15

COMPUTING WITH ARTIFICIAL GENE REGULATORY NETWORKS

Michael A. Lones

School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, Scotland, UK

15.1 INTRODUCTION

Gene regulatory networks (GRNs) are the fundamental mechanisms through which biological organisms control their growth, their dynamical behavior, their interaction with their environment, and which underlie much of the complexity we see in the biosphere. Biological complexity has long been an inspiration to computer scientists, and many seek to model the biological processes that generate this complexity, using these models to generate complex behavior that can then be used to solve problems in computer science and engineering. In this chapter, the term "artificial gene regulatory network" (AGRN) is used to refer to computational models of GRNs that are used in this manner.

Unlike other computational models of GRNs, AGRNs are predominantly used to solve computational and engineering problems, not biological problems. Hence, AGRNs often differ from these other models. This chapter reviews current understanding of AGRNs, discussing what is known about their computational properties, detailing how they have previously been applied to computational problems, and speculating about how they may be used in the future. Reflecting the theme explored in this book, we focus on approaches that have used evolutionary algorithms (EAs)

Evolutionary Computation in Gene Regulatory Network Research, First Edition.
Edited by Hitoshi Iba and Nasimul Noman.

to design AGRNs, a task for which they are presumably well suited given the close relationship between biological evolution and biological GRNs.

The chapter is organized as follows. Section 15.2 discusses what we know about biological GRNs, and the implications this has for the design of AGRNs. Section 15.3 presents the different motivations behind the development of AGRN models. Section 15.4 discusses the modeling decisions that have to be made when developing AGRN models. Sections 15.5 and 15.6 review previous work on using AGRNs for computation, focusing on theoretical and applied perspectives, respectively. Section 15.7 considers future directions for AGRN research, and Section 15.8 concludes.

15.2 BIOLOGICAL GRNs

Before discussing computational models of GRNs, it is instructive to first consider the system, which is being modeled—biological GRNs. The key feature of biological GRNs, and one that is captured in all computational models, is that a group of genes regulate one another's expression. However, something that is not often captured is the considerable heterogeneity and complexity of these regulatory processes. In a eukaryotic cell, all the following factors play important roles in regulating a gene's expression:

Gene accessibility Deoxyribonucleic acid (DNA) is supercoiled within a protein complex known as chromatin. Genes may only be expressed if they are accessible to the cellular machinery that carries out transcription, and this is only possible in regions where the chromatin-bound DNA has been unwound. Winding and unwinding, in turn, are subject to regulation by proteins expressed by other genes. These act in a number of different ways, for instance, through epigenetic processes such as DNA methylation [35].

Transcription preinitiation The transcription of all genes involves a core collection of proteins, which includes ribonucleic acid (RNA) polymerase and a group of stabilizing general transcription factors (TFs) that assemble at a gene's transcription start site (TSS). All these proteins are required to achieve a baseline level of transcription. Since they are finite in number, the transcription rate of a gene is also affected by the number and transcription rates of genes elsewhere within the cell's nucleus [24, 67].

Transcription initiation The preinitiation complex is insufficient in most cases, and is supplemented by special-purpose TFs that bind to regulatory regions upstream of the TSS, and act to up- or down-regulate the gene by either increasing or decreasing the stability of the transcription complex. TFs are proteins and, hence, the products of other genes. The number of TFs involved, and the complexity of their interactions, varies considerably between genes [75].

Transcription elongation Initiation is just the first stage of transcription. Regulation also takes place during transcription, typically through RNA-binding proteins that interfere with the progress or stability of the transcription complex. Again, these are the products of other genes [47].

Post-transcriptional modification Once a gene has been transcribed, the resulting strand of messenger RNA is acted upon by other proteins in order to form a mature transcript that is ready for translation. Regulation of this process is possible at all stages. Most prominent is the process of alternative splicing, where proteins differentially cleave away parts of the transcript, leading to the expression of different variants of the encoded protein [28].

Gene silencing During the last decade, it has become evident that RNA plays an important role in gene regulation. This process, known as gene silencing or RNA interference, involves microRNAs that are encoded by other genes, and which interfere with translation by binding to messenger RNA transcripts. This binding process remains poorly understood, but it is known that the specificity of individual microRNAs varies considerably, resulting in effects ranging from the silencing of individual genes to individual pathways to large-scale changes in gene expression [11].

Transcription and post-transcription Regulatory interactions also occur during protein synthesis, transportation, and degradation. An example of this is the regulation of the availability of chaperones, helper proteins that are required to fold certain proteins into their mature, functional, form [33, 41].

Metabolic availability The cellular machinery, including those sub-systems responsible for translation and transcription, requires various metabolites. For example, transcription requires a ready supply of transfer RNA and its associated amino acids. The availability of these metabolites places a global constraint on the cell's activities, and may also be regulated at a more local level [16].

These regulatory processes occur on different temporal and spatial levels. Hence, it is unlikely that a model operating on a single temporal or spatial level could accurately capture the overall behavior of a biological GRN. Furthermore, it is important to note that biological GRNs are only partially understood. It is quite likely that our knowledge of them is biased toward those components which are easiest to recognize and model, and quite possible that we do not yet understand all the fundamental mechanisms underlying their behavior. For example, the involvement of microRNA in genetic regulation is a relatively recent discovery, and understanding of its biochemical interactions remains incomplete. Nevertheless, although this incomplete understanding is an issue for research into AGRNs, it is also an opportunity, since the behavior of AGRNs could provide valuable insights into the working of their biological analogs.

15.3 COMPUTATIONAL MODELS

Historically, the development of computational models of GRNs has focused on their role in understanding biological systems. Typical uses include inferring computational models from measurement data in order to determine the structure of a biological GRN, simulating computational models in order to understand the

dynamics of a biological GRN, and using computational models to understand systems-level properties of biological pathways. However, for the area of research considered in this chapter, the justification for using computational models of GRNs is quite different. Rather than trying to understand how actual biological circuits work, the aim is to apply the principles of biological circuits to the design of computational and engineered systems. Typical motivations for this include the following:

Structures Biological systems are structurally complex, and this complexity is known to be a direct consequence of the behavior of GRNs. This has led to an interest in how computational models of GRNs can be used to generate intricate structures [9, 12, 13, 15, 21, 29, 36, 37, 43, 68, 72, 77, 81], something that is of interest to both engineers and people working in creative disciplines.

Dynamics Biological systems are also dynamically complex. They respond robustly and intelligently to their environment, and this is known to be a direct consequence of the signal processing and control behaviors exhibited by biological circuits. Hence, there is an interest in how computational models of GRNs can be used to perform these functions within engineered systems [5, 8, 14, 25, 26, 38, 45, 46, 52, 54, 56, 59, 61, 64, 73, 78].

Computability Biological systems appear to process information and carry out behaviors that are analogous to computation. However, the manner in which they achieve this is quite different to conventional computers. This has promoted interest in studying the computational characteristics of GRN models [7, 10, 31, 40, 51, 58]; for instance, their apparent ability to represent complex computations in a compact and robust form.

Evolvability Evolutionary algorithms have become a successful approach for carrying out design and optimization. They are modeled upon biological evolution. Since GRNs can be considered the "solution representation" used by biological evolution, and they are known to exhibit characteristics that make them evolvable, this has led to interest in using computational models of GRNs to represent artifacts (e.g., computer programs) evolved by EAs [4, 10, 53].

The diverse motivations for computational models of GRNs have led to an equally diverse range of computational models. Some of these, such as S-systems [66], are used exclusively for biological modeling. Some, such as our work on artificial biochemical networks (ABNs) [53, 55], are intended exclusively for computational applications. Others, such as Boolean networks, have been used at both ends of this spectrum [10, 15, 39, 64]. While it is beyond the scope of this chapter to review all the modeling approaches used in the literature (various examples can be seen elsewhere in this book), it is important to be aware that there are diverse models, that different models are appropriate for solving different problems, and, moreover, that we do not necessarily know which is the best model for a particular task. For applications in computer science and engineering, the latter point reflects both the infancy of the field and the incompleteness of biological knowledge.

15.4 MODELING DECISIONS

We know that biological GRNs are capable of generating complex behavior, but it is less clear which elements of biological GRNs are important or even necessary for achieving this. This uncertainty is reflected in the array of AGRN models that have been used in computational applications [4, 8, 15, 55, 81]. These vary considerably in the level of abstraction at which they model regulatory networks, and also vary in the accuracy with which they model individual biological processes and mechanisms. Before going on to talk in detail about these models and their applications, we briefly summarize some of the decisions faced when doing this kind of modeling, and the choices available.

Expression levels In biological systems, gene expression can take place at many different rates depending on the stability of the transcription complex and the presence of downstream regulatory processes. Hence, it may seem biologically accurate to capture gene expression as a continuous-valued state. Nevertheless, for many regulatory circuits it is known that the qualitative dynamics can be approximated by modeling gene expression as an on/off process [1], meaning that it is also feasible to capture gene expression as a binary state. While the consequences of choosing binary or continuous-valued expression levels are not fully understood, it is likely to represent a trade-off between model expressiveness and model simplicity, and the implementation consequences of the latter (for instance, ease of implementation in hardware [82]). This is a rather fundamental modeling choice, since the range of states determines the choice of regulatory functions. In practice, both decisions have been taken.

Regulatory functions Regulatory functions can be captured in a number of different ways. Most common is to use an additive or integrative function to combine the expression levels of the gene's regulatory inputs [5, 63]. This function may be weighted, so that some inputs are more influential than others. Inputs may also be negative weighted in order to represent negative regulatory interactions between genes. The regulatory function may be linear (e.g., simple addition) or nonlinear (e.g., sigmoidal). Some AGRN models (notably Boolean networks) use combinatorial functions, which are a more accurate model of many biological regulatory functions [80]. In particular, these allow logical and canalizing interactions between genes to be captured. However, in the general case, gene interactions are neither additive nor combinatorial. Rather, they are a result of complex physical interactions between many different biochemicals that occur over many different timescales. It is rarely feasible to model this process in detail, though some AGRN models attempt to capture notions of its complexity by using various kinds of complex nonlinear functions [8, 53].

Gene products The process by which one gene regulates another can be captured at differing levels of abstraction. From a high-level perspective, genes can be viewed as directly regulating one another, and indeed this is a simplification often used by biologists when modeling gene regulation [40]. A number of

AGRN models capture this direct interaction, with the expression level of a regulating gene becoming an input of the regulated gene's regulatory function [40, 52]. Other approaches explicitly model the intermediate gene products involved in gene regulation [4, 8]. Typically, this is done using analogs of TFs, using an additive or integrative function to combine the expression levels of all genes which are tagged as expressing a particular TF. The resulting value, which is an analog of chemical concentration, is then used as an input to the regulatory functions of genes which are tagged as being regulated by this TF. In one sense, these two representations are equivalent, since indirect regulatory interactions can be rewritten as direct interactions, following a suitable change to the regulatory function. Nevertheless, some authors have argued that an indirect representation brings advantages in terms of evolvability [5]. However, the explicit modeling of gene products has a computational overhead, and also reduces the apparent simplicity of the model.

Genome organization From an abstract perspective, a genome is an array of genes. In reflection of this, it is commonplace to encode an AGRN in an array structure when using an EA to optimize it. However, viewed through the lens of natural evolution, biological systems are not static entities. Rather, they are dynamically changing over the course of evolutionary history, and this is reflected in their genome, which is organized in a way that promotes meaningful change [42]. Since we are generating AGRNs using an evolutionary approach, it also makes sense to consider how elements of this biological evolvability may be captured within these models. Reflecting this, a number of modelers have considered the role of non-coding elements, and the manner in which biological complexity grows through a process of duplication and divergence [20, 81].

Time and space Another important modeling choice is how to capture temporal and spatial aspects of gene regulation. Time, in biological systems, is continuous. Genes do not express proteins synchronously, or even necessarily over the same timescales. This means that biological systems are able to make use of processes happening at different timescales, and this may be important for generating certain dynamical patterns. This kind of temporal variation can be captured through continuous models, such as systems of differential equations [5]. However, the simulation of differential equations is computationally demanding, especially when compared to discrete models, such as difference equations. Hence, in practice, it may be preferable to approximate continuous dynamics using discrete models, and this design decision is seen in many AGRNs [8, 52, 81]. Spatial organization is also an important factor in the behavior of biological GRNs, including its role in transcription complex formation, the role of DNA conformation and intergenic distances, and spatial processes such as diffusion and crosstalk. However, simulation of spatial processes is often computationally demanding. For this reason, it is often not considered in AGRNs, and when it is, it often appears in the form of a behavioral approximation [26].

Wider biological interactions GRNs do not exist within a vacuum. They interact with wider cellular and biological systems, and in many cases it is difficult, and arguably not useful, to delineate GRNs from their interactions with other systems. For example, the cell's GRN is closely integrated with its signaling and metabolic networks. It is also closely integrated with higher-level processes such as cell division. This has led to a number of models in which the AGRN is either generalized to model a broader system [53, 55], or hybridized with other forms of computational model, such as cellular automata (CA). The latter is especially commonplace in artificial development models, where the aim is often to capture a biological GRN's role in forming spatial structures [15, 21].

15.5 COMPUTATIONAL PROPERTIES OF AGRNs

Given the complexity of biological systems, and the key role that GRNs play in achieving this complexity, we expect AGRNs to show interesting computational properties. Nevertheless, we are still in the early stages of understanding the computational abilities of these models, and conversely their computational limits. This is for several reasons. First, historically there has not been much focus on this question, since most work on computational models of GRNs has been associated with biological understanding rather than computational understanding. Second, as the previous section attests, AGRN models are very diverse, making it unclear how computational understanding of one model applies to others. Furthermore, many AGRN models are descriptively complex, having many parameters, and therefore exhibiting many degrees of freedom. This makes it infeasible to exhaustively explore, or even sample with reasonable coverage, the space of possible networks, and therefore makes experimental investigations of their behavioral space intractable. Hence, our insight into their computational potential mostly comes from testing whether they can be optimized to carry out specific computational tasks. While sometimes revealing, this depends upon the suitability of the model for being optimized, making it difficult to distinguish between those behaviors that are theoretically possible and those that happen to be reachable using a particular optimization algorithm.

Nevertheless, some understanding of their computational potential can be uncovered by looking at the properties of relatively simple AGRN models, and their relationships to other, better understood, computational models. This is typically done using Boolean networks [39] (see Figure 15.1). These are a relatively simple and relatively abstract model of biological GRNs. Despite this, they have found uses in biological simulation [1], studies of systems-level understanding [2], and as AGRNs applied to computational applications [10, 64]. The model reduces gene expression to two values: true (expressed) and false (not expressed). Regulatory functions are then modeled as Boolean functions, such as AND, OR, and NOT, and gene expression updates are normally performed synchronously and deterministically.

The relationship between Boolean networks and CAs is important for understanding their computational potential. In particular, they are a generalization of elementary CAs (i.e., one-dimensional binary CAs), since the latter can be readily implemented

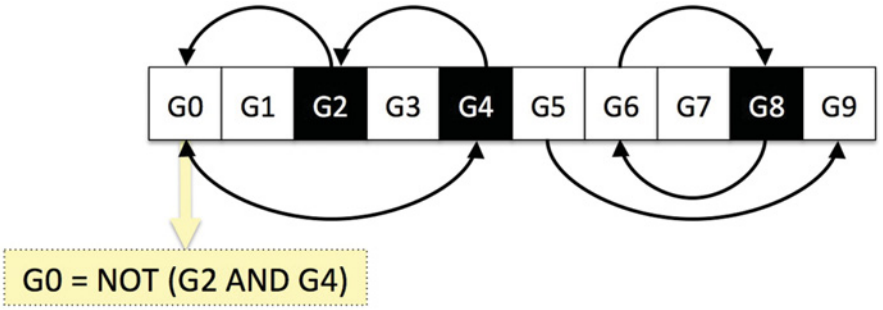


Figure 15.1 A Boolean network. Each gene has a binary state (shown in black or white), inputs from other genes, and a Boolean regulatory function (shown for G0) that determines its next state based upon the states of its inputs.

as a Boolean network by limiting connections to immediately neighboring genes and by using a single Boolean function for every gene [27]. It is well known that the use of certain update rules (such as Wolfram's rule 110 [76]) allow elementary CAs to simulate a Turing machine, and it follows from this that Boolean networks must likewise be capable of universal computation. Nevertheless, it should be taken into consideration that while assuring us that Boolean networks are computationally expressive, Turing completeness only tells us that there exists a Boolean network that can implement a particular computation. It does not tell us how small this network is, or how easy it is to construct, both of which are important in a practical sense.

In Ref. [58], the authors showed that Boolean networks perform better than CAs on tasks that measure the ability of distributed architectures to achieve global behaviors. Other work has given insights into the computational properties of particular kinds of Boolean network. For example, in Ref. [51], the authors noted that networks with a small-world topology exhibit a beneficial trade-off between information storage and propagation. In Ref. [40], it was hypothesized that computational abilities were maximized in the critical region between order and chaos, showing this to be the case when Boolean networks have an average connectivity of two inputs per gene. Various studies have observed a similar relationship between criticality and connectivity within evolved Boolean networks [31]. The relationship between criticality and evolvability in Boolean networks has also been highlighted [70]. Robustness is also an important issue, and it has been shown that certain classes of Boolean network can be strongly resistant to perturbations [3]. However, it has also been observed that changes to the Boolean network model can have a large impact upon its behavior. For instance, the introduction of stochastic or probabilistic elements considerably changes the appearance and stability of attractors [27]. The choice of functions and topologies also has significant consequences [32]. This suggests that care must be taken when trying to generalize these computational results to a wider range of AGRN models.

Various studies have considered the number and size of attractors found in Boolean network models [18]. These may give an indication of the computational capacity of a

particular network, and are of significant interest to researchers who use Boolean networks to study biological systems [34]. However, the relationship between attractors and computation remains unclear. In part, this may depend upon whether attractors are explicitly used during computation. For instance, if the inputs to a problem are encoded in the initial expression state of the network, the attractor to which the network converges can be used to signal a particular computational outcome. For cyclic attractors, the repeating series of states, which the network visits, may also be interpreted as a computational outcome; for example, in a robotics application, the series of expressions at particular nodes may be read as a pattern of actuator activations. Nevertheless, in these situations, the computation is arguably being performed by the transients rather than the attractors, since it is these that *compute* the attractors from the initial state (or from other attractors if inputs are introduced during execution).

15.6 AGRN MODELS AND APPLICATIONS

Knowledge of theoretical computational properties is important, since it can guide us in choosing an appropriate model for a particular task. However, from an engineering perspective, we might not care about whether or not a model is computationally universal, so long as it can be used to effectively and efficiently solve a particular task. In this respect, previous experimental results are an important indicator of future potential. In this section, we review some of the work done on developing AGRN models and applying them to real-world computational problems, and consider some of the perceived strengths and weaknesses of the different approaches.

The most popular application area for AGRNs is arguably in the development of control systems, mirroring one of the prominent roles of biological GRNs. Table 15.1 summarizes a number of previous contributions in this area. The majority of these involve controlling robots or other autonomous agents, performing tasks such as obstacle avoidance, target following, foraging, multirobot coordination, and legged robot locomotion. A number of researchers have also tackled classic control problems, notably several variants of the inverted pendulum control task. However, despite this apparent focus on control, AGRNs have also been applied to other problems. Table 15.2 lists some of these. Many of these applications are motivated by the observed ability of GRNs to generate complex patterns, both static and dynamically varying, and how these patterns can be mapped to physical or computational structures. For example, a popular application of this approach is in the design of electronic circuits [29, 81].

In this section, we discuss some of the AGRN models that have been used in applied work. We begin with Boolean networks. We then discuss artificial genome models, a group of AGRN models that developed from research in evolutionary computing and artificial life. Following this, we review work in the area of artificial development, where AGRNs are predominantly used for pattern generation. Then we discuss the fractal GRN, an AGRN model that attempts to capture the complexity of biochemical interactions. Finally, we discuss our own work on extending AGRN models to include wider biochemical interactions.

Table 15.1 Selected control applications of AGRNs

First author, Year [Ref.]	Problem	GRN model
Dellaert, 1996 [15]	Obstacle avoidance and path following tasks	Boolean network, developmental
Quick, 2003 [61]	Temperature regulation and robotic control	Artificial genome, discrete time, continuous valued
Bentley, 2004 [8]	High-level control of a simulated hexapod robot	Fractal GRN [8]
Kumar, 2004 [46]	Obstacle avoidance for simulated and real robots	Artificial genome, discrete time, continuous valued
Taylor, 2004 [68]	Simple patterning of an underwater robotic swarm	Artificial genome based on Ref. [63]
Lee, 2009 [48]	Inference of robotic controllers from data	RNN-based model
Joachimczak, 2010 [38]	Control of simulated foraging behavior	Artificial genome, operons
Krohn, 2010 [45]	Single and joint inverted pendulum problems	Fractal GRN [8]
Lones, 2010 [52]	Chaos control in numerical dynamical systems	Artificial biochemical networks
Nicolau, 2010 [59]	Control of an inverted pendulum	Artificial genome, based on Ref. [5] with minor changes
Trefzer, 2010 [71]	Obstacle avoidance in a simulated and real robot	Artificial genome, discrete
Zahadat, 2010 [78]	Control of a modular robot	Fractal GRN [8]
Lones, 2011 [56]	Locomotion of a simulated quadruped robot	Coupled artificial biochemical networks
Roli, 2011 [64]	Light following and avoidance in robots	Boolean network
Cussat-Blanc, 2012 [14]	Controller for intelligent agents in a video game	See Ref. [59]
Jin, 2012 [36]	Control of robotic swarm patterning and formation	Developmental, hierarchical model with diffusion
Joachimczak, 2012 [37]	Control of soft-bodied multicellular animats	Developmental, based on Ref. [38]
Fuente, 2013 [25]	Locomotion of quadruped robot on rough terrain	Artificial signaling network
Turner, 2013 [73]	Control of coupled inverted pendulums	Artificial epigenetic network
Yao, 2014 [77]	Control of robot swarm in dynamic environment	Developmental, condition-dependent model

GRN, gene regulatory network; AGRN, artificial gene regulatory network; RNN, recurrent neural network.

15.6.1 Boolean Networks

There have been a number of studies in which Boolean networks have been used to solve computational problems [10, 19, 31, 60, 64, 79]. An example of this is the work of Roli et al., who have explored the use of Boolean networks in robotic control. In

Table 15.2 Other applications of AGRNs

First author, Year [Ref.]	Problem	GRN model
Gordon, 2003 [29]	Digital circuit design	Artificial genome, developmental, discrete
Koopman, 2004 [43]	Hardware fault tolerance	Artificial genome, developmental, discrete
Bentley, 2005 [7]	Fault-tolerant code	Fractal GRN [8]
Bowers, 2005 [9]	Pattern generation	Artificial genome, developmental
Gordon, 2005 [30]	Scalable digital circuits	Artificial genome, developmental, discrete
Mattiussi, 2007 [57]	Analog circuit design	Artificial genome
Bull, 2009 [10]	Multiplexer design	Boolean networks
Zhan, 2009 [81]	Digital circuit design	Artificial genome, discrete
Liu, 2010 [50]	System on chip reconfiguration	See Ref. [71]
Trefzer, 2010 [72]	Image compression	See Ref. [71]
Cussat-Blanc, 2012 [13]	Evolutionary art	See Ref. [59]
Lones, 2013 [54]	Diagnosis of Parkinson's disease	Artificial biochemical networks

GRN, gene regulatory network; AGRN, artificial gene regulatory network.

Ref. [64], for instance, they describe a problem in which a robot must move toward or away from a light source, switching between these two behaviors based upon the perception of a sound. By optimizing a Boolean network, they were able to solve this problem in simulation and then use this solution on a physical robot. The ability to transfer a solution directly from a simulated environment to a real-world environment is particularly notable, and differs from the experience of many working in the field of evolutionary robotics, where it is commonplace for evolved solutions to overfit the simulation environment [44]. This may be a reflection of the way in which GRNs work, that is, by following robust attractors, rather than using arguably less robust mechanisms such as decision rules. This is also reflected in the authors' analysis of the controllers, which they found to be robust to external interference [64].

Another notable use of Boolean networks within an evolutionary context is the work of Bull et al. [10], who used this approach to optimize digital circuits, such as multiplexers. However, compared to the other models discussed in this section, there has been relatively little applied work using Boolean networks. In part, this may be due to the inherent restrictions of working with binary values. Although Boolean logic is efficient and can be readily implemented in hardware, the requirement for inputs and outputs to be binary encoded can be problematic. This is especially the case when there are many inputs and outputs to encode, or where each input or output requires a large number of bits (e.g., a floating-point number), since this requires either a correspondingly large number of input and output nodes, or for inputs to be delivered over a number of time steps.

There is also some concern regarding the accuracy with which particular behaviors can be evolved. For instance, while Boolean networks are able to model the qualitative

dynamics of many biological regulatory circuits, it has been recognized that they are not always able to capture the quantitative dynamics. This has prompted some authors to suggest the use of multivalued logics [22], which have been shown better able to capture these aspects of biological circuits. Nevertheless, it should be noted that the architectural simplicity of Boolean networks is a substantial advantage, since it promotes models that are relatively easy to comprehend, implement, and analyze.

15.6.2 Artificial Genome Models

Within the EA and artificial life communities, there has been significant interest in using AGRN models that capture a more detailed view of biological GRNs. An early example of this approach is the work by Riel [63], whose "artificial genome" (illustrated in Figure 15.2) introduced explicit analogs of genome organization and TFs. In this model, a genome is represented as a string of characters, analogous to DNA bases. Genes can be identified in this string by a sequence of characters that represent a transcription initiation site, with the sub-string immediately to the right of this sequence representing a TF and sub-strings to the left representing TF binding sites. It is possible for multiple genes to encode the same TF, and this leads to higher concentrations. Through a process of string matching, TFs are matched to regulatory elements, and the degree of this match, in tandem with the concentration of the TF, determines the gene's expression level. This model captures a number of elements that are missing from Boolean networks, such as the manner in which genetic sequences evolve, the indirect nature of gene interactions, and the presence of

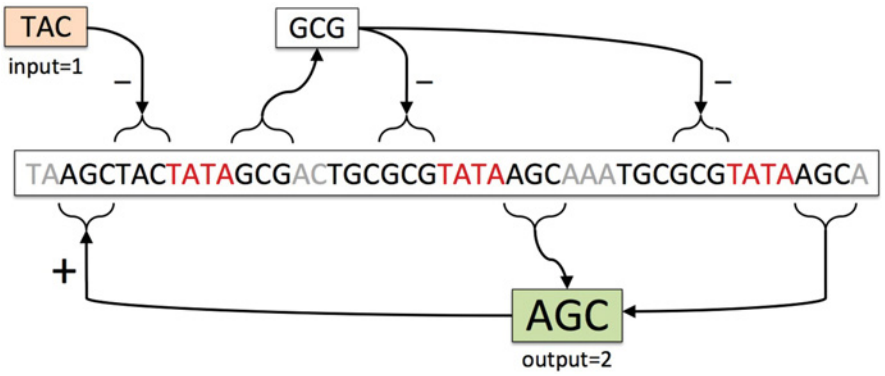


Figure 15.2 Example of an artificial genome model, loosely based on the model described in Ref. [63]. Genes are identified by the pattern TATA. The three letters to the right of this are interpreted as a gene product, and the six to the left are the regulatory region, divided into an activator and a repressor. Gray letters are comparable to non-coding DNA. Note that two genes produce the transcription factor AGC. Assuming all genes are initially expressed at the same level, this will lead to a relatively high concentration of AGC. Note also how external inputs can be encoded in the concentrations of transcription factors (TAC in this example) and how external outputs can be read from the concentrations of designated gene products (AGC).

differential expression levels. However, it does not model the combinatorial nature of gene regulation that is captured by Boolean networks, since regulation is implemented additively.

Reil originally used this model to study the dynamics of regulatory networks. The artificial genome approach was then adopted by Banzhaf [4, 5], who used a continuous-time variant and applied it to a number of computational problems. Particularly notable is the work described in Ref. [59], where the authors successfully evolved AGRNs that could control an inverted pendulum. This is a common benchmark in the control systems literature, and hence an important proof of concept. However, the authors did note that simulation time was an issue, a factor that is likely exacerbated by the use of a continuous-time model.

In more recent work, Cussat-Blanc et al. [14] applied a similar AGRN model to the problem of designing intelligent agents for use in video games and, in Ref. [65], analyzed the properties of controllers evolved to drive a virtual racing car. In common with Roli et al. [64], they found the controllers to be robust to added sensor noise, despite being evolved in the absence of noise. Again, this suggests that AGRN models are naturally robust, and that they capture at least some of the important properties of biological GRNs.

15.6.3 Artificial Development

Many of the earliest AGRN models emerged from the field of artificial development [15,21]. Central to this approach is the use of AGRNs to generate structures. Typically, this is achieved using gene products (which may or may not also be TFs) that encode various kinds of developmental signals. These signals become expressed in a certain pattern over the course of the AGRN's execution, and this pattern is then interpreted by a hybridized cellular growth model in such a way that a structure is produced. In addition, these models often capture spatial aspects of gene expression, such as diffusion, that allow communication between different parts of the developing structure; see Figure 15.3 for a simple example.

An early application of Boolean networks involved using this approach to generate the structure of a neurocontroller, which was then used to control the trajectory of an autonomous agent [15]. More recently, AGRNs have been used to generate ongoing dynamical behavior in addition to generating a structure. A good example of this is the work by Joachimczak et al. [37], who used AGRNs to both generate the morphology of a soft-bodied simulated creature and then control its locomotion by manipulating springs embedded within the resulting structure. Ideas such as this have also been extended to the case of controlling interactions between multiple autonomous elements. A prominent example is the work of Jin et al., who used AGRNs to develop patterns, which were then used to configure robot swarms [36] and modular robotic systems [77].

Another popular application area for developmental AGRN models is within the field of evolvable hardware, where EAs are used to design electronic systems [30, 43, 50, 81]. Often this involves evaluating evolved solutions directly within

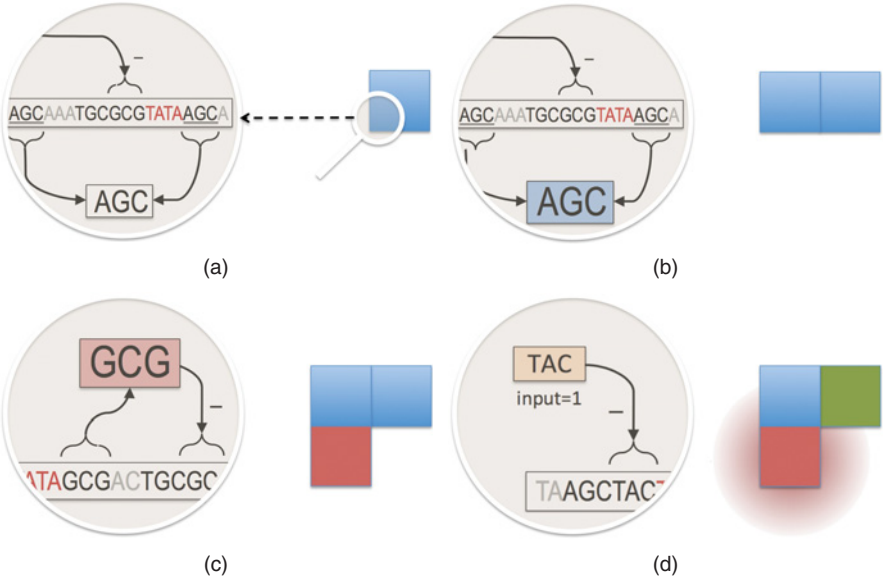


Figure 15.3 Example of how the AGRN from Figure 15.2 might be used to carry out artificial development: (a) maternal cell containing an AGRN; (b) when a designated gene product reaches a certain concentration, the cell divides; (c) different gene products can cause growth in different directions and using different cell types; and (d) AGRNs in different cells can affect each other using diffusive gene products.

reconfigurable hardware devices, and this requires AGRN models that are suitable to be efficiently implemented in this medium. Typically, this involves constraints on the complexity of the regulatory function, and the range and precision of numerical values. Elements of the patterns produced by these AGRNs can then be interpreted as electronic components, allowing them to be used for circuit design. Something that has been widely explored is the ability of developmental AGRNs to produce scalable generalized solutions, such as n-bit adders [30]. Another important aspect is the ability of the AGRN to perform ongoing dynamic behavior, allowing circuits to regenerate in the presence of faults [43], or to allow intelligent reconfiguration in system-on-chip devices [50]. Again, this makes use of the robustness present within AGRN models.

An interesting application outside of autonomous systems control and evolvable hardware is the work of Trefzer et al. [72], who used a developmental AGRN model to perform image compression. This worked by evolving the model so that it generates a time series of 8×8 pixel images, which are then used as a codebook for performing JPEG-like compression. This allows an image to be encoded as an AGRN alongside a sequence of time indices corresponding to elements of the codebook. The authors found that this produced a higher level of compression than JPEG while maintaining a similar level of quality.

15.6.4 Fractal Gene Regulatory Networks

The artificial genome and artificial development models described above capture the complexity of genome evolution and the role of intermediate biochemicals in gene regulation. However, they are arguably less successful at capturing the complexity of regulatory functions. Bentley [8] describes an interesting variant that goes some way toward addressing this. In this model, termed a fractal GRN, regulatory functions are implemented by interactions between regions of the Mandelbrot set. The intention behind this is to capture the degree of complexity of physico-chemical interactions between biochemicals without the inefficiency of implementing actual physico-chemical models. In a certain respect, this brings together AGRN models with the field of artificial chemistry [17], which aims to capture the expressiveness of chemical interactions within a purely computational or mathematical system.

Bentley [8] applied this model to a robotic control task that involved guiding a physical hexapod robot around a maze-like environment. Notably, the evolved controllers exhibited repeating nonlinear patterns of movement, which the author attributed to an implicit modularity present within the system. In another work [7], the author evolved fractal GRNs that could approximate the square root function. Significantly, it was noted that only AGRNs with a fractal chemistry could implement the requisite behavior: those without a complex regulatory function appeared unable to generate sufficient nonlinearity in their responses. Solutions to this problem were also found to be robust, degrading gracefully when intentionally damaged. Other authors have also applied fractal GRNs to control problems, including inverted pendulums in Ref. [45] and robotic modules in Ref. [78].

15.6.5 Artificial Biochemical Networks

In biological systems, genetic networks are closely coupled to other biochemical networks, and arguably it is the interaction between these different types of network that underlies the complexity seen within higher organisms. Our own work [25, 52–56, 73] focuses on modeling these wider interactions, and for this reason we refer to our models using the more generic term artificial biochemical network (or ABN). Our work is particularly motivated by the ability of biological systems to both express dynamics that are complex and diverse, and to switch robustly and sensitively between these dynamics in response to changes in their environment. This is similar to the situation faced in many real-world control tasks, where there is a need for controllers that can intelligently be transitioned between different robust control strategies in response to changes in the controlled system.

There are three main classes of biochemical network within a biological cell: genetic, metabolic, and signaling. The metabolic network, which emerges from interactions between enzyme-mediated reactions, determines the functional and biochemical properties of a cell, that is, the cell type. Since enzymes are gene products, there is a constructional relationship between the genetic and the metabolic network, with the former determining the make-up of the latter. Because of this, it is believed that transitions between the attractors of a GRN can lead to transitions between cell types [40].

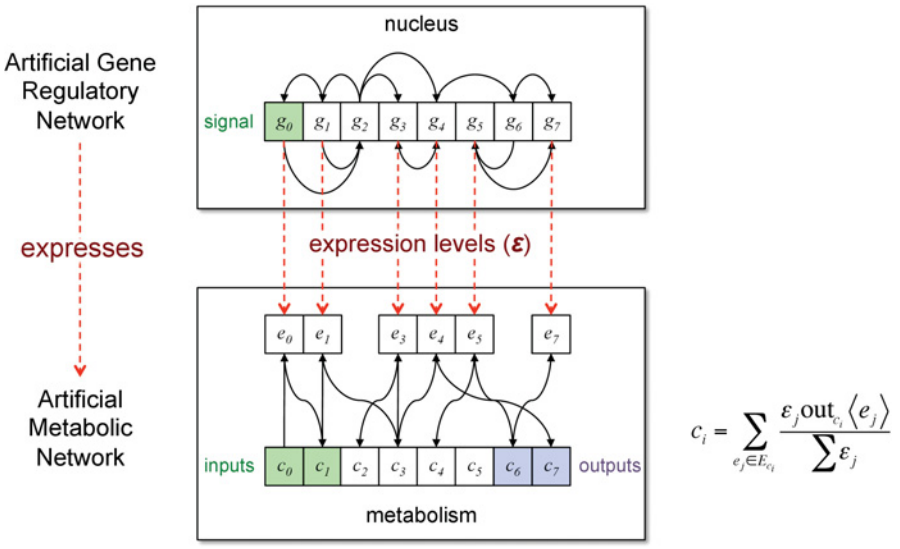
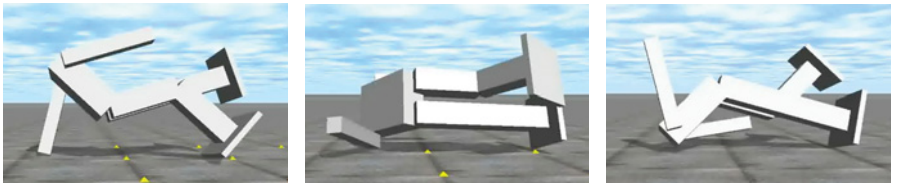


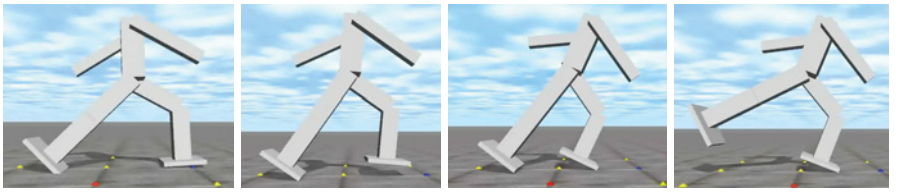
Figure 15.4 Example of an artificial biochemical network model in which an AGRN is coupled to an artificial metabolic network (AMN). The AMN models how a group of enzyme-mediated reactions modulate the concentrations of a group of chemical species. The AGRN, in turn, modulates the expression of the functional components of the AMN (i.e., its enzyme analogs), in effect switching different metabolic pathways on and off; see Ref. [53] for more details.

In Refs. [53, 56], we considered an ABN model in which an AGRN was used to express an artificial metabolic network (AMN, a simple artificial chemistry described in Ref. [52])—see Figure 15.4. This architecture was intended to capture the constructive relationship between genetic and metabolic networks, with the AGRN switching between different AMNs in response to an external signal, and the AMN then implementing the system's input–output mapping, that is, its current computational behavior. We applied this coupled ABN model to various control tasks, all of which required the ability to express multiple control behaviors and transition between these behaviors rapidly and in a context-sensitive manner. For example, in Ref. [53], we applied it to problems in chaos control and robotic locomotion, finding that coupled ABN-based controllers were generally easier to evolve and performed better than standalone AGRNs or AMNs. We also noted the value of using complex regulatory functions, in the form of discrete maps. In particular, coupled ABNs with discrete maps were observed to generate a diverse range of complex behaviors. To illustrate this, in Figures 15.5 and 15.6 we show examples of the kind of behaviors that can be generated by using coupled ABNs with discrete map regulatory functions.

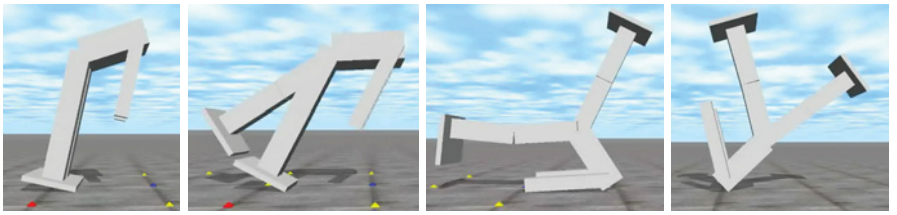
In biological systems, the cell's signaling network plays an important role in instigating transitions between GRN expression states. It achieves this via a complex spatio-temporal arrangement of signaling proteins that integrate and transduce the diverse signals received from the cell's environment, eventually relaying its response to the GRN through the manipulation of one or more TFs. It is not hard to imagine



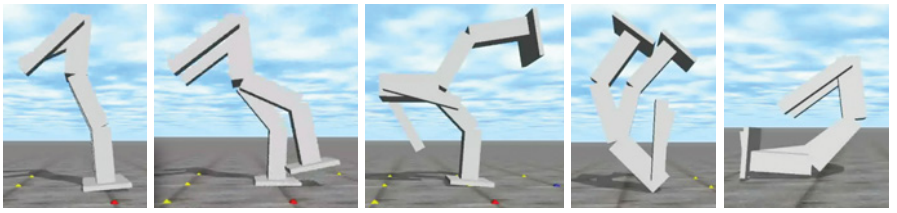
(a) Sideways roll. See http://youtu.be/hzMPB8EZGb8.



(b) Backward walk. See http://youtu.be/mT9qKZS7pds.



(c) Forward somersault. See http://youtu.be/urSNnVC1VqY.



(d) Backward somersault. See http://youtu.be/gSrQlsvbD98.

Figure 15.5 Diverse behaviors exhibited by coupled artificial biochemical networks evolved to control bipedal robots. All four controllers were found within the final population of a single multi-objective evolutionary algorithm run in which the objectives were to maximize distance covered and minimize energy used; see Refs. [53, 55] for more information about how ABNs have been used to control legged robots.

that this kind of flexible decision-making process could also be useful within real-world controllers that must respond to diverse incoming signals. Motivated by this, we considered how a particular facet of signaling network organization, crosstalk, could benefit AGRN models. Crosstalk has been identified as a means through which signaling networks achieve complex nonlinear responses. Our work with crosstalking ABNs suggests that it can carry out a similar role within a control systems context [26]. In Ref. [25], for example, we demonstrated how it can be used as the basis for

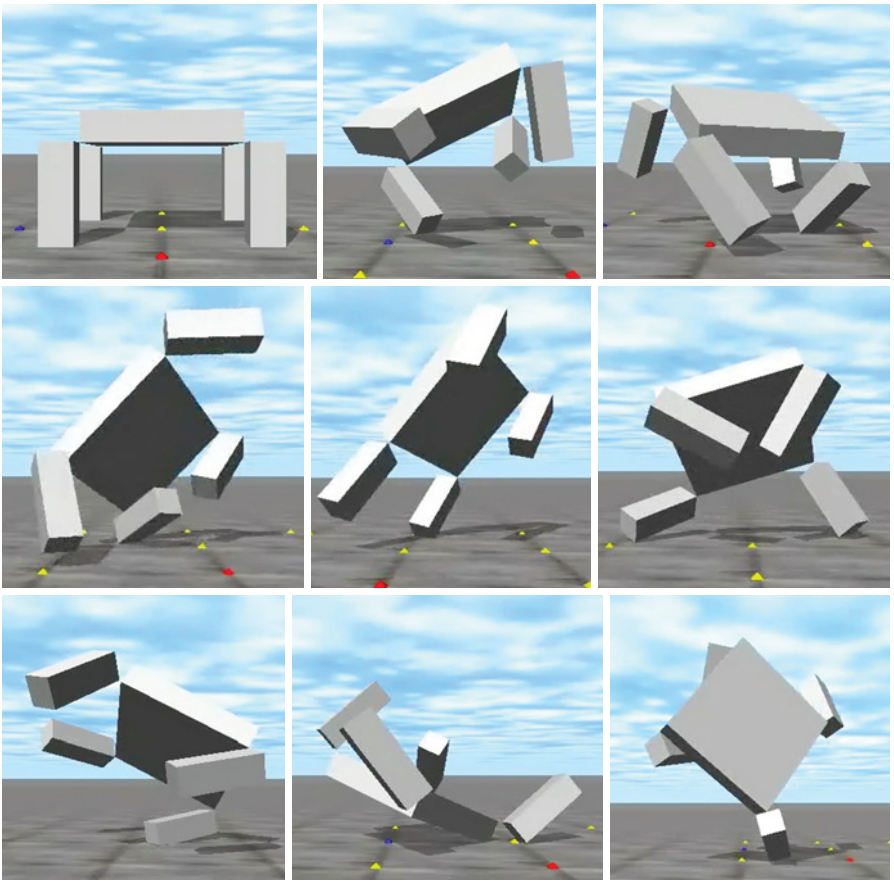


Figure 15.6 Expressive behaviors of a discrete map coupled artificial biochemical network controlling a four-legged robot with three degrees of freedom per leg. The objective was for the robot to move around as much as possible without moving away from its starting position, promoting movements that resemble dance; see: http://youtu.be/WzCmTUwtC3s.

a distributed adaptive locomotion controller for a legged robot, integrating signals from various internal and external sensors and generating a robust joint response.

Epigenetic processes also play a major role in cell fate determination, through their ability to guide the cell between different specialized states [49, 69]. Although these processes act through various biochemical mechanisms, for example, DNA methylation and histone modification, their effect is to change the accessibility of genes to the transcriptional machinery. This, in turn, causes modifications to the topology of the GRN, leading to changes in the expression state and hence the behavior of the cell. Since epigenetic modifications are the result of protein activity, this can be considered a form of self-modification, with the GRN being able to activate and deactivate different components of itself over the course of time. In Refs.[73, 74],

we investigated this idea within an AGRN context by introducing epigenetic switches to an AGRN model, each being able to sense the state of the network and turn other nodes on or off as a consequence. The resulting dynamical patterns of topological self-modification appeared to work well in control problems where there was a need to switch rapidly and robustly between different behaviors. For instance, in Ref. [73], we noted how the use of epigenetic switches allowed AGRN controllers to solve larger instances of the coupled inverted pendulums problem.

15.7 FUTURE RESEARCH DIRECTIONS

15.7.1 Wider Control Applications

Previous work has demonstrated that AGRNs can be used to solve a broad range of control problems. Hence, it makes sense to speculate about the wider potential for applications in this area. Certainly, one of the advantages of using EAs to optimize AGRNs is that we do not require a theoretical understanding of the system that we are attempting to control. This is in contrast to the majority of conventional control methods in common use, which generally require a good understanding of a system before they can be applied. In this sense, the range of potential applications is very broad, and there is scope for applying these methods in areas where the conventional control community has made few inroads.

One such area is the control of executable models. Executable modeling approaches, and especially agent-based modeling [62], have become popular in fields as diverse as economics, the social sciences, biology, and medicine. The approach involves constructing a dependable model of some aspect of a naturally occurring system, usually as a collaboration between computational modelers and domain specialists. The model can then be executed, and the resulting dynamics studied in order to infer understanding about the system being modeled. In addition to observing natural dynamics, for many of these systems there is an interest in how their state can be modified. For instance, for models of disease pathways, this interest relates to the problem of designing drug interventions, moving the system from a diseased to a disease-free state. For economic models, there is an interest in how we can transition away from bad economic states, such as recession. It is plausible that there are many control strategies that can achieve these outcomes, and the use of evolvable, expressive controllers in concert with executable models could be an effective way of exploring these. It is likely too that there are different trade-offs in the space of control strategies (for instance, ease of carrying out an intervention versus effectiveness of the intervention). In this situation, multiobjective EAs could provide a means of obtaining a more complete picture of how to control a system.

15.7.2 Better Computational Understanding

Previous work also suggests that AGRNs can be applied to broader range of applications. However, there has been little work in ascertaining what kind of applications

AGRNs are most suited for. By analogy with biological systems, we can expect them to be good at control, signal processing, and pattern generation, and existing computational applications in these areas reflect this. However, the nature of biological computation is often unclear, and it is limiting to assume that apparently computational mechanisms in biological systems cannot perform a wider range of behaviors simply because we have not observed them doing these. Studies of Boolean networks indicate that these models do have a range of interesting computational properties, including computational universality. However, focused studies are required in order to develop a better understanding of their practical usefulness for different classes of problem. We also need a better understanding of how modeling choices affect computability. In almost all cases, previous applied studies have focused on a single AGRN model out of all the many possible models. Consequently, it is unclear how these results generalize to AGRNs as a whole. Improved understanding of biological GRNs may one day help with this, but at present, it is difficult to say how the many different components of biological regulation contribute to the overall behavior of GRNs.

Another difficulty in studying the computational properties of AGRNs is the key role played by the optimization algorithm. It may well be possible to express any computational behavior with a particular AGRN model, but this is academic unless we can locate the AGRN that expresses the behavior we want expressed. More expressive models are often larger (i.e., have more parameters). This means that the optimization search space is also larger, which, in turn, may mean that it is harder to find a particular model instance. For EAs, evolvability must also be taken into account. If an AGRN model is not encoded in a way that is sufficiently evolvable, then it may be infeasible to find a particular instance regardless of the expressiveness of the model. Hence, future research in this area must also take into consideration the role of evolvability, and how to achieve it, if the computational potential of AGRNs is to be fulfilled.

15.7.3 Biological Computing

AGRNs also have considerable potential for understanding biological systems. As noted above, we do not fully understand how different regulatory components contribute to the overall behavior of a GRN. However, we can study how analogs of these regulatory components contribute to the overall behavior of an AGRN, and consider what this tells us about biological GRNs. If we argue that computational problems represent a relatively wide range of dynamical environments, we might even say that this tells us more about the general characteristics of GRNs than studying a particular biological problem. We might also consider this cybernetics-like approach to be a kind of generalized systems biology, telling us things about biological mechanisms that are not apparent from their direct biological setting. This argument is perhaps even more relevant to signaling networks, whose complex temporal–spatial organization makes them very difficult to study. For example, it is currently not possible to infer the components of signaling pathways by sampling the biomolecules present within a cell, due to the vast number of possible interactions between these biomolecules. One

way to guide this process would be to better understand the evolutionary pressures that constrain interactions between signaling molecules, and this could potentially be done by exploring the information processing capabilities of artificial signaling networks, identifying the topologies that are useful in a generalized, non-biological, context.

Another potentially strong application area for AGRNs is in synthetic biology [23]. Currently, this field is dominated by approaches that involve designing biological circuits based around computer engineering principles, for example, biochemical implementations of digital circuits [6]. Needless to say, biological systems are not based upon computer engineering principles. This suggests an alternative approach, which involves designing synthetic biology circuits based around the kind of computational models discussed in this chapter. There are, of course, challenges involved in this kind of work, but in many respects it would be more natural to use a controller derived from a GRN to control a biological process, rather than constructing a digital circuit to do this. It would also have the potential to be more expressive, opening up new application areas for synthetic biology, and leveraging the computational understanding of GRN models developed in AGRN studies.

15.8 CONCLUSIONS

Over the last decade, computer scientists and engineers have become increasingly interested in the role that GRNs play within the development and function of biological organisms. In particular, there has been a realization that computational models of GRNs can be used for computation, especially when the nature of this computation reflects the primary activities of biological GRNs. This has led to researchers exploring how these computational models, referred to here as AGRNs, can be used for applications that require intelligent control, pattern generation, and signal processing. The results of these investigations show considerable promise, demonstrating that AGRNs retain many of the desirable properties of their biological analogs, such as complex nonlinear behavior, robustness, compactness, and evolvability.

Nevertheless, the study of AGRNs is still in its early stages, and there is still much work to do in understanding their theoretical properties and computational potential. We have speculated about their future uses, but it is also important to gain a better understanding of what these computational models can do. By doing this, there is the possibility of informing understanding of biological systems, in addition to computer science and engineering. Biological science has made great progress in mapping and understanding GRNs, yet these systems remain only partially understood. AGRNs give us the opportunity to explore analogous behaviors within a more general setting, which, in turn, might lead to a better understanding of the general properties of GRNs. However, the reverse is also true, and it is likely that future progress in biology will continue to inform the design of AGRNs. It is also conceivable that one day the two areas of study might be unified within the context of synthetic biology.

REFERENCES

1. R. Albert. Boolean modeling of genetic regulatory networks. In: E. Ben-Naim, H. Frauenfelder, and Z. Toroczkai, editors, *Complex Networks*, volume 650 of *Lecture Notes in Physics*, pages 459–481. Springer, Berlin, 2004.
2. M. Aldana. Boolean dynamics of networks with scale-free topology. *Physica D: Nonlinear Phenomena*, 185(1):45–66, Oct. 2003.
3. J. Aracena, E. Goles, A. Moreira, and L. Salinas. On the robustness of update schedules in Boolean networks. *BioSystems*, 97(1):1–8, July 2009.
4. W. Banzhaf. Artificial regulatory networks and genetic programming. In: R. Riolo and B. Worzel, editors, *Genetic Programming Theory and Practice*, volume 6 of *Genetic Programming Series*, pages 43–61. Springer, New York, NY, 2003.
5. W. Banzhaf. On the dynamics of an artificial regulatory network. In: W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. Kim, editors, *Advances in Artificial Life*, volume 2801 of *Lecture Notes in Computer Science*, chapter 24, pages 217–227. Springer, Berlin, 2003.
6. Y. Benenson. Biomolecular computing systems: principles, progress and potential. *Nature Reviews Genetics*, 13(7):455–468, June 2012.
7. P. Bentley. Evolving fractal gene regulatory networks for graceful degradation of software. In: O. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, A. van Moorsel, and M. van Steen, editors, *Self-star Properties in Complex Information Systems*, volume 3460 of *Lecture Notes in Computer Science*, pages 21–35. Springer, Berlin, 2005.
8. P. J. Bentley. Controlling robots with fractal gene regulatory networks. In: L. Nunes de Castro and F. J. Von Zuben, editors, *Recent Developments in Biologically Inspired Computing*, pages 320–339. IGI Global, Hershey, PA, 2004.
9. C. P. Bowers. Formation of modules in a computational model of embryogeny. In: D. Corne, et al., editors, *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 537–542. IEEE Press, Piscataway, NJ, 2005.
10. L. Bull and R. Preen. On dynamical genetic programming: random Boolean networks in learning classifier systems. In: L. Vanneschi, S. Gustafson, A. Moraglio, I. De Falco, and M. Ebner, editors, *Genetic Programming*, volume 5481 of *Lecture Notes in Computer Science*, pages 37–48. Springer, Berlin, 2009.
11. S. E. Castel and R. A. Martienssen. RNA interference in the nucleus: roles for small RNAs in transcription, epigenetics and beyond. *Nature Reviews Genetics*, 14(2):100–112, Jan. 2013.
12. A. Chavoya, I. R. Andalon-Garcia, C. Lopez-Martin, and M. E. Meda-Campaña. Use of evolved artificial regulatory networks to simulate 3D cell differentiation. *BioSystems*, 102(1):41–48, Oct. 2010.
13. S. Cussat-Blanc and J. Pollack. Using pictures to visualize the complexity of gene regulatory networks. In: C. Adami, D. M. Bryson, C. Ofria, and R. T. Pennock, editors, *Artificial Life 13*, pages 491–498. MIT Press, Cambridge MA, 2012.
14. S. Cussat-Blanc, S. Sanchez, and Y. Duthen. Controlling cooperative and conflicting continuous actions with a gene regulatory network. In: A. J. Fernández Leiva, et al., editors, *Computational Intelligence and Games (CIG), The 2012 IEEE Conference on*, pages 187–194. IEEE Press, Piscataway, NJ, 2012.
15. F. Dellaert and R. D. Beer. A developmental model for the evolution of complete autonomous agents. In: P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson,

editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB 1996)*, pages 393–401, MIT Press, Cambridge, MA, 1996.

16. K. A. Dittmar, J. M. Goodenbour, and T. Pan. Tissue-specific differences in human transfer RNA expression. *PLoS Genetics*, 2(12):e221, Dec. 2006.
17. P. Dittrich, J. Ziegler, and W. Banzhaf. Artificial chemistries—a review. *Artificial Life*, 7(3):225–275, July 2001.
18. B. Drossel. Number of attractors in random Boolean networks. *Physical Review E*, 72(1):016110, July 2005.
19. E. Dubrova, M. Teslenko, and H. Tenhunen. A computational scheme based on random Boolean networks. In: C. Priami, F. Dressler, O. Akan, and A. Ngom, editors, *Transactions on Computational Systems Biology X*, volume 5410 of *Lecture Notes in Computer Science*, pages 41–58. Springer, Berlin, 2008.
20. 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*, 85(3):177–200, Sept. 2006.
21. P. Eggenberger Hotz. Evolving morphologies of simulated 3D organisms based on differential gene expression. In: P. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life (ECAL 1997)*, pages 205–213. MIT Press, Cambridge, MA, 1997.
22. R. Franke, F. J. Theis, and S. Klamt. From binary to multivalued to continuous models: the lac operon as a case study. *Journal of Integrative Bioinformatics*, 7(1), doi: 10.2390/biecoll-jib-2010-151, 2010.
23. P. S. Freemont and R. I. Kitney. *Synthetic Biology—A Primer*. Imperial College Press, London, UK, 2012.
24. N. J. Fuda, M. B. Ardehali, and J. T. Lis. Defining mechanisms that regulate RNA polymerase II transcription *in vivo*. *Nature*, 461(7261):186–192, Sept. 2009.
25. L. A. Fuente, M. A. Lones, A. P. Turner, L. S. Caves, S. Stepney, and A. M. Tyrrell. Adaptive robotic gait control using coupled artificial signalling networks, Hopf oscillators and inverse kinematics. In: C. A. Coello Coello, et al., editors, *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1435–1442. IEEE Press, Piscataway, NJ, 2013.
26. L. A. Fuente, M. A. Lones, A. P. Turner, S. Stepney, L. S. Caves, and A. M. Tyrrell. Computational models of signalling networks for non-linear control. *BioSystems*, 112(2):122–130, May 2013.
27. C. Gershenson. Introduction to random Boolean networks. In: M. Bedau, P. Husbands, T. Hutton, S. Kumar, and H. Suzuki, editors, *Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, pages 160–173. MIT Press, Cambridge, MA, 2004.
28. T. Glisovic, J. L. Bachorik, J. Yong, and G. Dreyfuss. RNA-binding proteins and post-transcriptional gene regulation. *FEBS Letters*, 582(14):1977–1986, June 2008.
29. T. G. W. Gordon. Exploring models of development for evolutionary circuit design. In: *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 2050–2057. IEEE Press, Piscataway, NJ, 2003.
30. T. G. W. Gordon and P. J. Bentley. Development brings scalability to hardware evolution. In: J. Lohn, et al., *Evolvable Hardware, 2005. Proceedings. 2005 NASA/DoD Conference on*, pages 272–279. IEEE Press, Piscataway, NJ, 2005.

31. A. Goudarzi, C. Teuscher, N. Gulbahce, and T. Rohlf. Emergent criticality through adaptive information processing in Boolean networks. *Physical Review Letters*, 108(12):128702, Mar. 2012.
32. F. Greil. Dynamics of Boolean networks. PhD thesis, Technical University of Darmstadt, 2009.
33. F. U. Hartl, A. Bracher, and M. Hayer-Hartl. Molecular chaperones in protein folding and proteostasis. *Nature*, 475(7356):324–332, July 2011.
34. S. Huang, I. Ernberg, and S. Kauffman. Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. *Seminars in Cell and Developmental Biology*, 20(7):869–876, Sept. 2009.
35. R. Jaenisch and A. Bird. Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals. *Nature Genetics*, 33:245–254, Mar. 2003.
36. Y. Jin, H. Guo, and Y. Meng. A hierarchical gene regulatory network for adaptive multirobot pattern formation. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 42(3):805–816, June 2012.
37. M. Joachimczak, T. Kowaliw, R. Doursat, and B. Wróbel. Evolving morphologies and controllers for soft-bodied multicellular animats using gene regulatory networks and artificial embryogenesis. In: S. Doncieux, Y. Jin, and J.-B. Mouret, editors, *Evo-Devo-Robo: Evolutionary Robotics and Developmental Robotics at the Fourteenth International Conference on Genetic and Evolutionary Computation*, GECCO Companion'12, pages 357–360. ACM, New York, NY, 2012.
38. M. Joachimczak and B. Wróbel. Evolving gene regulatory networks for real time control of foraging behaviours. In: H. Fellermann, M. Dörr, M. M. Hanczyc, L. L. Laursen, S. Maurer, D. Merkle, P.-A. Monnard, K. Stoy, and S. Rasmussen, editors, *Artificial Life XII: Proceedings of the 12th International Conference on the Simulation and Synthesis of Living Systems*, pages 348–355. MIT Press, Cambridge, MA, 2010.
39. S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, Mar. 1969.
40. S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, Oxford, 1st ed, 1993.
41. G. A. Khoury, R. C. Baliban, and C. A. Floudas. Proteome-wide post-translational modification statistics: frequency analysis and curation of the Swiss-Prot database. *Scientific Reports*, 1:90, doi: 10.1038/srep00090, Sept. 2011.
42. M. Kirschner and J. Gerhart. Evolvability. *Proceedings of the National Academy of Sciences of the United States of America*, 95(15):8420–8427, July 1998.
43. A. Koopman and D. Roggen. Evolving genetic regulatory networks for hardware fault tolerance. In: X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervós, J. Bullinaria, J. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature—PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 561–570. Springer, Berlin, 2004.
44. S. Koos, J. B. Mouret, and S. Doncieux. The transferability approach: crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145, Feb. 2013.
45. J. Krohn and D. Gorse. Fractal gene regulatory networks for control of nonlinear systems. In: R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 209–218. Springer, Berlin, 2010.

46. S. Kumar. A developmental biology inspired approach to robot control. In: J. Pollack, M. A. Bedau, P. Husbands, R. A. Watson, and T. Ikegami, editors, *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*. MIT Press, Cambridge, MA, 2004.

47. H. Kwak and J. T. Lis. Control of transcriptional elongation. *Annual Review of Genetics*, 47(1):483–508, Nov. 2013.

48. W.-P. Lee and T.-H. Yang. Combining GRN modeling and demonstration-based programming for robot control. In: D. Palmer-Brown, C. Draganova, E. Pimenidis, and H. Mouratidis, editors, *Engineering Applications of Neural Networks*, volume 43 of *Communications in Computer and Information Science*, pages 190–199. Springer, Berlin, 2009.

49. R. Lister, M. Pelizzola, Y. S. Kida, R. D. Hawkins, J. R. Nery, G. Hon, J. Antosiewicz-Bourget, R. O'Malley, R. Castanon, S. Klugman, M. Downes, R. Yu, R. Stewart, B. Ren, J. A. Thomson, R. M. Evans, and J. R. Ecker. Hotspots of aberrant epigenomic reprogramming in human induced pluripotent stem cells. *Nature*, 471(7336):68–73, Mar. 2011.

50. Y. Liu, J. Timmis, O. Qadir, G. Tempesti, and A. Tyrrell. A developmental and immune-inspired dynamic task allocation algorithm for microprocessor array systems. In: E. Hart, C. McEwan, J. Timmis, and A. Hone, editors, *Artificial Immune Systems*, volume 6209 of *Lecture Notes in Computer Science*, pages 199–212. Springer, Berlin, 2010.

51. J. T. Lizier, S. Pritam, and M. Prokopenko. Information dynamics in small-world Boolean networks. *Artificial Life*, 17(4):293–314, July 2011.

52. M. Lones, A. Tyrrell, S. Stepney, and L. Caves. Controlling complex dynamics with artificial biochemical networks. In: A. Esparcia-Alcázar, A. Ekárt, S. Silva, S. Dignum, and A. Uyar, editors, *Genetic Programming*, volume 6021 of *Lecture Notes in Computer Science*, pages 159–170. Springer, Berlin, 2010.

53. M. A. Lones, L. A. Fuente, A. P. Turner, L. S. D. Caves, S. Stepney, S. L. Smith, and A. M. Tyrrell. Artificial biochemical networks: evolving dynamical systems to control dynamical systems. *IEEE Transactions on Evolutionary Computation*, 18(2):145–166, Apr. 2014.

54. M. A. Lones, S. L. Smith, A. M. Tyrrell, J. E. Alty, and D. R. S. Jamieson. Characterising neurological time series data using biologically motivated networks of coupled discrete maps. *BioSystems*, 112(2):94–101, May 2013.

55. M. A. Lones, A. P. Turner, L. A. Fuente, S. Stepney, L. S. D. Caves, and A. M. Tyrrell. Biochemical connectionism. *Natural Computing*, 12(4):453–472, Dec. 2013.

56. M. A. Lones, A. M. Tyrrell, S. Stepney, and L. S. D. Caves. Controlling legged robots with coupled artificial biochemical networks. In: T. Lenaerts, M. Giacobini, H. Bersini, P. Bourgine, M. Dorigo, and R. Doursat, editors, *Advances in Artificial Life, European Conference on Artificial Life (ECAL) 2011*. MIT Press, Cambridge, MA, 2011.

57. C. Mattiussi and D. Floreano. Analog genetic encoding for the evolution of circuits and networks. *IEEE Transactions on Evolutionary Computation*, 11(5):596–607, Oct. 2007.

58. B. Mesot and C. Teuscher. Deducing local rules for solving global tasks with random Boolean networks. *Physica D: Nonlinear Phenomena*, 211(1–2):88–106, Nov. 2005.

59. M. Nicolau, M. Schoenauer, and W. Banzhaf. Evolving genes to balance a pole. In: A. Esparcia-Alcázar, A. Ekárt, S. Silva, S. Dignum, and Ş. Uyar, editors, *Genetic Programming*, volume 6021 of *Lecture Notes in Computer Science*, Chapter 17, pages 196–207. Springer, Berlin, 2010.

60. S. Patarnello and P. Carnevali. Learning capabilities of Boolean networks. In: I. Aleksander, editor, *Neural Computing Architectures: the Design of Brain-like Machines*, pages 117–129. MIT Press, Cambridge, MA, 1989.

61. T. Quick, C. Nehaniv, K. Dautenhahn, and G. Roberts. Evolving embodied genetic regulatory network-driven control systems. In: W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. Kim, editors, *Advances in Artificial Life*, volume 2801 of *Lecture Notes in Computer Science*, pages 266–277. Springer, Berlin, 2003.

62. S. F. Railsback and V. Grimm. *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton University Press, Princeton, NJ, 2011.

63. T. Reil. Dynamics of gene expression in an artificial genome—implications for biological and artificial ontogeny. In: D. Floreano, D. Nicoud, and F. Mondada, editors, *Advances in Artificial Life, 5th European Conference, ECAL'99*, volume 1674 of *Lecture Notes in Artificial Intelligence*, pages 457–466. Springer, Berlin, 1999.

64. A. Roli, M. Manfroni, C. Pinciroli, and M. Birattari. On the design of Boolean network robots. In: C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. Esparcia-Alcázar, J. Merelo, F. Neri, M. Preuss, H. Richter, J. Togelius, and G. Yannakakis, editors, *Applications of Evolutionary Computation*, volume 6624 of *Lecture Notes in Computer Science*, pages 43–52. Springer, Berlin, 2011.

65. S. Sanchez and S. Cussat-Blanc. Gene regulated car driving: using a gene regulatory network to drive a virtual car. *Genetic Programming and Evolvable Machines*, 15(4):477–511, Dec. 2014.

66. M. A. Savageau. Introduction to S-systems and the underlying power-law formalism. *Mathematical and Computer Modelling*, 11:546–551, Jan. 1988.

67. D. J. Taatjes. The human mediator complex: a versatile, genome-wide regulator of transcription. *Trends in Biochemical Sciences*, 35(6):315–322, June 2010.

68. T. Taylor. A genetic regulatory network-inspired real-time controller for a group of underwater robots. In: F. Groen, N. Amato, A. Bonarini, E. Yoshida, and B. Kröse, editors, *Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS-8)*, pages 403–412. IOS Press, Amsterdam, 2004.

69. V. B. Teif, Y. Vainshtein, M. Caudron-Herger, J.-P. Mallm, C. Marth, T. Höfer, and K. Rippe. Genome-wide nucleosome positioning during embryonic stem cell development. *Nature Structural and Molecular Biology*, 19(11):1185–1192, Nov. 2012.

70. C. Torres-Sosa, S. Huang, and M. Aldana. Criticality is an emergent property of genetic networks that exhibit evolvability. *PLoS Computational Biology*, 8(9):e1002669, Sept. 2012.

71. M. Trefzer, T. Kuyucu, J. Miller, and A. Tyrrell. Evolution and analysis of a robot controller based on a gene regulatory network. In: G. Tempesti, A. Tyrrell, and J. Miller, editors, *Evolvable Systems: From Biology to Hardware*, volume 6274 of *Lecture Notes in Computer Science*, pages 61–72. Springer, Berlin, 2010.

72. M. A. Trefzer, T. Kuyucu, J. F. Miller, and A. M. Tyrrell. Image compression of natural images using artificial gene regulatory networks. In: M. Pelikan and J. Branke, editors, *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO'10, pages 595–602. ACM, New York, NY, 2010.

73. A. P. Turner, M. A. Lones, L. A. Fuente, S. Stepney, L. S. D. Caves, and A. Tyrrell. The artificial epigenetic network. In: A. M. Tyrrell, et al., editors, *Evolvable Systems (ICES), 2013 IEEE International Conference on*, pages 66–72. IEEE Press, Piscataway, NJ, 2013.

74. A. P. Turner, M. A. Lones, L. A. Fuente, S. Stepney, L. S. D. Caves, and A. M. Tyrrell. The incorporation of epigenetics in artificial gene regulatory networks. *BioSystems*, 112(2):56–62, May 2013.

75. J. M. Vaquerizas, S. K. Kummerfeld, S. A. Teichmann, and N. M. Luscombe. A census of human transcription factors: function, expression and evolution. *Nature Reviews Genetics*, 10(4):252–263, Apr. 2009.

76. S. Wolfram. *A New Kind of Science*. Wolfram Media Inc., Champaign, IL, 2002.

77. Y. Yao, K. Marchal, and Y. Van de Peer. Improving the adaptability of simulated evolutionary swarm robots in dynamically changing environments. *PLoS One*, 9(3):e90695, Mar. 2014.

78. P. Zahadat, D. Christensen, U. Schultz, S. Katebi, and K. Stoy. Fractal gene regulatory networks for robust locomotion control of modular robots. In: S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.-A. Meyer, and J.-B. Mouret, editors, *From Animals to Animats 11*, volume 6226 of *Lecture Notes in Computer Science*, pages 544–554. Springer, Berlin, 2010.

79. M. Zanin and A. N. Pisarchik. Boolean networks for cryptography and secure communication. *Nonlinear Science Letters B*, 1(1):25–32, 2011.

80. J. Zañudo, M. Aldana, and G. Martínez-Mekler. Boolean threshold networks: virtues and limitations for biological modeling. In: S. Niiranen and A. Ribeiro, editors, *Information Processing and Biological Systems*, volume 11 of *Intelligent Systems Reference Library*, pages 113–151. Springer, Berlin, 2011.

81. S. Zhan, J. F. Miller, and A. M. Tyrrell. An evolutionary system using development and artificial genetic regulatory networks for electronic circuit design. *BioSystems*, 98(3):176–192, Dec. 2009.

82. N. M. Zivanov, A. Bresticker, D. Krishnaswamy, S. Venkatakrishnan, D. Marculescu, and J. R. Faeder. Emulation of biological networks in reconfigurable hardware. In: R. Grossman, A. Rzhetsky, S. Kim, and W. Wang, editors, *Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB'11, pages 536–540. ACM, New York, NY, 2011.