Reconstructing gene regulatory networks: from random to scale-free connectivity

J. Wildenhain and E.J. Crampin

Abstract: The manipulation of organisms using combinations of gene knockout, RNAi and drug interaction experiments can be used to reveal regulatory interactions between genes. Several algorithms have been proposed that try to reconstruct the underlying regulatory networks from gene expression data sets arising from such experiments. Often these approaches assume that each gene has approximately the same number of interactions within the network, and the methods rely on prior knowledge, or the investigator's best guess, of the average network connectivity. Recent evidence points to scale-free properties in biological networks, however, where network connectivity follows a power-law distribution. For scale-free networks, the average number of regulatory interactions per gene does not satisfactorily characterise the network. With this in mind, a new reverse engineering approach is introduced that does not require prior knowledge of network connectivity and its performance is compared with other published algorithms using simulated gene expression data with biologically relevant network structures. Because this new approach does not make any assumptions about the distribution of network connections, it is suitable for application to scale-free networks.

1 Introduction

The development of computational techniques to identify the transcription networks underlying observed gene expression patterns is an important challenge in the analysis of gene expression data. Significant progress has been made in the last few years in characterising regulatory interactions at the genomic level [1–9], including methods for identifying gene and protein interactions, regulatory modules occurring with high frequency in the genome and the identification of transcription motifs [10–13].

In parallel, several different approaches have been proposed by which gene regulatory networks can be identified directly from gene expression data sets [2, 14–16]. The aim of these so-called 'reverse-engineering' approaches is to allow investigators to analyse data sets directly without making prior assumptions about the underlying networks. Gene regulatory networks, identified purely from transcriptional data, reflect regulatory influences, rather than mapping direct physical interactions between molecules. Methods for gene network reconstruction have been proposed on the basis of statistical analyses such as Bayesian networks [17, 18], Boolean models [19] and graphical Gaussian models [15, 16]. In this report, we focus on analysing networks from gene perturbation experiments, analysed previously using singular value decomposition (SVD) [14, 20, 21] and a linear regression approach [2].

From the perspective of network reconstruction, one of the most important features of gene regulatory networks is that they are sparsely connected: the average number of connections per node in the network is small in comparison to the number of nodes [22]. Jeong *et al*. [23] analysed protein–protein interaction maps from *Saccharomyces cerevisiae* and estimated an average degree k_{av} per gene of 2.4. Guelzim *et al*. [24] showed for the yeast transcriptional network that the incoming and outgoing connections (in- and out-degrees) have a similar proportion in the genome. For this reason, progress can be made despite difficulties in achieving adequate coverage of all potential regulatory interactions in data sets (for example, there are typically fewer data points than interactions to be determined in gene expression data sets, especially for large-scale studies, including microarray approaches). Yeung *et al*. [14] have presented an approach based on SVD to infer the regulatory interactions, and they had success with limited amounts of data for networks of more than 200 genes. On the basis of this work, an improved approach to experimental design was developed, in which genes are selected iteratively for perturbation to reveal the architecture of the underlying network [20].

There is mounting evidence that many biological networks, including metabolic [25], protein–protein interaction [26] and transcriptional networks [27], as well as many other genomic indices [28], share the common property that the distribution of connections follows a power law, $P(k) \sim k^{-\gamma}$. Here the degree distribution $P(k)$ is the probability that a 'node' (gene) of the network is connected to exactly k other nodes. Networks with this property are known as scale-free networks, as the relative probability for two different connectivities k depends only on the ratio of the connectivities, rather than on their absolute values. Although the power-law degree distribution is the distinguishing feature of scale-free networks, few

IEE Proceedings online no. 20050092

doi:10.1049/ip-syb:20050092

Paper first received 15th November 2005 and in revised form 10th February 2006

J. Wildenhain was with the Bioinformatics Institute, School of Biological Sciences, University of Auckland, Private Bag 92019, Auckland, New Zealand and is now with Tyers Lab, Samuel Lunenfeld Research Institute, 600 University Avenue, Toronto, Ontario, Canada M5G 1X5

E.J. Crampin is with the Bioengineering Institute, Department of Engineering Science, University of Auckland, Private Bag 92019, Auckland, New Zealand

E-mail: e.crampin@auckland.ac.nz

reverse-engineering algorithms have considered this property, and several of the most prominent algorithms implicitly assume that networks are well characterised by their average network connectivity.

In this work, we consider the influence of the distribution of connections on network identification. Recently, Gardner *et al*. [2] described a reverse engineering algorithm based on a multiple regression approach, called NIR. Their approach requires that the average number of connections in the network be specified a priori, and that each node in the inferred network has this number of connections. This is clearly a limitation if biological networks are not well characterised by their average degree. Farina and Mogno [21] proposed the FAST algorithm that follows the SVD implementation by Yeung *et al*. [14], having a low computational complexity compared with the NIR algorithm.

Two new algorithms for reverse engineering gene regulatory networks are proposed in this paper, called simple-to-general (S2G) and general-to-specific (G2S) [29]. Both algorithms use an iterative model selection technique, described subsequently, to find the true underlying network. The two algorithms differ from one-another in that S2G iteratively builds up a network model, starting with no connections, whereas G2S starts with a fully connected network and sheds connections until an optimal network model is found. In either case, no assumptions need to be made about the network connectivity, distribution or average degree for the nodes, and it is this increased flexibility that makes these algorithms suitable for the identification of networks with more appropriate 'biological' degree distributions, such as scale-free networks.

Efforts to develop algorithms for identification of regulatory networks are, however, hindered by the quality and reliability of available data sets. In particular, benchmarking data sets on networks with known interactions are not readily available. For example, Gardner *et al*. [2] collected data for a sub-network of the SOS pathway in *Escherichia coli*, perturbing each gene in the sub-network in turn and recording the steady-state expression level. Using these data, Gardner *et al*. [2] applied the NIR algorithm and were able to infer many of the known interactions for what is a fairly well characterised pathway. However, the algorithm also identified a significant number of regulatory influences not previously recorded in the literature, which could correspond either to false positives or to the correct identification of previously unrecognised, yet genuine, regulatory interactions. This makes it difficult to use such data sets to assess the performance of their and other algorithms. In this situation, in silico modelling of gene networks can provide a platform by which to assess the performance of different algorithms [14, 20, 30–32]. In silico gene networks generate gene expression data with well-defined properties. The parameters of such artificial networks can be varied systematically, and the data sets obtained provide an objective comparison of reverse engineering algorithms on gene networks with different topologies, and with varying degrees of biological variation and measurement noise.

2 Methods

We developed a simulation environment similar to the one introduced by Mendes *et al*. [31]. This differential equation model, described subsequently, was used to generate steady-state gene expression data from perturbation experiments. To analyse the influence of network connectivity on the performance of the reverse engineering algorithms, we have used simulated data sets for random and scale-free networks of different sizes and noise levels in order to assess the performance of the existing and newly proposed algorithms.

2.1 Simulating gene expression data

Initially, we specify parameters defining the network connectivity in order to generate a network. Gene expression data can then be simulated for this network by specifying kinetic functions representing the regulatory interactions. In our model, the number of regulatory interactions of each gene is given by the degree, k, of the corresponding node, and the different network topologies correspond to different distributions of k across the nodes in the network. We simulated three types of networks: random [33], scale-free and hierarchical network topologies [34], shown in Fig. 1.

Random network. For a network of N genes, the random procedure connects each pair of nodes i and j with equal probability ζ, where the threshold for interaction ζ is equal to the average connectivity of the network, k_{av}.

Scale-free network. Following the work of Barabási and Albert [35], the scale-free topology is built using the preferential attachment rule

$$\zeta_i = \frac{k_i}{\sum_j k_j} \tag{1}$$

where k_i is the degree of node i and ζ_i the probability threshold for new interactions for gene i. Initially all nodes $i = 1, \ldots, N$ are assumed to have the same interaction probability. If a node acquires an interaction, the probability for new interactions increases according to (1). The algorithm selects two nodes randomly and tests against the probabilities $\zeta_i(k)$ and $\zeta_j(k)$. This procedure is terminated when the specified average connectivity for the network k_{av} is reached, which, providing it is well below fully connected, generates the power-law property.

Hierarchical network. In addition, we simulated hierarchical networks that are built by cloning a scale-free network and then adding connections between the clones, based on the probabilities $\zeta_i(k)$. These hierarchical networks retain the scale-free property.

For each network, the ratio of positive to negative connections is specified by the parameter r. If $r = 0.5$, then the probabilities for an edge to be activating or inhibiting are equal. Increasing r leads to more positive regulatory interactions, decreasing it to more negative interactions. Savageau [36] developed a theoretical basis that established some properties of either mode of regulation, but to our knowledge little research has been published about this proportion in genomes [37].

After defining the network structure, we generate large-scale expression data sets for the given topology, based on a model of genetic networks using ordinary differential equations (ODEs) [31]. Assuming that the levels of mRNA species are continuous and depend on the balance between transcription and degradation, (2) describes the general structure of the mathematical model

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = f_i(x_1, \ldots, x_N) - b_i x_i \tag{2}$$

where x_i represents the abundance of the mRNA of gene i, $f_i(x_1, \ldots, x_N)$ is the rate of transcription and b_i represents the breakdown rate of species i. The regulatory effects of activating and inhibitory genes are represented in the rate function f_i, which represents the influence of all regulatory

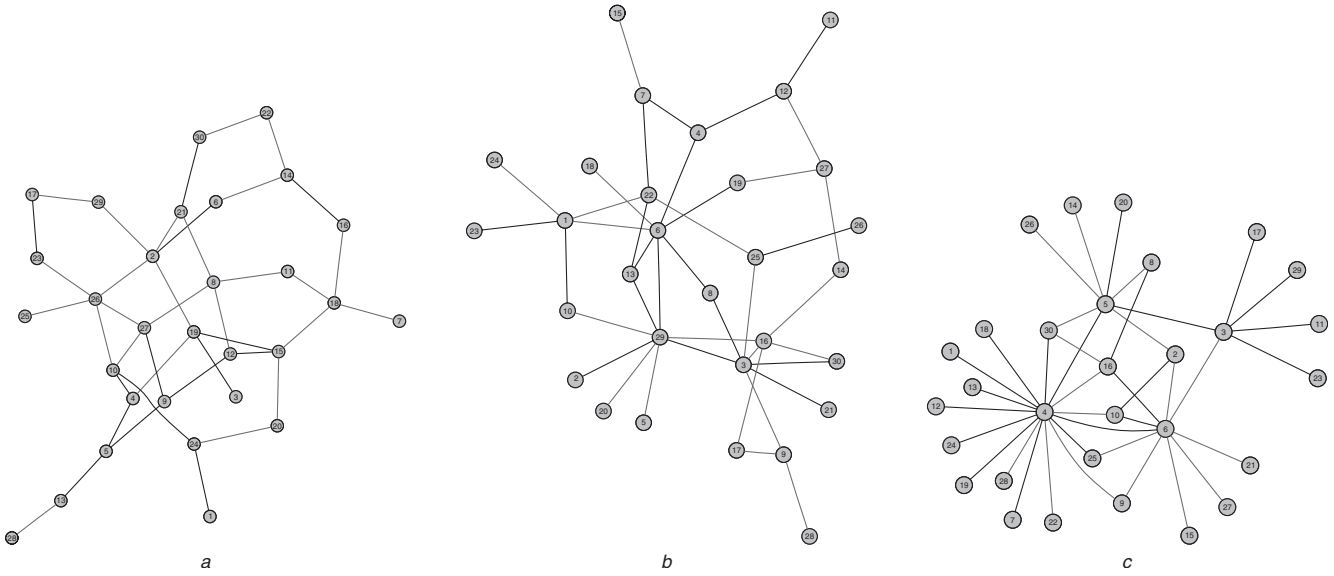

Fig. 1 *Networks of 30 nodes, with average degree $k_{av} = 3$*

a Random network topology ($k_{in} = 1.4$, $k_{out} = 1.5$)
b Scale-free network topology ($k_{in} = 1.5$, $k_{out} = 3$)
c Hierarchical network topology ($k_{in} = 1.5$, $k_{out} = 4.5$)

genes acting on gene i. Thus changes in the rate of transcription of each gene come about from changes in concentration of the other gene products. We assume that each gene exists only once in the network and transcription is catalysed by a limited number of transcription complexes. This characteristic leads to a general rate law of transcription

$$f_i(x_1, \ldots, x_N) = v_i \prod_j \left(\frac{\alpha_j^{m_j}}{I_j^{m_j} + \alpha_j^{m_j}} \right) \times \prod_j \left(\frac{1 + A_j^{m_j}}{A_j^{m_j} + \beta_j^{m_j}} \right) + u_i \quad (3)$$

where the term v_i symbolises the basal (non-inhibited) rate of expression and u_i represents an external perturbation. The regulatory interactions between genes consist of activation (A_j) or inhibition (I_j) of gene expression by transcription factors (activators and repressors), assumed proportional to gene transcript levels, with saturation constants α and β. The exponents m_j regulate the sigmoidicity of the interaction curve. These parameters are arbitrarily chosen from predefined ranges (Table 1). We generated large-scale expression data sets by simulating this ODE model, and checking the resulting data to guarantee a stable model.

Table 1: Parameter ranges for internal parameters used in the simulation model

Parameter	Description	Range
v_i	Basal rate of expression	$\boldsymbol{U}[1.05 \cdots 1.15]$
m_i	Hill coefficient	$\boldsymbol{U}[0 \cdots 6]$
α_i	Activator half-saturation constant	$\boldsymbol{U}[1 \cdots 2.5]$
β_i	Inhibitor half-saturation constant	$\boldsymbol{U}[1 \cdots 2.5]$
b_i	Degradation rate	$\boldsymbol{U}[0.75 \cdots 0.85]$

$\boldsymbol{U}[x \cdots y]$ denotes uniformly distributed random values in the given range. Biological variation in the kinetic parameters is determined by adding a uniformly distributed random value in the range $[-0.1 \cdots 0.1]$

2.2 Variability and noise

An important objective is to study the effects of variability on the performance of the algorithms. Two different sources are of major importance: biological variability and experimental noise. In biological systems, variability arises from genetic polymorphisms and from different environmental conditions. We simulated this by slightly varying the parameters v_i, b_i, α_i, β_i and m_i in the model between different simulated experiments. We simulated measurement error by adding Gaussian distributed noise to our simulation results, with zero mean and variance 0–100% according to the smallest estimated expression ratio of the final mRNA levels.

2.3 Steady-state perturbation experiments

Perturbation of the expression level of a gene will cause a change in the rate of expression of other genes in the regulatory network via their regulatory interactions, which may subsequently settle down into a new steady-state expression profile. To identify this system, a reverse engineering algorithm tries to infer the connections from the measurements obtained as a response to the perturbations made to the network. The identification of nonlinear behaviour is very challenging because of the increased complexity necessary to identify the nonlinear functional interactions. However, a simpler approach is to identify connections only by considering response to small perturbations from steady state, for which the behaviour of the network can be approximated by a linear system of equations. In this case, and in practice often for larger perturbations, the deviation from steady state $y_i = x_i - x_i^{ss}$ is well approximated by the linear equation

$$\frac{dy_i}{dt} = \sum_{j=1}^{N} a_{ij} y_j, \quad i = 1, \ldots, N \quad (4)$$

The 'connectivity matrix' a_{ij} represents the influence of gene j on gene i, and its entries are non-zero only when gene j acts

directly on gene i in the network. If a sustained external perturbation u_{il} is applied to the system, then a new steady state for each mRNA species may be established. The steady state resulting from the lth perturbation made to a network of N genes and M perturbation experiments is given by the following equation

$$0 = \sum_{j=1}^{N} a_{ij} y_{jl} + u_{il} + \epsilon_{il}, \quad i = 1, \ldots, N, \quad l = 1, \ldots, M \tag{5}$$

where y_{jl} is the steady state mRNA concentration for gene j following the perturbation in experiment l and ϵ_{il} represents the error term of the particular measurement. Our task is to identify the $N \times N$ coefficients a_{ij} (the connectivity matrix) describing the regulatory interactions between the genes; in particular, to identify the non-zero elements and their signs to distinguish between activating and inhibitory influences.

In practice, relative gene expression levels $\hat{y}_{il}$ are measured, where $\hat{y}_{il}$ is the ratio of the steady-state expression y_i under perturbation and the reference (unperturbed) steady state for gene i in experiment l

$$\hat{y}_{il} = \frac{y_{il}}{x_i^{ss}} = \frac{x_{il}}{x_i^{ss}} - 1 \tag{6}$$

Under this linear transformation of the data, the model that we wish to fit to the relative expression level data set is

$$0 = \sum_{j=1}^{N} \hat{a}_{ij} \hat{y}_j + \hat{u}_i \equiv \boldsymbol{E}_i, \quad i = 1, \ldots, N \tag{7}$$

where $\hat{\boldsymbol{y}}_j$ and $\hat{\boldsymbol{u}}_i = \boldsymbol{u}_i / x_i^{ss}$, are vectors of length M over the different experiments, $\hat{a}_{ij} = a_{ij} x_j^{ss} / x_i^{ss}$, which does not change the locations or the signs of the non-zero entries of the connectivity matrix, and $\boldsymbol{E}_i$ is the model reconstruction error (residual) for gene i over the experiments. We used model (2) and (3) to generate steady-state expression data $\hat{\boldsymbol{Y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N]$ for perturbations $\hat{\boldsymbol{U}} = [\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \ldots, \hat{\boldsymbol{u}}_N]$, and we applied different reverse engineering algorithms to infer the connectivity matrix and assessed their relative performance for different network parameters.

2.4 Comparison of reverse engineering algorithms

We used the NIR and FAST algorithms as described in the publications of Gardner *et al.* [2] and Farina and Mogno [21]. Additionally, two new algorithms have been adapted from the analysis of biochemical pathways [29]. In general, at least N independent measurements are required to uniquely determine the coefficients of the connectivity matrix from (7). Following Gardner *et al.* [2], in this work, we have assumed that each gene in the network is perturbed in turn, $M = N$ and $\hat{\boldsymbol{U}}$ is a diagonal matrix.

Given the perturbations $\hat{\boldsymbol{u}}_i$ and the steady-state expression data $\hat{\boldsymbol{y}}_i$, the coefficients $\hat{a}_{ij}$ can be found for each gene i in turn using a maximum likelihood approach. Assuming independent and normally distributed measurement errors, this reduces to least squares minimisation

$$C_{LS} = \frac{1}{M} \boldsymbol{E}_i \cdot \boldsymbol{E}_i \tag{8}$$

and can be solved using SVD. Typically, however, this approach will lead to over-fitting of the model to the data, including the noise. NIR and FAST get around this problem by assuming that each gene has around the same number of interactions, but as we have noted, this is inconsistent with the degree distributions found in many biological networks.

2.4.1 Simple-to-General and General-to-Specific: An alternative approach is to introduce a penalty term to the log-likelihood expression to prevent over-fitting, and hence find a sparse matrix a_{ij}. The algorithms S2G and G2S do not impose the number of interactions, k, a priori. Rather, they attempt to find a parsimonious model, using the Akaike Information Criterion (AIC) [38] to restrict the number of terms

$$C_{AIC}(K) = M \log \frac{1}{M} \boldsymbol{E}_i^{(K)} \cdot \boldsymbol{E}_i^{(K)} + 2K \tag{9}$$

where the number of terms in the model K is to be determined by the algorithm, from the data. This cost function seeks to minimise the least squares error and the second term penalises the use of an increasing number of interactions. (An alternative cost function, Rissanen's Minimum Description Length [39] based on minimising the coding length of a model and associated residual errors, could also be used here.)

We expect only a subset of the coefficients $\hat{a}_{ij}$ to be non-zero. Our model is then

$$\boldsymbol{E}_i^{(K)} = \sum_{k=1}^{K} \hat{a}_{ik} \hat{\boldsymbol{y}}_{\phi(k)} + \hat{\boldsymbol{u}}_i, \quad i = 1, \ldots, N \tag{10}$$

for a network with $K < N$ interactions, where $\phi(k) \in \{1, \ldots, N\}$ are the indices for the K interactions included in the model. The question is how to find K, and then which subset of K interactions to choose from the set of N possible interactions. An iterative procedure for constructing the model is proposed by Judd and Mees [40], who analysed the effect of adding and removing terms from the model. They showed that the term that can be added to increase the model size, giving the largest marginal improvement to the model approximation, is the element with largest absolute value in

$$\boldsymbol{\mu}_i = -\boldsymbol{V}^{T} \cdot \boldsymbol{E}_i^{(K)} \tag{11}$$

which is the projection of the model reconstruction error onto the matrix $\boldsymbol{V}$, the model design matrix, which is the set of all possible interactions. Similarly, they showed that the term that can be removed from the model doing the least damage to the approximation corresponds to the smallest coefficient $\hat{a}_{ij}$ [40]. Two iterative algorithms based on adding or removing interactions to improve the approximation are described in the Appendix. The S2G approach starts with a single interaction and iteratively considers terms to add to the model in order to find the set of interactions which minimises (9). Alternatively, the G2S approach uses as the initial set a fully connected graph and then applies (11) and (9) to eject terms until the optimal model is constructed.

2.4.2 Combining results using voting: Although G2S and S2G are built from the same iterative procedure, because of their different selection approaches we have found that the two algorithms tend to identify different networks. A simple voting scheme between the two algorithms was found to improve the performance. In the voting scheme, the networks indentified independently by the two algorithms are compared, and only those connections common to both networks are retained. This was found to decrease the number of false positive connections identified, with only a small impact on the false negative rate, thus increasing the performance of the approach.

3 Results

We measured the performance of the reverse engineering algorithms on simulated datasets with different network sizes and parameters. The NIR and FAST approaches required that the average network connectivity k_{av} was passed to the algorithms along with the steady-state data sets. An alternative strategy for NIR was suggested by di Bernardo *et al*. [7] where a maximal connectivity k_{max} is provided to the algorithm, and statistically insignificant edges subsequently pruned from the inferred network – we have not implemented this idea here. In addition, we compared the performance of the algorithms to a random allocation of gene–gene interactions, to calculate a threshold for selection of interactions by chance. We estimated statistics for randomly inferred connections for the given network parameters with respect to the topology, average number of connections k_{av}, network size N and the ratio r of repressing to activating interactions.

To compare the performance of the different algorithms we calculated the sensitivity (Sn: true positive rate, or 'power') and the false discovery rate (FDR), defined in the Appendix. Specificity (Sp: the true negative rate) is not a useful statistic here as it will be strongly affected by changing network size when comparing algorithm performance with different numbers of genes. In addition, we calculated the total number of errors (false positive count and false negative count).

Topology. Fig. 2 shows the performance of the four reverse engineering algorithms on sparse networks with random, scale free and hierarchical network topologies for different network sizes, along with the results of voting between the S2G and G2S approaches, and inference by chance. Simulated steady-state gene expression profiles were generated using networks with the desired topology and network size, and the gene regulatory networks inferred by each reverse engineering algorithm were compared with the network used to generate the data set. We used 50 independently generated networks at each data point in order to assess the performance of the reverse engineering algorithms.

S2G and G2S algorithms perform fairly uniformly across the different network topologies, and the sensitivity (true positive rate) is insensitive to network size for gene networks of 10–50 genes. In contrast, the NIR approach shows significantly worse performance for scale-free and hierarchical topologies than for random networks.

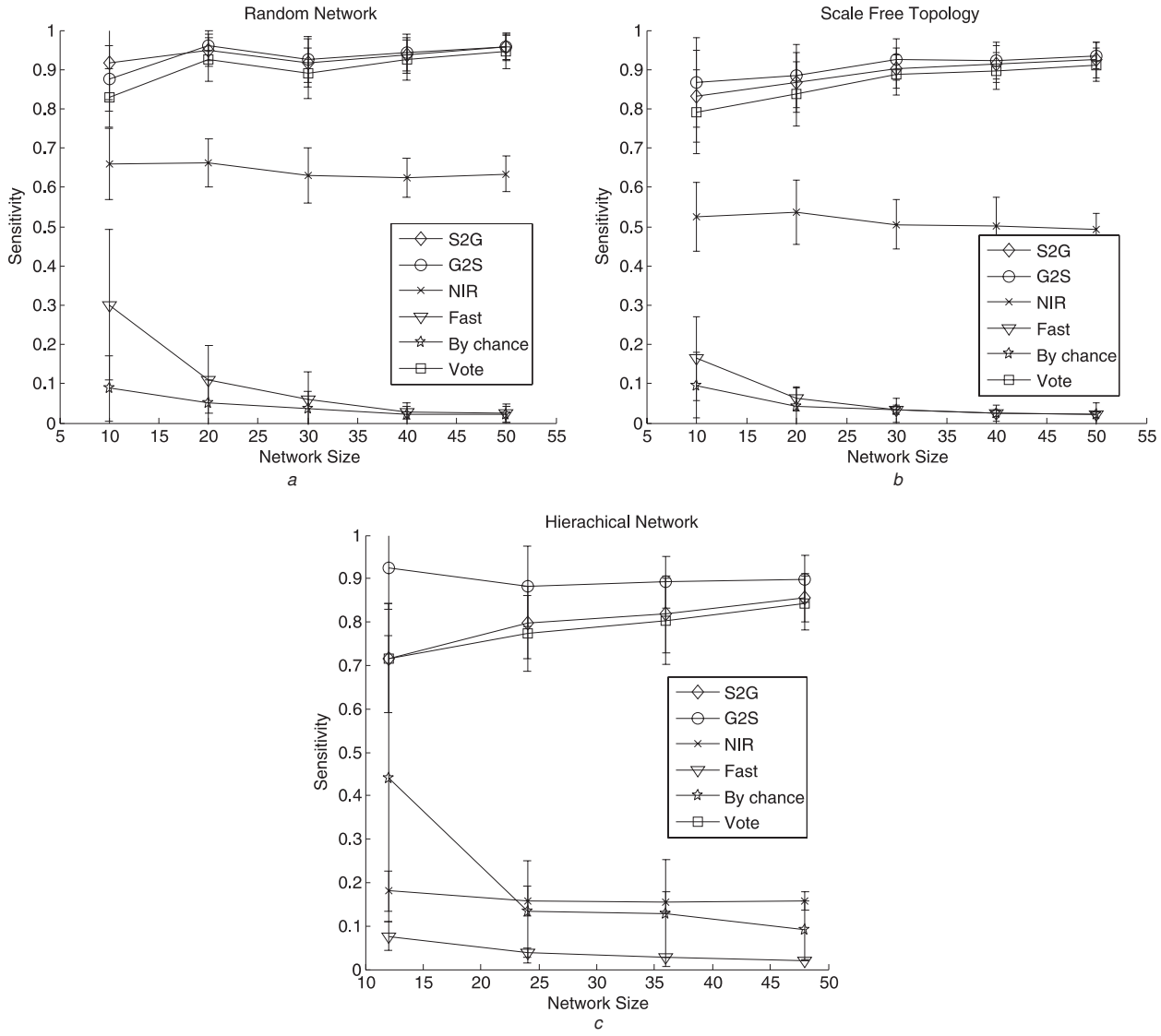

Fig. 2 *Performance of network inference algorithms for random, scale-free and hierarchical networks with 50 genes and* $k_{av} = 2$

a Random network topology
b Scale-free network topology
c Hierarchical topology

Table 2: Performance under different topologies with $k_{av}=2$ and network size of 50 after sampling 50 times

Algorithm	Random			Scale-free			Hierarchical		
	Sp	Sn	FDR	Sp	Sn	FDR	Sp	Sn	FDR
S2G	0.953	0.962	0.688	0.967	0.956	0.598	0.996	0.916	0.181
G2S	0.978	0.951	0.529	0.978	0.945	0.530	0.978	0.924	0.538
Vote	0.990	0.934	0.345	0.992	0.930	0.291	0.998	0.890	0.095
NIR	0.973	0.660	0.670	0.970	0.531	0.734	0.962	0.142	0.929
Fast	0.960	0.053	0.974	0.961	0.037	0.977	0.960	0.042	0.979
By chance	0.980	0.020	0.980	0.980	0.020	0.980	0.980	0.019	0.981

The performance of the FAST algorithm was found to be poor in all cases. The key network inference statistics for networks of size 50 are summarised in Table 2.

Connectivity. It has been shown that the connectivity varies in different components of transcriptional networks [10, 24]. Our simulations indicate that the level of connectivity also has a strong influence on the ability to infer networks. Fig. 3 shows that for networks of fixed size (40 genes) the performance of all algorithms deteriorates with increasing number of interactions in the network (average degree k_{av}) from two to five interactions per gene.

Noise. The most common problem in the analysis of data generated by high throughput experiments is the control of variability, both within and between experiments. In general, increasing the noise in our simulated data sets decreased the number of correctly identified connections, illustrated in Fig. 4*a*, and reduced the sensitivity with growing network size (data not shown). Specifically, the performance of the G2S approach drops towards the level of the NIR approach. This abrupt change in performance is presumably a result of loss of information content of smaller (near zero) expression ratio changes following perturbation of the network (Fig. 4*b*).

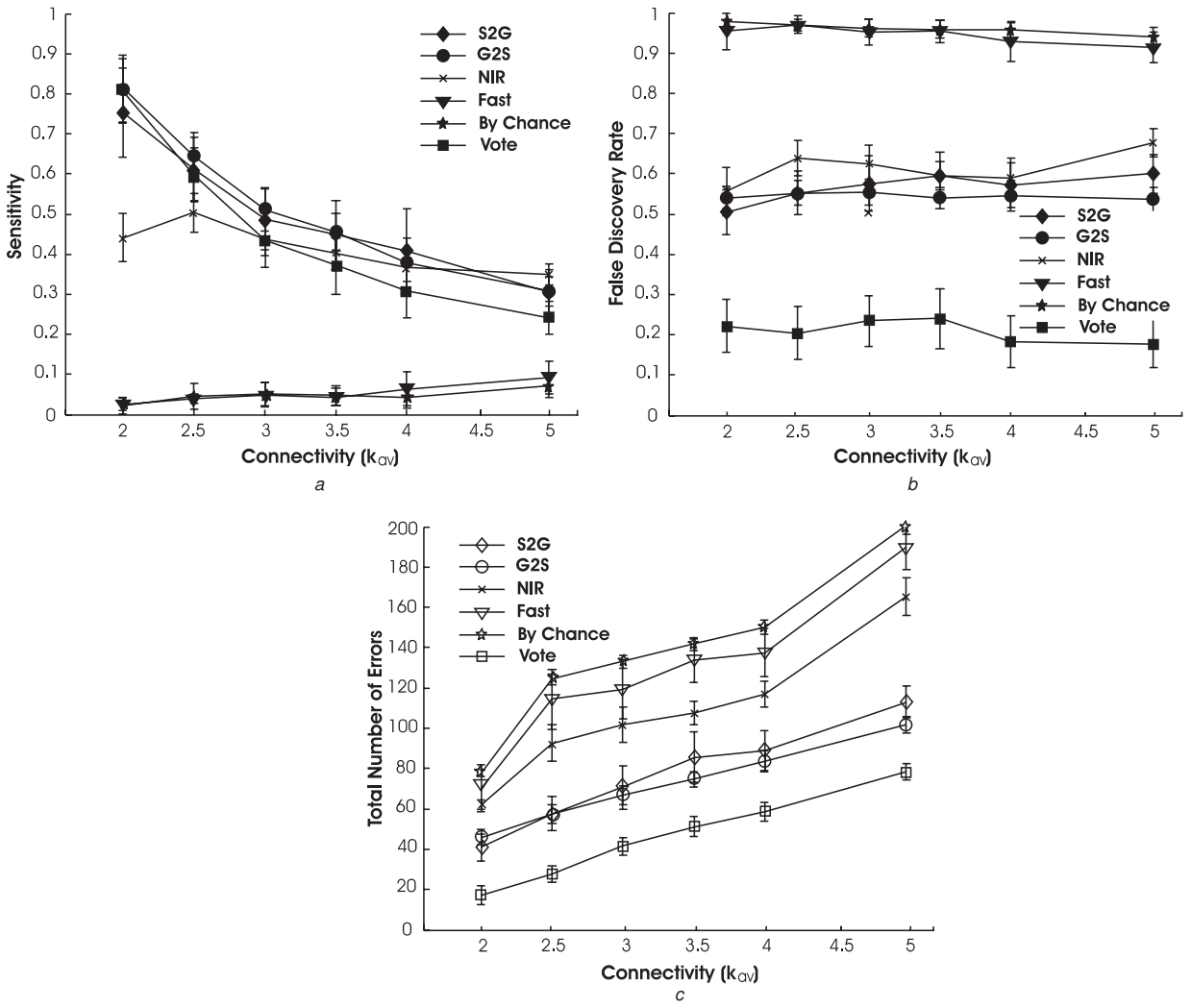

Fig. 3 *Influence of network connectivity on inference rate in scale-free networks with 40 genes*

a Sensitivity
b FDR
c Total number of errors

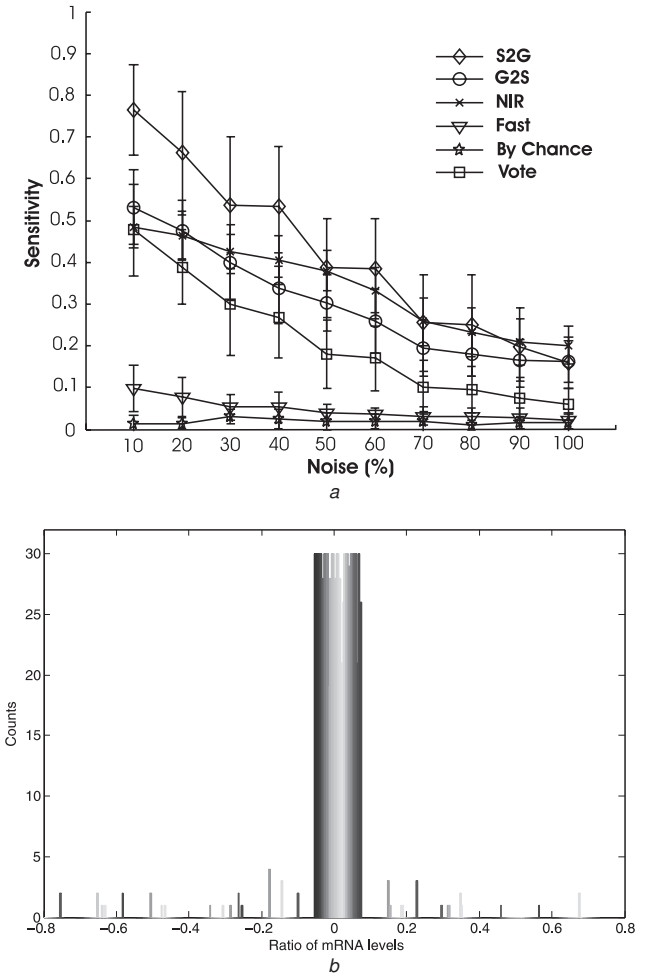

Fig. 4

a Effect of noise on network inference rate for scale-free networks with $k_{av} = 2$
Percentage of noise is calculated according to the expression levels per simulation
b Histogram of variation in individual mRNA species expression ratios for scale-free network with 30 genes and $k_{av} = 2$
Counts around zero indicate genes with small expression changes relative to control arising as a consequence of perturbation or variability

4 Discussion

In this work, we performed a direct comparison of reverse engineering algorithms, although it was necessary to pass information about the network structure (k_{av} and information about the noise distribution) to the NIR and FAST algorithms. Unsurprisingly, the S2G and G2S algorithms performed better on scale-free and hierarchical networks as these two algorithms make no assumption about the distribution of the number of regulatory interactions per gene. This additional flexibility does not appear to impair the performance of the S2G and G2S approaches for networks with random topology. In our simulations, the S2G algorithm showed the same low computational complexity as the FAST algorithm (but with the ability to infer regulatory interactions). The computational complexity of the NIR algorithm grows exponentially with increasing number of connections per gene. For dense networks, the cost to infer a network is similar to the G2S algorithm. Our analysis also highlights the effectiveness of ensemble approaches, such as the voting method used here, which combine predictions from different methods to reduce the FDR (inference error) without reducing the sensitivity (identification of true interactions), and thus improve predictive power.

The significance of degree distributions such as the scale-free and hierarchical topologies that we have considered here is their apparent prevalence in data on real biological networks. Recently, there has been some discussion as to whether these data indicate true power-law behaviour, or whether the biological networks in fact only approximate the scale-free property over some range of connectivities or sampling [41–43]. From the perspective of network inference, however, the key finding remains that biological networks are not randomly connected and therefore not well characterised by their average connectivity.

Our simulations with different network parameters show that the underlying structure of regulatory networks strongly influences the performance of reverse engineering algorithms. Simulations and reverse engineering approaches typically assume sparse networks with an average degree of around 2 [14, 16, 20], consistent with Jeong *et al.*'s estimate of $k_{av} \sim 2.4$ in protein–protein interaction maps from *S. cerevisiae* [23]. Guelzim *et al.* [24] recently found that transcriptional networks have at least an average degree of four interactions per node, but found incoming and outgoing connections (in and out degrees) have a similar proportion in the genome, whereas Luscombe *et al.* [44] have reported that the in and out degree may differ strongly in transcriptional networks ($k_{in} \ll k_{out}$). A high out degree indicates the existence of a large number of target genes that are at the end of signalling pathways, mostly regulated by central hubs [44]. Our simulations showed that the increase in the average degree impacts the inference and leads to an increase of false negatives in the inferred datasets, whereas the difference between in and out degrees is less worrying for successful inference. We also observed a significant deterioration of the sensitivity of the algorithms with increasing measurement noise, decreasing the probability of discovery of regulatory interactions. Measurement noise most severely affects our ability to infer networks with hierarchical structure and hub-like network motifs, which if obscured by noise will impact on the whole network inference. In contrast, we found that the biological variability of a system did not strongly influence the performance as network size was varied.

We used simulated gene perturbation data, in which individual genes were down or up regulated (Fig. 4*b*) and, if a steady state was reached, the corresponding steady network expression levels recorded. Equally, we could have applied the techniques to knockout data sets. Several authors have criticised the usage of knockout experiments because they cut essential regulatory interactions in a biological system. In our network simulations, we found that failure to reach a new steady state was not uncommon (for example, a perturbation resulting in periodic expression levels). We also observed that highly connected networks have a very high tolerance towards perturbations. This behaviour is known to be inherent to biological systems [9]. For the reverse engineering algorithms, these perturbations thus provide less information about the network, which may be responsible for the decline in the rate of inferred connections with increasing k_{av}.

There are at present no fully validated large-scale biological datasets available, and we must rely on simulation models to assess the performance of our algorithms. Indeed, the use of standard simulated data sets to compare algorithm performance has been suggested [15, 31, 45]. The caveat to

the use of simulated data sets, of course, is that they will only provide evidence on the performance with real data if the simulation model gives a faithful representation of real biological networks. Although models for large scale data analysis cannot be as detailed as tissue or pathway-specific models [46, 47], they can be made increasingly realistic using topological information such as k_{av}, in against out degree ratio, and power-law exponent γ, measured in real genomes [48, 49]. Using experimentally inferred values for DNA transcription and degradation rates for known gene families to generate a hierarchy of time scales, allows complex gene regulation models to be constructed.

Several alternative transcriptional network simulation environments are available. Regulatory interactions can be modelled using a mix of discrete Boolean logic and differential equations [50]. Zak *et al.* [47] consider the limitations in information content of gene expression data for reverse engineering regulatory networks. Kauffman [51] presents a proposal for using an ensemble approach to understand genetic regulatory networks. The S-system approach has been used to produce and analyse transcriptional perturbation data to find the correct regulatory mechanisms [52]. Probabilistic models simulating transcriptional data are presented by Mao and Resat [53] and Zhou *et al.* [32]. Vu and Vohradsky [30] published a gene network simulator that is based on a neural network principle. The availability of different models allows cross-validation of results from experimental data, as it seems reasonable that some models are better simulations for a particular biological problem.

From a practical perspective, establishing the size of the perturbations made to the system may in some circumstances be a difficult task. For over-expression of transcripts, for example using plasmids [2], the rate of mRNA production by the expression vector can be established. In cases where transcription is perturbed using other chemical or pharmaceutical means, for example, the target of the perturbation may not be known, let alone the perturbation strength. Several authors have addressed this issue in the context of regulatory network identification and shown that a scaled version of the connectivity matrix can be recovered without knowledge of the perturbations themselves [54, 55].

Finally, we reflect that underlying all gene regulatory network modelling is the assumption that correlations between transcript levels reveal regulatory interactions. Segal *et al.* [12] have shown for yeast that the transcription level and the protein levels do not have to be correlated. This would suggest that combining measurement data of mRNA and protein levels may improve the inference of regulatory interactions from experimental results.

In this paper, we have introduced two new approaches to analyse data from gene perturbation experiments, we have validated these and two published algorithms on an computational model simulating gene expression data. We showed that our selection method performs well on the simulated data sets. The FAST and NIR algorithms assume that each gene has the same number of regulatory interactions. The new approaches S2G and G2S do not make an assumption about the degree distribution of the genes, and are therefore well suited to the identification of scale-free and hierarchical networks.

5 Acknowledgments

We thank Drs Patrick E. McSharry, Santiago Schnell, Franz Pichler and Mik Black for their suggestions. Additionally, we would like to thank Drs Timothy Gardner and J.J. Collins for providing the NIR algorithm. J.W. was supported by the Center for Molecular Biodiscovery, and E.J.C. was supported by the NZ Institute of Mathematics and its Applications (NZIMA) and the Center for Molecular Biodiscovery, University of Auckland.

Software availability. A MATLAB program that implements the S2G and G2S algorithms is available on request to E.J.C.

6 References

1 Hughes, T.R., Marton, M.J., Jones, A.R., Roberts, C.J., Stoughton, R., Armour, C.D., Bennett, H.A., Coffey, E., Dai, H., He, Y.D., Kidd, M.J., King, A.M., Meyer, M.R., Slade, D., Lum, P.Y., Stepaniants, S.B., Shoemaker, D.D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M., and Friend, S.H.: 'Functional discovery via a compendium of expression profiles', *Cell*, 2000, **102**, pp. 109–126

2 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**, (5629), pp. 102–105

3 Haggarty, S.J., Clemons, P.A., and Schreiber, S.L.: 'Chemical genomic profiling of biological networks using graph theory and combinations of small molecule perturbations', *J. Am. Chem. Soc.*, 2003, **125**, pp. 10543–10545

4 Kamath, R.S., and Ahringer, J.: 'Genome-wide RNAi screening in *Caenorhabditis elegans*', *Methods*, 2003, **30**, pp. 313–321

5 Tewari, M., Hu, P.J., Ahn, J.S., Ayivi-Guedehoussou, N., Vidalain, P.O., Li, S., Milstein, S., Armstrong, C.M., Boxem, M., Butler, M.D., Busiguina, S., Rual, J.F., Ibarrola, N., Chaklos, S.T., Bertin, N., Vaglio, P., Edgley, M.L., King, K.V., Albert, P.S., Vandenhaute, J., Pandey, A., Riddle, D.L., Ruvkun, G., and Vidal, M.: 'Systematic interactome mapping and genetic perturbation analysis of a *C. elegans* TGF-beta signaling network', *Mol. Cell*, 2004, **13**, pp. 469–482

6 Parsons, A.B., Brost, R.L., Ding, H., Li, Z., Zhang, C., Sheikh, B., Brown, G.W., Kane, P.M., Hughes, T.R., and Boone, C.: 'Integration of chemical-genetic and genetic interaction data links bioactive compounds to cellular target pathways', *Nature Biotechnol.*, 2004, **22**, pp. 62–69

7 di Bernardo, D., Thompson, M.J., Gardner, T.S., Chobot, S.E., Eastwood, E.L., Wojtovich, A.P., Elliott, S.J., Schaus, S.E., and Collins, J.J.: 'Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks', *Nature Biotechnol.*, 2005, **23**, (3), pp. 377–383

8 Imoto, S., Tamada, Y., Araki, H., Yasuda, K., Print, C.G., Charnock-Jones, S.D., Sanders, D., Savoie, C.J., Tashiro, K., Kuhara, S., and Miyano, S.: 'Computational strategy for discovering druggable gene networks from genome-wide RNA expression profiles'. Proc. Pacific Symp. on Bioinformatics, 2006, **11**, pp. 559–571

9 Babu, M.M., Luscombe, N.M., Aravind, L., Gerstein, M., and Teichmann, S.A.: 'Structure and evolution of transcriptional regulatory networks', *Curr. Opin. Struct. Biol.*, 2004, **14**, pp. 283–291

10 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

11 Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., and Barkai, N.: 'Revealing modular organization in the yeast transcriptional network', *Nature Genet.*, 2002, **31**, pp. 370–377

12 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', *Nature Genet.*, 2003, **34**, pp. 166–176

13 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', *Nature Biotechnol.*, 2003, **21**, pp. 1337–1342

14 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

15 de la Fuente, A., Bing, N., Hoeschele, I., and Mendes, P.: 'Discovery of meaningful associations in genomic data using partial correlation coefficients', *Bioinformatics*, 2004, **20**, pp. 3565–3574

16 Schaefer, J., and Strimmer, K.: 'An empirical Bayes approach to inferring large-scale gene association networks', *Bioinformatics*, 2005, **21**, pp. 754–764

17 Pe'er, D., Regev, A., Elidan, G., and Friedman, N.: 'Inferring subnetworks from perturbed expression profiles', *Bioinformatics*, 2001, **17**, (Suppl 1), S215–S224

18 Husmeier, D.: 'Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks', *Bioinformatics*, 2003, **19**, (17), pp. 2271–2282
19 Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C.: 'Random Boolean network models and the yeast transcriptional network', *Proc. Natl. Acad. Sci. USA*, 2003, **100**, pp. 14796–14799
20 Tegner, 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**, (10), pp. 5944–5949
21 Farina, L., and Mogno, I.: 'A fast reconstruction algorithm for gene networks'. arXiv:qbio.QM/0401044v1, 2004
22 Thieffry, D., Huerta, A.M., Perez-Rueda, E., and Collado-Vides, J.: 'From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*', *Bioessays*, 1998, **20**, pp. 433–440
23 Jeong, H., Mason, S.P., Barabási, A.L., and Oltvai, Z.N.: 'Lethality and centrality in protein networks', *Nature*, 2001, **411**, (6833), pp. 41–42
24 Guelzim, N., Bottani, S., Bourgine, P., and Kepes, F.: 'Topological and causal structure of the yeast transcriptional regulatory network', *Nature Genet.*, 2002, **31**, pp. 60–63
25 Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., and Barabasi, A.L.: 'The large-scale organization of metabolic networks', *Nature*, 2000, **407**, (6804), pp. 651–654
26 Wagner, A.: 'The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes', *Mol. Biol. Evol.*, 2001, **18**, pp. 1283–1292
27 Featherstone, D.E., and Broadie, K.: 'Wrestling with pleiotropy: genomic and topological analysis of the yeast gene expression network', *Bioessays*, 2002, **24**, pp. 267–274
28 Luscombe, N.M., Qian, J., Zhang, Z., Johnson, T., and Gerstein, M.: 'The dominance of the population by a selected few: power-law behaviour applies to a wide variety of genomic properties', *Genome Biol.*, 2002, **3**, research0040.1–0040.7
29 Crampin, E.J., McSharry, P.E., and Schnell, S.: 'Extracting biochemical reaction kinetics from time series data', *Lect. Notes Artif. Intell.*, 2004, **3214**, pp. 329–336
30 Vu, T.T., and Vohradsky, J.: 'Genexp – a genetic network simulation environment', *Bioinformatics*, 2002, **18**, pp. 1400–1401
31 Mendes, P., Sha, W., and Ye, K.: 'Artificial gene networks for objective comparison of analysis algorithms', *Bioinformatics*, 2003, **19**, (Suppl. 2), pp. ii122–ii129
32 Zhou, X., Wang, X., Pal, R., Ivanov, I., Bittner, M., and Dougherty, E.R.: 'A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks', *Bioinformatics*, 2004, **20**, pp. 2918–2927
33 Erdős, P., and Renyi, A.: 'On the evolution of random graphs', *Publ. Math. Inst. Hung. Acad. Sci.*, 1960, **5**, pp. 17–61
34 Ravasz, E., and Barabási, A.L.: 'Hierarchical organization in complex networks', *Phys. Rev. E*, 2003, **67**, (2 Pt 2), p. 026112
35 Barabási, A.-L., and Albert, R.: 'Emergence of scaling in random networks', *Science*, 1999, **286**, pp. 509–512
36 Savageau, M.A.: 'Biochemical systems analysis' (Addison-Wesley, Reading, MA, 1976)
37 Papp, B., and Oliver, S.: 'Genome-wide analysis of the context-dependence of regulatory networks', *Genome Biol.*, 2005, **6**, p. 206
38 Akaike, H.: 'A new look at the statistical identification model', *IEEE Trans. Auto. Control*, 1974, **19**, pp. 716–723
39 Rissanen, J.: 'Consistent order estimates of autoregressive processes by shortest description of data' in Jacobs, O.L.R., *et al.* (Eds.): 'Analysis optimisation of stochastic systems' (Academic Press, New York, 1980)
40 Judd, K., and Mees, A.: 'On selecting models for nonlinear time series', *Physica D*, 1995, **82**, pp. 426–444
41 Thomas, A., Cannings, R., Monk, N.A.M., and Cannings, C.: 'On the structure of protein–protein interaction networks', *Biochem. Soc. Trans.*, 2003, **31**, pp. 1491–1496
42 Stumpf, M.P.H., Wiuf, C., and May, R.M.: 'Subnets of scale-free networks are not scale-free: Sampling properties of networks', *Proc. Natl. Acad. Sci. USA*, 2005, **102**, pp. 4221–4224
43 Khanin, R., and Wit, E.: 'How scale-free are biological networks'. Department of Statistics, University of Glasgow
44 Luscombe, N.M., Babu, M., Yu, H., Snyder, M., Teichmann, S.A., and Gerstein, M.: 'Genomic analysis of regulatory network dynamics reveals large topological changes', *Nature*, 2004, **431**, pp. 308–312
45 Husmeier, D.: 'Reverse engineering of genetic networks with Bayesian networks', *Biochem. Soc. Trans.*, 2003, **31**, pp. 1516–1518
46 Marnellos, G., Mjolsness, E., and Kintner, C.: 'Delta-Notch lateral inhibitory patterning in the emergence of ciliated cells in *Xenopus*: experimental observations and a gene network model', *Pac. Symp. Biocomput.*, 2000, **1**, pp. 329–340
47 Zak, D.E., Gonye, G.E., Schwaber, J.S., and Doyle, F.J.: 'Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network', *Genome Res.*, 2003, **13**, pp. 2396–2405
48 Pastor-Satorras, R., Smith, E., and Sole, R.V.: 'Evolving protein interaction networks through gene duplication', *J. Theor. Biol.*, 2003, **222**, (2), pp. 199–210
49 Teichmann, S.A., and Babu, M.M.: 'Gene regulatory network growth by duplication', *Nature Genet.*, 2004, **36**, (5), pp. 492–496
50 Bolouri, H., and Davidson, E.H.: 'Transcriptional regulatory cascades in development: initial rates, not steady state, determine network kinetics', *Proc. Natl. Acad. Sci. USA*, 2003, **100**, (16), pp. 9371–9376
51 Kauffman, S.: 'A proposal for using the ensemble approach to understand genetic regulatory networks', *J. Theor. Biol.*, 2004, **230**, pp. 581–590
52 Veflingstad, H., Almeida, J., and Voit, E.O.: 'Priming nonlinear searches for pathway identification', *Theor. Biol. Med. Model.*, 2004, **1**, pp. 716–723
53 Mao, L.Y., and Resat, H.: 'Probabilistic representation of gene regulatory networks', *Bioinformatics*, 2004, **20**, pp. 2258–2269
54 de la Fuente, A., Brazhnik, P., and Mendes, P.: 'A quantitative method for reverse engineering gene networks from microarray experiments using regulatory strengths'. Proc. Int. Conf. on Systems Biology, 2001
55 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

7 Appendix

The following algorithm, S2G, implements the selection approach by Judd and Mees [40]. We note that although this selection algorithm was originally developed for pseudo-linear models with non-orthogonal basis functions, it can equally well be applied to construct purely linear models [40], and we have found that it works well, as described in Section 3. In this case, to find the unknowns $\hat{a}_{ij}$, we can proceed gene by gene.

The dataset provided to the algorithm contains the scaled steady-state concentrations $\hat{\boldsymbol{Y}}$ and the perturbation strengths for gene i, $\hat{\boldsymbol{u}}_i$. The algorithm finds the number of interactions (basis functions) minimising the cost function $C_{\mathrm{AIC}}(K)$ by choosing a set of K basis functions, Φ, from the pool $\boldsymbol{V}$ containing all N possible, normalised basis functions (i.e. $\boldsymbol{V} = \{\hat{\boldsymbol{y}}_j\}$, $j = 1, \ldots, N$). $\boldsymbol{E}_i^{(K)} = \sum_{k=1}^{K} \hat{a}_{ik}\, \hat{\boldsymbol{y}}_{\phi(k)} + \hat{\boldsymbol{u}}_i$ is the model reconstruction error vector resulting from using K basis functions in Φ, where $\phi(k)$ are their indices.

1. Let Φ initially be an empty set and $k = 1$ the number of interactions to be included in the basis in this iteration. Define $\boldsymbol{E}^{(0)} = \hat{\boldsymbol{u}}_i$.
2. Let vector $\boldsymbol{\mu} = -\boldsymbol{V}^{\mathrm{T}} \cdot \boldsymbol{E}^{(k-1)}$ be the projection of the reconstruction errors onto the pool of basis functions. Let i_{in} be the index in $\boldsymbol{V}$ of the component of $\boldsymbol{\mu}$ with maximum absolute value. This will be the basis function included in the basis $\Phi = \Phi \cup \{i_{\mathrm{in}}\}$.
3. Calculate the coefficients a_{ij} associated with all the basis functions in the basis Φ [by finding the pseudo-inverse matrix for the linear equation (10)]. Let i_{out} be the index of the interaction having the coefficient with smallest absolute value. This basis function is a candidate for removal from the basis Φ.
4. If $i_{\mathrm{in}} \neq i_{\mathrm{out}}$, then remove i_{out} from the basis, $\Phi = \Phi \backslash \{i_{\mathrm{out}}\}$, and go to step 2.
5. Store the current $\Phi^{(k)} = \Phi$ and calculate the cost function $C_{\mathrm{AIC}}(k)$.
6. If $C_{\mathrm{AIC}}(k) < C_{\mathrm{AIC}}(k-1)$, then increase $k = k + 1$ and go to step 2.
7. The algorithm terminates, with $K = k - 1$ the optimum number of basis functions, and $\Phi^{(K)}$ the set of K basis functions in $\boldsymbol{V}$, which minimises the cost function.

This procedure can be repeated for each row to find all non-zero elements in $\hat{a}_{ij}$.

The G2S selection technique starts with the basis Φ containing all possible interactions in $\boldsymbol{V}$ and, in a similar manner, systematically removes basis functions until the cost function $C_{\mathrm{AIC}}(K)$ is minimised.

7.1 Assessment of algorithm performance

To assess the network reconstruction performance, we counted true positives (TP: correctly identified interactions), false positives (FP: incorrectly identified interactions), true negatives (TN: correctly identified zeros) and false negatives (FN: incorrectly identified zeros). From this information, we estimated the sensitivity (Sn: true positive rate, also called 'Power'), specificity (Sp: the true negative rate) and the FDR

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}, \quad \mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}}$$

These statistics were used to validate an existing interaction without taking the sign of the interaction into account. To consider the identification of positive and negative interactions (activation and repression), we also recorded the signed true positive rate and signed true negative rate. The accuracy in detecting the correct sign approaches 100% for NIR, S2G and G2S (data not shown).