BRIEFINGS IN BIOINFORMATICS. VOL 10. NO 4. 408–423 doi:10.1093/bib/bbp028

Computational methods for discovering gene networks from expression data

Wei-Po Lee and Wen-Shyong Tzou

Submitted: 14th March 2009; Received (in revised form): 8th May 2009

Abstract

Designing and conducting experiments are routine practices for modern biologists. The real challenge, especially in the post-genome era, usually comes not from acquiring data, but from subsequent activities such as data processing, analysis, knowledge generation and gaining insight into the research question of interest. The approach of inferring gene regulatory networks (GRNs) has been flourishing for many years, and new methods from mathematics, information science, engineering and social sciences have been applied. We review different kinds of computational methods biologists use to infer networks of varying levels of accuracy and complexity. The primary concern of biologists is how to translate the inferred network into hypotheses that can be tested with real-life experiments. Taking the biologists' viewpoint, we scrutinized several methods for predicting GRNs in mammalian cells, and more importantly show how the power of different knowledge databases of different types can be used to identify modules and subnetworks, thereby reducing complexity and facilitating the generation of testable hypotheses.

Keywords: *gene expression profiling; gene regulatory network; reverse engineering; transcription factor binding site; protein–protein interaction*

INTRODUCTION

In the post-genome era, biological researchers face the significant challenge of utilizing information churned out by multiple 'omics' technologies (genomics, transcriptomics, proteomics, glycomics, metabolomics). In general, biologists are aware of the need to move beyond the one gene or one protein approach and take a holistic view during all phases of research including data collection, information processing, interpretation, knowledge acquisition, domain discovery, hypothesis generation and subsequent experimental design. Systems biology is therefore more than just an emerging field: it represents a new way of thinking about biology with dramatic impact on the way that research is performed.

Under the command of transcription factors (TFs), each gene influences the activity of the cell by generating messenger RNA (mRNA) that guides the synthesis of proteins by ribosomes in the cytoplasm, which is the location in the cell where biochemical reactions and molecular events take place. Some of the proteins generated are themselves TFs that return to the nucleus (in eukaryotes) to control the expression of one or several genes. This complicated means of controlling gene expression can be represented as a gene regulatory network (GRN). The need to infer GRNs using available information derives from biologists' need to describe the complex phenomena of nature. This review focuses on gene regulation, in particular the regulation of

Corresponding authors. Wei-Po Lee, Department of Information Management, National Sun Yat-sen University, Kaohsiung, Taiwan. E-mail: wplee@mail.nsysu.edu.tw

Wen-Shyong Tzou, Institute of Bioscience and Biotechnology, National Taiwan Ocean University, Keelung, Taiwan. E-mail: wstzou@ntou.edu.tw

Wei-Po Lee is an associate professor at the Department of Information Management, National Sun Yat-sen University, Kaohsiung, Taiwan. His research interests include biological network simulation and modeling, system analysis and identification, and machine learning.

Wen-Shyong Tzou is an associate professor at the Institute of Bioscience and Biotechnology, National Taiwan Ocean University, Keelung, Taiwan. His research interests include the development of tool for the enriched motifs in the upstream sequences of mammalian genes from microarray platform and the elucidation of gene regulatory mechanism during embryonic development of zebrafish.

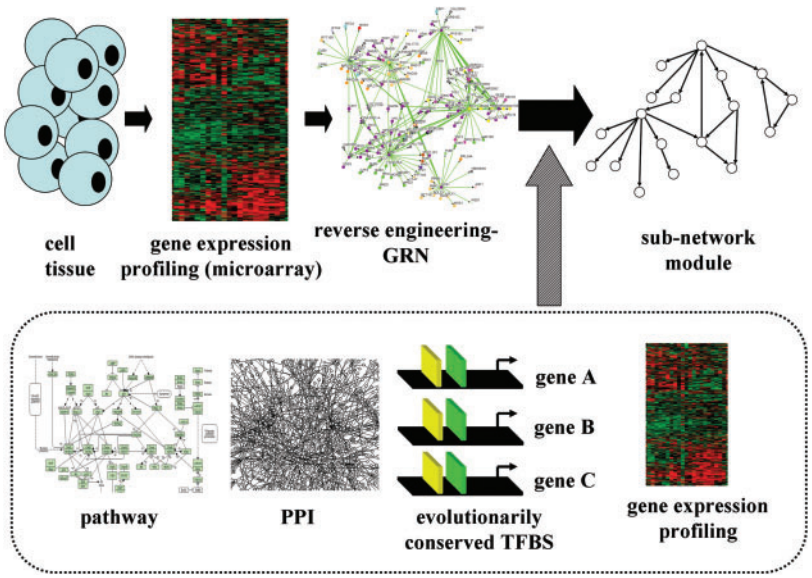

Figure 1: Inferring GRNs from gene expression profiling data. mRNA is extracted from cells or tissues, and subsequent microarray experiments are conducted. The output of the microarray experiments is gene expression profiling data, which computational methods then utilize to infer the GRN. Sometimes biologists need to consult databases of a different nature (such as pathway, PPI, evolutionarily conserved TFBS or gene expression profiling) to narrow down the search space. A subnetwork or module is expected to emerge that will serve as a final working model for biologists, allowing them to form hypotheses and design experiments.

transcription, and the ways that transcriptome data (primarily gene expression profiling data from microarray experiments) can be used to unravel the complex relationships between genes and proteins that comprise a GRN (Figure 1).

Gene expression profiling data from yeast (*Saccharomyces cerevisiae*) has become a platform for inferring gene networks. For example, Segal *et al.* [1] employed a probabilistic model that used an *S. cerevisiae* gene expression data set consisting of 2355 genes from 173 microarray experiments to infer condition-specific GRNs. This analysis predicted functions for several previously unannotated proteins. Recently, to approach a similar problem, a new method was proposed that uses centroid-like solutions extracted from an ensemble of possible statistical models to infer and interpret the network [2]. Furthermore, genome-wide identification of TF binding sites in *S. cerevisiae* has given researchers another dimension of information useful for the construction of GRNs [3–5]. This type of integration of data from different sources will lead to the development of GRNs that are both more accurate and more relevant. With the aid of ever-increasing computer power and rapidly advancing computational methods, the gene networks of mammalian cells (such as those from humans or mice) are now being described [6–10].

Gene network modeling uses gene expression profiling data to describe the phenotypic behavior of a system under study. Traditionally, to reconstruct a GRN from experimental data, an initial model is built that simulates the system's behavior in a certain experimental or environmental condition. After presenting the model with novel conditions, its predictions are compared with the observed gene expression data to give an indication of the adequacy of the model. If the experimental data are considered reliable and the predicted system behavior does not match the data, the model must be revised. The routine of manually constructing a model of the regulatory network, simulating the behavior of the system and testing the resulting predictions is repeated until an adequate model is obtained.

As the above procedure for network modeling takes a considerable amount of time, an automated procedure is desirable. Reverse engineering is a paradigm with great promise for analyzing and constructing biological networks [11–13]; it is an effective way of utilizing experimental data to determine the underlying network of a given model.

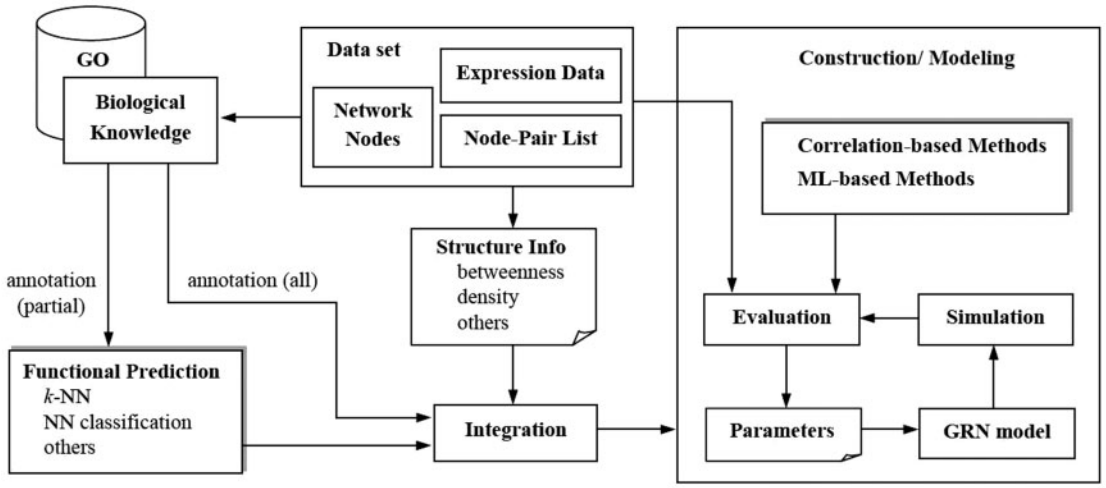

Figure 2: The general framework of a reverse engineering approach for modeling GRNs from measured expression data. The right side of this figure depicts the procedure of using computational methods to estimate network parameters and to use them to build, simulate and evaluate a model. The left side indicates that some useful information (such as functional knowledge or structure information) could also be derived from the data set to help the network reconstruction.

To reconstruct a network, experimental iterations and a priori knowledge are required until sufficient data are available to computationally infer the network structure. In the case of GRNs, this procedure involves altering the gene network in some way, observing the outcome, and using mathematics and logic (i.e. computational methods) to infer the underlying principles of the network. To derive a realistic model, available domain knowledge (including functional and structural information) can be integrated into the computational methods. Figure 2 illustrates the general procedure of a reverse engineering approach for modeling GRNs from quantitative expression data. The right-hand side of this figure indicates the procedure of employing computational methods to derive network parameters for a given model, to build and simulate the model, and to evaluate it by comparing the behavior of the inferred model with the original data set. As indicated in this figure, in addition to the direct use of expression data, some useful information can also be extracted from the data set for network reconstruction. For example, the gene name can be mapped into knowledge bases (e.g. gene ontology) to obtain biological knowledge (e.g. gene function) that can be furthermore used to determine the solutions. If part of the structure information is available, it can be used to measure important network features, such as edge betweenness or network density, to validate the inferred networks. In this review, we discuss, from both computational and biological perspectives, numerous methods that vary according to the size of the network the method can handle.

Now that different types of reverse engineering approaches have been developed, the next major concern is how to use the inferred GRN to generate hypotheses that can be tested in the laboratory. One recently developed strategy is to combine information from various sources to narrow down the search space in the network, thus shortening the time and effort required for validation and discovery. In this review, we also introduce some important attempts by biologists to apply predicted gene networks to bench experiments that have led to many new findings.

Readers are encouraged to consult several reviews on inferring gene networks [14–24].

NETWORK MODELING

In the process of reconstructing networks from expression data, the most important steps are selecting the network model and fitting the available data into the network's structural parameters. Many gene regulation models have been proposed. These models range from very abstract (involving Boolean values only) to very concrete (including fully biochemical interactions with stochastic kinetics), depending on the biological level to be studied.

Abstract models involve less biological detail and display only qualitative dynamic behavior; however, they are uniquely capable of implementing large-size networks. On the other hand, concrete models describe network dynamics in detail and are closer to biological reality; but they can only implement small size networks. In this section, we take the computational point of view and categorize gene regulatory models into two major types: those that use discrete and continuous variables, respectively, in the modeling procedure. The models and the methods used to infer them are described, and some network modeling tools recently developed (and currently available online) are listed in Table 1.

Models with discrete variables

Boolean network models

The first type of GRN model assumes that genes only exist in discrete states. This approximation is usually implemented by Boolean variables in which the gene is in either on (active or expressed) or off (inactive or unexpressed). Boolean networks are easy to simulate and are therefore less computationally taxing, but it has been proven that Boolean networks are not able to capture certain system behaviors that can be captured by continuous models [37].

To construct a Boolean network, many computational methods can be adopted. If there is only qualitative knowledge available, literature-based methods are useful. In these methods, sentences in different documents are analyzed and compared to extract the relationships and links between genes [38, 39]. Alternatively, if experimental data are available, the Boolean network can be inferred from time course data. Two classes of methods are often used to infer Boolean networks. One is based on correlation measurement, where different methods are employed to extract information about gene relationships, and this information is then used to model the topological connections between genes [40–43]. For example, the information-theoretic approach is commonly used to calculate the mutual information between genes, which can be used as a correlation measurement. The other class to infer Boolean networks is machine learning-based, in which genetic algorithm (GA) [44] is the most common method for network modeling (e.g. [45–47]). The regulatory functions of network nodes and the relationships between nodes are encoded in the string representations often used in GA, and adaptive operators (such as crossover and mutation) can be used to create new solutions. In addition to using a linear encoding scheme in GA, a graph structure (generally described as a tree) can also be used to represent a Boolean network, with subsequent genetic programming (GP [48]) used to infer the network structure directly.

Because traditional evolutionary algorithms are global search methods that mainly concentrate on exploring the solution space without considering local information, they cannot be optimized via local fine-tuning. Therefore, many enhanced methods have been proposed that combine GA with different local search techniques. These include taboo search, hill-climbing, simulated annealing and the simplex method; all exploit local information to determine promising directions in the search space. More recently, a new suite of intelligent population-based optimization techniques, known as swarm intelligence methods (including ant colony system [49] and particle swarm optimization [50]) was proposed as an alternative to the traditional evolutionary algorithms. Some hybrid methods have now been proposed to effectively exploit the qualities to each of these two types of methods. It is now commonly agreed that a hybrid model comprising both evolutionary algorithm and swarm intelligent methods can lead to further improvements in performance.

Probabilistic Boolean network models

Though Boolean networks are easy to simulate at a reduced computational cost, they are generally considered to be unable to capture many important system behaviors. The dynamics of a Boolean network are deterministic, and they depend on the initial node states. These unexpected features make the Boolean network model lack realism. To overcome this problem, the probabilistic Boolean network (PBN), a similar but revised model, was proposed [51, 52]. Uncertainty is introduced into this newly developed model by providing each network node with multiple regulatory functions, each with a predefined probability. The function at each node is then determined probabilistically. At each time step, the regulatory function of each node is randomly selected from this pool of functions according to their given probabilities. With this randomizing effect, PBNs are stochastic, and the dynamics of the network are no longer deterministic. Any given set of initial node states can result in multiple subsequent network states.

Table I: Recent online tools for GRN inference

Tool	Category	Model	Method	URL
GenYsis [25]	Discrete	Boolean network	Algebra (reduced ordered binary decision diagrams)	http://si2.epfl.ch/~garg/genysis.html
GeNESiS [26]	Discrete, continuous	Graph	Machine learning (GA)	http://genomics.iab.keio.ac.jp/genesis.html
LICORN [27]	Discrete, continuous	Graph	Data mining (Apriori algorithm), heuristic measurement,	http://www.lri.fr/~elati/licorn.html
SIRENE [28]	Discrete, continuous	Graph (probabilistic)	Machine learning (SVM)	http://cbio.Ensmp.fr/sirene
mi3 [29]	Discrete, continuous	Graph (probabilistic)	Information theory (mutual information)	http://sysbio.engin.umich.edu/~luow/downloads.php
ARACNE [7]	Discrete, continuous	Graph (probabilistic)	Information theory (mutual information), statistics (Gaussian Kernel estimator)	http://amdec-bioinfo.cu-genome.org/html/ARACNE.htm
SEBINI [30]	Continuous	Graph, Bayesian network	Information theory (mutual information), statistics (Pearson correlation), Bayesian inference	http://www.emsl.pnl.gov/NIT/NIT.html
LibB [19]	Continuous	Graph, Bayesian networks	Bayesian inference, hill-climbing algorithm	http://www.cs.huji.ac.il/labs/compbio/LibB/
BANJO [31]	Continuous	Dynamic Bayesian network	Bayesian inference	http://www.cs.duke.edu/~amink/software/
FastNCA [32]	Continuous	ODEs (linear model)	Algebra (approximation)	http://www.eee.hku.hk/~cqchang/FastNCA.htm
NIR [33]	Continuous	ODEs (linear model)	Algebra (regression)	http://dibernardo.tigem.it/wiki/index.php/Network_Inference_by_Reverse-engineering_NIR
TSNI [33]	Continuous	ODEs (linear model)	Algebra (regression)	http://dibernardo.tigem.it/wiki/index.php/Time_Series_Network_Identification_TSNI-integral
NetRec [34]	Continuous	ODEs	Statistics (Pearson correlation, partial Pearson correlation and Graphical Gaussian models)	http://people.sissa.it/~altafini/papers/SoBiAI07/
Ometer [35]	Continuous	ODEs	Statistics (Pearson partial correlation)	http://mendes.vbi.vt.edu/tiki-index.php?page=Software
Gepasi [36]	Continuous	ODEs	Statistics (Pearson partial correlation)	http://mendes.vbi.vt.edu/static/Metabolomics04/

The first step for generating a PBN model is to identify some candidate Boolean networks by the aforementioned methods. Once the candidates have been identified, the next step is to compile the functions that the different candidate networks ascribe to each node into n sets of predictor functions with certain designated probabilities. The details of this compilation process can be found in the literature [52]. The main disadvantage of PBN, perhaps not surprisingly, is its increased computational complexity. The methods used in Boolean network inferencing can be revised and applied to PBN, but much more computation time is required to calculate the predicator probabilities. It thus becomes difficult to scale this approach to large networks. Some heuristic methods have been proposed to reduce the amount of computation. For example, Ivanov and Doherty [53] developed two methods (mapping reduction and projection) and considered their effect on the original probability structure of a given PBN. Marshall *et al.* [54] also proposed a method that separates the data sequence into subsequences, infers a Boolean network for each given sub-sequence, and then infers the probabilities of perturbation as well as the selection probabilities governing which network is to be selected.

Bayesian network models

Another popular discrete variable model is the Bayesian network. A Bayesian model is a directed acyclic graph that explicitly establishes probabilistic relationships between network nodes. It describes the probability distribution governing a set of variables by specifying a set of conditional independence assumptions in addition to a set of conditional probabilities. Thus, the arcs between nodes not only indicate the regulatory relationships, but also describe the conditional dependencies (i.e. a family of joint probability distributions of all nodes) between them. In this way, the joint probability for any desired assignment of values to the network variables can be computed. More details of Bayesian networks can be found in relevant volumes [55]. The variables in a Bayesian network can also be continuous, though Bayesian networks are classified as discrete variable models here.

Bayesian methods can be classified as either static or dynamic, depending on whether temporal expression profiles are used for the consideration of dynamics. As the directed network graphs are acyclic by definition, there can be no auto-regulation and no time-course regulation. With these limits, the static Bayesian methods cannot be used to infer regulatory networks with feedback loops. To consider the dynamic processes of networks, dynamic Bayesian methods were thus developed, and these can yield more accurate models. It should be noted, however, that the computational complexity increases significantly [56–58].

Modeling Bayesian networks involves two steps: model selection (or structure learning) and parameter learning. Model selection involves creation of the network structure, and parameter learning involves estimation of the probability values in the tables associated with each network node. The network structure can be inferred by employing a Bayesian scoring metric for model evaluation. For each possible model, the metric defines the score as the logarithm of the probability that the model correctly describes a given set of data. To avoid the overfitting problem, this likelihood is averaged over all parameter values that could possibly define the conditional probability distribution for each model. It should be noted that though Bayesian models have rich statistics and probability semantics, the learning network structure for such models is computationally expensive. For this purpose, some supplementary methods, such as network decomposition (for dimension reduction), and Monte Carlo strategies using random sampling have been developed to enhance performance [59–61].

Models with continuous variables

Differential equation models

Unlike the above discrete variable models, the second type of model uses continuous variables. One of the popular continuous variable models is based on differential equations, which can describe the system dynamics of a GRN more accurately. Many models based on differential equations have been proposed, including some that use linear ordinary differential equations (ODEs) and others that use nonlinear power law differential equations [33, 62–65]. In general, the change in the expression level of a gene at a certain time (discrete or continuous) is characterized by a function that takes the regulatory influence (activation or inhibition) of other genes into account. It can be described as:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = f_i(x_1, x_2, \ldots, x_n, p, u)$$

in which x_i $(1 \leq i \leq n)$ is the expression level of gene i at time t, n is the number of genes, p is parameter

set of the system and u is an external perturbation to the system. The function f_i can be linear, piecewise linear, pseudo-linear, or continuously nonlinear, each describing the system dynamics with a different level of complexity. Detailed descriptions of ODE models can be found in the literature [37].

The most popular and well-researched ODE model, the S-system, is a power law model. It consists of a particular set of ODEs in which the component processes are characterized by power law functions. In the S-system model, the systematic structure can be described as:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{n} x_j^{g_{i,j}} - \beta_i \prod_{j=1}^{n} x_j^{h_{i,j}}$$

Here x_i is the expression level of gene i and n is the number of genes in a genetic network. The non-negative parameters α_i and β_i are rate constants that indicate the direction of mass flow. The real number exponents $g_{i,j}$ and $h_{i,j}$ are kinetic orders that reflect the intensity of interaction from gene j to i. The above set of parameters defines an S-system model.

Compared to the discrete variable models, the differential equation models can more accurately represent the underlying physical phenomena by virtue of their use of continuous variables. In addition, theoretical aspects of systems analysis and of control design in dynamic systems support this type of model. In particular, with nonlinear ODE models (such as S-system models), a given system's steady-state evaluation, control analysis and sensitivity analysis can be established mathematically [66]. It should be noted, however, that these models depend on numerical parameters that are often difficult to establish experimentally. It is thus important to address the stability issue when using this type of model for network construction; it is necessary to investigate whether the system's behavior depends on the exact values of these parameters and/or their initial settings.

Though nonlinear ODEs can more accurately model the system dynamics of gene networks, they are difficult to solve. To derive a model, several methods have been proposed in which the network is often described as a set of linear ODEs and linear algebra methods are then used to solve the equations. For example, Di Bernardo *et al.* [33] developed an approach that took a series of steady-state gene expression measurements following transcriptional perturbations and reduced the linear ODEs to a form of linear regression to solve the equations with relevant techniques accordingly. Researchers have in fact considered the estimation of all parameters in a model as a large-scale parameter optimization problem. The network parameters are identified and extracted from the model, and a cost function is defined. Then, different numerical and approximation methods, such as steady-state analysis and evolutionary algorithms [67–69], are used to solve the optimization problem. As mentioned earlier, evolution-based methods search the solution space with heuristics that are predominantly global; thus, many local search techniques have been integrated in order to enhance their performance.

Neural network models

The other commonly used type of continuous variable model is the neural network-based model. The most successful of these is the recurrent neural network (RNN) [70–73]. This model is biologically plausible and noise-resistant. It is continuous in time, and uses a transfer function to transfer the inputs into a shape similar to that observed in natural processes. In addition, its nonlinear characteristics provide information about the principles of control, as well as about the natural interactions of elements of the modeled system. Most importantly, this model considers feedback loops and takes internal states into account in its operating process. Internal state is the key point that allows this kind of network to oscillate in the absence of external feedback, and this effect does not require the addition of any particular element other than the concentrations of reactants. Therefore, this model is capable of generating oscillatory and periodic activities, and is better able represent the dynamic behavior of systems over time.

There are several RNN architectures ranging from restricted classes of feedback to full interconnection between nodes. As a representative example, we here discuss the type of RNN model that is most widely used for GRN modeling, the fully recurrent neural network. In a fully recurrent net, each node has a link to every node of the net, including itself. In the case of GRNs, this assumes that each node represents a particular gene, and the wiring between the nodes defines the regulatory interactions. The level of expression of a gene at any time can be assessed by the other gene nodes, and the output of a node at the next time step is derived from the expression levels and connection weights

of all the genes connected to it. In other words, the regulatory effect on a certain gene can be regarded as a weighted sum of all of the other genes capable of regulating it. To calculate the expression rate of a gene, the following transformation rules are generally used:

$$\frac{dx_i}{dt} = k_{1,i}G_i - k_{2,i}x_i$$

$$G_i = \left\{1 + e^{-\left(\sum_j w_{i,j}x_j + b_i\right)}\right\}^{-1}$$

where x_i is the actual concentration of the i-th gene product; $k_{1,i}$ and $k_{2,i}$ are the accumulation and degradation rate constants of gene product, respectively; G_i is the regulatory effect on each gene that is defined by a set of weights (i.e. $w_{i,j}$) estimating the regulatory influence of gene j on gene i, and an external input b_i representing the reaction delay parameter.

The construction of a GRN with this model involves generating a network with nodes and edges corresponding to levels of gene expression as measured in microarray experiments, and deriving correlation coefficients describing the relationships between genes.

By introducing a scoring function for network performance evaluation, the above task can be regarded as a parameter estimation problem with the goal of maximizing network performance (or minimizing an equivalent error measure). Greedy algorithms based on the gradient descent method, such as back-propagation through time (BPTT) [74], have been developed to efficiently update the relevant parameters of recurrent networks in discrete time steps. However, in the learning procedure, the error surface often has a different gradient along each weight direction, so each weight requires a separate learning rate. It is thus difficult to achieve efficient training due to the requirement that appropriate values in all learning rates be chosen simultaneously. To facilitate the process of choosing appropriate learning rates, heuristic algorithms such as delta-bar-delta [75] have been implemented for automatic parameter adjustment. In addition, global parameter optimization techniques, including evolutionary algorithms and the swarm intelligence methods described previously, can be employed in conjunction with local search methods to facilitate estimation. However, due to its computational complexity, this modeling approach can currently only be applied to very small systems.

NETWORK MODELING INCORPORATING MULTIPLE BIOLOGICAL DATA

GRN construction using correlation and TFBS information

Correlation is often used to discover sets of genes with similar expression profiles. However, the quality of expression data can be variable and changes in the expression of some TFs is not easily detectable. Even functionally related genes do not always appear to be co-expressed. By utilizing expression data from different experimental conditions, a first-order correlation coefficient between two genes (a doublet) can be calculated. Zhou *et al.* [76] introduced a second-order coefficient, in which first-order correlation coefficients are used as descriptors for comparing two doublets (yielding a quadruplet). Using 618 yeast expression profiles consisting of 39 cDNA and Affymetrix array experiments, 278 799 quadruplets out of 13 million pairs of 5142 doublets were identified to have correlation coefficients greater than 0.6. While 84% of the identified quadruplets are functionally related genes, only 54% of the 4186 co-expressed gene pairs [77] are functionally related. This method for grouping gene pairs allowed 79 functional annotations to be appended to 67 unknown genes.

GRNs can be inferred from information regarding TFs and their binding sites in the *cis*-regulatory sequences of their target genes [78]. As mentioned in Zhou *et al.* [76], the assignment of TFs to biological processes is a difficult task. Despite this, a GRN constructed by this method has been used to provide evidence that the TF *GAT3* plays a role in both mitosis and meiosis. This was a novel prediction, and though it has not yet been validated experimentally, several lines of evidence support it [79–81].

GRN construction by an information-theoretic approach

This method, termed ARACNe (algorithm for the reconstruction of accurate cellular networks), identifies co-regulated gene pairs of high statistical significance by using mutual information and data processing inequality (DPI) to eliminate indirect relationships [7, 82]. ARACNe can reconstruct a hierarchical and scale-free network in which the link count (number of interactions) of a node (gene) follows a broad-tailed distribution that is approximated as a power-law, $P(k) \sim k^{-\gamma}$, where k is the link count and γ is the degree exponent [83–86].

The expression profiles of 336 genes (corresponding to 336 B-cell phenotypes) were used to infer ~129 000 interactions. This hierarchical and scale-free GRN is characterized by densely-linked hubs. The top 5% of the hubs accounted for nearly 40% of the gene–gene relationships, and a Gene Ontology [87] biological processes analysis revealed that the genes linked to these hubs were preferentially involved in cell cycle regulation, protein synthesis, catabolism, RNA processing and metabolism.

The proto-oncogene *MYC* is on the list of major hubs. A subnetwork centered on *MYC* indicated that 56 genes are direct targets of *MYC*, and 27 of these were new targets. Chromatin immunoprecipitation (ChIP) revealed that *MYC* binds to the transcription initiation regions of 12 of the 27 new targets. This analysis further revealed that the network predicted *MYC* targets with a success rate of more than 90%. One of the intriguing properties of the MYC subnetwork is that some of the genes directly connected to *MYC* are also busy hubs, with more than 100 gene neighbors. This suggests that *MYC* regulates biological processes in a hierarchical way.

Inference of direct targets in a perturbed GRN

Sometimes the construction of an entire network is not the immediate goal. The widespread use of perturbations such as chemicals, drugs, gene knock-downs (by morpholino, antisense RNA or RNAi) and gene knock-outs have prompted the development of computational methods for determining the direct targets of these perturbations. Time-series network identification (TSNI) [10, 88] is a method that uses ODE, assuming that the changes of the transcript concentration of one gene are proportional to those of other genes at previous time and perturbation. TSNI was used to search for the direct target of *TRP63*, a TF required for the morphogenesis and maintenance of all stratified epithelia [89, 90]. Tamoxifen-induced expression of *TRP63* in primary mouse keratinocytes changed the expression of 786 transcripts corresponding to 639 genes. TSNI selected 67 down-regulated genes and 33 up-regulated genes, most of which were involved in cell differentiation. Predicted targets were validated by another round of gene expression profiling in *TRP63* knock-down primary mouse keratinocytes. ChIP-chip experiments were also employed to confirm TSNI predictions. Surprisingly, the expression of several components of the *AP-1* complex was found to be under the control of *TRP63*.

Inference of modules using correlation, pathway, and enriched TFBS data

Most reverse engineering approaches rely solely on temporal gene expression profiling to construct GRNs. However, in the post-genome era, biologists tend to collect information regarding the phenomenon of interest from as many sources as possible. Databases with a wide variety of content are readily available. These databases compile gene expression profiles, protein–protein interactions (PPIs), TF binding sites, pathways and research literature. One might be lucky enough to streamline the data flow in such a way that the output of one program could be fed into the input dialog box of a subsequent program. Mobini *et al.* [6] employed an integrated approach that aims to identify the disease genes involved in allergies. The gene expression profiles of 36 patient and control samples were used to calculate Pearson's correlation coefficients and to build symmetric correlation matrices. Representations were used to construct a correlation graph with vertices joined by edges. Gene ontology was used to prioritize the maximum clique, and genes were selected if they were differentially expressed, correlated and cliquified [91]. The authors identified 103 genes from correlation matrices that contained millions of correlated pairs.

These 103 genes were subsequently input into a manually curated pathway database [ingenuity pathway analysis, (IPA)] that collates physical, transcriptional and enzymatic interactions between mammalian orthologs. IPA predicted that the T-cell receptor (TCR) pathway was most likely to be the common pathway among the disease-associated genes. The expression patterns of the genes in the TCR pathway were investigated in allergen-challenged $CD4^+$ cells from patients with seasonal allergic rhinitis (SAR) to confirm their involvement in the disease. Though several of the genes were differentially expressed, only one gene, *ITK*, was selected; this was due to its involvement in the polarization of Th2 cells and its location in a chromosomal susceptibility region for allergy. A subsequent *in vivo* test using *ITK* mice confirmed the importance of *ITK* in allergic inflammation of the upper and lower airways. Remarkably, the authors went on to find the upstream signal by

locating evolutionarily conserved transcription binding sites in the promoter regions of *ITK* and two other TCR pathway genes, *CD3D* and *LEF1*. *GATA3* appears to be the master regulator of allergies because it is highly expressed in CD4+ cells from atopic patients. Gene expression patterns in allergen-challenged CD4+ cells from patients with SAR were analyzed, and 37 genes were found to correlate with the expression pattern of *GATA3*. Of these 37 genes, *IL7R* was singled out as the gene with the highest number of interactions (five) with known disease-associated genes. The inhibitory effect of *IL7R* in allergic inflammation was proposed to result from a shift in the balance between Th1 and Th2 cells.

Emergent method by the integrated use of databases of various sources and forms

We here generalize the approach adopted by Mobini *et al.* [6], which emphasizes the importance of utilizing information from multiple sources to narrow down the search space after a GRN is constructed from gene expression profiling data (Figure 3, Table 2). After constructing a GRN with reverse engineering methods (such as maximum clique computations, ODE, ARACNe, among others) [6, 7, 10, 82, 88] (step I in Figure 3 and Table 2), pathway databases (Ingenuity Pathway Analysis, Pathway Studio, KEGG, DAVID, Cytoscape, etc.) [92–97] are used to match the inferred GRN to a module or subnetwork (step II). The module or subnetwork is then used to search for genes with similar gene expression profiles (step IV). Since the genes so identified are likely to be under the control of the same TF, their upstream sequences are searched for common *cis*-regulatory elements (step III) (Genomatix, oPOSSUM, PAP, Amadeus, OTFBS, GeneXPress, Array2BIO, etc.) [98–104]. Genes encoding any TFs so identified are subsequently used to search for interacting proteins in the PPI database (GRID, PIPS, DIP) [105–107] and to search for genes with similar gene expression profiles (step IV).

The sequence of these steps (step I to step IV in Figure 3 and Table 2) can be customized. Biologists can choose the workflow that best suits their research needs by shuffling the order of the steps and the number of times each step is used, as depicted in Figure 3. In general, many steps can work together. Each step builds on the previous step, refines the previous step's information and/or adds to the previous step's impact. In this way, several steps can be woven together with different methodologies, thereby increasing the chances of discovery.

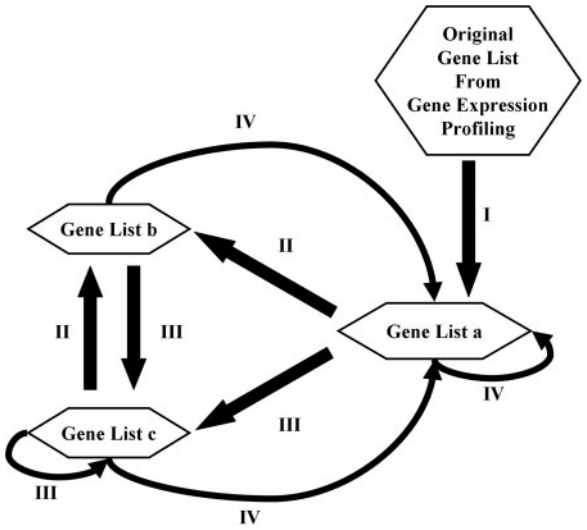

Figure 3: The workflow of searching for modules or subnetworks via integrated use of GRNs and databases of various sources and forms. Biologists can start by producing a gene list from gene expression profiling data and inferred GRNs (step I). At step II, pathway-dependent gene matching is conducted in the network, identifying one or several pathways most relevant to the research question. Upstream sequences of genes involved in these pathways are investigated for evolutionarily conserved TFBSs (step III). TF genes are subsequently used to find genes with similar expression patterns in the gene expression profiling data, and to find interacting protein partners in the PPI data (step IV). The sequence of these steps is not restricted but can be customized. For example, after step I biologists can also follow a different path, starting from step III, and extract evolutionarily conserved TFBSs in the upstream sequences of a set of genes. TF genes can then be used to search for the possible pathways the TF might be involved with (step II). Genes in the pathways are subsequently used to look for those with similar expression patterns in the gene expression profiling data, and/or interacting protein partners in the PPI data (step IV). This scheme is modified from Mobini *et al.* [6]. Step numbers are the same as in Table 2.

SUMMARY AND FUTURE DIRECTIONS

In this article, we introduced the primary network modeling methods used to infer GRNs, and described the application of GRNs to biological

Table 2: Workflow to search for the module and subnetwork based on the integrated use of GRN and multiple databases

Step[a]	Method[b]	Tools[c]	Input gene set[d]	Database feature[e]	Output gene set[f]
I	Inferring GRN based on gene expression profiling data	Reverse engineering (ODE, ARACNe, maximum clique computations, etc.)	Millions of correlate gene pairs	In-house gene expression profiling data	103 genes
II	Gene matching and prioritization in biological pathway	Finding the most relevant pathway (Ingenuity Pathway Analysis, Pathway Studio, KEGG, DAVID, Cytoscape, etc.)	103 genes	Gene, chemical entities and concept database, million articles, pathway and PPI database	One pathway (11 genes)
IV_1	Correlate with the gene expression profiling data and PPI data	Correlate with the gene expression profiling data	11 genes	In-house gene expression profiling data	6 genes
III	Evolutionarily conserved TFBS searching	Up one level in the hierachy of signal transduction cascade (Genomatix, oPOSSUM, PAP, Amadeus, OTFBS, GeneXPress, Array2BIO, etc.)	3 genes	More than 600 DNA binding profiles of TFs	3 TFs
IV_2	Correlate with the gene expression profiling data and PPI data	Correlate with the gene expression profiling data	3 TF	In-house gene expression profiling data	1 TF
IV_3	Correlate with the gene expression profiling data and PPI data	Correlate with the gene expression profiling data and PPI data	1 TF	PPI database and in house gene expression profiling data	37 genes

[a]Steps employed in [6], following the sequence of I, II, IV_1, III, IV_2, IV_3. Codes of steps are the same as in Figure 3.
[b]Methods used in the workflow.
[c]Available tools of the corresponding methods. The tools include website application and software.
[d]Input gene set fed into the corresponding tool.
[e]Source and type of data.
[f]Output gene set generated by the corresponding tool.

questions. Various models have been proposed for simulation of GRNs, and computational methods have been developed for the construction of networks from expression data. Many GRN models share similar ideas and principles; they have improved the performance of network modeling over iterations, and preliminary results support the idea of reverse engineering. However, individual studies were implemented using different techniques. Studies analyzing qualitative network behavior often use abstract models involving fewer biological details. These types of studies have the potential to construct large networks. On the other hand, studies focusing on systems dynamics prefer concrete models with a higher level of biological feasibility. These more complex models can only represent small networks due to their increased computational complexity. In order to provide a broad perspective of the construction of gene networks, this review has described a variety of methods from both computational and biological perspectives. In addition, we have also described how researchers have used GRNs to generate new experiment and new findings.

However, some computational issues remain in the field of network modeling. One such issue is scalability. As the number of genes and interactions in a regulatory network increase, the number of network parameters increases even more rapidly. Therefore, even though many intelligent computing techniques for parameter approximation have been proposed, the solution to this multi-dimensional problem remains to be found. Dimension reduction is one strategy for dealing with the problems of a high-dimensional solution space. Two such strategies have been applied to the optimization problem. The first is the incremental method, which gradually derives the overall solution from partial solutions. In this method, a solution is built to solve a simpler version of the original complex task, such that in the solution, region of the original task is more accessible from the region of the new task. More task versions with incremental complexity can be arranged so that the original task can be achieved progressively. The general strategy of the incremental method is to transform the goal task into another task that is more achievable. In the process of task transformation, the underlying structure of the environment and the goal of the overall task must be preserved. This can be achieved either by arranging the task sequence manually or by an automated procedure.

In addition to the above search-based technique, the other promising way to resolve the high-dimensional problem is to tackle the problem in a 'divide and conquer' manner that involves a network decomposition procedure to reduce task complexity. Clustering is a useful exploratory technique for network decomposition. The hypothesis used by gene clustering is that genes in a cluster may share common functions or regulatory elements and therefore can be considered and modeled together. This technique is used to group genes into tightly coupled small-scale networks that are based on correlations between expression data and function. These small networks can be similarly re-decomposed until the resulting networks can be directly modeled. Thus, small networks are derived directly from expression data. Once all the small networks have been constructed, they can be regarded as self-contained components of the original system, and assembled together manually or by using the learning algorithms described above.

Other issues requiring further investigation are network robustness and reliability. Even though gene networks can be inferred from expression data via computational methods, the limited scope and quality of the available gene expression data raises questions about the biological validity of such networks. It is therefore important not only to study the dynamic properties governed by the particular network parameters, but also to further investigate the robustness and reliability of the overall system.

To infer a robust gene network, the method of adding noise to the training data is useful. In fact, it has been shown that adding noise (of small amplitude) to a set of training data is equivalent to a Tikhonov regularization [108]. When a regularization term is hard to find in the modeling process, adding noise provides a simple alternative. In this way, the predictive capability of the inferred network for the training data set will no longer be perfect, but the robustness will be optimal and the solution will be stable.

Reliability measures how well a constructed network fits in with externally collected expression data that contain internal variation. Parameters that are very sensitive to internal variation can cause fragility in the system. Therefore, two types of analyses are required in reverse engineering. One is local parameter sensitivity analysis, which explores one parameter at a time and deals with small perturbations of the model. The other is global parameter

sensitivity analysis, which explores simultaneous parameter effects and can provide a comprehensive analysis [109, 110]. These analyses provide important ways to understand which parameters need to be considered carefully in order to develop a reliable model.

The construction of a GRN is not just the end of reverse engineering; it is also the start of an inference algorithm with more biological relevance. In this review, we have surveyed recently developed methods that are used to retrieve subnetworks or modules from the wide variety of biological databases currently available. Our aim was not only to provide an overall view of state-of-art methods of reverse engineering, but more importantly to emphasize the ways that biologists can take part in GRN construction by proposing hypotheses and designing experiments.

In this post-genome era, the development of computational methods for inferring GRNs will have to rely on the technologies of information science, engineering and biology. Meanwhile, researchers will have to take into account biological information from new and different sources for GRNs to become workable models. The next generation of GRN software should be part of a customized workflow that better meets the demands of researchers.

Key Points

- Biologists in the post-genome era often obtain a prodigious amount of data churned out by high-throughput techniques, but now face the problem of how to actually utilize these data. Inferring GRNs from gene expression profiling data gives biologists an unprecedented opportunity to determine the intricate, multilayered processes that control gene regulation.
- Many gene regulation models have been proposed, ranging from very abstract models to very concrete ones. Abstract models with less biological detail display only qualitative dynamic behavior; however, they are computationally tractable and therefore provide the opportunity to examine large systems. Concrete models describe network dynamics in detail and are closer to biological reality; however, they are currently only able to implement small networks.
- Various modeling techniques have been developed to solve this problem; these techniques vary according to the type of network model being constructed and include information-theoretic approaches, machine learning approaches and various other optimization methods.
- The use of data and concepts from different sources such as information on pathways, protein–protein interactions (PPIs), transcription factor binding sites and evolutionary conservation can effectively narrow down the search space for the inferred GRN and lead to a workable hypothesis.

FUNDING

National Science Council of Taiwan (NSC 97-2627-B-019-002, NSC 96-2113-M-019-002-MY2 to W.S.T; NSC 97-2221-E-110-063-MY2 to W.P.L).

References

1. Segal E, Shapira M, Regev A, *et al*. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet* 2003;**34**:166–76.
2. Joshi A, De Smet R, Marchal K, *et al*. Module networks revisited: computational assessment and prioritization of model predictions. *Bioinformatics* 2009;**25**:490–6.
3. Xu X, Wang L, Ding D. Learning module networks from genome-wide location and expression data. *FEBS Lett* 2004;**578**:297–304.
4. Bar-Joseph Z, Gerber GK, Lee TI, *et al*. Computational discovery of gene modules and regulatory networks. *Nat Biotechnol* 2003;**21**:1337–42.
5. Cokus S, Rose S, Haynor D, *et al*. Modelling the network of cell cycle transcription factors in the yeast Saccharomyces cerevisiae. *BMC Bioinformatics* 2006;**7**:381.
6. Mobini R, Andersson BA, Erjefalt J, *et al*. A module-based analytical strategy to identify novel disease genes shows an inhibitory role for interleukin 7 Receptor in allergic inflammation. *BMC Syst Biol* 2009;**3**:19.
7. Margolin AA, Nemenman I, Basso K, *et al*. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 2006;**7**(Suppl. 1):S7.
8. Tuck DP, Kluger HM, Kluger Y. Characterizing disease states from topological properties of transcriptional regulatory networks. *BMC Bioinformatics* 2006;**7**:236.
9. Li X, Rao S, Jiang W, *et al*. Discovery of time-delayed gene regulatory networks based on temporal gene expression profiling. *BMC Bioinformatics* 2006;**7**:26.
10. Della GG, Bansal M, Ambesi-Impiombato A, *et al*. Direct targets of the TRP63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. *Genome Res* 2008;**18**:939–48.
11. Kitano H. Perspectives on systems biology. *New Generat Comput* 2000;**18**:199–216.
12. Cho KH, Choo SM, Jung SH, *et al*. Reverse engineering of gene regulatory networks. *IET Syst Biol* 2007;**1**:149–63.
13. Csete ME, Doyle JC. Reverse engineering of biological complexity. *Science* 2002;**295**:1664–9.
14. Stigler B, Jarrah A, Stillman M, *et al*. Reverse engineering of dynamic networks. *Ann NY Acad Sci* 2007;**1115**:168–77.
15. Schlitt T, Brazma A. Current approaches to gene regulatory network modelling. *BMC Bioinformatics* 2007;**8**(Suppl. 6):S9.
16. Gilbert D, Fuss H, Gu X, *et al*. Computational methodologies for modelling, analysis and simulation of signalling networks. *Brief Bioinform* 2006;**7**:339–53.
17. Bansal M, Belcastro V, Ambesi-Impiombato A, *et al*. How to infer gene networks from expression profiles. *Mol Syst Biol* 2007;**3**:78.

18. Bellazzi R, Zupan B. Towards knowledge-based gene expression data mining. *J Biomed Inform* 2007;**40**:787–802.
19. Friedman N. Inferring cellular networks using probabilistic graphical models. *Science* 2004;**303**:799–805.
20. Markowetz F, Spang R. Inferring cellular networks—a review. *BMC Bioinformatics* 2007;**8**(Suppl. 6):S5.
21. Karlebach G, Shamir R. Modelling and analysis of gene regulatory networks. *Nat Rev Mol Cell Biol* 2008;**9**:770–80.
22. Baumbach J, Tauch A, Rahmann S. Towards the integrated analysis, visualization and reconstruction of microbial gene regulatory networks. *Brief Bioinform* 2009;**10**:75–83.
23. Ernst J, Vainas O, Harbison CT, *et al*. Reconstructing dynamic regulatory maps. *Mol Syst Biol* 2007;**3**:74.
24. Hecker M, Lambeck S, Toepfer S, *et al*. Gene regulatory network inference: data integration in dynamic models—a review. *Biosystems* 2009;**96**:86–103.
25. Gary A, Di Cara A, Xenarios I, *et al*. Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics* 2008;**24**:1917–25.
26. Kratz A, Tomita M, Krishnan A. GeNESiS: gene network evolution simulation software. *BMC Bioinformatics* 2008;**9**:541.
27. Elati M, Neuvial P, Bolotin-Fukuhara M, *et al*. LICORN: learning cooperative regulation networks from gene expression data. *Bioinformatics* 2007;**23**:2407–14.
28. Mordelet F, Vert J. SIRENE: supervised inference of regulatory networks. *Bioinformatics* 2008;**24**:i76–i82.
29. Luo W, Hankenson KD, Woolf PJ. Learning transcriptional regulatory network from high throughput gene expression data using continuous three-way mutual information. *BMC Bioinformatics* 2008;**9**:467.
30. Taylor RC, Shah A, Treatman C, *et al*. SEBINI: software environment for biological network inference. *Bioinformatics* 2006;**22**:2706–8.
31. Bernard A, Hartemink AJ. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. *Pac Symp Biocomput* 2005;**10**:459–70.
32. Chang C, Ding Z, Hung YS, *et al*. Fast network component analysis (FastNCA) for gene regulatory network reconstruction from microarray data. *Bioinformatics* 2008;**24**:1349–58.
33. Di Bernardo D, Gardner TS, Collins JJ. Robust identification of large genetic networks. *Pac Symp Biocomput* 2004;**9**:486–97.
34. Soranzo N, Bianconi G, Altafini C. Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. *Bioinformatics* 2007;**23**:1640–7.
35. de la Fuente A, Bing N, Hoeschele I, *et al*. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics* 2004;**20**:3565–74.
36. Camacho D, de la Fuente A, Mendes P. The origins of correlations in metabolomics data. *Metabolomics* 2005;**1**:53–63.
37. de Jong H. Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol* 2002;**9**:67–103.
38. Chiang JH, Yu HC. MeKE: discovering the functions of gene products from biomedical literature via sentence alignment. *Bioinformatics* 2003;**19**:1417–22.
39. Hernminger BM, Saelim B, Sullivan PF, *et al*. Comparison of full-text searching to metadata searching for genes in two biomedical literature cohorts. *J Am Soc Inf Sci Tec* 2007;**58**:2341–52.
40. Chen CC, Zhong S. Inferring gene regulatory networks by thermodynamic modeling. *BMC Genomics* 2008;**9**(Suppl. 2):S19.
41. Laubenbacher R, Stigler B. A computational algebra approach to the reverse engineering of gene regulatory networks. *J Theor Biol* 2004;**229**:523–37.
42. Liang S, Fuhrman S, Somogyi R. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput* 1998;**3**:18–29.
43. Mehra S, Hu WS, Karypis G. A Boolean algorithm for reconstructing the structure of regulatory networks. *Metab Eng* 2004;**6**:326–39.
44. Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin Heidelberg, Germany: Springer-Verlag, 1994.
45. Iba H, Mimura A. Inference of a gene regulatory network by means of interactive evolutionary computing. *Information Sciences* 2002;**145**:225–36.
46. Keedwell E, Narayanan A. Discovering gene networks with a neural-genetic hybrid. *IEEE/ACM Trans Comput Biol Bioinform* 2005;**2**:231–42.
47. Repsilber D, Liljenstrom H, Andersson SG. Reverse engineering of regulatory networks: simulation studies on a genetic algorithm approach for ranking hypotheses. *Biosystems* 2002;**66**:31–41.
48. Koza J. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MA, USA: MIT Press, 1992.
49. Dorigo M, Stützle T. *Ant Colony Optimization*. MA, USA: MIT Press, 2004.
50. Kennedy J, Eberhart R. *Swarm Intelligence*. CA, USA: Morgan Kaufmann Publishers, 2001.
51. Li P, Zhang C, Perkins EJ, *et al*. Comparison of probabilistic Boolean network and dynamic Bayesian network approaches for inferring gene regulatory networks. *BMC Bioinformatics* 2007;**8**(Suppl. 7):S13.
52. Shmulevich I, Dougherty ER, Kim S, *et al*. Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 2002;**18**:261–74.
53. Ivanov I, Dougherty ER. Reduction mappings between probabilistic Boolean networks. *EURASIP J Appl Si Pr* 2004;**2004**:125–31.
54. Marshall S, Yu L, Xiao Y, *et al*. Inference of a probabilistic Boolean network from a single observed temporal sequence. *EURASIP J Bioinform Syst Biol* 2007;**2007**:32454.
55. Neapolitan R. *Learning Bayesian Networks*. NJ, USA: Prentice Hall, 2003.
56. Hartemink AJ, Gifford DK, Jaakkola TS, *et al*. Bayesian methods for elucidating genetic regulatory networks. *IEEE Intelligent Systems* 2002;**17**:37–43.
57. Kim SY, Imoto S, Miyano S. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Brief Bioinform* 2003;**4**:228–35.
58. Ong IM, Glasner JD, Page D. Modelling regulatory pathways in *E. coli* from time series expression profiles. *Bioinformatics* 2002;**18**(Suppl. 1):S241–S48.
59. Zou M, Conzen SD. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory

networks from time course microarray data. *Bioinformatics* 2005;**21**:71–9.

60. Werhli AV, Husmeier D. Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat Appl Genet Mol Biol* 2007;**6**:15.
61. Kim CS. Bayesian Orthogonal Least Squares (BOLS) algorithm for reverse engineering of gene regulatory networks. *BMC Bioinformatics* 2007;**8**:251.
62. Kimura S, Ide K, Kashihara A, *et al*. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics* 2005;**21**:1154–63.
63. Kikuchi S, Tominaga D, Arita M, *et al*. Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics* 2003;**19**:643–50.
64. Cinquemani E, Milias-Argeitis A, Summers S, *et al*. Stochastic dynamics of genetic networks: modelling and parameter identification. *Bioinformatics* 2008;**24**:2748–54.
65. Savageau MA. Rules for the evolution of gene circuitry. *Pac Symp Biocomput* 1998;**3**:54–65.
66. Voit E. *Computational Analysis of Biochemical Systems*. Cambridge, UK: Cambridge University Press, 2000.
67. Curto R, Voit EO, Sorribas A, *et al*. Validation and steady-state analysis of a power-law model of purine metabolism in man. *Biochem J* 1997;**324**:761–75.
68. Ho SY, Hsieh CH, Yu FC, *et al*. An intelligent two-stage evolutionary algorithm for dynamic pathway identification from gene expression profiles. *IEEE/ACM Trans Comput Biol Bioinform* 2007;**4**:648–60.
69. Noman N, Iba H. Inferring gene regulatory networks using differential evolution with local search heuristics. *IEEE/ACM Trans Comput Biol Bioinform* 2007;**4**:634–47.
70. Vohradsky J. Neural network model of gene expression. *FASEB J* 2001;**15**:846–54.
71. Xu R, Venayagamoorthy GK, Wunsch DC. Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. *Neural Networks* 2007; **20**:917–27.
72. Blasi MF, Casorelli I, Colosimo A, *et al*. A recursive network approach can identify constitutive regulatory circuits in gene expression data. *Physica A* 2005;**348**:349–70.
73. Lee WP, Yang KC. A clustering-based approach for inferring recurrent neural networks as gene regulatory networks. *Neurocomputing* 2008;**71**:600–10.
74. Werbos PJ. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 1990;**78**:1550–60.
75. Jacobs RA. Increased rates of convergence through learning rate adaptation. *Neural Networks* 1988;**1**:295–307.
76. Zhou XJ, Kao MC, Huang H, *et al*. Functional annotation and network reconstruction through cross-platform integration of microarray data. *Nat Biotechnol* 2005;**23**: 238–43.
77. Stuart JM, Segal E, Koller D, *et al*. A gene-coexpression network for global discovery of conserved genetic modules. *Science* 2003;**302**:249–55.
78. Lee TI, Rinaldi NJ, Robert F, *et al*. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* 2002;**298**: 799–804.
79. Chu S, DeRisi J, Eisen M, *et al*. The transcriptional program of sporulation in budding yeast. *Science* 1998;**282**: 699–705.
80. Giaever G, Chu AM, Ni L, *et al*. Functional profiling of the *Saccharomyces cerevisiae* genome. *Nature* 2002;**418**: 387–91.
81. Primig M, Williams RM, Winzeler EA, *et al*. The core meiotic transcriptome in budding yeasts. *Nat Genet* 2000; **26**:415–23.
82. Basso K, Margolin AA, Stolovitzky G, *et al*. Reverse engineering of regulatory networks in human B cells. *Nat Genet* 2005;**37**:382–90.
83. Barabasi AL, Oltvai ZN. Network biology: understanding the cell's functional organization. *Nat Rev Genet* 2004;**5**: 101–13.
84. Jeong H, Mason SP, Barabasi AL, *et al*. Lethality and centrality in protein networks. *Nature* 2001;**411**:41–2.
85. Wagner A. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol Biol Evol* 2001;**18**:1283–92.
86. Yook SH, Oltvai ZN, Barabasi AL. Functional and topological characterization of protein interaction networks. *Proteomics* 2004;**4**:928–42.
87. Gene Ontology, http://www.geneontology.org/GO.annotation.shtml (28 May 2009, date last accessed).
88. Bansal M, Gatta GD, di Bernardo D. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics* 2006;**22**: 815–22.
89. Mills AA, Zheng B, Wang XJ, *et al*. p63 is a p53 homologue required for limb and epidermal morphogenesis. *Nature* 1999;**398**:708–13.
90. Yang A, Schweitzer R, Sun D, *et al*. p63 is essential for regenerative proliferation in limb, craniofacial and epithelial development. *Nature* 1999;**398**:714–18.
91. Voy BH, Scharff JA, Perkins AD, *et al*. Extracting gene networks for low-dose radiation using graph theoretical algorithms. *PLoS Comput Biol* 2006;**2**:e89.
92. Nikitin A, Egorov S, Daraselia N, *et al*. Pathway studio—the analysis and navigation of molecular networks. *Bioinformatics* 2003;**19**:2155–57.
93. The Ingenuity Pathway Analysis tool (IPA), http://www.ingenuity.com/products/pathways_analysis.html (28 May 2009, date last accessed).
94. Dennis G, Jr., Sherman BT, Hosack DA, *et al*. DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome Biol* 2003;**4**:3.
95. Okuda S, Yamada T, Hamajima M, *et al*. KEGG Atlas mapping for global analysis of metabolic pathways. *Nucleic Acids Res* 2008;**36**:W423–W426.
96. Yeung N, Cline MS, Kuchinsky A, *et al*. Exploring biological networks with Cytoscape software. *Curr Protoc Bioinformatics* 2008;Chapter 8: Unit 8.13.
97. Bindea G, Mlecnik B, Hackl H, *et al*. ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. *Bioinformatics* 2009.
98. Genomatix, http://www.genomatix.de/products/Gene2Promoter/index.html (28 May 2009, date last accessed).
99. Ho Sui SJ, Mortimer JR, Arenillas DJ, *et al*. oPOSSUM: identification of over-represented transcription factor binding sites in co-expressed genes. *Nucleic Acids Res* 2005; **33**:3154–64.
100. Chang LW, Nagarajan R, Magee JA, *et al*. A systematic model to predict transcriptional regulatory mechanisms

based on overrepresentation of transcription factor binding profiles. *Genome Res* 2006;**16**:405–13.

101. Linhart C, Halperin Y, Shamir R. Transcription factor and microRNA motif discovery: the Amadeus platform and a compendium of metazoan target sets. *Genome Res* 2008;**18**:1180–9.
102. Zheng J, Wu J, Sun Z. An approach to identify over-represented *cis*-elements in related sequences. *Nucleic Acids Res* 2003;**31**:1995–2005.
103. GeneXpress, http://genexpress.stanford.edu/tutorials/attribute_analysis.html (28 May 2009, date last accessed).
104. Loots GG, Chain PS, Mabery S, *et al*. Array2BIO: from microarray expression data to functional annotation of co-regulated genes. *BMC Bioinformatics* 2006;**7**:307.
105. Breitkreutz BJ, Stark C, Tyers M. The GRID: the General Repository for Interaction Datasets. *Genome Biol* 2003;**4**:R23.
106. Xenarios I, Eisenberg D. Protein interaction databases. *Curr Opin Biotechnol* 2001;**12**:334–9.
107. McDowall MD, Scott MS, Barton GJ. PIPs: human protein–protein interaction prediction database. *Nucleic Acids Res* 2009;**37**:D651–D656.
108. Bishop CM. Training with noise is equivalent to tikhonov regularization. *Neural Computation* 1994;**7**:108–16.
109. Ihekwaba AE, Broomhead DS, Grimley RL, *et al*. Sensitivity analysis of parameters controlling oscillatory signalling in the NF-kappaB pathway: the roles of IKK and IkappaBalpha. *Syst Biol* 2004;**1**:93–103.
110. Zhang Y, Rundell A. Comparative study of parameter sensitivity analyses of the TCR-activated Erk-MAPK signalling pathway. *Syst Biol* 2006;**153**:201–11.