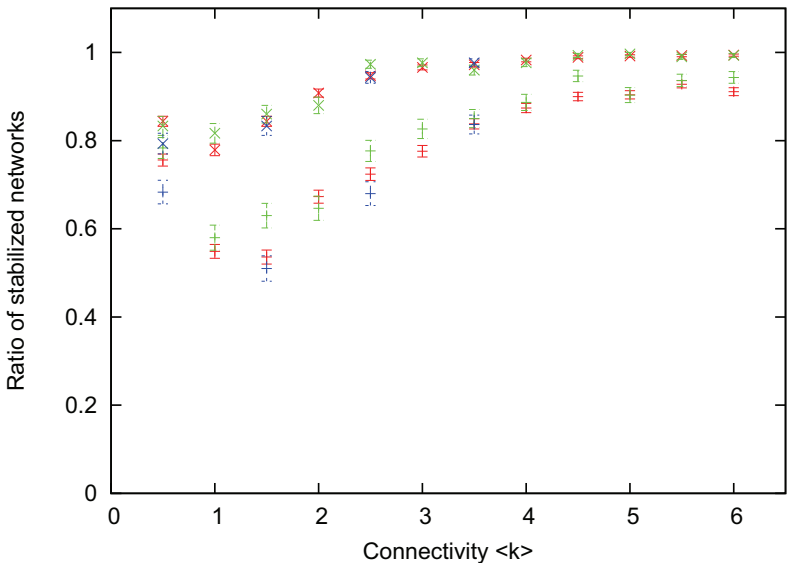

Figure 9.13: Comparison of different system sizes in the functional attractor stabilization. System sizes are distinguished by color: green points for $N = 12$, red for $N = 16$, and blue for $N = 20$. Upper points ($\times$): neutral mutations. Lower points ($+$): single link rewirings. For these system sizes, the results are very similar.

instabilities of an attractor under consideration. For this, we count explicitly for every configuration of switches whether it leads to unstable dynamics in the sense defined above and call the total number of unstable configurations u_t. The fitness score is then given as $1/1 + (u_t)$ (which keeps the fitness score between 0 and 1 and is monotonically falling for a rising u_t). Now, a mutant is selected if the fitness score of the functional attractor is higher than the mother network. The results for this modified evolution process are given in figure 9.14. The curve lies between the single-step and the neutral curves, which is not surprising as this selection criterion is less restrictive than the single-step criterion but compared to the neutral selection criterion does not allow those mutations that reduce the improved fitness score.

9.3.6 Rewire inlinks only

The last model variation that we want to discuss concerns the mutation process. Up to now we have considered mutations that remove one link and add another link anywhere in the network. As both of these effects have to happen at the same time, one might consider it an artificial situation. It seems more natural that a gene mutates in such a way that a binding site vanishes and another binding site is created at the same time. Thus, we have also tested a mutation process that leaves the target node of a link unchanged and

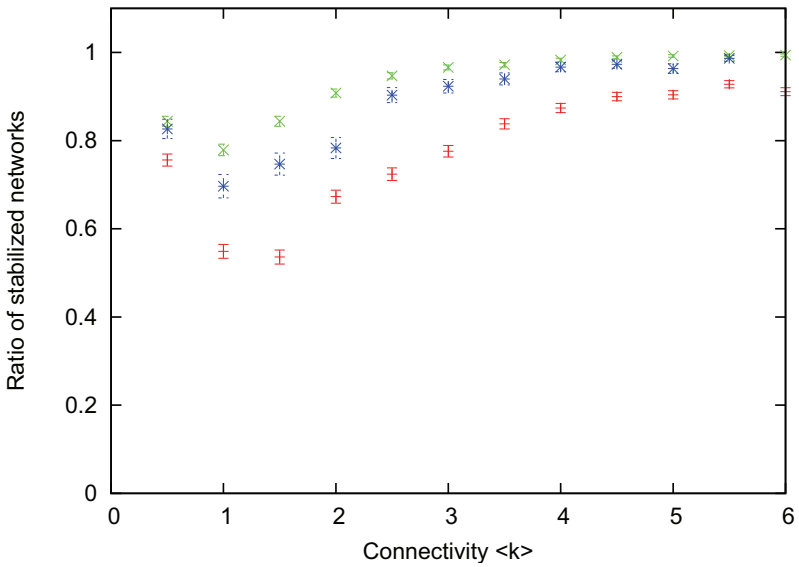

Figure 9.14: Alternative fitness criterion (‘∗’), compared to single-step (‘+’) and neutral selection (‘×’).

only moves the tail, thus keeping the in-degrees of all nodes constant but changing the out-degrees during the evolution. We present the results of this modified prescription in figure 9.15 and compare with the original mutation scheme. One can see that both curves are quite similar over the whole range of connectivities. Interestingly, there is a small difference at higher connectivities, meaning that there are some networks that cannot be evolved towards reliability if the in-degree of each node is kept. Still, the main message remains that most networks can be easily evolved towards a reliable functional attractor.

9.4 Discussion

In this chapter we have investigated the evolvability of networks towards reliability of dynamics against small perturbations of the signal event times. Starting from synchronous dynamics and systematically testing all infinitesimal perturbations to the synchronous sequence, we obtain a deterministic measure of the reliability of an attractor of a given network.

Using this criterion we investigate two different evolution processes. First we show that a completely stable attractor landscape can be obtained within very few rewiring steps. Thus, starting from any random network, a network with completely reliable dynamics is close in network structure space. This means that the features that can be measured by averaging over the whole network, such as degree distribution, clustering coefficient or motif distribution cannot directly dictate the stability of the resulting network dynamics. Small

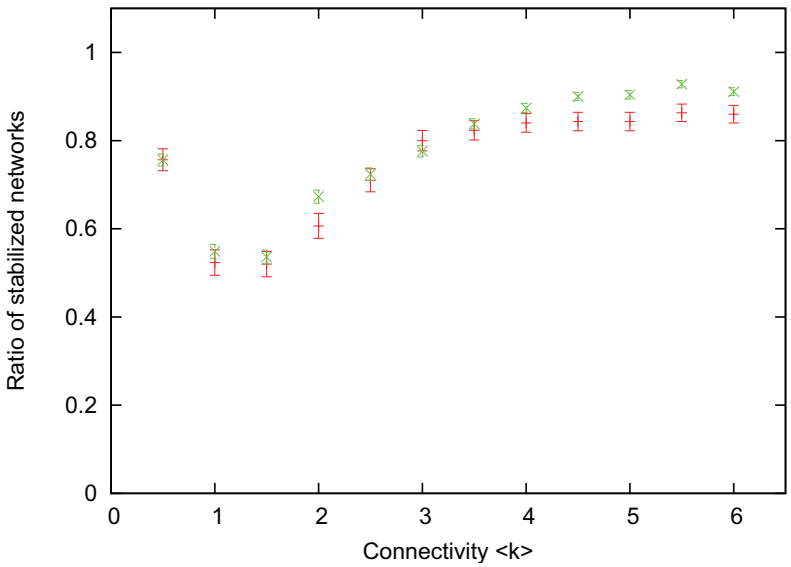

Figure 9.15: Different mutation scheme: instead of freely moving a link from one place in the network to any other, only those mutations are conducted, in which the affected node is kept the same. Only the affecting node can change. Again, the ratio of stabilized networks is plotted against the average conectivity. The original mutation scheme is marked by '×' and the modified scheme by '+'

changes in the rewiring can have dramatic effects on the attractor landscape, including complete stabilization.

In the second part, we define one attractor of the initial network as the functional prototype and seek stabilization of this attractor through evolution. Even within small topological changes, it is often possible to find networks which exhibit the same attractor, but perform it in a reliable way. This means that the synchronous state sequence of an attractor does not in general determine its ability to run in a stable fashion. The same attractor can often be performed reliably if a suitable network is found through the evolution process.

We have discussed several variations of the model and conclude that our results are very robust against the variations discussed. Significant enhancements of the evolvability in the functional attractor scheme could be found when neutral mutations were also allowed. During the evolution, a wider search through network space can be performed which leads to a higher ratio of stabilized networks.

In all cases considered, we found that the evolvability of networks towards reliability of dynamics is higher for networks with higher connectivities. Although in the landscape stabilization process, sparsely connected networks are quickly evolved (in the sense of number of mutation steps necessary to complete the evolution), the actual ratio of rewired links is smaller for more densely connected networks. Also in the functional attractor

scheme, the ratio of stabilized networks is higher for more densely connected networks, causing an easier evolvability towards reliability.

10 Evolution under finite noise

In the previous chapter we have used a deterministic model to quantify the evolvability of networks towards reliable dynamics. While the deterministic nature of the reliability criterion ensures that the evolution process is least hampered by noise in the fitness criterion, it also has some disadvantages. Apart from its abstract nature and conceptual complexity, it imposes serious limitations on the sizes of the investigated networks, as it requires complete enumeration of the state space of the synchronous dynamics.

Also, in [155] it was discussed that this criterion, which was also used in [46], misjudges some specific cases of reliability assessment. In this chapter, the dynamical model of chapter 8 will be used to investigate the question of evolvability towards functional attractors. This model overcomes problems of the infinitesimal scheme by explicitly modeling the time evolution.

10.1 Model description

We again use the notion of functional attractors to define what is the desired dynamical behavior of the network under investigation. Here, we simply determine the functional attractor by running the synchronous model with random initial conditions until an attractor is found. During the evolution process we again demand each network instance 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.

The evolution procedure again is a simple version of a mutation and selection process. We start by creating a directed random network with the prescribed number of links (self-links are allowed) and determine the functional attractor. During the evolution procedure, we mutate the current network by 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 connected.

The fitness of a given network is assessed by comparison of the asynchronous dynamics with the prescribed functional attractor. The discrete activity state of all nodes are set according 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 time course of the system is followed according to equations (5.6), (5.5). A detailed description of this continuous time algorithm is given in appendix A.

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 only take those networks for the evolution which lead to a limit

cycle from the random initialization. Others are discarded before the start of the evolution process.

In the evolution process, the fitness of a mutant is compared to the fitness that the mother network scored. Thus, a network is only selected, 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 might actually be less reliable than the mother network can still be selected in the evolution due to variability in the fitness score.

If a network follows the attractor until the prescribed maximal step number, it is thought to be "reliable". If during the network process a given number of mutation tries is exceeded, the evolution process is aborted. The maximal number of mutation tries in the evolution is $a_{\text{step}} = 20000$ 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 parameters in the results part. The delay time t_d is set to unity, the charging 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. We follow the dynamics for maximally 10^6 time steps and the fitness score is given by the number of steps correctly following the synchronous behavior, divided by this maximal step number.

In figures 10.1, 10.2 we show an example of a typical evolution process for a small network of $N = 12$. During three steps, the network is evolved towards a stable realization. The initial network (upper-left in figure 10.1) displays three synchronous attractors (top panel in figure 10.2) of which the first is chosen as the functional attractor. The structural changes are depicted in figure 10.1 by a grey arrow for the removed link and a plus-sign for the newly added link. As is typical for these evolution processes and was shown already in chapter 9, the attractor landscape is affected dramatically during the evolution. In this example, only the functional attractor survives the evolution procedure.

10.2 Results

We have performed the described network evolution for a variety of different network sizes as well as connectivities. For system sizes of $N = 16, 32, 50$ and connectivities between 0.5 and 6 the ratio of networks that were stabilized during the evolution is shown in figure 10.3. 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 is expected as a very similar dependence on the connectivity was found for

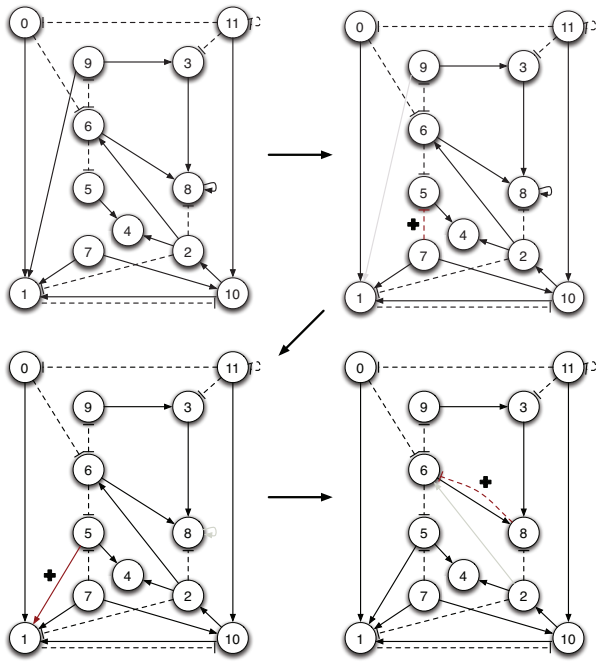

Figure 10.1: A typical example of an evolution process for a network of size $N = 12$. In this example, three steps suffice for stabilization. The structure of each network during the evolution is shown, with the arrows denoting the subsequent step in the evolution. In every step, one link is lost (shown in grey color) and a new link is added (denoted by the plus sign). The change of the dynamical structure of the network is given in figure 10.2.

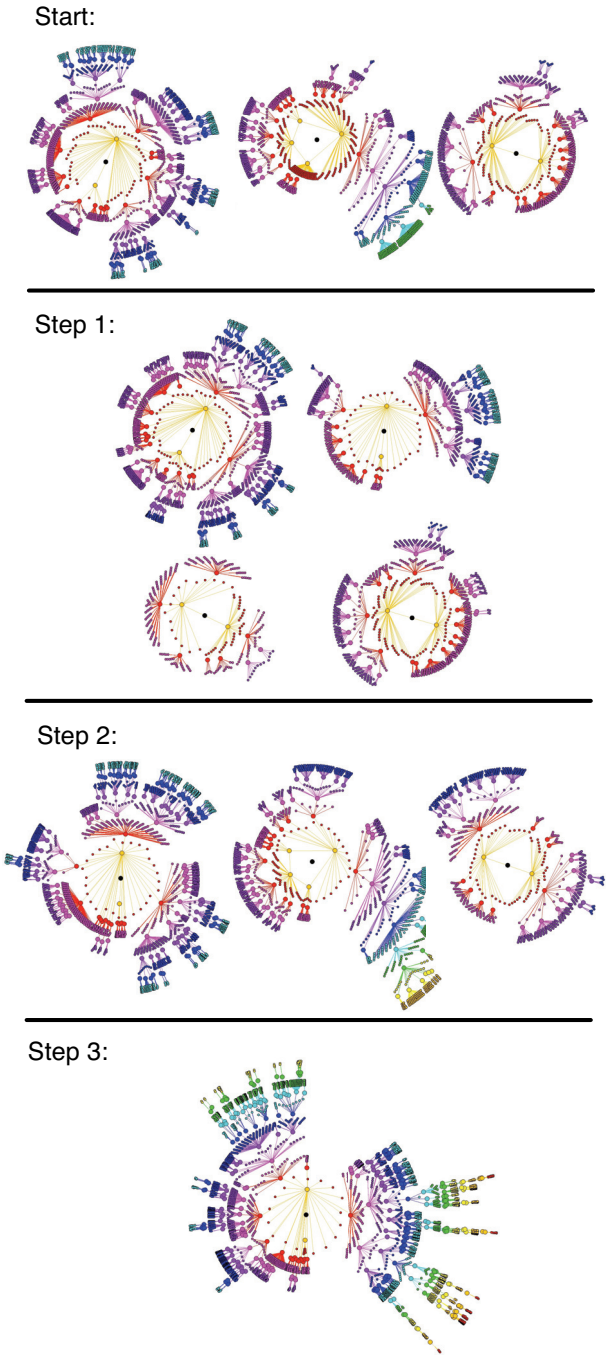

Figure 10.2: Change of the (synchronous) attractor landscape during evolution – corresponding to the network structures shown in figure 10.1. For every step, the full attractor landscape is shown. Every dot denotes a state, the subsequent state is connected via a line. The limit cycle is shown in the center of each attractor basin. The functional attractor is shown as the first attractor in all steps.

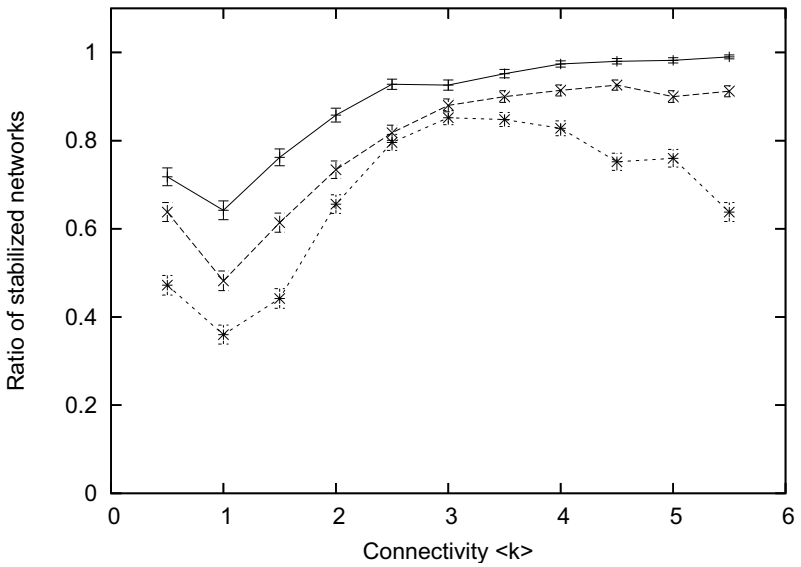

Figure 10.3: Ratio of networks that were stabilized during the evolution plotted against the average connectivity of the networks for network sizes of N=16 (straight line), N=32 (long dashes), N=50 (short dashes).

16 nodes in the infinitesimal scheme in chapter 9, although for slightly different stability definition and evolution procedure.

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 of the stabilization ratio for connectivities below two. 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 that actually affects the dynamically relevant nodes at all without destroying the 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. This effect was also seen for small networks in chapter 9.

However, considering large connectivities and large system sizes, the ratio of stabilized networks drops again. This can also be understood and has to do with the well-known increase of 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 figure 10.4 shows the average number of rewiring steps necessary until a reliable network realization is found for networks of 32 nodes. For all connectivities, this number is remarkably low, considering that 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 quicky finds a realization exhibiting reliable dynamics. Interestingly, the number

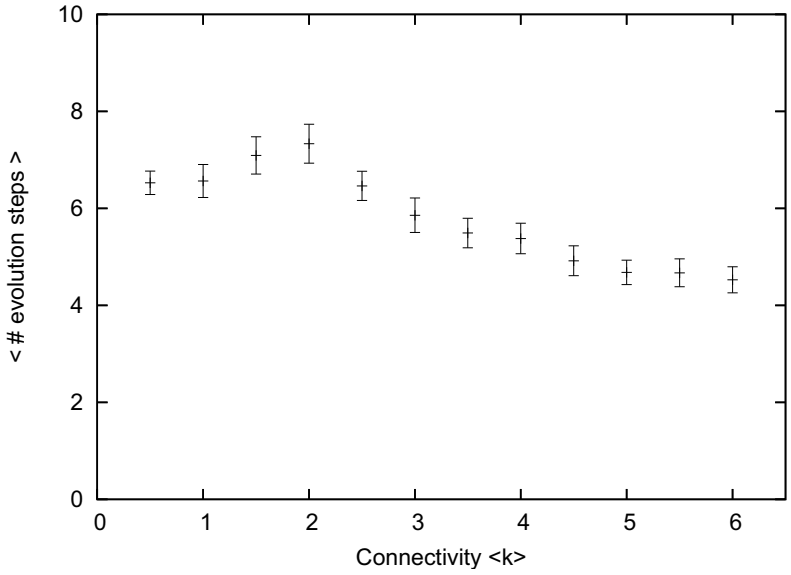

Figure 10.4: Average number of evolution steps until stable realization is reached – $N = 32$

of evolution steps does not monotonically grow with the connectivity, but instead drops for connectivities larger than two.

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).

Next, we further want to investigate the dependence on network size by repeating the evolution procedure with system sizes up to $N = 400$. This is shown in figure 10.5 in a log-linear plot of the ratio of stabilized networks against the system size. We find that the ability of the process to stabilize a given network decreases with the system size. The straight 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 rather 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 $N = 16$, 20000 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 figure 10.5 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 figure 10.6 we again show the ratio of stabilized networks against system size, but this time for two different parameter values – the original parameter set with $a_{\text{step}} = 2 \cdot 10^4$ (denoted by '+') and for an increased

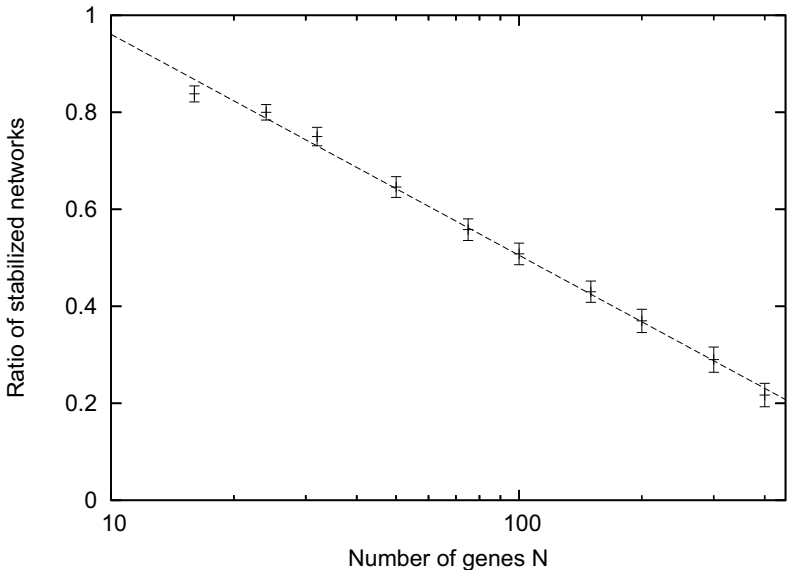

Figure 10.5: Ratio of networks that were stabilized during the course of evolution plotted against the number of nodes in the networks for an average connectivity of 2. The dashed line is given by a logarithmic fit of the data.

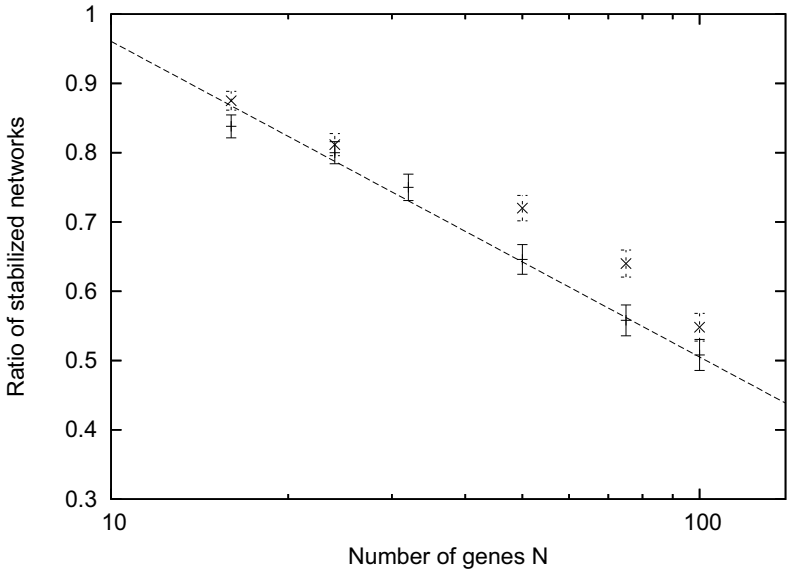

Figure 10.6: Comparison of parameter values. Ratio of networks that were stabilized against the number of nodes in the networks for an average connectivity of 2. Original set of parameters marked with '+', points obtained with increased step number marked with '×'.

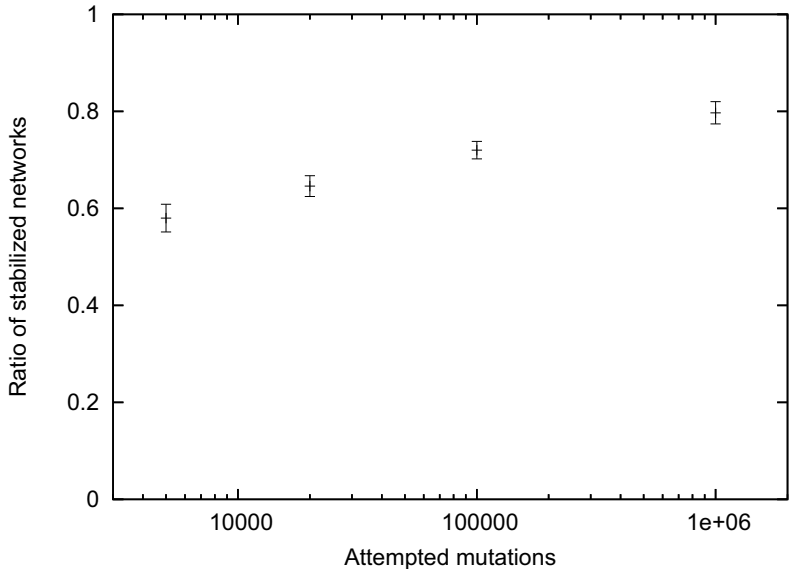

Figure 10.7: Effect of parameter a_{step} on the results for $N = 50$. The ratio of stabilized networks is plotted against the value of the paramter a_{step}, giving the maximal number of attempted mutations per evolution step.

value of $a_{\text{step}} = 10^5$ (denoted by '$\times$'). For small networks, the value of this parameter obviously does not significantly affect the results, but for $N > 50$, differences can be clearly seen. For $N = 50$ the ratio rises from 0.65 ± 0.02 at $a_{\text{step}} = 2 \cdot 10^5$ 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 figure 10.7. 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 trying more mutations in every 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 viable as the chance to find exactly the right mutation 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. We see this as a model for small networks of key generators as were described in recent Boolean models of biological systems [13, 21]. 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.

10.3 Discussion

We have seen that a high ratio of random networks can evolve towards instances displaying reliable dynamics. In accordance with other recent work [42, 43], it is 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 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 [156, 157].

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. Nevertheless, this decrease with system size is rather slow and shows that even large networks can often be stabilized.

11 Conclusions

In this work questions of dynamical reliability and evolvability of gene regulatory networks were studied. Using simple discrete dynamical models that capture the essential features of transcriptional regulation, the interplay of network structure and reliability of dynamics was investigated.

A model was introduced that generalizes the synchronous Boolean model to continuous time and allows stochastic fluctuations of the signal transmission times. The main ingredients of this model are a signal delay and a low-pass filter, that allow in principle to create dynamically reliable circuits. We have devised a modeling framework using linear differential equations whose piecewise solutions can be assembled in a simple numerical algorithm to obtain the full dynamical time course. In this framework one can naturally incorporate stochastic fluctuations by varying the delay times of the individual signals, which implements small perturbations of the dynamical behavior of a regulatory network.

Using examples of small circuits that are either known from organisms or can be constructed as synthetic networks, the reliability criterion was explored and several simple conclusions were drawn. Additionally to the established fact that a given network structure can exhibit both reliable and unreliable dynamics, it was shown that the same dynamical pattern can be reliably or unreliably performed by different networks. Furthermore, simple mechanisms underlying reliable dynamical behavior were discussed.

With this simple reliability criterion the question of how reliability can emerge in more complex control structures can be asked. This question has several levels and in this thesis different aspects were investigated. All of these studies have in common that a conceptually very simple form of network dynamics was employed: Boolean network nodes influence each other via threshold transmission functions, such that the functional output of a node is given only by the structure of the network, the state of the influencing node and the type – activating or inhibiting – of the respective links.

First, the reliability criterion was used to investigate the reliability of a real regulatory network. For the well-characterized system of the budding yeast cell cycle, the Boolean model of [13] was extended to the stochastic asynchronous model introduced in this work. It was shown that the system displays an astonishing reliability against fluctuating signal transmission times and two main features of the network's dynamical organization were identified that lead to this reliability: First, the discrete states characterizing the time course of protein concentrations persist for extended periods of time, which limits the effect of fluctuating signal times. Second, phases of simultaneous concentration changes of several proteins are interspersed with steps in which only a single protein species changes its activity level. These steps act as "catcher states" that remove any divergence of signal times from the dynamics.

The main focus of this work lies on the question which constraints the demand for reliable dynamics imposes on network structure. It was investigated whether the specific motif distributions found in real-world signaling networks can emerge from a network evolution process. Starting from the recently suggested explanation that these structures might stem from the reliability of the individual triads as observed in [33], a simple evolution process was devised.

In each evolution run, a set of initial states is defined as the different environmental conditions in which the network has to function reliably. Networks are mutated by the rewiring of single links and are selected during the evolution if they display a higher reliabiliy than the mother networks. In summary, it was found that despite the known differences in reliability of isolated triads, the corresponding network evolution cannot account for the motif distributions in real-world networks. A number of model variants was investigated and no typical network structures can be identified that emerge from this evolution process. The effects of specific wiring patterns overrule the influence of statistically measurable quantities such as the motif distribution. In particular, it was demonstrated that reliable networks can display virtually any distribution of triads, which rules out the possibility that motif structures are primarily caused by a selection pressure towards dynamical reliability.

To investigate whether the constraint of reliable dynamics has any measurable influence on the motif structure of evolved networks, averages over many networks that result from the evolution process were investigated. Again, several model and parameter variations were tested and no significant deviation of the evolved networks from the random expectation was found. Thus, we conclude that while the constraint of reliable dynamics has implications on the selection of isolated triads, the effect on the overall network structure is practically not measurable in our model and does not seem feasible as an explanation for the motif structures of real-world networks.

Another conclusion drawn from this result is that adaptations in the functional behavior can occur without significant effects on the network structure. To further investigate this issue, we sought to reduce all possible sources of noise in the fitness measure. We devised a reliability criterion based on the criterion from [46], which uses infinitesimal perturbations and checks whether any perturbation persists during the course of the attractor or dies out. By fully enumerating the state space of the synchronous dynamics, noise in the fitness criterion associated with the explicit simulation of the dynamics is eliminated and a deterministic reliability measure for small networks can be constructed.

With this deterministic reliability measure we used two evolution methods to quantify the evolvability towards reliable network dynamics. First, the entire landscape of a network was selected to be reliable. It was obtained that an attractor landscape consisting only of reliable attractors can be reached from any random network within surprisingly few rewiring steps. In addition, when looking for the optimal rewiring at each evolutionary time step, it was found that practically every network is within one mutation step from a completely reliable network. This also explains why stastically measured properties such as the motif distribution typically do not differ significantly from the expectation of random networks. It was also shown that small changes of the topological structure can dramatically influence the dynamical properties of networks.

For the second evolution process, the concept of functional attractors is introduced, which refers to a sequence of network states that constitute the desired dynamical behavior of a network. In contrast to the previous evolution processes, in which the only requirement was that of reliable dynamics, the functional attractor criterion requests a specific dynamical behavior to be reproduced. Only if a network does this correctly, the reliability of the dynamics is assessed. This means that instead of the stabilization of any network behavior it is now requested that a particular dynamical behavior has to be reliably reproduced. Surprisingly, many networks can be stabilized in this criterion and a sizable number can even be stabilized within a single rewiring. This means that the synchronous state sequence of an attractor does not in general imply whether it can be performed reliably or not. By investigating a number of model variations, some interesting findings were obtained: the evolvability of networks with very sparse connectivities significantly depends on the ability to form self-links. If one explicitly forbids self-coupling, the ratio of stabilized networks drops practically to zero. Furthermore, it is found that networks with high connectivities can almost always be stabilized while evolution processes involving networks of intermediate connectivities around three have a higher probability to fail. Surprisingly, the number of evolution steps to reach a reliable instance of a network, is dramatically higher for sparse networks than it is for denser networks.

Although the model has the great advantage of supplying a deterministic reliability score, it suffers from the fact that it is rather abstract and unbiological. We therefore sought to investigate the questions of evolvability towards reliability also in the biologically more plausible model of finite fluctuations that was already used in the investigation of the motif structures.

The ability of a network to display a reliable functional attractor can be easily investigated in this framework as well, as it requires the simulation to be run only starting from a single initial condition. First, it was shown that the results from using the deterministic criterion can be reproduced in this framework, i.e. small networks are often easily evolved towards reliability of the functional attractor and super-critical connectivities strongly enhance this evolvability.

The advantage of explicitly modeling the dynamics compared to the deterministic enumeration scheme is that larger system sizes can be investigated. It was found that both for very sparse and for dense networks, the evolvability significantly drops for larger system sizes, whereas for intermediate connectivities of two to four the decline is quite slow. For a connectivity of two and system sizes between 16 and 400, a systematic investigation of the dependence on system size was conducted and it was found that the decline is approximately logarithmic with the number of nodes.

It has to be emphasized that the chosen dynamical model displays a special simplicity in the space of possible Boolean rules: as only threshold functions are used, no tinkering with specific (and potentially) unrealistic Boolean functions can be utilized in the evolution process – the evolution has to take place on network space only. Also, as no additional parameters affect the model, this evolution is limited to the rather strong individual change imposed by rewiring of links. In this regard, it is quite astonishing that networks can evolve quickly towards reliable dynamics.

It would be very interesting to investigate extensions of our model. As real-world networks obviously are not simply random, neither in their dynamical behavior, nor in their network structure, one should strive for the inclusion of specific traits into these models: the modularity typical for gene regulatory networks could specifically be incorporated (which would limit the sizes of groups of interacting components), just as non-Poissonian degree distributions. Also, requirements for dynamics could be improved. One possible extension would be to define input and output nodes and regard the rest of the network as a hidden "computational" machinery that does not have to display a given behavior but has to work reliably. Recently, it was shown [145] that a modular definition of dynamical goals can lead to a modular network structure. Potentially, this could also affect the outcome of the evolution processes discussed here.

How do our results fit in with other investigations of dynamical robustness? In a recent work [42] using synchronous dynamics it was observed that evolving networks with canalizing Boolean functions towards complete robustness against single node state changes in the initial states can always be accomplished. In a similar model [43], where also node state perturbations are considered, similar results to our "functional" attractor criterion were observed. It was shown that all network topologies that reach a given activity state when subject to a predefined initial state, can be traversed by single link mutations. The comparatively small number of networks that display high robustness properties are thus accessible to gradual evolution. This is in agreement with our observation that networks reproducing a predefined pattern can usually be quickly evolved towards robust functioning. We note that the requirement of reproduction of the complete attractor sequence as required in our criterion is much more restrictive than the constraint to reach a given final state.

We also want to emphasize that in these previous models the perturbations take place before the dynamics is started. We here have amended these studies by investigating the reliability against fluctuations during the dynamical behavior.

In summary, we have investigated different models of gene regulatory networks regarding the reliability of their dynamics. It was found that reliable dynamics can be easily evolved over a wide-range of network sizes and connectivities, and that even specific attractors can be performed in a reliable fashion, given the network has sufficient "buffer" in the network structure in the form of links that can be rewired without destroying the dynamical attractor. This quick evolvability towards reliable attractors on the other hand leads to the fact that non-random motif structures as found in real-world networks cannot be recovered from such evolution processes as the structural effect caused by the constraint for reliable dynamics is rather small. Applying our criterion to a real genetic network, it was found that simple mechanisms can be observed that lead to reliable functioning.

All of our result lead to the same general conclusion: the common approach of characterizing dynamical behavior of networks by statistical measures such as degree distribution is substantially limited. In addition to structural properties, specific local wiring patterns and roles of individual nodes have to be assessed. As was shown in our simple model, the effect of local changes in the structure can result in drastically different dynamical features.

Acknowledgements

I wish to thank Stefan Bornholdt for giving me the opportunity to write this thesis under his supervision, for countless interesting discussions and for his relentless correction work on papers and talks which taught me a lot about scientific presentation; Klaus Pawelzik for agreeing to be second referee although the date of submission fell exactly into his vacation time; and Hans-Günther Döbereiner and John Burrows for agreeing to be members of the colloquium.

I also want to thank all members of the institute for an open-minded and cooperative atmosphere, especially Fabian Zöhrer and Jörg Reichardt with whom I shared many hours discussing all important matters of life.

I also want to thank Stanislaus Brachmann for keeping our Macs in great shape and for his commitment to solve all computer-related issues that arose.

A very warm special thanks to Agnes Janßen for all the great advice, help with formalities and bureaucracy and the invaluable moral support.

I certainly want to thank the Deutsche Forschugsgemeinschaft for the financial support of this work under grants BO1242/5-1, BO1242/5-2 and the University of Bremen for supplying me with a scholarship to finish my thesis.

Without the support of my family and friends, this work would not have been possible. Thank you all so much for your confidence and patience, for entertaining hours and the great feeling that there is so much in life besides work. Special thanks and a big kiss to Jenny, who had to endure a lot in the final weeks of my work but never lost her calmness and patience.

A Description of the finite noise algorithm

The asynchronous algorithm is implemented such that no discretized "ticking" time 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)
- $s_{i,\mathrm{current}}$: current discrete state of node i
- $s_{i,\mathrm{aim}}$: discrete state of node i that would result from the current states of all nodes: $s_{i,\mathrm{aim}} = \Theta\left(\sum_{j=1}^{n} a_{ij} s_{j,\mathrm{current}} - 1/2\right)$,

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 $s_{i,\mathrm{current}}$ 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 $s_{i,\mathrm{aim}}$ differs from the current state $s_{i,\mathrm{current}}$. 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 $b_i = s_{i,\mathrm{current}}$, 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 equation (5.7) for t with $c_i = 0.5$:

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

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's 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 equation (5.7) 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.

B Supplementary plots for chapter 8

In this appendix, some plots will be presented that supplement the plots of chapter 8. They are mentioned in the main text and are shown here for the sake of completeness.

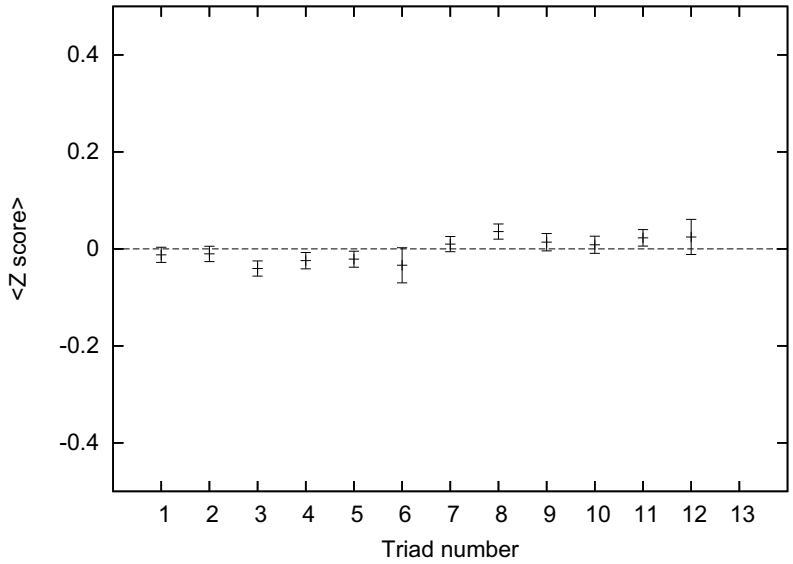

Figure B.1: Abundance of triads in evolved networks as compared to a randomized ensemble. Fixed points are counted as unstable. The networks consist of 32 nodes with connectivity 3. Average over 4000 evolution runs.

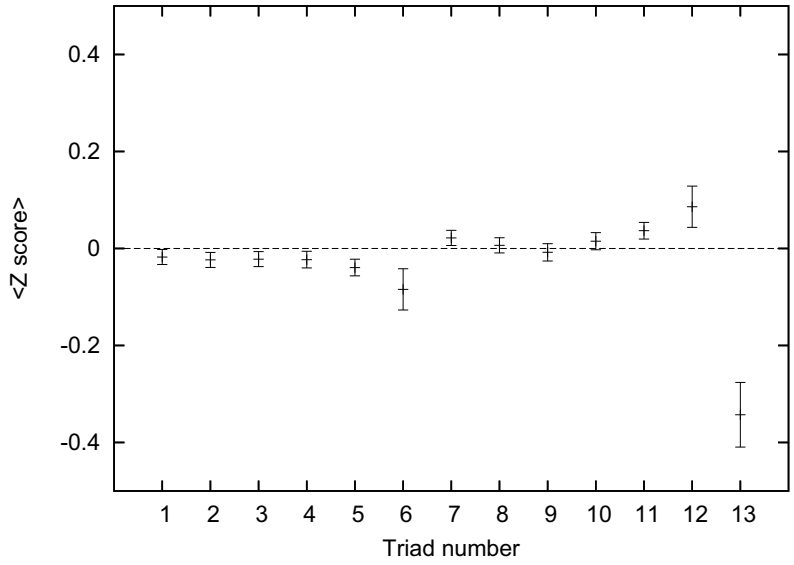

Figure B.2: Abundance of triads in evolved networks as compared to a randomized ensemble. Fixed points are counted as unstable. The networks consist of 32 nodes with connectivity 2. Average over 4000 evolution runs.

Bibliography

[1] H. Kitano, "Biological robustness," *Nat. Rev. Genet.*, vol. 5, no. 11, pp. 826–837, 2004.

[2] H. H. McAdams and A. Arkin, "It's a noisy business! genetic regulation at the nanomolar scale," *Trends Genet*, vol. 15, no. 2, pp. 65–69, 1999.

[3] J. M. Raser and E. K. O'Shea, "Noise in gene expression: Origins, consequences, and control," *Science*, vol. 309, no. 5743, pp. 2010–2013, 2005.

[4] A. Becskei and L. Serrano, "Engineering stability in gene networks by autoregulation," *Nature*, vol. 405, no. 6786, pp. 590–593, 2000.

[5] M. Thattai and A. van Oudenaarden, "Intrinsic noise in gene regulatory networks," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 15, pp. 8614–8619, 2001.

[6] N. Barkai and S. Leibler, "Robustness in simple biochemical networks," *Nature*, vol. 387, no. 6636, pp. 913–917, 1997.

[7] U. Alon, M. G. Surette, N. Barkai, and S. Leibler, "Robustness in bacterial chemotaxis," *Nature*, vol. 397, pp. 168–171, 1999.

[8] J. Stelling, E. D. Gilles, and I. Doyle, Francis J., "From the cover: Robustness properties of circadian clock architectures," *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 36, pp. 13210–13215, 2004.

[9] G. von Dassow, E. Meir, E. M. Munro, and G. M. Odell, "The segment polarity network is a robust developmental module," *Nature*, vol. 406, pp. 188–192, 2000.

[10] A. Eldar, R. Dorfman, D. Weiss, H. Ashe, B.-Z. Shilo, and N. Barkai, "Robustness of the bmp morphogen gradient in drosophila embryonic patterning," *Nature*, vol. 419, no. 6904, pp. 304–308, 2002.

[11] R. Albert and H. G. Othmer, "The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster," *J. Theor. Biol.*, vol. 223 (1), pp. 1–18, 2003.

[12] M. Chaves, R. Albert, and E. D. Sontag, "Robustness and fragility of boolean models for genetic regulatory networks," *J. Theor. Biol.*, vol. 235, pp. 431–449, 2005.

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

[14] C. Espinosa-Soto, P. Padilla-Longoria, and E. R. Alvarez-Buylla, "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*, vol. 16, no. 11, pp. 2923–2939, 2004.

[15] M. Kollmann, L. Lovdok, K. Bartholome, J. Timmer, and V. Sourjik, "Design principles of a bacterial signalling network," *Nature*, vol. 438, no. 7067, pp. 504–507, 2005.

[16] A. Wagner, "Circuit topology and the evolution of robustness in two-gene circadian oscillators," *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 33, pp. 11775–11780, 2005.

[17] W. Ma, L. Lai, Q. Ouyang, and C. Tang, "Robustness and modular design of the drosophila segment polarity network," *Mol Syst Biol*, vol. 2, p. 70, 2006.

[18] S. Bornholdt, "Systems biology: Less is more in modeling large genetic networks," *Science*, vol. 310, no. 5747, pp. 449–451, 2005.

[19] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, pp. 437–467, 1969.

[20] Y. Zhang, M. Qian, Q. Ouyang, M. Deng, F. Li, and C. Tang, "Stochastic model of yeast cell cycle network," *Physica D*, vol. 219, no. 1, pp. 35–39, 2006. arXiv:q-bio/0605011.

[21] M. I. Davidich and S. Bornholdt, "Boolean network model predicts cell cycle sequence of fission yeast," *PLoS ONE*, vol. 3, no. 2, p. e1672, 2008.

[22] S. Bornholdt and K. Klemm, "Robust gene regulation: Deterministic dynamics from asynchronous networks with delay." arXiv:q-bio/0409022, 2003.

[23] A. L. Barabási and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nat. Rev. Genet.*, vol. 5, pp. 101–113, 2004.

[24] M. M. Babu, N. M. Luscombe, L. Aravind, M. Gerstein, and S. A. Teichmann, "Structure and evolution of transcriptional regulatory networks," *Current Opinion in Structural Biology*, vol. 14, no. 3, pp. 283–291, 2004.

[25] B. Derrida and Y. Pomeau, "Random networks of automata: a simple annealed approximation," *Europhys. Lett.*, vol. 1, pp. 45–49, 1986.

[26] J. J. Fox and C. C. Hill, "From topology to dynamics in biochemical networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 11, no. 4, pp. 809–815, 2001.

[27] M. Aldana and P. Cluzel, "A natural class of robust networks," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 15, pp. 8710–8714, 2003.

[28] M. Aldana, "Boolean dynamics of networks with scale-free topology," *PHYSICA D*, vol. 185, pp. 45–66, Oct 2003.

[29] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of escherichia coli," *Nature genetics*, vol. 31, pp. 64–68, 2002.

[30] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[31] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, "Superfamilies of evolved and designed networks," *Science*, vol. 303, no. 5663, pp. 1538–1542, 2004.

[32] A. Vázquez, R. Dobrin, D. Sergi, J. P. Eckmann, Z. N. Oltvai, and A. L. Barabási, "The topological relationship between the large-scale attributes and local interaction patterns of complex networks," *Proc. Natl. Acad. Sci. U S A*, vol. 101, December 2004.

[33] K. Klemm and S. Bornholdt, "Topology of biological networks and reliability of information processing," *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 51, pp. 18414–18419, 2005.

[34] R. J. Prill, P. A. Iglesias, and A. Levchenko, "Dynamic properties of network motifs contribute to biological network organization," *PLoS Biol*, vol. 3, no. 11, p. e343, 2005.

[35] P. Dwight Kuo, W. Banzhaf, and A. Leier, "Network topology and the evolution of dynamics in an artificial genetic regulatory network model created by whole genome duplication and divergence," *Biosystems*, vol. 85, no. 3, pp. 177–200, 2006.

[36] P. Kaluza and A. S. Mikhailov, "Evolutionary design of functional networks robust against noise," *Europhysics Letters*, vol. 79, no. 4, p. 48001, 2007.

[37] I. Lodato, S. Boccaletti, and V. Latora, "Synchronization properties of network motifs," *Europhysics Letters*, vol. 78, p. 28001, Apr. 2007.

[38] S. Bornholdt and K. Sneppen, "Neutral mutations and punctuated equilibrium in evolving genetic networks," *Phys. Rev. Lett.*, vol. 81, pp. 236–239, Jul 1998.

[39] S. Bornholdt and K. Sneppen, "Robustness as an evolutionary principle," *Proc. R. Soc. Lond. B*, vol. 267, p. 2281, 2000.

[40] A. Wagner, "Does evolutionary plasticity evolve?," *Evolution*, vol. 50, pp. 1008–1023, 1996.

[41] M. Aldana, E. Balleza, S. Kauffman, and O. Resendiz, "Robustness and evolvability in genetic regulatory networks," *J. Theor. Biol.*, vol. 245, pp. 433–448, 2007.

[42] A. Szejka and B. Drossel, "Evolution of canalizing boolean networks," *Eur. Phys. J. B*, vol. 56, pp. 373–380, 2007.

[43] S. Ciliberti, O. C. Martin, and A. Wagner, "Robustness can evolve gradually in complex regulatory gene networks with varying topology," *PLoS Comp. Biol.*, vol. 3, no. 2, 2007.

[44] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, B. D., and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Mol. Biol. Cell*, vol. 9, pp. 3273–3297, 1998.

[45] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young, "Transcriptional regulatory networks in saccharomyces cerevisiae," *Science*, vol. 298, no. 5594, pp. 799–804, 2002.

[46] K. Klemm and S. Bornholdt, "Stable and unstable attractors in boolean networks," *Phys. Rev. E*, vol. 72, no. 5, p. 055101(R), 2005.

[47] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walte, *Molecular Biology of the Cell*. Garland Science, New York, 1996.

[48] H. Grosshans and W. Filipowicz, "Molecular biology: the expanding world of small rnas," *Nature*, vol. 451, no. 7177, pp. 414–6, 2008.

[49] V. Reinke, "Germline genomics." WormBook, ed. The C. Elegans Research Community, http://www.wormbook.org, 2006.

[50] P. Jaluria, K. Konstantopoulos, M. Betenbaugh, and J. Shiloach, "A perspective on microarrays: current applications, pitfalls, and potential uses," *Microb Cell Fact*, vol. 6, no. 1475-2859 (Electronic), p. 4, 2007.

[51] M. J. Buck and J. D. Lieb, "Chip-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments," *Genomics*, vol. 83, no. 3, pp. 349–60, 2004.

[52] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, "Stochastic Gene Expression in a Single Cell," *Science*, vol. 297, no. 5584, pp. 1183–1186, 2002.

[53] N. Rosenfeld, J. W. Young, U. Alon, P. S. Swain, and M. B. Elowitz, "Gene regulation at the single-cell level," *Science*, vol. 307, no. 5717, pp. 1962–1965, 2005.

[54] I. Golding, J. Paulsson, S. M. Zawilski, and E. C. Cox, "Real-time kinetics of gene activity in individual bacteria," *Cell*, vol. 123, no. 6, pp. 1025–36, 2005.

[55] N. Maheshri and E. K. O'Shea, "Living with noisy genes: how cells function reliably with inherent variability in gene expression," *Annu Rev Biophys Biomol Struct*, vol. 36, no. 1056-8700 (Print), pp. 413–34, 2007.

[56] J. M. Levsky and R. H. Singer, "Fluorescence in situ hybridization: past, present and future," *J Cell Sci*, vol. 116, no. 14, pp. 2833–2838, 2003.

[57] H. Maamar, A. Raj, and D. Dubnau, "Noise in Gene Expression Determines Cell Fate in Bacillus subtilis," *Science*, vol. 317, no. 5837, pp. 526–529, 2007.

[58] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Nature*, vol. 403, no. 6767, pp. 339–42, 2000.

[59] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, pp. 335–338, 2000.

[60] C. V. Rao, D. M. Wolf, and A. P. Arkin, "Control, exploitation and tolerance of intracellular noise," *Nature*, vol. 420, pp. 231–237, 2002.

[61] M. Kaern, T. C. Elston, W. J. Blake, and J. J. Collins, "Stochasticity in gene expression: from theories to phenotypes.," *Nat Rev Genet*, vol. 6, no. 6, pp. 451–64, 2005.

[62] P. S. Swain, M. B. Elowitz, and E. D. Siggia, "Intrinsic and extrinsic contributions to stochasticity in gene expression," *PNAS*, vol. 99, no. 20, pp. 12795–12800, 2002.

[63] I. L. Ross, C. M. Browne, and D. A. Hume, "Transcription of individual genes in eukaryotic cells occurs randomly and infrequently," *Immun. Cell. Biol.*, vol. 72, pp. 177–185, 1994.

[64] H. H. McAdams and A. Arkin, "Stochastic mechanisms in gene expression," *Proc. Natl. Acad. Sci. USA*, vol. 94, no. 3, pp. 814–819, 1997.

[65] A. M. Kierzek, J. Zaim, and P. Zielenkiewicz, "The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression," *J. Biol. Chem.*, vol. 276, no. 11, pp. 8165–8172, 2001.

[66] T. B. Kepler and T. C. Elston, "Stochasticity in transcriptional regulation: Origins, consequences, and mathematical representations," *Biophys. J.*, vol. 81, no. 6, pp. 3116–3136, 2001.

[67] E. M. Ozbudak, M. Thattai, I. Kurtser, A. D. Grossman, and A. van Oudenaarden, "Regulation of noise in the expression of a single gene," *Nat Genet*, vol. 31, no. 1, pp. 69–73, 2002.

[68] M. S. H. Ko, "A stochastic model for gene induction," *Journal of Theoretical Biology*, vol. 153, no. 2, pp. 181–194, 1991.

[69] W. J. Blake, M. Kaern, C. R. Cantor, and J. J. Collins, "Noise in eukaryotic gene expression," *Nature*, vol. 422, no. 6932, pp. 633–637, 2003.

[70] J. M. Raser and E. K. O'Shea, "Control of Stochasticity in Eukaryotic Gene Expression," *Science*, vol. 304, no. 5678, pp. 1811–1814, 2004.

[71] L. H. Hartwell and T. A. Weinert, "Checkpoints: controls that ensure the order of cell cycle events," *Science*, vol. 246, no. 4930, pp. 629–34, 1989.

[72] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, June 1998.

[73] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, p. 509, 1999.

[74] R. Albert and A.-L. Barabasi, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, p. 47, 2002.

[75] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, p. 167, 2003.

[76] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[77] R. Albert, H. Jeong, and A.-L. Barabasi, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[78] D. Ekman, S. Light, A. K. Bjorklund, and A. Elofsson, "What properties characterize the hub proteins of the protein-protein interaction network of saccharomyces cerevisiae?," *Genome Biol*, vol. 7, no. 6, p. R45, 2006.

[79] H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[80] N. Guelzim, S. Bottani, P. Bourgine, and F. Kepes, "Topological and causal structure of the yeast transcriptional regulatory network," *Nat Genet*, vol. 31, no. 1, pp. 60–63, 2002.

[81] S. Maslov and K. Sneppen, "Specificity and stability in topology of protein networks," *Science*, vol. 296, no. 5569, pp. 910–913, 2002.

[82] J. Bonner, *The evolution of complexity.* Princeton University Press, Princeton, NJ., 1988.

[83] M. E. J. Newman, "Detecting community structure in networks," *Eur. Phys. J. B*, vol. 38, pp. 321–330, 2004.

[84] Y. Qi and H. Ge, "Modularity and dynamics of cellular networks.," *PLoS Comput Biol*, vol. 2, no. 12, p. e174, 2006.

[85] R. V. Sole and S. Valverde, "Spontaneous emergence of modularity in cellular networks.," *J R Soc Interface*, vol. 5, no. 18, pp. 129–33, 2008.

[86] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *J. Phys. Chem.*, vol. 81, pp. 2340–2361, December 1977.

[87] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *The Journal of Chemical Physics*, vol. 115, pp. 1716–1733, 2001.

[88] A. Arkin, J. Ross, and H. H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells," *Genetics*, vol. 149, no. 4, pp. 1633–48, 1998.

[89] J. T. Mettetal, D. Muzzey, J. M. Pedraza, E. M. Ozbudak, and A. van Oudenaarden, "Predicting stochastic gene expression dynamics in single cells," *Proc. Natl. Acad. Sci. USA*, vol. 103, no. 19, pp. 7304–7309, 2006.

[90] K. C. Chen, A. Csikasz-Nagy, B. Gyorffy, J. Val, B. Novak, and J. J. Tyson, "Kinetic analysis of a molecular model of the budding yeast cell cycle," *Mol. Biol. Cell*, vol. 11, no. 1, pp. 369–391, 2000.

[91] F. Cross, V. Archambault, M. Miller, and M. Klovstad, "Testing a mathematical model of the yeast cell cycle," *Mol. Biol. Cell*, vol. 13, pp. 52–70, 2002.

[92] K. C. Chen, L. Calzone, A. Csikasz-Nagy, F. R. Cross, B. Novak, and J. J. Tyson, "Integrative analysis of cell cycle control in budding yeast," *Mol Biol Cell*, vol. 15, no. 8, pp. 3841–62, 2004.

[93] A. Sveiczer, A. Csikasz-Nagy, B. Gyorffy, J. Tyson, and B. Novak, "Modeling the fission yeast cell cycle: Quantized cycle times in wee1-cdc25 mutant cells," *Proc. Natl. Acad. Sci. U S A*, vol. 97(14), pp. 7865–7870, 2000.

[94] T. Chen, H. He, and G. Church, "Modeling gene expression with differential equations," *Pac. Symp. Biocomput.*, pp. 29–10, 1999.

[95] N. Holter, A. Maritan, M. Cieplak, N. Fedoroff, and J. Banavar, "Dynamic modeling of gene expression data," *Proc. Natl Acad. Sci. USA.*, vol. 98, pp. 1693–1698, 2001.

[96] W. Bialek, "Stability and noise in biochemical switches," in *NIPS*, pp. 103–109, 2000, arXiv:q-bio/0605004.

[97] J. Hasty, J. Pradines, M. Dolnik, and J. J. Collins, "Noise-based switches and amplifiers for gene expression.," *Proc. Natl. Acad. Sci. U S A*, vol. 97, no. 5, pp. 2075–80, 2000.

[98] R. Thomas, "Boolean formalization of genetic control circuits," *J. Theor. Biol.*, vol. 42, pp. 563–585, 1973.

[99] O. Brandman, J. Ferrell, James E., R. Li, and T. Meyer, "Interlinked Fast and Slow Positive Feedback Loops Drive Reliable Cell Decisions," *Science*, vol. 310, no. 5747, pp. 496–498, 2005.

[100] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle.," *Bioinformatics*, vol. 22, no. 14, pp. e124–31, 2006.

[101] L. Sanchez and D. Thieffry, "A logical analysis of the drosophila gap-gene system," *J Theor Biol*, vol. 211, pp. 115–141, 2001.

[102] A. Rubinstein, V. Gurevich, Z. Kasulin-Boneh, L. Pnueli, Y. Kassir, and R. Y. Pinter, "Faithful modeling of transient expression and its application to elucidating negative feedback regulation," *Proc. Natl. Acad. Sci. USA*, vol. 104, no. 15, pp. 6241–6246, 2007.

[103] Z. Szallasi and S. Liang, "Modeling the normal and neoplastic cell cycle with 'realistic boolean genetic networks': their application for understanding carcinogenesis and assessing therapeutic strategies," *Pac. Symp. Biocomput.*, pp. 66–76, 1998.

[104] A. Chaos, M. Aldena, C. Espinosa-Soto, B. de Leon, G. Arroyo, A, and E. Alvarez-Buylla, "From genes to flower patterns and evolution: Dynamic models of gene regulatory networks," *J. Plant. Growth. Regul.*, vol. 25, pp. 278–289, 2006.

[105] A. Samal and S. Jain, "The regulatory network of e. coli metabolism as a boolean dynamical system exhibits both homeostasis and flexibility of response." arXiv:q-bio/0703059, 2007.

[106] D. Thieffry, M. Colet, and R. Thomas, "Formalization of regulatory networks: a logical method and its automation," *Math. Model Sci. Comput.*, vol. 55, pp. 144–151, 1992.

[107] H. McAdams and L. Shapiro, "Circuit simulation of genetic networks," *Science*, vol. 269, no. 5224, pp. 650–656, 1995.

[108] L. Mendoza, D. Thieffry, and E. Alvarez-Buylla, "Genetic control of flower morphogenesis in arabidopsis thaliana: a logical analysis," *Bioinformatics*, vol. 15, pp. 593–606, 1999.

[109] P. Smolen, D. Baxter, and J. Byrne, "Modeling transcriptional control in gene networks–methods, recent results, and future directions," *Bull. Math. Biol.*, vol. 62, pp. 247–292, 2000b.

[110] P. Smolen, D. A. Baxter, and J. H. Byrne, "Mathematical modeling of gene networks," *Neuron*, vol. 26, pp. 567–580, 2000.

[111] T. Schlitt and A. Brazma, "Current approaches to gene regulatory network modelling.," *BMC Bioinformatics*, vol. 8 Suppl 6, no. 1471-2105 (Electronic), p. S9, 2007.

[112] C. Chaouiya, "Petri net modelling of biological networks," *Brief Bioinform.*, vol. 8, no. 4, p. 210, 2007.

[113] A. Ma'ayan, S. L. Jenkins, S. Neves, A. Hasseldine, E. Grace, B. Dubin-Thaler, N. J. Eungdamrong, G. Weng, P. T. Ram, J. J. Rice, A. Kershenbaum, G. A. Stolovitzky, R. D. Blitzer, and R. Iyengar, "Formation of Regulatory Patterns During Signal Propagation in a Mammalian Cellular Network," *Science*, vol. 309, no. 5737, pp. 1078–1083, 2005.

[114] S. A. Kauffman, *The Origins of Order*. Oxford University Press, New York, 1993.

[115] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, "Genetic networks with canalyzing boolean rules are always stable," *Proc. Natl. Acad. Sci. USA*, vol. 101, p. 17102, 2004.

[116] K. E. Kürten, "Critical phenomena in model neural networks," *Phys. Lett. A*, vol. 129, pp. 157–160, 1988.

[117] K. E. Kürten, "Correspondence between neural threshold networks and kauffman boolean cellular automata," *Journal of Physics A: Mathematical and General*, vol. 21, no. 11, pp. L615–L619, 1988.

[118] B. Derrida and G. Weisbuch, "Evolution of overlaps between configurations in random boolean networks," *J. Physique*, vol. 47, p. 1297, 1986.

[119] H. J. Hilhorst and M. Nijmajer, "On the approach of the stationary state in kauffman's random boolean network," *J. Physique*, vol. 48, pp. 185–191, 1987.

[120] B. Luque and R. V. Solé, "Phase transitions in random networks: Simple analytic determination of critical points," *Phys. Rev. E*, vol. 55, pp. 257–260, Jan 1997.

[121] H. Flyvbjerg, "An order parameter for networks of automata," *J Phys A*, vol. 21, no. 19, p. L955, 1988.

[122] U. Bastolla and G. Parisi, "Closing probabilities in the kauffman model: An annealed computation," *Physica D*, vol. 98, pp. 1–25, 1996.

[123] B. Samuelsson and C. Troein, "Superpolynomial growth in the number of attractors in kauffman networks," *Phys. Rev. Lett.*, vol. 90, p. 098701, Mar 2003.

[124] B. Drossel, T. Mihaljev, and F. Greil, "Number and length of attractors in a critical kauffman model with connectivity one," *Phys. Rev. Lett.*, vol. 94, p. 088701, Mar 2005.

[125] B. Drossel, "Number of attractors in random boolean networks," *Phys. Rev. E*, vol. 72, p. 016110, Jul 2005.

[126] T. Rohlf and S. Bornholdt, "Criticality in random threshold networks: annealed approximation and beyond," *Physica A*, vol. 310, pp. 245–259, 2002.

[127] T. Rohlf, N. Gulbahce, and C. Teuscher, "Damage spreading and criticality in finite random dynamical networks," *Phys. Rev. Lett.*, vol. 99, p. 248701, 2007. cond-mat/0701601.

[128] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks.," *Bioinformatics*, vol. 18, no. 2, pp. 261–74, 2002.

[129] D. Cornforth, D. G. Green, and D. Newth, "Ordered asynchronous processes in multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 204, no. 1-2, pp. 70–82, 2005.

[130] I. Harvey and T. Bossomaier, "Time out of joint: Attractors in asynchronous random boolean networks.," in *Proceedings of the Fourth European Conference on Articial Life*, pp. 67–75, 1997.

[131] E. A. D. Paolo, "Rhythmic and non-rhythmic attractors in asynchronous random boolean networks," *Biosystems*, vol. 59, no. 3, pp. 185–195, 2001.

[132] F. Greil and B. Drossel, "Dynamics of critical kauffman networks under asynchronous stochastic update," *Phys. Rev. Lett.*, vol. 95, no. 4, p. 048701, 2005.

[133] M. Chaves, E. D. Sontag, and R. Albert, "Methods of robustness analysis for boolean models of gene control networks," *Systems Biology, IEE Proceedings*, vol. 153, no. 4, pp. 154–167, 2006.

[134] L. Glass, "Classification of biological networks by their qualitative dynamics.," *J. Theor. Biol.*, vol. 54, no. 1, pp. 85–107, 1975.

[135] H. Hirata, S. Yoshiura, T. Ohtsuka, Y. Bessho, T. Harada, K. Yoshikawa, and R. Kageyama, "Oscillatory expression of the bhlh factor hes1 regulated by a negative feedback loop," *Science*, vol. 298, no. 5594, pp. 840–843, 2002.

[136] M. H. Jensen, K. Sneppen, and G. Tiana., "Sustained oscillations and time delays in gene expression of protein hes1," *FEBS Letters*, vol. 541, Issues 1-3, pp. 176–177, 2003.

[137] P. Nurse, "Universal control mechanism regulating onset of M-phase," *Nature*, vol. 344, pp. 503–508, 1990.

[138] V. Archambault, C. Li, A. Tackett, R. Wasch, B. Chait, M. Rout, and F. Cross, "Genetic and biochemical evaluation of the importance of cdc6 in regulating mitotic exit," *Mol. Biol. Cell*, vol. 14, pp. 4592–4604, 2003.

[139] M. Tyers, "Cell cycle goes global," *Curr. Opin. Cell Biol.*, vol. 16, pp. 602–613, 2004.

[140] G. Stoll, J. Rougemont, and F. Naef, "Few crucial links assure checkpoint efficiency in the yeast cell-cycle network," *Bioinformatics*, vol. 22, no. 20, pp. 2539–46, 2006.

[141] R. Friedberg, "A learning machine," *IBM Journal, Research and Development Part II*, vol. 3, pp. 183–191, 1959.

[142] G. P. Wagner and L. Altenberg, "Complex adaptations and the evolution of evolvability," *Evolution*, vol. 50, no. 3, pp. 967–976, 1996.

[143] S. A. Kauffman and R. G. Smith, "Adaptive automata based on darwinian selection," *Physica D*, vol. 22, pp. 68–82, 1986.

[144] M. D. Stern, "Emergence of homeostasis and "noise imprinting" in an evolution model," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 19, pp. 10746–10751, 1999.

[145] N. Kashtan and U. Alon, "Spontaneous evolution of modularity and network motifs," *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 39, pp. 13773–13778, 2005.

[146] P. Oikonomou and P. Cluzel, "Effects of topology on network evolution," *Nat Phys*, vol. 2, no. 8, pp. 532–536, 2006.

[147] K. Kaneko, "Evolution of robustness to noise and mutation in gene expression dynamics.," *PLoS ONE*, vol. 2, no. 5, p. e434, 2007.

[148] S. Ohno, *Evolution by gene duplication.* Springer-Verlag, New York, 1970.

[149] Z. J., "Evolution by gene duplication: an update," *Trends in Ecology and Evolution*, vol. 18, pp. 292–298(7), June 2003.

[150] T. Reil, "Dynamics of gene expression in an artificial genome – implications for biological and artificial ontogeny," *Advances in Artificial Life, Proceedings, Lecture Notes in Artificial Intelligence*, vol. 1674, pp. 457–466, 1999.

[151] A. P. Quayle and S. Bullock, "Modelling the evolution of genetic regulatory networks," *J. Theor. Biol.*, vol. 238, no. 4, pp. 737–753, 2006.

[152] S. Mangan, A. Zaslaver, and U. Alon, "The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks," *Journal of Molecular Biology*, vol. 334, pp. 197–204(8), 21 November 2003.

[153] A. Mazurie, S. Bottani, and M. Vergassola, "An evolutionary and functional assessment of regulatory network motifs," *Genome Biology*, vol. 6, no. 4, p. R35, 2005.

[154] A. Wuensche, *Artificial Life III*, ch. The Ghost in the Machine: Basins of Attraction of Random Boolean Networks. Addison-Wesley, Reading, MA, 1994.

[155] J. Norrell, B. Samuelsson, and J. E. S. Socolar, "Attractors in continuous and boolean networks," *Phys. Rev. E*, vol. 76, no. 4, p. 046122, 2007.

[156] G. Tononi, O. Sporns, and G. M. Edelman, "Measures of degeneracy and redundancy in biological networks," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 6, pp. 3257–3262, 1999.

[157] A. Wagner, "Distributed robustness versus redundancy as causes of mutational robustness," *Bioessays*, vol. 27, no. 2, pp. 176–88, 2005.