

Reverse engineering of gene regulatory networks

K.-H. Cho, S.-M. Choo, S.H. Jung, J.-R. Kim, H.-S. Choi and J. Kim

Abstract: Systems biology is a multi-disciplinary approach to the study of the interactions of various cellular mechanisms and cellular components. Owing to the development of new technologies that simultaneously measure the expression of genetic information, systems biological studies involving gene interactions are increasingly prominent. In this regard, reconstructing gene regulatory networks (GRNs) forms the basis for the dynamical analysis of gene interactions and related effects on cellular control pathways. Various approaches of inferring GRNs from gene expression profiles and biological information, including machine learning approaches, have been reviewed, with a brief introduction of DNA microarray experiments as typical tools for measuring levels of messenger ribonucleic acid (mRNA) expression. In particular, the inference methods are classified according to the required input information, and the main idea of each method is elucidated by comparing its advantages and disadvantages with respect to the other methods. In addition, recent developments in this field are introduced and discussions on the challenges and opportunities for future research are provided.

1 Introduction

In order to have a better understanding on complex biological phenomena and disease mechanisms, we need to unravel the interaction structure of molecular components involved in the cellular processes rather than just characterising the properties of individual components. In this paper, we focus on gene regulatory networks (GRNs) representing the interaction structure of genes [1]. In general, a GRN is represented by a directed graph composed of nodes (genes) and links (regulatory relationships). The regulatory relationship can be either an activation (i.e. inducing transcription of other genes) or an inhibition (i.e. repressing transcriptional activity). Two nodes without a link imply that no regulatory relationship exists between them. Inference of such a GRN for a specific part or the entire genome can help us to unravel the gene interaction mechanism for a particular stimulation and we can further utilise this information to predict adverse effects of new drugs or to identify a new drug target. Owing to the development of new high-throughput measurement technologies such as DNA microarray, ChIP (CHromatin ImmunoPrecipitation)-chip experiments and protein-protein interaction measurements (see Section 2), there is a renewed interest in unravelling the hidden GRN. The inference of such a GRN from either

gene expression profiles [more precisely, messenger ribonucleic acid (mRNA) expression profiles] or DNA sequence information is often called 'reverse engineering'. Various reverse engineering methods have been developed to infer such a GRN; however, because of both experimental limitations and methodological complexities, a large majority of these methods have been not so successful as there are (i) a dimensionality problem: too many genes with too few available sampling time points, (ii) a computational complexity problem: exponential complexity if a priori information is unavailable for regulatory genes and (iii) an experimental measurement problem: no guidelines for an appropriate experimental design for distinguishing direct and indirect influences among genes. Hence, we need to understand the essential features of each method before we apply any particular method and to choose a most suitable one considering given conditions and available data. In this respect, we review the previously developed reverse engineering methods. There are some review papers on reverse engineering [2–6], but we approach in a different way. We explicitly classify each method depending on the required input data and the inference outputs, elucidate the key idea of each method and compare the advantages and disadvantages.

The overall procedure of reverse engineering GRNs is illustrated in Fig. 1. We need to understand the required input information of each reverse engineering method as some methods presume specific types of data produced from experiments having a particular design. The input information for inference methods can be either gene expression data or biological information data such as DNA sequences and annotations (see Section 4). As a large number of gene expression levels can be measured simultaneously using DNA microarray, we can use gene expression data for reverse engineering of a GRN. However, there is a limitation in inferring a GRN using only the expression data and hence, it has been proposed to make further use of diverse biological information. The inference methods requiring expression data include Boolean methods (Section 3.1), Bayesian methods (Section 3.2) and regulation matrix methods (Section 3.3), and

© The Institution of Engineering and Technology 2007

doi:10.1049/iet-syb:20060075

Paper first received 7th April 2006 and in revised form 30th January 2007

K.-H. Cho is with the College of Medicine, Seoul National University, Jongno-gu, Seoul 110-799, South Korea

S.-M. Choo is with the School of Electrical Engineering, University of Ulsan, Ulsan 680-749, South Korea

S.H. Jung is with the Department of Information and Communication Engineering, Hansung University, Seoul 136-792, South Korea

J.-R. Kim is with the Bio-MAX Institute, Seoul National University, Gwanak-gu, Seoul 151-818, South Korea

H.-S. Choi and J. Kim are with the Interdisciplinary Program in Bioinformatics, Seoul National University, Gwanak-gu, Seoul 151-747, South Korea

K.-H. Cho is also with the Bio-MAX Institute, Seoul National University, Gwanak-gu, Seoul 151-818, South Korea

E-mail: ckh-sb@snu.ac.kr (K.-H. Cho)

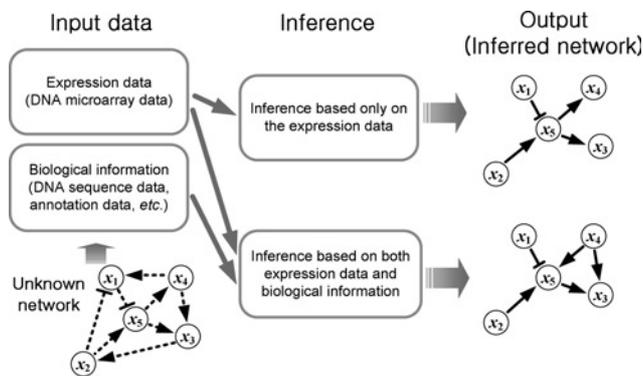


Fig. 1 Schematic diagram of typical procedures for reverse engineering of GRNs

those requiring biological information as well as expression data include the MODEM (MODule construction using gene Expression and sequence Motif) and GRAM (Genetic RegulATory Modules) methods (Section 4).

We also note that each inference method provides us with various forms of the inferred GRNs. For example, some methods only provide correlations between genes, whereas others may provide detailed regulatory relations such as activation and inhibition. In some cases, the regulatory relations are represented by probability and the inferred networks are represented by module regulatory networks. Hence, we need to understand what kind of inference ‘output’ can be obtained from each reverse engineering method in order to choose the most appropriate one for a given purpose. In this paper, we review various reverse engineering methods in these respects and then provide a useful guide for researchers who are interested in investigating new reverse engineering methods.

2 DNA microarray experiments and sequence motifs

In this section, we briefly review DNA microarray experiments and DNA sequence motifs, both of which provide useful input information for reverse engineering of GRNs. Further details are to be found in [7–28].

2.1 DNA microarray experiments

We can measure genome-wide gene expression through a DNA microarray experiment which is basically an extended version of Northern blotting [7] measuring the abundance of mRNAs separated by electrophoresis [8–10]. A DNA microarray experiment is devised to simultaneously measure ten to hundreds of thousands of mRNA expression levels of a given sample [29], whereas Northern blotting is to measure a single mRNA expression level of a selected gene with more accuracy. The central principle of measuring mRNA expression levels is the base pairing between unknown sequences of mRNAs in sample cells and known complementary DNA sequences of the target genes [i.e. A (adenine) pairs with T (thymine), and G (guanine) pairs with C (cytosine)]. There are two types of microarrays: cDNA microarray and oligonucleotide microarray [11]. The overall procedures of DNA microarray experiments are illustrated in Fig. 2 considering cDNA microarray.

As illustrated in Fig. 2, DNA microarray experiments are composed of multiple steps which imply several possible noise sources. For instance, there might be problems

caused by different binding affinities depending on DNA sequences [12], physical contamination, saturation effect of each pixel depending on the laser excitation intensity [8, 13], different dye effects [14, 15] and so on. In particular, we note that the microarray data table (Fig. 2i) can be changed depending on the definition of the representative value in data1 and data2 (Fig. 2h) within the same experiment [18]. Hence, we need to carefully design DNA microarray experiments to minimise such potential errors and to normalise the measured expression data by statistically eliminating any systematic errors [8, 12–17, 28], which are the main topics for microarray bioinformatics but beyond the scope of this paper.

2.2 DNA sequence motifs

In addition to gene expression data, we can also make use of various biological information for reverse engineering of GRNs including the followings: transcription factor (TF) binding sites (sequence motifs) [19], ChIP–chip data for TF binding information based on ChIP [20], gene annotations [e.g. the gene annotation of SWI4 is ‘DNA binding component of SBF complex (Swi4p–Swi6p)’] [20] and protein–protein interaction data. Such information can be used in the following processes: identification of candidate genes producing TFs, investigation of the TF binding sites [21] and search for genes in relation with the TF binding sites [22]. We cannot, however, determine all the binding sites of TFs through experimentation [21, 23, 24] because the number of TFs is exceedingly large (2000–3000 for the human) and the genome size is also even larger (more than 3 billion base pairs for human) [21]. As a result of this large scale, some *in silico* approaches have been developed [19, 21, 25, 26]. Among them, a weight matrix method (Fig. 3) has been generally used [19, 21]; sequences known to bind with a TF are collected (Fig. 3a) and then the distribution of bases at each position is computed (Fig. 3b). The consensus sequence of 6 bp (Fig. 3c), in which the base at each position is the most frequent one, can be found from this distribution, where a TF can bind the consensus sequence. However, as a TF can also bind with other sequences as seen in the experimental results of Fig. 3a, an alternate consensus sequence is also adopted (Fig. 3c). The alternate consensus sequence is the sequence permitting the possible plural bases at some position in the consensus sequence [21]. As seen in Fig. 3c, ‘GYNGAG’ can be an alternate consensus sequence where N represents an arbitrary base and Y denotes a pyrimidine (i.e. cytosine or thymine). R is conventionally employed to indicate a purine (i.e. adenine or guanine). A method of using a position weight matrix was developed (Fig. 3d) in order to evaluate how strongly a TF binds to the corresponding sequence. The weight $W(b, l)$ of base b at l th position can be calculated from the distribution of bases at each position (Fig. 3b) using the equation $W(b, l) = 10(2 + \log_2 f(b, l))$ where $f(b, l)$ denotes the frequency of base b at l th position. This kind of *in silico* approach for finding a relationship between TFs and genes based on only binding sites, however, still has the following two fundamental limitations: (i) there are too many candidate binding sites; (ii) it is not clear whether a binding site will be the *cis*-regulatory element (region of DNA or RNA which regulates the expression of genes located on that same strand) of a particular gene. These are because the protein coding region is relatively very small compared with the whole genome size in many evolved complex organisms (about 5% for human) or the binding site can be far away from the transcription initiation site and even be placed in the

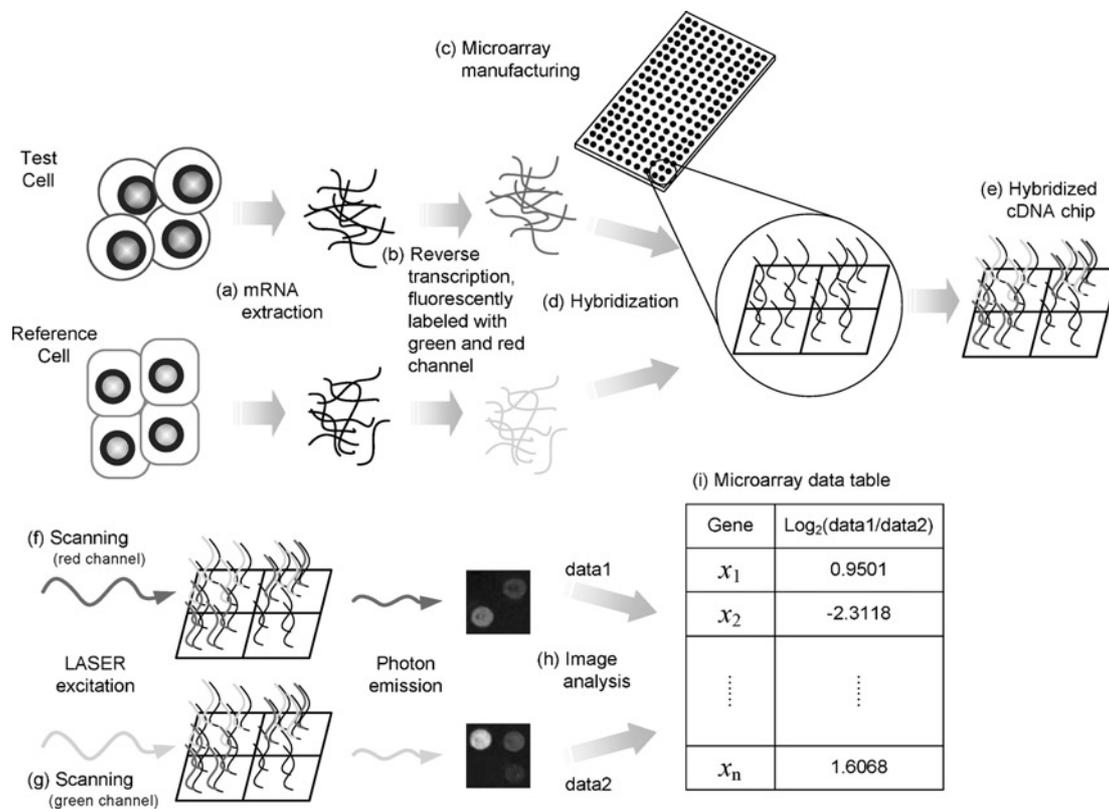


Fig. 2 Schematic diagram of the overall procedures of cDNA microarray experiments [8]

- a mRNA extraction from test cells and reference cells (e.g. tumour cells and normal cells, respectively)
- b Reverse transcription of the mRNA samples of test (reference) cells into the DNAs fluorescently labelled with red (green) fluorescent dye
- c Microarray chip is fabricated with spots having complementary DNA sequences of the target genes
- d Hybridisation of the two mRNA samples (b) on the microarray chip of (c)
- e Hybridised cDNA chip obtained from (d)
- f, g Image data obtained by converting the emitted photon amounts into signal intensities after applying light to stimulate red fluorescence and green fluorescence, respectively
- h Data1 and data2 are obtained by calculating the representative value (e.g. mean, median) of intensities of each spot in image data (f) and (g), respectively
- i Microarray data table is constructed using data1 and data2 (h)

protein coding region [21, 27]. In particular, the binding site of a TF cannot be determined only from the consensus sequences as the binding site depends on multiple factors including chromatin structures [21].

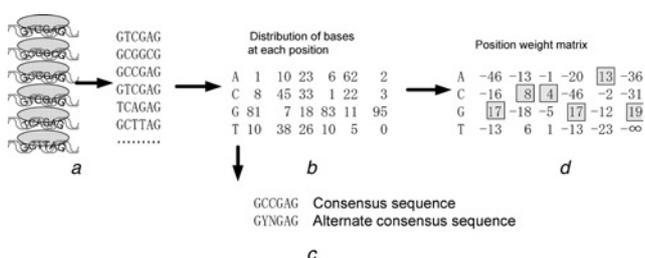


Fig. 3 Schematic diagram illustrating the weight matrix method [19, 21]

- a Protein binding sequences can be obtained from protein–DNA binding assays; grey-coloured ellipses indicate a common TF
- b Distribution of bases at each position: the j th column in the table means the j th position and the number in the i th row and j th column in the table means how many times the base in the i th row appears in the j th position
- c Base of first position in the consensus sequence is ‘G’ as ‘G’ appears the most frequently at first position
In an alternate consensus sequence, the base of second position is ‘Y’ as ‘C’ and ‘T’ appear dominantly at the second position
- d Score (i.e. the possibility of binding between TF and DNA of the sequence) can be evaluated from the position weight matrix
For instance, the binding site ‘GCCGAG’ of a TF has the highest score ($17 + 8 + 4 + 17 + 13 + 19 = 78$)

3 Reverse engineering based on expression profiles

To date, various approaches have been developed to infer a GRN from measured gene expression profiles. GRNs in many cases cannot be unravelled precisely, however, because of measurement noise and the limited number of data sets compared with the number of genes that are involved. In other words, we always have to process insufficient and uncertain data information in reverse engineering of GRNs. Hence, there cannot be one universal method applicable to all practical cases at present. Therefore in order to choose an appropriate method for inferring a GRN, we must understand the fundamental idea of each method and should choose the most appropriate one by considering available data sets and all underlying constraints. In this section, we review the reverse engineering methods based on expression profiles with respect to the main idea, the resulting output, the merits and the limitations of each method. In particular, the Boolean, Bayesian and the regulation matrix methods are considered in this section.

3.1 Boolean methods

Boolean methods are used to infer GRNs by applying Boolean logic to the discretised gene states (e.g. 0 or 1 states in Fig. 4) which indicate the discretised mRNA expression levels. Here 0 and 1 mean an ‘off’ (i.e. inactive

Input data

	Input state			Output state		
	x_1	x_2	x_3	x_1	x_2	x_3
State transition pair 1	0	0	0	0	0	0
State transition pair 2	0	0	1	0	1	0
State transition pair 3	0	1	0	1	0	0
State transition pair 4	0	1	1	1	1	1
State transition pair 5	1	0	0	0	1	0

a

Output data

Argument		x_1	Argument		x_2	Argument				
x_2	x_1		x_1	x_3		x_1	x_2	x_3	x_1	x_2
0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	1	0	0	0
		1	0	1	0	1	0	0	0	0
					0	1	1	1	1	1
					1	0	0	0	0	0

b

Fig. 4 Illustration of the input and output of Boolean methods

a State transition pair table of a network with three nodes and two states (0 or 1)

b First table is the state transition pair table of the first node x_1 with an argument x_2 as an output using REVEAL (REVERSE ENGINEERING ALGORITHM) [39]

The rest are the state transition pair tables of nodes x_2 and x_3 , respectively

or unexpressed) state and an ‘on’ (i.e. active or expressed) state of genes, respectively. Let $x_i(t)$ be the discretised state of a network node x_i ($1 \leq i \leq n$) at t and a pair of two lists $(x_1(t), \dots, x_n(t))$ and $(x_1(t+1), \dots, x_n(t+1))$ be a state transition pair (the first list is an input state and the second list is an output state) where the i th component of an input (output) state is called the input (output) state of x_i . For instance, there exist $n - 1$ state transition pairs for data measured at n time points. Boolean methods are to find a Boolean relation (a gene regulation rule) of each node which explains the influence of input states on the node and then these methods can be applied to data sets measured at two time points at least. In order to obtain a Boolean function or a state transition pair table of each node as output data (see Fig. 4b), a state transition pair table of the network (input data for the Boolean method) is formulated by summarising these input states and output states (see Fig. 4a).

Discretisation of mRNA expression levels makes Boolean methods useful when there are noisy input data [30]. We lose information during the discretisation of states, however, which may result in unrealistic inference outcomes. Moreover, we cannot always find an optimal inference result because of the computational complexity that grows exponentially according to the number of network nodes. To deal with this problem, the maximum number of arguments of each Boolean function is assumed to be bounded by some constant, and various algorithms, such as the REVEAL, BOOL-2 [31, 32], temporal Boolean [33], Discrete Function Learning [34], computational algebra [35, 36] and Probabilistic Boolean Network (PBN) algorithms [37], have been developed. The following is a review of the characteristic properties of each aforementioned algorithm.

REVEAL makes use of mutual information [38] in information theory as a measure of interrelationships [33, 34, 39–42]. For instance, consider a network with nodes having two discretised states (e.g. 0 or 1) and formulate the state transition pair table of the network as an input data. And then find the k arguments whose mutual information with the node x_i is identical to the self-information of the node x_i by starting from $k = 1$. If there is no such case for x_i then we repeat this procedure after increasing k

by 1. Finally, we can construct a state transition pair table of each node if we find such arguments for all nodes. REVEAL has been extended to cases of multiple discretised states [40, 41] and has formed a basis for applying information theory to reverse engineering of GRNs. Mutual information has been widely employed in reverse engineering of GRNs. For instance, mutual information is used in Bayesian methods by incorporating REVEAL into their package and a modified mutual information criterion is also used to overcome the difficulties in network learning [43]. In addition, data transmission theory, which asserts mutual information of indirect interaction is smaller than that of direct interaction, is used to identify the indirect relationships among the interrelated genes [41]. REVEAL has the disadvantage in that it requires an exhaustive search for all pair-wise mutual information by increasing the indegree (the number of arguments). One way of dealing with such a difficulty is to confine the search space into the set $S \cup \{x_j\}$ where S consists of k nodes having the highest mutual information with x_i when we increase the indegree of x_i from k to $k + 1$ [34]. Moreover, Zheng and Kwok [34] have extended REVEAL by allowing some error ranges in consideration of noise measurement. Alternatively, another extension of the Boolean method by considering the values of arguments not only at t but also at $t - 1, \dots, t - (T - 1)$ for a given node at $t + 1$ has been proposed [33] where T is an index representing the dependency of the algorithm on the time window. Other approaches developed to find Boolean functions using logical operations and algebraic theory instead of mutual information have been developed [32, 35, 36]. For instance, the BOOL-2 algorithm, using logical operations, was proposed to deal with experimental noise effects by considering only the Boolean functions of nodes which logically explain the influence of input states on corresponding nodes with probability of more than a threshold [31, 32]. However, this algorithm does not provide any practical guideline for selecting the threshold, which is critical to the identification results of the BOOL-2 algorithm.

Laubenbacher and Stigler [35, 36] have proposed the algorithms of searching for Boolean functions from a set of polynomial functions with discretised coefficients by applying computational algebraic theory. They have shown that Boolean functions can be represented by polynomial functions with the coefficients of 0 or 1 through translation of Boolean logical operators into algebraic operators (e.g. $x \vee y := x + y + xy$). They have then assigned to each node a polynomial function realising the relationship between the input state and the output state of each state transition pair by employing Lagrange interpolation or the Chinese Remainder Theorem [44]. They have further utilised the Buchberger algorithm [45] to eliminate any term of the constructed polynomial function which has 0 input state to obtain a final Boolean function. Hence, we can find Boolean functions using this algebraic algorithm without any exhaustive search; however, this algorithm is sensitive to noise because of the procedure of fitting input states to the output states.

There are probabilistic extensions of Boolean methods by considering many Boolean functions $f_{i_1}, f_{i_2}, \dots, f_{i_k}$ of each node x_i and the probabilities with which each Boolean function f_{i_j} is chosen to predict the state of x_i [37, 46, 47]. The PBN algorithm [46, 47] can account for the embedded uncertainty of data and models by allowing some error bounds in the Boolean functions. There are, however, too many parameters to be estimated (e.g. $2^8 \approx 10^{77}$ parameters only for eight nodes). Ching *et al.* [37] have proposed an extended PBN algorithm that reduces

the number of parameters to be estimated by making use of a homogeneous first-order discrete-time Markov chain and regression while keeping the advantages of the PBN algorithms [46, 47]. This algorithm, however, does not guarantee improved accuracy of the inference result.

3.2 Bayesian methods

Bayesian methods make use of the Bayes' rule to reverse engineer GRNs by inferring the causal relationship between two network nodes based on conditional probability distributions (CPDs) [48–58] and then use statistical theories with various types of biological data. Bayesian methods can be classified into static Bayesian and dynamic Bayesian methods depending on the use of temporal expression profiles for considerations of dynamics.

Let us consider static Bayesian methods first. Using these methods, we can infer a directed acyclic graph (DAG) and a CPD from the given expression data (refer to the network structure and probabilities at each node in Fig. 5). Note that the DAG means a directed graph without any loop. As these methods do not take account of temporal dynamics, they can be widely used for reverse engineering of other biomolecular networks based on static information [50, 59].

The procedure of inferring a DAG and CPD from an input data D is as follows. First, we assume either discrete variables (Fig. 5a) or continuous variables (Fig. 5b) for network nodes and further assume one of the possible DAGs. Next, let us consider a probabilistic model of each

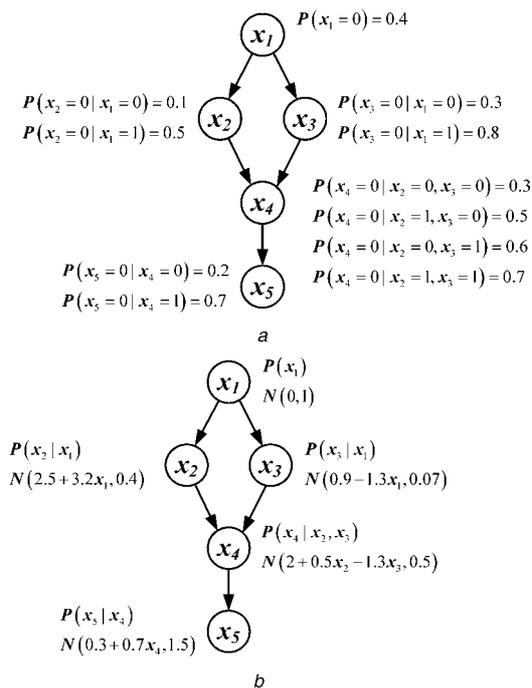


Fig. 5 Illustration of the static Bayesian method

a Inferred network composed of discrete nodes having multinomial distributions

Conditional probability $P(x_3 = 0 | x_1 = 1)$ means $P(x_3 = 0 \text{ and } x_1 = 1) / P(x_1 = 1)$

b Inferred network composed of continuous nodes having normal distributions

Conditional probability $P(x_4 | x_2, x_3) \sim N(2 + 0.5x_2 - 1.3x_3, 0.5)$ means 'the child variable x_4 is normally distributed around the mean $2 + 0.5\alpha - 1.3\beta$ with the variance 0.5 given its parents $x_2 = \alpha$, $x_3 = \beta$, which is computed using a linear regression model $P(x_4 | x_2, x_3) \sim N(a + bx_2 + cx_3, 0.5)$ for the child variable x_4 given its parents x_2, x_3

Here the variance 0.5 of the variable x_4 is independent of its parents x_2, x_3

child node (e.g. x_4 in Fig. 5) for the given parent nodes (e.g. x_2 and x_3 in Fig. 5) such as a multinomial distribution model (Fig. 5a) or a linear Gaussian model (Fig. 5b). Finally, we estimate the CPD of each node using the DAG and D . We make a score that describes the fitness of the DAG with respect to D using the estimated CPD [60]. The DAG and CPD with the highest scores become the inference outputs, and the procedures of obtaining those are called 'model selection (or structure learning)' and 'parameter learning', respectively [61]. A particular distribution over nodes must be assumed for parameter learning, although there is no specific guideline for this assumption [62]. Bayesian approaches and mutual information are often used for this to reflect the fitness of a DAG with the input data [63, 64]. For a continuous node variable, there is an additional computational complexity for integration in the scoring procedure, but a robust inference result can occur as all possible parameter values are probabilistically reflected in the scoring. In model selection, we need to confine the search space as there are so many possible DAGs (e.g. $O(n^{18}) \simeq 10^{18}$ DAGs for only ten nodes). For this purpose, heuristic approaches are usually employed and any further details on such learning procedures can be found in [65].

Friedman *et al.* have determined the statistical confidence of features (network properties of interest) between two nodes x and y [e.g. Markov relation features (x and y are parents of another node) or order relation features (x precedes y)] from input data D using a bootstrap method [66]: The input data D_i ($1 \leq i \leq m$) with the same number of samples as D are constructed by random selection and replacement from D , and an optimal DAG G_i ($1 \leq i \leq m$) is obtained by applying the learning procedure to the input data D_i . The confidence of each feature is computed by counting the frequency of G_i ($1 \leq i \leq m$) containing the feature. On the other hand, Pe'er *et al.* [67] have proposed an extended algorithm for the discrete network studied by Friedman *et al.* [48] such that detailed regulation types (activation or inhibition) can be inferred from the input data of perturbation experiments (e.g. gene deletion or over-expression, kinetic mutations and external treatments such as environmental stresses).

The static Bayesian methods have a critical limitation that any regulatory network containing a feedback loop (i.e. a directed cycle) cannot be inferred using these methods. As the feedback loop has a most important network feature that can cause homeostasis, other methods are needed to overcome this limitation and take account of the temporal dynamics. Dynamic Bayesian methods were developed from such motivations. Basically, these are simple extensions of the static Bayesian methods using time-series input data (see Fig. 6). Dynamic Bayesian methods also focus on the probabilistic causal relationship between two network nodes and assume that these relationships do not change over time like the static Bayesian methods. Dynamic Bayesian methods can, however, result in a better inference result as the dynamics of networks are reflected [56].

There are software toolkits for both the static Bayesian method and the dynamic Bayesian method: Mocapy Toolkit (<http://sourceforge.net/projects/mocapy>), Bayes Net Toolbox (BNT) [55] and Deal (<http://www.math.aau.dk/~dethlef/novo/deal>) [52]. Note that BNT employs the REVEAL algorithm for structure learning. Ong *et al.* [51] have constructed a Bayesian network structure using BNT and unravelled a transcriptional regulatory pathway through parameter learning. Kim *et al.* [58] have proposed to infer a GRN using a nonparametric regression model

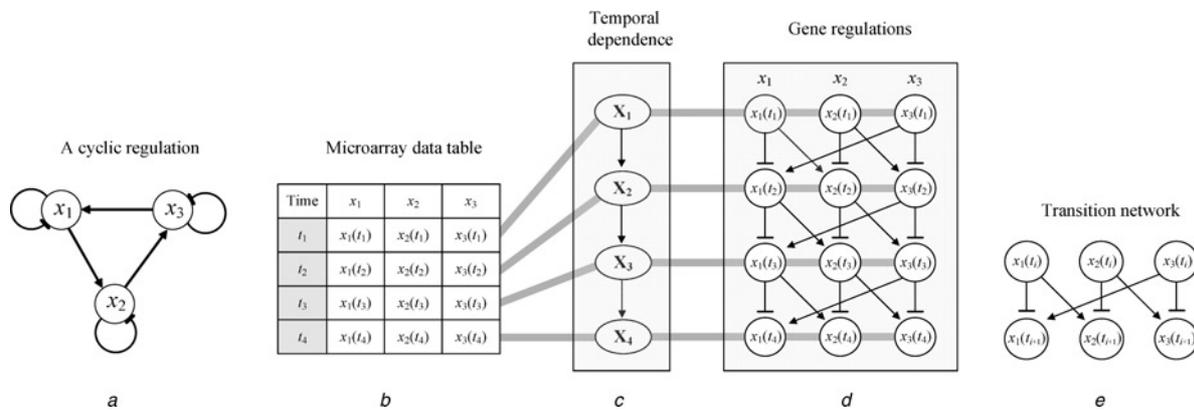


Fig. 6 Illustration of the dynamic Bayesian method

- a Cyclic regulatory network with three nodes
- b Microarray time-series data produced from (a)
- c First-order Markov relations with $X_i = (x_1(t_i), x_2(t_i), x_3(t_i))$ and $t_1 < t_2 < t_3 < t_4$
- d The network structure and the CPD of each node are assumed to be time-invariant
- e Transition network representing the causal relationship between X_i and X_{i+1} , which is to be learned [64]

and employing the dynamic Bayesian algorithm. Yu *et al.* [57] have proposed an influence scoring to infer detailed regulatory types (activation or inhibition) and the relative magnitude of node interactions. They have further shown that the combination of the Bayesian Dirichlet equivalence (BDe) scoring metric based on Bayesian posteriori probability [68] and a greedy search algorithm results in best inference outputs among the combinations of the various scoring metrics (e.g. BDe, Bayesian information criterion) and search algorithms (e.g. greedy algorithm with random restarts, simulated annealing, genetic algorithm). Li and Chan [53] have reported that they have successfully inferred some subnetworks such as tricarboxylic acid and urea cycles by combining several Bayesian methods. Recently, Zou and Conzen [54] have investigated a new dynamic Bayesian algorithm in consideration of time-lag effects.

3.3 Regulation matrix methods

In general, a GRN can be represented by an ordinary differential equation $d\mathbf{x}(t)/dt = \mathbf{f}(\mathbf{x}(t))$ or a discrete-time equation $\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t))$, where $\mathbf{x}(t) = (x_1(t), \dots, x_n(t))$ is a vector of nodes $x_i (1 \leq i \leq n)$ representing gene expression levels at time t and $\mathbf{f} = (f_1, \dots, f_n)$ is a vector-valued function from the real n -dimensional space \mathbb{R}^n into \mathbb{R}^n . The function \mathbf{f} can be a linear [69–73], piecewise linear [2, 74], pseudo-linear (i.e. composite function of a sigmoid function and a linear function) [75, 76] or continuous (differentiable) nonlinear function such as a power-law function represented by S-systems [77]. Detailed descriptions of various models can be found in [2, 78]. Note that the discrete-time equation corresponds to the Boolean model if \mathbf{f} takes its value from $\{0,1\}^n$. There was also another development employing partial differential equations to describe the spatial location [2]. As most bio-molecular networks, in general, have nonlinear dynamics, we should consider nonlinear models, but this causes much more difficulties in estimating parameters from the limited number of data samples. This is the reason why no effective inference method has been suggested yet in such direction and many reverse engineering methods infer the regulatory relationship by solving a linear (more exactly, linearised) system instead of directly inferring the nonlinear function \mathbf{f} . Hence, in this section, we focus on those methods based on linear systems, categorised as regulation matrix methods. In regulation matrix methods, we

want to find a solution $A = (a_{ij})$ of the linear system $\tilde{y}_i = b_i + \sum_{j=1}^n a_{ij}\tilde{x}_j (1 \leq i \leq n)$ derived from $d\mathbf{x}(t)/dt = \mathbf{f}(\mathbf{x}(t))$ or $\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t))$. Here b_i , \tilde{x}_i and \tilde{y}_i are directly computed from experimental data (\tilde{x}_i and \tilde{y}_i can have various forms; see Sections 3.3.1 and 3.3.2), and a_{ij} denotes $\partial f_i / \partial x_j$ or $(\partial f_i / \partial x_j) / (-\partial f_i / \partial x_i)$ which are the regulatory relationships of x_i on x_j (i.e. $\partial(dx_i/dt)/\partial x_j$ or $\partial x_i / \partial x_j$). Thus, if $a_{ij} > 0$, x_j activates x_i by enhancing (the net rate of) the production of x_i ; if $a_{ij} < 0$, x_j inhibits x_i by reducing (the net rate of) the production of x_i and $a_{ij} = 0$ implies that x_j has no regulatory relation on x_i . In this respect, the matrix A is called a regulation matrix (see Sections 3.3.1 and 3.3.2).

Although the Boolean method assumes discretised expression levels, regulation matrix methods directly make use of continuous expression levels without loss of any information caused by discretisation. Moreover, contrary to the Bayesian methods, which are based on probabilistic concepts, regulation matrix methods do not rely on such probabilistic notions, but instead make use of linear algebra such as linear regression, principal components analysis, singular value decomposition (SVD), Gaussian elimination and so on. This means that regulation matrix methods can infer GRNs in a more quantitative manner than the Boolean methods if proper data measurements are conducted. Using regulation matrix methods, we can also estimate the strength of interactions. On the other hand, if given data contain many noises, regulation matrix methods might result in poor inference results compared with the Boolean and Bayesian methods. Regulation matrix methods can be further classified according to the required data types: steady-state or time-series expression data (see Fig. 7 for an overall sketch).

3.3.1 Regulation matrix methods based on steady-state data:

Regulation matrix methods basically utilise the linear system $\tilde{y}_i = b_i + \sum_{j=1}^n a_{ij}\tilde{x}_j (1 \leq i \leq n)$ around a steady state. In general, steady-state measurements of gene expression levels are required before/after gene perturbations such as variations of temperature, pH and using plasmids. In some cases, parameters that indirectly influence a set of particular genes are perturbed instead of direct gene perturbations. We need to design a sophisticated perturbation experiment so that the characteristics of a GRN are well reflected in the steady-state data because it is impossible to infer the interaction relationships from a

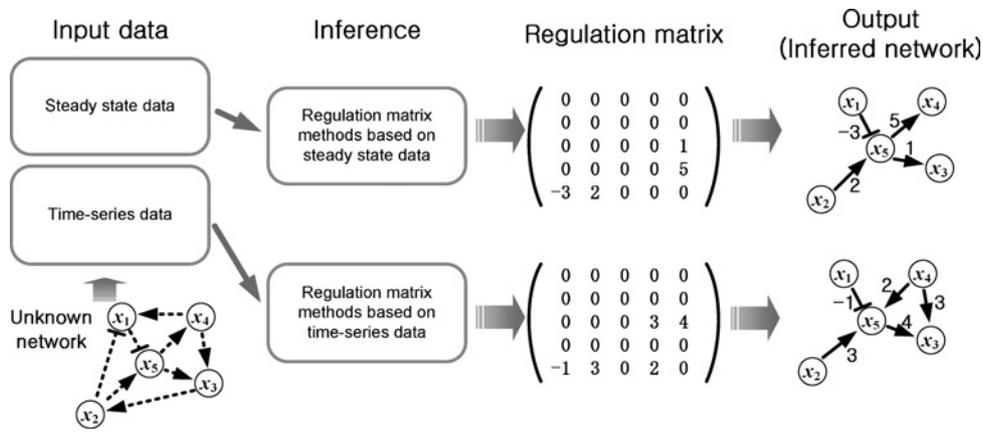


Fig. 7 Illustration of the regulation matrix methods based on either steady-state expression data or time-series data

single sampling at the steady state. We also note that the perturbation amounts can be critical because too large variations cannot be used for linearised models.

Several methods have been suggested to solve this linear equations using steady-state data and to obtain the regulation matrix A under different circumstances [72, 79–86] (see Fig. 8).

Yeung *et al.* [87] have proposed a method of utilising the estimate of $d(x_i - x_i^s)/dt$ obtained from linear interpolation without assuming $d(x_i - x_i^s)/dt = 0$. They have computed the regulation matrix by employing SVD [71, 88] to decompose the data matrix and regression to identify the sparsest network (Fig. 8c) [89]. Gardner *et al.* [72] have made use of the sparseness of a regulation matrix and assumed that the maximal number of nonzero elements at each row of A is k (the upper bound of the indegree of a gene). On the basis of this assumption, they reduced the number of variables to estimate from n^2 into kn . In other words, they converted the problem into an over-determined one and computed A using the multiple linear regression (Fig. 8d). An example of a small network composed of nine genes related to the SOS pathway was used for illustration of this method based on perturbation experiments with quantitative polymerase chain reaction. Tegnér *et al.* [82] have developed an algorithm for determining the least number of genes to be perturbed and applied the algorithm to the previously described method. Di Bernardo *et al.* [81] have also employed the ‘Forw-TopD-reest-K’ search algorithm [90] for reverse engineering of large-scale networks. The basic idea is to choose D optimal solutions for $k = 1$ and extend these solution networks by adding other connections (one by one) for each incremental change of k and choose the solution with the smallest error for $k = K$.

The aforementioned methods have a common difficulty of estimating the perturbation amount from the measured data after perturbation. To handle this difficulty, Kholodenko *et al.* [79] proposed another type of perturbation method – perturbing the parameters that indirectly affect the activity of some modules or sets of genes in a network. They have assumed a GRN that can be decomposed into modules x_i ($1 \leq i \leq n$) and a corresponding parameter p_k ($k \neq i$), and proposed a method of inferring a_{ij} ($i \neq j$) which satisfies $\Delta x_{i,k}^s / x_i^s(p_k) = \sum_{1 \leq j \leq n, j \neq i} a_{ij} \Delta x_{j,k}^s / x_j^s(p_k)$ ($1 \leq k \leq n$, $k \neq i$) based on steady-state data $x_i^s(p_k)$ (before perturbation) and $x_i^s(p_k + \Delta p_k)$ (after perturbation) where $\Delta x_{i,k}^s = x_i^s(p_k + \Delta p_k) - x_i^s(p_k)$ (Fig. 8e). This method has advantages in that the amount of perturbation does not need to be measured and a subnetwork structure affecting only selected modules of interest can be inferred. This method, however, requires a priori information on

parameters that indirectly affect each module and does not take any experimental noise into consideration. Compared to the previous methods of constraining the upper bound of indegree k [72, 81, 82], this method cannot be applied to a large-scale network as we need to perturb as many parameters as the number of network nodes. A similar method was developed for metabolic control analysis [83, 91] and Andrec *et al.* [84] have extended the foregoing method of Kholodenko *et al.* [79] by considering experimental noise under a normal distribution. They have not applied, however, this method to authentic experimental data.

3.3.2 Regulation matrix methods based on time-series data: The steady-state data obtained after perturbation does not usually reflect the dynamical characteristics of the gene regulatory system, but we can capture these characteristics through time-series data. To make use of such time-series data for reverse engineering of GRNs, we compute the regulation matrix over these time-series data.

Assuming a GRN model of $\mathbf{x}(t+1) = A\mathbf{x}(t)$, we will review the methods of computing A [71, 92–94]. van Someren *et al.* [92] have proposed a method scaling down the plausible solution network space by preprocessing the expression data and clustering them, where preprocessing means thresholding (i.e. choosing only the genes with significant variations) and normalisation of the data. After preprocessing the data, they have reduced the size of a network to be inferred by clustering the genes with Euclidean distance, and hence converted the under-determined problem of finding the solution A of $\mathbf{x}(t+1) = A\mathbf{x}(t)$ into an over-determined problem. In this way, they have inferred the network of the prototypes (i.e. representatives) of the clusters (Fig. 8f). This method overcame the dimensionality problem which means a difficulty (under-determined problem) in reverse engineering GRNs because of the large scale of the networks with a little data, but it can only reveal the regulatory network among those prototype clusters and not between individual genes.

Next, let us assume a GRN model of $\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$ and consider those methods of computing A from time-series data [69–71, 73, 75, 76, 95–98]. Chen *et al.* [69] assumed a GRN model of $\dot{\mathbf{r}}(t) = C\mathbf{p}(t) - V\mathbf{r}(t)$, $\dot{\mathbf{p}}(t) = L\mathbf{r}(t) - U\mathbf{p}(t)$ and represented the inferred GRN as $\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$, $\mathbf{x} = (\mathbf{r}, \mathbf{p})$ where \mathbf{r} and \mathbf{p} denote the concentration vectors of mRNA and protein, respectively. They employed an approximated difference equation and made use of the sparseness of GRNs to compute the regulation matrix A . They also suggested a method to compute C using the Fourier transform for stable systems. This method infers a GRN in consideration of both transcription and translation using the concentration

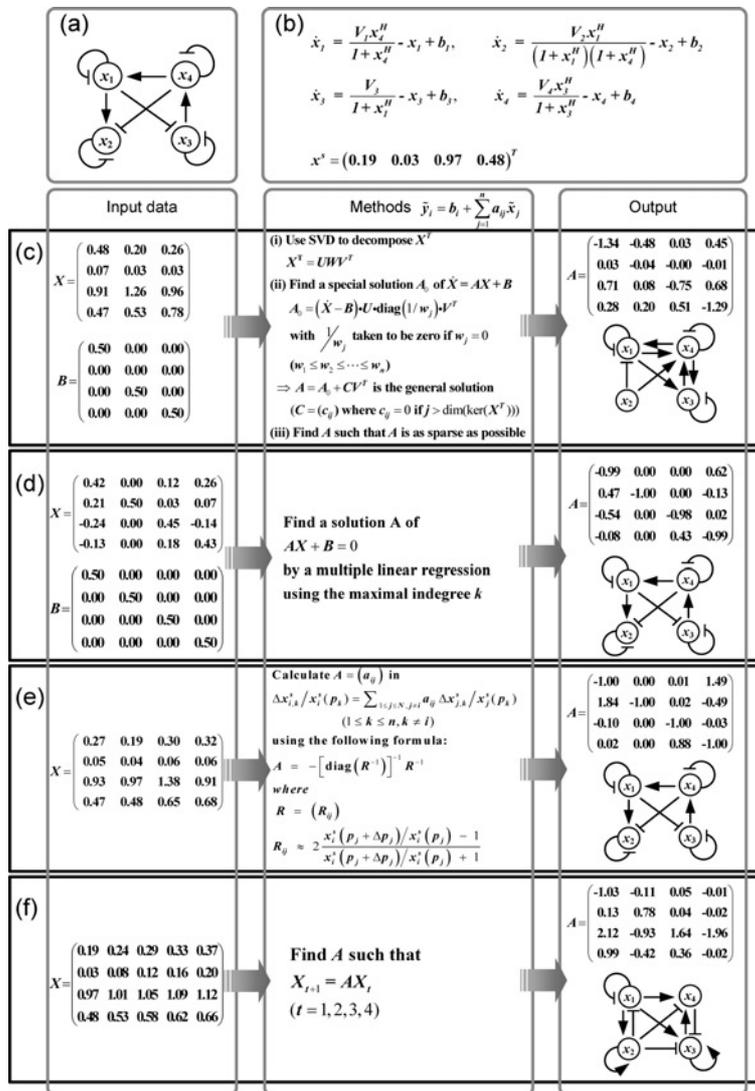


Fig. 8 Illustrative examples of the regulation matrix methods

a Graph of the artificial network

b Mathematical model to generate input data with $H = 2$, $V_1 = V_2 = V_3 = V_4 = 1$. b_i ($1 \leq i \leq 4$) are detailed in the description of each method j th Entry x_j^s of the column vector x^s is the measured expression level of the node x_j before perturbations

c Yeung *et al.*'s [87] method: the j th entry of the i th column vector of X is the measured expression level of the node x_j at $t = 2$ after the perturbation of (b_1, b_2, b_3, b_4) which corresponds to the i th column vector of B ($1 \leq i \leq 3, 1 \leq j \leq 4$)

In this method, \tilde{y}_i and \tilde{x}_j represent $d(x_i(t) - x_i^s)/dt$ and $x_j(t) - x_j^s$, respectively

Derivatives are calculated using the mean ratios of changes

d Gardner *et al.*'s [72] method: the j th entry of the i th column vector of X is the measured steady-state expression level of the node x_j at $t = 100$ after the perturbation of (b_1, b_2, b_3, b_4) which corresponds to the i th column vector of B ($1 \leq i, j \leq 4$)

In this method, \tilde{y}_i and \tilde{x}_j represent $d(x_i(t) - x_i^s)/dt$ and $x_j(t) - x_j^s$, respectively

e Kholodenko *et al.*'s [79] method: the j th entry of the i th column vector of X is the measured steady-state expression level of the node x_j at $t = 100$ after the parameter perturbation $V_i = 1.2$ ($1 \leq i, j \leq 4$)

In the method, $\Delta x_{i,k}^s$ denotes the difference of $x_i^s(p_k + \Delta p_k) - x_i^s(p_k)$

All the other notations are detailed in the main text

In (d) and (e), we used four times bigger perturbations to infer the network and thereby we could correctly infer the network using the methods

f van Someren *et al.*'s [92] method: the j th entry of the i th column vector X_t of X is the expression level of the node x_j at $t = 0.1(i - 1)$ after the perturbation $(b_1, b_2, b_3, b_4) = (0.5, 0.5, 0.5, 0.5)$ ($1 \leq i \leq 5, 1 \leq j \leq 4$)

In (c)–(f), we used 0.1 as the threshold for significant relations

profiles of both proteins and mRNAs as input data. This method was proposed at the early stage of a GRN studies and was based on differential equation models. But this method lacks detailed information on experimental design such as sampling time intervals. There is an extension of this method which considers time delays in $\dot{x}(t) = Ax(t)$ [95]. Another extension has been made by considering a non-linear model $\dot{x}(t) = f(x(t))$ with a specific form of $f(x)$ and introducing a genetic algorithm for parameter estimation [76].

The aforementioned methods have assumed time-invariant regulatory relationships among genes as they

have used the linear system near steady states (i.e. the regulation matrix A has been assumed as a constant). Sontag *et al.* [99] have relaxed this assumption and proposed a method of inferring the regulation matrix $A(t)$ at each time t . They have assumed a GRN model of $\dot{x}_i(t) = f_i(x(t), p)$ ($1 \leq i \leq n$) where p is a set of parameters, and developed a method of inferring $A(t) = (a_{ij}(t))$ by making use of the time-series data $x_j(t, p_{i_k})$, $x_j(t, p_{i_k} + \Delta p_{i_k})$ ($1 \leq j \leq n$) measured before and after, respectively, the perturbation of a specific parameter p_{i_k} ($1 \leq k \leq n$) that indirectly affects a node x_i . In this case,

$a_{ij}(t)$ denotes the solution of the linear system $(R_{ip_{ik}}(t + \Delta t) - R_{ip_{ik}}(t))/\Delta t = \sum_{1 \leq j \leq N} a_{ij}(t)R_{jp_{ik}}(t) (1 \leq k \leq n)$ with $R_{jp_{ik}}(t) = x_j(t, p_{ik} + \Delta p_{ik}) - x_j(t, p_{ik})$. This method does not require a small perturbation as it is not based on a linearised model near steady states. Moreover, it does not require measuring the perturbation amount Δp_{ik} and can be applied to reverse engineering of a subnetwork around a specific module of interest as the method of Kholodenko *et al.* [79]. Furthermore, this method provides the information on temporal variation of regulatory relationships. This method, however, also requires as many parameter perturbations as the number of network nodes and therefore cannot be applied to reverse engineering of a large-scale network. Cho *et al.* [100] have expounded on the fundamental concepts of the methods proposed by Kholodenko *et al.* [79] and Sontag *et al.* [99], and presented a comprehensive unified framework. The basic idea was that n independent equations are required to uniquely solve the system with n unknowns, and these n linearly independent equations can be obtained by a properly chosen set of n parameter perturbations. Cho *et al.* [101], however, have also proposed a very simple but effective reverse engineering method based on the temporal ascending or descending slope information from given time-series measurements instead of computation through the measured absolute values.

4 Reverse engineering based on both gene expression profiles and biological information

In the previous section, we reviewed the reverse engineering methods that use only expression profiles and exposed the fundamental limitations to such methods related to dimensionality, computational complexities and data uncertainties. Hence, for better inference results, it is necessary to adopt additional information processes. In this regard, this section will review the reverse engineering methods based not only on expression profiles, but also on available biological information such as a TF binding DNA sequence (a sequence motif shortly mentioned), which is a 5' upstream sequence of genes recognised by a common TF, gene function annotations (e.g. gene *lacZ* encodes protein β -galactosidase) [63], ChIP-chip data [102] and protein-protein interaction data (see Input data in Fig. 9). We can also acquire various inference outputs by integrating these different types of biological information.

There have been extensive studies on reverse engineering of GRNs by finding sequence motifs. Tavazoie *et al.* [103] have applied k -means clustering algorithm (a popular technique for clustering given data into k partitions) to cell-cycle time-series data and investigated the clusters of similarly expressed genes under a particular growth condition. They identified the function and the sequence motif of the genes within the same cluster using the MIPS

(Munich Information Center for Protein Sequence) category and the AlignACE (Aligns Nucleic Acid Conserved Elements) program [104], respectively, and then found the transcriptional sub-network regulated by the known TF recognising this sequence motif. It is difficult to correctly identify the whole set of regulators of genes only from identification of the sequence motif of these genes as we cannot determine the chromatin modification based only on sequence information. Moreover, this method cannot explain the combinatorial regulation of TFs. To solve this problem, we need the information based on which we can choose sequence motifs involved in combinatorial regulation. To obtain the information, Beer and Tavazoie [105] have employed time-series data and diverse biological information as input data: upstream DNA sequences of 800 bp of genes in the 5' direction, the space between two sequence motifs, the distance between a sequence motif and ATG (the starting point of genes), the orientation of genes (the right direction or the left direction of genes in the gene map) and the order of the genes in the chromosomes. As a result, they were able to explain the combinatorial regulation through clustering and inference of the relations between sequence motifs and gene expression patterns.

There also have been attempts to infer a GRN by identifying regulators and the relationships of these regulators that control the mRNA expression levels. Segal *et al.* [106] have determined the set of the triples (a module, a set of regulators and a regulation tree) using a Bayesian score [65] where a module is a set of functionally coherent genes, regulators are the controllers of the module and a regulation tree is a tree composed of regulators for its nodes. The input data are mRNA expression data and candidate regulatory genes (known and putative TFs and signal transduction molecules). To analyse the biological meaning of the obtained triples, each triple was tested for enrichment of the sequence motifs and gene annotations. They have applied the method to the following input data: a set of 173 *Saccharomyces cerevisiae* gene expression microarrays data and candidate regulator genes based on the *Saccharomyces* Genome Database and Yeast Proteome Database. Using this approach, we can infer which modules are regulated under what stress conditions and can further determine the genes that regulate a specific process under a given condition of experiments.

The aforementioned clustering-based methods can, however, result in many false-positives because of indirect influences included during clustering similarly expressed genes and searching sequence motifs. To overcome such difficulty, a genome-wide location analysis based on ChIP-chip data [102] has been proposed to identify all possible target genes that can bind a given TF [102, 107–115]. Bar-Joseph *et al.* [107] have proposed the following

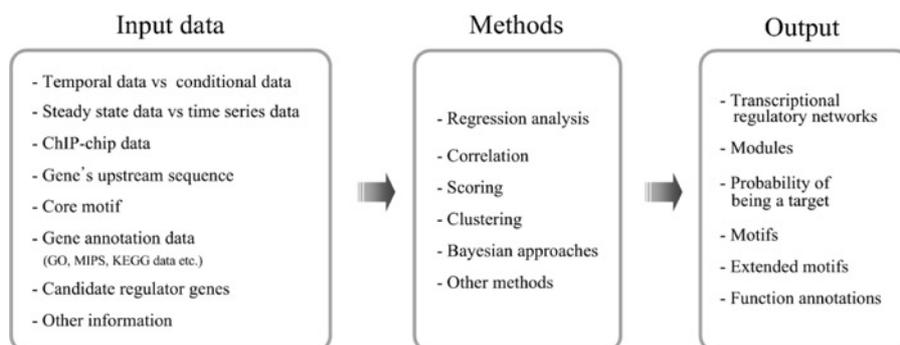


Fig. 9 Reverse engineering methods based on biological information

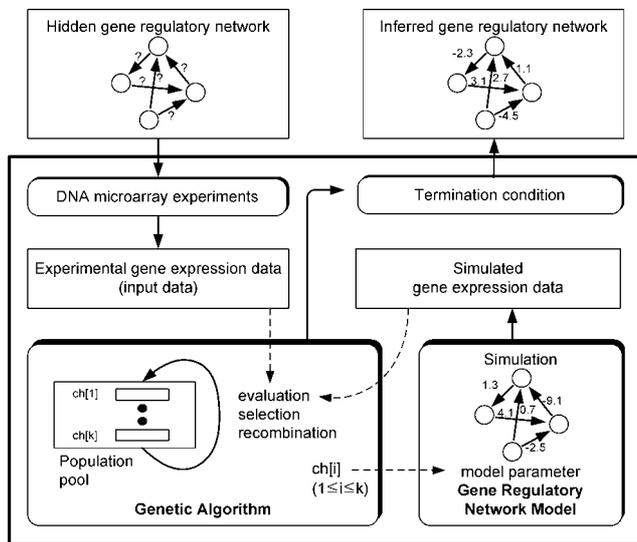


Fig. 10 Overview of reverse engineering of GRNs by GAs

GRAM algorithm using protein–DNA binding data and mRNA expression data as input data: First, all possible combinations of transcriptional regulators are constructed using the binding data. Although there are potentially an exponential number of combinations, the combinations of interest are quite limited. This is because GRAM selects a gene g and then chooses the combinations T_1, \dots, T_k of transcriptional regulators which bind to the selected gene g . Next, for each T_i , GRAM finds a collection C_i of genes to which all the transcriptional regulators in T_i commonly binds. Then, GRAM finds a gene c_i in C_i that have a ‘core’ expression profile and construct a set S of genes in C_i which show highly correlated expressions with c_i . Through relaxation of the binding criterion, GRAM identifies additional genes which are similarly expressed with c_i and to which all the transcriptional regulators in T_i commonly bind. GRAM extends S by adding similarly expressed genes. A module M_i can be obtained by such a repeated extension through relaxation and the function of M_i can be categorised by MIPS. GRAM chooses again a gene which is not included in $\bigcup_{1 \leq i \leq k} M_i$ and repeat the previous steps. In order to accurately identify the binding site of a TF, however, the ChIP–chip data must be obtained using the activated TF [109]. An extension of this method involves multivariate regression techniques to estimate the binding affinity of each TF at a promoter (the region recognised by RNA polymerase for the gene transcription). This extension infers target regulatory genes through the correlation between this binding affinity and gene expression profiles [108] and also through the correlation between DNA binding profiles and gene expression patterns [116].

Most of the previous studies on searching target genes regulated by a TF are based on the clustering of expression data [102, 103, 105–108, 110–115, 117–120]. These methods cannot be applied to the expression data measured at only one time point. Moreover, we note that many genes having the sequence motif recognised by a TF are not direct target genes of the TF. To handle these problems, Wang *et al.* [109] have proposed another method (MODEM algorithm) with the input data: a ChIP–chip data or TF perturbation experimental data measured at only one time point, a core motif (a sequence motif of about 6–8 bp length) recognised by the TF and promoter sequences of genes. An extended core motif of 12 bp was obtained from a position-specific frequency matrix. On the basis of the joint probability of the extended core motif and the

expression data, they computed the probability of a gene being a target gene of the TF. This method is advantageous using single time point data from a single experiment and also not requiring clustering to look for target genes. Moreover, we can search the overlapped part in several sequence motifs using the extended motif and thereby also reveal that several TFs work competitively in the same sequence motif. This method, however, still cannot distinguish direct and indirect target genes even though they have used sequence motifs as the procedure of finding sequence motifs depends on expression values measured at one time point.

GRNs are complicated, and experimental data sets have intrinsic errors and diverse biological information. Thus, in order to handle the errors and use the diverse biological information, it is necessary to integrate various types of biological data [59, 121–123]. In this regard, Lee *et al.* [59] made use of various heterogeneous functional genomics data: mRNA expression data (Stanford Microarray Database), gene context data [Rosetta stone data (gene-fusion data), phylogenetic profiles], experimental protein–protein interaction data (database of interacting proteins, mass spectrometry, yeast two-hybrid assay, synthetic lethal assay data), literature minig data (Medline abstracts) and five benchmark sets (Kyoto-based KEGG database, Gene Ontology annotation, the cluster orthologous group annotation, the yeast protein localisation data generated from genome-wide GFP-tagging and microscopy, MIPS). They have obtained a *S. cerevisiae* network consisting of gene–gene linkages with scores (e.g. a linkage between open reading frame (ORF) YLR206W and ORF YLR290C with a log likelihood score 8.826), where benchmark sets were used to test the correct assignment of linkages. Only some links represented direct protein–protein interaction and the other linkages represented probabilistic functional associations. The score of each linkage was calculated with respect to four categories (co-expression, co-citation, protein sequence comparison, protein interaction) based on the Bayesian approach. Using these scores, each linkage has been assigned a score through a unified scoring scheme. Higher scores mean more confident linkages. We can incorporate new biological information to increase the confidence of linkages through this approach.

5 Machine learning approaches

In contrast to the previous methods based on logical analysis or mathematical developments, this section reviews the reverse engineering methods employing machine learning techniques such as genetic algorithms, genetic programming, neural networks and fuzzy logic. Machine learning techniques have been used not only for inference algorithms [75, 124–132], but also for the clustering of gene expression data [133–136] and the modelling of GRNs [137, 138]. Of particular importance, genetic algorithms (GAs) and genetic programming (GP) have been widely used to reconstruct GRNs and the clustering of gene expression data, whereas neural networks (NNs) have often been used for modelling GRNs. Recently, NNs and fuzzy logic algorithms were fused to form a cooperative framework for the inference of GRNs [135, 136].

In GA-based inference methods, we assume that a GRN is represented by a mathematical model whose parameters are to be estimated from gene expression data using GAs. The estimation of model parameters is achieved by evolution of the chromosomes in a population pool. GAs have

been typically used for this evolutionary procedure as they are widely known as robust and systematic optimisation tools applicable to general scientific and engineering problems [139–141]. GAs evolve the chromosomes (in this application, candidate GRNs) by means of evaluating their fitness, selecting parents for a next generation and doing crossover as well as mutation operations. Therefore if any types of problems are represented by chromosomes (typically, represented by bit strings), GAs can efficiently find a (sub)optimal solution within predefined generations. Owing to this feature, GAs have mainly been applied to functional or combinatorial optimisation problems (e.g. bit pattern matching problems) and parameter estimation problems. In parameter estimation problems, the parameters are encoded to chromosomes, and GAs find the optimal estimates by evolving the chromosomes in a population pool within a predefined resolution depending on the encoding scheme. Wahde and Hertz [75] and Repsilber *et al.* [124] have used GAs to estimate the parameters of GRN models from both artificial and experimental microarray data. Iba and Mimura [132] also employed GAs to reconstruct GRNs from time-series gene expression data, and in particular, introduced an exon (an active link that can have any value) and an intron (an inactive link fixed to zero) in encoding chromosomes of GAs. The number of exons was limited to approximately 5 for each population pool (they maintained several population pools depending on the constraints of the exons and introns) to achieve better estimates in a rapid manner. Their methods also provided an interactive platform to infer GRNs through human intervention under graphic user interface environments. Similar methods have been developed with different modelling frameworks of GRNs [125–128].

Fig. 10 illustrates an overview of reverse engineering of GRNs through parameter estimation of GAs. In Fig. 10, experimental gene expression data obtained from DNA microarray experiments are presented as input data. The chromosomes of GAs in a population pool encode the parameters of a GRN model. The network model can be one of the Boolean networks [31], linear networks [71, 92–94], Bayesian networks [48], differential equations [69] and recurrent NNs [75, 142, 143]. Any other types of network models can also be applied if their parameters can be encoded into chromosomes. But the recent works [75] have mainly focused on the recurrent NN models. Like the other applications of GAs, there are no other constraints on the network models.

The initial values of encoded parameters are set to random numbers or predefined values from a priori knowledge. Note that each chromosome in the population pool represents one candidate GRN that can be obtained by decoding the chromosome. This candidate GRN produces a simulated gene expression data set and the GA uses these data to score the fitness of the corresponding chromosome by comparing the simulated data with the experimental gene expression data. All chromosomes in the population pool are evaluated in the same way and the chromosomes with higher fitness scores have more chances to be selected as parents for the next generation by a selection algorithm. The selected chromosomes are reproduced by crossover and mutation operations. GAs drive chromosome evolution over time and converge to an inferred GRN that corresponds to the inference output. The evolution process stops by a predefined criterion on the fitness value such that it results in a (sub)optimal inference result.

One of the key factors for successful reverse engineering is the selection of an appropriate GRN model. Wahde and Hertz [75] and Jung and Cho [128] have used recurrent

NN models; Iba and Mimura [132] and Kimura *et al.* [127] have used S-system models; Repsilber *et al.* [124] have used transition table models; Xiong *et al.* [125] have used linear structural equation models; Swain *et al.* [126] have used mutuality models. Which model is most appropriate to represent the dynamics of GRNs remains unknown.

We cannot guarantee the quality of an inference result obtained by the GA-based reverse engineering method even if the fitness value is very high and therefore the simulated gene expression data of an inferred GRN are very close to the experimental gene expression data. This shortcoming is because there can be many GRN models that can still generate similar gene expression data to the experimental gene expression data. Such difficulty originates from the fact that the number of samples in DNA microarray experiments is usually much smaller than the number of parameters desired for estimation – this issue is referred to as a ‘small sampling problem’. This problem can be alleviated by reducing the number of parameters through clustering of gene expression data or increasing the number of samples through interpolation [73]. The ‘small sampling problem’ is a major hurdle especially for reverse engineering of large-scale GRNs. In addition, the GA-based reverse engineering method cannot account for the noise effect introduced in microarray experiments.

There is an extension of the GA, termed GP, in which candidate solutions are represented by a tree structure [144]. This tree structure makes GP more adequate for estimating the structure or topology of a network than the parameter of a network [145]. Like the GAs, candidate solutions in GP (corresponding to the chromosomes in GAs) evolve through selection, crossover and mutation operations. GPs have been widely used not only for reverse engineering of GRNs, but also for unravelling metabolic networks. Ando *et al.* [129] have employed GPs for the estimation of network structures and used a least mean square method to determine the parameters. GPs produce similar inference results as GAs and also have similar limitations.

As mentioned previously, a major problem in reverse engineering of GRNs is the ‘small sampling problem’ which can be alleviated by reducing the number of parameters through clustering of gene expression data. The clustering of gene expression data is therefore an important preprocessing step for reverse engineering. Toronen *et al.* [133] have applied self-organising maps (SOMs) to the clustering of gene expression data and Huang *et al.* [134] have further extracted the relationship between clusters by using an artificial neural network. Kasabov [135] employed neuro-fuzzy style NNs, knowledge-based neural networks (KBNNs), for the classification of clusters and reverse engineering of GRNs. In KBNNs, ‘if-then’ rules for input–output relationships are extracted and this provides us with insights into the causal relationships of GRNs. Chan *et al.* [136] have used GAs for the selection of initial cluster centres for expectation maximisation (EM) clustering as the EM (a hill-climbing-like local optimiser) results in a very sensitive performance with respect to the initial cluster centres. As mentioned above, the NNs (especially SOMs) are useful tools for clustering and these are also often used for reverse engineering of GRNs.

NNs have been widely applied to the modelling of GRNs, clustering of gene expression data and reverse engineering of GRNs. Reinitz and Sharp [142] have introduced a recurrent NN model of GRNs in the form of $\tau_i \dot{x}_i(t) = g(b_i + \sum_{j=1}^n a_{ij}x_j(t)) - x_i(t)$ where g is an activation

function, b_i is a bias level of x_i , and τ_i is time constant of x_i . In general, a sigmoid function $g(z) = (1 + e^{-z})^{-1}$ is employed for the activation function to account for the saturation effects. Vohradsky [137] has proposed another recurrent NN model as $\tau_i \dot{x}_i(t) = f_i(\theta_i + \sum_{j=1}^n a_{ij}x_j(t)) - x_i(t)$ where f_i is a nonlinear transfer function and θ_i is the external input to the gene x_i . These two models are similar with each other and represent nonlinear models for GRNs. Alternatively, a simplified linear model $\dot{x}_i(t) = p_i + \sum_{j=1}^n a_{ij}x_j(t)$ has recently been adopted and applied by Gardner and Faith [146] where p_i denotes an external perturbation.

In summary, we note that the machine learning techniques have been broadly applied to reverse engineering of GRNs. GAs and GPs have been most popularly employed for parameter estimation in reverse engineering, whereas NNs have been used for the clustering of gene expression data and the modelling of GRNs (sometimes for reverse engineering). Recently, neuro-fuzzy style NNs have been brought into reverse engineering studies. Sokhansanj *et al.* [138] introduced a linear fuzzy gene network model that represents a set of fuzzy ‘if-then’ rules for GRNs. These fusion approaches help to alleviate the fundamental limitations of the previously reviewed reverse engineering methods. The most recent concept of an artificial genome (AG) [147] – emulating gene expression mechanisms upon artificial chromosomes – may provide a strong lead for resolving current problems in reverse engineering. Thus, machine learning techniques can be expected to improve continuously and be incorporated into the applications of reverse engineering methodology in systems biology.

6 Summary

In this paper, we have provided our review on the methods for reverse engineering of GRNs. The reverse engineering methods based only on expression profiles were considered in Section 3 and those utilising expression profiles plus biological information were revisited in Section 4. In Section 5, we considered machine learning approaches and explored in considerable detail, an estimation of parameters in the mathematical model of a GRN. As most of the proposed methods have different advantages and disadvantages, we recognise a need to improve our understanding of the fundamental idea for each method and to consider available input data and constraints in choosing an appropriate reverse engineering method.

The reverse engineering methods based on discretised states of expression profiles, such as the Boolean methods, are useful to capture simplified interaction structures. But these methods undergo a loss of information caused by discretisation. Despite such difficulties, these methods can be useful for certain cases when accuracy in describing the system of interest is not great. For instance, de Magalhaes and Toussaint [30] applied the Boolean method to infer a GRN related to human aging, which is complex and poorly understood, with the aim of investigating anti-aging intervention.

Bayesian methods are also useful in representing causal relationships. We can avoid over-fitting using Bayes’ rule even if expression data contain noise and uncertainties [50]. Moreover, we can use heterogeneous information as input data for Bayesian methods [63]. For example, we can infer a GRN by computing the log likelihood scores of such variable input data as gene expression, gene fusion, phylogenetic profiles, gene annotation and protein interaction [59]. As more high-throughput

heterogeneous data become available, these integrative methods, based on the Bayesian approach, will receive increasing attention.

The regulation matrix methods can avoid information loss caused by discretisation and can infer a GRN in a relatively rapid way by employing algebraic computations. These methods require large amounts of experimental data for a large network. To deal with this problem, GRN is usually assumed to be composed of sparse connections resulting in a sparse structure of the regulation matrix. There is, however, another drawback of these methods in that inference results can be strongly affected by noise in the data. One means to overcome this limitation is to use a threshold for each gene expression level to filter out noise effects and to choose those genes of significant variations prior to actually applying the regulation matrix methods for reverse engineering. Nonetheless, these methods have not been successful in reverse engineering of a large-scale network because of insufficient data. Insufficient data are often supplemented by in numero data, but this does not improve the performance of inference. As the data insufficiency commonly occurs in many other reverse engineering methods based on expression profiles, the methods of integrating additional biological information, such as sequence motifs, are increasingly being studied.

Available biological information includes DNA sequences, ChIP–chip data and functional annotation data. As such heterogeneous information can be incorporated into reverse engineering methods the inferred network can be interpreted from various biological viewpoints. The extraction of most useful information from the integrated heterogeneous data, however, still requires further investigation. Both spatial and temporal variations need to be considered in reverse engineering, which at present is not the case.

When compared to the logical inference based on rigorous mathematical frameworks, the reverse engineering methods employing machine learning approaches are less restrictive as these can always result in some suboptimal inference output for given input data based on a presumed network model and without further assumptions or constraints. The machine learning methods (e.g. GA), however, usually require extensive computation time due to the required evolution of parameter values, and cannot guarantee the quality of inference results. These methods do not provide us with a reproducible inference output and we cannot take account of the noise characteristics of input data in these methods. This latter problem can be resolved, however, by introducing the NNs and fuzzy inference schemes that have been developed recently for a more robust inference with noisy input data.

7 Acknowledgments

This work was supported by a grant from the Korea Ministry of Science and Technology (Korean Systems Biology Research Grant, M10503010001-05N030100111), by the 21C Frontier Microbial Genomics and Application Center Program, Ministry of Science and Technology (Grant MG05-0204-3-0), Republic of Korea, by a Grant from NITR/KOREA FDA for the National Toxicology Program in Korea (KNTP) and in part by 2005-B0000002 from the Korea Bio-Hub Program of Korea Ministry of Commerce, Industry & Energy. K.-H. Cho, H.-S. Choi and J. Kim were supported by the second-stage Brain Korea 21 Project in 2006.

8 References

- 1 Brazhnik, P., de la Fuente, A., and Mendes, P.: 'Gene networks: how to put the function in genomics', *Trends Biotechnol.*, 2002, **20**, pp. 467–472
- 2 de Jong, H.: 'Modeling and simulation of genetic regulatory systems: a literature review', *J. Comput. Biol.*, 2002, **9**, pp. 67–103
- 3 D'Haeseleer, P., Liang, S., and Somogyi, R.: 'Genetic network inference: from co-expression clustering to reverse engineering', *Bioinformatics*, 2000, **16**, pp. 707–726
- 4 van Someren, E.P., Wessels, L.F., Backer, E., and Reinders, M.J.: 'Genetic network modeling', *Pharmacogenomics*, 2002, **3**, pp. 507–525
- 5 Baldi, P., and Hatfield, G.W.: 'DNA microarrays and gene expression' (Cambridge University Press, 2002) Ch. 8
- 6 Crampin, E.J., Schnell, S., and McSharry, P.E.: 'Mathematical and computational techniques to deduce complex biochemical reaction mechanisms', *Prog. Biophys. Mol. Biol.*, 2004, **86**, pp. 77–112
- 7 Alwine, J.C., Kemp, D.J., and Stark, G.R.: 'Method for detection of specific RNAs in agarose gels by transfer to diazobenzoyloxymethyl-paper and hybridization with DNA probes', *Proc. Natl. Acad. Sci. USA*, 1977, **74**, pp. 5350–5354
- 8 Leung, Y.F., and Cavalieri, D.: 'Fundamentals of cDNA microarray data analysis', *Trends Genet.*, 2003, **19**, pp. 649–659
- 9 Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P., and Trent, J.M.: 'Expression profiling using cDNA microarrays', *Nat. Genet.*, 1999, **21**, pp. 10–14
- 10 van Hal, N.L., Vorst, O., van Houwelingen, A.M., Kok, E.J., Peijnenburg, A., Aharoni, A., van Tunen, A.J., and Keijer, J.: 'The application of DNA microarrays in gene expression analysis', *J. Biotechnol.*, 2000, **78**, pp. 271–280
- 11 Lipshutz, R.J., Fodor, S.P., Gingeras, T.R., and Lockhart, D.J.: 'High density synthetic oligonucleotide arrays', *Nat. Genet.*, 1999, **21**, pp. 20–24
- 12 Binder, H., Preibisch, S., and Kirsten, T.: 'Base pair interactions and hybridization isotherms of matched and mismatched oligonucleotide probes on microarrays', *Langmuir*, 2005, **21**, pp. 9287–9302
- 13 Wang, X., Ghosh, S., and Guo, S.W.: 'Quantitative quality control in microarray image processing and data acquisition', *Nucleic Acids Res.*, 2001, **29**, pp. E75–E85
- 14 Yang, Y.H., and Speed, T.: 'Design issues for cDNA microarray experiments', *Nat. Rev. Genet.*, 2002, **3**, pp. 579–588
- 15 Kerr, M.K., and Churchill, G.A.: 'Statistical design and the analysis of gene expression microarray data', *Genet. Res.*, 2001, **77**, pp. 123–128
- 16 Park, T., Yi, S.G., Lee, S., and Lee, J.K.: 'Diagnostic plots for detecting outlying slides in a cDNA microarray experiment', *Biotechniques*, 2005, **38**, pp. 463–471
- 17 Tseng, G.C., Oh, M.K., Rohlin, L., Liao, J.C., and Wong, W.H.: 'Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects', *Nucleic Acids Res.*, 2001, **29**, pp. 2549–2557
- 18 Chen, Y., Dougherty, E.R., and Bittner, M.: 'Ratio-based decisions and quantitative analysis of cDNA microarrays images', *Biomed. Opt.*, 1997, **2**, pp. 313–314
- 19 Stormo, G.D.: 'DNA binding sites: representation and discovery', *Bioinformatics*, 2000, **16**, pp. 16–23
- 20 Ren, B., Robert, F., Wyrick, J.J., Aparicio, O., Jennings, E.G., Simon, I., Zeitlinger, J., Schreiber, J., Hannett, N., Kanin, E., Volkert, T.L., Wilson, C.J., Bell, S.P., and Young, R.A.: 'Genome-wide location and function of DNA binding proteins', *Science*, 2000, **290**, pp. 2306–2309
- 21 Bulyk, M.L.: 'Computational prediction of transcription-factor binding site locations', *Genome Biol.*, 2003, **5**, 201 pp
- 22 Elkon, R., Linhart, C., Sharan, R., Shamir, R., and Shilo, Y.: 'Genome-wide in silico identification of transcriptional regulators controlling the cell cycle in human cells', *Genome Res.*, 2003, **13**, pp. 773–780
- 23 Gold, L., Brown, D., He, Y., Shtatland, T., Singer, B.S., and Wu, Y.: 'From oligonucleotide shapes to genomic SELEX: novel biological regulatory loops', *Proc. Natl. Acad. Sci. USA*, 1997, **94**, pp. 59–64
- 24 Klug, S.J., and Famulok, M.: 'All you wanted to know about SELEX', *Mol. Biol. Rep.*, 1994, **20**, pp. 97–107
- 25 Eddy, S.R.: 'Profile hidden Markov models', *Bioinformatics*, 1998, **14**, pp. 755–763
- 26 Stormo, G.D., Schneider, T.D., Gold, L., and Ehrenfeucht, A.: 'Use of the 'Perceptron' algorithm to distinguish translational initiation sites', *E. coli*, *Nucleic Acids Res.*, 1982, **10**, pp. 2997–3011
- 27 Remenyi, A., Scholer, H.R., and Wilmanns, M.: 'Combinatorial control of gene expression', *Nat. Struct. Mol. Biol.*, 2004, **11**, pp. 812–815
- 28 Black, M.A., and Doerge, R.W.: 'Calculation of the minimum number of replicate spots required for detection of significant gene expression fold change in microarray experiments', *Bioinformatics*, 2002, **18**, pp. 1609–1616
- 29 Baldi, P., and Hatfield, G.W.: 'DNA microarrays and gene expression' (Cambridge University Press, 2002), Ch. 2
- 30 de Magalhaes, J.P., and Toussaint, O.: 'How bioinformatics can help reverse engineer human aging', *Ageing Res. Rev.*, 2004, **3**, pp. 125–141
- 31 Akutsu, T., Miyano, S., and Kuhara, S.: 'Identification of genetic networks from a small number of gene expression patterns under the Boolean network model', *Pac. Symp. Biocomput.*, 1999, **4**, pp. 17–28
- 32 Akutsu, T., Miyano, S., and Kuhara, S.: 'Inferring qualitative relations in genetic networks and metabolic pathways', *Bioinformatics*, 2000, **16**, pp. 727–734
- 33 Silvescu, A., and Honavar, V.: 'Temporal Boolean network models of genetic networks and their inference from gene expression time series', *Complex Syst.*, 2001, **13**, pp. 54–75
- 34 Zheng, Y., and Kwok, C.K.: 'Reconstruction Boolean networks from noisy gene expression data'. Paper Presented at the International Conf. on Control, Automation, Robotics and Vision, 2004
- 35 Laubenbacher, R., and Stigler, B.: 'A computational algebra approach to the reverse engineering of gene regulatory networks', *J. Theor. Biol.*, 2004, **229**, pp. 523–537
- 36 Stigler, B.S.: 'Algebra approach to reverse engineering with application to biochemical networks' (Virginia Polytechnic Institute and State University, 2005)
- 37 Ching, W.K., Ng, M.M., Fung, E.S., and Akutsu, T.: 'On construction of stochastic genetic networks based on gene expression sequences', *Int. J. Neural Syst.*, 2005, **15**, pp. 297–310
- 38 Yeung, R.W.: 'A first course in information theory' (Kluwer Academic/Plenum Publishers, 2002)
- 39 Liang, S., Fuhrman, S., and Somogyi, R.: 'Reveal, a general reverse engineering algorithm for inference of genetic network architectures', *Pac. Symp. Biocomput.*, 1998, **3**, pp. 18–29
- 40 Ciccarese, P., Mazzocchi, S., Ferrazzi, F., and Sacchi, L.: 'GENIUS: a new tool for gene networks visualization'. Paper Presented at the European Conf. on Artificial Intelligence, 2004
- 41 Basso, K., Margolin, A.A., Stolovitzky, G., Klein, U., Dalla-Favera, R., and Califano, A.: 'Reverse engineering of regulatory networks in human B cells', *Nat. Genet.*, 2005, **37**, pp. 382–390
- 42 Andrecut, M., and Kauffman, S.A.: 'A simple method for reverse engineering causal networks', *J. Phys. A Math. Gen.*, 2006, **39**, L647–L655
- 43 Liu, T.-F., Sung, W.-K., and Mittal, A.: 'Learning multi-time delay gene network using Bayesian network framework'. Proc. 16th IEEE Int. Conf. on Tools with Artificial Intelligence, 2004, pp. 640–645
- 44 Yu, L.R., and Luo, L.B.: 'The generalization of the Chinese remainder theorem', *Acta Math. Sin.*, 2002, **18**, pp. 531–538
- 45 Abbott, J., Bigatti, A., Kreuzer, M., and Robbiano, L.: 'Computing ideals of points', *J. Symbolic Comput.*, 2000, **30**, pp. 341–356
- 46 Shmulevich, I., Dougherty, E.R., Kim, S., and Zhang, W.: 'Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks', *Bioinformatics*, 2002, **18**, pp. 261–274
- 47 Shmulevich, I., Dougherty, E.R., and Zhang, W.: 'From Boolean to probabilistic Boolean networks as models of genetic regulatory networks', *Proc. IEEE*, 2002, **90**, pp. 1778–1792
- 48 Friedman, N., Linial, M., Nachman, I., and Pe'er, D.: 'Using Bayesian networks to analyze expression data', *J. Comput. Biol.*, 2000, **7**, pp. 601–620
- 49 Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A., and Nolan, G.P.: 'Causal protein-signaling networks derived from multiparameter single-cell data', *Science*, 2005, **308**, pp. 523–529
- 50 Needham, C.J., Bradford, J.R., Bulpitt, A.J., and Westhead, D.R.: 'Inference in Bayesian networks', *Nat. Biotechnol.*, 2006, **24**, pp. 51–53
- 51 Ong, I.M., Glasner, J.D., and Page, D.: 'Modelling regulatory pathways in *E. coli* from time series expression profiles', *Bioinformatics*, 2002, **18**, (Suppl. 1), pp. S241–S248
- 52 Bötcher, S.G., and Dethlefsen, C.: 'deal: a package for learning Bayesian networks', *J. Stat. Software*, 2003, **8**
- 53 Li, Z., and Chan, C.: 'Inferring pathways and networks with a Bayesian framework', *FASEB J.*, 2004, **18**, pp. 746–748
- 54 Zou, M., and Conzen, S.D.: 'A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data', *Bioinformatics*, 2005, **21**, pp. 71–79
- 55 Murphy, K.: 'An introduction to graphical models' (Computer Science Division, UC Berkeley, 2001)
- 56 Hartemink, A.J.: 'Reverse engineering gene regulatory networks', *Nat. Biotechnol.*, 2005, **23**, pp. 554–555
- 57 Yu, J., Smith, V.A., Wang, P.P., Hartemink, A.J., and Jarvis, E.D.: 'Advances to Bayesian network inference for generating causal

- networks from observational biological data', *Bioinformatics*, 2004, **20**, pp. 3594–3603
- 58 Kim, S., Imoto, S., and Miyano, S.: 'Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data', *Biosystems*, 2004, **75**, pp. 57–65
- 59 Lee, I., Date, S.V., Adai, A.T., and Marcotte, E.M.: 'A probabilistic functional network of yeast genes', *Science*, 2004, **306**, pp. 1555–1558
- 60 Cooper, G.F., and Herskovits, E.: 'A Bayesian method for the induction of probabilistic networks from data', *Mach. Learn.*, 1992, **9**, pp. 309–347
- 61 Chen, X.W., Anantha, G., and Wang, X.: 'An effective structure learning method for constructing gene networks', *Bioinformatics*, 2006, **22**, pp. 1367–1374
- 62 Eddy, S.R.: 'What is Bayesian statistics?', *Nat. Biotechnol.*, 2004, **22**, pp. 1177–1178
- 63 Friedman, N.: 'Inferring cellular networks using probabilistic graphical models', *Science*, 2004, **303**, pp. 799–805
- 64 Göransson, L., and Koski, T.: 'Using a dynamic Bayesian network to learn genetic interactions', 2002. Technical Report, available online at: <http://www.mai.liu.se/~tikos/dynbayesian.pdf>. Accessed 16th April 2007
- 65 Heckerman, D.: 'A tutorial on learning with Bayesian networks' (Microsoft Research Advanced Technology Division, Microsoft Corporation, 1998)
- 66 Efron and Tibshirani: 'An introduction to the bootstrap' (Chapman & Hall: 1993)
- 67 Pe'er, D., Regev, A., Elidan, G., and Friedman, N.: 'Inferring subnetworks from perturbed expression profiles', *Bioinformatics*, 2001, **17**, (Suppl. 1), S215–S224
- 68 Friedman, N., Murphy, K., and Russell, S.: 'Learning the structure of dynamic probabilistic networks'. Paper Presented at the Fourteenth Conf. on Uncertainty in Artificial Intelligence, San Francisco, CA, 1998
- 69 Chen, T., He, H.L., and Church, G.M.: 'Modeling gene expression with differential equations', *Pac. Symp. Biocomput.*, 1999, **4**, pp. 29–40
- 70 de Hoon, M.J., Imoto, S., Kobayashi, K., Ogasawara, N., and Miyano, S.: 'Inferring gene regulatory networks from time-ordered gene expression data of *Bacillus subtilis* using differential equations', *Pac. Symp. Biocomput.*, 2003, **8**, pp. 17–28
- 71 D'Haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R.: 'Linear modeling of mRNA expression levels during CNS development and injury', *Pac. Symp. Biocomput.*, 1999, **4**, pp. 41–52
- 72 Gardner, T.S., di Bernardo, D., Lorenz, D., and Collins, J.J.: 'Inferring genetic networks and identifying compound mode of action via expression profiling', *Science*, 2003, **301**, pp. 102–105
- 73 Gutknecht, R., Moller, U., Hoffmann, M., Thies, F., and Topfer, S.: 'Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection', *Bioinformatics*, 2005, **21**, pp. 1626–1634
- 74 de Jong, H., Gouze, J.L., Hernandez, C., Page, M., Sari, T., and Geiselmann, J.: 'Qualitative simulation of genetic regulatory networks using piecewise-linear models', *Bull. Math. Biol.*, 2004, **66**, pp. 301–340
- 75 Wahde, M., and Hertz, J.: 'Coarse-grained reverse engineering of genetic regulatory networks', *Biosystems*, 2000, **55**, pp. 129–136
- 76 Deng, X., Geng, H., and Ali, H.: 'EXAMINE: a computational approach to reconstructing gene regulatory networks', *Biosystems*, 2005, **81**, pp. 125–136
- 77 Savageau, M.A.: 'Biochemical systems analysis. II. The steady-state solutions for an *n*-pool system using a power-law approximation', *J. Theor. Biol.*, 1969, **25**, pp. 370–379
- 78 Soule, C.: 'Mathematical approaches to differentiation and gene regulation', *CR Biol.*, 2006, **329**, pp. 13–20
- 79 Kholodenko, B.N., Kiyatkin, A., Bruggeman, F.J., Sontag, E., Westerhoff, H.V., and Hoek, J.B.: 'Untangling the wires: a strategy to trace functional interactions in signaling and gene networks', *Proc. Natl. Acad. Sci. USA*, 2002, **99**, pp. 12841–12846
- 80 Ideker, T.E., Thorsson, V., and Karp, R.M.: 'Discovery of regulatory interactions through perturbation: inference and experimental design', *Pac. Symp. Biocomput.*, 2000, **5**, pp. 305–316
- 81 Di Bernardo, D., Gardner, T.S., and Collins, J.J.: 'Robust identification of large genetic networks', *Pac. Symp. Biocomput.*, 2004, **9**, pp. 486–497
- 82 Tegnér, J., Yeung, M.K., Hasty, J., and Collins, J.J.: 'Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling', *Proc. Natl. Acad. Sci. USA*, 2003, **100**, pp. 5944–5949
- 83 de la Fuente, A., Brazhnik, P., and Mendes, P.: 'Linking the genes: inferring quantitative gene networks from microarray data', *Trends Genet.*, 2002, **18**, pp. 395–398
- 84 Andrec, M., Kholodenko, B.N., Levy, R.M., and Sontag, E.: 'Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy', *J. Theor. Biol.*, 2005, **232**, pp. 427–441
- 85 Cho, K.-H., Choi, H.-S., and Choo, S.-M.: 'Unraveling the functional interaction structure of a biomolecular network through alternate perturbation of initial conditions', *J. Biochem. Biophys. Methods*, 2007, in press
- 86 Schmidt, H., Cho, K.H., and Jacobsen, E.W.: 'Identification of small scale biochemical networks based on general type system perturbations', *FEBS J.*, 2005, **272**, pp. 2141–2151
- 87 Yeung, M.K., Tegner, J., and Collins, J.J.: 'Reverse engineering gene networks using singular value decomposition and robust regression', *Proc. Natl. Acad. Sci. USA*, 2002, **99**, pp. 6163–6168
- 88 Holter, N.S., Maritan, A., Cieplak, M., Fedoroff, N.V., and Banavar, J.R.: 'Dynamic modeling of gene expression data', *Proc. Natl. Acad. Sci. USA*, 2001, **98**, pp. 1693–1698
- 89 Gustafsson, M., Hornquist, M., and Lombardi, A.: 'Constructing and analyzing a large-scale gene-to-gene regulatory network – lasso-constrained inference and biological validation', *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 2005, **2**, pp. 254–261
- 90 van Someren, E.P., Wessels, L.F., Reinders, M.J., and Backer, E.: 'Searching for limited connectivity in genetic network models'. Paper Presented at the Proceedings of the International Conf. on Systems Biology, Pasadena, CA, 2001
- 91 de la Fuente, A., and Makhecha, D.P.: 'Unravelling gene networks from noisy under-determined experimental perturbation data', *Syst. Biol. (Stevenage)*, 2006, **153**, pp. 257–262
- 92 van Someren, E.P., Wessels, L.F., and Reinders, M.J.: 'Linear modeling of genetic networks from experimental data', *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 2000, **8**, pp. 355–366
- 93 Weaver, D.C., Workman, C.T., and Stormo, G.D.: 'Modeling regulatory networks with weight matrices', *Pac. Symp. Biocomput.*, 1999, **4**, pp. 112–123
- 94 Wessels, L.F., van Someren, E.P., and Reinders, M.J.: 'A comparison of genetic network models', *Pac. Symp. Biocomput.*, 2001, **6**, pp. 508–519
- 95 Dasika, M.S., Gupta, A., and Maranas, C.D.: 'A mixed integer linear programming (MILP) framework for inferring time delay in gene regulatory networks', *Pac. Symp. Biocomput.*, 2004, **9**, pp. 474–485
- 96 Mjolsness, E., Sharp, D.H., and Reinitz, J.: 'A connectionist model of development', *J. Theor. Biol.*, 1991, **152**, pp. 429–453
- 97 Cho, K.H., Kim, J.R., Baek, S., Choi, H.S., and Choo, S.M.: 'Inferring biomolecular regulatory networks from phase portraits of time-series expression profiles', *FEBS Lett.*, 2006, **580**, pp. 3511–3518
- 98 Kim, J., Bates, D.G., Postlethwaite, I., Heslop-Harrison, P., and Cho K.H.: 'Least-squares methods for identifying biochemical regulatory networks from noisy measurements', *BMC Bioinform.*, 2007, **8**, 8 pp
- 99 Sontag, E., Kiyatkin, A., and Kholodenko, B.N.: 'Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data', *Bioinformatics*, 2004, **20**, pp. 1877–1886
- 100 Cho, K.H., Choo, S.M., Wellstead, P., and Wolkenhauer, O.: 'A unified framework for unravelling the functional interaction structure of a biomolecular network based on stimulus-response experimental data', *FEBS Lett.*, 2005, **579**, pp. 4520–4528
- 101 Cho, K.H., Shin, S.Y., and Choo, S.M.: 'Unravelling the functional interaction structure of a cellular network from temporal slope information of experimental data', *FEBS J.*, 2005, **272**, pp. 3950–3959
- 102 Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., Zeitlinger, J., Jennings, E.G., Murray, H.L., Gordon, D.B., Ren, B., Wyrick, J.J., Tagne, J.B., Volkert, T.L., Fraenkel, E., Gifford, D.K., and Young, R.A.: 'Transcriptional regulatory networks in *Saccharomyces cerevisiae*', *Science*, 2002, **298**, pp. 799–804
- 103 Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., and Church, G.M.: 'Systematic determination of genetic network architecture', *Nat. Genet.*, 1999, **22**, pp. 281–285
- 104 Roth, F.P., Hughes, J.D., Estep, P.W., and Church, G.M.: 'Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation', *Nat. Biotechnol.*, 1998, **16**, pp. 939–945
- 105 Beer, M.A., and Tavazoie, S.: 'Predicting gene expression from sequence', *Cell*, 2004, **117**, pp. 185–198
- 106 Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N.: 'Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data', *Nat. Genet.*, 2003, **34**, pp. 166–176

- 107 Bar-Joseph, Z., Gerber, G.K., Lee, T.I., Rinaldi, N.J., Yoo, J.Y., Robert, F., Gordon, D.B., Fraenkel, E., Jaakkola, T.S., Young, R.A., and Gifford, D.K.: 'Computational discovery of gene modules and regulatory networks', *Nat. Biotechnol.*, 2003, **21**, pp. 1337–1342
- 108 Gao, F., Foat, B.C., and Bussemaker, H.J.: 'Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data', *BMC Bioinform.*, 2004, **5**, 31 pp
- 109 Wang, W., Cherry, J.M., Nochomovitz, Y., Jolly, E., Botstein, D., and Li, H.: 'Inference of combinatorial regulation in yeast transcriptional networks: a case study of sporulation', *Proc. Natl. Acad. Sci. USA*, 2005, **102**, pp. 1998–2003
- 110 Scott, M.S., Perkins, T., Bunnell, S., Pepin, F., Thomas, D.Y., and Hallett, M.: 'Identifying regulatory subnetworks for a set of genes', *Mol. Cell Proteomics*, 2005, **4**, pp. 683–692
- 111 Banerjee, N., and Zhang, M.Q.: 'Identifying cooperativity among transcription factors controlling the cell cycle in yeast', *Nucleic Acids Res.*, 2003, **31**, pp. 7024–7031
- 112 Qian, J., Lin, J., Luscombe, N.M., Yu, H., and Gerstein, M.: 'Prediction of regulatory networks: genome-wide identification of transcription factor targets from gene expression data', *Bioinformatics*, 2003, **19**, pp. 1917–1926
- 113 Ihmels, J., Friedlander, G., Bergmann, S., Sariq, O., Ziv, Y., and Barkai, N.: 'Revealing modular organization in the yeast transcriptional network', *Nat. Genet.*, 2002, **31**, pp. 370–377
- 114 Pilpel, Y., Sudarsanam, P., and Church, G.M.: 'Identifying regulatory networks by combinatorial analysis of promoter elements', *Nat. Genet.*, 2001, **29**, pp. 153–159
- 115 Berman, B.P., Nibu, Y., Pfeiffer, B.D., Tomancak, P., Celniker, S.E., Levine, M., Rubin, G.M., and Eisen, M.B.: 'Exploiting transcription factor binding site clustering to identify *cis*-regulatory modules involved in pattern formation in the *Drosophila* genome', *Proc. Natl. Acad. Sci. USA*, 2002, **99**, pp. 757–762
- 116 Gilchrist, M., Thorsson, V., Li, B., Rust, A.G., Korb, M., Kennedy, K., Hai, T., Bolouri, H., and Aderem, A.: 'Systems biology approaches identify ATF3 as a negative regulator of Toll-like receptor 4', *Nature*, 2006, **441**, pp. 173–178
- 117 Brazma, A., Jonassen, I., Vilo, J., and Ukkonen, E.: 'Predicting gene regulatory elements in silico on a genomic scale', *Genome Res.*, 1998, **8**, pp. 1202–1215
- 118 Bussemaker, H.J., Li, H., and Siggia, E.D.: 'Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis', *Proc. Natl. Acad. Sci. USA*, 2000, **97**, pp. 10096–10100
- 119 Bussemaker, H.J., Li, H., and Siggia, E.D.: 'Regulatory element detection using a probabilistic segmentation model', *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 2000, **8**, pp. 67–74
- 120 Li, H., Rhodius, V., Gross, C., and Siggia, E.D.: 'Identification of the binding sites of regulatory proteins in bacterial genomes', *Proc. Natl. Acad. Sci. USA*, 2002, **99**, pp. 11772–11777
- 121 Ideker, T., Thorsson, V., Ranish, J.A., Christmas, R., Buhler, J., Eng, J.K., Bumgarner, R., Goodlett, D.R., Aebersold, R., and Hood, L.: 'Integrated genomic and proteomic analyses of a systematically perturbed metabolic network', *Science*, 2001, **292**, pp. 929–934
- 122 Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung S., Emili, A., Snyder, M., Greenblatt, J.F., and Gerstein M.: 'A Bayesian networks approach for predicting protein–protein interactions from genomic data', *Science*, 2003, **302**, pp. 449–453
- 123 von Mering, C., Huynen, M., Jaeggi, D., Schmidt, S., Bork, P., and Snel, B.: 'STRING: a database of predicted functional associations between proteins', *Nucleic Acids Res.*, 2003, **31**, pp. 258–261
- 124 Repsilber, D., Liljenstrom, H., and Andersson, S.G.: 'Reverse engineering of regulatory networks: simulation studies on a genetic algorithm approach for ranking hypotheses', *Biosystems*, 2002, **66**, pp. 31–41
- 125 Xiong, M., Li, J., and Fang, X.: 'Identification of genetic networks', *Genetics*, 2004, **166**, pp. 1037–1052
- 126 Swain, M., Hunniford, T., Dubitzky, W., Mandel, J., and Palfreyman N.: 'Reverse-engineering gene-regulatory networks using evolutionary algorithms and grid computing', *J. Clin. Monit. Comput.*, 2005, **19**, pp. 329–337
- 127 Kimura, S., Ide, K., Kashihara, A., Kano, M., Hatakeyama, M., Masui, R., Nakagawa, N., Yokoyama, S., Kuramitsu, S., and Konagaya, A.: 'Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm', *Bioinformatics*, 2005, **21**, pp. 1154–1163
- 128 Jung, S.H., and Cho, K.-H.: 'Identification of gene interaction networks based on evolutionary computation' (Springer, 2005)
- 129 Ando, S., Sakamoto, E., and Iba, H.: 'Evolutionary modeling and inference of gene network', *Inform. Sci.*, 2002, **145** pp. 237–259
- 130 Eriksson, R., and Olsson, B.: 'Adapting genetic regulatory models by genetic programming', *Biosystems*, 2004, **76**, pp. 217–227
- 131 Sugimoto, M., Kikuchi, S., and Tomita, M.: 'Reverse engineering of biochemical equations from time-course data by means of genetic programming', *Biosystems*, 2005, **80**, pp. 155–164
- 132 Iba, H., and Mimura, A.: 'Inference of a gene regulatory network by means of interactive evolutionary computing', *Inform. Sci.*, 2002, **145**, pp. 225–236
- 133 Toronen, P., Kolehmainen, M., Wong, G., and Castren, E.: 'Analysis of gene expression data using self-organizing maps', *FEBS Lett.*, 1999, **451**, pp. 142–146
- 134 Huang, J., Shimizu, H., and Shioya, S.: 'Clustering gene expression pattern and extracting relationship in gene network based on artificial neural networks', *J. Biosci. Bioeng.*, 2003, **96**, pp. 421–428
- 135 Kasabov, N.K.: 'Knowledge-based neural networks for gene expression data analysis modelling and profile discovery', *Biosilico*, 2004, **2**, pp. 253–261
- 136 Chan, Z.H., Karabov, N., and Collins, L.J.: 'A two-stage methodology for gene regulatory network extraction from time-course gene expression data', *Expert Syst. Appl.*, 2006, **30**, pp. 59–63
- 137 Vohradsky, J.: 'Neural model of the genetic network', *J. Biol. Chem.*, 2001, **276**, pp. 36168–36173
- 138 Sokhansanj, B.A., Fitch, J.P., Quong, J.N., and Quong, A.A.: 'Linear fuzzy gene network models obtained from microarray data by exhaustive search', *BMC Bioinform.*, 2004, **5**, p. 108 pp
- 139 Holland, J.H.: 'Adaptation in natural and artificial systems' (MIT Press, 1992)
- 140 Goldberg, D.E.: 'Genetic algorithms in search, optimization, and machine learning' (Addison-Wesley, 1989)
- 141 Davis, L.: 'Handbook of genetic algorithms' (Van Nostrand Reinhold, 1991)
- 142 Reinitz, J., and Sharp, D.H.: 'Mechanism of eve stripe formation', *Mech. Dev.*, 1995, **49**, pp. 133–158
- 143 Takane, M.: 'Inference of gene regulatory networks from large scale gene expression data' (McGill University, Montreal, 2003)
- 144 Koza, J.R.: 'Genetic programming: on the programming of computers by means of natural selection' (MIT Press, 1992)
- 145 Wahde, M., and Hertz, J.: 'Modeling genetic regulatory dynamics in neural development', *J. Comput. Biol.*, 2001, **8**, pp. 429–442
- 146 Gardner, T.S., and Faith, J.J.: 'Reverse-engineering transcription control networks', *Phys. Life Rev.*, 2005, **2**, pp. 65–88
- 147 Quayle, A.P., and Bullock, S.: 'Modelling the evolution of genetic regulatory networks', *J. Theor. Biol.*, 2006, **238**, pp. 737–753