

Supporting Online Material

Plasmids, strains, growth conditions, and chemicals. The pBADX53 expression plasmid was constructed by making the following modifications to the pBAD30 plasmid obtained from American Type Culture Collection (ATCC): (i) the origin of replication was replaced with the low-copy SC101 origin of replication; (ii) the *araC* gene was removed, leaving the *araC* promoter intact; (iii) the ribosome binding site from the P_{bad} promoter in the pBAD18s (ATCC) plasmid was inserted for use with the luciferase gene in control cells; and (iv) an *n-myc* DNA fragment was inserted upstream of the *rrn T1/T2* transcription terminators to provide an alternative unique priming site for real-time PCR. Plasmids were constructed using basic molecular cloning techniques described in standard cloning manuals (1, 2). Copies of all transcripts in the SOS test network were obtained by PCR amplification of cDNA using PfuTurbo. cDNA was prepared from total RNA as described below. PCR primers included overhanging ends containing the appropriate restriction sites for cloning into the pBADX53 plasmid. Endogenous ribosome binding sites were included in the cDNA fragments for all SOS test network genes that were cloned into the pBADX53 plasmid. Sequences of the cloned SOS test network genes and their ribosome binding sites were verified using an Applied Biosystems Prism 377 Sequencer. All cloning was performed by TSS transformation (1).

The host cell for all cloning and experiments was wild-type *E. coli* strain MG1655. All cells were grown in LB medium with 50 µg/ml ampicillin, 0.5 µg/ml Mitomycin C and L-arabinose at 37±0.5°C as indicated in the main text and figures.

Antibiotics, media and chemicals were obtained from Sigma-Aldrich or Fisher Scientific, unless otherwise indicated. PfuTurbo polymerase was purchased from Stratagene. All other enzymes were purchased from New England Biolabs, unless otherwise indicated. All synthetic oligonucleotides were purchased from Integrated DNA Technologies.

RNA extraction and reverse transcription. Eight replicate *E. coli* cultures containing the pBADX53/luciferase vector (control group) and eight replicate cultures containing the pBADX53/perturbed-gene vector (perturbed group) were grown to a density of $\approx 5 \times 10^8$ cells/mL as measured by absorbance at 600nm in a Tecan SPECTRAFluor Plus plate reader (Tecan, Research Triangle Park, NC). 0.5 mL samples of each replicate culture were stabilized in 1 mL of RNAprotect Bacterial Reagent (Qiagen, Valencia, CA). Approximately 25 μ g total RNA was extracted with Qiagen RNeasy Mini spin columns using Lysozyme for bacterial cell-wall disruption. Total RNA was treated with RNase-free DNase (Ambion, Austin, TX), and its integrity was routinely verified using ethidium bromide-stained agarose gel electrophoresis. For each replicate, reverse transcription of 1 μ g total RNA was performed with 1.25 units/mL MultiScribe Reverse Transcriptase (Applied Biosystems, Foster City, CA) using 2.5 mM random hexamers in a total volume of 50 μ L, according to the manufacturer's instructions. Reactions were incubated 10 minutes at 25 °C for hexamer annealing, 30 minutes at 48 °C for reverse transcriptase elongation, and 5 minutes at 95 °C for enzyme inactivation.

Real-time quantitative PCR. Quantitative PCR primers for each transcript in the SOS test network and the normalization transcripts, *gapA* and *rrsB*, were designed using Primer Express Software v2.0 (Applied Biosystems, Foster City, CA), according to the recommendations of the manufacturer for SYBR Green detection. Primers were selected such that all amplicons were 100-107 bp, calculated primer annealing temperatures were 60 °C, and probabilities of primer-dimer/hairpin formations were minimized. DNA sequences for primer selection were obtained from the EcoGene database (<http://bmb.med.miami.edu/EcoGene/EcoWeb/>). PCR reactions were prepared using 1.4 μ L cDNA (corresponding to 30 ng of total RNA) in a total volume of 10 μ L containing 10 nM of forward and 10 nM of reverse primers and 5 μ L 2 \times SYBR Green Master Mix (Applied Biosystems, Foster City, CA). Duplicate PCR reactions were performed for each of the replicate samples. Reactions were carried out on 384-well op-

tical microplates (Applied Biosystems) using an ABI Prism 7900 for real-time amplification and SYBR Green I detection. PCR parameters were: denaturation (95 °C for 10 minutes), 40 cycles of two-segment amplification (95 °C for 15 seconds, 60 °C for 60 seconds). The thermal cycling program was concluded with a dissociation curve (60 °C ramped to 95 °C, 15 seconds at each 1 °C interval) to detect non-specific amplification or primer-dimer formation; specificity was confirmed during optimization reactions by agarose gel electrophoresis/ethidium bromide staining. All RNA extractions were checked for genomic DNA contamination by using 1 μ g total RNA in PCR reactions containing primers specific for the *gapA* and *rrsB* (16S) RNA amplicons. No-template control reactions for every primer pair were also included on each reaction plate to check for external DNA contamination.

Quantitative PCR data analysis. C_t (crossing-point threshold) and real-time fluorescence data were obtained using the ABI Prism Sequence Detection Software v2.0. Default software parameters were used except for adjustments made to the pre-exponential phase baseline used to calculate C_t for the higher abundance RNAs.

The PCR reaction efficiency of each amplicon in each reaction was calculated from the real-time fluorescence data by fitting the equation $V = E^n$ to the three data points closest to C_t , where V is the normalized fluorescence, E is the reaction efficiency, and n is the PCR cycle number. Aberrant and inefficient reactions were removed from the data set by eliminating reactions with E or C_t values outside of their joint 95% confidence interval. (Formally, the j^{th} reaction was excluded if $\text{Prob}(E_j = \text{median}(E)) \times \text{Prob}(C_{tj} = \text{median}(C_t)) < 0.05$. The errors on E and C_t are assumed to be independent.) The values of E remaining from all 32 reactions performed for each amplicon in each perturbation experiment (2 reactions/sample \times 8 samples/group \times 2 groups) were averaged. The values of C_t remaining from all 16 reactions performed for each amplicon in each experimental group in each perturbation experiment were averaged. For each gene, i , the RNA expression ratio between the perturbed and control groups

of cells were calculated from:

$$\frac{[\text{RNA}_i]^{\text{pert}}}{[\text{RNA}_i]^{\text{cont}}} = \frac{\hat{E}_i^{\hat{C}_{iu} - \hat{C}_{ip}}}{\hat{E}_r^{\hat{C}_{ru} - \hat{C}_{rp}}},$$

where

\hat{E}_i is the mean PCR efficiency for gene i ,

\hat{E}_r is the mean PCR efficiency for the *gapA* or *rrsB* normalization gene,

\hat{C}_{ip} is the mean C_t for gene i in the perturbed cell group,

\hat{C}_{iu} is the mean C_t for gene i in the control (unperturbed) cell group,

\hat{C}_{rp} is the mean C_t for the normalization gene in the perturbed cell group, and

\hat{C}_{ru} is the mean C_t for the normalization gene in the control (unperturbed) cell group.

RNA expression changes were calculated as:

$$x_i = \frac{[\text{RNA}_i]^{\text{pert}}}{[\text{RNA}_i]^{\text{cont}}} - 1,$$

and were provided to the NIR algorithm for calculation of the network model and prediction of compound bioactivity targets. The standard errors, S_{x_i} , on the expression changes, x_i , were calculated from the standard errors on \hat{E}_i , \hat{E}_r , \hat{C}_{ip} , \hat{C}_{iu} , \hat{C}_{rp} , and \hat{C}_{ru} using the propagation of error formula:

$$S_{x_i} = \sqrt{\left(\frac{\partial x_i}{\partial \hat{E}_i} S_{\hat{E}_i}\right)^2 + \dots + \left(\frac{\partial x_i}{\partial \hat{C}_{ru}} S_{\hat{C}_{ru}}\right)^2}.$$

Numerics. All computations and data analysis were performed using Matlab (Mathworks, Waltham, MA) unless otherwise specified.

NIR algorithm and computational methods. A variety of mathematical models may be used to describe genetic networks, including Boolean logic (3, 4), Bayesian networks (5), graph theory (6), and ordinary differential equations (7). Once a genetic network model has been chosen, it is possible to recover its parameters from experimental data (8–11), i.e., to infer the network.

Here we represent the network by a set of ordinary differential equations (7) describing the time evolution of the mRNA concentration of the genes in the network¹:

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \quad (1)$$

where \underline{x} represents the mRNA concentrations of the genes in the network, and \underline{u} is a set of transcriptional perturbations. We assume that the cell under investigation is at equilibrium near a stable steady-state point, and we can apply a small perturbation to each of its genes. A perturbation is small if the system returns to the original steady-state point after removal of the perturbation and if the magnitude of the response is roughly proportional to the magnitude of the perturbation. (More formally, a perturbation is small if it does not drive the network out of the basin of attraction of the stable steady-state point and if the stable manifold in the neighborhood of the steady-state point is approximately linear.) With these assumptions, we can linearize the set of nonlinear rate equations near its stable state-steady point. Thus, for each gene, i , in a network of N genes we can write the following equation:

$$\dot{x}_{il} = \sum_{j=1}^N a_{ij}x_{jl} + u_{il} = \underline{a}_i^T \cdot \underline{x}_l + u_{il}, \quad i = 1 \dots N, l = 1 \dots M, \quad (2)$$

where x_{il} is the mRNA concentration of gene i following the perturbation in experiment l ; a_{ij} represents the influence of gene j on gene i ; and u_{il} is an external perturbation to the expression of gene i in experiment l . For all N genes, Eqs. 2 can be rewritten in more compact form using matrix notation:

$$\dot{\underline{x}}_l = \mathbf{A} \cdot \underline{x}_l + \underline{u}_l, \quad l = 1 \dots M, \quad (3)$$

where \underline{x}_l is an $N \times 1$ vector of mRNA concentrations of the N genes in experiment l , \mathbf{A} is an $N \times N$ connectivity matrix, composed of elements a_{ij} , and \underline{u}_l is an $N \times 1$ vector of the perturbations applied to each of the N genes in experiment l . Inferring the network in this context means to retrieve matrix \mathbf{A} . This can be accomplished by measuring the mRNA

¹(from now on we will use the following notation: \underline{x} represents a column vector, \underline{x}^T is a row vector, x is a scalar and \mathbf{A} is a matrix)

concentrations of all the N genes at steady state (i.e., $\dot{x}_l = 0$) in M experiments and solving the system of equations:

$$\mathbf{A} \cdot \mathbf{X} = -\mathbf{U}, \quad (4)$$

where \mathbf{X} is an $N \times M$ matrix composed of columns \underline{x}_l ; \mathbf{U} is an $N \times M$ with each column, \underline{u}_l . Equation 4 can be solved only if $M \geq N$. However, the recovered weights, \mathbf{A} , will be extremely sensitive to noise both in the data, \mathbf{X} , and in the perturbations, \mathbf{U} , and thus unreliable unless we overdetermine the system of Eqs. 4. This can be accomplished either by increasing the number of experiments ($M > N$), or, by assuming the maximum number of regulators acting on each gene, k , is less than M (i.e., the network is not fully connected (11–13)), thus reducing the number of weights a_{ij} to be recovered.

The algorithm solves Eqs. 4 using multiple linear regression (14, 15) with the following hypotheses: (i) the data \mathbf{X} are stochastic variables normally distributed with known variances; (ii) the perturbations, \mathbf{U} , are stochastic variables normally distributed with known variances; and (iii) the solution is the one that minimizes the least-squared error.

For each row of \mathbf{A} , the algorithm computes the least-squared solution for all possible combinations of k out of N regulatory inputs per gene. It then selects the solution with the least-squared error as the best approximation to the solution of Eqs. 4. The algorithm also returns the significance of the regression (goodness of fit) and the standard error of each of the recovered weights. A lack of significance of the regression for a given gene implies that its main regulators must lie outside of the set of genes included in the model.

Algorithm in detail. A genetic network can be described by the system of linear differential equations, Eqs. 2. For each gene i at steady state ($\dot{x}_{il} = 0$) in experiment l , we can therefore write:

$$-u_{il} = \underline{a}_i^T \cdot \underline{x}_l, \quad (5)$$

where u_{il} is the transcriptional perturbation applied to gene i in experiment l , \underline{a}_i^T is a row of \mathbf{A} , and \underline{x}_l ($N \times 1$) are the mRNA concentrations at steady state following the perturbation in experiment l . The algorithm assumes that only k out of the N weights in \underline{a}_i for gene i are different from zero. Hence, for each possible combination of k out of N weights, the algorithm computes the solution to the following linear regression model:

$$y_{il} = \underline{b}_i^T \cdot \underline{z}_l + (\epsilon_{il} - \underline{b}_i^T \cdot \underline{\gamma}_l), \quad (6)$$

where $y_{il} = -u_{il}$ is the perturbation applied to gene i in experiment l ; ϵ_{il} represents normally distributed (zero mean) measurement noise on the perturbation of gene i in experiment l ; \underline{b}_i is a $k \times 1$ vector representing one of (N choose k) possible combinations of the elements of \underline{a}_i ; $\underline{z}_l = \underline{x}_l + \underline{\gamma}_l$ is a $k \times 1$ vector of mRNA concentrations (corresponding to the k elements selected from \underline{a}_i) that result from the perturbation in experiment l ; and $\underline{\gamma}_l$ represents normally distributed (zero mean) measurement noise on the mRNA concentrations in experiment l . Equation 6 represents a multiple linear regression model with noise $\eta_{il} = \epsilon_{il} - \underline{b}_i^T \cdot \underline{\gamma}_l$, with zero mean and variance:

$$\text{var}(\eta_{il}) = \sum_{j=1}^k b_{ij}^2 \text{var}(\gamma_{jl}) + \text{var}(\epsilon_{il}) \quad (7)$$

(if ϵ_{il} and $\underline{\gamma}_l$ are uncorrelated).

If we collect data in M different experiments, then we can write Eq. 6 for each experiment and obtain the system of equations:

$$\underline{y}_i^T = \underline{b}_i^T \cdot \mathbf{Z} + \underline{\eta}_i^T, \quad (8)$$

where \underline{y}_i is an $M \times 1$ vector of measurements of y_{il} in the M experiments; \mathbf{Z} is a $K \times M$ matrix, where each column is the vector \underline{z}_l for one of the M experiments; and $\underline{\eta}_i$ is an $M \times 1$ vector of noise in the M experiments. From Eqs. 8, it follows that a predictor for \underline{y}_i given the data matrix \mathbf{Z} is:

$$\hat{\underline{y}}_i^T = \tilde{\underline{b}}_i^T \cdot \mathbf{Z}, \quad (9)$$

where \tilde{b}_i is an estimate of b_i , the true model weights. To find the best estimate, we search for values of \tilde{b}_i that minimize the sum squared errors (SSE) cost function:

$$\text{SSE}_i^k = \sum_{l=1}^M (y_{il} - \hat{y}_{il})^2 = \sum_{l=1}^M (y_{il} - \tilde{b}_i^T \cdot \underline{z}_l)^2. \quad (10)$$

The solution is (14):

$$\tilde{b}_i = (\mathbf{Z} \cdot \mathbf{Z}^T)^{-1} \cdot \mathbf{Z} \cdot \underline{y}_i. \quad (11)$$

If some of the regressors in \mathbf{Z} are colinear, it may be necessary to use the ridge regression technique (14). In addition, we note that \tilde{b}_i in Eq. 11 is not the maximum likelihood estimate for the parameters b_i when the regressors \mathbf{Z} are stochastic variables, but nevertheless it is a good estimate provided that the variation in \mathbf{Z} is sufficiently large (14, 15).

We calculate \tilde{b}_i for each of the (N choose k) combinations of weights for gene i . We then select the estimate, \tilde{b}_i with the smallest sum squared errors as the best approximation of a_i in Eqs. 2.

We now estimate the variance on the estimated parameters \tilde{b}_i . Because the noise is uncorrelated and normally distributed with zero mean, the covariance matrix is given by (9):

$$\text{Cov}(\tilde{b}_i) = (\mathbf{Z} \cdot \mathbf{Z}^T)^{-1} \cdot \mathbf{Z} \cdot \Sigma_\eta \cdot \mathbf{Z}^T \cdot (\mathbf{Z} \cdot \mathbf{Z}^T)^{-1}, \quad (12)$$

where Σ_η is an $M \times M$ diagonal matrix with diagonal elements equal to the noise variance for gene i in the M experiments, $\text{var}(\eta_{i1}), \dots, \text{var}(\eta_{im})$. We assume that we can estimate the noise variances by substituting the parameters \tilde{b}_i estimated with Eq. 11 into Eq. 7:

$$\text{var}(\eta_{il}) = \sum_{j=1}^k \tilde{b}_{ij}^2 \text{var}(\gamma_{jl}) + \text{var}(\epsilon_{il}), \quad (13)$$

Testing the significance of the regression. The significance of the regression determines if there exists a linear relationship between y_i , the dependent variable in the regression, and z_j , the regressor variables. The test discriminates between two hypotheses:

$$\begin{aligned} H_0 &: b_j = 0 \quad j = 1, \dots, k, \\ H_1 &: b_j \neq 0 \quad j = 1, \dots, k. \end{aligned} \quad (14)$$

Rejection of the null hypothesis, H_0 , implies that at least one regressor z_j contributes significantly to the model, i.e., the fit of the model to the data is significant. The appropriate test statistic is (16):

$$F = \frac{(\text{SSE}_0 - \text{SSE}_k)/k}{\text{SSE}_k/(M - k)}, \quad (15)$$

where $\text{SSE}_0 = \sum_l (y_{il})^2$ is the sum of the squared errors for the H_0 regression model (i.e., no regressor variables), and $\text{SSE}_k = \sum_l (y_{il} - \hat{y}_{il})^2$ is the sum of squared errors for the H_1 regression model (i.e., k regressor variables). Both $\text{SSE}_0/\text{var}(\eta)$ and $\text{SSE}_k/\text{var}(\eta)$ follow a χ^2 -distribution. Thus, F follows an F -distribution, the ratio of two χ^2 -distributions. The F statistic may also be obtained from R^2 , the coefficient of determination for the regression model (16):

$$F = \frac{(M - k)R^2}{k(1 - R^2)}, \quad (16)$$

where $R^2 = 1 - \text{SSE}_k/\text{SSE}_0$. If F exceeds the acceptance threshold, F^* , then sum squared errors using model H_1 is significantly smaller than the sum squared errors using model H_0 . Thus, we reject model H_0 . F^* is determined for a desired confidence level from an F -distribution with k and $M - k$ degrees of freedom. (Note, finding a statistically significant fit for a regression model with k regressor variables does not imply that k variables provides the optimal fit. The optimal number of regressors must be determined from additional criteria, such as the residual mean square criterion (14), or those described below.)

Treatment of missing data points. Our algorithm can also be modified to recover the network when some of its genes have not been perturbed. If for example, gene i has not been perturbed, then the i_{th} row of matrix \mathbf{U} will be null and Eqs. 4 for gene i become:

$$\underline{a}_i^T \cdot \mathbf{X} = -\underline{u}_i^T = 0, \quad (17)$$

where \underline{a}_i^T is the i_{th} row of matrix \mathbf{A} , and \underline{u}_i^T is the i_{th} row of matrix \mathbf{U} . The trivial solution to Eq. 17 is $\underline{a}_i = 0$, i.e., gene i is not regulated. To recover a non-trivial solution, we followed

an alternative approach. We can rewrite Eq. 17 by dividing all the coefficients a_i by the self regulation of gene i , a_{ii} . The new vector, $\underline{\alpha}_i$, will have its n_{th} element equal to a_{in}/a_{ii} , and hence its i_{th} element equal to 1. We can then rewrite Eq. 17 as:

$$\sum_{j=1, j \neq i}^N \alpha_{ij} x_{jl} = -x_{il}, \quad l = 1 \dots M. \quad (18)$$

Equations 18 now can be solved in the same way as Eqs. 5. However, the magnitude of the recovered weights α_{ij} will be relative to the self-regulation weight, a_{ii} .

Target prediction. The targets of a chemical compound, or some other perturbation with unknown targets, can be computed from measurements of the expression response of the cell to that perturbation using:

$$\hat{\underline{u}}_p = -\tilde{\mathbf{A}} \cdot \underline{x}_p, \quad (19)$$

where \underline{x}_p is a vector of the expression changes of the species in the network following the perturbation; $\tilde{\mathbf{A}}$ is the estimated network model; and $\hat{\underline{u}}_p$ is a vector of predicted values of the perturbation. The variance on the predicted perturbation of gene i can be computed as (14):

$$var(\hat{u}_{pi}) = \underline{x}_p^T \cdot (\mathbf{Z} \cdot \mathbf{Z}^T)^{-1} \cdot \mathbf{Z} \cdot \Sigma_{\eta} \cdot \mathbf{Z}^T \cdot (\mathbf{Z} \cdot \mathbf{Z}^T)^{-1} \cdot \underline{x}_p + \sum_{j=1}^k \tilde{b}_{ij}^2 var(x_{pj}). \quad (20)$$

The statistical significance of each predicted perturbation can be estimated using the calculated variances. Genes with statistically significant values in $\hat{\underline{u}}_p$ are considered to be targets of the chemical compound or other perturbation with unknown targets.

Remarks on determining the optimal connectivity. The connectivity (k) chosen for the model can affect the significance of the fit of the model to the data, the dynamic stability of the model, the number of regulatory interactions correctly recovered (coverage), and the number of false interactions recovered (false positives).

We calculated the significance of the fit of the network model to the data, as described above, for $k = \{3, 4, 5, 6\}$ and significant fits were obtained for $k = 4$, $k = 5$, and $k = 6$ (see

Table S3). However, we did not obtain a significant fit for regulatory inputs to the *recF* gene for any value of k . This suggests that, under the growth conditions used in the experiments, *recF* is not significantly regulated by any of the genes included in the test network. Therefore, input connections to *recF* were set to zero in the network model.

We next tested the dynamic stability of the recovered networks. In dynamically stable linear networks, the expression levels of the network species will eventually settle over time to steady state. In a dynamically unstable network, expression levels will continue to grow without limit over time. Because we measured our test network in steady state, we know that the experimental network is dynamically stable. Therefore, the solved network model must also be dynamically stable. Only models with $k = 5$ and $k = 6$ were dynamically stable.

Next we examined the effect of various values of k on the coverage and false positives in the solved network model. To this end, we simulated networks of 9 transcripts with an average of 5 input connections per gene. We used the NIR method to solve for network models of the simulated networks using three different values of k (Fig. S2). As k was decreased, we found that the fraction of false positives decreases by a larger relative amount than the coverage of correct connections. Thus, to obtain the best balance between coverage and false positives, we selected the model solved with $k = 5$ for further analysis.

As described in the main text and the supplement, the use of $k < N$ enables the solution of accurate network models even with high noise and small data sets. However, a full set of model parameters ($k = N$) may be used with no alterations to the NIR or experimental approach. In this case, a researcher must only collect a larger number of data points. Resource limitations may make this difficult for some researchers. But for many research groups and institutions (e.g., biotech and pharmaceutical companies with the ability to perform thousands of microarray experiments per month), this increased data requirement will present no difficulties. Nevertheless, we view the use of reduced parameters to be a significant advantage because it yields the best model given the available data. If large data sets are available, then, large values

of k may prove effective.

Remarks on target prediction To confirm our experimental results which showed our solved model has high predictive power, we performed simulations of random networks composed of 9 transcripts. We used the NIR method to obtain network models from the simulated network data, and tested the ability of the model to predict perturbations, as was done for the experimental network. Consistent with our results from the experimental data from the SOS test network, the results from simulated data show that the network model can identify perturbed genes with high coverage and specificity even at high levels of measurement noise (Fig. S4).

To further evaluate the predictive power of method, we tested a worst case scenario in which the model was recovered using a seven-perturbation training set that excluded the *lexA* and *recA* training perturbations. The ability of the reduced model to predict transcriptional targets was nearly as good as the model recovered using a full training set (Fig. S3). For the MMC perturbation, it again identified *recA* as a target, and it also identified two false targets, *umuDC* and *lexA* ($1/1 = 100\%$ coverage, $6/8 = 75\%$ specificity). For the *lexA-recA* double perturbation, it identified *lexA* but not *recA* as a target with no false positives ($1/2 = 50\%$ coverage, $7/7 = 100\%$ specificity). These results agree with simulations showing that a reduced model retains high coverage and specificity in predicting perturbation targets, albeit slightly reduced from that of a full model (Fig. S4).

In the past, a large compendium of transcriptional responses to genetic perturbations, combined with pairwise clustering, has been used to identify gene targets of pharmacological compounds (17). Although this method is successful under certain conditions, it can break down when a compound's bioactivity is mediated by multiple interacting genes or pathways, or when a perturbation to the targeted gene or pathway is not represented in the compendium. Moreover, it cannot differentiate between genes that are highly interconnected in a pathway. As shown in Fig. S5, neither pairwise hierarchical clustering nor pairwise correlation unambiguously identi-

fied the targets of MMC activity in the test network.

Because we measured only transcriptional changes, protein and metabolite species cannot be explicitly represented in our network model. Consequently, the network model can specifically identify only the transcriptional targets of a compound's bioactivity, but not the protein or metabolite targets of a compound. Nevertheless, using biological databases, the protein or metabolite regulators of the transcripts can be identified. With modest additional experimental effort, such regulators can be confirmed as the true targets. Thus, the network model can accelerate the identification of protein and metabolite targets of a compound, even when proteins and metabolites are not explicitly represented. In addition, with advances in high-throughput protein and metabolite assay technologies, it may soon be possible to explicitly include protein and metabolite species in the model.

Remarks on noise and error. The recovered network model, A , is a linear representation of a nonlinear system. Nonlinear behaviors that are sometimes exhibited by gene, protein, and metabolite networks, including bifurcations, thresholds, and multistability, cannot be described by A . Nevertheless, the linear approximation is topologically equivalent to the nonlinear system near a steady-state point. Therefore, to apply the NIR algorithm, it is necessary to remain near a single steady state during the course of all experiments. From a practical perspective, this means that cells must be maintained under consistent and constant environmental and physiological conditions, and the applied perturbations must be relatively small. If these conditions are not met, the recovered model may contain a certain degree of nonlinear error, or, in the extreme, it may not be possible to adequately fit a linear model.

In practice, it is generally straightforward to keep the cells in a constant environmental and physiological state, but due to the presence of measurement noise, it can be challenging to meet the condition of small perturbations. For errors due to noise, we can improve the signal-to-noise ratio (S/N) by increasing the size of the perturbations. However, larger perturbations can lead

to larger nonlinear errors. Thus, the experimenter must identify an acceptable balance between noise and nonlinear error. In practice, one should identify a measurement technology, a number of replicates, and a perturbation level that provides noise levels comparable to or better than that obtained in this work ($S/N = 1.5$).

Remarks on scalability. One of the more promising aspects of this method, from a practical standpoint, is its scalability. Computationally, the NIR algorithm is easily applied to larger networks. Experimentally, the scalability of the method depends primarily on the speed with which perturbations can be delivered. The perturbations use transcriptional overexpression and are delivered from episomal expression plasmids. (In general, we expect overexpression alone will be sufficient for model recovery, except in special cases where transcription of a gene is saturated in the baseline state.) Thus, the perturbations are easily applied to any gene and they require no labor intensive and physiologically unpredictable chromosomal modifications. In this study, the experiments were performed at a rate of approximately one new perturbation per day. With process improvements, this rate could be further improved. The simplicity of the perturbation approach also suggests that the method may be extended to eukaryotic cell lines.

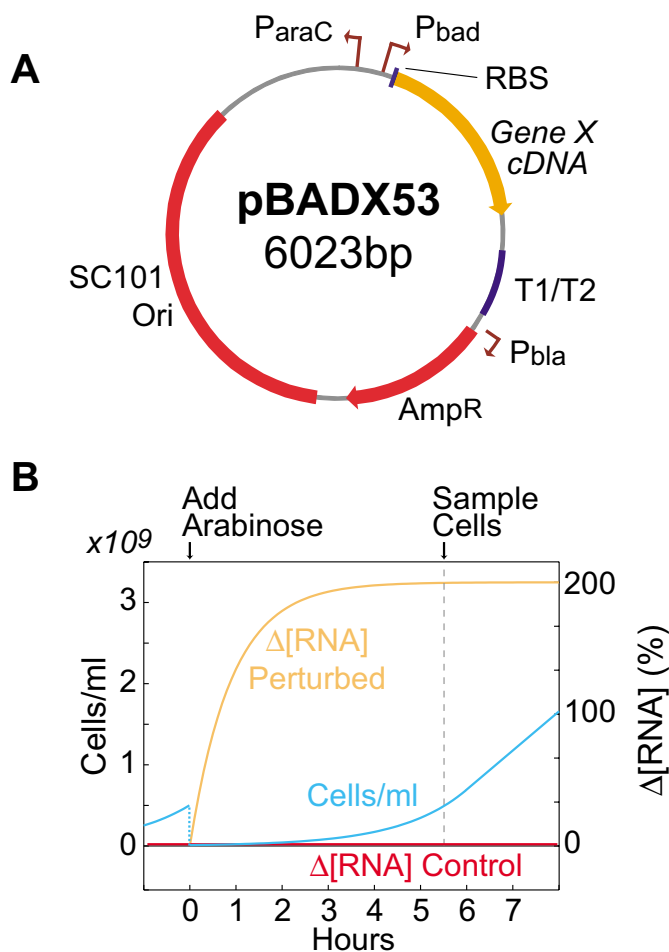


Figure S1: Experimental methods. **(A)** pBADX53 expression plasmid used to perturb expression of transcripts in the test network, where gene X is one of the nine test-network genes. The endogenous ribosome binding site (RBS) for each gene X is included in the plasmid. **(B)** Experimental design. In the baseline condition, batch cultures of cells containing the pBADX53 plasmid were maintained in exponential growth in LB medium with 0.5 $\mu\text{g/ml}$ MMC and 50 $\mu\text{g/ml}$ Ampicillin (to maintain plasmid survival). MMC is a highly specific DNA-damaging agent and was applied to ensure moderate activation of the SOS response. One group of cells (the perturbed group) was grown in the baseline condition with the pBADX53 plasmid coupled to one of the test-network genes. A second group of cells (the control group) was grown in the baseline condition with the pBADX53 plasmid coupled to the luciferase reporter gene. Transcriptional perturbations were then induced by adding an amount of arabinose sufficient to induce expression of the perturbed gene at levels typically 100-500% in excess of endogenous expression levels. Although arabinose was added to both the perturbed and control cell groups, the luciferase gene does not interact with the SOS pathway. Thus, luciferase RNA was used to estimate the level of overexpression of the perturbed gene. RNA expression ratios ($[\text{RNA}]_{\text{perturbed}}/[\text{RNA}]_{\text{control}}$) were assayed using real-time PCR and the *gapA* or *16s* gene as a normalization reference (18).

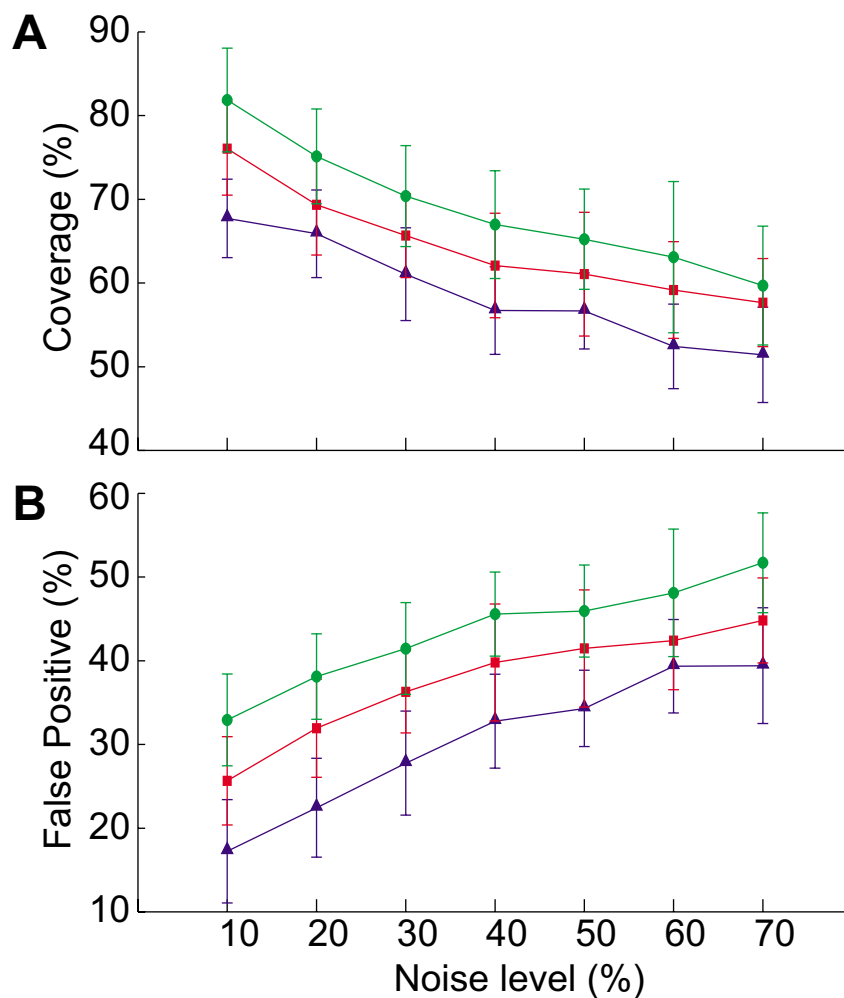


Figure S2: Effect of k on the recovery of randomly connected networks of nine genes with an average of five regulatory inputs per gene. Simulated perturbations of magnitude $u_i = 1$ (arbitrary units) were applied to fifty such randomly connected networks. For each perturbation to each random network, the mRNA concentrations at steady state were calculated, and normally-distributed, uncorrelated noise was added both to the mRNA concentrations and to the perturbations to represent measurement error. The noise (noise on $x = S_x/x$, where S_x is the standard error on the mean of x , μ_x) on the perturbations was set to 20% (equivalent to that observed on perturbations in our experiments). The noise on the mRNA concentrations was varied from 10% to 70%. Coverage and false positives were calculated for $k = 6$ (green circles), $k = 5$ (red squares), $k = 4$ (blue triangles) in the NIR algorithm.

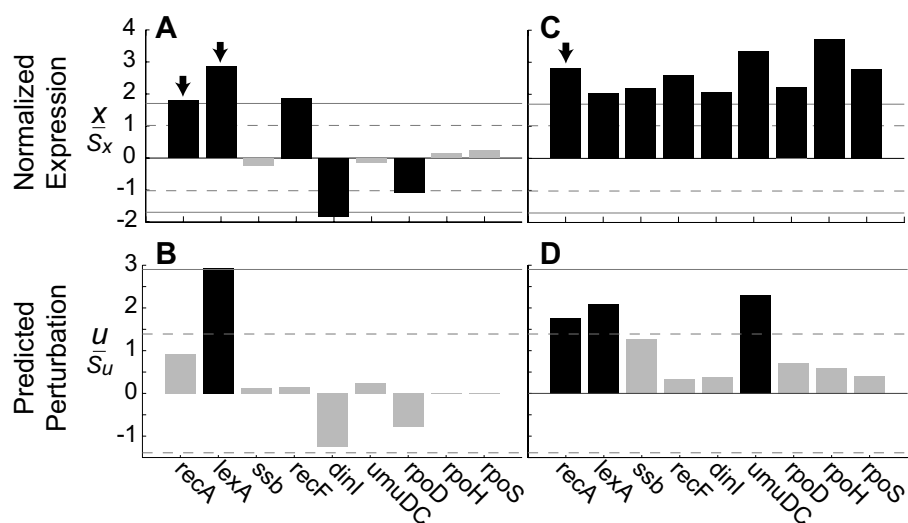


Figure S3: Identification of perturbed genes using a network model solved with an incomplete training data set that excluded the *lexA* and *recA* perturbations. Cells were perturbed either with a *lexA-recA* double perturbation or MMC. The mean relative expression changes (x), normalized by their standard deviations (S_x), are illustrated for the *lexA-recA* double perturbation (**A**) and the MMC perturbation (**C**). Arrows indicate the genes known to be targeted by the perturbation. Predicted perturbations in the *lexA-recA* experiment (**B**) and the MMC experiment (**D**) were calculated from the expression data in **A** and **C** using the SOS model solved with the seven-perturbation training set (18). The predicted perturbations to each gene (u) were normalized by their standard deviations (S_u) to determine statistical significance. In all panels, black bars indicate statistically significant, and grey bars indicate statistically non-significant. Horizontal lines denote significance levels: $P = 0.3$ (dashed), $P = 0.1$ (solid).

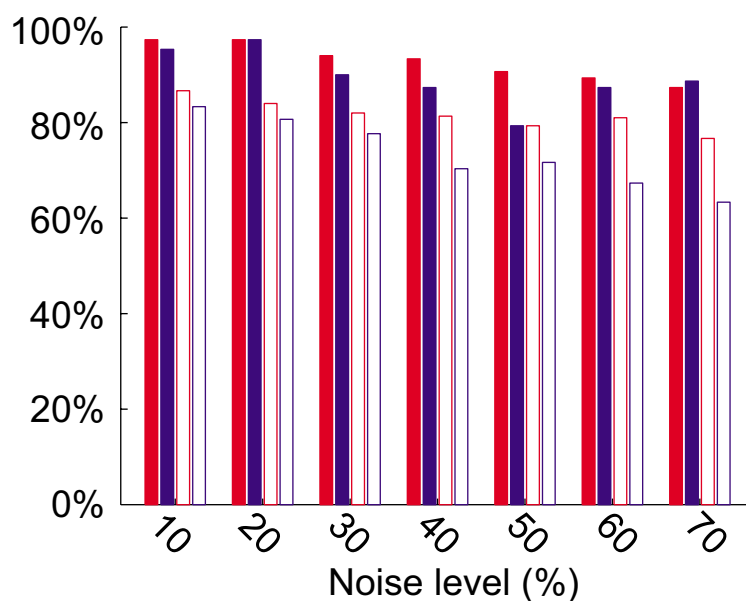


Figure S4: Perturbation recovery performance for simulated networks. Three randomly selected genes were perturbed in each of fifty randomly connected networks of nine genes with an average of five regulatory inputs per gene. For each perturbation to each random network, the mRNA concentrations at steady state were calculated, and normally-distributed, uncorrelated noise was added both to the mRNA concentrations and to the perturbations to represent measurement error. The noise ($\text{noise} = S_x/\mu_x$, where S_x is the standard deviation of the mean of x , μ_x) on the perturbations was set to 20% (equivalent to that observed on perturbations in our experiments). The noise on the mRNA concentrations was varied from 10% to 70%. Coverage (genes correctly identified as perturbed by the network model / total number of perturbed genes) and specificity (genes correctly identified as unperturbed by the network model / total number of unperturbed genes) were calculated for models recovered using a nine-perturbation training set (red bars) and a seven-perturbation training set (blue bars). Solid bars denote coverage; open bars denote specificity.

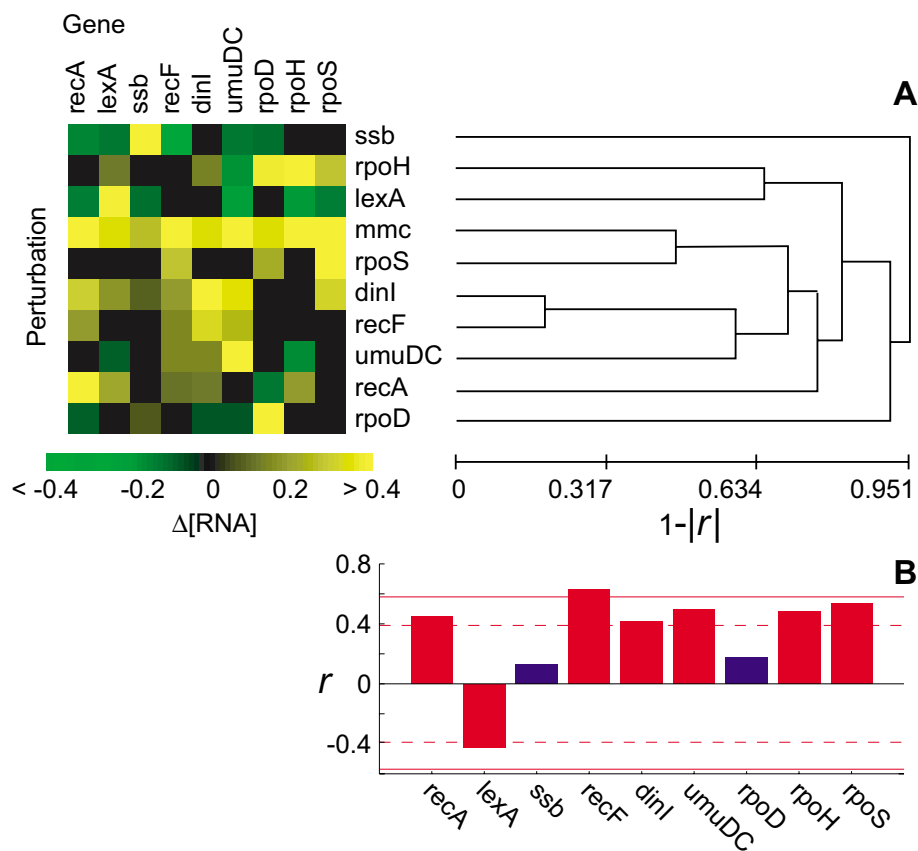


Figure S5: Performance of clustering and correlation for identifying perturbed genes. **(A)** Expression profiles for the MMC perturbation and all perturbations in the training set are compared using average-linkage clustering with the absolute linear uncentered correlation metric (i.e., $1 - |r|$ where r is the uncentered correlation coefficient) (19). The MMC perturbation profile is incorrectly clustered with the *rpoS* perturbation profile. **(B)** Pair-wise correlation of the MMC perturbation profile with each perturbation in the training set. All but two perturbations show statistically significant correlation with the MMC perturbation. Red bars indicate statistically significant; blue bars indicate statistically non-significant. Red lines denote significance levels: $P = 0.3$ (dashed), $P = 0.1$ (solid).

	<i>recA</i>	<i>lexA</i>	<i>ssb</i>	<i>recF</i>	<i>dinI</i>	<i>umuDC</i>	<i>rpoD</i>	<i>rpoH</i>	<i>rpoS</i>
<i>recA</i>	-0.597	-0.179	-0.010	0	0.096	0	-0.011	0	0
<i>lexA</i>	0.387	-1.670	-0.014	0	0.087	-0.068	0	0	0
<i>ssb</i>	0.044	-0.189	-1.275	0	0.053	0	0.027	0	0
<i>recF</i> [†]	-0.1808	0.2377	-0.0251	-1	-0.0554	0	0	0	0.39
<i>dinI</i>	0.281	0	0	0	-2.094	0.156	-0.037	0.012	0
<i>umuDC</i>	0.112	-0.403	-0.016	0	0.205	-1.147	0	0	0
<i>rpoD</i>	-0.171	0	-0.017	0	0.025	0	-1.513	0.021	0
<i>rpoH</i>	0.096	0	0.001	0	-0.009	-0.031	0	-0.483	0
<i>rpoS</i>	0.217	0	0	-1.678	0.672	0	0.077	0	-3.921

Table S1: Recovered model (A). Each row in the matrix shows the influences of the genes listed in the columns on the gene in the row. The values on the diagonal represent self-feedback. A positive self-feedback is any value greater than -1, a negative feedback is any value less than -1. [†]Indicates statistically non-significant fit for the row.

	<i>recA</i>	<i>lexA</i>	<i>ssb</i>	<i>recF</i>	<i>dinI</i>	<i>umuDC</i>	<i>rpoD</i>	<i>rpoH</i>	<i>rpoS</i>
<i>recA</i>	0.199	0.176	0.006	0	0.039	0	0.013	0	0
<i>lexA</i>	0.248	0.859	0.015	0	0.081	0.084	0	0	0
<i>ssb</i>	0.118	0.307	0.087	0	0.043	0	0.025	0	0
<i>recF</i> [†]	0.189	0.352	0.011	0	0.072	0	0	0	0.236
<i>dinI</i>	0.243	0	0	0	0.583	0.113	0.046	0.011	0
<i>umuDC</i>	0.150	0.405	0.013	0	0.091	0.311	0	0	0
<i>rpoD</i>	0.122	0	0.013	0	0.066	0	0.336	0.011	0
<i>rpoH</i>	0.047	0	0.005	0	0.015	0.024	0	0.134	0
<i>rpoS</i>	0.470	0	0	1.765	0.355	0	0.112	0	1.794

Table S2: Standard errors on weights of recovered model (A). [†]Indicates statistically non-significant fit for the row.

	Row of A								
	<i>recA</i>	<i>lexA</i>	<i>ssb</i>	<i>recF</i>	<i>dinI</i>	<i>umuDC</i>	<i>rpoD</i>	<i>rpoH</i>	<i>rpoS</i>
$k = 4$	0.0109	0.1654	0.0000	0.9841	0.0023	0.0023	0.0002	0.0000	0.1067
$k = 5$	0.0094	0.1675	0.0000	0.9846	0.0022	0.0013	0.0002	0.0000	0.0960
$k = 6$	0.0077	0.1242	0.0000	0.9852	0.0020	0.0012	0.0002	0.0000	0.0891

Table S3: Significance of regression for the weights in each row of A. The values in the table are the P -values of the fit, i.e., the probability that all the fitted weights for that row of A are zero.

	<i>recA</i>	<i>lexA</i>	<i>ssb</i>	<i>recF</i>	<i>dinI</i>	<i>umuDC</i>	<i>rpoD</i>	<i>rpoH</i>	<i>rpoS</i>
<i>recA</i>	+	-	-	+	+	-	+	0	0
<i>lexA</i>	+	-	-	+	+	-	+	0	0
<i>ssb</i>	+	-	-	+	+	-	+	0	0
<i>recF</i>	0	0	0	0	0	0	+	0	+
<i>dinI</i>	+	-	-	+	+	-	+	0	0
<i>umuDC</i>	+	-	-	+	+	-	+	0	0
<i>rpoD</i>	+	-	-	+	+	-	+	+	0
<i>rpoH</i>	0	0	0	0	0	0	+	+	0
<i>rpoS</i>	0	0	0	0	0	0	+	0	+

Table S4: Known regulatory interactions in the SOS test network, which were derived from published literature, as explained in the main text. +, -, or 0 indicates a positive, negative, or no regulatory input from the gene in the column to the gene in the row.

	<i>recA</i>	<i>lexA</i>	<i>ssb</i>	<i>recF</i>	<i>dinI</i>	<i>umuDC</i>	<i>rpoD</i>	<i>rpoH</i>	<i>rpoS</i>
Gain Matrix (G)									
<i>recA</i>	-	-17.49	-1.08	0.00	6.75	1.95	-1.39	0.10	0.00
<i>lexA</i>	38.01	-	-0.89	0.00	3.8	-2.82	-0.39	0.07	0.00
<i>ssb</i>	0.43	-9.11	-	0.00	1.72	0.77	1.37	0.10	0.00
<i>recF</i>	0.00	0.00	0.00	-	0.00	0.00	0.00	0.00	0.00
<i>dinI</i>	22.43	-4.05	-0.20	0.00	-	6.92	-1.37	1.14	0.00
<i>umuDC</i>	6.23	-22.19	-0.94	0.00	8.11	-	-0.26	0.19	0.00
<i>rpoD</i>	-17.31	1.99	-0.75	0.00	0.03	-0.19	-	2.86	0.00
<i>rpoH</i>	31.02	-2.00	0.01	0.00	-0.06	-5.50	-0.23	-	0.00
<i>rpoS</i>	12.35	-1.62	-0.11	-42.8	8.82	1.29	0.98	0.26	-
Mean(G_j)	14.20	6.49	0.44	4.76	3.25	2.16	0.67	0.52	0.00

Table S5: Gain Matrix, $\mathbf{G} = -\mathbf{A}^{-1}$, for the recovered model (as % change in expression). Each column shows the percentage change in expression of genes following a 100% perturbation of the expression of the gene indicated in the column. The mean(|G_j|) is the mean of all absolute responses to the perturbation of gene, *j*, indicated by the column. Self-feedback effects were not included in the calculation of mean(|G_j|). We considered major regulators to be those transcripts that, when perturbed, cause mean changes in expression of the other genes in the network. The gain matrix correctly identifies *recA* and *lexA* as the major regulators in the network.

Genes	Training Perturbations									Test Perturbations	
	<i>recA</i>	<i>lexA</i>	<i>ssb</i>	<i>recF</i>	<i>dinI</i>	<i>umuDC</i>	<i>rpoD</i>	<i>rpoH</i>	<i>rpoS</i>	double	MMC
<i>recA</i>	0.906	-0.132	-0.139	0.187	0.291	-0.061	-0.077	-0.017	-0.025	0.313	0.496
<i>lexA</i>	0.212	0.383	-0.117	0.064	0.169	-0.087	0.039	0.125	0.084	0.688	0.321
<i>ssb</i>	0.018	-0.107	10.524	0.061	0.080	0.013	0.064	0.089	-0.070	-0.028	0.251
<i>recF</i>	0.104	-0.050	-0.273	0.139	0.180	0.146	0.069	-0.004	0.275	0.441	0.523
<i>dinI</i>	0.119	-0.097	0.056	0.315	2.147	0.142	-0.068	0.135	0.113	-0.240	0.334
<i>umuDC</i>	0.076	-0.189	-0.124	0.250	0.347	2.017	-0.067	-0.172	-0.022	-0.022	0.834
<i>rpoD</i>	-0.122	-0.047	-0.102	-0.107	-0.011	0.104	3.068	0.365	0.217	-0.139	0.327
<i>rpoH</i>	0.178	-0.183	0.036	-0.070	-0.034	-0.155	0.008	26.633	0.087	0.026	0.786
<i>rpoS</i>	0.072	-0.128	0.073	0.081	0.305	0.051	-0.061	0.274	0.672	0.035	0.672

Table S6: Expression data. Relative RNA expression changes, $x_i = [\text{RNA}_i]^{\text{pert}} / [\text{RNA}_i]^{\text{cont}} - 1$, for SOS test network genes in all perturbation experiments.

Genes	Training Perturbations									Test Perturbations	
	<i>recA</i>	<i>lexA</i>	<i>ssb</i>	<i>recF</i>	<i>dinI</i>	<i>umuDC</i>	<i>rpoD</i>	<i>rpoH</i>	<i>rpoS</i>	double	MMC
<i>recA</i>	0.128	0.107	0.080	0.112	0.057	0.077	0.057	0.104	0.098	0.174	0.177
<i>lexA</i>	0.092	0.180	0.075	0.088	0.067	0.078	0.058	0.120	0.109	0.240	0.158
<i>ssb</i>	0.071	0.102	0.677	0.089	0.060	0.104	0.057	0.095	0.076	0.118	0.115
<i>recF</i>	0.095	0.117	0.097	0.103	0.069	0.100	0.070	0.101	0.136	0.235	0.201
<i>dinI</i>	0.096	0.111	0.101	0.120	0.187	0.096	0.064	0.126	0.118	0.130	0.161
<i>umuDC</i>	0.095	0.113	0.094	0.116	0.102	0.271	0.064	0.078	0.096	0.162	0.248
<i>rpoD</i>	0.062	0.124	0.082	0.136	0.089	0.123	0.259	0.164	0.184	0.131	0.148
<i>rpoH</i>	0.063	0.104	0.103	0.086	0.055	0.091	0.059	3.607	0.120	0.183	0.212
<i>rpoS</i>	0.082	0.108	0.131	0.118	0.096	0.090	0.063	0.198	0.256	0.150	0.240

Table S7: Standard errors on expression data.

	Gene Perturbed in Training Perturbations									
	<i>recA</i>	<i>lexA</i>	<i>ssb</i>	<i>recF</i>	<i>dinI</i>	<i>umuDC</i>	<i>rpoD</i>	<i>rpoH</i>	<i>rpoS</i>	
Magnitude (u_i)	0.6529	1.1711	13.4120	1.6705	4.5415	2.3555	4.7083	12.8658	4.1089	
Std. Error (S_{u_i})	0.1752	0.2003	0.2884	0.6940	1.1635	0.5227	0.9566	3.1169	1.0454	

Table S8: Perturbations. Relative magnitude of perturbation delivered in each of the nine training experiments. $u_i = [\text{RNA}_i]^{\text{vec}} / [\text{RNA}_i]^{\text{cont}}$, for a perturbation to gene i . $[\text{RNA}_i]^{\text{vec}}$ is the concentration of gene i RNA synthesized from the overexpression vector pBADX53 in each training experiment.

References and Notes

1. F. M. Ausubel, *Current Protocols in Molecular Biology* (Wiley, New York, 1987).
2. J. Sambrook, E. F. Fritsch, T. Maniatis, *Molecular Cloning: A Laboratory Manual* (Cold Spring Harbor Laboratory Press, Plainview, NY, 1989).
3. I. Shmulevich, E. R. Dougherty, S. Kim, W. Zhang, *Bioinformatics* **18**, 261 (2002).
4. S. Liang, S. Fuhrman, R. Somogyi, *Proc. Pacific Symp. Biocomp.* **3**, 18 (1998).
5. A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, R. A. Young, *Proc. Pacific Symp. Biocomp.* **7**, 437 (2002).
6. A. Wagner, *Bioinformatics* **17**, 1183 (2001).
7. H. de Jong, *J. Comp. Biol.* **9**, 67 (2002).
8. P. Brazhnik, A. de la Fuente, P. Mendes, *Trends Biotechnol.* **20**, 467 (2002).
9. L. Ljung, *System Identification: Theory for the User* (Prentice Hall, Upper Saddle River, NJ, 1999).
10. Pacific Symposium on Biocomputing 1999, online proceedings, <http://psb.stanford.edu/psb-online/>.
11. M. K. S. Yeung, J. Tegnér, J. J. Collins, *Proc. Natl. Acad. Sci. U.S.A.* **99**, 6163 (2002).
12. Z. N. Oltvai, A. L. Barabási, *Science* **298**, 763 (2002).
13. J. Tegner, M. K. Yeung, J. Hasty, J. J. Collins, *Proc. Natl. Acad. Sci. U.S.A.* **100**, 5944 (2003).
14. D. Montgomery, E. A. Peck, G. G. Vining, *Introduction to Linear Regression Analysis* (John Wiley & Sons, Inc., New York, 2001).

15. W. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge University Press, Cambridge, UK, 1993).
16. A. J. Miller, *Subset Selection in Regression* (Chapman and Hall, London, 1990).
17. T. R. Hughes, et al., *Cell* **102**, 109 (2000).
18. Materials, methods and supporting data are available as supporting material on Science Online.
19. Clustering was performed using the European Bioinformatics Institute EPCLUST tool available at <http://www.ebi.ac.uk/microarray/ExpressionProfiler/ep.html>.