Modeling transcriptional regulatory networks

Hamid Bolouri[1,2]* and Eric H. Davidson[2]

Summary

Developmental processes in complex animals are directed by a hardwired genomic regulatory code, the ultimate function of which is to set up a progression of transcriptional regulatory states in space and time. The code specifies the gene regulatory networks (GRNs) that underlie all major developmental events. Models of GRNs are required for analysis, for experimental manipulation and, most fundamentally, for comprehension of how GRNs work. To model GRNs requires knowledge of both their overall structure, which depends upon linkage amongst regulatory genes, and the modular building blocks of which GRNs are heirarchically constructed. The building blocks consist of basic transcriptional control processes executed by one or a few functionally linked genes. We show how the functions of several such building blocks can be considered in mathematical terms, and discuss resolution of GRNs by both ''top down'' and ''bottom up'' approaches. *BioEssays* 24:1118–1129, 2002.

Introduction

This article reviews the role of modeling in understanding animal genetic regulatory networks (GRNs). For reviews of modeling prokaryotic GRNs, see Refs. 1,2. Why focus on animal GRNs? Because ultimately, all species-specific characteristics must be explicable at the level of genetically inherited information. Given the remarkable commonality of protein families in the animal kingdom, we must conclude that the morphological differences between animal species arise primarily through differential regulation of genes and their products, and that the information for this differential regulation must be encoded in the inherited DNA (see Davidson, Ref. 3, for in-depth discussion). Here we consider the use of computer modeling as an aid to unraveling and quantifying this process.

[1]Institute for Systems Biology, USA and Science & Technology Research Centre, University of Hertfordshire, UK.
[2]Division of Biology 156-29, California Institute of Technology, USA.
Funding agency: NIH grant (GM-61005).
*Correspondence to: Hamid Bolouri, Institute for Systems Biology, 1441 North 34th Street, Seattle, WA 98103-8904.
E-mail: hbolouri@systemsbiology.org
DOI 10.1002/bies.10189
Published online in Wiley InterScience (www.interscience.wiley.com).

What follows will be primarily focused on transcriptional regulatory networks.

Unraveling cellular processes is usually complicated by the experimental difficulty of identifying all the interactions that take place in vivo and, equally difficult, measuring the kinetic parameters associated with in vivo biochemical, biomechanical, or electrophysiological interactions. GRNs offer significant advantages in this respect, as follows.

- Whole genome sequencing can, in principle (though not yet in practice), identify all potential macromolecular players (since they must ultimately be encoded in DNA).
- Large-scale technologies for gene discovery (arrayed mRNA expression assays, parallel quantitative PCR measurement of the effects of perturbations, DNA-sequence searching algorithms) are currently better developed for general use than their proteomic counterparts.
- GRN models ultimately lead to DNA-specific predictions such as the existence of putative binding sites for transcription factors hypothesized to regulate a downstream gene. Such predictions can be experimentally validated or falsified in straightforward ways.
- Transcription and translation are usually much slower than many protein–protein and enzymatic reactions. Thus, in GRN-based models, it is frequently possible to model these faster reactions as instantaneous events and the interactions that govern their activity as switches (of the type ''if <condition>then<outcome>''). This abstraction often permits large-scale GRN models of cellular processes.

Developmental GRNs, as they are encoded in animal genomes, are of course a product of evolution. They are not organized according to laws of parsimony, and they are not particularly streamlined. They have been assembled during evolution by addition of novel regulatory linkages to preexisting regulatory linkages; they are a mosaic of old and new features.[4] Robustness and fail-safe mechanisms that ensure stability of given states of expression are prominent features of the GRN structures that evolution has produced, as shown for example in the sea urchin endomesodermal GRN that we and our colleagues recently reported.[5] All this means is that the structure of a developmental GRN is likely to

be very different indeed from that of an easily intuited enzymatic pathway, or a neatly designed, maximally efficient logic circuit.

In the following, we review some mathematical descriptions of basic transcriptional regulatory processes that are useful as computational building blocks for modeling GRNs. The examples are focused on single genes and small modular network subelements consisting of several genes linked in simple regulatory circuits. But real-life GRNs for developmental processes are large and intricate, relative to these network building blocks. Adequate models for developmental GRNs must be anchored in sufficient biological information so that the overall structure of the GRN can be derived. Here we (1) provide a brief overview of the experimental technologies, and data-processing techniques commonly used for unraveling GRNs, (2) review the theoretical framework for GRN modeling, (3) present some putative GRN building blocks, and (4) outline arguments in favor of a two-pronged approach to reverse engineering GRNs (starting with a coarse-grained picture of a system and iteratively increasing the resolution, and building the network from the bottom up by first searching for elementary building blocks, then combinations of these, and so on).

Reverse engineering GRNs

In contrast to forward engineering, i.e., building new systems, "reverse engineering" is concerned with unraveling the operational principles underlying existing systems. It is used widely in engineering to understand competitors' products (see for example http://www.chipworks.com/FAQ.htm). Necessary components of such understanding are:

- A parts list (for instance, as revealed by large-scale gene expression assays).
- An understanding of the characteristics of the parts (as described in protein and gene databases, for example http://www.brenda.uni-koeln.de/).
- A map of how the parts fit together (as in "pathway" databases, see for example http://us.expasy.org/cgi-bin/search-biochem-index).
- A description of the outcome of the interactions among the parts. For a static structure, such as a house, this means understanding the utility of walls, rooms, stairs, doors, windows, etc. Most systems of interest, however, are dynamic; that is, they dissipate energy and carry out work. In such systems, in addition to the static structure, one needs to understand the dynamic behavior of the system as a function of the interactions of its component parts. Thus, to reverse engineer a car, we would need to understand not only the significance of the chassis and wheels, but also the dynamic behavior of the car as a function of fuel combustion, torque transmission, steering, etc. As hinted at in this example, a key aspect to understanding the dynamics of large-scale systems is the hierarchical division of a system into smaller modules with distinct functionality. *Engineered* systems are invariably built from hierarchical groupings of such modules and we expect the same will be true for GRNs. We will return to this issue at the end of this paper.

Methodologies and tools for unraveling GRNs

Methodologies, experimental technologies and data processing techniques for unraveling GRNs have been widely discussed.[1,6–10] Here, we present a brief overview merely to provide a context for the modeling issues that form the focus of the rest of this paper. Typically, the reverse engineering of a GRN will involve the following.

- Gene discovery through arrayed gene expression assays. The aim here is to discover the members of a network and to group these together in crudely defined categories. Typically, time-course-of-expression profiles (in both wild-type and disturbed cells), the functional nature of the protein products, and the spatial domains of expression of genes are used to form clusters of putative linkage groups.
- Regulatory linkage analysis. Typically, the activity of an upstream gene is disrupted (e.g., using RNAi, dominant negative, morpholino, or engrailed technologies), and/or ectopic expression is induced, in order to identify downstream regulated genes. In addition to identifying transcription factor target genes, regulatory linkage analysis can reveal whether regulatory inputs incident on a gene are required in combination (logical and) or individually (logical or), and whether they activate (logical imply) or repress (logical inversion, or not) the expression of the regulated gene (regulatory logic is discussed in Figs. 1–3). Thus, linkage analysis can provide both the connectivity structure of a GRN and a logical description of the interactions between genes.
- Prediction and experimental verification of transcription factor binding sites on *cis*-regulatory DNA. Assuming transcriptional regulation, the linkage map established in the preceding step indicates what binding sites may be expected to exist. Here, the aim is to verify the existence of specific stretches of regulatory DNA sequence as the binding site for each incident transcription factor. This is typically performed by experimental *cis*-regulatory analysis, increasingly with the aid of statistical sequence searching such as comparison of putative regulatory sequences of evolutionarily close species. An additional outcome of such analysis may be the discovery of multiple binding sites for some transcription factors. This has implications about the dynamics of the interaction

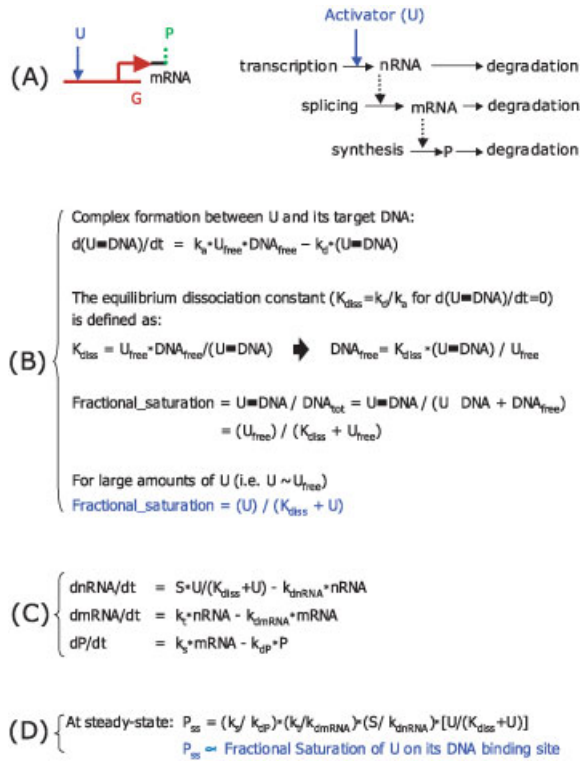

Figure 1. A three-step model of gene regulation. For simplicity, a gene (*G*) with a single ubiquitous transcriptional activator (U) is presented. **A:** Diagrammatic representation of the circuit in terms of the constituent components (left), and the processes involved (right). In this simple model, all the many processes and molecules involved in gene expression are reduced to just three steps. The top process stands for all the complex events that control the transcription and loss of nuclear RNA. Similarly, the multi-step processes that generate and degrade mRNA and the protein product of the gene (P) are summarized as single steps. **B:** Derivation of the relationship between transcription factor concentration and its occupancy (fractional saturation) on its target DNA binding site. k_a, k_d association and dissociation rates of U and its DNA-binding site. K_{diss} is the corresponding equilibrium dissociation constant. U ■ DNA represents the bound U–DNA complex. Note that this commonly used formulation of occupancy includes the assumption that the concentration of U is high enough that the amount of unbound U is approximately equal to total U. **C:** The three equations of the model. The latter two steps are presented as simple linear processes. The first process (nuclear RNA production) is intrinsically a non-linear function of the concentration of U. k_t and k_s represent the rates of mRNA and protein production respectively. k_d's represent degradation rates. "S" stands for activator strength, a measure of the efficiency with which a given complex activates transcription. **D:** At steady state, P is proportional to the fractional saturation of U, and a non-linear function of U concentration. The implications of this are discussed in Figs. 2 and 3.

of that factor with its target gene(s), as discussed in Figs 2 and 3.

- Measurement of kinetic data. A GRN with a given connectivity and set of logical interactions can exhibit multiple behaviors depending on the kinetic parameters (see examples in Fig. 4). Frequently, which behavior is most likely can be guessed by the context of the GRN and verified/falsified experimentally. Otherwise, it is necessary to measure kinetic data such as transcription factor association/dissociation rates, the rates of transcription and translation, and mRNA and protein degradation rates. In addition, it may be pertinent to characterize upstream cellular processes such as diffusion and transport, or electrical and mechanical interactions in order to explain the overall behavior of interest.
- Network modeling and simulation. At each of the above steps, computer modeling and simulation can be used to explore how well the relationships discovered explain features of the cellular process of interest. Models can reveal contradictions in our understanding, and can aid in posing experimentally falsifiable questions (hypothesis formulation). For example, we may ask whether the discovered families of genes are sufficient to explain a behavior of interest; alternatively, we may ask whether the structure of the network constrains its behavior to certain classes, or we may build a model of the revealed gene interactions and ask whether the resulting behavior faithfully reflects all experimental observations.

The choice of modeling formalism

All modeling is an abstraction of reality. The only exact model of any system is the system itself. So, when we set out to build a model of a system (here a GRN), we must make a choice about the level of detail and type of features that the model should represent. To a large extent, this is dictated by the characteristics of the system being studied, the type of experimental data available, and the type of questions that we wish to address through modeling. For example, phenomenological electrophysiological models are widely used to study the electrical activity of neuronal and cardiac cells without any explicit modeling of the underlying molecular mechanisms. At the other extreme, studies of systems of small numbers of molecules usually require stochastic models where the probability of each molecular interaction is computed from Gibbs Free Energy considerations (for examples of stochastic models and related modeling theory, see Refs. 11,12. For an early example of the application of phenomenological modeling to genetic networks, see Ref. 13). In between these extremes, lies a plethora of modeling formalisms.

Eukaryotic gene regulation is a complex process involving a very large number of physical, mechanical, and

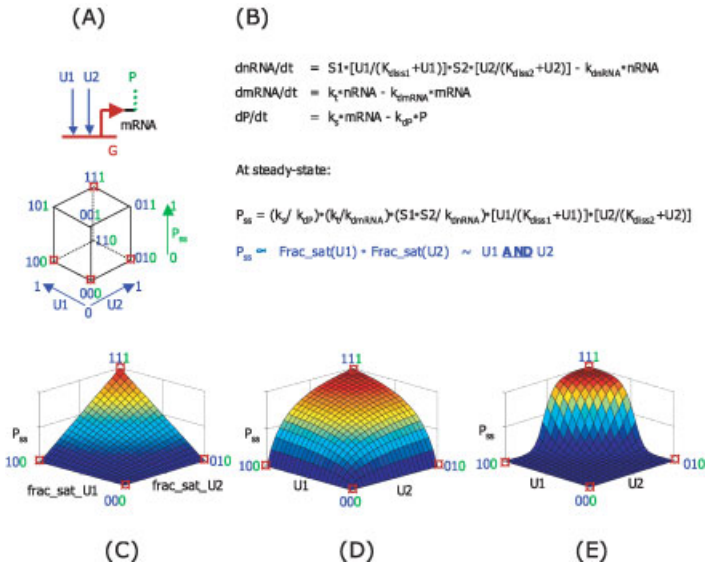

Figure 2. A simple model of a hypothetical gene with just two activators. In this example, both activators are necessary for transcription initiation. **A:** Top, schematic diagram; bottom, the effect of the two activators on steady-state protein expression level, shown on the corners of a unit (Boolean) cube; discussed below in conjunction with (C–E). **B:** Three-stage model of the gene (cf. Fig. 1). S1 and S2 are the activation strengths of the corresponding transcription factors. They could be combined into a single value for the overall strength of the U1-U2 complex. For given levels of U1 and U2 (i.e., at steady state), the concentration of the protein (P) is proportional to the product of the two fractional saturations and may be approximated by a Boolean logic and function. Such a function is visualized in the bottom drawing in (A). Consider a unit cube whose vertex coordinates in the (x,y) direction represent the four Boolean (all or none) combinations of the concentrations of U1 and U2. Let the z-axis represent the Boolean values of the protein product P (P = 0 in the lower cube face, and P = 1 in the upper cube face). We can then use a three-digit index to mark each vertex (here, blue digits for (U1,U2) values and green for P). The corners marked represent Boolean and functionality (contrast with Fig. 3A which presents the corresponding case for a Boolean or function). The same four points (representing Boolean and) are marked on the plots in (C–E). Note how they coincide with the extrema of each plotted surface. **C:** The steady-state value of P as a function of U1 and U2 fractional saturations. **D:** The steady-state value of P as a function of the concentrations of U1 and U2. Note the non-linear characteristic of the graph. If the transcription factors have multiple binding sites, multimerize, and/or interact with each other cooperatively, the steady state level of P behaves like a threshold function of the concentrations of U1 and U2, as illustrated in **E**. Note how, for most values of U1 and U2, P_{ss} has a value close to zero or one (i.e., P is approximately Boolean).

chemical interactions. In Fig. 1, we present a highly abstract, minimally simple, ODE-based model of transcriptional gene regulation for a ''cartoon'' gene with just one transcriptional regulator (symbolically illustrated in Fig. 1A). As summarized in Fig. 1B, the entire process is abstracted into three steps: (1) regulation of transcriptional activation by one or more transcription factors, (2) mRNA production/decay, and (3) protein synthesis/decay. For simplicity, the latter two steps are modeled as linear processes. However, the first process is intrinsically a nonlinear function of the concentration of the activating transcription factor (see Fig. 1B and its caption for an explanation). At steady state, concentration of the protein product of the modeled gene is proportional to the fractional saturation (occupancy) of the activating transcription factor on its DNA-binding site (see Fig. 1D for an explanation).

The relationship between models based on Ordinary Differential Equations (ODEs), continuous algebraic equations, and Boolean logic is explored and presented in more detail in the ''cartoon'' or ''toy'' examples shown in Figs 2 and 3. The cartoon gene of Fig. 2 has just two activating transcription factors, both of which are necessary for initiation of transcription. By contrast, the gene modeled in Fig. 3 has two inputs each of which is sufficient for transcription. See figure captions for details. The figures illustrate the simple manner in which the steady-state response of a gene to its regulatory factors can be modeled as an algebraic function of the occupancies of the regulatory factors. Boolean models may be viewed as discrete versions of these algebraic representations.

For our studies of GRNs underlying sea urchin embryonic development,[5,14–17] we have found a mixture of Boolean

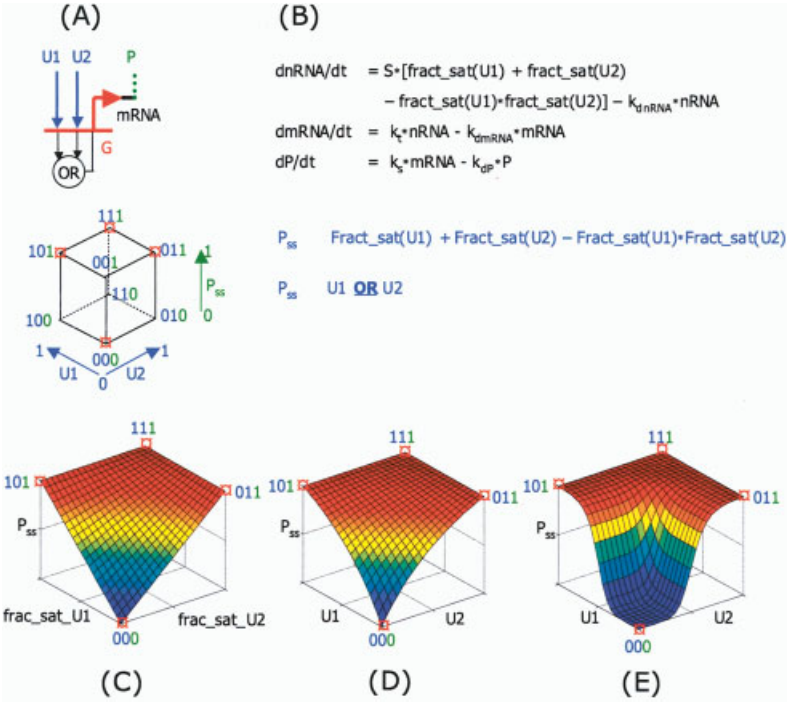

Figure 3. Model of a hypothetical gene with just two activators. In this example, each activator can independently cause transcription initiation, and for simplicity we assume that each transcription factor is capable of driving the gene at the maximal rate. **A:** Schematic diagrams using the representation styles introduced in Fig. 2. Top: To avoid confusion, we usually indicate the logical interactions between bound transcription factors, just below the line representing regulatory DNA (for further examples, see Refs. 15–17). Bottom: Boolean cube representation, as in Fig. 2A. **B:** Three-stage model and steady-state equation for the protein product. In this case, the steady-state level of the protein product is proportional to union of the fractional saturations of the two regulators. The union represents the three cases where each regulator is active alone, and when both are active together. Since the algebraic sum of two variables covers their overlap (U1 and U2 both active) twice, the product is subtracted once from the sum in the equations. The relationship is illustrated graphically in (C–E). **C:** Illustration of the steady-state value of P as a function of U1 and U2 fractional saturations. **D:** The relationship between steady-state level of P and U1, U2 concentrations. **E:** If we assume cooperative scenarios such as those discussed in Fig. 2E, then P behaves like a threshold

(i.e., discrete) and algebraic (i.e., continuous) logic most useful because changes in protein concentrations occur on a much faster timescale than successive developmental states. As discussed later in this paper (see Fig. 4), the use of ODEs necessitates the introduction of a large number of parameters for which experimental values are often not available. Thus, following Ockham's Razor (law of parsimony), we use the representation with the fewest number of free parameters. As discussed in Ref. 17, this formalism results in the same type of grammar tree as that used by structured programming languages and so can represent any and all regulatory interactions (so long as the relevant experimental data is available!).

The choice of modeling formalism and its implications are explored further in Figs. 4 and 5. Consider a hypothetical single-gene negative feedback circuit. Such a circuit can exhibit two classes of behavior: constant steady-state, or oscillation. Suppose experimental data indicates that the gene oscillates. Figure 4 shows some possible logical and ODE-based models of the circuit. A major issue in using ODE-based models is the form of equations used. Note that all four ODE models in Fig. 4 are highly abstract. Each equation summarizes the phenomenological behavior of a large number of biochemical/mechanical/physical steps. None of the equations correspond to actual biochemical reactions, so the use of Michaelis–Menten and other rate laws is purely symbolic.

Nonetheless, as illustrated in Fig. 4 and discussed in the caption, each modeling formalism implies a specific set of constraints on the model parameters. Because the models are mathematical abstractions, it is very important to interpret the biological/biochemical implications of these constraints

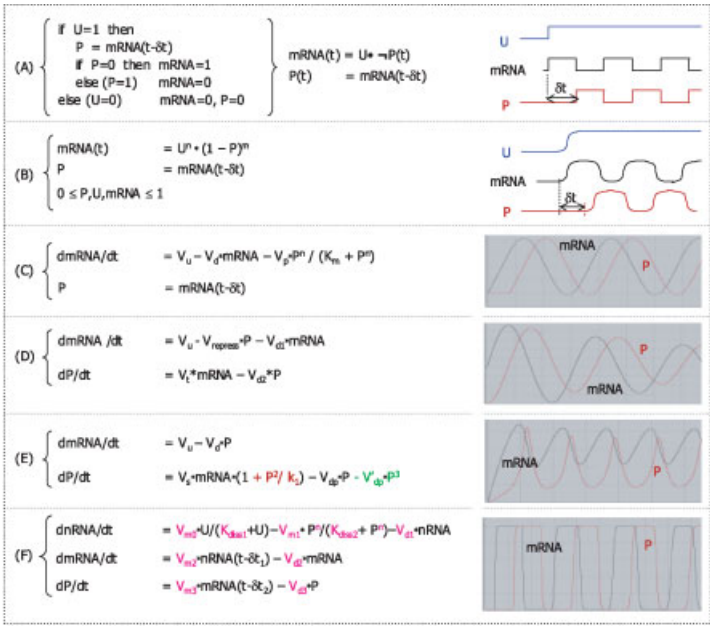

Figure 4. Models of a hypothetical single-gene oscillator where a gene's protein product represses its own transcription. The oscillator is activated/inactivated by a ubiquitous factor (U). Because of its extreme structural simplicity, we explore the behavior of this circuit as an example of the utility and limitations of different modeling formalisms. **A:** A Boolean logic model, in which the modeled variables can only be either fully active or fully inactive (binary). The model is described verbally (as in a computer program) by the "if <condition>then<outcome>" statements within the { } brackets, and using symbolic-logic shorthand (on the right-hand side). The symbol ¬ indicates logical inversion (not). The symbol ● indicates a logical and (a Boolean operator which returns high (active) if both conditions are true (high)). The shorthand (t-δt) indicates a delay of δt applied to the preceding variable (in this case mRNA). The top shorthand statement can be read as "mRNA level equals (U) and (not(P))". Thus U activates the circuit. If U is low/inactive, no expression takes place. If U is high/active, mRNA levels oscillate because the mRNA level is always the opposite of its own value some time earlier. The above Boolean model will oscillate so long as the delay δt is greater than zero. The period of oscillations (the length of time between expression peaks) equals 2δt. δt here includes the rise and fall times of the concentrations of all the intermediate molecular species involved. In our minimal Boolean model, the values of U, mRNA and P (the protein product of the gene) change instantly and the delay between changes in mRNA and P is modeled as a separate statement for simplicity. We could instead model the delay between mRNA and P changes by assigning each variable a finite rise and fall time (as is done in electronic circuit simulators). Because these rise and fall times are not known for most genes, we have lumped all of them into one term, thus simplifying the model. But our single parameter now stands for a mixture of several and its value does not reveal much about the modeled system. This is a basic, often unavoidable, trade-off in modeling. In general, delays determine when the "condition" in an "If <condition>then<outcome>" statement becomes true. Changes in assumed rise/fall times can result in radically different network behaviors. Electronic circuit designers spend considerable effort to ensure their circuits are robust to unavoidable manufacturing variations in the rise/fall times of their circuit elements. Such robustness may also be expected in developmental GRNs, which must progress through specific states at specific times and in specific cells, and cannot therefore vary with random variations in reaction kinetics. One implication of this observation is that robust developmental processes can be modeled with Boolean and algebraic logic without explicit formulation of the reaction kinetics. **B:** The same circuit modeled using an algebraic (continuous logic) formalism: the variables are now continuous valued, and the interaction functions not limited to those specified in Boolean logic. Here the Boolean and function is replaced with an algebraic multiplication. The powers "n" and "m" allow the modeler to give the interaction function smoother (low n, m values) or sharper, more threshold-like characteristics (higher n, m). The repressive effect of P on mRNA is modeled with a simple subtraction (causing the mRNA value to decrease in proportion to P, as P increases). Note, however, that the interaction and repression functions could be defined by any number of other algebraic formulae, so long as they are well behaved. For example, instead of $(1-P)$, the repression of mRNA by P could be modeled as $(1-P)/(1+P)$. This model reveals a circuit characteristic not apparent in the Boolean model: it is possible for P and mRNA to "hover" at $P = mRNA = 0.5$. This state is highly unstable. Random noise would normally drive the circuit out of this state. However, the existence of this meta stable steady state can be significant; for example it is widely used in microelectronics to implement very sensitive amplifiers for reading the contents of computer memory "chips". For oscillatory behavior, we need to avoid the preceding scenario. For this, the delay between mRNA and P must be non-negligible, and the repressive action of P on the gene must be non-linear. Thus, we see that an algebraic (continuous logic) model involves a larger number of parameters and interaction functions to be defined, but also can reveal more of the dynamical range of behaviors that may be exhibited by a circuit. In practice, lack of appropriate data for the choice of interaction functions and parameters limits the utility of this modeling formalism. However, such models have the advantage that, by normalizing all variables to the range 0–1, it is straightforward to mix this continuous notation with the Boolean notation

correctly. For example, for the Boolean model (Fig. 4A), the assumed total delay between changes in mRNA and protein levels equals half the period of oscillation. However, it would be wrong to imagine that this constraint implies a limit on the time it takes for mRNA molecules to be transcribed, modified, transported to the cytoplasm and translated. The reason is that we could equally have formulated the model based on a delay between protein level and the resulting change in transcribed mRNA level. In that case, we might have concluded (equally incorrectly) that there is a limit on the time it takes for transcription initiation, RNA elongation, and mRNA editing. In fact, all we can conclude from the Boolean model is that the sum of all non-linearities (including delays) in the modeled circuit must equal the period of oscillation.

The Boolean model of Fig. 4A has only one parameter: the mRNA–protein level delay. ODE-based models introduce many more parameters (10 in Fig. 4F), thus requiring even more careful interpretation. Some examples are discussed in the caption to Fig. 4.

Figure 4 is by no means a comprehensive list of all the modeling formalisms that could be usefully employed. Notably, stochastic[11,12] and multi-valued logic[18] models are not discussed. Stochastic frameworks may prove essential for modeling situations involving few molecules or where diversity generation is important (e.g., in the immune system). Multi-valued logic offers a useful balance between model complexity and representation / analysis powers.

Modeling gene regulation during development

Developmental genetic regulatory models are distinguished by the need for multicellular representation of gene activity. At any given developmental stage, cells within each territory will share the same set of active genetic regulatory interactions. Cells in different territories will have different gene activities. We visualize the set of gene interactions specific to a particular cell type at a particular time as the ''View From the Nucleus'' (VFN) of that cell type.[6] By contrast, the set of all gene interactions (in all cell types and at all developmental stages) constitutes the ''View From the Genome'' (VFG). From a modeling and simulation point of view, each VFN is a specific subset of the genes and interactions represented in the VFG. During simulation, each cell inherits the entire genome (i.e., the VFG), but its gene

Figure 4. (*Continued*)
described in (A). **C:** Same model as (A) and (B), represented with a delay-differential equation (after Ref. 22). Differential equation models have the advantage that the modeled interactions can take the form of ''pseudo chemical reactions'' (where the rate of change of a product is modeled as the algebraic difference between the rates of its production and degradation). Here the rate of mRNA transcription is increased in direct proportion (V_u) to the ubiquitous activator. It is decreased by a function of P that mimics the occupancy of P on its target *cis*-regulatory binding site and the effectiveness with which bound P inactivates transcription. The power ''n'' on the P occupancy function models the degree of non-linearity of the effect of P on transcription. With this formalism, we discover an additional property of the circuit: irrespective of the value of δt, true oscillations require that the repressive effect of P on mRNA transcription be non-linear with $n > 1$. A linear repression function, such as $(1-P)$ can only produce damped oscillations (where the amplitude of successive peaks increases indefinitely, or decreases to zero over time). See (**D**) for example damped oscillations. **D:** Although all of the above models require a finite delay between mRNA and P, oscillatory behavior can also be obtained without explicit delays. The simplest such model is shown in (D). As the mRNA and protein degradation rates (V_{d1} and V_{d2}) are reduced to zero, the oscillations of the model approach constant amplitude. Although this model is attractive for its simplicity, it has the undesirable property that its variables cannot represent concentrations or activity levels since they must take negative values for oscillations to occur. One can assume a 2nd set of variables $P' = P + C_p$ and $mRNA' = mRNA + C_{mRNA}$ such that P' and $mRNA'$ are always positive; but it is difficult to explain such a transformation biologically. **E:** Another two-equation model that oscillates without an explicit delay term. This model has the advantage that its variables are always positive and can therefore represent concentrations or activity levels. The model here is based on a mechanism proposed in Ref. 23. Instead of delay, the rate of change of P has two thresholds: one for when P is decreasing, and another for when P is increasing (this is due to the two non-linear functions highlighted in red and green). The outcome is similar to (C) in that P follows the mRNA level with a time lag (with additional non-linearity). Comparison of (C) and (E) provides a cautionary lesson in interpreting such phenomenological models too literally. Both models generate oscillatory behavior. But the most distinct feature of (C), the requirement for an explicit delay term between mRNA and P, is absent in (E). This apparent contradiction is just because it is possible to model the same phenomenon using very different mathematical approximations. The processes modeled as delay in (C), are represented by non-linear interactions in (E). Neither model is more right or wrong than the other; and neither should be interpreted too literally. Ultimately, both models lead to the same observation: oscillations in these abstract networks require non-linear regulatory interactions. **F:** As with the delay-based models, the phenomenological interactions in (E) cannot be ascribed to specific molecular processes. The simplest model whose terms could be interpreted as concentrations is shown in (F). The first equation models transcription of nuclear RNA (nRNA). The repression of transcription by P is modeled using a ''non-competitive inhibition'' type formalism. The second and third equations model mRNA and protein production/decay. Even though (F) is a massive simplification of the many molecular processes involved, it nonetheless ''boasts'' 10 ''kinetic'' parameters (shown in lavender). To have any confidence in the model as a reflection of the modeled network, it is necessary to estimate these parameter values and demonstrate that the model is insensitive to all plausible variations in parameter estimates.

expression state is the VFN inherited from its parent cell type. This VFN may then be modified through intracellular and intercellular interactions.

In our NetBuilder (http://strc.herts.ac.uk/bio/maria/NetBuilder/) network capture and simulation software,[7] we use the convention that inactive genes (and their outputs) in VFNs are either shown in light gray, or omitted from the view, but the position and connectivity of all genes remains the same as represented by the VFG. Figure 7B shows an example VFG; while Fig. 7C shows two VFNs of the same network. See Refs. 5,6,14 for more examples.

Putative building blocks of animal GRNs

We suspect that animal GRNs are modular in structure in that there is an enumerably small set of GRN "building blocks" from which larger GRNs are constructed. As with engineered systems, it is likely that larger modules will be hierarchically built up from combinations of smaller ones. Thus, at the top of the hierarchy, there can be a very wide variety of large GRN modules, while at the bottom of the hierarchy, the number of small building blocks can be limited. For example, consider the different ways in which a gene may regulate its own activity. Mechanistically, there are only two possibilities: enhancement, or repression. Depending on kinetic parameter values (e.g., association/dissociation constants), each of these can lead to just a few canonical forms of behavior. A self-enhancing feedback loop (also known as positive feedback or auto-regulation) can:

- amplify the effect of an incident regulatory input such as an intrinsic, cell-specific factor, or an intercellular signal;
- rapidly drive the expression level of an activated gene;
- maintain gene expression in response to a transient activating signal.

Similarly, self-repression (negative feedback) by a gene may:

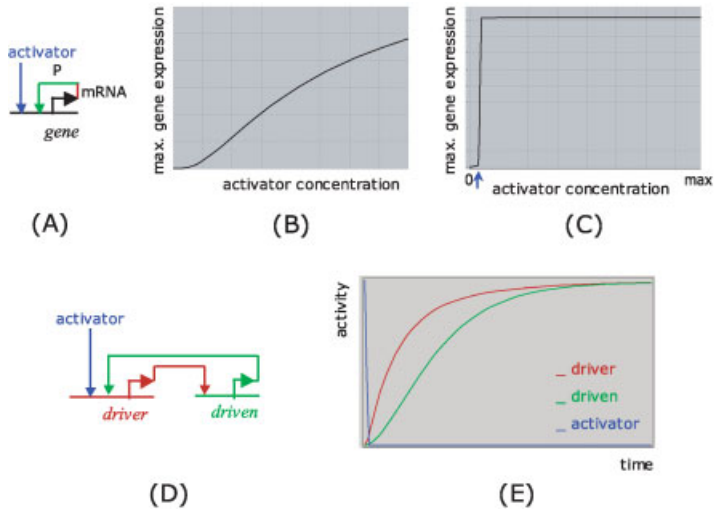

Figure 5. Putative intracellular transcriptional positive feedback lock-on switches. **A:** Single gene with autocatalytic feedback. The gene is initially activated by one or more regulatory factors (in early development, the activator may be maternal). Once activated, the gene remains maximally active because its protein product is a transcriptional enhancer of itself. **B:** Simulated gene expression (protein concentration) level as a function of activator concentration for a gene without positive feedback (i.e., circuit and equations equivalent to Fig. 1). **C:** As (B) but with autocatalytic feedback (i.e., the circuit shown in A). Note the threshold-like response curve. This is an example of a system feature brought into focus through simulation. While this observation may be obvious in the current toy model, it serves to illustrate the explanatory utility of simulation. In larger-scale models, similar insights can be much easier to arrive at through modeling and simulation than through "thought experiments" with box and arrow diagrams **D:** Example of an intracellular positive feedback lock-on switch involving more than one gene. Here *driver* and *driven* activate each other, forming a two-gene positive feedback loop. **E:** Simulation of expression profiles of the two genes over time. The *driver* gene is the one that receives an initial activating input. Note how the activity of the second (*driven*) gene closely follows that of the first. In this example, the activator signal (shown in blue) is only "on" very briefly at the beginning of the simulation, but is enough to trigger the positive feedback loop between the genes such that their expression levels rise on to saturation even though the activator is no longer present. Once activated, the genes remain on (presumably until one or more dominant repressors (not shown) disrupt the feedback).

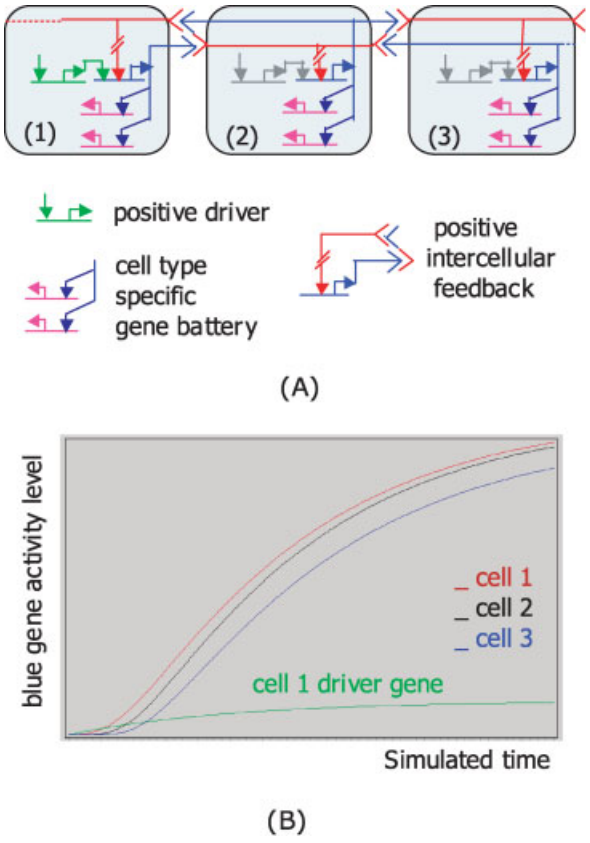

Figure 6. Putative intercellular positive feedback loops implementing a "Community Effect."[19] **A:** Schematic wiring diagram showing the necessary interactions between genes in three cells. Note that the genetic network in all three cells is identical, and signaling only takes place between neighboring cells. The blue genes are in a positive feedback loop with their counterparts in neighboring cells (through the ligand–receptor interactions symbolically represented by double chevrons). The green genes are upstream activators of the blue genes. As with the circuits in Fig. 5, there is a threshold below which the activator signal cannot trigger the circuit on, but any signal level above this value is enough for activation. For the illustrative simulation results in (**B**), we have specified the extreme condition where only the green gene in cell 1 is active, and that at only 10% of its maximal rate (green curve). Nonetheless, as the graphs show, the downstream genes in all three cells are activated to near their maximal rates due to the intercellular positive feedback loops. Note that cell 3, which is farthest from cell 1, receives the effect of the activator in cell 1 only via cell 2, and as a result is activated slightly later but approaches near maximal expression levels nonetheless. Note also how the gene in cell 2 has a slightly sharper expression profile because it receives positive feedback from two neighboring cells (whereas cells 1 and 2 have only one neighbor each). In Fig. 5E, the activator signal was turned on maximally, but only for a brief moment. Here, the activator gene is turned on at a low level but left on indefinitely. In both cases, even an activation signal that is both low-level (but above threshold) and short-duration is enough to drive the genes in the loop to saturation expression levels.

- limit the rate of expression of the gene to a fixed steady-state level; or
- cause the expression level of the gene to oscillate.

It is straight-forward to construct combinations of the above behaviors by mixing positive and negative feedback. For instance, one may imagine a gene that expresses at a fixed level in response to very small or transient extracellular changes. Such a gene might use autoregulation to amplify the signal and self-repression to produce a predetermined level of expression once an input has been detected.

Figures 5–7 describe some example putative GRN building blocks that we have come across while reverse engineering a GRN that operates very early on during the development of sea urchin embryos.[5,14] These include examples of:

- Single and two-gene positive feedback loops that appear to be used as a means of ensuring the unidirectional progress of developmental processes (Fig. 5).
- Positive feedback (community effect, Ref. 19) between genes in different cells, mediated by complex signaling pathways (Fig. 6). These appear to be used in development to ensure that all cells within a territory adopt the same fate.
- Repression gene cascades, which appear to be used to define sharp spatial boundaries between cells of different future territories (Fig. 7).

The use of modeling to explore the dynamics of each of the above putative building blocks is discussed in the accompanying figure captions. Needless to say, the putative building blocks presented in these figures represent a small proportion of the likely total number that must be utilized in animal GRNs. We hope readers of this article will seek, find, and present many more in future.

Discussion

As discussed earlier and illustrated in Fig. 4, abstract models have relatively few parameters and so, on the one hand, it is easy to explore their behavior and build models with them. On the other hand, the parameters they do have are combinations of many factors. In contrast, more detailed models suffer from an explosion in the number of their parameters; as a comparison of Fig. 4A and F reveals. As illustrated in cartoon form in Fig. 8, a large number of parameters can make it very difficult to compare alternative models. Paradoxically, the opposite can also be true. The network structure of cellular processes is sometimes so intricately defined as to make their behavior largely independent of parameter values, see for example Refs. 20,21.

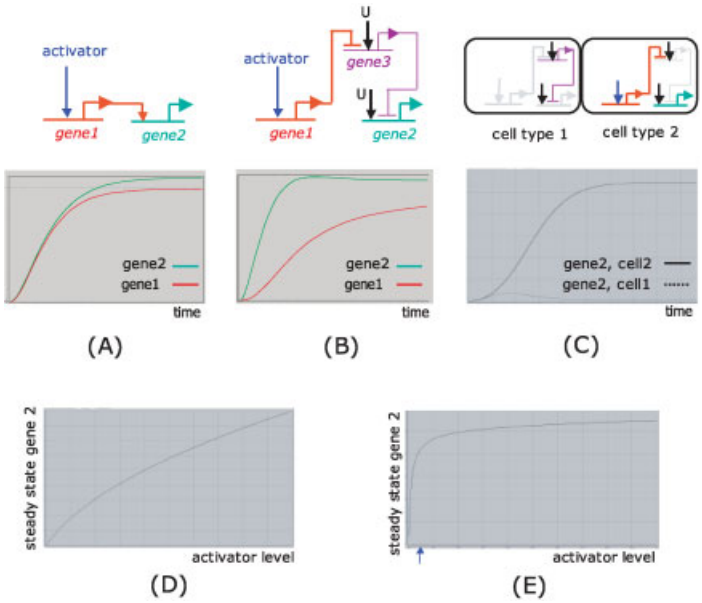

Figure 7. An example putative circuit block for defining sharp spatial boundaries. Suppose *gene1* is differentially expressed in two groups of cells. **A:** The simplest circuit for boundary definition between the two cell types would involve *gene1* directly activating other genes. If, for simplicity, we model all genes with the same kinetic equations and parameters, then *gene2* expression will closely follow *gene1* expression, as shown in the simulation results below. **B:** Activation through a double-repression cascade (top: "View from the Genome," bottom: simulated gene expression profiles). Again, all genes are modeled with the same kinetics. Lifting a repression can result in a sharper rise in activity for the downstream gene, thus allowing greater control of its expression. This is due to the possible "priming" role of ubiquitous activators (black arrows). The effect is like locking down (repressing) a catapult while it is loaded (acted on by ubiquitous activators), then suddenly releasing the lock (lifting the repression). **C:** Illustration of how the repressor cascade can lead to differential expression in neighboring cells. Top, "Views From the Nuclei" of two cells in which *activator* is differentially regulated. Gray text and lines indicate inactive network components in a cell. Bottom, simulated expression curves for the gene 2 protein in the two cell types. Cell 2 has inherited a higher concentration of an activating factor than cell 1. As with the circuits in Figs 5–6, there is a concentration threshold below which the activating factor cannot trigger the target gene (see E). In cell 1, activator concentration is below this threshold and hence *gene1* is not active. The result is activation of the *Repressor* (lavender) gene (by the ubiquitous driver, black arrow) and repression of *gene2* (green). In cell 2, the situation is reversed and *gene2* is repressed (after a brief small transient while the repressor gene turns on). **D:** Direct activation of one gene by another has no distinct activation/inactivation threshold and results in a fairly smooth direct relationship between activator concentration and *gene2* activity. **E:** In contrast, the activator–response curve of a repression cascade has a distinct activation threshold (blue arrow) due to the highly non-linear dependence of *gene2* expression on activator level. When the activator is an asymmetrically distributed factor, such a repression cascade will define a sharp gene-expression boundary between cells with factor concentrations just below and just above the threshold.

The reverse-engineering methodology outlined earlier address the above conundrum by seeking to identify parameter correlations before parameter values are considered. This is achieved by resolving the network structure first, then identifying the types of interaction between the nodes in the network, and only then considering the dynamic effects of parameter values. The particularity of a given large GRN does not lie in the specific set of small GRN modules of which it is composed. These building blocks are utilized over and over again in diverse GRNs that accomplish different developmental tasks, such as building different parts of the body. The particularity of large GRNs is to be found at the highest level of their organization, and this is what is made explicit in the network structure. There are probably a very large number of different large GRNs because there is a very large number of different ways that a limited set of elemental modular mechanisms can be linked together. This is the GRN feature that underlies the diversity of developmental process, and this is also why the primary task in GRN analysis is to resolve the network structure and its linkages. A useful complementary approach may be to first identify minimal complexity building blocks, then search for larger blocks using these minimal blocks, and so on hierarchically. The combination of these top-down and bottom-up

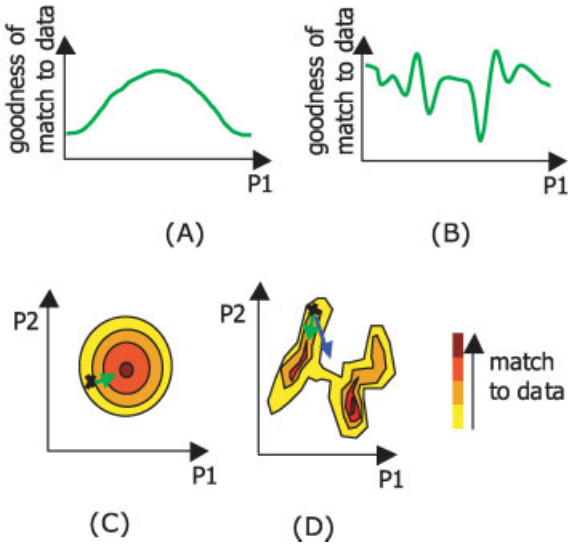

Figure 8. Cartoon illustration of the "curse of dimensionality," Models are usually developed to mimic some experimentally observed feature of a system. For instance, for the oscillator models in Fig. 4, one may be interested in matching the period of oscillations of the models to a set of experimental expression profiles. The models in Fig. 4 have between one and ten parameters. For simplicity, consider a model with just one parameter P_1 (e.g. model (A) in Fig. 4, where the only parameter is δt). The graph in (A) represents one possible scenario in which P_1 affects the behavior of the model in a simple, easily predictable way. Once the regulatory gene linkages have been identified (the network connectivity is determined) and model interactions (equations) defined, one may search for the value of P_1 that results in model behavior most closely matching experimental data. **A:** For the simple scenario, one need only start with a random guess for the value of P_1 (a random location along the P_1 axis in the figure) and iteratively change P_1 in the direction that increases the goodness of fit to data. Unfortunately, things are frequently not so simple. **B:** Represents a more usual scenario. Here, the optimal fit of P_1 to experimental data lies at the peak of one of several local optima. Starting from a random value for P_1, it is now necessary to look not just for the nearest peak, but for the *highest* or *broadest* peak (i.e., the best fit to data or the least sensitive to variations in P1). This requires considerably more computation. **C,D:** Illustration of the increasing difficulty of searching multi-dimensional spaces. Same scenario as above, but now for a hypothetical model with two parameters (for example parameters δt and n in the model in Fig. 4C). The goodness of fit of the model (e.g., the difference between experimentally measured and model oscillation periods) is plotted as a function of the values of the two parameters. Coloring denotes the degree of match between model behavior and experimental data. The figures can be viewed as "contour maps" of the goodness of fit landscape. (C) Is analogous to (A), while (D) is analogous to (B). The colored areas in (C) and (D) are roughly the same size; but, starting from a random estimate of P1, P2 values (e.g., the black cross), it is much harder to find the maroon-colored peak in (D). In (C), one can start from any random position (e.g., black cross) and find the maroon area (best fit) by simply climbing up the local contour gradient (green arrow). This strategy does not work in (D). Instead of climbing up the local gradient (green arrow), one needs to identify the direction towards the highest peak (blue arrow). Put another way, correlations between the effects of parameters on system behavior give the colored area in (D) a concave characteristic, so that—unlike (C)—it is not possible to find the optimum parameter fit by simply climbing up the local gradient. Thus, the parameter search problem becomes considerably more difficult as the number of dimensions (number of parameters to be estimated) increases. One solution to this problem is to guide the search by identifying some of the dependencies (correlations) between parameter values prior to any search. This can be done for instance, by first identifying the connectivity structure of a network (what interactions exist) before quantifying the nature of the interactions, as embodied in our reverse-engineering methodology (see text).

strategies offers exciting opportunities in reverse-engineering GRNs.

Acknowledgments

We are grateful to Drs. Mark Borisuk, Brian Ingalls, Herbert Sauro, Maria Schilstra, Denis Thieffry, Olaf Wolkenhauer, Adam Wilkins and Tau Mu Yi for helpful discussion and comments on early drafts of this manuscript.

References

1. Smolen P, Baxter DA, Byrne JH. Modeling transcriptional control in gene networks—Methods, recent results, and future directions. Bull Math Biol 2000;62:247–292.
2. Rao CV, Arkin AP. Control motifs for intracellular regulatory networks. Ann Rev Biomed Eng 2001;3:391–419.
3. Davidson EH. Genomic Regulatory Systems—Development and Evolution, San Diego: Academic Press; 2001.
4. Monod J. Chance and Necessity; an Essay on the Natural Philosophy of Modern Biology. Translated from the French by Austryn Wainhouse. New York: Knopf; 1971.

5. Davidson EH, et al. A genomic regulatory network for development. Science 2002;295:1669–1678.
6. Bolouri H, Davidson EH. Modeling DNA sequence-based *cis*-regulatory gene networks. Dev Biol 2002;246:2–13.
7. Brown CT, et al. New computational approaches for analysis of *cis*-regulatory networks. Dev Biol 2002;246:86–102.
8. de Jong H. Modeling and simulation of genetic regulatory systems: a literature review. J Comput Biol 2002;9:67–103.
9. Wyrick JJ, Young RA. Deciphering gene expression regulatory networks. Curr Opin Genet Dev 2002;12:130–136.
10. D'haeseleer P, Liang SD, Somogyi R. Genetic network inference: From co-expression clustering to reverse engineering. Bioinformatics 2000;16: 707–726.
11. Shea MA, Ackers GK. The Or control system of the bacteriophage lambda—a physical-chemical model for gene regulation. J Mol Biol 1985;181:211–230.
12. McAdams HH, Arkin A. Stochastic mechanisms for gene expression. Proc Natl Acad Sci USA 1997;94:814–819.
13. Mjolsness E, Sharp DH, Reinitz J. A connectionist model of development. J Theor Biol 1991;152:429–453.
14. Davidson EH, et al. A provisional regulatory gene network for specification of enomesoderm in the sea urchin embryo. Dev Biol 2002;246:162–190.
15. Yuh C-H, Bolouri H, Davidson EH. Genomic *cis*-regulatory logic: Functional analysis and computational model of a sea urchin gene control system. Science 1988;279:1896–1902.
16. Yuh C-H, Bolouri H, Davidson EH. *cis*-Regulatory logic in the *endo16* gene: Switching from a specification to a differentiation mode of control. Development 2001;128:617–628.
17. Yuh C-H, Bolouri H, Bower JM, Davidson EH. A logical model of *cis*-regulatory control in a eukaryotic system. In: Bower JM, Bolouri H. editors, Computational Modeling of Genetic and Biochemical Networks. Cambridge. MIT Press. 2001. pp. 73–100.
18. Sanchez L, Thieffry D. A logical analysis of the *Drosophila* gap-gene system. J Theor Biol 2001;211:115–141.
19. Gurdon JB. A community effect in animal development. Nature 1988; 336:772–774.
20. von Dassow G, Mier E, Munro M, Odell M. The segment polarity network is a robust development module. Nature 2000;406:188–192.
21. Barkai N, Leibler S. Robustness in simple biochemical networks. Nature 1997;387:913–917.
22. Glass L, MacKey MC. From clocks to chaos—the rhythms of life. Princeton University Press. 1988.
23. Tyson JJ. Modeling the cell division cycle—CDC2 and cyclin interactions. Proc Natl Acad Sci USA 1991;88:7328–7332.