



Available online at www.sciencedirect.com



**PHYSICS of LIFE
reviews**

ELSEVIER

Physics of Life Reviews 2 (2005) 65–88

www.elsevier.com/locate/plrev

Reverse-engineering transcription control networks

Timothy S. Gardner*, Jeremiah J. Faith

Department of Biomedical Engineering, Boston University, 44 Cummington St., Boston, MA 02215, USA

Received 13 January 2005; accepted 13 January 2005

Abstract

Microarray technologies, which enable the simultaneous measurement of all RNA transcripts in a cell, have spawned the development of algorithms for reverse-engineering transcription control networks. In this article, we classify the algorithms into two general strategies: physical modeling and influence modeling. We discuss the biological and computational principles underlying each strategy, and provide leading examples of each. We also discuss the practical considerations for developing and applying the various methods.

© 2005 Elsevier B.V. All rights reserved.

PACS: 87.16.-b; 87.16.Ac; 87.16.Xa; 87.16.Yc; 87.17.-d; 87.17.Aa; 87.80.Vt

Keywords: Gene networks; Reverse-engineering; Machine learning; Transcription control; Gene regulation

Contents

1. Introduction	66
2. Gene expression in a nutshell	67
2.1. Mechanisms of RNA regulation	68
2.2. A simple model of transcription	69
3. Reverse-engineering: an introductory example	70
4. Two general reverse-engineering strategies	72
5. The physical strategy: identifying TF interactions	73
6. The influence strategy: inferring gene networks	75
6.1. Differential equation models	76

* Corresponding author.

E-mail address: tgardner@bu.edu (T.S. Gardner).

6.2. Boolean network models	77
6.3. Bayesian network models	80
6.4. Association networks	82
7. Challenges	84
8. Summary	85
Acknowledgements	86
References	86

1. Introduction

One decade ago, DNA microarray technologies [1–3] were developed which enabled an experimenter to simultaneously measure the concentration of thousands of RNA transcripts from a single sample of cells or tissues. Such data offered the possibility to infer, or “reverse-engineer,” a model of a cell’s underlying transcription control systems (Fig. 1). Engineers and scientists have previously developed reverse-engineering techniques in the fields of computer science, engineering, and statistics, which are respectively called machine learning, system identification, and statistical learning. Hence, not long after DNA microarray technology emerged, researchers proposed plausible approaches to reverse-engineer the mechanism of transcription control in cells [4–10].

Continuing research [11–15] has dramatically expanded the types of reverse-engineering approaches and their application to experimental data. Still, the development of reverse-engineering methods remains a challenging and active area of research. Challenges principally arise from the nature of the data; they are typically noisy, high dimensional, and significantly undersampled. Moreover, well-understood and standardized benchmark systems for validating algorithm performance are not available. Thus, significant questions still remain regarding experimental design, the reliability of the predicted networks, and the utility of various approaches for particular applications.

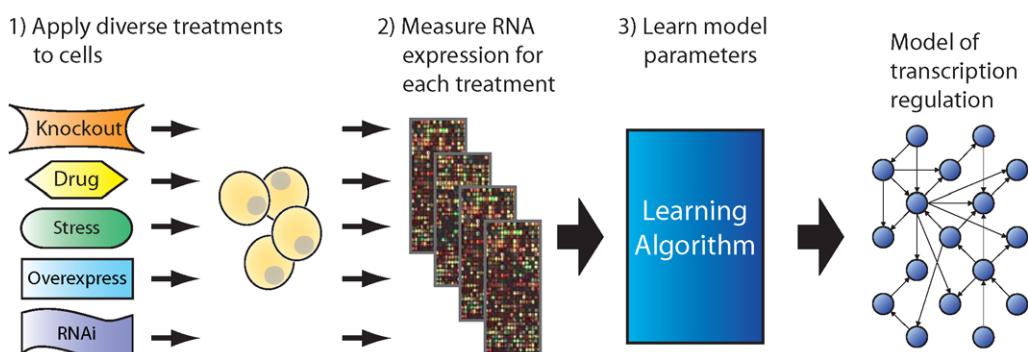


Fig. 1. The general strategy for reverse-engineering transcription control systems. (1) The experimenter perturbs cells with various treatments to elicit distinct responses. (2) After each perturbation, the experimenter measures the expression (concentration) of many or all RNA transcripts in the cells. (3) A learning algorithm calculates the parameters of a model that describes the transcription control system underlying the observed responses. The resulting model may then be used in the analysis and prediction of the control system function.

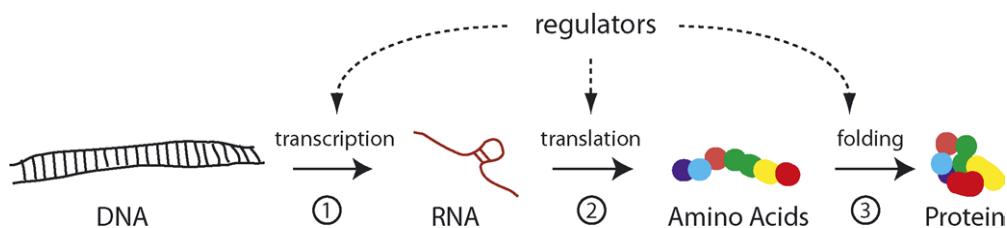


Fig. 2. Gene expression is a three step process. (1) A region of DNA (a gene) is *transcribed* into RNA. (2) RNA is *translated* into a chain of amino acids. (3) The amino acid chain is *folded* into its functional 3-dimensional protein structure. Regulation of cell dynamics can occur at any of these steps.

In this article, we review promising examples of several classes of algorithms for reverse-engineering transcriptional control systems using RNA transcript measurements. We group these approaches into two general strategies: “physical” approaches, which seek to identify the molecules that physically control RNA synthesis, and “influence” approaches, which seek to model causal relationships between RNA transcripts. The causal relationships may or may not correspond to true molecular interactions. Within both strategies, similar computational techniques may be applied, including linear regression, Boolean inference, Bayesian learning, correlation and mutual information.

We begin with a review of the fundamental concepts of gene expression related to reverse-engineering methods. We then provide an introductory example of reverse-engineering applied to transcription regulation. We next discuss the principal reverse-engineering strategies developed to date. Finally, we conclude with brief comments on the practical considerations for developing and applying the various methods.

2. Gene expression in a nutshell

Gene expression refers to the process by which cells produce proteins from the instructions encoded into DNA [16]. In essence, one may consider it to be a three step process (Fig. 2): first, specialized proteins *transcribe* a region of DNA (a gene) into an RNA molecule (also called a *transcript*); second, additional proteins process the RNA transcript and *translate* it into an polypeptide chain; third additional proteins fold and modify the polypeptide chain into a biochemically active protein. Regulatory molecules can control the concentration and form of the product of each step. These regulators are usually fully-formed proteins, but any of the intermediate products (RNA, polypeptides, or proteins) also may act as regulators of gene expression.

Reverse-engineering techniques have principally focused on decoding the mechanisms of transcription control, the first step in gene expression. This is because DNA microarray technology has enabled researchers to efficiently measure the concentration of all RNA transcripts in a cell, making such data abundant. Measuring peptide, protein and metabolite regulators of gene expression is generally more difficult, and such data are not often available. But with improved technologies for protein and metabolite measurement, reverse-engineering techniques may be extended to the second and third steps of gene expression.

Microarray technology consists of glass slides or silicon chips containing thousands, or millions, of DNA probes, each of which is complementary to a specific RNA species in the cell [1–3]. Each probe can bind to, and quantitatively measure, the concentration of an individual RNA species. Due to variations in probe sensitivities, the technology can reliably measure only relative changes in RNA concentrations.

Thus, one reports RNA measurements as concentration ratios for each transcript relative to its baseline state. For example, we can determine that two different transcripts both double in concentration, but we will not know if one of the two transcripts is more abundant than the other.

2.1. Mechanisms of RNA regulation

RNA levels in a cell can be controlled via the rate of synthesis or the rate of elimination (degradation or modification to another form). The rate of degradation of RNA can be modulated by RNA-degrading proteins. One usually assumes that RNA degradation is nonspecific, and thus, not a dominant mode of control. There are exceptions, however, such as the *mazF* protein in *E. coli* [17–19], which appears to degrade specific RNA species under stress conditions.

RNA synthesis (transcription) is controlled by the activity of RNA polymerase (RNAP), the protein complex that reads the DNA and copies it into RNA. Transcription of DNA begins when RNAP recognizes and binds to a *promoter*, which is a control sequence of DNA upstream of the transcribed region (Fig. 3a). After binding, RNAP opens the DNA double helix and slides along the DNA sequence, elongating the RNA message by adding ribonucleotides that match the DNA sequence. Transcript elongation proceeds until the RNAP encounters a stop sequence in the DNA. Factors that bind directly to the RNAP complex can modulate the binding rate, the binding specificity, the rate of RNA elongation, and the termination of elongation. Examples include the ppGpp molecule, which alters the binding specificity of

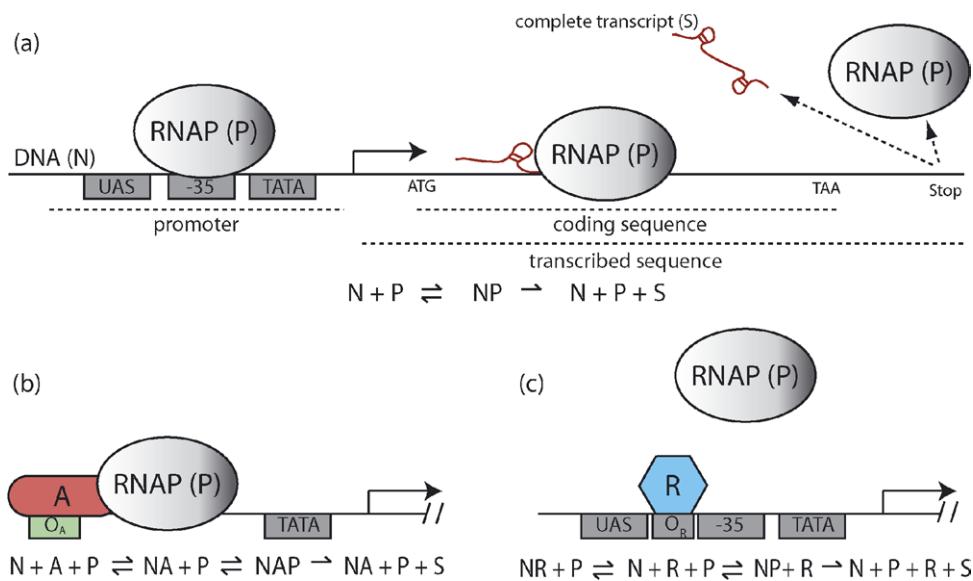


Fig. 3. (a) RNA transcription begins when an RNA polymerase (RNAP) binds to upstream recognition sequences on the DNA sequence (promoters). The two strands of DNA are separated and RNAP moves along the DNA, transcribing an RNA copy of it until a stop region on the DNA knocks the polymerase off. (b) Some genes have an activator protein that binds to a motif (O_A) near the promoter. This protein increases the affinity of RNAP for the promoter. (c) Other genes have repressor proteins that bind to motifs (O_R) in the promoters. Repressor proteins can act by blocking access of RNAP to important regions of the promoter.

RNAP under stress conditions in many microbes [20], and the N protein, which alters the termination site of RNAP in bacteriophage lambda DNA [21].

Transcription can be controlled also by an auxiliary protein called a *transcription factor* (TF). A TF binds to a short, unique DNA sequence, called a *motif*, in or near the promoter (Fig. 3b). When bound to its motif in a promoter, a TF can recruit RNAP to atypical promoters, or block the binding of RNAP to promoters. Because this form of regulation is highly specific to particular RNA transcripts, it appears to be the predominant form of RNA regulation in cells.

2.2. A simple model of transcription

We can obtain a simple model of transcription by considering the RNAP (P), the DNA promoter (N), and the RNA transcript (S) as chemical species in a well-stirred environment. Applying the principles of rate-law kinetic modeling, we obtain:



where k_{+1} and k_{-1} describe the on and off rates of RNAP binding, and k_{+2} reflects the transcript elongation rate. Under the assumption that k_{+2} is small, or the assumption that the concentration of promoter (N) is much smaller than the concentration of RNAP (P), we can write a differential equation for transcription as:

$$\frac{ds}{dt} = \frac{V_m p}{p + K_m} - \delta s, \quad (2)$$

where the lowercase letters denote the concentrations of the respective species, $V_m = k_{+2} n$ is the maximum rate of synthesis, $K_m = (k_{-1} + k_{+2})/k_{+1}$ is the Michaelis constant, and δ accounts for a constant rate of degradation of RNA transcript. More formally, the above assumptions create fast and slow time scales in the reaction, and Eq. (2), describes the slow dynamics. K_m describes the activation threshold for p . For concentrations of p above K_m , the rate of synthesis begins to saturate at its maximum value, V_m .

Given the small number of reacting molecules, the omission of several other chemical species involved in transcription, and the inhomogeneous nature of the cell contents, Eq. (2) is an imperfect model. Nevertheless, it has proven effective for capturing qualitative and quantitative features of gene expression [22–31].

We can also incorporate regulatory control into the rate-law model of transcription. For example, RNAP does not recognize some promoters without the help of an activating TF that first binds to the promoter (Fig. 3b). This reaction scheme leads to the following modification of Eq. (2):

$$\frac{ds}{dt} = \left(\frac{a^\alpha}{a^\alpha + K_a} \right) \frac{V_m p}{p + K_m} - \delta s, \quad (3)$$

where a denotes the concentration of A , the activator, and K_a is the activation threshold for A . The activator simply modulates the maximum rate of transcription. The parameter α in the equation is a cooperativity exponent. It is determined by the number of copies of the binding motif for A in the promoter. If M_a is the number of binding motifs that must be occupied by A in order to recruit RNAP, then $\alpha = M_a$.

Transcription from promoters may also be inhibited by repressor TFs. Multiple forms of inhibition are possible. A common one is competitive inhibition, which means that the binding of the repressor TF and the RNAP are mutually exclusive (Fig. 3c). For competitive inhibition, Eq. (2) is modified as:

$$\frac{ds}{dt} = \frac{V_m p}{p + K_m(1 + r^\beta/K_r)} - \delta s, \quad (4)$$

where r denotes the concentration of the repressor, R , and K_r is the inhibition threshold for R . The repressor can be viewed as modulating the activation threshold, K_m , of the promoter by RNAP. The parameter β , is similar to α , and is determined by, M_b , the number of copies of the binding motif for R in the promoter.

3. Reverse-engineering: an introductory example

Identifying the binding motifs of TFs in promoters is a principal challenge in reverse-engineering the mechanisms of transcriptional control. Bussemaker, et al. introduced a promising strategy for combining RNA expression data and genome sequence data to identify the binding motifs of transcription factors in promoters [11]. The method assumes that the rate of transcription from a particular promoter will reflect the number of binding motifs for a TF in that promoter. The method begins by exhaustively enumerating all possible binding motifs of a given length of nucleotides, and then counting the number of copies of each possible motif in every promoter in a cell. Most of these putative motifs are not real. True motifs are distinguished because the copy number of true binding motifs in a promoter will correlate linearly with the logarithm of RNA levels measured using a microarray.

The rationale for the approach follows from the model of transcription that we developed in the previous section. Under steady state expression conditions (i.e., $ds/dt = 0$), the measured RNA level provides an estimate of the rate of synthesis. For example, Eq. (2) gives:

$$s = \left(\frac{1}{\delta}\right) \frac{V_m p}{p + K_m}. \quad (5)$$

We assume that the TF concentrations are far from saturating a promoter at its maximum rate of transcription, e.g., $a^\alpha < K_a$ and $r^\beta > K_r(p/K_m + 1)$. Near saturation, RNA measurements are uninformative because changes in TF activity or motif copy number will have little influence on transcription; thus, the motif-finding method will work poorly in this regime. With the saturation and steady state assumptions, we can write Eq. (3) for activation as:

$$s = a^\alpha \left(\frac{1}{\delta K_a} \frac{V_m p}{p + K_m} \right),$$

and Eq. (4) for repression as:

$$s = r^{-\beta} \left(\frac{K_r V_m p}{\delta K_m} \right).$$

The quantities inside the parenthesis are constant. Taking the logarithm gives:

$$\log s = \alpha \log(a) + \text{constant},$$

for activation, and

$$\log s = -\beta \log(r) + \text{constant},$$

for repression. We recall that $\alpha = M_a$ and $\beta = M_b$. Assuming that both repression and activation are simultaneously possible in a single promoter, we obtain a general expression relating motif copy number to the logarithm of RNA levels:

$$\log s = \sum_i \gamma_i M_i + \gamma_0, \quad (6)$$

where M_i is the copy number of motif i in the promoter, γ_i represents the log concentration of the TF i in the cell, and γ_0 is a constant.

The Bussemaker algorithm determines the coefficients, γ_i , of Eq. (6) by regressing the copy number of each motif in each promoter against the log RNA level measured for each promoter in a particular experiment. (In practice, one uses the log *ratio* of RNA relative to a reference state.) The algorithm assumes that the combination of motifs providing the most significant regression fit is the true set of active motifs in the measured experiments.

Selecting the best combination of motifs can be a computationally hard task. The number of putative binding motifs in promoters is large, and only a small subset of them are true motifs. For example, more than 20 000 motifs of 7 nucleotides exist. The Bussemaker algorithm may detect on the order of 10 of them as active in one experiment. Exhaustive scoring of every subset of motifs is infeasible due to the huge number of combinations. Thus, the algorithm uses an iterative, greedy search for the best-fitting combination. In each iteration, the algorithm fits motifs one at a time and selects the best fitting motif. It then subtracts the influence of that motif from the expression data and fits another motif to the residual signal. The algorithm repeats the process until the addition of motifs does not improve the fit. This search scheme assumes that the contribution of each motif to the RNA expression is independent.

Due to the size and complexity of transcription control networks and limitations on the amount of experimental data, a combinatorial search problem like that encountered in the Bussemaker algorithm is almost universal in reverse engineering approaches. Researchers have applied several strategies to address the problem, including heuristic strategies like the Bussemaker approach, Monte Carlo methods [32,33], dimensional reduction [34–36], and prior information [13,37]. Selection of an optimal strategy for each approach remains a challenge.

The success of the Bussemaker approach depends on a number of assumptions regarding the dynamics of transcription. Most notably, it relates the influence of combinations of TFs as a log-linear function of RNA levels. Such a highly constrained model may lead to errors in predictions. Indeed, the authors of the study point out that the log-linear model captured at best 30% of the signal present a test data set of yeast expression experiments. More complex nonlinear models of transcription may be more suitable. One such approach [14] which also captures location and orientation features of binding motifs, is described below. Despite its limitations, the Bussemaker method successfully analyzed yeast sequence and expression data; it identified many of the known binding motifs present in previously-studied promoters.

Another limitation of the Bussemaker approach is that it cannot determine which TFs bind to the discovered motifs. One strategy to overcome this problem requires that RNA levels are measured after directly perturbing the expression of a TF [38]. The most significant binding motifs that the Bussemaker algorithm detects in that experiment are assumed to be those bound by the perturbed TF. In tests on

experimental data, this method typically ranked the true binding site of a TF as the first or second most significant identified motif. Thus, it may provide a means to systematically determine the network of TF regulators and their binding motifs in a cell.

4. Two general reverse-engineering strategies

The Bussemaker approach is an example of a “physical” strategy [39] for reverse-engineering transcription regulation using RNA expression data. The physical approach seeks to identify the protein factors that regulate transcription, and the DNA motifs to which the factors bind. In other words, it seeks to identify true physical interactions between regulatory proteins and their promoters. An advantage of this strategy is that it can reduce the dimensionality of the reverse-engineering problem by restricting possible regulators to TFs. It also enables the use of genome sequence data, in combination with RNA expression data, to enhance the sensitivity and specificity of predicted interactions. The limitation of this approach is that it cannot describe regulatory control by mechanisms other than transcription factors.

A second strategy, which we call the “influence” approach, seeks to identify regulatory influences between RNA transcripts. In other words, it looks for transcripts that act as “inputs” whose concentration changes can explain the changes in “output” transcripts (Fig. 4). Each transcript may act as both an input and an output. The input transcripts can be considered the regulators of transcription. By construction, such a model does not generally describe physical interactions between molecules since transcription is rarely controlled directly by RNA (and never by messenger RNA, which is the type of RNA predominantly measured by DNA microarrays). Thus, in general, the regulator transcripts may exert their effect indirectly through the action of proteins, metabolites and effects on the cell environment (Fig. 5). Nevertheless, in some cases, the regulator transcripts may encode the TFs that directly regulate transcription. In such cases, the influence model may accurately reflect a physical interaction.

An advantage of the influence strategy is that the model can implicitly capture regulatory mechanisms at the protein and metabolite level that are not physically measured. That is, it is not restricted to describing only transcription factor/DNA interactions. As described in the section on differential equation models, an influence model may be advantageous when trying to predict the global response of the cell to

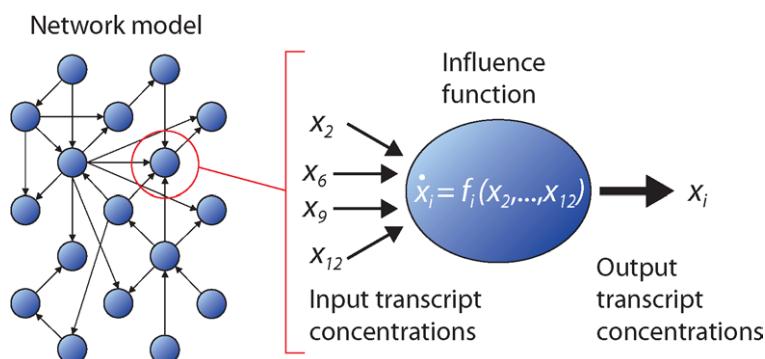


Fig. 4. Gene network models are represented as directed graphs describing the influence of the levels of one set of transcripts (the inputs) on the level of another transcript (the output). One usually assumes that networks are sparse, i.e., only a small subset of transcripts act as inputs to each transcript. The relation between inputs and outputs is specified by an interaction function (f_i).

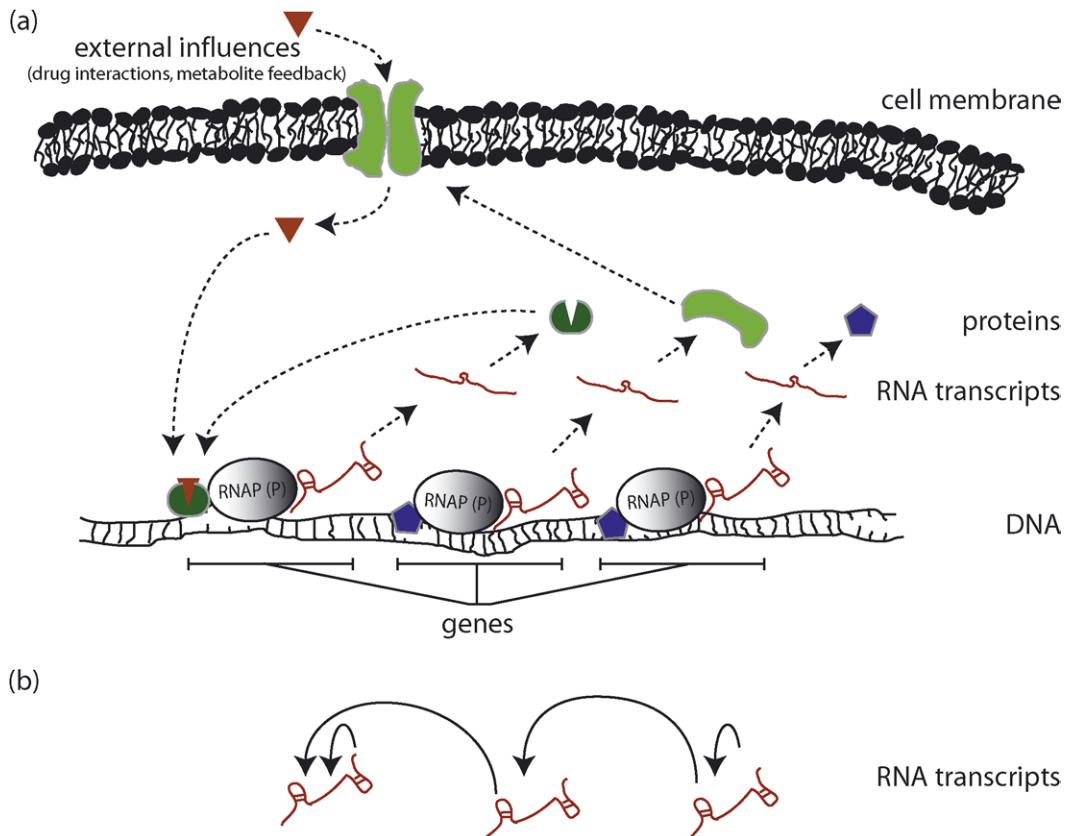


Fig. 5. Biological networks are regulated at many levels. (a) shows an example network where protein transcription factors (blue and green shapes) influence the expression of different transcripts (brown lines). One protein is a membrane-bound metabolite transporter. The metabolite it imports (brown triangle) binds one of the transcription factors enabling it to bind DNA and initiate transcription. (b) A gene network model of the real network in (a). Because the model is inferred from measurements of transcripts only, it describes transcripts as directly influencing the level of each other, even though they do not physically interact. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

stimuli. The limitation of this approach is that the model can be difficult to interpret in terms of the physical structure of the cell, and therefore difficult to integrate or extend with further research. Moreover, the implicit description of hidden regulatory factors may lead to prediction errors.

5. The physical strategy: identifying TF interactions

Here, we describe two additional physical approaches. The objective of both, like the Bussemaker approach, is to identify the binding motifs of transcription factors. One of the first methods to accomplish this task was introduced by Tavazoie and colleagues [10]. The method assumes that transcripts controlled by the same TFs will exhibit similar expression changes under a variety of experimental conditions. With a sufficient number of RNA expression experiments, the method clusters transcripts based on the similarity of their changes across all the experiments. Then the method applies a motif-finding algorithm

to identify homologous DNA sequences in the promoter regions of the clustered transcripts. The approach assumes that homologous DNA sequences are probable TF binding motifs.

The authors of the method applied it to a data set of 15 RNA expression profiles measured over two cell cycles of synchronized yeast cultures. Each profile measured the expression of more than 6000 yeast transcripts. The algorithm used a k -means algorithm with a Euclidean distance metric to cluster the expression profiles for 3000 of the transcripts that exhibited significant expression changes. The authors then applied the AlignACE [40] algorithm to identify putative TF binding motifs. The method found 18 motifs among 12 of the 30 clusters. Seven of the motifs corresponded to previously validated TF binding sites. The others were potentially novel motifs.

One limitation of this method is that transcripts sharing a motif may be regulated similarly under some conditions and differently under other conditions due to the presence of different combinations of motifs in their promoters. This may explain the failure of the approach to identify significant motifs in many of the clusters. One possible solution to this problem is to use condition-specific clustering [41] in which genes may be grouped together in one subset of conditions, but also may be grouped in other clusters under other conditions. Thus, genes may participate in more than one cluster. Each cluster is more likely to share common binding sites.

Another limitation of the Tavazoie method, and the Bussemaker method, is that they account only for the presence and copy number of motifs in the promoter region. In many promoters, a binding motif for a transcription factor may exist but have no regulatory function. Thus, location, orientation, methylation and other properties of a motif may also influence the regulatory effect of TFs on their target promoters. A method to partially account for such properties was recently introduced by Beer and Tavazoie [14]. This method may also overcome the clustering issues related to condition-specific regulation.

The Beer and Tavazoie begins the same way as the Tavazoie method; it clusters transcripts according to the similarity of their expression profiles across many experimental conditions, and then identifies the overrepresented motifs in each cluster using the AlignACE algorithm. The method then finds all transcripts that contain the identified motifs in their promoter regions, even if they are not in the original cluster. It determines the position, orientation, order, spacing and degree of homology of these motifs in each promoter. It then learns a Bayesian network model that describes the influence of these various properties on transcript expression.

In the approach, motif properties are encoded as binary variables, e.g., the motif exists or doesn't exist, the spacing is less than 50 bases or more than 50 bases, etc. These binary variables are then treated as parent nodes (i.e., regulator variables) in the Bayesian network model. The parent nodes that best predict the expression data are then learned through a iterative, greedy search procedure. In each iteration, each variable is tested and the most predictive variable is added to the model until the overall score of the model no longer improves. The model score is the probability that the model is correct for the observed data and is computed using Bayes theorem with equal prior probabilities for all models. More details on Bayesian network methods are provided below.

The approach successfully identified many of the known regulatory motifs in yeast as well as many new ones. More importantly, it succeeded in determining which transcripts were influenced by the motifs, and the regulatory logic underlying the motif properties. One result from their study (Fig. 6) shows the correlation of all transcripts containing two TF motifs in their promoters. Promoters with the motifs in specific locations show dramatically higher correlation of expression. Using five-fold cross-validation,

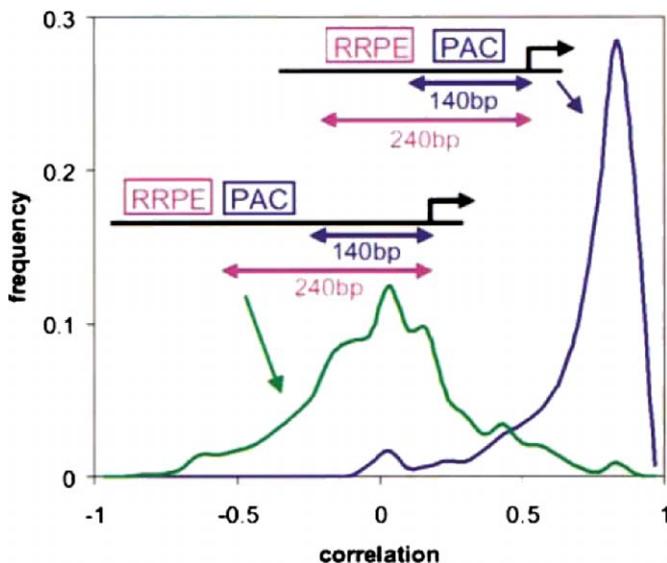


Fig. 6. Dependence of gene expression on motif position. Reproduced from Beer and Tavazoie [14]. The green curve shows a histogram of correlation coefficients for all pairs of transcripts that contain both the RRPE and PAC motifs in their promoter. The purple curve shows a histogram of correlation coefficients for just those transcripts that contain PAC in within the first 140 nucleotides and RRPE within the first 240 nucleotides of the promoter. Correlation of transcript expression shows a marked increase, which indicates that position has a significant influence on the regulatory effect of transcription factors. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

the authors showed that the regulatory logic identified by their algorithm could predict the expression pattern for 73% of the 2587 transcripts examined. In addition, it showed a marked improvement over the predictive capabilities of the Bussemaker algorithm.

6. The influence strategy: inferring gene networks

The influence strategy for reverse-engineering seeks a model in which transcripts serve as regulatory inputs that control the rate of synthesis of other RNAs. This type of model is sometimes called a *gene regulatory network* or a *gene network*. Gene network reverse-engineering algorithms do not use or model protein and metabolite data. Since the concentrations of many transcription factor RNAs do not correlate with the activity of their respective protein products [42,43], a gene network model does not usually describe physical relationships between regulators and transcripts [12,42,43]. Nevertheless, gene network models provide a global view of gene regulation that is not restricted to TF/promoter interactions. They can implicitly capture the protein and metabolite factors that may influence gene expression. Such models can provide valuable insight into the important regulators of cell responses. The models are also of value in predicting transcription responses to new cell treatments.

In general, one can represent a gene network model as a directed graph (Figs. 4 and 9a). Depending on the reverse-engineering approach used, one can describe this graph mathematically as a system of differential equations, as a Boolean network, as a Bayesian network, or as an association network [44]. The representations provide different degrees of simplification of cell regulation, lend themselves to

different computational strategies, and may suitable for different applications. We will discuss them in order.

6.1. Differential equation models

One can describe a gene network as a system of differential equations. The rate of change in concentration of a particular transcript, x_i , is given by a nonlinear influence function, f_i , of the concentrations of other RNA species.

$$\frac{dx_i}{dt} = f_i(x_1, \dots, x_N), \quad (7)$$

where N is the number of transcripts in the network. In general, only a small subset of all RNA species regulate a particular transcript. In other words, the networks are sparse. The objective of reverse engineering is to determine which RNA species regulate particular transcripts, and the parameters of the functions relating them.

An algorithm usually presupposes the form of the influence functions, f_i . Researchers have studied various functions, including sigmoidal functions [9] and linear functions [7,8,12]. Thus far, the linear functions have proven to be the most versatile in the analysis of experimental data sets [8,12]. In part, this is due the simplifying power of linear functions; they dramatically reduce the number of parameters needed to describe the influence function and avoid problems with overfitting. Thus, the amount of data required to solve a linear model is much less than that required by more complex nonlinear models. This advantage is crucial in light of the high cost of experimental data and the high dimensionality of the systems. However, this gain in experimental efficiency comes at the cost of placing strong constraints on the nature of regulatory interactions in the cell. The restrictions may lead to errors in the network model, and may restrict the conditions for which the model accurately describes the regulatory network.

Algorithms can infer the influence functions using time-series RNA expression data or steady state data. If the algorithm uses time-series data, it must estimate the rates of change of the transcripts (dx/dt) from the series. This can be problematic because calculating the derivative can amplify the measurement errors in the data. Alternately, one can convert the model to a discrete dynamical system [34,45], which does not require calculation of the derivatives. However, after inferring the model, one may need to transform it back to the continuous-time domain. On the other hand, if the algorithm uses steady-state data, $dx/dt = 0$, and the algorithm does not need to compute derivatives.

We illustrate the linear approach with an algorithm which we developed and applied successfully to a small network of transcripts in the bacterium *E. coli* [12]. In our method, called Network Identification by multiple Regression (NIR), the rate of change of a transcript is represented as a weighted sum of the levels of the other genes in the network:

$$\frac{dx_i}{dt} = \sum_j w_{ij}x_j + p_i, \quad (8)$$

where w_{ij} is the influence coefficient of transcript j on transcript i , and p_i is an externally applied perturbation to the level of transcript i . To learn the model, we collect training data by artificially synthesizing extra copies of individual transcripts to perturb the cells, and then measuring the steady-state levels of all transcripts in the network. Because steady-state data is used, $dx_i/dt = 0$, and $p_i = -\sum_j w_{ij}x_j$. This means that changes in the inputs to transcript i must balance the external perturbation. Therefore, the

measurement of the perturbation, p_i , provides an estimate of the steady-state rate of synthesis of transcript i from its promoter.

The algorithm learns the weights of the model using least-squares regression, where p_i serves as the dependent variable, and the RNA concentrations, x_j , as the regressors. Typically, the number of experimental data points is much less than the number of parameters in Eq. (8), and thus multiple solutions exist. To address this problem, we apply additional constraints to account for the sparse nature of transcription networks. The NIR algorithm forces each transcript to have a maximum of k regulators (k nonzero weights), where k is much less than the number of experiments. To find the k nonzero weights, the algorithm employs a greedy search scheme [45,46] in which the weights are added iteratively. In each iteration, it selects the weight providing the greatest reduction in squared error.

Alternately, one may account for the sparsity of the network by using statistical strategies that reduce the dimension of the data. These strategies include regression against clusters of transcripts [34] and principal component regression schemes [35,36]. We tested the NIR method on the SOS network in *E. coli*. This network regulates the cell's response to DNA damage and involves more than 100 genes. Because the SOS network was well described in the literature, it served as a good system for validating the NIR method. As a starting point, we applied the NIR method to a subset of nine transcripts at the core of the network. We used a synthetic DNA construct called a *plasmid* to synthesize extra copies of each transcript; we used the plasmid to perturb one transcript per experiment in nine separate experiments. We measured the resulting steady-state changes in the levels of all nine transcripts in each experiment. From this data, NIR algorithm calculated a network model that correctly identified 25 of the previously known regulatory relationships between the nine transcripts, as well as 14 relationships that may be novel regulation pathways, or possibly false positives (Fig. 7). These results were obtained with a noise-to-signal ratio of approximately 68%.

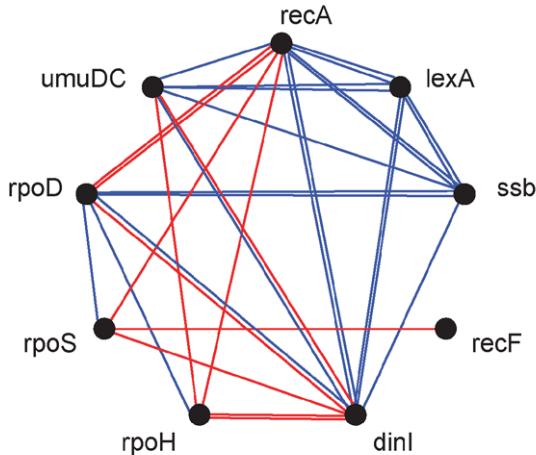
We subsequently used the network model as a predictive tool for analyzing new RNA expression data. A common problem in drug discovery is to identify the molecular targets of potential drug compounds. RNA expression data offers the possibility to find such targets by measuring all transcript responses to a drug. However, responses of the true targets of a drug are often masked by secondary changes in tens or hundreds of transcripts. For example, as illustrated in Fig. 8, when we applied the antibiotic mitomycin C to *E. coli*, we observed changes in all nine measured SOS transcripts. But the known mediator of mitomycin C is only the *recA* gene. The network model obtained by the NIR algorithm enables us to separate secondary changes from those that are due to the interaction with the drug. As illustrated in Fig. 8, we used the network model to filter the mitomycin C expression data and correctly identify the *recA* transcript as the target. The same target was identified for treatments with UV irradiation and the antibiotic pefloxacin, both of which stimulate the *recA* transcript, but not for novobiocin, a drug which is not known to stimulate *recA*.

6.2. Boolean network models

Boolean network models describe transcript levels as binary variables. A “1” represents that a transcript is expressed, or that it is changed relative to a reference state. A “0” represents that a transcript is not expressed or unchanged. The state of a transcript is determined as a Boolean function of the state of the input transcripts, i.e.,

$$x_i(t+1) = f_i^B(x_1(t), \dots, x_N(t)), \quad (9)$$

(a) Network inferred by NIR method



(b) Network model: connection weight matrix

	recA	lexA	ssb	recF	dinI	umuD	rpoD	rpoH	rpoS
recA	0.40	-0.18	-0.01	0	0.10	0	-0.01	0	0
lexA	0.39	-0.67	-0.01	0	0.09	-0.07	0	0	0
ssb	0.04	-1.19	-0.28	0	0.05	0	0.03	0	0
recF	0	0	0	0	0	0	0	0	0
dinI	0.28	0	0	0	-1.09	0.16	-0.04	0.01	0
umuD	0.11	-0.40	-0.02	0	0.20	-0.15	0	0	0
rpoD	-0.17	0	-0.02	0	0.03	0	-0.51	0.02	0
rpoH	0.10	0	0	0	0.01	-0.03	0	0.52	0
rpoS	0.22	0	0	-1.68	0.67	0	0.08	0	-2.92

Fig. 7. Inference of a nine-transcript subnetwork of the SOS pathway in *E. coli* using the NIR algorithm. (a) Graph depiction of the network model identified by the NIR algorithm. Previously known regulatory influences are marked in blue, novel influences (or false positives) are marked in red. The strengths and directions of the identified connections are not labeled in the graph. (b) The network model is also depicted as a matrix of interaction strengths. The colors are the same as in panel (a). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

where f_i^B is a Boolean function for transcript i . Boolean models were pioneered by Kauffman [47] to theoretically explore possible dynamic properties of gene networks. More recently they were studied as a model for reverse-engineering gene networks [4,6,48].

Researchers have explored two primary strategies to learn the connectivity of transcripts and the interaction functions in Boolean networks. The first strategy computes the mutual information between sets of two or more transcripts and tries to find the smallest set of input transcripts that provides complete information on the output transcript [4]. The second approach looks for the most parsimonious set of input transcripts whose expression changes are coordinated or consistent with the output transcript [6,48,49]. In both approaches, one may employ trimming strategies to eliminate redundant connections in the network.

Solving Boolean networks requires large amount of experimental data because it does not place constraints on the form of the Boolean interaction functions, f_i^B . Thus, to completely determine the

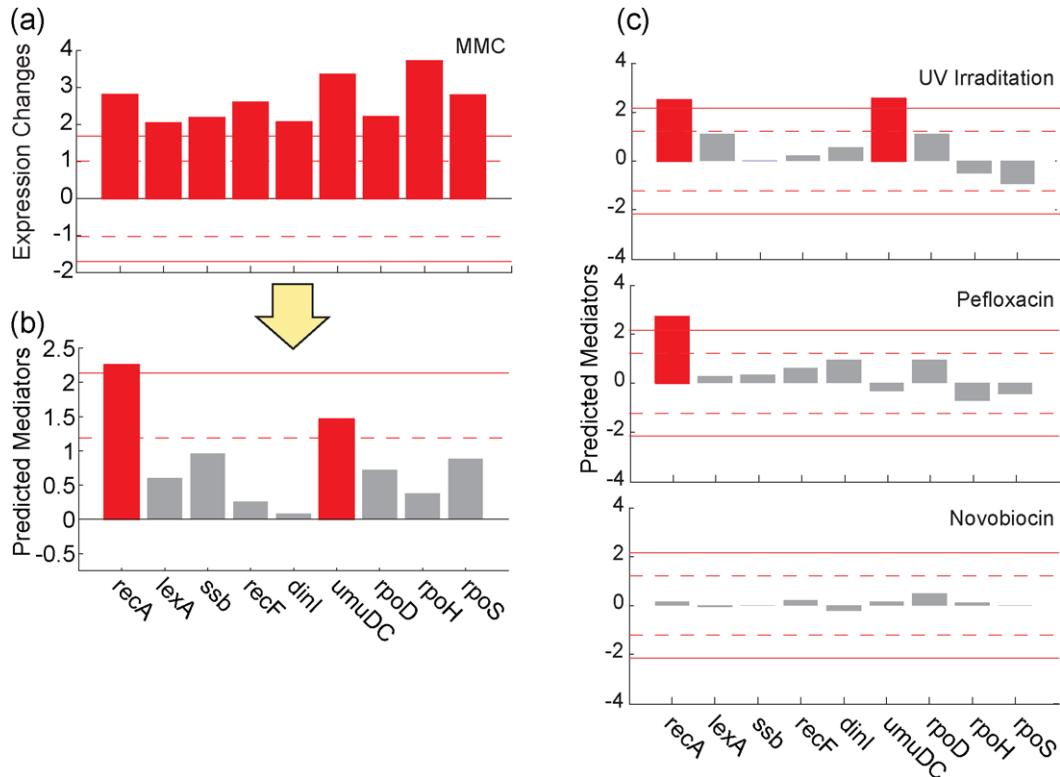


Fig. 8. The NIR-inferred network model is used to predict the gene targets in *E. coli* of different cell treatments. (a) Transcript expression changes following treatment with the antibiotic mitomycin C. Red indicates a statistically significant change. Lines denote significance levels: $P = 0.3$ (dashed), $P = 0.1$ (solid). (b) The NIR-inferred network model filters the expression data to show that *recA*, and possibly *umuDC* are the only targets of the drug. (c) The network model also predicts the targets of UV irradiation and two additional antibiotics. The expression data (not shown) were obtained from public microarray data sets. In the case of UV irradiation and pefloxacin treatment, both DNA-damaging, the *recA* gene is correctly predicted as the mediator of the expression response. For novobiocin, which does not damage DNA, *recA* is not predicted as the mediator of the expression response. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

interaction functions from data, an algorithm must have samples of all possible combinations of input expression states. (See Fig. 9.) Unlike linear or other more constrained functions, an algorithm cannot interpolate between states that are measured experimentally. Consequently, for a fully connected Boolean network, an algorithm would require approximately 2^N data points (all transcripts being measured in each data point) to infer all interaction functions, assuming no measurement errors [50]. Often, one assumes networks are sparsely connected, which reduces the demand for data. Nevertheless, the data requirements are still considerable.

Most of the current work on Boolean networks assumes that RNA transcript data has been already been discretized into binary variables. Because measurement error on transcript data is often quite large, discretization is not trivial. One must choose an appropriate statistical test and a confidence threshold to perform this discretization. Regardless of the approach used, false positive and false negative errors are inevitable, which will lead to degradation of performance of the algorithm. Although problems stemming

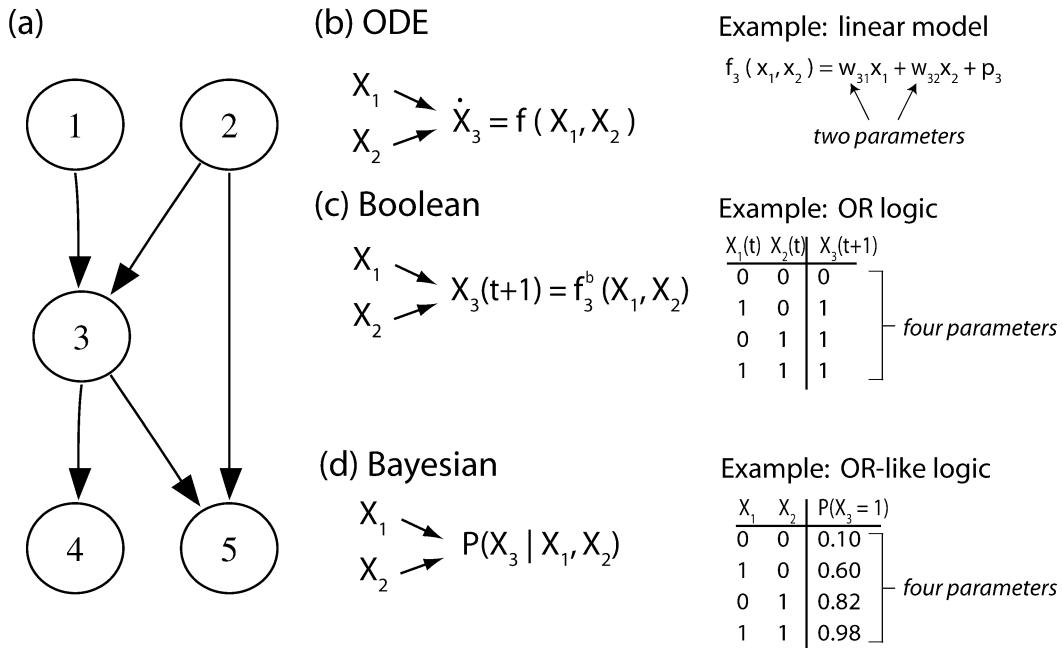


Fig. 9. (a) An example transcript control network with five transcripts and five connections. Different models may be used to represent the relationship between input and output transcripts, as illustrated for transcript 3 panels (b), (c), and (d). (b) Differential equation model of the input/output relationship. For a linear relationship, the model requires two parameters to relate transcripts 1 and 2 to transcript 3. (c) Boolean model of input/output relationship. The model requires four parameters to relate transcripts 1, 2 and 3. “OR” logic is illustrated. (d) Bayesian network model of the input/output relationship. The model is similar to the Boolean model; it requires four parameters to relate transcripts 1, 2 and 3. But the parameters specific probability distributions rather than deterministic relationships. “OR-like” logic is illustrated.

from noisy data are significant, they have received only limited direct attention [51] in studies exploring Boolean reverse-engineering approaches.

Most work on Boolean networks has examined only simulated data sets. Thus, it is difficult to assess their practical utility. Although Boolean networks offer much promise for describing the combinatorial complexity of transcriptional regulation [52], it seems that issues regarding noise and data requirements will impede their practical use in reverse-engineering.

6.3. Bayesian network models

Bayesian network models provide a way to account for the noise and data limitations inherent in expression studies, and also retain the combinatorial logic of transcription regulation. Bayesian networks represent the state of a transcript as a random variable, X_i . The random variable is specified by a probability distribution function, f_i , which is dependent on (i.e., *conditioned on*) a set of regulator transcripts, X_j . We can write this as:

$$\text{Prob}(X_i = x_i | X_j = x_j) = f_i(x_i | x_j), \quad (10)$$

where $j = 1, \dots, N$, and $j \neq i$, and a lowercase x denotes a particular value of X . For simplicity, we can write the left hand side of the equation as $P(X_i | X_j)$. This conditional distribution has one further

restriction, a transcript may be a regulator of any other transcript, provided that the network contains no *cycles* (i.e., no feedback loops). This restriction is the principal limitation of Bayesian network models. It stems from the joint probability distribution function for all transcripts, f , from which we derived Eq. (10). For example, the state of transcripts in the network in Fig. 9 is given by the joint distribution:

$$\text{Prob}(X_1 = x_1, \dots, X_5 = x_5) = f(x_1, \dots, x_5).$$

Again, one can write the left hand side more simply as $P(X_1, \dots, X_5)$. Applying the chain rule, we can rewrite the joint probability distribution as the product of conditional probabilities:

$$P(X_1, \dots, X_5) = P(X_5|X_4, \dots, X_1)P(X_4|X_3, X_2, X_1)P(X_3|X_2, X_1)P(X_2|X_1)P(X_1).$$

Thus, we obtain the form of the conditional distributions given by Eq. (10). This factoring shows that the joint distribution cannot be satisfied by a network topology that contains cycles.

In the terminology of Bayesian networks, the regulators of a transcripts are called its *parents*. If we assume that the probability of each transcript depends only on the state of its *parents*, then we can further simplify the joint probability to:

$$P(X_1, \dots, X_5) = P(X_5|X_3, X_2)P(X_4|X_3)P(X_3|X_2, X_1)P(X_2)P(X_1).$$

Only the parent transcripts appear as regulators in the conditional distribution functions. Thus, to reverse-engineer a Bayesian network model of a transcription network, we must find two sets of parameters: the model topology (i.e., the regulators of each transcript), and the conditional probability functions relating the state of the regulators to the state of the transcripts. If the network is sparsely connected, each transcript will have few regulators, which helps to minimize the number of parameters in the model.

As is the case for differential equation models, the model learning algorithm usually presupposes the form of the conditional probability function, f_i . Any function may be used, including Boolean and linear functions. But there will be a trade-off between model realism and model simplicity. More realistic models will have more parameters, which will require more experimental data and greater computational effort to solve.

As an example, consider the network in Fig. 9. If we discretize transcript levels into binary random variables, the algorithm must learn four parameters to specify $P(X_3|X_2, X_1)$, the conditional distribution for transcript 3. One parameter exists for each of the 2^k combinations of the states of the k parents. Thus, to fully specify the distribution function for a transcript, one must experimentally observe each of the 2^k states at least once (as is the case for Boolean networks). Because this is an impractically large number of experiments, training data sets for such models are often incomplete; i.e., they do not measure all possible states. Fortunately, the Bayesian network structure enables an algorithm to partially specify the conditional distribution function for at least those states that are observed. A set of partially complete distribution functions can be sufficient to determine the topology of the network.

The network structure is usually determined using a heuristic search, such as a greedy-hill climbing approach or Markov Chain Monte Carlo [33] method. For each network structure visited in the search, an algorithm learns the maximum likelihood parameters for the conditional distribution functions. It then computes a score for each network using the Bayes Information Criteria [33] or some other metric that measures the overall fit of the model to the data. One may then select the highest-scoring network as the correct network.

Often, because training data are incomplete, the learning problem is underdetermined and several high-scoring networks are found. To address this problem, one can use model averaging or bootstrapping

to select the most probable regulatory interactions and to obtain confidence estimates for the interactions. For example, if a particular interaction between two transcripts repeatedly occurs in high-scoring models, one gains confidence that this edge is a true biological dependency. Alternatively, one can augment an incomplete data set with prior information to help select the most likely model structure. The probabilistic structure of a Bayesian network enables straight-forward incorporation of prior information via application of Bayes rule [33].

Typically algorithms use steady-state transcription measurements for the training data. However, such data do not permit the modeling of network dynamics. To capture dynamics behaviors, one may use Dynamic Bayesian networks [53,54] along with time-series data. These models also offer the possibility of capturing feedback loops. However, Dynamic Bayesian networks may moderately increase the complexity and data demands of the model. They are also subject to the challenges of using time-series data, which are discussed below.

Bayesian models are well-suited to dealing with incomplete data sets, and allow for the incorporation of prior data about the structure of a regulatory network. Consequently, researchers have devoted considerable attention in recent years to the use of Bayesian network approaches for reverse-engineering gene networks [13,37,54–56]. In our own work, we have successfully applied a Bayesian network model to reverse-engineer transcription network interactions in *E. coli* (Fig. 10).

6.4. Association networks

Association networks assign interactions to pairs of transcripts that exhibit high statistical similarity (i.e., statistical dependence) in their responses in all experiments in a training data set. To measure similarity, algorithms often use Pearson correlation, which assumes linear dependence between variables, or mutual information, which makes no assumptions about the form of dependence between variables. Because association networks measure only similarity, they cannot assign direction to the connections between transcripts.

This class of algorithms requires a training data set of transcript expression levels measured over many different experimental conditions. Algorithms begin by adding connections between all transcript pairs with expression profiles that exceed a threshold of similarity. Ideally, connections in this graph will describe true input–output (parent–child relationships). However, many connections in this initial graph may associate transcripts that are regulated by a common parent transcript, or transcripts that are a few nodes upstream in the network. In other words, the first step of the algorithm does not distinguish similar and causal relations, nor between direct and indirect relations. To address this problem, a pruning process is undertaken to remove connections that are better explained by a more direct path through the graph. What remains are the connections that are more likely to be causal interactions.

As an example, in Fig. 9a, the first step of the algorithm will likely find high correlation between transcript 1 and 5, even though there is no direct interaction between them, because some of the influence of transcript 1 on 3 is transferred to transcript 5. By considering higher-order dependences, pruning algorithms try to remove this type of connection. The connection between transcript 1 and 5 is removed because the similarity is lowest among the three pairs of transcripts 1, 3 and 5. Likewise, a false connection between transcripts 3 and 5 would be removed among transcripts 3, 4 and 5. Clearly this method is also susceptible to errors. For example, the relation between transcripts 4 and 5 might be removed.

One recent association network algorithm uses partial correlation [15]. The algorithm first uses Pearson correlation or Spearman rank correlation to connect all similar transcripts. To remove redundant

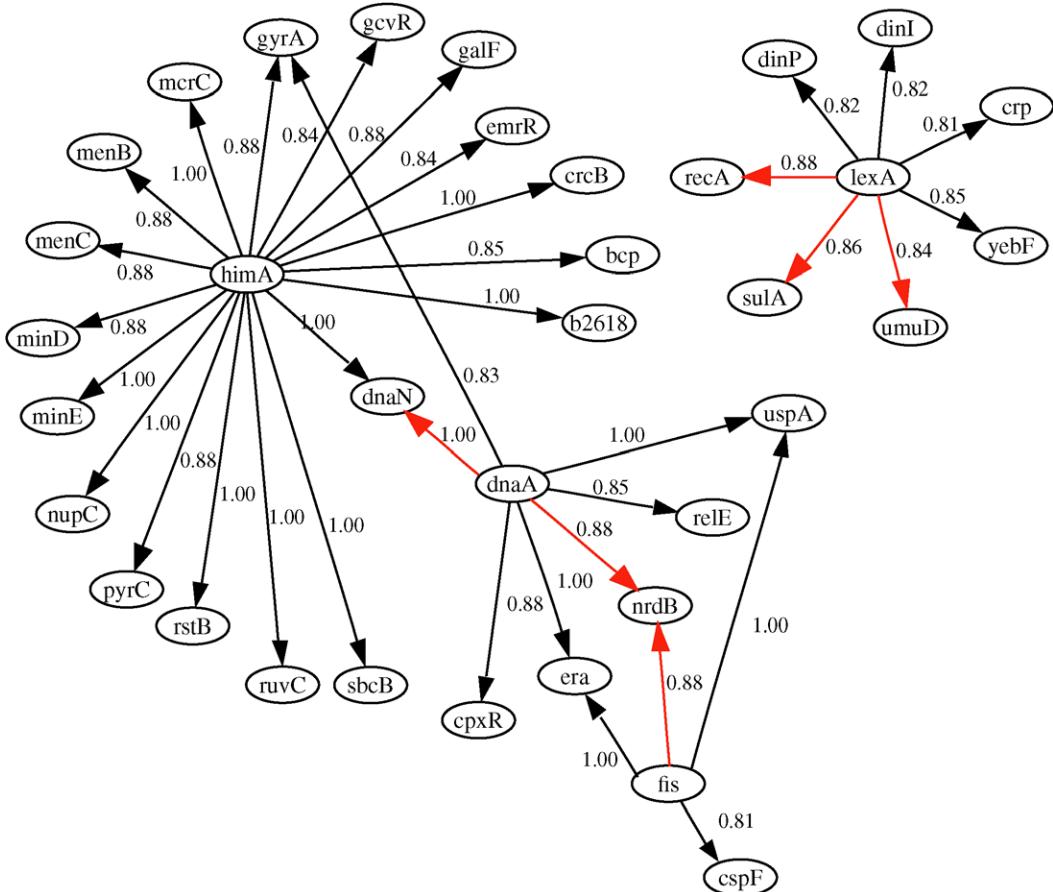


Fig. 10. A Bayesian network of the *E. coli* SOS pathway, which we inferred from a set of 56 microarrays measuring cell response to single-transcript perturbations. We included only the 56 perturbed transcripts in the inferred network and discretized expression measurements into binary values. We used a Markov Chain Monte Carlo (MCMC) search with a Bayesian Information Criteria (BIC) score to find the best set of transcript dependencies. We determined confidence values (the numbers on the arrows) for connections as the frequency at which a connection occurred in samples drawn from the MCMC search. Previously known connections (as determined from regulon DB [69]) are shown in red. Only edges with a frequency above 0.8 were retained in the final network. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

connections, the algorithm then prunes the network using partial correlation coefficients. Partial correlation measures the correlation between two variables after subtracting the correlation between each variable and one or more additional variables. In other words, partial correlation measures the correlation between variables, *conditioned on* the additional variables. The number of conditioning variables defines the *order* of the partial correlation; e.g., a first-order partial correlation is conditioned on one variable.

The pruning step calculates the first and second-order partial correlation between two transcripts given every possible combination of the conditioning transcripts. It removes the connection between the two transcripts if their partial correlation falls below some significance threshold for any combination of conditioning transcripts. A low partial correlation implies that the dependency between the two transcripts is better explained by the conditioning transcripts.

For computational efficiency, the algorithm considers only up to second-order partial correlation, but the authors use simulations to show that improvements for higher-orders are minor. The power of this approach is not high (only 20–40% of interactions were inferred in simulations), but it produces a collection of small sparse subnetworks with high-confidence.

A second algorithm, called ARACNE, uses mutual information to associate transcripts [57,58]. It was written to address the difficult problem of network reconstruction in mammalian systems where the number of transcripts is higher and the network topology is more complex. The algorithm creates an initial graph by connecting all transcript pairs with a mutual information value above a *p*-value derived threshold (the mutual information, $I(X_i, X_j)$, between two variables X_i and X_j is zero when they are independent). The algorithm then prunes indirect interactions by considering all triplets of transcripts. A triplet is a fully connected set of three transcripts. The algorithm removes the connection with the lowest mutual information (least dependence). This process results in locally tree-like networks, and would not be able to infer subgraphs like that of X_3 , X_4 , and X_5 in Fig. 9. Alternatively, one can specify a tolerance threshold to prevent pruning of more significant 3-transcript loops.

A third algorithm utilizes the dynamic information encoded in time-series data [59,60]. It calculates the time-delayed cross-correlation coefficient between pairs of transcripts. The algorithm retains highly correlated transcript pairs, and prunes the connections according the length of delay at which the maximum correlation occurs. The method assumes that direct connections will exhibit maximum correlation at shorter delays. Thus, the algorithm eliminates redundant connections better explained by connections with shorter delays.

7. Challenges

A challenge common to all reverse-engineering techniques is noisy, limited data. Without simplifying assumptions, the models used to represent transcriptional control networks are underdetermined. As a first step, one chooses a model structure that places constraints on the form of interaction functions, thereby limiting the number of model parameters. Linear models and discretization of variables are two examples of such simplifications. One can obtain further reduction in parameters by assuming sparse net structure. But this comes at the cost of computational complexity; a heuristic or Monte Carlo search for the best combination of regulators of each transcript is required. The sparsity assumption may be addressed also using statistical techniques such as principal components analysis or clustering to group similarly-regulated genes. Regardless of the simplification strategy employed, the resulting models may provide an incomplete or coarse-grained portrait of the underlying transcriptional network.

A potentially more promising way to deal with limited data is to use prior information collected with distinct experimental techniques. In several examples of this strategy [61–64], researchers assumed that the complete topology of a TF/promoter interactions was known a priori. Algorithms were applied to determine the strength of interactions and the activation state of the TF proteins in particular experimental conditions. The resulting models provided a quantitative basis for the timing and strength of the observed transcription responses, and helped to connect transcription regulation with protein and metabolite regulators.

Unfortunately, information on the topology of transcription networks is often spotty. Thus, algorithms that can appropriately utilize partial prior information are of great value. To date, such most algorithms have used the *intersection* of data collected through distinct experimental modes, e.g., sequence, annota-

tion, expression, and chromosome-binding data. For example, when analyzing a yeast expression data set, Segal, et al. [13] restricted the set of possible regulators to those that had been annotated with regulatory roles. However, care must be taken to avoid excessive bias in the results. For example, if annotations are incomplete or incorrect, an algorithm may not identify correct regulators. Other algorithms that combine data types have been explored [11,37,65–67], including the physical algorithms discussed above. Such approaches remain a promising area of research.

Decreases in the cost of experimental technology may also help to alleviate problems concerning the quantity and quality of data. One will be able obtain more replicates to improve measurement errors, and measure more experimental conditions to observe different transcriptional states of a cell. Even with such improvements, experimental data will remain finite, and questions concerning optimal experimental design will remain. That is, how can one get the biggest bang for their buck? A significant question is whether to use a time-series design or a steady-state design. The steady-state design may miss dynamic events that are critical for correctly inferring the control structure of a transcription network, but it enables one to observe more diverse experimental conditions. On the other hand, time-series experiments can capture dynamics, but many of the data points may contain redundant information leading to inefficient use of experimental resources. In our own work, we have found the steady-state design to be more informative for inferring topology [12], and simpler to implement experimentally, but both approaches have been explored [10,59].

Another key challenge for reverse-engineering transcription networks is validation of the methods. The number and complexity of algorithms is expanding rapidly, but no generally accepted benchmark criteria exist for evaluating their performance. In part, this is due to the differing objectives of the algorithms, and limited availability of standardized data sets (experimental or computational). Thus, it is currently almost impossible to rationally judge the relative merits of the algorithms. Moreover, most algorithms have limited or no experimental validation. Hence, their practical utility often remains unknown. As the field of biochemical reverse-engineering continues to develop, it is important that algorithms be tested against standardized data sets [68] using benchmark performance criteria. Only with such testing, and thorough application of algorithms to experimental data sets, will it become clear what are the strengths, weaknesses, and practical utility of various approaches.

8. Summary

The number of approaches for reverse-engineering transcription networks is large and expanding. In this review, we have classified the approaches into two general categories: physical and influence. The influence approach produces a *gene network* model that will not usually have meaning in terms of direct physical interactions between cellular molecules, and thus, may be difficult to interpret or extend through additional experimental studies. But it can capture system-wide regulatory relationships between genes that provide practical value. For example, in our own work, we have used gene network models to predict the targets of drugs in *E. coli* [12] and yeast [70].

The physical approach, on the other hand, does not necessarily provide a system-wide picture of regulatory influences in a cell, but it does provide estimates of the physical topology of regulatory interactions. Thus, such an approach provides a more permanent and extendable model of transcription regulation. This approach appears particularly promising for decoding mechanisms of transcription factor regula-

tion. It may soon be possible to map the transcriptome with the same certainty, speed and efficiency as is now possible for genome sequencing.

Although the objectives and applications of the two strategies are different, they share many of the same computational principles and challenges. For example, linear, Bayesian, and association approaches have been applied in both strategies. And both strategies encounter similar challenges with respect to data limitations and computational complexity. As these approaches are increasingly applied to benchmark simulated and experimental data sets, it will become clear what are their relative merits and practical utility. We hope that a few particularly effective approaches will emerge as standards for systematic application to many organisms.

Acknowledgements

This work was supported the Department of Energy, the Pharmaceutical Research and Manufacturers of America Foundation, the Army Research Office, and the Whitaker Foundation. We also thank Diego di Bernardo, David Byrne, and Michael Thompson for their comments on the manuscript.

References

- [1] Schena M, Shalon D, Davis RW, Brown PO. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 1995;270:467.
- [2] Chee M, et al. Accessing genetic information with high-density DNA Arrays. *Science* 1996;274:610.
- [3] Lockhart DJ, et al. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol* 1996;14:1675.
- [4] Liang S, Fuhrman S, Somogyi R. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Proc Pacific Symp Biocomp* 1998;3:18.
- [5] Szallasi Z. Genetic network analysis in light of massively parallel biological data acquisition. In: *Pacific symp. biocomput.*; 1999. p. 5–16.
- [6] Akutsu T, Miyano S, Kuhara S. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In: *Pacific symp. biocomput.*; 1999. p. 17–28.
- [7] Chen T, He HL, Church GM. Modeling gene expression with differential equations. In: *Pacific symp. biocomput.*; 1999. p. 29–40.
- [8] D'Haeseleer P, Wen X, Fuhrman S, Somogyi R. Linear modeling of mRNA expression levels during CNS development and injury. *Proc Pacific Symp Biocomp* 1999;4:41.
- [9] Weaver DC, Workman CT, Stormo GD. Modeling regulatory networks with weight matrices. *Proc Pacific Symp Biocomp* 1999;4:112.
- [10] Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. *Nat Genet* 1999;22:281.
- [11] Bussemaker HJ, Li H, Siggia ED. Regulatory element detection using correlation with expression. *Nat Genet* 2001;27:167.
- [12] Gardner TS, di Bernardo D, Lorenz D, Collins JJ. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 2003;301:102.
- [13] Segal E, et al. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet* 2003;34:166.
- [14] Beer MA, Tavazoie S. Predicting gene expression from sequence. *Cell* 2004;117:185.
- [15] de la Fuente A, Bing N, Hoeschele I, Mendes P. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics* 2004;20:3565.
- [16] Lodish H, Berk A, Matsudaira P. Molecular cell biology. 4 ed. New York: Freeman; 2000.

- [17] Munoz-Gomez AJ, Santos-Sierra S, Berzal-Herranz A, Lemonnier M, Diaz-Orejas R. Insights into the specificity of RNA cleavage by the *Escherichia coli* MazF toxin. FEBS Lett 2004;567:316.
- [18] Zhang Y, et al. MazF cleaves cellular mRNAs specifically at ACA to block protein synthesis in *Escherichia coli*. Mol Cell 2003;12:913.
- [19] Christensen SK, Pedersen K, Hansen FG, Gerdes K. Toxin–antitoxin loci as stress-response-elements: ChpAK/MazF and ChpBK cleave translated RNAs and are counteracted by tmRNA. J Mol Biol 2003;332:809.
- [20] Cashel M, Gentry DR, Hernandez VJ, Vinella D. In: *Escherichia coli and Salmonella: cellular and molecular biology*. 2 ed. Washington, DC: American Society for Microbiology; 1996 [chapter 92].
- [21] Ptashne M. A genetic switch: phage λ and higher organisms. 2 ed. Cambridge, MA: Cell Press; 1992.
- [22] Barkai N, Leibler S. Robustness in simple biochemical networks. Nature 1997;387:913.
- [23] Leloup JC, Goldbeter A. A model for circadian rhythms in *Drosophila* incorporating the formation of a complex between the PER and TIM proteins. J Biol Rhythms 1998;13:70.
- [24] Bhalla US, Iyengar R. Emergent properties of networks of biological signaling pathways. Science 1999;283:381.
- [25] Gardner TS, Cantor CR, Collins JJ. Construction of a genetic toggle switch in *Escherichia coli*. Nature 2000;403:339.
- [26] Elowitz MB, Leibler S. A synthetic oscillatory network of transcriptional regulators. Nature 2000;403:335.
- [27] Becskei A, Séraphin B, Serrano L. Positive feedback in eukaryotic gene networks: cell differentiation by graded to binary response conversion. EMBO J 2001;20:2528.
- [28] Bhalla US, Ram PT, Iyengar R. MAP kinase phosphatase as a locus of flexibility in a mitogen-activated protein kinase signaling network. Science 2002;297:1018.
- [29] Vilar JMG, Kueh HY, Barkai N, Leibler S. Mechanisms of noise-resistance in genetic oscillators. Proc Natl Acad Sci USA 2002;99:5988.
- [30] Isaacs F, Hasty J, Cantor CR, Collins JJ. Prediction and measurement of an autoregulatory genetic module. Proc Natl Acad Sci USA 2003;100:7714.
- [31] Atkinson MR, Savageau MA, Myers JT, Ninfa AJ. Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. Cell 2003;113:597.
- [32] Tadesse MG, Vannucci M, Lio P. Identification of DNA regulatory motifs using Bayesian variable selection. Bioinformatics 2004;20:2553.
- [33] Neapolitan RE. Learning Bayesian networks (artificial intelligence). New York: Prentice-Hall; 2004.
- [34] van Someren EP, Wessels LF, Reinders MJ. Linear modeling of genetic networks from experimental data. Proc Int Conf Intell Syst Mol Biol 2000;8:355.
- [35] Yeung MKS, Tegner J, Collins JJ. Reverse engineering gene networks using singular value decomposition and robust regression. Proc Natl Acad Sci USA 2002;99:6163.
- [36] Holter NS, Maritan A, Cieplak M, Fedoroff NV, Banavar JR. Dynamic modeling of gene expression data. Proc Natl Acad Sci USA 2001;98:1693.
- [37] Hartemink AJ, Gifford DK, Jaakkola TS, Young RA. Combining location and expression data for principled discovery of genetic regulatory network. Proc Pacific Symp Biocomp 2002;7:437.
- [38] Wang W, Cherry JM, Botstein D, Li H. A systematic approach to reconstructing transcription networks in *Saccharomyces cerevisiae*. Proc Natl Acad Sci USA 2002;99:16893.
- [39] Yeang C-H, Ideker T, Jaakkola T. Physical network models. J Comput Biol 2004;11:243.
- [40] Hughes JD, Estep PW, Tavazoie S, Church GM. Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. J Mol Biol 2000;296:1205.
- [41] Ihmels J, et al. Revealing modular organization in the yeast transcriptional network. Nat Genet 2002;31:370.
- [42] Herrgård MJ III, Covert MW, Palsson BO. Reconciling gene expression data with known genome-scale regulatory network structures. Genome Res 2003;13:2423.
- [43] Gutiérrez-Ríos RM, et al. Regulatory network of *Escherichia coli*: consistency between literature knowledge and microarray profiles. Genome Res 2003;13:2435. Evaluation studies.
- [44] de Jong H. Modeling and simulation of genetic regulatory systems: a literature review. J Comp Biol 2002;9:67.
- [45] van Someren EP, Wessels LFA, Reinders MJ, Backer E. Searching for limited connectivity in genetic network models. In: Proc. 2nd int. conf. on systems biol.; 2001. p. 222–30.
- [46] Di Bernardo D, Gardner TS, Collins JJ. Robust identification of large genetic networks. In: Pacific Symp. Biocomput.; 2004. p. 486–97.
- [47] Kauffman SA. Origins of order: self-organization and selection in evolution. Oxford: Oxford Univ. Press; 1989.

- [48] Ideker TE, Thorsson V, Karp RM. Discovery of regulatory interactions through perturbation: inference and experimental design. In: Pacific Symp. Biocomput.; 2000. p. 305–16.
- [49] Kyoda KM, Morohashi M, Onami S, Kitano H. A gene network inference method from continuous-value gene expression data of wild-type and mutants. Genome Inform Ser Workshop Genome Inform 2000;11:196.
- [50] D'haeseleer P, Liang S, Somogyi R. Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics 2000;16:707.
- [51] Akutsu T, Miyano S, Kuhara S. Inferring qualitative relations in genetic networks and metabolic pathways. Bioinformatics 2000;16:727.
- [52] Yuh C-H, Bolouri H, Davidson EH. Genomic *cis*-regulatory logic: experimental and computational analysis of a sea urchin gene. Science 1998;279:1896.
- [53] van Berlo RJP, van Someren EP, Reinders MJT. Studying the conditions for learning dynamic Bayesian networks to discover genetic regulatory networks. Simulation 2003;79:689.
- [54] Nachman I, Regev A, Friedman N. Inferring quantitative models of regulatory networks from expression data. Bioinformatics 2004;1(20 Suppl):I248.
- [55] Segal E, Taskar B, Gasch A, Friedman N, Koller D. Rich probabilistic models for gene expression. Bioinformatics 2001;1(17 Suppl):243.
- [56] Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. J Comput Biol 2000;7:601.
- [57] Margolin AA, et al. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context; 2004. Preprint, <http://arxiv.org/abs/q-bio/0410037>.
- [58] Margolin AA, Banerjee N, Nemenman I, Califano A. Reverse engineering of the yeast transcriptional network using the ARACNE algorithm; 2004. <http://www.menem.com/~ilya>.
- [59] Schmitt WAJ, Raab RM, Stephanopoulos G. Elucidation of gene interaction networks through time-lagged correlation analysis of transcriptional data. Genome Res 2004;14:1654.
- [60] Arkin A, Shen P, Ross J. A test case of correlation metric construction of a reaction pathway from measurements. Science 1997;277:1275.
- [61] Liao JC, et al. Network component analysis: reconstruction of regulatory signals in biological systems. Proc Natl Acad Sci USA 2003;100:15522.
- [62] Kao KC, et al. Transcriptome-based determination of multiple transcription regulator activities in *Escherichia coli* by using network component analysis. Proc Natl Acad Sci USA 2004;101:641.
- [63] Ronen M, Rosenberg R, Shraiman BI, Alon U. Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. Proc Natl Acad Sci USA 2002;99:10555.
- [64] Kalir S, Alon U. Using a quantitative blueprint to reprogram the dynamics of the flagella gene network. Cell 2004;117:713.
- [65] Tamada Y, et al. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. Bioinformatics 2003;2(19 Suppl):II227.
- [66] Bar-Joseph Z, et al. Computational discovery of gene modules and regulatory networks. Nat Biotechnol 2003;21:1337.
- [67] Segal E, Wang H, Koller D. Discovering molecular pathways from protein interaction and gene expression data. Bioinformatics 2003;1(19 Suppl):264. Evaluation studies.
- [68] Mendes P, Sha W, Ye K. Artificial gene networks for objective comparison of analysis algorithm. Bioinformatics 2003;2(19 Suppl):II122.
- [69] Salgado H, et al. RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. Nucl Acids Res 2004;32:D303. Database issue.
- [70] di Bernardo, et al. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. Nat Biotechnol 2005; in press.