BMC Bioinformatics

BioMed Central

Methodology article

Using large-scale perturbations in gene network reconstruction

Thomas MacCarthy*[1], Andrew Pomiankowski[1,2] and Robert Seymour[1,3]

Address: [1]COMPLEX, University College London, 4 Stephenson Way, London NW1 2HE, UK, [2]Department of Biology, University College London, 4 Stephenson Way, London NW1 2HE, UK and [3]Department of Mathematics, University College London, Gower Street, London WC1E 2BT, UK

Email: Thomas MacCarthy* - t.maccarthy@ucl.ac.uk; Andrew Pomiankowski - a.pomiankowski@ucl.ac.uk;
Robert Seymour - rms@math.ucl.ac.uk

* Corresponding author

Published: 19 January 2005

BMC Bioinformatics 2005, **6**:11 doi:10.1186/1471-2105-6-11

Received: 20 October 2004
Accepted: 19 January 2005

This article is available from: http://www.biomedcentral.com/1471-2105/6/11

Abstract

Background: Recent analysis of the yeast gene network shows that most genes have few inputs, indicating that enumerative gene reconstruction methods are both useful and computationally feasible. A simple enumerative reconstruction method based on a discrete dynamical system model is used to study how microarray experiments involving modulated global perturbations can be designed to obtain reasonably accurate reconstructions. The method is tested on artificial gene networks with biologically realistic in/out degree characteristics.

Results: It was found that a relatively small number of perturbations significantly improve inference accuracy, particularly for low-order inputs of one or two genes. The perturbations themselves should alter the expression level of approximately 50–60% of the genes in the network.

Conclusions: Time-series obtained from perturbations are a common form of expression data. This study illustrates how gene networks can be significantly reconstructed from such time-series while requiring only a relatively small number of calibrated perturbations, even for large networks, thus reducing experimental costs.

Background

Recent technological advances have led to an explosive growth in high-throughput genomic and proteomic data such as DNA microarrays. The rapid growth in available data has led in turn to a need for novel quantitive methods for analysis. As a consequence of this need, the reconstruction of gene network architectures from DNA microarray expression data has become a major goal in the field of systems biology. An increased understanding of the network architectures and their respective dynamics will enable novel approaches to disease treatments by allowing us, for example, to identify drug targets *in silico* which manipulate the functional outputs of these networks. This process is expected to lead to novel classes of drug based on a network approach to cellular dynamics.

Frequently, the gene expression data itself is derived from perturbation experiments such as stress conditions, temperature shifts, and chemical treatments; for example, the widely used yeast cell-cycle datasets of Cho [1] and Spellman [2]. Although these global perturbations are carried out in order to reveal causality between genes, it is not always clear how experiments should be designed so as to reveal as much causality as possible, while both minimising costly experimentation and remaining computationally tractable.

A range of computational and mathematical techniques have been adopted in the effort to find a successful gene network reconstruction technique. Reconstruction methods often have to negotiate a tradeoff between intensive (often intractable) computations, and having to perform a large number of costly experiments. Certain progress can be achieved by making simplifications, such as imposing a limit on the number of inputs to each gene, or making steady state assumptions about the system [3,4]. Some techniques described in the literature offer efficient algorithms, but require a large number of experiments, perhaps as many as there are genes [5-7]. On the other hand, theoretical work on Boolean models has shown [8] that perhaps as few as $O(log(n))$ experiments (input/output pairs) might be required for n genes, but that to infer these relationships requires the use of computationally costly enumeration methods.

In this paper, we propose to explore the issue of how perturbation microarray experiments might be designed, and to suggest how such experiments might be optimised so as to maximize inference capability. Logical gene network models have previously been used to investigate gene network robustness [9], perturbation dynamics [10] and evolutionary potential [11], and form the basis of the inference method used in this study. This inference method [11] is similar to others in which networks with a minimal number of connections are reconstructed through enumeration [12,13]. Given the significant speed advantage of integer computation over floating point computation, and that most genes are expected to have few inputs (93% have between 1 and 4 [14]), the method is considered to be adequate for this investigation. In this study, exhaustive evaluation was performed up to a maximum of 4 inputs of both positive and negative sign (see Methods). Enumeration is computationally feasible on an ordinary desktop computer for medium-sized networks ($n \sim 100$), and still tractable for large networks ($n \sim 1000$), though this would require some parallelisation. The global perturbations themselves are simulated by changing the state of each gene at random. A perturbation intensity measure q, defines the probability that each gene will change state (see Methods).

Results and discussion

A limited number of perturbations significantly improve accuracy

A discrete dynamical model was used to generate time series data from random networks (see Methods). To measure the effect of adding perturbations on inference ability, inference *sensitivity* (defined as true positives/true positives + false negatives, see Methods) was measured against P, the number of additional perturbations. Figure 1 shows the results for predicted solutions with one and two inputs, as well as overall sensitivity. The top graph in figure 1 shows that overall sensitivity is clearly enhanced by including more perturbation experiments, with lower order solutions (one and two inputs) reaching higher levels of sensitivity. The bottom graph shows the corresponding inverse relationship for the standard deviation of the sensitivity (lower for higher P).

It should be noted that the algorithm tends to underestimate the number of inputs a gene may have. This is to be

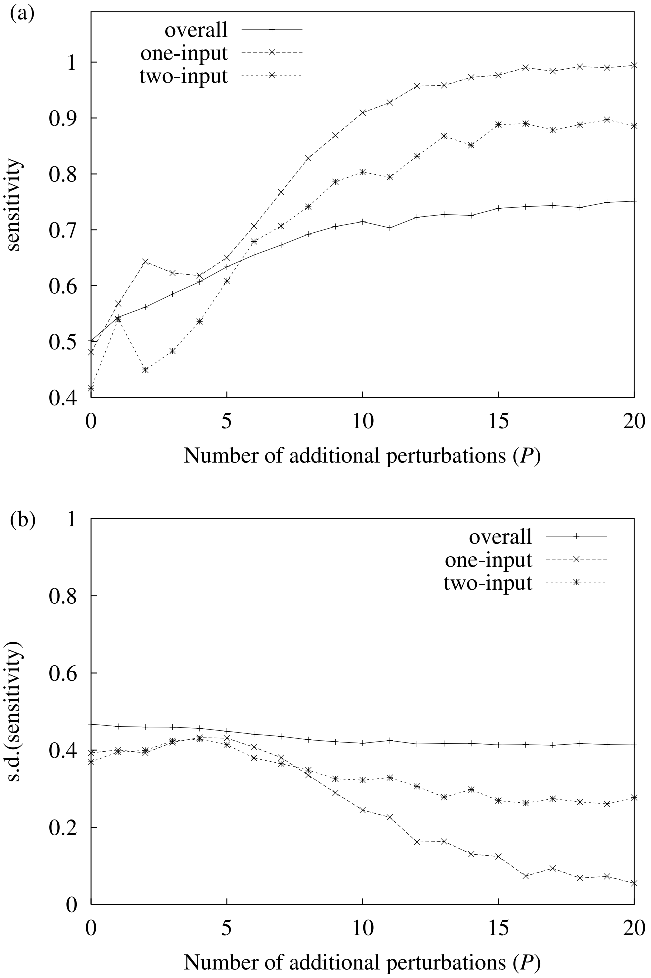

Figure 1

Sensitivity vs. *P*. (a) Sensitivity vs. number of additional perturbations used. (b) The corresponding standard deviation is shown here separately for clarity. The curves represent results for overall (i.e. all solutions) sensitivity, and specific sensitivity for (predicted) one and two-input solutions. Sensitivity is generally lower for higher order of inputs. Accuracy increases significantly with the number of additional perturbations used. The results shown are average values for 250 random networks at each data point. The remaining parameters are fixed: network size $N = 50$, perturbation intensity $q = 0.5$.

Table 1: Solution set sizes Distribution for the inferred solution set sizes, compared to the distribution of indegree in the actual network for the simulations. These statistics were produced from 250 random networks run using the following parameter values: *N* = 50, *P* = 12, and *q* = 0.5. The table illustrates how the algorithm overestimates the number of solutions with zero inputs.

$\|Y_i\|$	0	1	2	3	≥ 4
inferred	0.57	0.12	0.07	0.05	0.19
actual	0.37	0.24	0.15	0.10	0.14

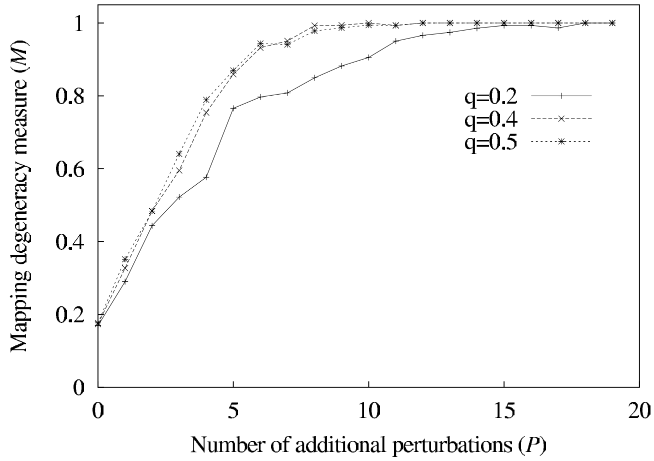

Figure 2
***M* vs. *P*.** M (the number of distinct "concatenated" vectors S_i divided by N, the number of genes) increases in value, as the number of perturbations (*P*) is increased. The graph shows curves for three values of perturbation intensity *q*.

expected in genes for which dynamics cannot be informative: for example, consider a gene *i* which has one or more negative inputs, as well as having default value OFF. Since the discrete dynamics for this gene will be the same as if it had no inputs at all (i.e. zero gene expression for $t > 0$), the presence of the inputs is impossible to infer. This underestimation effect is clear in table 1, which compares the distribution of inferred solution set sizes ($|Y_i|$, see Methods) with the actual solution sizes (i.e. the indegree distribution), and shows that the method is only able to produce roughly half the number of one and two input solution sets that actually exist.

The increase in sensitivity with *P* can be explained at least partially, in the following way. Since the time series are discrete, many of the genes may have identical behaviour over time despite having different inputs (i.e. $s_i(t) = s_j(t)$ for two different genes *i* and *j*). If we define a "concatenated" time series vector $S_i = \{(s_i^0(t), s_i^1(t), ..., s_i^P(t)) : t \geq 0\}$ for gene *i*, and then map each gene *i* onto S_i, we obtain a many-to-one mapping. As we increase the number of perturbations, we might expect the number of distinct time series also to increase. We define a simple measure to quantify this mapping, $M = n'/N$ where n' is the number of distinct vectors S_i, and *N* is the number of genes. The maximum value of $M = 1$ indicates that the mapping of genes to time series is one-to-one, whereas lower values indicate degenerate mappings. The manner in which *M* increases with the number of perturbations is shown in figure 2, and shows how the increase in M reflects the corresponding increase in sensitivity (figure 1).

Network size and optimal perturbation intensity

The experiments described above were repeated to consider variations in two other parameters: the network size *N*, and the perturbation intensity parameter *q* (roughly, the proportion of genes whose initial expression level is changed in each perturbation experiment – see Methods).

To consider the first case, the minimum number of perturbations P^* required to reach a given high accuracy criterion was measured for different values of the network size *N*. The high accuracy criterion was defined as average sensitivity = 0.95 for one-input solution sets (average sensitivity is found using a default value q = 0.5 and averaging for all the sensitivity measurements obtained from 250 random networks). To find P^*, we first find the number of perturbations P^+, such that average sensitivity $P^+ \geq 0.95$, and average sensitivity $(P^+ - 1) < 0.95$. If average sensitivity $P^+ > 0.95$, we use simple linear interpolation to find the (real) value of P^* between P^+ and $(P^+ - 1)$ for which average sensitivity = 0.95.

The resulting values for P^* are shown in figure 3. Since the relationship is expected to be logarithmic [8], the plot shows *log*(*N*) against P^* (logarithms used are base 10). A least squares best fit gives $P^* \simeq 1.75\ log(N) + 7.02$, which, for $N = 1000$, gives $P^* \simeq 12.26$. In order to obtain a measure of variance for P^*, we would need to calculate P^*-equivalent values for many individual networks separately, then consolidate these values to obtain the relevant statistics. However, because it was only feasible to consider medium-sized networks ($20 \leq N \leq 70$), and for any such network we often find only a small number of one-input solution sets, such statistics were found to be unreliable.

The second case (varying perturbation intensity) suggests an optimal range for *q*. Figure 4a shows the inference sensitivity over a range of values for *q*, and figure 4b shows

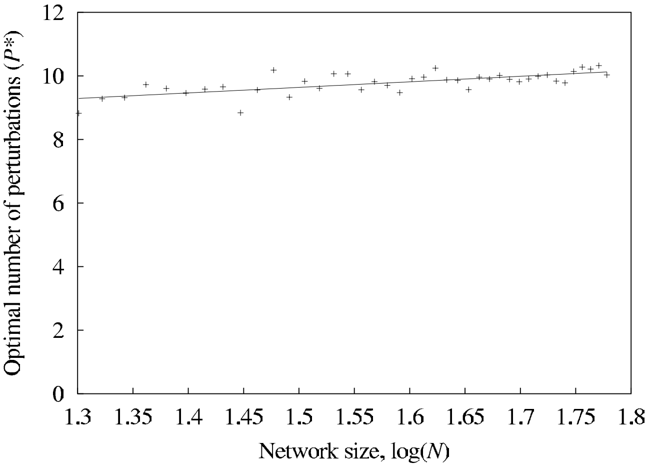

Figure 3
Perturbations required for high accuracy The minimum number of perturbations (P*) required to reach the high accuracy criterion (average sensitivity = 0.95) for different values of the network size N. Each point represents the average value for 250 random networks inferred. This is equivalent to finding the value of P for which sensitivity = 0.95 on the one-input curve of figure 1(a) for different values of N (figure 1(a) shows N = 50). A linear fit is also shown.

the corresponding standard deviation. Again, inference sensitivity for one-input solutions is higher than for two-input solutions, which in turn is higher than overall sensitivity. For one-input solutions, the results show a clear peak for sensitivity close to the range $0.5 < q < 0.6$. Together with a corresponding minimisation of the standard deviation in this interval (though it still remains fairly high in absolute terms), these results suggest that perturbation intensity should be close to this range to optimise inference accuracy.

Conclusions

A recent analysis of the yeast genetic network has shown that 93% of genes are regulated by between 1 and 4 genes [14]. This suggests that enumerative network reconstruction methods can be useful within computationally feasible limits. Experiments involving large-scale perturbations (such as temperature shifts, chemical stress) are a standard way of obtaining time-series of gene expression data [1,2]. A key result of [14] is that indegree appears to follow an exponential distribution, whereas outdegree follows a scale-free distribution, which has enabled the generation of realistic artificial gene networks used here. A logical model [11] was used to simulate the perturbed expression data. Subsequently, experimental parameters were considered in relation to inference accuracy, namely: a) number of perturbations required, P, and b) perturbation intensity, q.

The inference method itself is most useful for low order inputs, with inference accuracy maximized for predicted single input genes. More accurate methods have been proposed, though these generally require a much larger number of experiments [5,15]. Methods such as the one proposed here, which infer relationships from expression data may well be more successful when used in conjunction with other methods such as promoter analysis [16,17], or when used to drive experimental procedure [18]. Here, the results show that only a relatively small number of perturbations are necessary in order to achieve a substantial inference accuracy, even for large N. These relatively modest experimental requirements would presumably imply lower experimental costs. The results also suggest that the perturbations should be calibrated (by changing stress intensity, for example), so as to alter the expression levels of approximately half the genes in each experiment. Generating perturbations which alter the expression level of half the genes at random may be difficult to achieve in practice, though experiments can be designed to come as close to this goal as possible. Even in the absence of optimal perturbations, we hope the simulation approach described here will still serve as a useful tool for planning experiments.

Methods

Discrete dynamical model

For a system of N genes, the state of each gene s_i (i = 1, .., N) is represented by the binary values 0(OFF) and 1(ON). Additionally, each gene is assigned a default ON/OFF state $\theta_i \in \{0, 1\}$. The gene interactions are described by an ($N \times N$) matrix C, composed of elements $C_{ij} \in \{-1, 0, +1\}$, representing the positive(+1), zero(0) or negative(-1) influence of gene j on gene i. State transitions are calculated as follows:

$$s_i(t+1) = \sigma(u_i(t)) \qquad (1)$$
$$\text{where} \quad u_i(t) = \sum_j C_{ij} s_j(t), \ \ \sigma(x) = \begin{cases} 1 & \text{if} \quad x > -\theta_i \\ 0 & \text{otherwise} \end{cases}$$

The state of the ith gene at the next timestep, $s_i(t + 1)$, is therefore determined by the balance of positive versus negative inputs which are ON at the previous timestep t. If the balance is positive, then $u_i(t) > 0$ and the next state will be 1(ON). Similarly, if the balance is negative, then $u_i(t) < 0$ and the next state will be 0(OFF). If $u_i(t) = 0$ (indicating either that there are no active input connections, or that they balance out), then the default value θ_i determines the next state. This default value needs to be given *a priori*, and for the purpose of this study will be random.

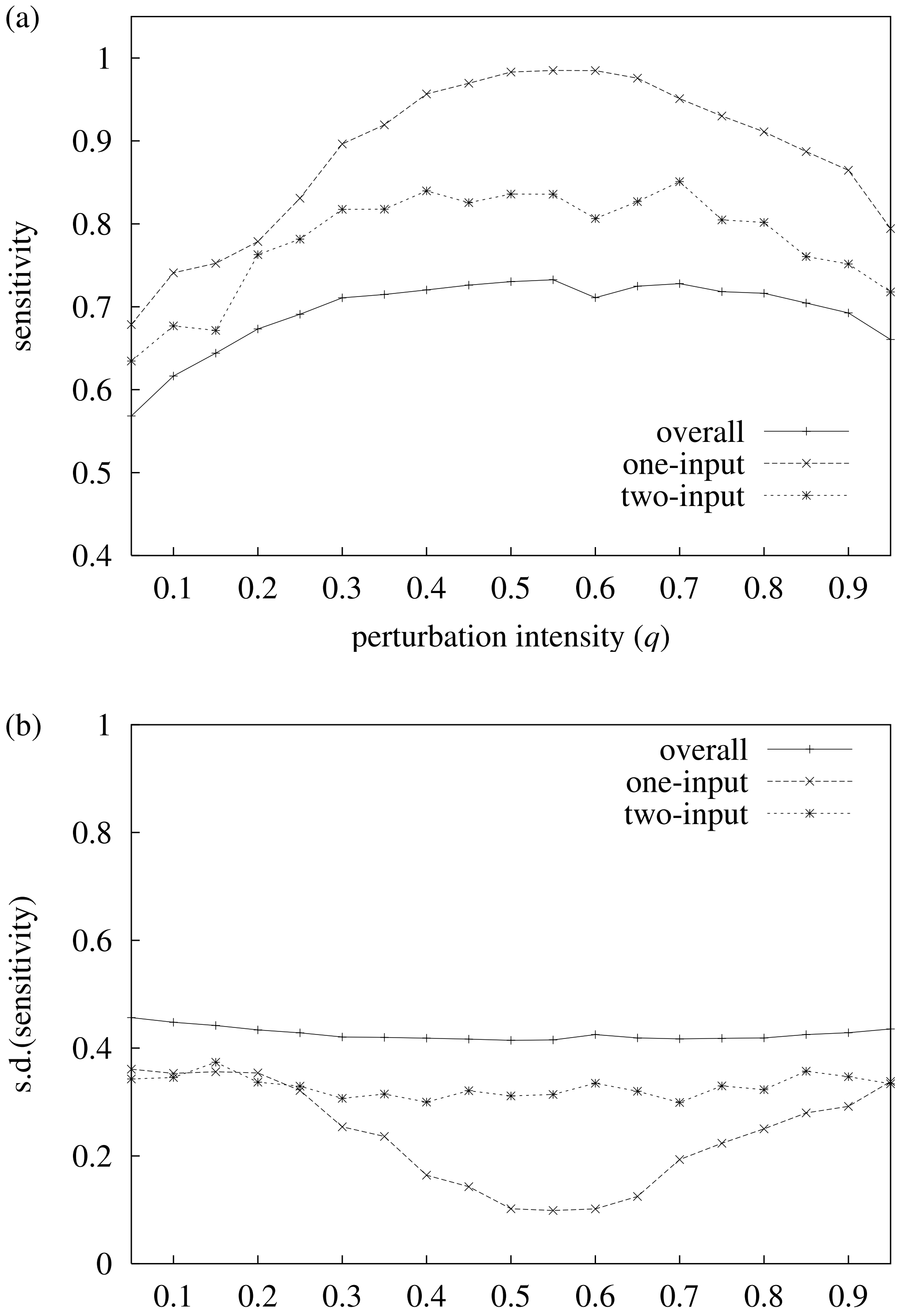

Figure 4

Sensitivity vs. *q*. (a) Average inference sensitivity vs. perturbation intensity *q*. (b) The variance (one standard deviation) is shown here separately for clarity. The results show sensitivity for (predicted) one and two-input solutions being generally higher than the overall case. The results shown are average values for 250 random networks inferred. The remaining parameters are fixed: network size *N* = 50 and *P* = 12.

Network inference method

Assuming we are given the state dynamics $s(t)$ and the default vector θ, the problem is to find the necessary model parameters (C) which will reproduce these dynamics. Specifically, a system initialised at $s(0)$ should reproduce the given dynamics $s(t)$ for $t > 0$. Note that multiple $s(t)$ expression patterns may be defined, which will be denoted as $s^r(t)$ for $r = 0, .., P$, corresponding to time series with different initial states $s^r(0)$. Our problem is to find at least one interaction matrix that will reproduce all given dynamics $s^r(t)$. The problem of finding an appropriate matrix C may be broken up into N sub-problems, since in this system, each gene i may be solved independently from the others. More precisely, the inputs to gene i (i.e. C_i, the ith row of C), can be found independently of the other genes. This reduces the search space from $O(3^{N^2})$ down to $O(N3^N)$. Each input z^i to gene i is represented as an ordered pair (j, g), $j \in \{1, .., N\}$, $g \in \{\pm 1\}$, indicating an input from gene j of sign g. A solution $\gamma(i)$ for gene i is a set of K inputs $\{z_1^i, z_2^i, ..., z_K^i\}$ (with $\gamma(i) = \phi$ if $K = 0$). For K inputs there are $\begin{pmatrix} N \\ K \end{pmatrix} 2^K$ solutions to evaluate. Starting with $K = 0$ (no inputs), we progress up to a maximum of $K = 4$, exhaustively evaluating all possible solutions for each K. However, making a parsimony assumption, if solutions are found for some $K_s < 4$, the method no longer evaluates for $K > K_s$. Note that the method does not stop as soon as a solution is found, but evaluates all possible solutions for K_s. The failure rate (percentage of genes for which no solution was found for $K \leq 4$) never exceeded 3% of the genes in any single network for which reconstruction was attempted.

Global perturbations and the perturbation intensity measure

The control time series $s^0(t)$ is generated by setting $s^0(0) = \theta$. The other time series $s^r(t)$, $r > 0$ are obtained from initial conditions which are perturbations of θ, and correspond to standard experiments such as stress conditions, or chemical treatments. Since, experimental perturbations can usually be modulated in intensity (for example, a temperature shift), this was represented using modulated artificial perturbations. Perturbed initial states $s^r(0)$ were generated by randomly changing each state $s^0(0)$ with probability q. This means that, on average, there will be qN random state differences between each perturbed initial state $s^r(0)$, and θ.

Measuring inference accuracy

Assuming one or more solutions $\gamma_1(i)$, $\gamma_2(i)$, ... are found for gene i, these are consolidated into a solution set, $Y_i = \cup_l\{\gamma_l(i)\}$. Note that some information about the solutions has been lost using this approach. For example, a solution set $Y_i^{(2)}$ obtained from a single two-input ($K = 2$) solution: $Y_i^{(2)} = \{\gamma(i)\} = \{z_1^i, z_2^i\}$, may be equal to another solution set $Y_i^{(1)}$ resulting from two single-input ($K = 1$) solutions: $Y_i^{(1)} = \{\gamma_1(i), \gamma_2(i)\}$ with $\gamma_1(i) = \{z_1^i\}$ and $\gamma_2(i) = \{z_2^i\}$.

However, this consolidation is convenient in that the solution set is easily compared with the known network structures using standard accuracy measures such as *sensitivity* and *specificity*. Here, accuracy was measured using *sensitivity*, defined as true positives / (true positives + false negatives). The relatively large number of true negatives, makes *specificity*, defined as true negatives / (true negatives + false positives), an uninformative statistic. Here, *true positives* are members of the solution set Y_i which are also true inputs (since the networks will be generated artificially, true inputs are known), and *false negatives* are those true inputs which are not members of the solution set Y_i.

Accuracy statistics were gathered from inferences performed on a large number of medium-sized random networks ($20 \leq N \leq 70$). Inferences on R random networks (each with N genes), will produce approximately RN *sensitivity* measurements (slightly fewer due to the nonzero failure rate).

Artificial gene network generation

It appears to be the case in gene networks that indegree follows an exponential distribution, whereas outdegree appears to follow a scale-free distribution. More specifically, for the yeast network, the probability distribution for indegree k follows $p_k \sim C_{in}e^{-\beta k}$ with $\beta \sim 0.45$, whereas the distribution for outdegree follows $p_k \sim C_{out}k^{-\tau}$, with $\tau \sim 1$ (C_{in}, C_{out} constants) [14]. Here, artificial gene networks [19] were created using the algorithm for generating directed graphs with arbitrary in/out degree distributions described in [20]. The exponential probability distribution for indegree k is given by:

$$p_k = (1 - e^{-\beta})e^{-\beta k},$$

where $\beta = 0.45$ is a constant. Similarly, the power law distribution (including an exponential cutoff term which is both biologically realistic and necessary analytically when $\tau < 2$ [20]) for outdegree k is described by:

$$p_k = Ck^{-\tau} e^{-\gamma k},$$

where C, γ, and $\tau = 1$ are constants. Since the algorithm begins by generating in/out-degree pairs for each node, we

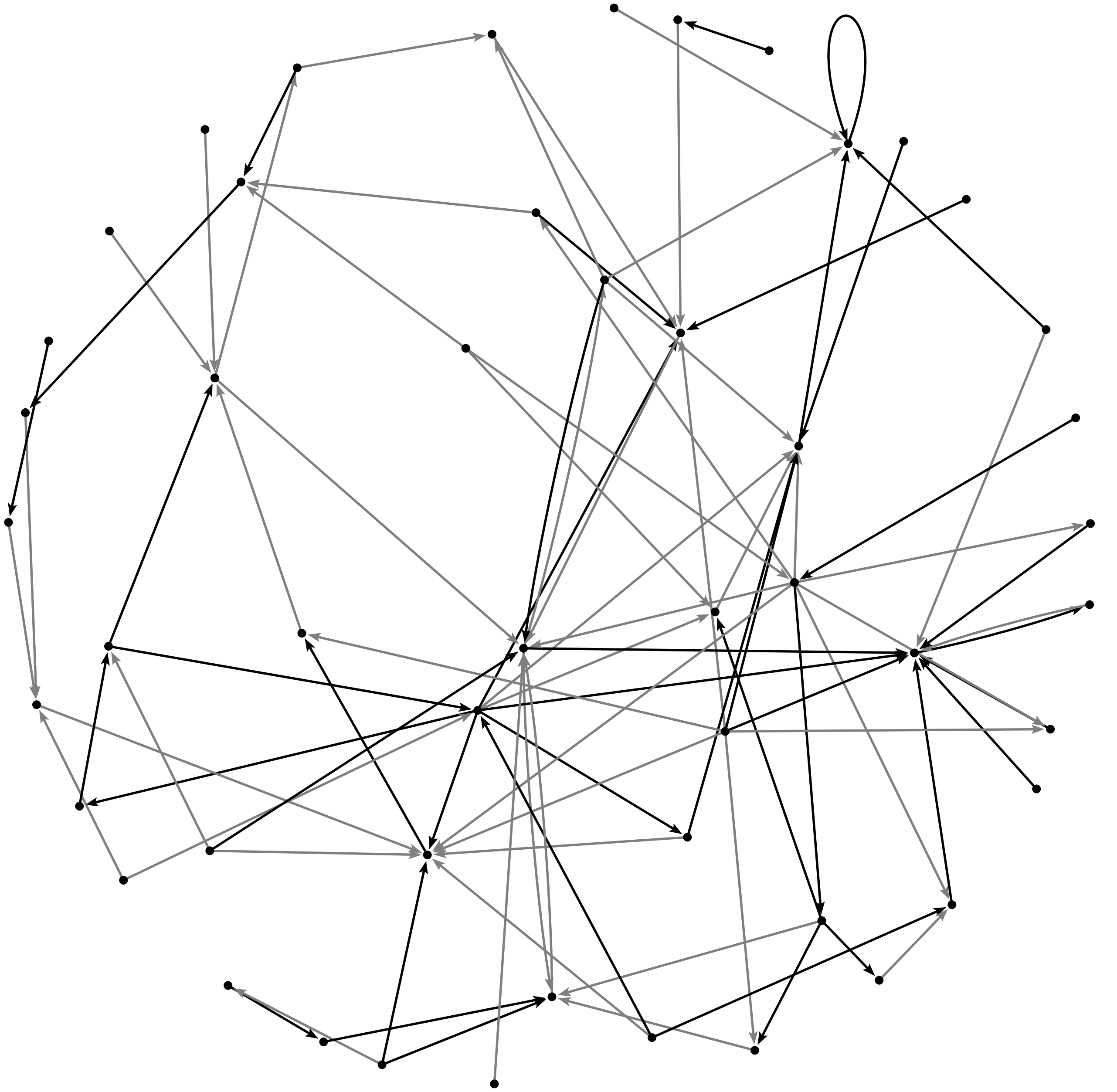

Figure 5

Example network. Example of an artificial gene network with *N* = 50. Positive interactions are shown in black, negative interactions in grey. Note the autoregulatory interaction on the upper right hand side. This diagram was generated using Pajek http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

require equal means for both indegree ($<k_{in}>$) and outdegree ($<k_{out}>$). Following [20], we obtain expressions for the mean in/out degree:

$$< k_{in} >= \frac{e^{-\beta}}{1-e^{-\beta}} \quad , \quad < k_{out} >= \frac{-e^{-\gamma}}{(1-e^{-\gamma})\ln(1-e^{-\gamma})}$$

Since β is given, we obtain a value $<k_{in}> = 1.76$, and fit the free parameter $\gamma = 0.436$ to obtain $<k_{out}> = <k_{in}>$ Since the resulting networks are unweighted, non-zero weights ($C_{ij} \in \{-1, +1\}$) are assigned at random with probability 0.5, as in [19]. It should be noted that autoregulatory interactions can be (and indeed were) generated, and that these present no particular problem for the inference method. An example of a network which was used in the analysis is shown in figure 5.

Authors' contributions

TM devised and implemented the experiments and drafted the manuscript. RS and AP supervised the project. All authors read and approved the final manuscript.

Acknowledgements

TM was supported by a grant from the UK Medical Research Council (MRC).

References

1. Cho R, Campbell M, Winzeler E, Steinmetz L, Conway A, Wodicka L, Wolfsberg T, Gabrielian A, Landsman D, Lockhart D, Davis R: **A genome-wide transcriptional analysis of the mitotic cell cycle.** *Mol Cell* 1998, **2:**65-73.
2. Spellman P, Sherlock G, Zhang M, Iyer V, Anders K, Eisen M, Brown P, Botstein D, Futcher B: **Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.** *Mol Biol Cell* 1998, **9(12):**3273-3297.
3. Tegner J, Yeung M, Hasty J, Collins J: **Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling.** *Proc Natl Acad Sci U S A* 2003, **100(10):**5944-5949.
4. Gardner T, di Bernardo D, Lorenz D, Collins J: **Inferring genetic networks and identifying compound mode of action via expression profiling.** *Science* 2003, **301(5629):**102-105.
5. Tringe S, Wagner A, Ruby S: **Enriching for direct regulatory targets in perturbed gene-expression profiles.** *Genome Biol* 2004, **5(4):**60.
6. Wagner A: **Reconstructing pathways in large genetic networks from genetic perturbations.** *J Comput Biol* 2004, **11:**53-60.
7. Kholodenko B, Kiyatkin A, Bruggeman F, Sontag E, Westerhoff H, Hoek J: **Untangling the wires: a strategy to trace functional interactions in signaling and gene networks.** *Proc Natl Acad Sci U S A* 2002, **99(20):**12841-12846.
8. Akutsu T, Miyano S, Kuhara S: **Identification of genetic networks from a small number of gene expression patterns under the Boolean network model.** *Pac Symp Biocomput* 1999:17-28.
9. Li F, Long T, Lu Y, Ouyang Q, Tang C: **The yeast cell-cycle network is robustly designed.** *Proc Natl Acad Sci U S A* 2004, **101(14):**4781-4786.
10. Serra R, Villani M, Semeria A: **Genetic network models and statistical properties of gene expression data in knock-out experiments.** *J Theor Biol* 2004, **227:**149-157.
11. MacCarthy T, Seymour R, Pomiankowski A: **The evolutionary potential of the Drosophila sex determination gene network.** *J Theor Biol* 2003, **225(4):**461-468.
12. Liang S, Fuhrman S, Somogyi R: **Reveal, a general reverse engineering algorithm for inference of genetic network architectures.** *Pac Symp Biocomput* 1998:18-29.
13. Yeung M, Tegnér J, Collins J: **Reverse engineering gene networks using singular value decomposition and robust regression.** *Proc Natl Acad Sci U S A* 2002, **99(9):**6163-6168.
14. Guelzim N, Bottani S, Bourgine P, Képès F: **Topological and causal structure of the yeast transcriptional regulatory network.** *Nat Genet* 2002, **31:**60-63.
15. Stark J, Brewer D, Barenco M, Tomescu D, Callard R, Hubank M: **Reconstructing gene networks: what are the limits.** *Biochem Soc Trans* 2003, **31(Pt 6):**1519-1525.
16. Yu H, Luscombe N, Qian J, Gerstein M: **Genomic analysis of gene expression relationships in transcriptional regulatory networks.** *Trends Genet* 2003, **19(8):**422-427.
17. Sudarsanam P, Pilpel Y, Church G: **Genome-wide co-occurrence of promoter elements reveals a cis-regulatory cassette of rRNA transcription motifs in Saccharomyces cerevisiae.** *Genome Res* 2002, **12(11):**1723-1731.
18. Covert M, Knight E, Reed J, Herrgard M, Palsson B: **Integrating high-throughput and computational data elucidates bacterial networks.** *Nature* 2004, **429(6987):**92-96.
19. Mendes P, Sha W, Ye K: **Artificial gene networks for objective comparison of analysis algorithms.** *Bioinformatics* 2003, **19(Suppl 2):**122-129.
20. Newman M, Strogatz S, Watts D: **Random graphs with arbitrary degree distributions and their applications.** *Phys Rev E Stat Nonlin Soft Matter Phys* 2001, **64(2 Pt 2):**026118-026118.